LY20-0897-7
File No. S370-36

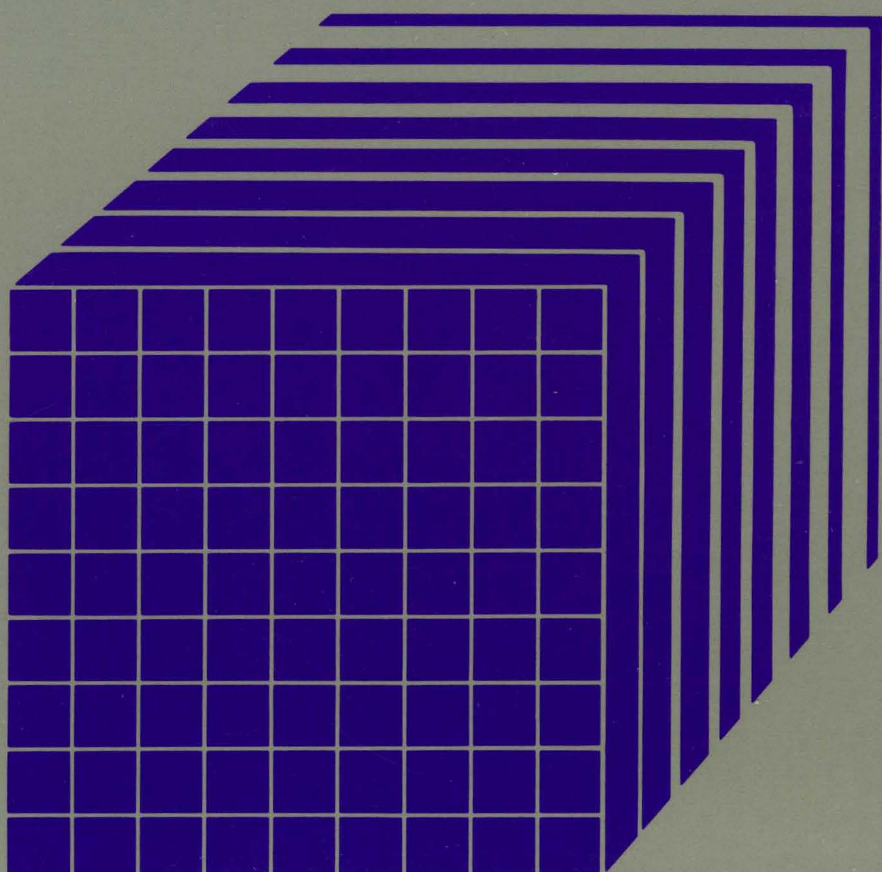
IBM

Virtual Machine/ System Product High Performance Option

System Logic and Problem Determination Guide–CP

Release 5

LY20-0897-7



LY20-0897-7
File No. S370-36

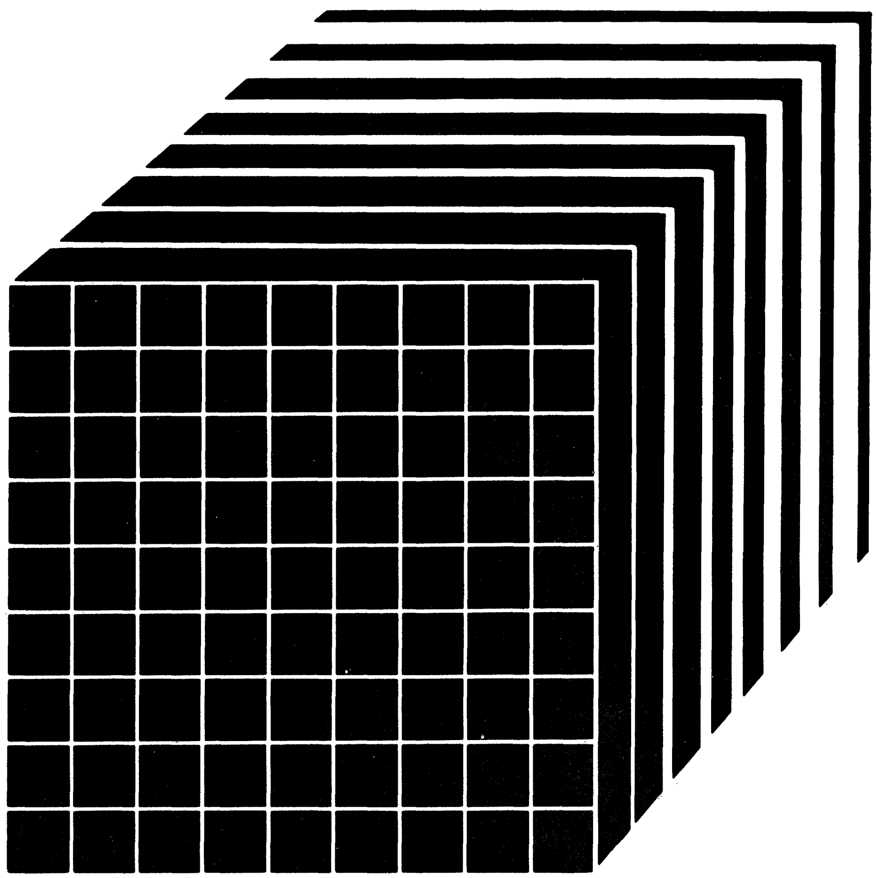
IBM

Virtual Machine/ System Product High Performance Option

System Logic and Problem Determination Guide—CP

Release 5

LY20-0897-7



The term "VM/SP High Performance Option" applies to the VM/SP High Performance Option Licensed Program when used in conjunction with the VM/System Product Licensed Program.

Eighth Edition (August 1987)

This is a major revision of LY20-0897-6. See "Summary of Changes" on page iii for the changes made to this manual. Technical changes or additions made to the text and illustrations for this release are indicated by a vertical line to the left of the change.

This edition applies to Release 5.0 of IBM Virtual Machine/System Product High Performance Option (Program Number 5664-173), and to later releases and modifications until otherwise indicated in new editions or Technical Newsletters. This (eighth) edition is a revision of the seventh edition (Release 4.2).

To order the previous edition that still applies to Release 4.2, use the following temporary order number:

Release 4.2 Seventh Edition LT00-1912

Changes are made periodically to the information herein; before using this publication to operate IBM systems, consult the latest *IBM System/370, 30xx, and 4300 Processors Bibliography*, GC20-0001, for the editions that are applicable and current.

Summary of Changes

Summary of Changes
for LY20-0897-7
as updated August 1987
for VM/SP HPO Release 5.0

SELECTION OF PAGES ON A SYSTEMWIDE "LEAST RECENTLY USED" BASIS

Changed: Programming Support

These changes will improve performance in several ways:

- Improving the memory management of large working sets, shared pages, and the <16 Mb area. The free list becomes the major source for page replenishment. The flush list will be deemphasized. Core table scan becomes the primary method for free list replenishment—the disposable page collector is eliminated.
- Streamlining the QDROP and QADD processes. When a virtual machine drops from queue, its pages will no longer be logically swapped and trimmed.
- Making Expanded Storage (called Paging Storage in this manual) more attractive as a swapping device.
- Preserving interactive response times in CMS intensive environments.
- Simplifying tuning.

SPOOL FILE LIMIT ENHANCEMENTS

Changed: Programming Support

The former limit of 9900 spool files on a system has been removed. With this change, there may exist up to 9900 spool files *for each user*. (Actually, the systemwide maximum is determined by the size of the checkpoint area. At present, this theoretical maximum would be over 100,000 for most systems. It is reasonable to assume that no system would ever approach that amount.)

Spool files will now have a user-unique spool ID as well as a systemwide ID. Reader spool file blocks (SFBLOKs) will now be kept in the virtual storage of a special userid, SYSSPOOL. Printer and punch SFBLOKs will remain in FREE storage.

In conjunction with these enhancements, the checkpoint/forced start process has been improved to reconstruct the spool files more rapidly.

A new module, DMKCKR, has been generated.

SCHEDULER ENHANCEMENT

Changed: Programming Support

With this enhancement, virtual machines are moved from the eligible list to the run list only if processor time and sufficient main storage are available. This will help eliminate storage overcommitment and reduce response times.

NEW 'NOQ2' OPTION ON THE 'SET QDROP OFF' COMMAND

New: Programming Support

Specify the SET QDROP OFF NOQ2 command for service virtual machines (like GCS/VSCS) that use system resources in small, frequent bursts. This command will keep that virtual machine in Q1, improving performance for the users of that service.

EXPANDED STORAGE (PAGING STORAGE) ENHANCEMENTS

New: Programming Support

A new macro, SYSXSTOR, is added. This macro controls the allocation of Paging Storage. You may continue to use SYSPAG to allocate Paging Storage, but SYSXSTOR offers these advantages:

- It is easier to use than SYSPAG
- For the 3090 Model 400, it allows you to generate Paging Storage greater than 64 increments.

A new module, DMKXST, has been generated.

AUTOMATIC REORDERING OF SYSPAG AREAS

New: Programming Support

A new parameter is added to the SYSPAG macro, ORDER=SYSTEM/USER. If ORDER=SYSTEM (the default) is coded, HPO will automatically order the devices allocated on the SYSPAG macro to decrease I/O contention by evenly distributing the I/O activity over the available I/O paths.

ERROR RECORDING ENHANCEMENTS

New: Programming Support

For a 3090 Processor, the channel check handler and machine check handler will now provide more information in the error records. Specifically, these records will now tell you whether or not the hardware error actually affected system performance.

NEW 'NOVF' PARAMETER ON THE 'OPTION' DIRECTORY ENTRY

New: Programming Support

Specifying the NOVF parameter on a user's OPTION control statement in the directory will deny that user access to the Vector Facility.

4381 PROCESSOR COMPLEX MODELS 11, 12, 13, AND 14

Changed: Hardware Support

The 4381 Processor Complex Models 1, 2, and 3 are replaced and extended by the Models 11, 12, 13, and 14.

3090 PROCESSOR COMPLEX MODELS 150, 180, AND 400

Changed: Hardware Support

VM/SP HPO now supports the 3090 Processor Complex Models 150, 180, and 400.

LOGICAL DEVICE HOST LIMIT RELIEF

Changed: Programming Support

This support removes the restriction that no more than 8 virtual machines can create and use logical devices concurrently. Now, any number of virtual machines can create up to 512 logical devices as long as the number of logical devices in the system does not exceed 4096.

3480 VOLUME SERIAL ERROR RECORDING

Changed: Programming Support

Users can now keep track of the error frequency for 3480 tapes by examining the volume serial in 3480 outboard records (OBR) or miscellaneous data records (MDR).

3422 MAGNETIC TAPE SUBSYSTEM

Changed: Programming Support

VM/SP HPO provides programming support for the 3422 Magnetic Tape Subsystem.

TRANSPARENT SERVICES ACCESS FACILITY (TSAF)

New: Programming Support

The transparent services access facility lets users connect to and communicate with local or remote virtual machines within a group of systems. This facility consists of the TSAF virtual machine component, APPC/VM, and two CP system services. APPC/VM is a modified subset of IUCV. With the TSAF virtual machine, it provides services within a single system and throughout a group of systems, unlike IUCV, which provides services only within a single system. The TSAF virtual machine component handles communication between systems by letting APPC/VM paths span more than one system. The QUERY CPTRAP command has also been added.

The following new modules have been generated: DMKCQC, DMKIDR, DMKIUB, DMKIUN, DMKIUP, DMKIUS, DMKTRX, DMKCRM.

NATIONAL LANGUAGES SUPPORT

New: Programming Support

VM/SP HPO now supports a variety of national languages for use in their native countries. Updates have been made to modules and data areas providing this support, specifically, those handling CP messages. The following new modules have been generated: DMKHVF, DMKMES, DMKVBM.

ALTERNATE NUCLEUS SUPPORT

New: Programming Support

Alternate nucleus support makes it easier to create and IPL backup copies of the CP nucleus when the primary nucleus is damaged or unavailable.

PRINTER SUPPORT ENHANCEMENTS

Changed: Programming Support

The printer support enhancements include the addition of a SPOOL system service facility which provides support for a printer subsystem. The DESTination option allows you to select a specific printer or punch to process your print, punch, or console file. Two new DIAGNOSE codes allow a user to specify additional information about a print file. The CMS PRINT command has been enhanced to support an OVersize option and a special carriage control character to allow a longer data line.

LOGON/LOGOFF ENHANCEMENTS

Changed: Programming Support

The LOGON/LOGOFF enhancements improve system availability to users and resolve the problem of conflicting messages during LOGOFF processing. A new module, DMKUSQ, has been generated.

ERROR LOGGING SYSTEM SERVICE

Changed: Programming Support

The error logging system service, a new CP system service, allows a virtual machine to receive a copy of all records currently written to the CP error recording area.

SPOOL FILE COMPRESSION SUPPORT ENHANCEMENT

Changed: Programming Support

An enhancement to SPOOL File Compression Support improves the reliability of transmitting spooled data between systems.

ASCII ENHANCEMENTS

Changed: Programming Support

Various enhancements have been made to the support of ASCII devices.

OTHER MODULE SPLITS

New: Programming Support

Modules(s) Being Split	**Resulting Modules**
DMKACR	DMKACR and DMKACS
DMKCCW	DMKCCW, DMKCCD, DMKCCF, DMKCCO, DMKCCS, and DMKCCT
DMKCNS	DMKCNS and DMKCNT
DMKCPT	DMKCPT and DMKCPN
DMKCQH	DMKCQH and DMKCPI
DMKCSU	DMKCSU and DMKCSW
DMKCSV	DMKCSV, DMKCSX, and DMKCSY
DMKDIA	DMKDIA and DMKDIF
DMKGRF	DMKGRF, DMKGRD, DMKGRE, DMKGRF, and DMKGRI
DMKIUA, DMKIUC, and DMKIUE	DMKIUA, DMKIUC, DMKIUE, DMKIUB, DMKIUN, DMKIUP, and DMKIUS
DMKQCN	DMKQCN and DMKQCQ
DMKSVC	DMKSVC and DMKSVD
DMKTRT and DMKTRU	DMKTRT, DMKTRU, and DMKTRX
DMKTTY	DMKTTY and DMKTTX
DMKVCP	DMKVCP and DMKVCU
DMKVDD	DMKVDD and DMKVDF

DOCUMENTATION CHANGES

Minor editorial and technical changes have been made throughout this publication.

Summary of Changes for LY20-0897-6 for VM/SP HPO Release 4.2

AUTO-DEACTIVATION OF RESTRICTED PASSWORDS AND DIRECTORY ENHANCEMENTS

New: Programming Support

Adds support to enhance system integrity by minimizing the exposure of unauthorized system access through the use of restricted passwords. The directory enhancements removes the restriction on the number of USER entries that can be defined in the directory. Also, directory PROFILE support provides a means by which installations can optimize the number of commonly repeated control statements in USER entries in the source directory.

ACCESS VERIFICATION ROUTINES

New: Programming Support

While VM/SP HPO provides many security functions, added support for access verification routines provides a standard interface to the RACF/VM Support PRPQ or user-written routines that can provide a higher level of security. Although the access verification routines support does not by itself provide security functions, it allows you to install software that does.

For example, to increase security of minidisk accesses, logon passwords, and movement of spool files, you can install access verification routines with the Resource Access Control Facility (RACF) (Program Number 5740-XXH) and RACF/VM Support PRPQ (Program Number 5767-002).

VECTOR FACILITY

New: Hardware Support

Support is provided for the Vector Facility in System/370 mode configured to a 3090 Processor. The Vector Facility is a synchronous vector/scalar instruction processor that can manipulate values (usually floating-point) at a high speed. Compiled engineer and scientific FORTRAN applications can use the array processing capability of the Vector Facility. VM/HPO supports multiple virtual machines' use of this facility.

PAGE MIGRATION

Changed: Programming support

Page migration is changed to select pages (rather than segments) for migration on a reference basis instead of by time-stamp (age basis). Also, pages are migrated down the demand page hierarchy, instead of being migrated directly to the pre-allocated migration area. This improves the time required to retrieve those pages that become active in the near future.

Because migration of swap tables is sometimes necessary even when page migration is not actively moving pages, swap table migration is now invoked independently of page migration (rather than after page migration). Swap table migration is further improved by migrating swap tables regardless of whether all the pages in the segment have been migrated.

DOCUMENTATION CHANGES

Minor technical and editorial changes have been made throughout this publication.

Summary of Changes for LY20-0897-5 for VM/SP HPO Release 4

Note: Release 4 does not support 3090 processors. 3090 processors are supported by Release 3 Modification 6. For information on Release 3 Modification 6, order the manual using the pseudo number shown above.

GROUP CONTROL SYSTEM (VM/SP HPO GCS)

New: Programming Support

This new component of VM/SP HPO is a virtual machine supervisor that provides simulated MVS services and supports a multitasking environment. For more information on the Group Control System (GCS), refer to the *VM/SP Group Control System Guide*. A new module, DMKVMG, has been generated.

STAND-ALONE DUMP

Changed: Programming Support

The Stand-alone Dump facility is an enhancement to VM/SP HPO Serviceability. It provides the support personnel with capability of dumping up to 16 megabytes of real storage. It is required to dump real storage when VM/SP HPO cannot create a CP abend dump. A new module, DMKSAD, has been generated.

CP FRET TRAP

New: Programming Support

The CP FRET trap can be used as an aid in solving problems caused by improper use of CP storage and to solve many storage overlay problems.

VMDUMP ENHANCEMENTS

Changed: Programming Support

DIAGNOSE Code X'94' is available to allow a virtual machine to request dumping of its virtual storage. Also, the three address range restriction has been removed from the VMDUMP command. A new module, DMKVME, has been generated.

SHARED/NONSHARED RESTRICTION

New: Programming Support

With the addition of this support, any attempt to construct a virtual device configuration that would mix SHARED and NONSHARED device types on the same virtual control unit is rejected.

IUCV Enhancements

Changed: Programming Support

The Inter-User Communications Vehicle (IUCV) has been enhanced to support data movement from discontiguous buffers on the SEND, RECEIVE, and REPLY functions. The IUCV macro has been updated to handle a "BUFLIST=" parameter on the SEND and RECEIVE functions, and an "ANSLIST=" parameter on the SEND and REPLY functions. New modules generated are DMKVCQ and DMKVCS.

CPTRAP ENHANCEMENTS

Changed: Programming Support

CPTRAP is a service aid used in problem determination. Enhancements to the CPTRAP command provide two additional functions, GROUPID and WRAP, and one additional entry type, X'3D'.

Enhancements to TRAPRED makes reviewing the trap data easier by providing more selectivity for X'3D', X'3E', and X'3F' entries and by providing a way to display formatted output of the trapped data. A new module, DMKTRU, has been generated.

TERMINAL ENHANCEMENTS

New: Programming Support

The DIAL command is now supported for SNA devices. A new module, DMKRGE, has been generated.

EXPANSION OF USER CLASSES

Changed: Programming Support

The user class structure has been modified such that the user may now define up to 32 privilege classes, beyond (or in place of) the seven IBM defined privilege classes.

REMOTE SPOOLING COMMUNICATIONS SUBSYSTEM NETWORKING VERSION 2

Changed: Programming Support

With the release of the Remote Spooling Communications Subsystem Networking Version 2 program product (5664-188), any reference to RSCS in this manual applies to RSCS Version 2. Information about RSCS can be found in the *VM/SP Remote Spooling Communications Subsystem Version 2 General Information.*

VM/SP HPO 3800 MODEL 3 COMPATIBILITY SUPPORT

New: Hardware Support

Compatibility support allows VM/SP users to access the 3800 Model 3 Printing Subsystem. Existing programs designed to produce 3800 Model 1 printer output may produce output for the 3800 Model 3 printer with little or no program change. Use of this support provides improved print quality (240 x 240 pel resolution) and the addition of a 10 lines-per-inch (LPI) vertical space option. The following new modules have been generated: DMKRSF (split from DMKRSE), DMKSEQ (split from DMKSEP), DMKVSX (split from DMKVSP).

3480 MAGNETIC TAPE SUBSYSTEM SUPPORT

New: Hardware Support

The 3480 is a buffered magnetic subsystem consisting of one control unit which can address up to 8 drives, or two control units which can each address up to 16 drives.

3370 DIRECT ACCESS STORAGE MODELS A2 AND B2

New: Hardware Support

Support is added for the 3370 Direct Access Storage Models A2 and B2. A new module, DMKVDB, is generated.

DMKFRE/DMKFRT SPLIT

Changed: Programming Support

The module DMKFRE has been split into two modules, DMKFRE and DMKFRT. DMKFRE handles all requests for free storage and calls to DMKFRET to release free storage. DMKFRT handles all requests to return free storage.

DMKCKP SPLIT

Changed: Programming Support

The module DMKCKP has been split into six modules: DMKCKD, DMKCKF, DMKCKH, DMKCKM, DMKCKN, and DMKCKP.

- DMKCKD is called by DMKCKP and issues HALT I/O instructions to system generated devices and drains existing I/O interrupts. It also finds a usable path to a desired device and executes all processor switch request.

- DMKCKF is called by DMCKP and saves vital system data on the system's warm start cylinders.

- DMKCKH is called by many different routines and contains several utility routines used by the checkpoint program. These routines do tasks such as checking the validity of the warm start cylinders, handling record buffers, saving system spool files, and handling I/O interrupts.

- DMKCKM is called by many different routines and saves selected virtual machines during system shutdown or abend. It also displays messages for the checkpoint program.

- DMKCKN is called by DMKCKM and handles I/O requests for the checkpoint program. It also preprocesses various fields in RDEVBLOKs so that they can be used by DMKCKN and DMKCKM.

- DMKCKP is the main driver for the checkpoint program. It contains the IPL entry point for system initialization, system shutdown, and system abend. It also displays error messages and loads disabled wait state PSWs.

DMKCPI/DMKCPS REORGANIZATION

Changed: Programming Support

The logic of DMKCPI and DMKCPJ has been reorganized to create six new modules: DMKALO, DMKIDU, DMKMNT, DMKMNT, DMKOPE, DMKSEG, DMKTOD.

- DMKALO is called by DMKMNT and builds the system ALOCBLOKs for CP-owned volumes that are online at IPL.

- DMKIDU is called by DMKCPJ and locates DASD space for and allocates the CP DUMP if DASD space is found. It also clears TDISK space at IPL.

- DMKMNT is called by DMKCPI and checks all devices in DMKRIO and marks them online if the device is operational.

- DMKOPE is called by DMKCPI to locate the system operator's console and to disconnect the operator on a system restart if the operator was disconnected before the system abend. It is also called by DMKCPJ to start the auto MONITOR, to autolog users, and to logon the system operator.

- DMKSEG is called by DMKCPI to initialize CP's PAGE, SEGMENT, CORE, and SWAP tables. It is also called by DMKCPJ to page in CP pageable modules and to write the symbol table (DMKSYM) and DMKVMI to DASD.

- DMKTOD is called by DMKCPT to initialize the TOD clock.

3880 STORAGE SUBSYSTEM MODEL 21 SUPPORT

New: Hardware Support

VM/SP HPO now supports the 3880 Storage Subsystem Model 21. The 3880 Model 21 is designed as a high-performance paging and swapping subsystem. It is an enhancement over the 3880 Model 11. In addition, the 3880 Storage Subsystem Model 11 is supported for swapping.

4381 MODEL 3

New: Hardware Support

VM/SP HPO now supports the 4381 Processor Model 3 (dyadic). ECPS:VM/370 level 22 specifically supports the 4381 Model 3. It includes:

- Assists for VM/SP HPO free storage and dispatching algorithms and a CP assist for DIAGNOSE code '18' (Disk I/O).

- Enhancements to existing assists to support the extended real addressing facility of VM/SP HPO.

- Enhancements to existing assists to support a multiprocessor system.

OTHER MODULE SPLITS

New: Programming Support

Modules(s) Being Split	Resulting Modules
DMKCPU	DMKCPM and DMKCPU
DMKCSO	DMKCPM and DMKCPU
DMKCQP	DMKCPM and DMKCPU
DMKEMA, DMKEMB, and DMKEMC	DMKEMA, DMKEMB, DMKEMC, DMKEMD, and DMKEME
DMKIOS	DMKIOQ and DMKIOS
DMKLOG	DMKLOG and DMKLOJ
DMKLOH	DMKLOH and DMKLOM
DMKSST	DMKSST and DMKSSV
DMKTAP	DMKTAP and DMKTAQ
DMKVCA	DMKVCA and DMKVCB

Modules(s) Being Split	Resulting Modules
DMKVDG	DMKVDG and DMKVDH
DMKVRR	DMKVRR and DMKVRS
DMKPTS	DMKPTS and DMKPTT

DOCUMENTATION CHANGES

Minor technical and editorial changes have been made throughout this publication.

Preface

Purpose

This manual provides information needed to analyze problems that may occur on the IBM Virtual Machine/System Product: High Performance Option (VM/SP High Performance Option) when used with VM/System Product Release 5. The descriptions are general and serve as a guide in understanding VM/SP High Performance Option. They are not meant to be a detailed analysis of VM/SP High Performance Option programming and cannot be used as such.

Audience

This manual is intended for persons who are analyzing errors to the Control Program (CP) component of VM/SP High Performance Option. Persons performing this task could be:

- IBM support personnel
- Customer systems programmers.

Organization

This manual contains a logic description for the designated component. It is divided into four parts: Introduction, CP Method of Operation and Program Organization, CP Directories, and CP Diagnostic Aids.

CP Method of Operation and Program Organization contains the functions and relationships of the program routines in VM/SP High Performance Option.

CP Directories contains descriptions of all the assemble modules in CP. It also contains extensive cross-references between modules and labels within a VM/SP High Performance Option component.

CP Diagnostic Aids contains additional information useful for determining the cause of a problem.

Appendix A contains a description of VM/370 Extended Control-Program Support (ECPS:VM/370).

Appendix B describes support for the IBM 3850 Mass Storage System (MSS).

Appendix C contains information about MVS/System Extensions and MVS/System Product Support.

Appendix D describes in detail the access verification routines (DMKRPW, DMKRPI, and DMKRPD).

Related Information

Information on the Conversational Monitor System (CMS) is contained in *System Logic and Problem Determination Guide Volume 2 (CMS)*, Order No. LT00-1604.

Information on the Remote Spooling Communications Subsystem (RSCS), a VM/370 Release 6 component, is contained in *VM/370 System Logic and Problem Determination Guide, Volume 3 Remote Spooling Communications Subsystem (RSCS),* Order No. SY20-0888.

The control blocks supporting the RSCS Logic are contained in *VM/SP Data Areas and Control Block Logic- Volume 2 (CMS),* Order No. LT00-1606.

Logic Information on the Interactive Problem Control System (IPCS), a VM/370 Release 6 component, is totally contained in *VM/SP HPO Service Routines Program Logic,* Order No. LY20-0898.

For information on the Group Control System, see the *VM/SP Group Control System Guide*, Order No. SC24-5249.

For information on VM/VTAM, see the *ACF/VTAM Version 3 General Information (for VM/SP)*, Order No. GC30-3246.

How to Use This Manual

- Isolate the component of VM/SP High Performance Option in which the problem occurred.
- Use the list of restrictions in *VM/SP HPO Planning Guide and Reference* to be certain that the operation that was being performed was valid.
- Use the directories and use *VM/SP High Performance Option Data Areas and Control Block Logic – CP* to help you to isolate the problem.
- Use the "CP Method of Operation and Program Organization" sections, if necessary, to understand the operation that was being performed.

Device Terminology

For terms in this manual that are unfamiliar to you, refer to *IBM Vocabulary for Data Processing, Telecommunications, and Office Systems*, GC20-1699.

For device terminology that may be specific to VM/SP High Performance Option, refer to *VM/SP HPO Library Guide.*

Contents

Figures

Introduction

This part contains the following information:

- Introduction to the Control Program (CP)
- Program States
- Using Processor Resources
- Functional Information
- Performance Guidelines
- Interrupt Handling.

The Control Program (CP)

The VM/SP High Performance Option (VM/SP HPO) Control Program manages the resources of a single computer so that multiple computing systems appear to exist. Each "virtual" computing system, or virtual machine, is the functional equivalent of a single processor IBM System/370.

A virtual machine is configured by recording appropriate information in a directory called the *system directory*. The virtual machine configuration includes counterparts of the components of a real IBM System/370:

- A virtual operator's console
- Virtual storage
- A virtual processor
- Virtual I/O devices.

CP makes these components appear real to whichever operating system is controlling the work flow of the virtual machine.

The virtual machines operate concurrently via multiprogramming techniques. CP overlaps the idle time of one virtual machine with execution in another.

Each virtual machine is managed at two levels:

- An operating system manages the work the virtual machine is to do.
- The control program (CP) manages the concurrent execution of multiple virtual machines.

Some system functions perform differently when running in attached processor mode or multiprocessor mode. For a description of the additional processing performed when in attached processor mode, see "Virtual Machine Storage Management" on page 5.

Introduction to the Control Program

A virtual machine is created for a user when he logs on VM/SP HPO, on the basis of information stored in his system directory entry. The entry for each user identification includes a list of the virtual input/output devices associated with the particular virtual machine.

Additional information about the virtual machine that also is kept in the directory entry includes:

- A list of the virtual I/O devices associated with that virtual machine
- The command privilege class
- Accounting data
- Normal and maximum virtual storage sizes
- Dispatching priority
- Optional virtual machine characteristics, such as extended control mode.

The Control Program supervises the execution of virtual machines by (1) permitting only problem state execution except in its own routines, and (2) receiving control after all real computing system interrupts. Generally, CP intercepts each privileged instruction[1] and simulates it if the current program status word of the issuing virtual machine indicates a virtual supervisor state; if the virtual machine is executing in virtual problem state, the attempt to execute the privileged instruction is reflected to the virtual machine as a program interrupt. All virtual machine interrupts (including those caused by attempting privileged instructions) are first handled by CP, and are reflected to the virtual machine if an analogous interrupt would have occurred on a real machine.

Virtual Machine Time Management

The real processor simulates multiple virtual processors. Virtual machines that are executing in a conversational manner are given access to the real processor more frequently than those that are not; these conversational machines are assigned the smaller of two possible time slices. CP determines execution characteristics of a virtual machine at the end of each time slice on the basis of the recent frequency of its console requests or terminal interrupts. The virtual machine is queued for subsequent processor utilization according to whether or not it is a conversational user of system resources.

A virtual machine can gain control of the processor only if it is not waiting for some activity or resource. The virtual machine itself may enter a virtual wait state after an input/output operation has begun. The virtual machine cannot gain control of the real processor if it is waiting for a page of storage, if it is waiting for an input/output operation to be translated and started, or if it is waiting for a CP command to finish execution.

A virtual machine can be assigned a priority of execution. Priority is a parameter affecting the execution of a particular virtual machine as compared with other virtual machines that have the same general execution characteristics. It is a parameter in the virtual machine's directory entry. The system operator can reset the value with the privilege class A SET command.

---

1 When preferred machine assist is active, the MVS virtual machine runs in the real supervisor state in the V=R area. Also, when an MVS virtual machine runs in single processor mode and preferred machine assist in an AP, MP, or dyadic processor, it runs in the supervisor state.

Virtual Machine Storage Management

The normal and maximum storage sizes of a virtual machine are defined as part of the virtual machine configuration in the directory. The user can redefine virtual storage size to any value that is a multiple of 4K and not greater than the maximum defined value. VM/SP HPO implements this storage as virtual storage. The storage may appear as paged or not paged to the virtual machine, depending upon whether or not the extended control mode option was specified for that virtual machine. This option is required if operating systems that control virtual storage, such as OS/VS1, VM/SP HPO, or VM/370, are run in the virtual machine.

Storage in the virtual machine is logically divided into 4096-byte areas called *pages*. Contiguous 64K areas of virtual storage are called *segments*. A complete set of segment and page tables is used to describe the storage of each virtual machine. These tables are updated by CP and reflect the allocation of virtual storage pages to blocks of real storage. These page and segment tables allow virtual storage addressing in a System/370 machine. Storage in the real machine is logically and physically divided into 4096-byte areas called *page frames*.

If the virtual machine is executing in extended control mode with translate on, then two additional sets of segment and page tables are kept. The virtual machine operating system is responsible for mapping the virtual storage created by it to the storage of the virtual machine. CP uses this set of tables in conjunction with the page and segment tables created for the virtual machine at logon time to build *shadow* page tables for the virtual machine. These shadow tables map the virtual storage created by the virtual machine operating system to the storage of the real computing system. The tables created by the virtual machine operating system may describe any page and segment size permissible in the IBM System/370.

When there is a shortage of real storage available, CP keeps only referenced virtual storage pages in real storage. CP can bring pages into any available page frames. During program execution, a combination of CP and dynamic address translation on the System/370 relocates a page. (During relocation, new absolute addresses are assigned to a page so that the program can execute in the assigned area of real storage.) The active pages from all logged-on virtual machines and from the pageable routines of CP compete for available page frames. When the number of page frames available for allocation falls below a threshold value, CP determines which virtual storage pages currently allocated to real storage are relatively inactive, and starts page-out operations for them.

Inactive pages are kept on a direct access storage device (DASD). DASD space is managed so that only one copy of a given page exists at one time. CP assigns certain inactive pages to a paging device according to the following rules:

- If the page was referenced during virtual machine execution, and if space is available in a TYPE=SW area, the page is assigned to that area (regardless of whether or not the page was changed during virtual machine execution).

- If the page was referenced and changed during virtual machine execution, and no space is available in a TYPE=SW area, the page is assigned a TYPE=PP area.

- If the page was changed during virtual machine execution but not referenced, it is assigned to a TYPE=PP area.

Note: If a page is read in from TYPE=SW, it is marked as "changed" because the backup copy in the TYPE=SW area is released once the page is read in.

(See the heading "DASD and Paging Storage Management" for more information.)

A virtual machine program can use the DIAGNOSE instruction to tell CP that the information from specific pages of virtual storage is no longer needed. CP then releases the DASD areas that were assigned to hold the specified pages.

CP can prepage a number of virtual machine pages that were resident and referenced at the last queue drop. It prepages when a virtual machine is added to queue.

When a page fault is resolved for a page in a swap set, all the pages in the swap set are paged in. While CP is paging for one virtual machine, another virtual machine can be executing. Any paging operation started by CP is transparent to the virtual machine.

Storage and Processor Utilization

The system operator can assign reserved page frames to multiple virtual machines by using the SET RESERVE command. This command assigns to a virtual machine a specific amount of storage from the real machine, enabling the specified number of a virtual machine's active pages to remain in real storage. CP will dynamically build up a set of reserved real storage page frames for a virtual machine during its execution until the maximum number of reserved frames is reached. The page frames that are reserved are defined by core table scan, so that referenced pages are more likely to be held in storage than unreferenced pages. See the *VM/SP HPO Operator's Guide* for more information on the SET RESERVE command.

During CP system generation, the installation may specify an option called virtual=real. With this option, the virtual machine's storage is allocated directly from real storage at the time the virtual machine logs on (if it has the VIRT=REAL option in its directory). All pages except page zero are allocated to the corresponding real storage locations. In order to control the real computing system, real page zero must be controlled by CP. Consequently, the real storage size must be large enough to accommodate the CP nucleus, the entire virtual=real virtual machine, and the remaining pageable storage requirements of CP and the other virtual machines.

The virtual=real option improves performance in the selected virtual machine since it removes the need for CP paging operations for the selected virtual machine. The virtual=real option is necessary whenever programs that contain dynamically modified channel programs (except those of OS ISAM and OS/VS TCAM Level 5) are to execute under control of CP. For additional information on running systems with dynamically modified channel programs, see the *VM/SP HPO CP for System Programming*.

Virtual Storage Preservation

CP tries to preserve the contents of a virtual machine if the system operator forces the machine off the system, if the system abnormally terminates the machine, or if the system itself abnormally terminates.

At system generation time, you can specify which virtual machines are to be saved. The contents of these virtual machines are written out and saved in DASD space that must be previously allocated during system generation; the sequence in which virtual machines are saved can also be established. If a sequence for saving systems is not defined, then the systems are saved in the order in which virtual storage preservation was invoked for each. After the user logs on to the system again, the saved DASD area is restored by issuing the IPL command, specifying the name of the defined DASD area. System generation parameters also allow another designated user to IPL the named SAVESYS area.

Either the VMSAVE directory option or the SET VMSAVE command may be used for saving the contents of a specific virtual machine. The VMSAVE facility can be nullified by SET VMSAVE OFF, SYSTEM CLEAR, DEFINE STORAGE, or normal LOGOFF.

The V=R area (if active) of the real machine is preserved if the system is performing a warm start. The V=R area is cleared if the system terminates to a hard wait state or if a different V=R user logs on.

You can specify multiple VMSAVE target areas (areas in which the virtual machine is to be saved) for a single user; you do this by including in the DMKSNT module more than one NAMESYS macro with the same USERID= operand. Different target areas are required if a user wishes to IPL a VMSAVE system and have the VMSAVE option enabled at the same time. Once the VMSAVE is enabled, the IPL command cannot refer to the area until a recovery operation has taken place. Similarly, if a VMSAVE area currently contains a saved system, it can be released only by the user who caused the system to be stored there. Until the user releases that area, no other user can use it as a VMSAVE target area.

For more information on the VMSAVE facility, refer to *VM/SP HPO Planning Guide and Reference* and *VM/SP HPO CP for System Programming*.

MVS/SP V=R Virtual Machine Recovery

When the system performs an automatic warm start due to an abnormal termination, CP recovers the MVS/SP V=R virtual machine. VM/SP HPO saves the status of the MVS/SP virtual machine environment after CP software ABENDs. MVS/SP virtual machine recovery restores pending interrupts, dedicated I/O devices, I/O control blocks, real storage, and SPMODE (including AP/MP support if active). All open spool files are closed and checkpointed as in a normal system shutdown.

The recovered V=R machine will have its **original** directory characteristics. For example, if a different storage size was defined after logon, the original directory storage size will be in effect after recovery.

When recovery is completed, the MVS/SP virtual machine immediately resumes execution. However, the virtual machine is in a disconnected state.

CP saves the MVS virtual machine after an ABEND if the following are true:

- Dump to disk was specified using the SET DUMP AUTO command
- NOTRANS is set ON in the V=R machine
- The MVS/SP V=R machine is logged on at system ABEND
- Module DMKVRR is present in the system.

MVS/SP virtual machine recovery is not possible if CP is unable to dump, checkpoint, and re-IPL the system. If the system operator requests a dump using RESTART, recovery may be only partially successful. IUCV, VMCF, and SNA applications are not recovered for the MVS/SP virtual machine operating system.

IBM recommends that the guest operating system operator's console be separate from the virtual machine operator's console so that no console output is lost. All devices the guest virtual machine uses except virtual console and virtual spool devices should be dedicated devices with real addresses corresponding to virtual addresses. I/O interrupt data is saved only for dedicated devices.

Virtual Machine I/O Management

A real disk device can be shared among multiple virtual machines. Virtual device sharing is specified in the directory entry or by a user command. If specified by the user, an appropriate password must be supplied before gaining access to the virtual device. A particular virtual machine may be assigned read-only or read/write access to a shared disk device. CP checks each virtual machine input/output operation against the parameters in the virtual machine configuration to ensure device integrity.

Virtual Reserve/Release support can be used to further enhance device integrity for data on shared *minidisks.* Reserve/Release operation codes are simulated on a virtual basis for minidisks, including full-extent minidisks. For details on Reserve/Release support, refer to "Reserve/Release" under "Scheduling I/O Requests" in this part.

The virtual machine operating system is responsible for the operation of all virtual devices associated with it. These virtual devices may be defined in the directory entry of the virtual machine, or they may be attached to (or detached from) the virtual machine's configuration, dynamically, for the duration of the terminal session. Virtual devices may be dedicated, as when mapped to a fully equivalent real device; shared, as when mapped to a minidisk or when specified as a shared virtual device; or spooled by CP to intermediate direct access storage.

There is a limit to the number of virtual devices that can be defined for a virtual machine. This limit is established by the MAXDEV xxxx option in the machine's directory. The maximum limit is obtained from the formula:

Maximum Device Limit = 7FFF/VDEVSIZE + 1

There is, however, one restriction: the device limit for a BC mode virtual machine cannot be greater than 1536.

If the MAXDEV xxxx option is not specified in a machine's directory, the device limit is 410.

Consult *VM/SP HPO CP for System Programming* for current values of VDEVSIZE and the maximum and default device limits.

In a real machine running under control of OS, input/output operations are normally initiated when a problem program requests OS to issue a START I/O instruction to a specific device. Device error recovery is handled by the operating system. In a virtual machine, OS can perform these same functions, but the device address specified and the storage locations referenced will both be virtual. It is the responsibility of CP to translate the virtual specifications to real. Using the Diagnose '98' command, a virtual machine can execute its own real channel programs. See *VM/SP HPO CP for System Programming* for more information.

In attached processor or multiprocessor environments, virtual I/O can be initiated by either processor; in attached processor systems, all real I/O requests must be executed by the main processor and all I/O interrupts must be received on the main processor (the processor with I/O capability). Any I/O requests by the attached processor (the processor without I/O capability) are transferred to the main processor. In a multiprocessor system, real I/O can be handled by both processors as both processors have I/O capability.

In addition, the interrupts caused by the input/output operation (including channel errors) are reflected to the virtual machine for its interpretation and processing. If input/output errors occur, CP records them but does not initiate error recovery operations. The virtual machine operating system must handle error recovery, but does not record the error (if SVC 76 is used).

Input/output operations initiated by CP for its own purposes (paging and spooling) are performed directly and are not subject to translation.

See Appendix B for an explanation of additional processing when the virtual I/O request results in a real I/O request to an MSS 3330V volume.

Dedicated Channels

In most cases, the I/O devices and control units on a channel are shared among many virtual machines as minidisks and dedicated devices, and shared with CP system functions such as paging and spooling. Because of this sharing, CP has to schedule all the I/O requests to achieve a balance between virtual machines. In addition, CP must reflect the results of the subsequent I/O interruption to the appropriate storage areas of each virtual machine.

By specifying a dedicated channel (or channels) for a virtual machine via the Class B ATTACH CHANNEL command, the CP channel scheduling function is bypassed for that virtual machine. A virtual machine assigned a dedicated channel has that channel and all of its devices for its own exclusive use. CP translates the virtual storage locations specified in channel commands to real locations and performs any necessary paging operations, but does not perform any device address translations. The virtual device addresses on the dedicated channel must match the real device addresses; thus, a minidisk cannot be used.

Spooling Functions

A virtual unit record device that is mapped directly to a real unit record device is said to be dedicated. The real device is then controlled completely by the virtual machine's operating system.

CP facilities allow multiple virtual machines to share unit record devices. Since virtual machines controlled by CMS ordinarily have modest requirements for unit record input/output devices, such device sharing is advantageous, and it is the standard mode of system operation.

Spooling operations cease if the direct access storage space assigned to spooling is exhausted, and the virtual unit record devices appear in a not-ready status. The spooling operator may make additional spooling space available by using the class D SPTAPE command to dump output spool files to tape. He can also use the SPTAPE command to retrieve those files from the tape for output processing when spooling space requirements are not critical. See the description of the SPTAPE command in the *VM/SP HPO Operator's Guide* for further information. In an extreme situation, the system operator may make additional spooling space available by purging existing spool files or by assigning additional direct access storage space to the spooling function.

Specific files can be transferred from the spooled card punch or printer of a virtual machine to the card reader of the same or another virtual machine. Files transferred between virtual unit record devices by the spooling routines are not physically punched or printed. With this method, files can be made available to multiple virtual machines, or to different operating systems executing at different times in the same virtual machine.

CP spooling includes options for the virtual machine user and the real machine operator. These options include printing multiple copies of a single spool file, backspacing any number of printer pages, and defining spooling forms and classes and for the scheduling of real output. Each output spool file has associated with it a 136-byte area known as the spool file tag. The information contained in this area and its syntax are determined by the originator and receiver of the file. Both programs expect to find the destination identification in the file tag. Tag data is set, changed, and queried using the CP TAG command.

It is possible to spool terminal input and output. All data sent to the terminal, whether it be from the virtual machine, the control program, or the virtual machine operator, can be spooled. Spooling is particularly desirable when a virtual machine is run with its console disconnected. Console spooling is usually started via the command

```
SPOOL CONSOLE START
```

An exception to this is when a system operator logs on using a graphics device. In this instance, console spooling is automatically started and continues in effect even if the system operator should disconnect from the graphics device and log on to a nongraphic device. In order to stop automatic console spooling, the system operator must issue the command

```
SPOOL CONSOLE STOP
```

Spool File Recovery

If the system should suffer an abnormal termination, there are three degrees of recovery for the system spool files: warm start (WARM), checkpoint start (CKPT), and force start (FORCE). Warm start is automatically invoked if SET DUMP AUTO is in effect. Otherwise, the choice of recovery method is selected when the following message is issued:

```
Start ((WARM|CKPT|FORCE|COLD)(DRAIN))|(SHUTDOWN):
```

Note that a cold (COLD) start does not recover any spool files.

Warm Start

After a system failure, the warm start procedure copies the following data to the warm start area on an auxiliary DASD:

- Print spool files
- Punch spool files
- Open reader spool files
- Reader hash table (plus extension pages)
- SYSSPOOL's virtual storage
- Accounting data
- System message data.

When the system is reloaded, this information is retrieved and restored to its original status. If the warm start procedure cannot be implemented, because certain required areas of storage are invalid, the operator is notified to take other recovery procedures.

Checkpoint Start

Any new or revised status of spool file blocks, spooling devices, and spool hold queue blocks is dynamically copied to checkpoint area on an auxiliary DASD as it occurs. When a checkpoint (CKPT) start is requested, the information is used to re-create the users' spool file chains. It differs from warm start data in that only spool file data is restored—accounting and system messages information is not recovered. Also, the order of spool files on any particular restored chain is not the original sequence but a random one.

Force Start

A force start is required when checkpoint start encounters I/O errors while reading files, or invalid data. The procedure is the same as for checkpoint start except that unreadable or invalid files are erased. They cannot be recovered.

CP Commands

The CP commands allow you to control the virtual machine from the terminal, much as an operator controls a real machine. Each CP command is defined by a COMMD macro entry in module DMKCFC. Entries for logged-on users are placed beyond label COMNBEG1. Module DMKCMD also contains COMMD macro entries for subcommands. The COMMD macro has parameters defining command or subcommand name, class, type, entry point label, and the label of valid subcommands in DMKCMD.

Virtual machine execution can be stopped at any time by use of the terminal's attention key (for 3066 and 3270 terminals, the ENTER key is used); it can be restarted by entering the appropriate CP command. External, attention, and device ready interrupts can be simulated on the virtual machine. Virtual storage and virtual machine registers can be inspected and modified, as can status words such as the PSW and the CSW. Extensive trace facilities are provided for the virtual machine, as well as a single-instruction mode. Commands are available to invoke the spooling and disk sharing functions of CP.

CP commands are classified by privilege classes. The directory entry for each user assigns one or more privilege classes. The IBM-supplied classes are primary system operator (class A), system resource operator (class B), system programmer (class C), spooling operator (class D), system analyst (class E), service representative (class F), and general user (class G). Commands in the system analyst class may be used to inspect real storage locations, but may not be used to make modifications to real storage. Commands in the operator class provide real resource control capabilities. System operator commands include all commands related to virtual machine performance options, such as assigning a set of reserved page frames to a selected virtual machine. For descriptions of all the CP commands, see the *VM/SP HPO CP Command Reference* and the *VM/SP HPO Operator's Guide.*

You can extend the eight IBM-defined classes to up to 32 classes by adding an optional Class Control statement to the Directory. See *VM/SP HPO CP for System Programming* and the *VM/SP HPO Planning Guide and Reference* for more information.

CP Messages/National Language Support

CP dynamically builds some of its messages in modules; these messages are then issued directly from the modules. CP, however, issues most of its messages from a single static repository file.

CP dynamically creates a table for each language that a user sets. These tables determine the page of the message repository that contains the text associated with a particular message. Each table contains the range of message IDs on each page of the repository and the corresponding virtual address for that page.

The LANGBLOK contains the message table for a particular language. One LANGBLOK exists for each language, and these are chained together. A user's VMBLOK points to the LANGBLOK for the language set for the user's virtual machine session. Changing a language means changing the LANGBLOK pointer in the user's VMBLOK.

During system initialization, CP creates the LANGBLOK for the installation default language's message repository. This repository is part of the CP nucleus, so virtual storage has already been allocated from the SYSTEM VMBLOK for the pages of the repository. The installation default language is always the first LANGBLOK in the LANGBLOK chain.

To handle an additional message repository for another language, CP creates a new VMBLOK through the virtual buffer manager. This VMBLOK has the userid LANGUAGE; it allows CP to access enough virtual storage for the message repository pages.

Unlike the repository in the nucleus, message repositories defined in the System Name Table (DMKSNT) must have virtual storage specifically allocated. When building the LANGBLOK for a message repository defined in the System Name Table, CP allocates a virtual page from the LANGUAGE VMBLOK for each page of the message repository on DASD. CP then saves the virtual addresses in the LANGBLOK and initializes the swap table entry for each virtual page using the DASD address indicated in DMKSNT.

When a message is to be issued for a user, DMKERM finds the appropriate LANGBLOK. It then uses the table in LANGBLOK to get the virtual address of the repository page containing the message text. For the installation default language, CP uses information in the SYSTEM VMBLOK's page and swap tables to bring messages into real storage. For other supported languages, CP uses information in the LANGUAGE VMBLOK's page and swap tables to bring messages into real storage.

The CP Message Repository

The message compiler, the GENMSG command, generates a pageable text file from a message repository source file. The message repository text file consists of:

- A single general information page
- A variable number of data pages.

Information Page: The first page of a repository contains general information about the repository. This information page is made up of an identifier, a header, and table entries. CP uses the information on this page to build various control blocks.

Figure 1 shows the layout of the information page.

```
 0  | MSGREP identifier                     | not used |
 8  | Information used to build the                    |
    |   LANGBLOK control block                         |
18  |                                                  |
20  | Information used to build the                    |
    | LANGNTRY extensions to the LANGBLOK              |
    |                                                  |
```

Figure 1. Information Page for a Message Repository

- Identifier

  The compiler sets the first six bytes of the information page to "MSGREP". This identifies the file as a message repository. The next two bytes are unused.

- Header

  This information is used to build the control block for a language, LANGBLOK. The compiler initializes this portion of the repository as follows:

Label	**Value**
LANGNEXT	0
LANGPAGE	Number of data pages in the file (which corresponds to the number of table entries on the information page).
LANGFLAG	X'80' for a DBCS repository X'00' otherwise
LANGLOCK	0

Label	**Value**
LANGLANG	5-character langid, left-justified, padded with blanks
LANGVMBK	0
All reserved fields	0

CP initializes or modifies LANGNEXT, LANGFLAG, LANGLOCK, and LANGVMBK as appropriate when it builds the LANGBLOK control block for this repository.

- Table entries

  These are used to build LANGNTRY extension on the LANGBLOK. The compiler generates one table entry for each data page in the repository; CP then builds one LANGNTRY extension to the LANGBLOK for each table entry on this page.

  The compiler initializes each table entry as follows:

Label	**Value**
LANGLOW	Message-ID of the lowest number stored on the data page
LANGHIGH	Message-ID of the highest number stored on the data page
LANGADDR	0

  CP initializes LANGADDR when it builds the LANGNTRY extension.

Data Pages: All pages following the repository page are data pages. Data pages contain CP error messages and responses. DMKERM references these data pages in order to display CP messages. Each data page consists of a header, message entries, and index entries. (Refer to Figure 2.)

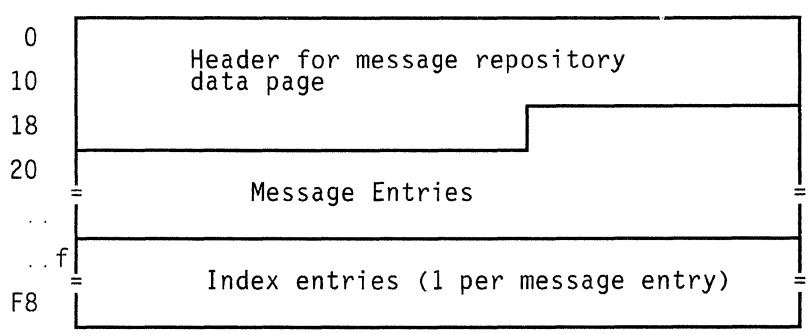


Figure 2. Data Page for a Message Repository

- Header

The header on repository data pages is mapped by REPHEAD DSECT in module DMKERM.

Label	**Value**
REPREP	String identifying the page as part of the message repository.
REPLANG	String indicating the language. This should be 5 bytes, with the language left-justified and padded with blanks.
REPAPPL	String indicating this is a CP repository. The compiler sets this to DMK for a CP message repository.
REPSUB	Substitution character that was specified in the message repository source file.
REPHDRLN	Number of digits to display in the error message. (It is specified in the repository source file; however, CP ignores this value and always displays a 3-digit message number.)
REPCNT	Number of message entries contained on this data page.
REPINXPT	Displacement from the start of the 4K page to the index portion of the data page.
REPTXTPT	Displacement from the start of the 4K page to the text of the first message on this data page.
REPDBCS	Indicates whether the repository is DBCS (X'80') or not (X'00').

- Message entries

Message entries are grouped together and follow immediately after the header. Message entries are variable in length; they consist of:

- An action character (1 byte)
- A length field, which indicates the length of the message text (1 byte)
- Message text.

- Index entries

  Index entries on repository data pages are grouped together and follow the message entries. There is one index entry for each message entry on the data page. Index entries are mapped by the MSGINDEX DSECT in module DMKERM. They consist of:

  – A message identifier
    – Message number (2 bytes)
    – Format number (1 byte)
    – Line number (1 byte).

    For example, X'00050201' is message 5, format 2, line 1.

  – The displacement from the start of the 4K page to the message entry on the data page.

Module DMKERM

Many CP modules call DMKERM to display error messages. Module DMKERM builds the message-ID from the input parameters and gets the address of the LANGBLOK that identifies the message repository used for the virtual machine. (The VMBLOK contains this LANGBLOK address.)

DMKERM then:

- Scans the LANGNTRY extensions to determine which data page of the message repository has the message to be displayed
- Pages in the repository data page containing the message
- Does a binary search on the index to find the index entry for the message to be displayed
- Gets the displacement to the message from the index entry
- Calculates the address of the message
- Sets up the message to be displayed.

Note: The message repository object file maintains 4K page boundaries:

- *Message texts do not cross page boundaries.*
- *All lines of a multiple line message are on the same 4K page.*
- *Not all formats for a given message have to be on the same 4K page.*

Program States

When instructions in the Control Program are being executed, the real computer is in the supervisor state; at all other times, when running virtual machines, the real computer is in the problem state. Therefore, privileged instructions cannot be executed by the virtual machine. Programs running on a virtual machine can issue privileged instructions; but such an instruction either (1) causes an interruption that is handled by the Control Program, or (2) is intercepted and handled by the processor, if the virtual machine assist feature or VM/370 Extended Control-Program Support is enabled and supports that instruction. CP examines the operating status of the virtual machine PSW. If the virtual machine indicates that it is functioning in supervisor mode, the privileged instruction is simulated according to its type. If the virtual machine is in problem mode, the privileged interrupt is reflected to the virtual machine.

The control program and the MVS/SP virtual machine using preferred machine assist may operate in supervisor state on the real processor. All other programs operate in problem state. The hardware microcode routes interrupts for the preferred virtual machine directly to this virtual machine. All other user interrupts, including those generated by privileged operations, are handled by either the control program or the processor (if the processor has the virtual machine assist feature or VM/370 Extended Control Program Support is available). Only those interrupts that the user program would expect from a real machine are reflected to it. A problem program will execute on the virtual machine in a manner identical with its execution on a real System/370 processor, as long as it does not violate the CP restrictions. See the *VM/SP HPO Planning Guide and Reference* for a list of the restrictions.

Using Processor Resources

CP allocates the processor resource to virtual machines according to their operating characteristics, priority, and the system resources available.

Virtual machines are dynamically categorized at the end of each *time slice* as interactive or noninteractive, depending upon the frequency of operations to or from either the virtual system console or a terminal controlled by the virtual machine.

Virtual machines are dispatched from one of three queues, called Queue 1, Queue 2, and Queue 3. In order to be dispatched from one of these queues, a virtual machine must be considered executable (that is, not waiting for some activity or for some other system resource). Virtual machines are not considered dispatchable if the virtual machine:

- Enters a virtual wait state after an I/O operation has begun
- Is waiting for a page frame of real storage
- Is waiting for an I/O operation to be translated by CP and started
- Is waiting for CP to simulate its privileged instructions
- Is waiting for a CP console function to be performed.

Selecting a Virtual Machine to Run

CP uses several queues and lists to determine which CP task or virtual machine should next receive a time slice from the processor. Figure 3 shows the relationship between these queues and lists, which are:

- The eligible list, which contains virtual machines waiting to be added to the run list. Virtual machines on the eligible list are not considered to be in queue.
- The run list, which is a list of virtual machines that are considered in queue but not necessarily runnable.
- The dispatch request queue, which contains pointers to CP tasks like CPEXBLOKs, IOBLOKs, and TRQBLOKs. In AP and MP systems, there is a dispatch request queue for each processor.
- The dispatch list, also called the true run list, which contains pointers to virtual machines that are both in queue and runnable. In AP and MP systems, there is a dispatch list for each processor.

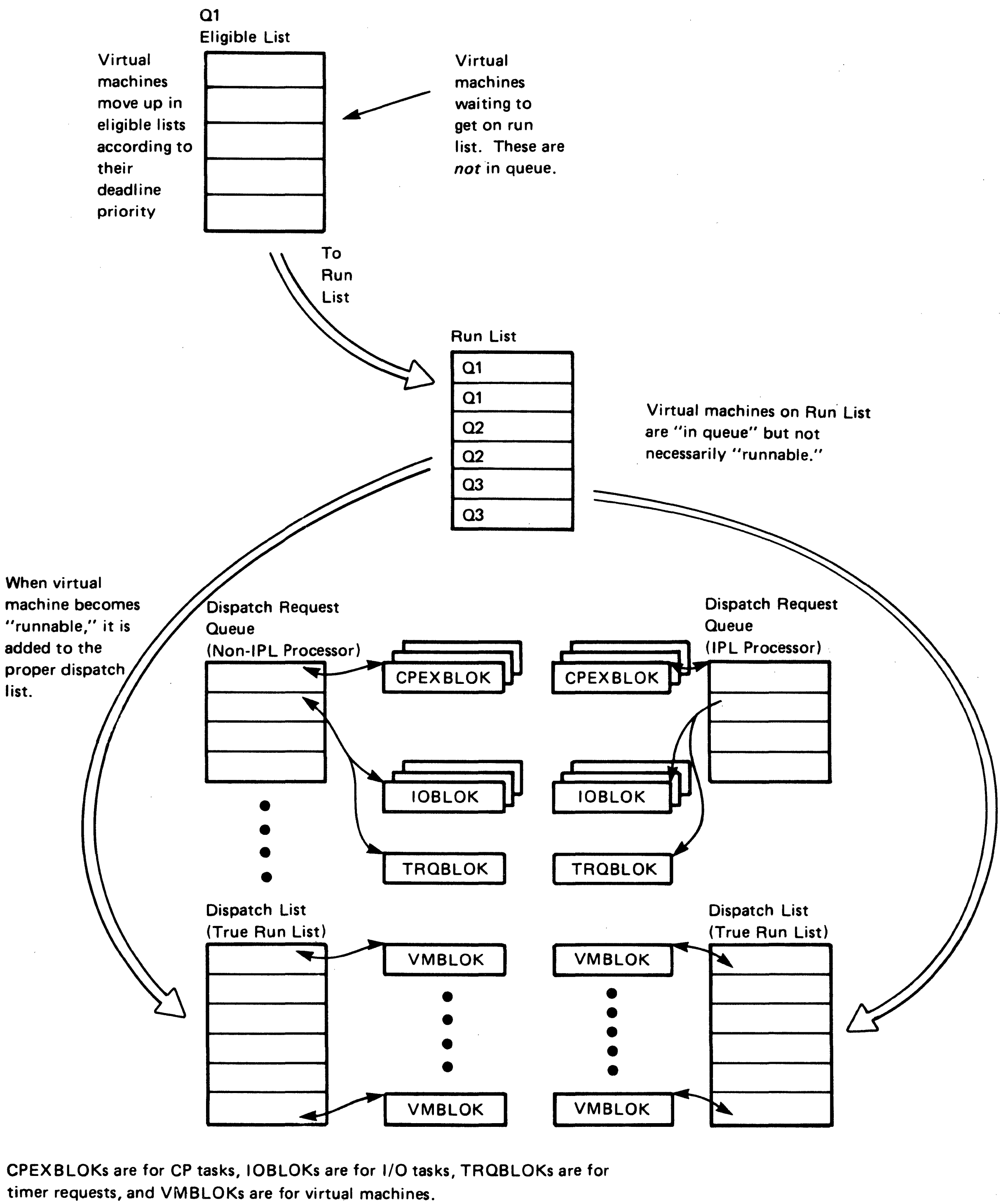


Figure 3. Queues and Lists That CP Uses to Select a Virtual Machine to Run

Queue 1

Virtual machines in Q1 are considered conversational or interactive users and enter this queue when an interrupt is reflected to the virtual machine user after it has been idle for at least 300 milliseconds. The Q1 virtual machines are ordered in the dispatch list by their deadline priorities. A deadline priority is a value calculated by the scheduler every time a user is dropped from a queue (queue drop time). This value is based on paging activity, processor usage, the load on the system, and user priority. Deadline priority is used to determine when the virtual machine receives its next time slice.

A Q1 virtual machine will usually have a better (earlier) deadline priority than a Q2 virtual machine. For information on how CP calculates deadline priorities, refer to "Dispatching and Scheduling" under "CP Program Organization" in the second part of this manual.

When a virtual machine completes its time slice, it is dropped from the run list (dropped from queue) and placed on an eligible list. A virtual machine that enters CP command mode is also dropped from the run list.

When a write is pending to a Q1 or Q2 virtual machine's console, the scheduler may keep it in queue when the virtual machine would otherwise be dropped. The scheduler uses a 300-millisecond delay to eliminate queue drops for such virtual machines.

Queue 2

Virtual machines in Q2 can be either interactive or noninteractive. In CP mode, Q1 virtual machines are normally dispatched before Q2 virtual machines. This means that CMS users entering commands that do not involve disk or tape I/O operations should get fast responses from the system even if the system has many Q1 virtual machines. Some virtual machines in Q2 are dispatched before virtual machines in Q1 because of their user priority, current resource level, or for other reasons.

For Q1 virtual machines, the user bias factor is divided by 8 because the Q1 time slice is 1/8th the Q2 time slice. Compared with a Q2 virtual machine, a Q1 virtual machine receives 1/8th the amount of processor time, eight times as often. Operating constantly in either queue, a virtual machine should receive the same amount of processor resources over an extended period of time. The first time that CP changes your virtual machine from Q1 to Q2, CP classifies it as interactive. If your virtual machine completes a time slice without moving back to Q1, the scheduler changes your virtual machine to noninteractive.

Queue 3

Virtual machines in Q3 are considered noninteractive. CP changes your virtual machine from Q2 to Q3 when it has used six consecutive Q2 time slices without entering long idle wait. The differences between Q2 and Q3 virtual machines are reflected in their deadline priority calculations and the amounts of such processor time they are allowed in queue. Q3 virtual machines are allowed eight consecutive Q2 processor time slices before they are dropped from the queue. Because of the eightfold increase in processor time allowed each time in queue, the user bias factor is multiplied by 8 before adding to the current time-of-day to form the deadline priority. Q3 virtual machines should receive eight times as much processor time each time in queue as Q2 virtual machines, but only 1/8th as often.

To reiterate the Q1/Q2 statement: Operating constantly in any queue, a virtual machine should receive the same amount of processor resources over an extended period of time. This does not necessarily mean that a virtual machine will perform the same when operating in Q3 mode as when operating in standard Q2 mode. An amount of overhead (roughly proportional to the small number of resident pages) is used for each virtual machine when it drops from queue. When operating in Q3 mode, a virtual machine may perform much better than in normal Q2 mode because it is undergoing fewer queue drops.

You can alter the queue scheme by using the NOQ2 or NOQ3 option on the SET QDROP OFF command. Specifying NOQ3 will force a virtual machine to be kept in Q1 or Q2. Specifying NOQ2 will force a virtual machine to be kept only in Q1.

When the SET QDROP userid OFF command is specified with the USERS operand, the QDROP OFF status is extended to any virtual machine communicating through VMCF or IUCV to the service virtual machine specified. See the *VM/SP HPO CP Command Reference* for more information on the SET QDROP command.

Functional Information

The functional diagrams that follow describe the program logic associated with various control program functions. Not all CP functions are described. These functional diagrams are meant to describe the CP functions about which you may want more detailed information if you are debugging, modifying, or updating CP.

Figure 4 describes the CP initialization process.

Figure 5 describes system shutdown and automatic warm start.

Figure 6 and Figure 7 describe the real and virtual I/O control blocks used by CP in its I/O control.

Figure 8, Figure 9, and Figure 10 show how CP handles SVC, external, and program interrupts.

Figure 11 describes the CP paging function.

Figure 12 and Figure 13 describe the CP spooling function (both virtual and real).

Figure 14 shows how virtual tracing is performed.

Figure 15 describes CP PER command processing.

Figure 16 describes CP PER interrupt processing.

Figure 17 shows the steps involved in translating a virtual address to a real address and gives an example of address translation.

Figure 18 shows how SNA Consoles Communication Services (SNA CCS) communicates with the VTAM Communications Network Application (VM/VCNA) and with the rest of VM/SP HPO.

Figure 19 shows the structure of SNA Console Communication Services (SNA CCS) control blocks.

Figure 20 shows the structure of SNA DIAL control blocks.

The functional information contained in these diagrams is intended for system programmers and IBM Field Engineering program support representatives.

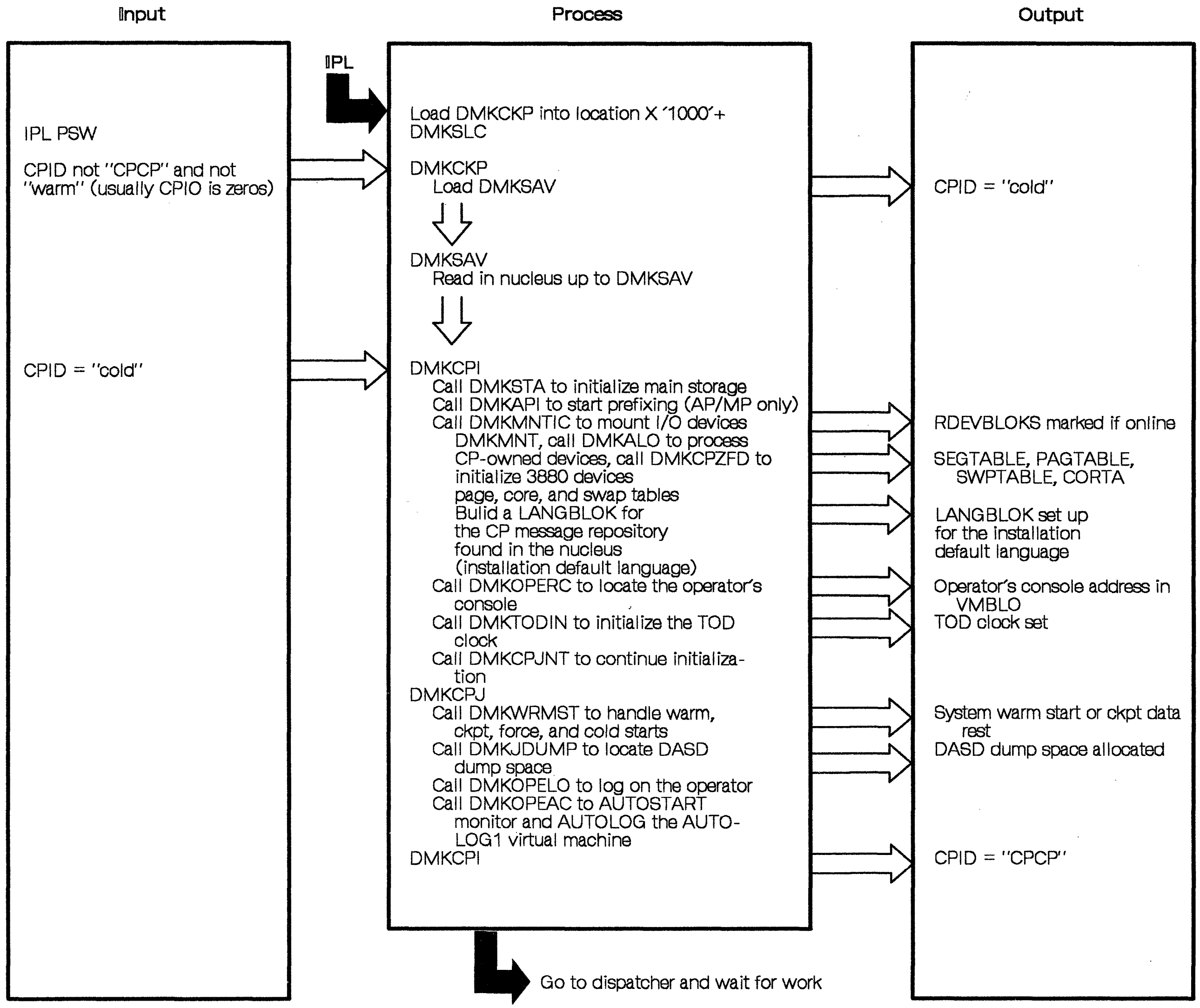


Figure 4. Initialization on a Cold Machine

Input	Process	Output
Shutdown or Shutdown REIPL	DMKCPS Halt the system Go to DMKDMPRS	CPID = "CPCP" for shutdown CPID = "warm" for shutdown REIPL
System abend	DMKDMP Dump storage to the dump device ↓ DMKDMPRS Load DMKCKP ↓	CPID = "CPCP" after dump to printer or tape and CPID = "warm" after dump to DASD
CPID = "CPCP" for shutdown or following dump to printer or tape CPID = "warm" for shutdown REIPL or following dump to DASD	DMKCKP Read in DMKCKD, DMKCKF, DKMCKH, DMKCKM, DMKERP, DMKCKW, and DMKCKN Call DMKCKDEV to drain I/O interrupts, and issue a HIO to each device Call DMKCKFIL to checkpoint the warm system	Log message, accounting data, SFBLOKS, Reader Hash Table plus extension pages, SPUBLOKS, allocation RECBLOKS, and SHOBLOKS saved on warm start area of SYSRES
	Call DMKCKM to save virtual machines enabled for VMSAVE	Virtual machines saved
	If CPID = "CPCP", then load a wait state PSW (code 008)	CPID = "SHUT"
	If CPID = "warm" then: DMKCKP Load DMKSAV ↓ DMKSAV Processing continues as indicated for DMKSAV and DMKCPI (see "Initialization on a cold machine," page 18) except that CPID = "wrm"	CPID = "wrm"

Figure 5. System Shutdown and Automatic Warm Start

The real machine configuration is represented by a set of related control blocks. These blocks are:

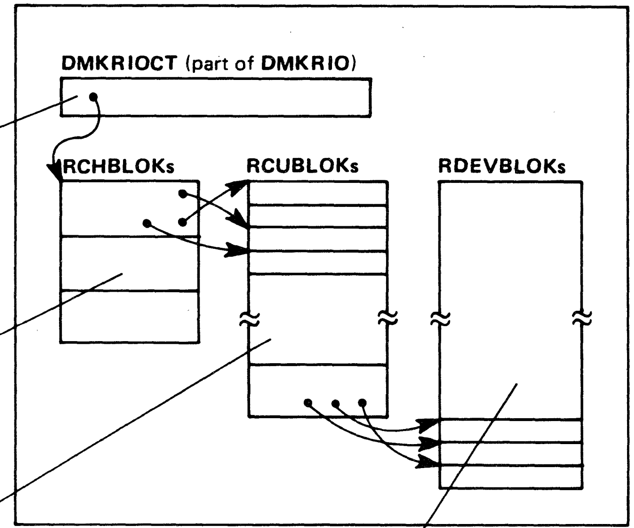
- in the nucleus
- built from macros during system generation
- located at system **IPL** and initialized then for operation.

There is one control block per channel, per control unit, and per device.

The characteristics of real **I/O** control are:

- Block multiplexing (**BMPX**) with **RPS** (Rotational Position Sensing) is used.
- Multi-path scheduling is not used.
- All **I/O** operations are handled by **CP** scheduling and interrupt handling.

Relationship of Real I/O Control Blocks



DMKRIOCT – real channel table[1]

XXXX – negative value (**FFFF**) indicates that no channel exists
– positive value is an index to the **RCHBLOK**

RCHBLOK – real channel block[1]

Channel identification
Scheduling Control
.
.
.

XXXX	XXXX	XXXX	XXXX	Control Unit Index Table
XXXX	XXXX		XXXX	

XXXX if negative (**FFFF**), no control unit exists
if positive, that value is an index to the **RCUBLOK**

RCUBLOK – real control unit block[1]

Control Unit identification
Scheduling Control
.
.
.

XXXX	XXXX	XXXX	XXXX	Device Index Table
XXXX		XXXX	XXXX	

XXXX if negative (**FFFF**), no device exists
if positive, that value is an index to **RDEVBLOK**

RDEVBLOK – real device block[1]

Device identification
Scheduling Control
Terminal Control
Spooling Control
Dedicated Control
Error Recovery
Allocation Control
.
.
.
.

Part of the **RDEVBLOK** pertains to functions that are device independent, that part of the **RDEVBLOK** is used in the same way for all devices. However, some of the fields in the **RDEVBLOK** have multiple uses, depending on the device type and function.

[1] For a complete description of CP control blocks, see *IBM Virtual Machine/System Product High Performance Option: Data Areas and Control Blocks – CP.*

Figure 6. Real I/O Control Blocks

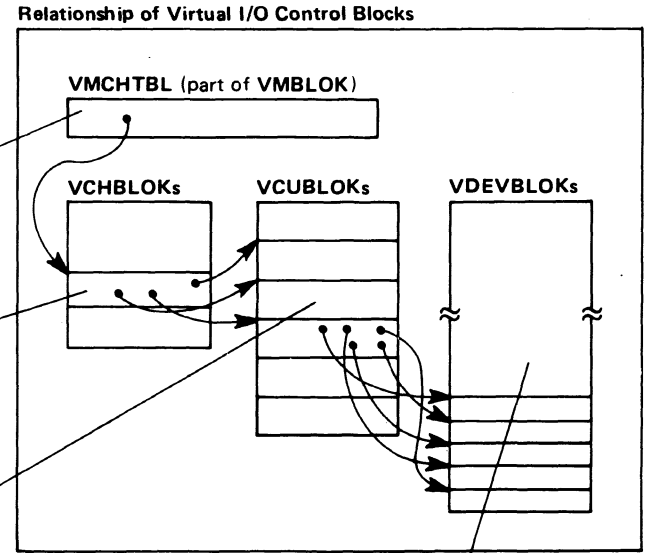
The virtual machine configuration is represented by a set of related control blocks. These blocks are:

- built by **CP** at **LOGON** from data in directory
- modified by user commands (for example, **DETACH**, **LINK**, **DEFINE**)

There is one control block per channel, per control unit, and per device.

The characteristics of virtual **I/O** control are:

- **BMPX** (block multiplexing) is supported
- **RPS** (rotational position sensing) is supported
- the virtual machine operating system performs scheduling
- virtual **I/O** control blocks simulate real hardware interface
- virtual unit record devices use spooling
- virtual console is simulated on terminal
- minidisks simulate **DASD**
- dedicated devices are supported



VMCHTBL – virtual channel index table

VCHBLOK – virtual channel block[1]

Channel identification status . . .			
XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX

XXXX — if negative (**FFFF**), no control unit exists; if positive, the value is an index to the **VCUBLOK**

VCUBLOK – virtual control unit block[1]

Control unit identification status . . .			
XXXX	XXXX	XXXX	XXXX
XXXX			
XXXX			

Device Index Table

XXXX — if negative (**FFFF**), no device exists; if positive, the value is an index to the **VDEVBLOK**

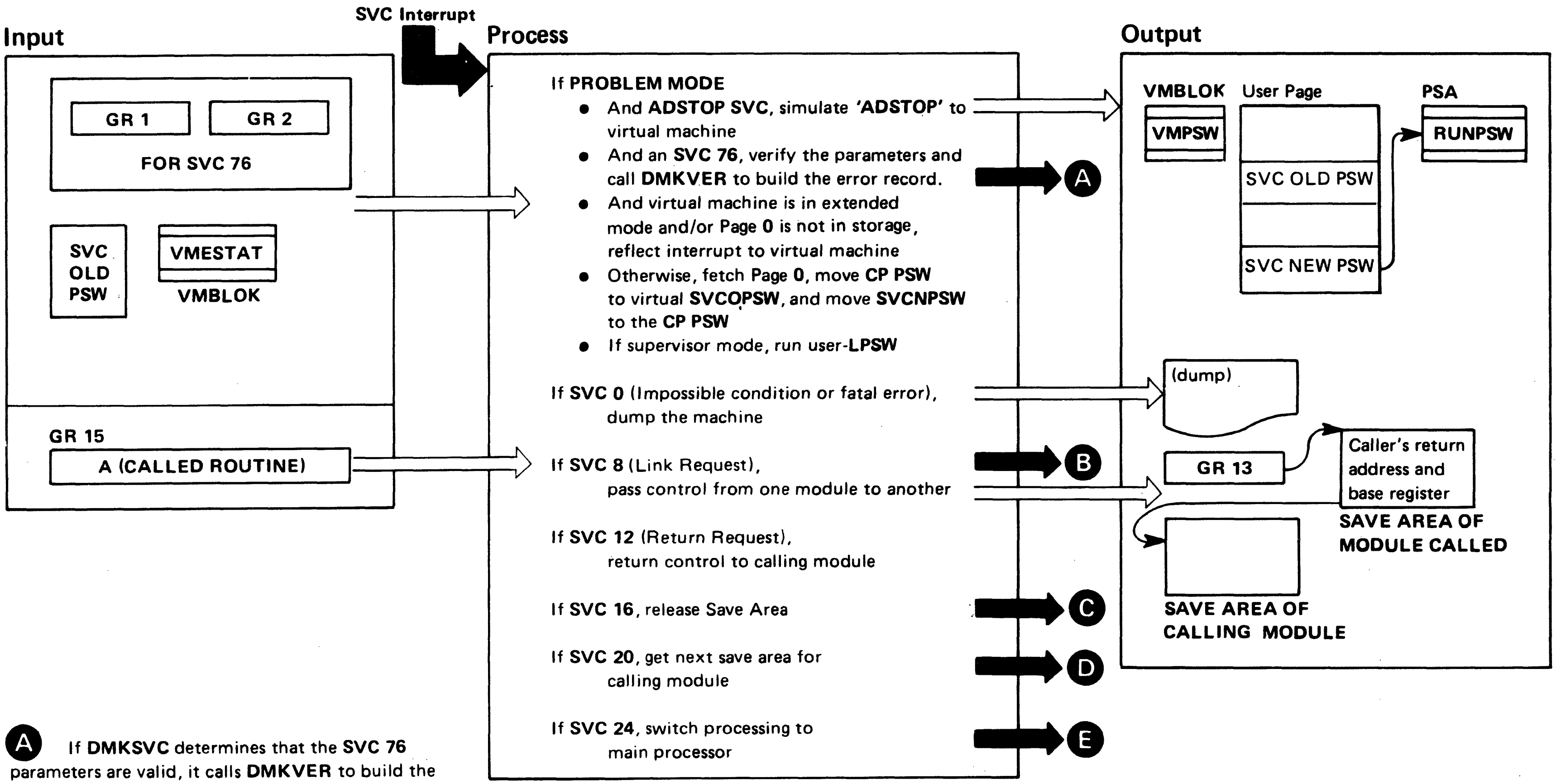
VDEVBLOK – virtual device block[1]

Device identification
Status pending
Positioning
Terminal control
Spooling control
. . .
RDEVBLOK Pointer

Part of the **VDEVBLOK** contains device independent information and is used identically in all **VDEVBLOKs**. However, some fields of the **VDEVBLOKs** have multiple uses, depending on the device type.

[1] For a detailed description of the CP control blocks, see *VM/SP HPO: Data Areas and Control Blocks – CP.*

Figure 7. Virtual I/O Control Blocks



A If **DMKSVC** determines that the **SVC 76** parameters are valid, it calls **DMKVER** to build the error record. If the parameters are not valid or if **DMKVER** cannot build the error record, **DMKSVC** reflects the **SVC** back to the virtual machine. If the error record is recorded, **DMKVER** gives control to the dispatcher with the user's running status set to return to the next sequential instruction following the **SVC 76**.

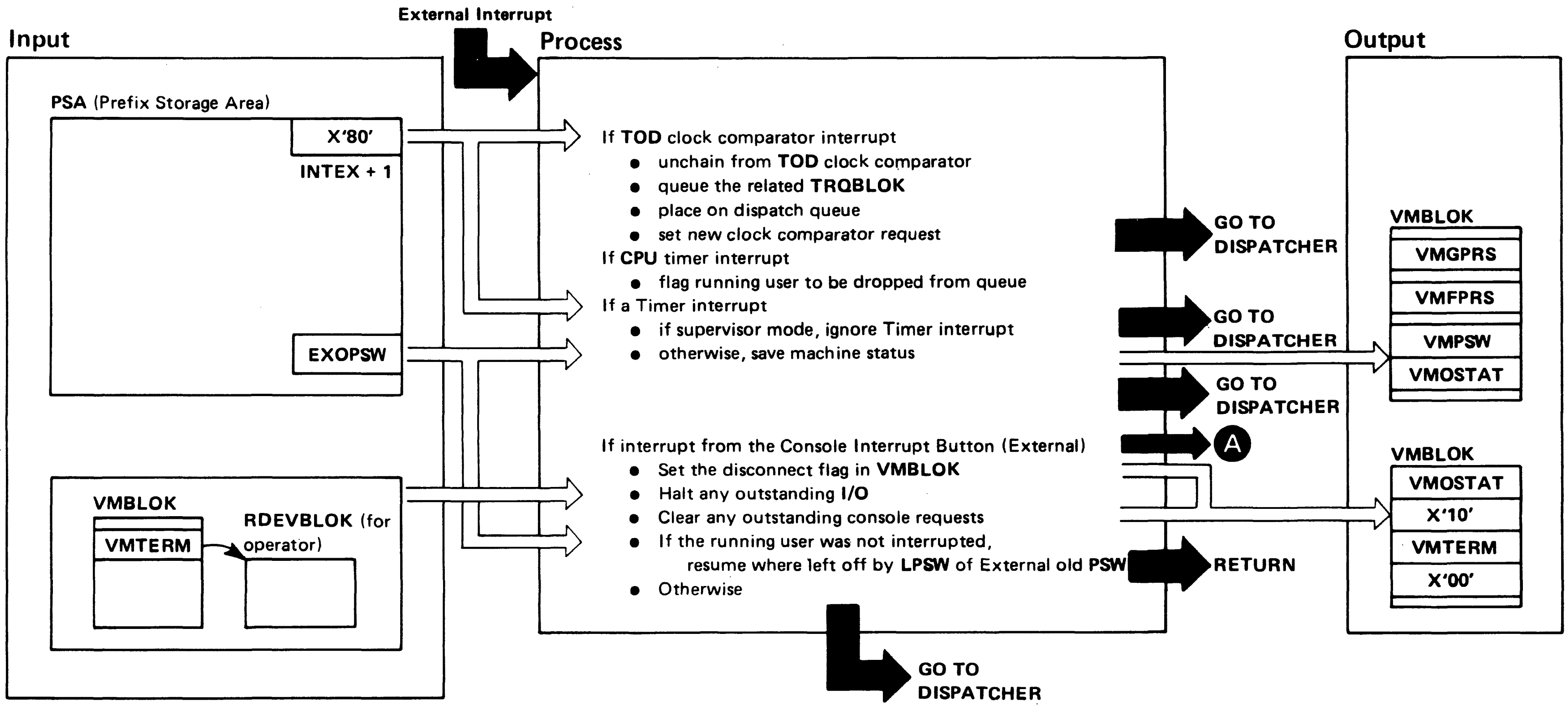
B A new save area is acquired and passed on. The caller's addressability register (**R 12**), the save area address (**R 13**), and the return address (**SVCOPSW**) are saved in the new save area.

C Control is returned to module issuing **SVC 16**, rather than to calling module as in **SVC 12**.

D Return is to module issuing **SVC 20**.

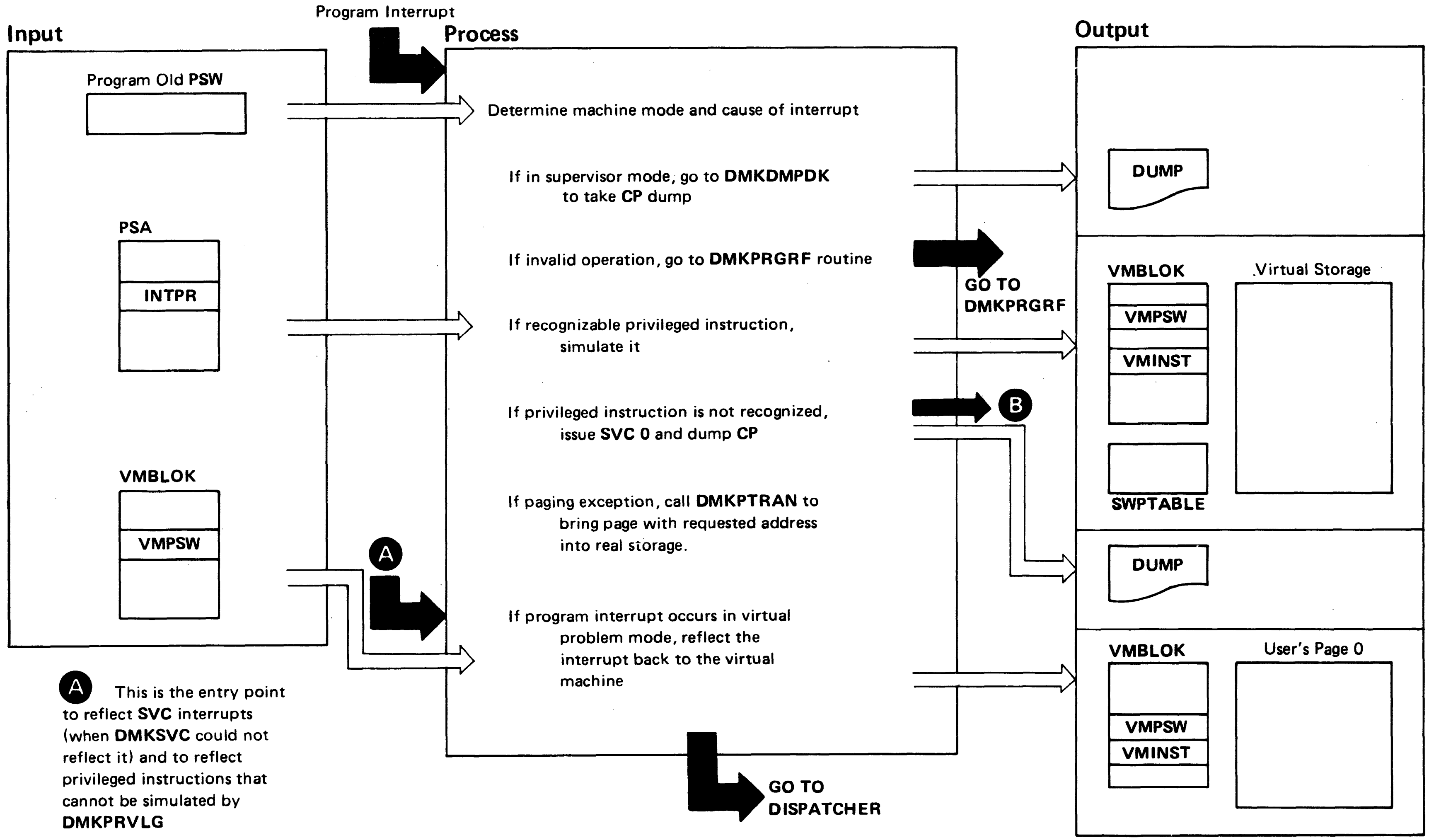
E Return is on other processor to module issuing **SVC 24**.

Figure 8. SVC Interrupt Handling



A External interrupt from control panel is used to disconnect the system operator's terminal. The system operator may reconnect at any other terminal via the **LOGON** command.

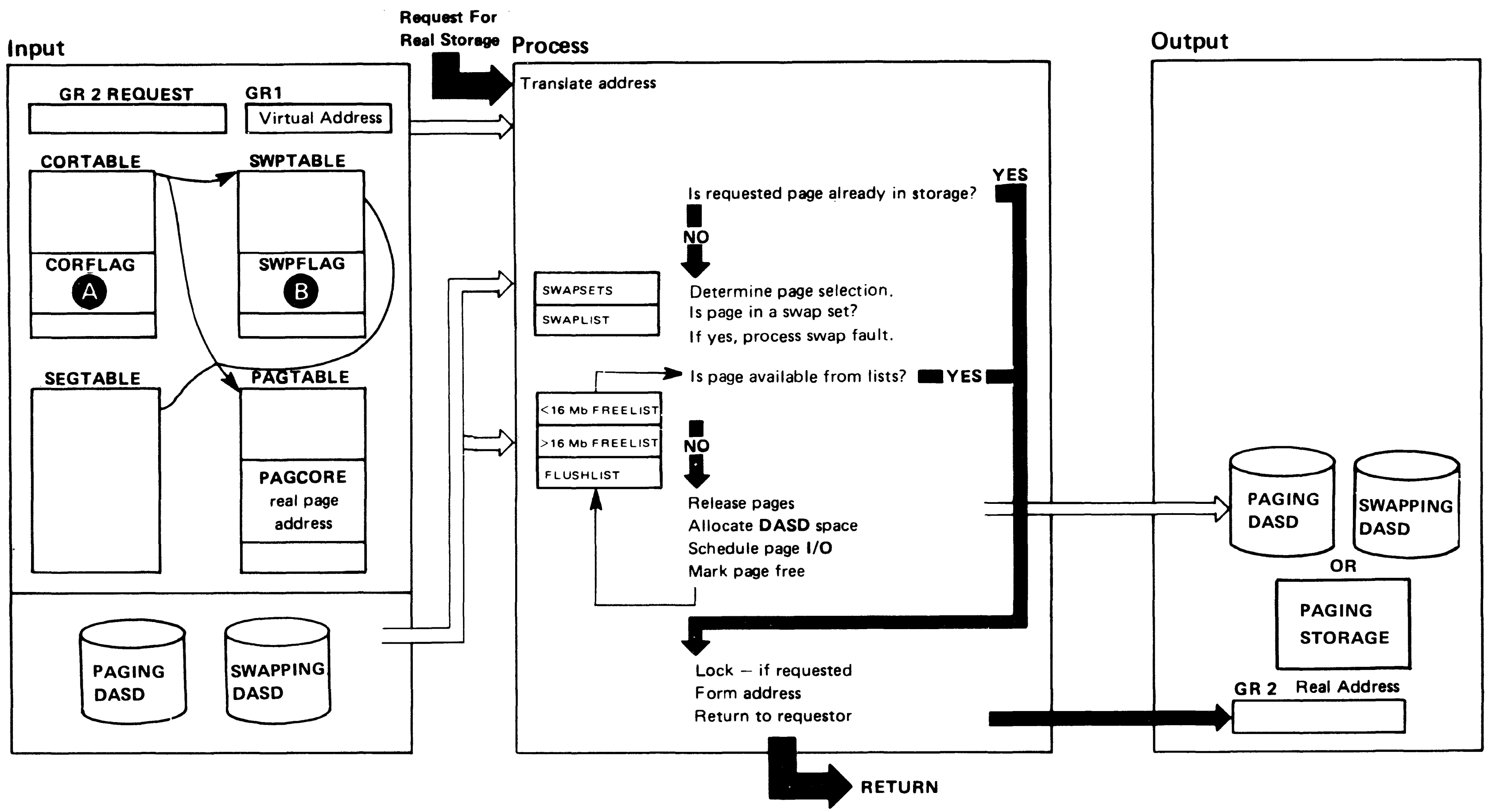
Figure 9. External Interrupt Handling



A This is the entry point to reflect **SVC** interrupts (when **DMKSVC** could not reflect it) and to reflect privileged instructions that cannot be simulated by **DMKPRVLG**

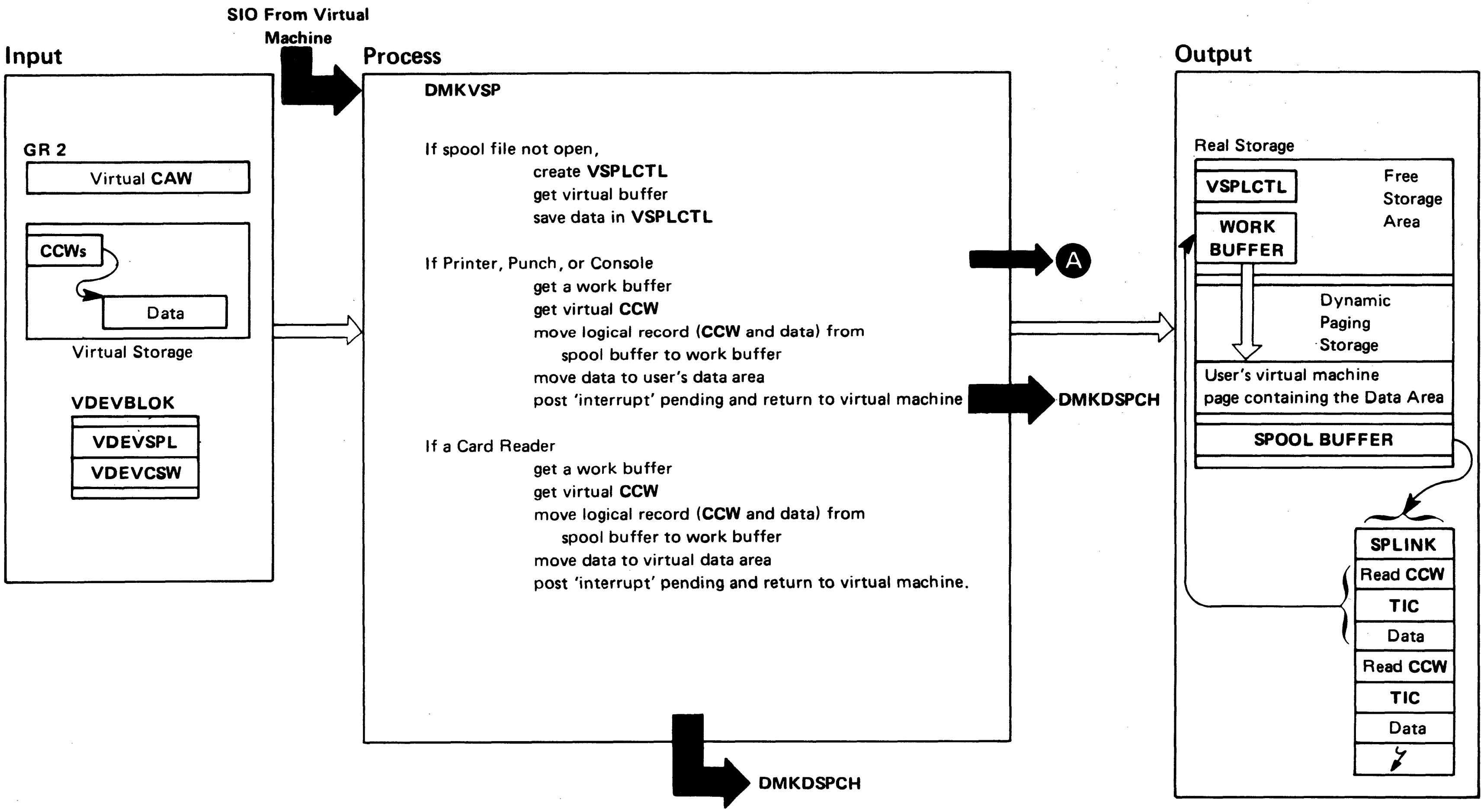
B Invalid operation code is in **GR 0**. The **VMINST** field of the **VMBLOK** contains the image of the privileged instruction that caused the interrupt

Figure 10. Program Interrupt Handling



A Bits defined for CORFLAG				B Bits defined for SWPFLAG			
CORIOLCK	EQU	X'80'	Page locked for I/O	SWPTRANS	EQU	X'80'	Page in transit
CORCFLCK	EQU	X'40'	Page locked by console function	SWPRECMP	EQU	X'40'	Page permanently assigned
CORFLUSH	EQU	X'20'	Page is in flush list	SWPALLOC	EQU	X'20'	Page enqueued for allocation
CORFREE	EQU	X'10'	Page is in free list	SWPSHR	EQU	X'10'	Page shared
CORSHARE	EQU	X'08'	Page is shared	SWPREF1	EQU	X'08'	1st half page referenced
CORSWAP	EQU	X'04'	Page is in swap list	SWPCHG1	EQU	X'04'	1st half page changed
CORCP	EQU	X'02'	Page belongs to CP	SWPREF2	EQU	X'02'	2nd half page referenced
CORDISA	EQU	X'01'	Page disable – not available	SWPCHG2	EQU	X'01'	2nd half page changed

Figure 11. Paging



A Virtual console spooling is the same as printer spooling except that:

- A skip to channel one **CCW** is inserted every 60 lines of output
- The operator's virtual console spool buffer is written for every 16 lines of output
- The Virtual spool buffer is written to the allocated spool device when the first **CCW** is placed in the Virtual buffer. The buffer is kept in a pseudo closed state so that checkpoint saves the buffer in the event of a system failure.

Figure 12. Virtual Spooling

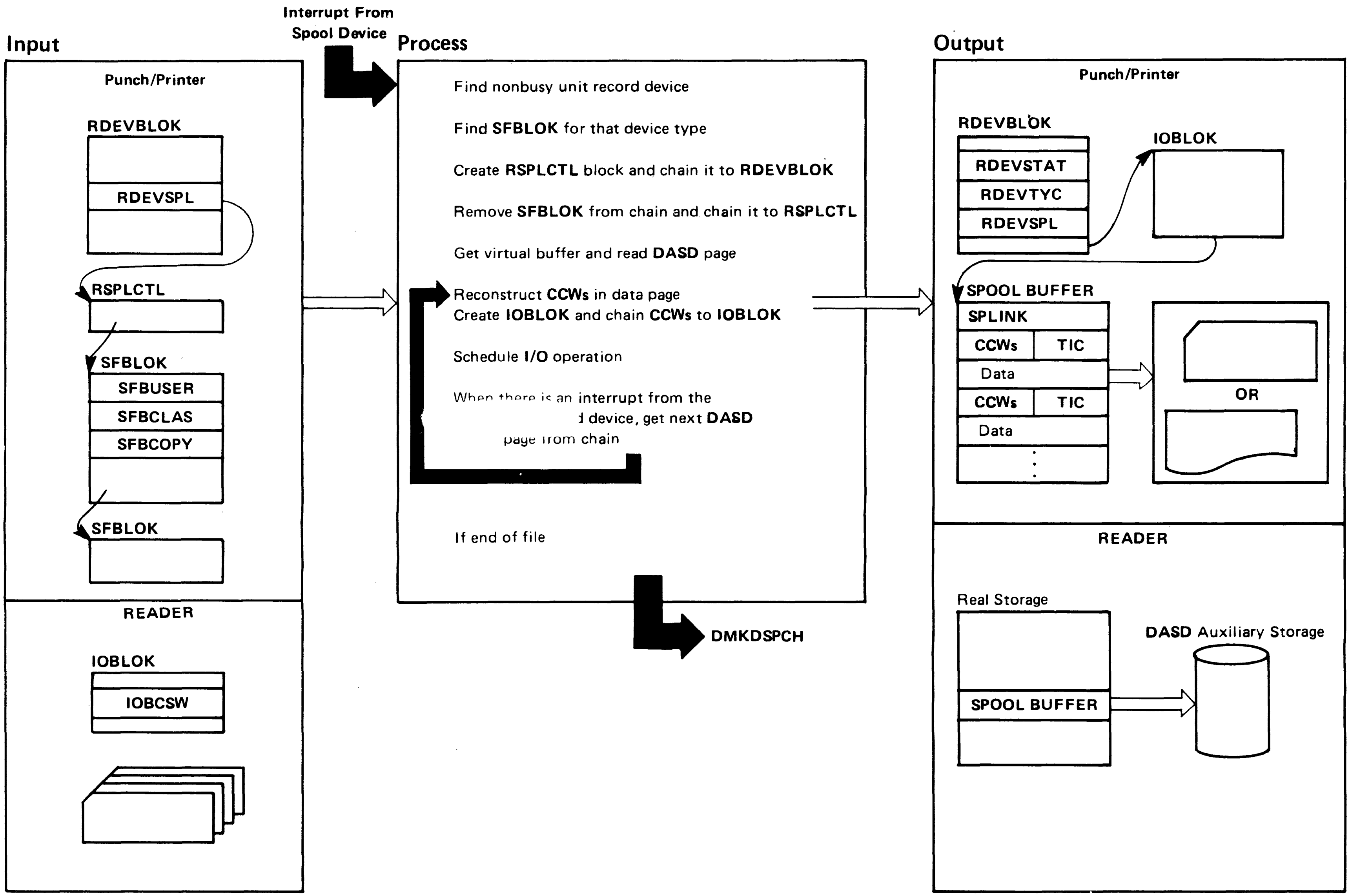
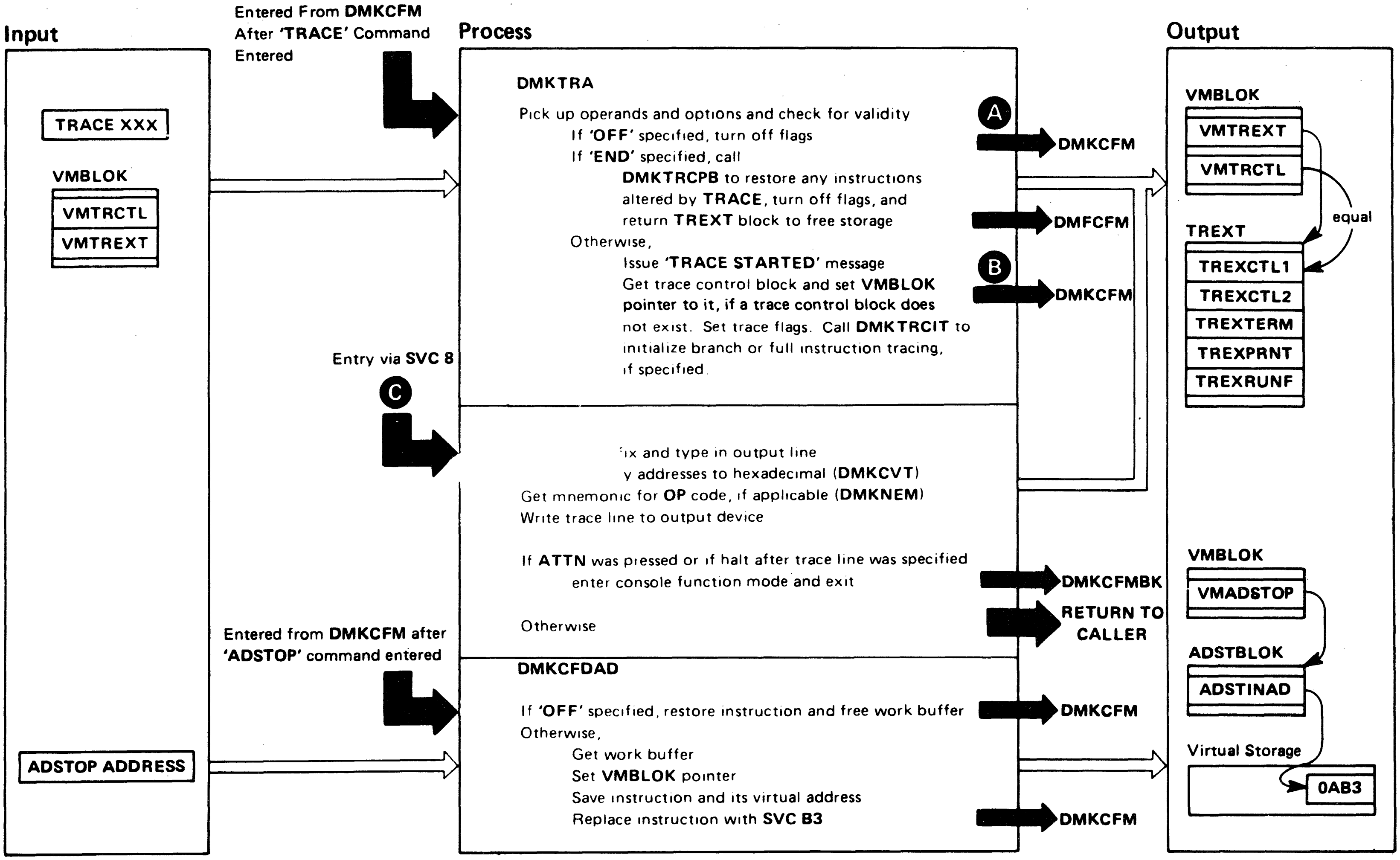


Figure 13. Real Spooling



Comments

A If this turns off the last flag, then the **TREXT** block is returned to free storage. If branch and instruction tracing are both turned off, **DMKTRCPB** is called to put back any instructions altered by **TRACE**.

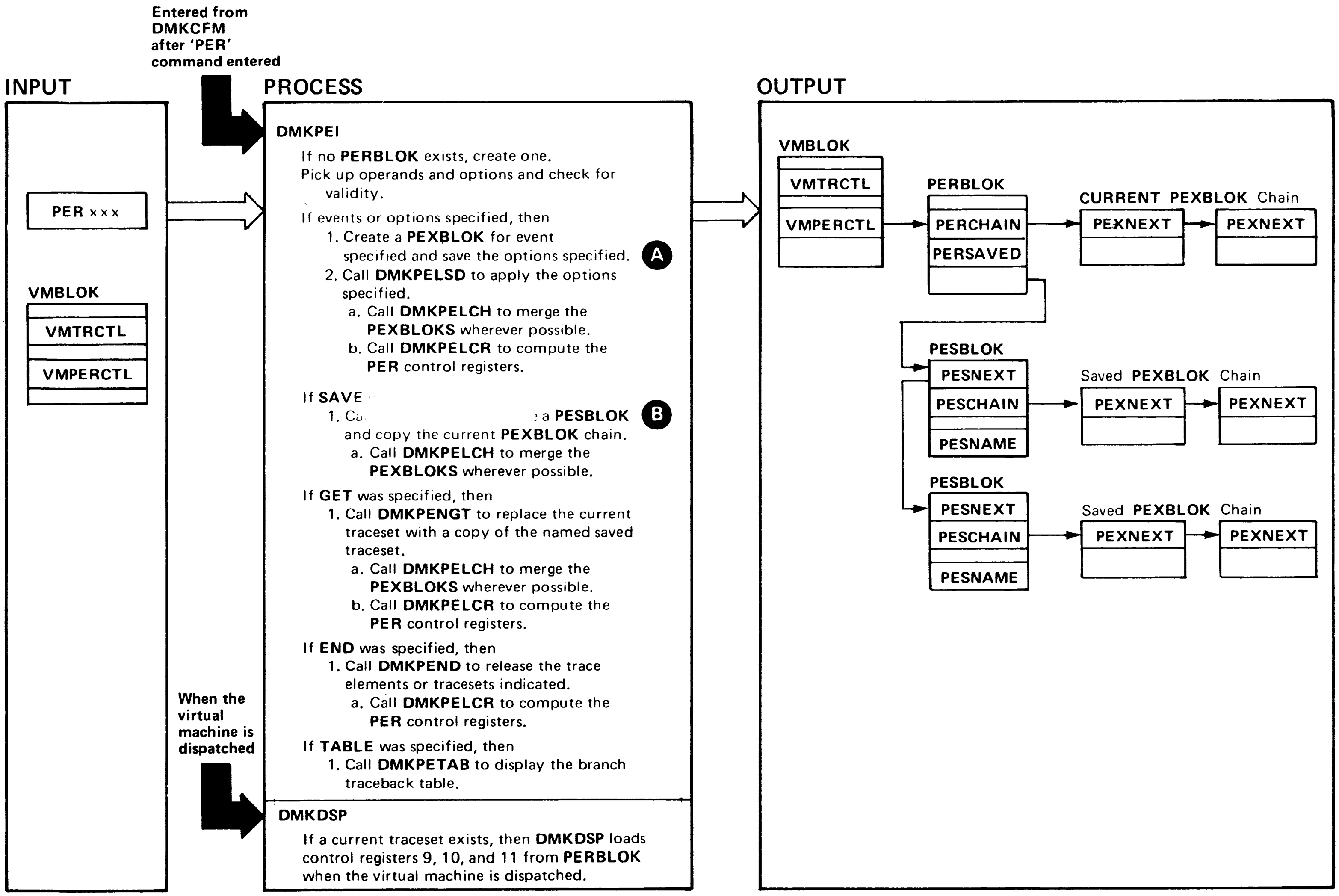
B **VMTRCTL** and **TREXCTL1** are identical.

C Entry via **SVC 8** as follows

	Entry Point	From
External Interrupt	DMKTRCEX	DMKDSP
I/O Interrupt	DMKTRCIO	DMKDSP
Program Interrupt	DMKTRCPG	DMKPRG
Privileged Instructions	DMKTRCPV	DMKPRV
I/O Operations	DMKTRDSI	DMKVIOEX
Virtual and Real CSWs	DMKTRCSW	DMKVIOIN

	Entry Point	From
SVC, branch or full instruction trace	DMKTRCSV	DMKPSA
Restore user instructions altered by tracing	DMKTRCPB	DMKTRA
Initialize instruction tracing	DMKTRCIT	DMKTRA

Figure 14. Virtual Tracing



Comments

A Each trace element is represented by a **PEXBLOK**.

B The **PEXBLOK**s for each saved traceset are chained from a **PESBLOK**.

Figure 15. CP PER Command Processing

INPUT

VMBLOK
- VMTRCTL
- VMPERCTL

PERBLOK
- PERADDR
- PERCDE

PROCESS

DMKPRG entered via PSW swap as a result of a program interrupt.

DMKPRG

If this is a **PER** interrupt, then
1. Save the **PER** event information
   a. in the **PERBLOK** if **CP PER** is active
   b. in the **ECBLOK** if user **PER** is active
2. Set **PER** pending flag (**VMPERND**)

DMKPRV is entered via a **GOTO** from **DMKPRG** to simulate privileged instructions. **DMKTMR** is entered via a **GOTO** from **DMKPRV**.

DMKPRV, DMKTMR

If the instruction simulated by **DMKPRV** or **DMKTMR** alters storage or a register and **PER** is active, then **DMKPRV** or **DMKTMR** will simulate a **PER** event by loading the **PER** information into
1. the **PERBLOK** if **CP PER** is active
2. the **ECBLOK** if user **PER** is active

When the virtual machine is dispatched.

DMKDSP

If a **PER** interrupt is pending due to **CP PER**, then call **DMKPERIL** and then turn off the **PER** pending flag.

If a **PER** interrupt is pending due to a user **PER**, then reflect the **PER** interrupt to the user's virtual machine.

Entry via SVC 08 from dispatcher.

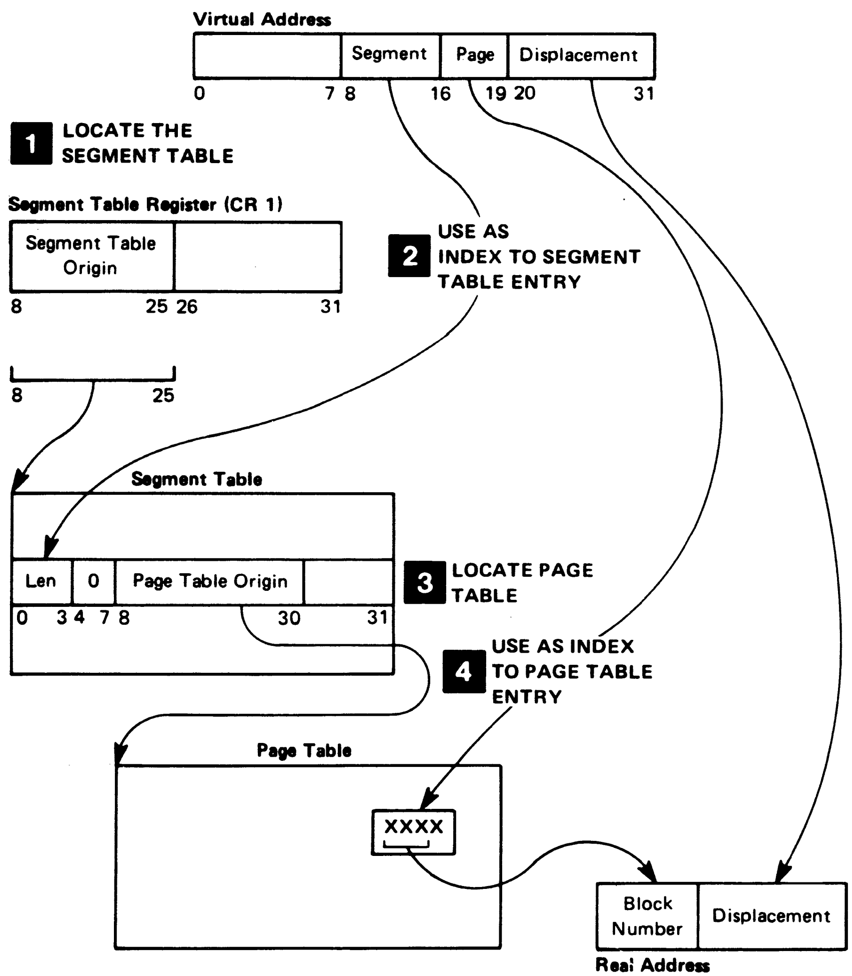
DMKPERIL

Scan the **PEXBLOK**s in the current traceset to determine if the **PER** interrupt satisfies the conditions in any of the **PEXBLOK**s. If it does, then call **DMKPETAL** to display the instruction.

OUTPUT

The instruction display.

Figure 16. CP PER Interrupt Processing



Example

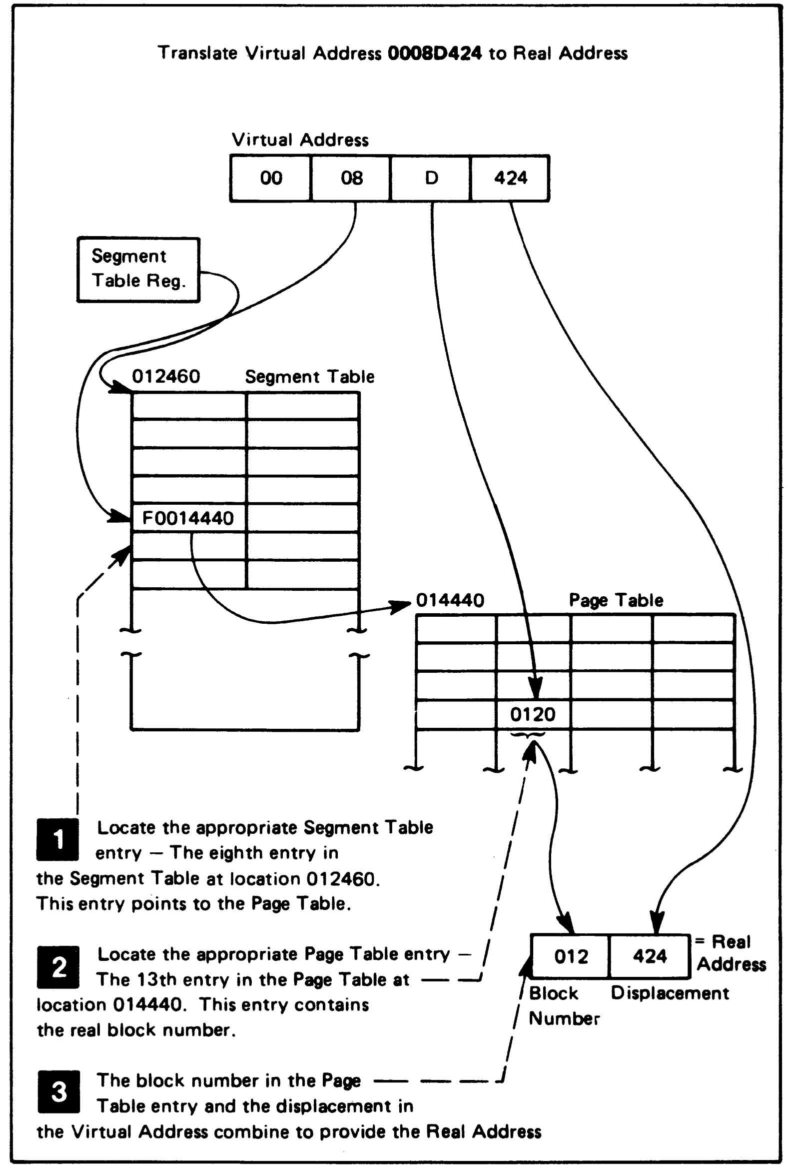


Figure 17. Virtual-to-Real Address Translation

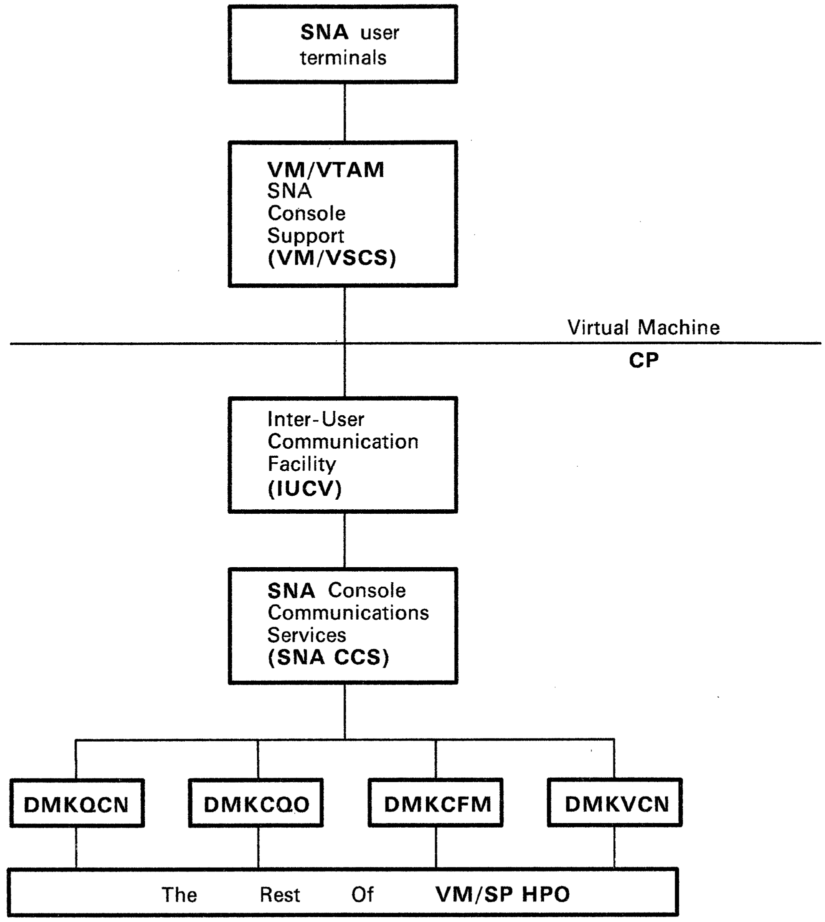


Figure 18. SNA CCS Interfaces

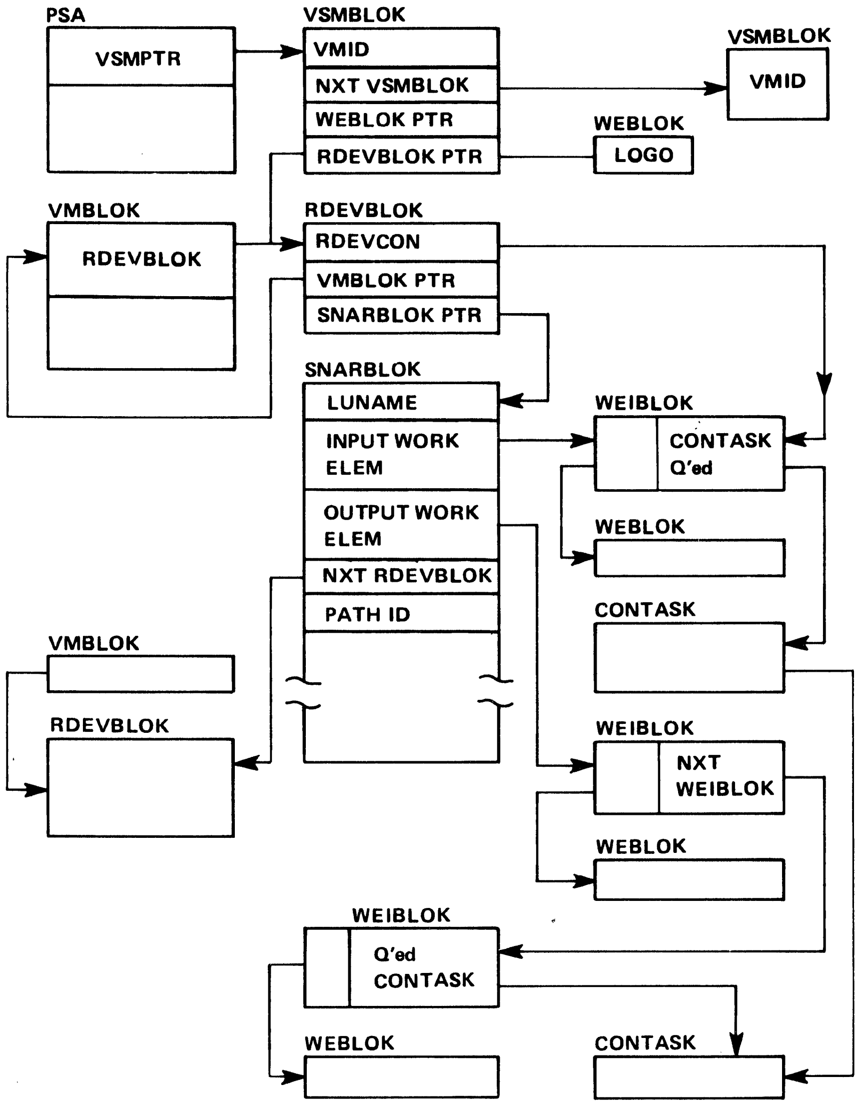


Figure 19. SNA CCS Control Block Structure

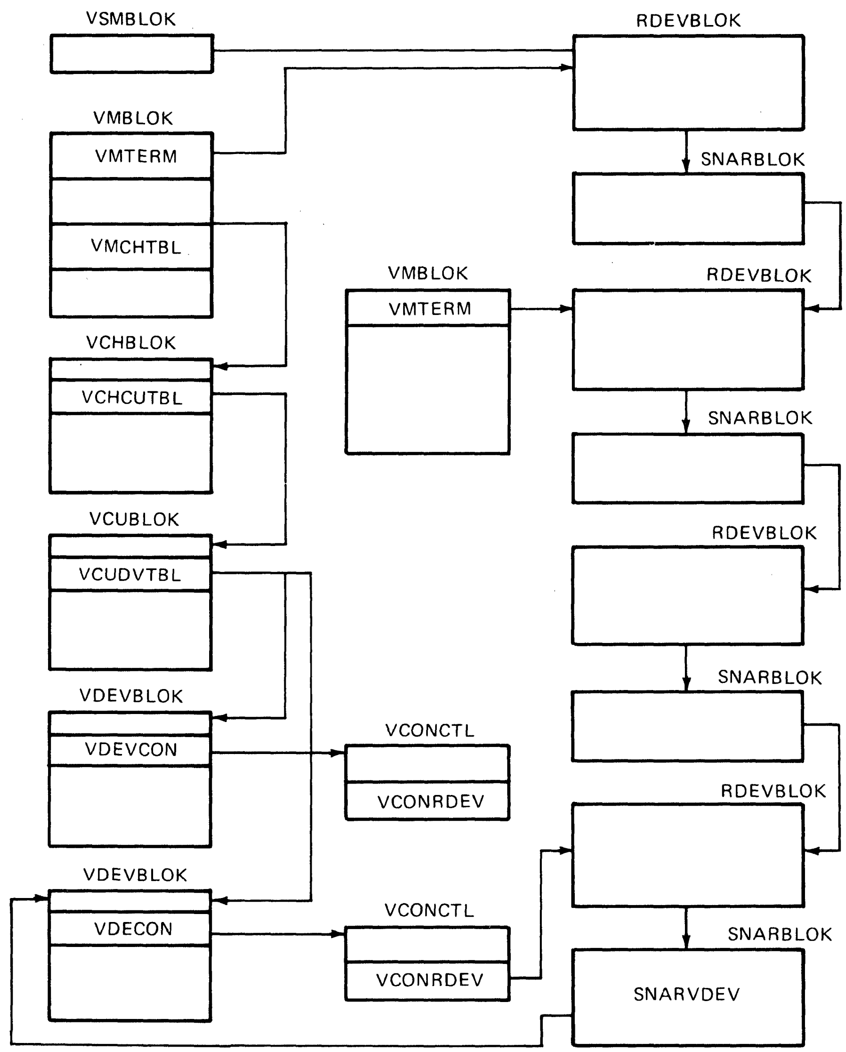


Figure 20. SNA DIAL Control Block Structure

Performance Guidelines

The performance characteristics of an operating system, when it is run in a virtual machine environment, are difficult to predict. This unpredictability is a result of several factors:

- The System/370 model used
- The characteristics of the operating system and its work level
- The total number of virtual machines executing
- The type of work being done by each virtual machine
- The speed, capacity, and number of the paging devices
- The order in which devices are selected for swapping, paging, and spooling
- The amount of real storage available
- The degree of channel and control unit contention, as well as arm contention, affecting the paging device
- The type and number of VM/SP HPO performance options in use by one or more virtual machines
- The degree of access to MSS 3330V volume
- The amount of fixed-head paging storage (drum, 3340, 3350).

Performance of any virtual machine may be improved up to some limit by the choice of hardware, operating system, and VM/SP HPO options. The topics discussed in this section address:

1. The options available to improve the performance of a particular virtual machine.
2. The system options and operational characteristics of operating systems running in virtual machines that will affect their execution in the virtual machine environment.

The performance of a specific virtual machine may never equal that of the same operating system running standalone on the same System/370, but the total throughput obtained in the virtual machine environment may equal or better that obtained on a real machine.

When executing in a virtual machine, any function that cannot be performed wholly by the hardware causes some degree of degradation in the virtual machine's performance. As the control program for the real machine, CP initially processes all real interrupts. A virtual machine operating system's instructions are usually executed in *problem* state.[2] Any privileged instruction issued by the virtual machine causes a real privileged instruction exception interruption. The amount of work to be done by CP to analyze and handle a virtual machine-initiated interrupt depends upon the type and complexity of the interrupt.

The simulation effort required of CP may be trivial, as for a supervisor call (SVC) interrupt (which is generally reflected back to the virtual machine), or may be more complex, as in the case of a Start I/O (SIO) interrupt, which initiates extensive CP processing.

When planning for the virtual machine environment, consider the number and type of privileged instructions to be executed by the virtual machines. Any reduction in the number of privileged instructions issued by the virtual machine's operating system will reduce the amount of extra work CP must do to support the machine.

Directory Considerations

Two techniques involving the user directory will improve system performance by:

- Decreasing the size of the directory
- Decreasing the search time needed to locate a USER control statement when attaching that user's virtual machine.

To decrease the size of the directory, commonly used control statements can be contained in a directory profile. The profile is defined by a PROFILE control statement, and each user entry may reference this profile via an INCLUDE control statement. The advantage gained is that these commonly used statements are defined only once (in the profile), instead of several times (in each user's entry). See the "DMKDIR - The Directory Program" section in the *VM/SP HPO Service Routines Program Logic* for more information.

To decrease the search time needed to locate USER control statements, you should alphabetically sort the directory according to the userid field in the USER control statements. Routines that search the directory list (for example, when logging on or spooling) will do so more efficiently and therefore greatly reduce the search time. For more information, see the *VM/SP HPO Planning Guide and Reference* or *VM/SP HPO CP for System Programming.*

---

2 When preferred machine assist is active, the MVS virtual machine runs in supervisor state in the V=R area and does I/O operations on dedicated channels.

Virtual Machine I/O

To support I/O processing in a virtual machine, CP must translate all virtual machine channel command word (CCW) sequences to refer to real storage and real devices and, in the case of minidisks, real cylinders or blocks. When a virtual machine issues an SIO, CP must:

1. Intercept the virtual machine SIO interrupt

2. Allocate real storage space to hold the real CCW list to be created

3. Translate the virtual addresses referred to in the virtual CCWs to real addresses

4. Page into real storage and lock, for the duration of the I/O operation, all virtual storage pages required to support the I/O operation

5. Generate a new CCW sequence building a Channel Indirect Data Address list if the real storage locations cross page boundaries

6. Schedule the I/O request

7. Present the SIO condition code to the virtual machine

8. Intercept, retranslate, and present the channel end and device end interrupts to the appropriate virtual machine, where they must then be processed by the virtual machine operating system.

CP's handling of SIOs for virtual machines can be one of the most significant causes of reduced performance in virtual machines.

The overhead associated with SIO operations required by a virtual machine can be significantly reduced in several ways:

- Use of large blocking factors (of up to 4096 bytes) for user data sets to reduce the total number of SIOs needed.
- Use of preallocated data sets.
- Use of virtual machine operating system options (such as chained scheduling in OS) that reduce the number of SIO instructions.
- Substitution of a faster resource (virtual storage) for I/O operations, by building small temporary data sets in virtual storage rather than using an I/O device.
- Use of NOTRANS option with dedicated channels for the V=R virtual machine operating system.

Frequently, there can be a performance gain when CP paging is substituted for virtual machine I/O operations. The performance of an operating system such as OS can be improved by specifying as resident as many frequently used OS functions (transient subroutines, ISAM indexes, and so forth) as are possible. In this way, paging I/O is substituted for virtual machine-initiated I/O. In this case, the only work to be done by CP is to place into real storage the page that contains the desired routine or data.

The following performance options are available to reduce the CP overhead associated with virtual machine I/O instructions or other privileged instructions the virtual machine's I/O supervisor uses:

- The virtual=real option eliminates the need for CP to perform storage reference translation and paging before each I/O operation for the V=R virtual machine. Only one V=R virtual machine can be executing at any time.
- The virtual machine assist feature reduces the real supervisor state time used by the control program.
- VM/370 Extended Control Program Support further reduces the real supervisor state time used by the control program.

The section *Virtual Machine Performance Options* describes assigning and using these options.

See the *VM/SP HPO Planning Guide and Reference* for a list of processors that support virtual machine assist and VM/370 Extended Control Program Support. This publication, along with the *VM/SP HPO Installation Guide*, describes procedures for setting up a V=R area and the requirements for using preferred machine assist.

Paging Considerations

When virtual machines refer to virtual storage addresses that are not currently in real storage, they cause a paging exception and the associated CP paging activity.

The addressing characteristics of programs executing in virtual storage have a significant effect on the number of page exceptions experienced by that virtual machine. Routines that have widely scattered storage reference tend to increase the paging load of a particular virtual machine. When possible, modules of code that are dependent upon each other should be located in the same page. Reference tables, constants, and literals should also be located near the routines that use them. Exception or error routines that are infrequently used should not be placed within main routines, but located elsewhere.

CP assigns certain inactive pages to a paging device according to the following rules:

- If the page was referenced during virtual machine execution, and if space is available in a TYPE=SW area, the page is assigned to that area (regardless of whether or not the page was changed during virtual machine execution).
- If the page was referenced and changed during virtual machine execution, and no space is available in a TYPE=SW area, the page is assigned a TYPE=PP area.

- If the page was not referenced during virtual machine execution, it is assigned to a TYPE=PP area.

Note: *If a page is read in from TYPE=SW, it is marked as "changed" because the backup copy in the TYPE=SW area is released once the page is read in.*

Virtual machines that reduce their paging activity by controlling their use of addressable space improve resource management for that virtual machine, the VM/SP HPO System, and all other virtual machines. The total paging load that must be handled by CP is reduced, and more time is available for productive virtual machine use.

The system programmer can gain additional dynamic paging storage by using the SYSCOR macro statement at system generation time. One way he can do this is by reducing the number of free storage page frames and prime storage page frames to be allocated at system load time. The amount of free storage to be allocated is specified in the FREE operand of the SYSCOR macro. The amount of prime storage to be allocated is specified in the PRIME operand of the SYSCOR macro. Reducing free storage, however, is not a good idea on large systems. Another way to gain additional dynamic paging storage is to reduce the size of the trace table (again, by using the SYSCOR macro). See the *VM/SP HPO Planning Guide and Reference* for more information.

Another way the system programmer can add dynamic paging storage is to specify a value of more than 16 Mb in the RMSIZE or RSSIZE operand of the SYSCOR macro. This should be done *only* on processors that can support extended storage. The area between the 16 Mb line and RMSIZE will be used as additional dynamic paging storage; however, these page frames can be used for virtual machine pages only. This area above the 16 Mb line cannot be used for CP pageable pages, virtual page 0, or virtual machine pages requiring service by CP. When CP requires access to a virtual machine page that is resident above the 16 Mb line, that page must be moved to the dynamic paging area below the 16 Mb line before CP can reference it. CP references virtual machine storage for such functions as privileged operation simulation, channel program translation, interrupt reflection, and console functions.

See "Extended Storage Support" under "CP Interrupt Handling" in this part of the manual for more information.

CP provides four performance options (locked pages, reserved page frames, QDROP OFF option, and a virtual=real area) to reduce the paging requirements of virtual machines. Generally, these facilities require some dedication of real storage to the chosen virtual machine and, therefore, improve its performance at the expense of other virtual machines.

Locked Pages Option

The LOCK command, which is available to the system operator (with privilege class A), can be used to permanently fix or lock specific user pages of virtual storage into real storage. In so doing, all paging I/O for these page frames is eliminated.

Since this facility reduces total real storage resources (real page frames) that are available to support other virtual machines, only frequently used pages should be locked into real storage. Since page zero (the first 4096 bytes) of a virtual machine storage is referred to and changed frequently (for example, whenever a virtual machine interrupt occurs or when a CSW is stored), it should be the first page of a particular virtual machine that an installation considers locking. The virtual machine interrupt handler pages are also good candidates for locking.

Other pages to be locked depend upon the work being done by the particular virtual machine and its usage of virtual storage.

The normal CP paging mechanism selects unreferenced page frames in real storage for replacement by active pages. Page frames belonging to inactive virtual machines will all eventually be selected and paged out if the real storage frames are needed to support active virtual machine pages.

When virtual machine activity is initiated on an infrequent or irregular basis, such as from a remote terminal in a teleprocessing inquiry system, some or all of its virtual storage may have been paged out before the time the virtual machine must begin processing. Some pages will then have to be paged in so that the virtual machine can respond to the teleprocessing request compared with running the same teleprocessing program on a real machine. This paging activity may cause an increase in the time required to respond to the request compared with running the teleprocessing program on a real machine. Further response time is variable, depending upon the number of paging operations that must occur.

Locking specific pages of the virtual machine's program into real storage may ease this problem, but it is not always easy or possible to identify which specific pages will always be required.

In general, once a page is locked, it remains locked until one of the following occurs:

- The user logs off
- The system operator (privilege class A) issues the UNLOCK command for that page
- The user re-IPLs his system by device address and specifies the CLEAR option
- The user re-IPLs his system by name (Shared System), and the locked pages are not in the shared segment

- The user re-IPLs his system by name, the locked pages are in the shared segment, and the user who is re-IPLing is the last user of that shared segment
- The user issues DIAGNOSE instruction X'14', X'30', X'34', or X'38' against a locked page.

However, if the user re-IPLs his system by device address and does not specify the CLEAR option, all pages remain locked except the page given to DMKVMI for IPL.

The SYSTEM CLEAR command, when invoked, clears virtual machine storage and unlocks the user's locked pages.

You cannot lock pages into the greater than 16 Mb area. Page frames above this limit are dedicated to the preferred machine assist guest or are used by CP as additional dynamic paging area for user pages only. Pages above the 16 Mb line are moved to the less than 16 Mb area before they are locked.

Shared Pages

In a system generated for attached processor or multiprocessor operation, no shared pages are locked. If the system operator attempts to lock a shared page or an address range containing one or more shared pages, he will receive the message

```
DMKCPV165I Page hexloc not locked; shared page
```

for each of the shared pages within the range.

Reserved Page Frames Option

A more flexible approach than locked pages is the reserved page frames option. This option provides a specified virtual machine with an essentially private set of real page frames, the maximum number of frames being designated by the system operator, when he issues the CP SET RESERVE command line. Pages will not be locked into these frames. The most recently referenced pages, as determined by the core table scan, will be held in storage. When a temporarily inactive virtual machine having this option is reactivated, these page frames are immediately available. If the program code or data required to satisfy the request was in real storage at the time the virtual machine became inactive, no paging activity is required for the virtual machine to respond.

This option is usually more efficient than locked pages in that the pages that remain in real storage are those pages with the greatest amount of activity at that moment, as determined automatically by the system.

The syntax of the SET RESERVE command is:

```
SET RESERVE userid nnnn
```

where nnnn is the maximum number of reserved page frames required. Nnnn can be a value from 1 to 4096. The number of frames reserved will be nnnn or number of pages resident for the user, whichever is smaller.

Note: The sum of all reserved pages should never approach the total available pages, since CP overhead is substantially increased in this situation, and excessive paging activity is likely to occur in other virtual machines.

The reserved page frames user's paging activity is generally consistent from run to run. This can be especially valuable for production-oriented virtual machines with critical schedules, or those running teleprocessing applications where response times must be kept as short as possible.

Note: Multiple machines may have reserved page frames.

QDROP OFF Option

By setting QDROP OFF for a specific virtual machine, DMKSEL will examine that user's resident pages less often during core table scan (and, therefore, those pages will be more likely to remain resident). This increased page residency will improve performance for that user. See the *VM/SP HPO CP Command Reference* for more information on the SET QDROP command.

Virtual=Real Option

The virtual=real option eliminates CP paging for the selected virtual machine. All pages of virtual machine storage, except page zero, are locked in the real storage locations they would use on a real computer. CP controls real page zero, but the remainder of the CP nucleus is relocated and placed beyond the virtual=real machine in real storage. For a more detailed discussion of this option, see "Preferred Virtual Machine Initialization" under "Preferred Machine Assist Feature."

Since the entire address space required by the virtual machine is locked, these page frames are not available for use by other virtual machines except when the virtual=real area has been unlocked. This option often increases the paging activity for other virtual machine users, and in some cases for VM/SP HPO. (Paging activity on the system may increase substantially, since all other virtual machine storage requirements must be managed with fewer remaining real page frames.)

The virtual=real option may be desirable or mandatory in certain situations. The virtual=real option is desirable when running a virtual machine operating system (like DOS/VS or OS/VS) that performs paging of its own because the possibility of double paging is eliminated. The option must be used to allow programs that execute self-modifying channel programs or have a certain degree of hardware timing dependencies to run under VM/SP HPO.

Virtual Machine Performance Options

VM/SP HPO provides several functions that create a special virtual machine environment. The following functions improve the performance of a selected virtual machine:

- Favored execution
- User priority
- Reserved page frames
- Virtual=real option
- QDROP OFF option
- Affinity
- Multiple shadow table support
- Shadow table bypass
- Single processor mode
- Dynamic system control program transition to or from native mode
- Preferred machine assist.

Although these functions can be applied to different virtual machines, they are usually applied to only one if optimum performance is required for that specific virtual machine.

The following functions improve the performance of the VM/SP HPO system. They can be applied to as many virtual machines as desired:

- Reserved page frames
- Virtual machine assist
- VM/370: Extended Control Program Support
- Dual address space assist
- MVS page fault assist.

Favored Execution

The favored execution options allow an installation to modify the normal CP deadline priority calculations in the scheduler to force the system to devote more of its processor resources to a given virtual machine than would ordinarily be the case. The options provided are:

1. The basic favored execution option.
2. The favored execution percentage option.

The basic favored execution option means that the virtual machine so designated is to remain in the dispatch list at all times, unless it becomes nonexecutable. When the virtual machine is executable, it is to be placed in the dispatchable list at its normal priority position. However, any active virtual machine represents either an explicit or implicit commitment of main storage. An explicit storage commitment can be specified by either the virtual=real option or the reserved page frames option. An implicit commitment exists if neither of these options is specified, and the scheduler recomputes the virtual machine's projected work-set at what it would normally have been at queue-drop time. Multiple virtual machines can have the basic favored execution option set. However, if their combined main storage requirements exceed the system's capacity, performance can suffer because of thrashing.

If the favored task is highly compute bound and must compete for the processor with many other tasks of the same type, you can define the processor allocation to be made. In this case, you can select the favored execution percentage option. This option specifies that the selected virtual machine, in addition to remaining in queue, is guaranteed a specified minimum percentage of the total processor time if it can use it.

Note: The percentage of processor time that has been requested via the SET FAVORED command with the percentage option is not an absolute value.

The percentage actually received by the favored user will vary depending on the total load and/or the type of load on the system. Generally, it will remain close to the percentage specified in the command. However, if the run list contains multiple virtual machines that are compute bound, the favored user may not receive the requested percentage of processor time. The favored execution option can only be invoked by a system operator with command privilege class A. The format of the command is as follows:

```
SET FAVORED userid  nnn
                    OFF
```

where:

userid identifies the virtual machine to receive favored execution status.

nnn is any value from 1 through 100 and specifies the percentage of the in-queue time slice that the system will attempt to provide for this virtual machine. In addition, specifying 100 causes the user to be kept at the top of the dispatch list.

OFF specifies that the virtual machine is to be removed from favored execution status.

If a percentage is not specified, a virtual machine with the favored execution option active is kept in the dispatch list except under the following conditions:

- Entering CP console function mode
- Loading a disabled PSW
- Loading an enabled PSW with no active I/O in process
- Logging on or off.

When the virtual machine becomes executable again, it is put back on the dispatch list in Q1. If dropped from Q1, the virtual machine is placed directly in the Q2 dispatch list. If the percentage option of the SET FAVORED command is specified, the deadline priority is calculated at queue drop time by:

```
current time-of-day + length of allowed processor in-queue time slice
                      -----------------------------------------------
                                   favored percentage
```


For example, if the processor in-queue time slice is 1 second, and the specified percentage is 10 percent (1/10), then the value added to the current time-of-day is 10 seconds. The virtual machine should receive one processor time slice (1 second) once every 10 seconds.

Although the SET FAVORED command prevents specifying more than 100% for a particular virtual machine, nothing is done to prevent allocating more than 100% to a number of virtual machines. Where more than 100% has been allocated, the favored virtual machines compete for the available resources on a pro-rata basis. That is, an individual virtual machine's allocation is, roughly, proportional to the percentage allocated to it, divided by the total percentage allocated to all virtual machines. The effect of allocating more than 100% of the system on interactive (Q1) responses is unpredictable.

User Priority

The system operator can assign specific priority values to different virtual machines. In so doing, the virtual machine with a higher priority is allocated a larger share of the system resources before a virtual machine with a lower priority. User priorities are set by the following class A command:

```
SET PRIORITY userid nn
```

where userid is the user's identification and nn is an integer value from 1 to 99. The value of nn affects the user's dispatching priority in relation to other users in the system. The priority value (nn) is one of the factors considered in the calculation of the deadline priority. The deadline priority is the basis on which all virtual machines in the system are ordered on both the eligible list and the dispatch list. The deadline priority calculation is based on the assumption that the average or normal (default) user priority is 64.

Virtual=Real

For this option, the nucleus must be reorganized to provide an area in real storage large enough to contain the entire virtual=real machine. In the virtual machine, each page from page 1 to the end is in its true real storage location; only its page zero is relocated. The virtual machine is still run in dynamic address translation mode, but since the virtual page address is the same as the real page address, no CCW translation is required. Since CCW translation is not performed, no check is made to ensure that I/O data transfer does not occur into page zero or any page beyond the end of the virtual=real machine's storage.

Systems that are generated with the virtual=real option use the system loader (DMKLD00E). For information about generating a virtual=real system, see the *VM/SP HPO Installation Guide*.

Figure 21 is an example of a real storage layout with the virtual=real option. The V=R area is 8Mb and real storage is 32Mb.

There are several considerations for the virtual = real option that affect overall system operation:

1. The area of contiguous storage built for the virtual = real machine must be large enough to contain the entire addressing space of the largest virtual = real machine. The virtual = real storage size that a VM/SP HPO system allows is defined during system generation when the option is selected.

2. The storage reserved for the virtual = real machine can only be used by a virtual machine with that option specified in the system directory. It is not available to other users for paging space, nor for VM/SP HPO use until released from virtual = real status by a system operator via the CP UNLOCK command. Once released, VM/SP HPO must be loaded again before the virtual = real option can become active again.

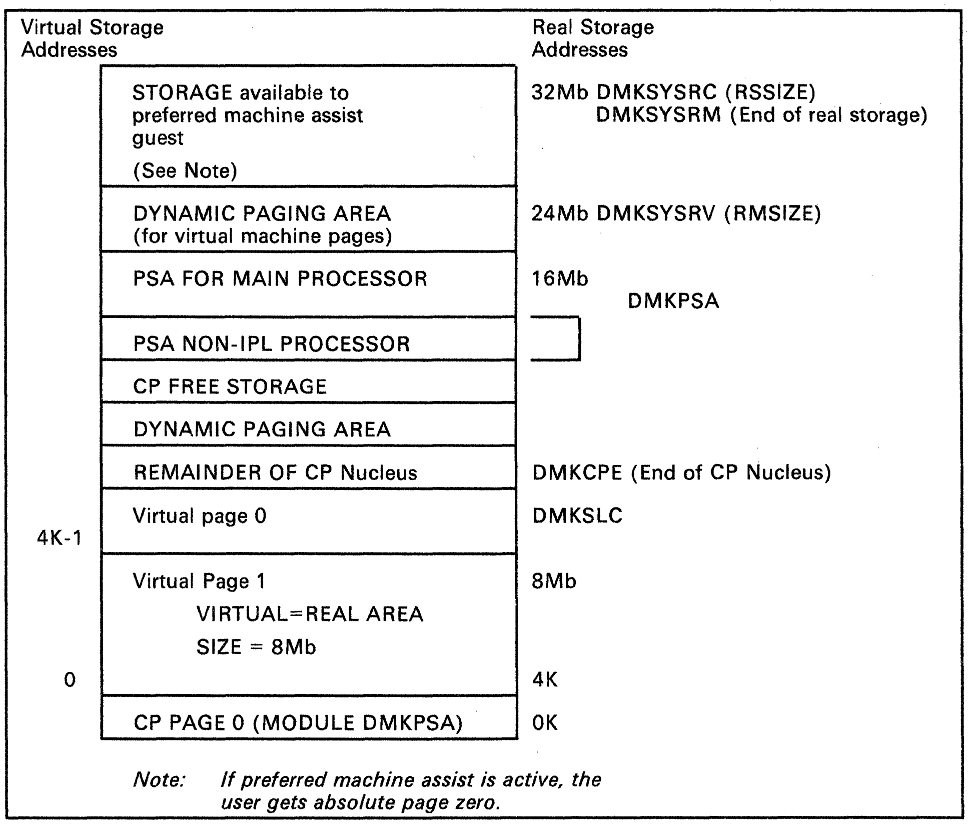


Figure 21. Storage Layout in a Virtual = Real Machine with Extended Storage Support

3. The virtual machine with the virtual = real option operates in the preallocated storage area with normal CCW translation in effect until the CP SET NOTRANS ON command is issued. At that time, with several exceptions, all subsequent I/O operations are performed from the virtual CCWs in the virtual = real space without translation. The exceptions occur under any of the following conditions:

   - SIO tracing active
   - First CCW not in the V = R region

- I/O operation is a sense command
- I/O device is a dial-up terminal
- I/O is for a nondedicated device (spooled unit record console virtual CTCA or minidisks that are less than a full volume)
- Pending device status
- I/O device has an alternate path.

Any of the above conditions will force CCW translation. Since minidisks are nondedicated devices, they may be used by programs running in the V=R region even though CP SET NOTRANS ON is in effect.

4. If the virtual=real machine performs a virtual reset or IPL, then the normal CCW translation goes into effect until the CP SET NOTRANS ON command is again issued. This permits simulation of an IPL sequence by CP. Only the virtual=real virtual machine can issue the command. A message is issued if normal translation mode is entered.

5. A virtual=real machine is not allowed to IPL a named or shared system. It must IPL by device address.

6. When NOTRANS is in effect for a virtual=real machine, no meaningful SEEK data is collected by MONITOR operations.

7. The reliability and availability of an operating system running virtual=real on a 308x, 3090, or 4381 processor can be enhanced when the TEST BLOCK instruction (TB) is used to validate the V=R storage area.

Affinity

This virtual machine option allows virtual machines that operate on attached processor systems or multiprocessor systems to select, if desired, the processor of their choice for program execution. The selection can be made by the system directory OPTION statement, or it can be made dynamically by an operand of the CP SET command:

For class G users

```
SET AFFINITY   nn
               OFF
```

For class A users

```
SET AFFINITY userid   nn
                      ON
                      OFF
```

where nn is the processor address of a processor in an attached or multiprocessor configuration.

In application, the affinity setting of a virtual machine implies a preference of operation to either (or neither) processor. Affinity of operation for a virtual machine means that the program of that virtual machine will be executed on the selected or named processor. It does not imply that supervisory functions and the CP housekeeping functions associated with that virtual machine will be handled by the same processor.

In attached processor systems, all real I/O operations and associated interrupts are handled by the main processor. Virtual I/O initiated on the attached processor that is mapped to real devices must transfer control to the main processor for real I/O execution. Therefore, benefits may be realized in a virtual machine "mix" by relegating those virtual machines that have a high I/O-to-compute ratio to the main processor, and those virtual machines that have a high compute-to-I/O ratio to the attached processor. Such decisions should be carefully weighed as every virtual machine is in contention with other virtual machines for resources of the system.

In multiprocessor configurations, both processors have the capability of executing real I/O. If the path to a user's primary minidisk or to a user's dedicated volumes is configured asymmetrically to one processor, performance benefits for the virtual machine may be derived by setting the virtual machine's affinity to that processor.

A more important use of the affinity setting would be in applications where there are virtual machine program requirements for special hardware features that are available on one processor and not the other. Such features could be a performance enhancement such as virtual machine assist (described later in the text) or a special RPQ that is a requirement for a particular program's execution.

Multiple Shadow Table Support

To reduce the number of shadow table purges when the virtual machine changes control register 1 (CR1) values, CP maintains a queue of segment table origins (STO) and associated shadow tables for the virtual machine. Thus, each time an MVS or SVS system dispatches a new address space (changes CR1), CP can use the proper shadow table.

Multiple shadow table support adds one control block to CP, the segment table origin control block (STOBLOK) pointed to by the ECBLOK. The STOBLOK, created by DMKVAT, contains all information pertaining to the shadow segment table, the shadow segment table itself and the virtual CR1 value. It also provides forward and backward queue pointers to the next STOBLOK on the queue. The first STOBLOK on the queue always contains the shadow STO to be loaded into CR1 when the virtual machine is dispatched in translation mode. The queue of STOBLOKs is maintained by DMKVAT in the following manner:

1. If the virtual machine loads a new CR1 value, DMKVATAB searches the queue of STOBLOKs for the virtual CR1 value.
2. If the proper STO is found, the STOBLOK is ordered first on the queue.
3. If the proper STO is not found, the maximum STO count is checked.

4. If the number of STOBLOKs equals the maximum STO count, DMKVATAB steals the last STOBLOK, purges the shadow tables, and reinitializes and reuses the STOBLOK by chaining it first on the queue.

5. If the number of STOBLOKs is less than the maximum STO count, then free storage is obtained from VM/SP HPO, and the STOBLOK is reinitialized and chained first on the queue.

Multiple shadow table support is controlled by the SET STMULTI command.

Shadow Table Bypass

Shadow table bypass is controlled by the SET STBYPASS command.

Note: If virtual machine assist is enabled on the system, the virtual machine must have the STFIRST directory option to be allowed to issue the SET STBYPASS command.

Shadow Table Bypass for the V=V User

This technique is based on several characteristics of VS systems:

1. VS systems have a large area of addressing space starting with location zero where the virtual address is equal to the real address.

2. This addressing space is common to each segment table when multiple segment tables are used (MVS or SVS address space).

3. The VS system never pages within this fixed area.

Thus, an area starting with location zero can be established where the second-level address equals the third-level address or virtual-virtual=virtual-real (VV=VR). This allows a high-water mark, the highest VV=VR address, for a VS system to be established. Because the second-level address is the same as the third-level address, a reverse translation allows the shadow tables to be indirectly indexed. Then, whenever VM/SP HPO steals a page from the VV=VR area, it invalidates the shadow page table entry and executes a PTLB instruction before redispatching the VS system's virtual machine.

In addition, whenever a shadow table is purged because a page frame was stolen from above the high-water mark or because the virtual machine executed a PTLB or LCTL instruction, the invalidation starts above the high-water mark, thus reducing purge and revalidation time.

Shadow Table Bypass for the V=R User

Using V=R shadow table bypass, you can eliminate both the shadow tables and the overhead associated with maintaining them. Do this by modifying the virtual operating system's page table to relocate virtual page zero to the highest real address within the V=R area. Then dispatch the virtual machine with control register 1 pointing to its own virtual page and segment tables.

Single Processor Mode

In multiprocessor (MP) and attached processor (AP) systems, single processor mode allows you to dedicate a processor to an MVS V=R virtual machine. With single processor mode active, VM/SP HPO executes in uniprocessor mode on the IPL processor while the V=R virtual machine, which continues to operate under VM/SP HPO, has exclusive control of the non-IPL processor for AP or MP operations.

You activate single processor mode for the system by issuing the CP command SPMODE ON, but only after the processor has been varied off to VM. Before using single processor mode, you should generate VM/SP HPO as an AP or MP system with a V=R area defined. You can generate VM/SP HPO as a UP system if you have the MP feature installed on the machine. CP issues an error message if you violate any of the requirements.

Virtual Prefixing

When your VM/SP HPO AP or MP system is operational, CP supports virtual prefixing before you activate single processor mode. This allows you to use the functions of the shadow table bypass assist by creating an environment in which the MP feature is made available to the V=R guest even though only one processor (the VM/SP HPO IPL processor) is online to the guest.

CP performs virtual prefixing by modifying the VMSEG page table entry, segment tables, and shadow tables. CP stores the prefix value for the V=R virtual machine in segment zero of the page table entry for virtual page zero. Thus, when the V=R virtual machine is not in translate mode (the translation bit is off in the EC mode PSW), access to virtual page zero is through the prefix value that is already stored in the page table entry for the V=R virtual machine. If the V=R virtual machine references its virtual prefix page, absolute page zero is accessed through reverse prefixing.

When the V=R virtual machine is in translate mode, virtual prefixing is accomplished through a shadow segment table. This table is essentially a copy of the segment table for the virtual machine. CP maintains a private shadow page table for segment zero and for the segment containing the virtual prefix page. The shadow page table entries for virtual page zero and the virtual prefix page are maintained in the same way as they are maintained in the VMSEG segment tables.

Since shadow tables are necessary to simulate virtual prefixing, the SET STBYPASS VR command is effectively treated as a no-operation function. Use the SET STMULTI command to minimize shadow table maintenance overhead. VM/SP HPO support for virtual prefixing allows CP to simulate the MP instructions CONCS, STAP, SPX, STPX, and SIGP even though single processor mode is not activated.

Real Prefixing

When you activate single processor mode (using the SPMODE ON command), CP sets the VMSEG page table entry that corresponds to the virtual prefix page to access absolute page zero. When the virtual prefix is not zero, the page table entry for virtual page zero is set to the virtual prefix page. When the virtual prefix is zero, the page table entry for virtual page zero is set to access absolute page zero.

If the shadow table bypass assist feature is present on the processor, CP activates the STNOSM and STOSM functions of the assist by setting the appropriate control bits in the MICBLOK. The V=R virtual machine running in single processor mode can issue the SET STBYPASS command to set a high-water mark, so you do not have to do so using the STFIRST directory option.

If you issue SPMODE OFF and the V=R virtual machine has a nonzero prefix, the system returns to virtual prefixing. CP adjusts the page table entries in the VMSEG tables to access virtual page zero instead of absolute page zero and reestablishes shadow tables. The shadow table bypass assist bits are set to zero in the MICBLOK.

When a PSW RESTART function is invoked under normal use, it forces VM/SP HPO to take a storage dump (PSA002). When the V=R virtual machine is running in single processor mode, the PSW RESTART function causes a restart interrupt to the V=R virtual machine. If you require a forced storage dump (PSA002) while running in single processor mode, set a flag byte in DMKPSA before invoking the PSW RESTART function.

For information about the commands used in this support, see the *VM/SP HPO CP Command Reference*. A full description of the operating procedures for this support is contained in *VM/SP HPO Operating Systems in a Virtual Machine*.

Dynamic SCP Transition to or from Native Mode

This function allows the operating system running in the V=R virtual machine to make a transition from the VM/SP HPO environment to the native environment and back again. It eliminates the necessity of system shutdown and reinitialization for the operating system and VM/SP HPO.

To make the transition to native mode, the indicated users must perform the following steps:

- All I/O used by the V=R virtual machine must be dedicated and the virtual I/O addresses must be the same as the real I/O addresses.
- The virtual machine operator must stop the virtual spooling devices (nondedicated) and detach them from the virtual machine.
- The system operator must drain the real unit-record devices.
- All users, except the V=R virtual machine and the system operator, must be logged off.

When these conditions are met, the system operator can issue the CP command

```
QVM userid
```

The VM/SP HPO machine's page zero is saved, and the V=R virtual machine's page zero is moved to absolute page zero. The timers, control registers, and general registers are initialized to the virtual machine's values. Control is then given to the V=R virtual machine in native mode.

The operating system in the V=R virtual machine is not given native control if any of the following conditions are detected:

- CP wait conditions are present
- I/O interruptions are pending
- External interruptions are pending
- I/O requests are queued within VM/SP HPO
- An ADSTOP has been set or tracing is active.

The transition to native mode is handled by the DMKQVMRT subroutine in the DMKQVM module.

This support also allows the transition from native mode back to VM/SP HPO. When the V=R virtual machine is given native control, the RESTART PSW is modified to point to an entry point in the DMKQVM module.

The system operator must set a flag byte within DMKQVM and perform a PSW RESTART function. This gives control to a subroutine in the DMKQVM module (DMKQVMRS or DMKQVMRX for 370E systems), and the transition from native mode back to the VM/SP HPO environment is performed. After this transition is completed, the operating system is running in the V=R virtual machine again.

When making the transition back to the VM/SP HPO environment, the operating system operator should ensure that the operating system's I/O configuration and environment are the same as when the transition to native mode took place. If a PSW RESTART function is invoked while running in native mode and without setting the flag byte to return to VM/SP HPO, a restart interruption is reflected to the native system.

Information about the commands used in this support is found in the *VM/SP HPO CP Command Reference.* A full description of the operating procedures for this support is contained in the *Virtual Machine Running Guest Operating Systems.*

Virtual Machine Assist Feature

The virtual machine assist feature is a processor hardware feature. It improves system performance. Virtual storage operating systems, which run in problem state under CP, use many privileged instructions and SVCs that cause interrupts that CP must handle. With the virtual machine assist feature, many of these interrupts are intercepted and handled by the processor; and, consequently, performance is improved. See the functional characteristics manual for your processor model to see if the virtual machine assist feature is available.

The virtual machine assist feature intercepts and handles interrupts caused by SVCs (other than SVC 76), invalid page conditions, and several privileged instructions. An SVC 76 is never handled by the assist feature; it is always handled by CP.

Although the assist feature was designed to improve system performance, virtual machines may see a performance improvement because more resources are available for virtual machine users.

Using the Virtual Machine Assist Feature

Whenever you IPL VM/SP HPO on a processor with the virtual machine assist feature, the feature is available for all virtual machines. However, the system operator's SET command can make the feature unavailable to CP and, subsequently, available again for all users. The format of the system operator's SET command is:

```
SET SASSIST  ON   [[PROC] xx]
             OFF
```

If you do not know whether or not the virtual machine assist feature is available, use the class A and E QUERY command. For a complete description of the Class A and E QUERY and SET commands, see the *VM/SP HPO CP Command Reference.*

If the virtual machine assist feature is available when you log on your virtual machine, it is also supported for your virtual machine. If your directory entry has the SVCOFF option, the SVC handling portion of the assist feature is not available when you log on. The class G SET command can disable the assist feature (or only disable SVC handling). It can also enable the assist feature, or if the assist feature is available, enable the SVC handling. The format of the command is:

```
SET ASSIST  [ON] [SVC  ] [TMR  ]
                 [NOSVC] [NOTMR]
             OFF
```

You can use the class G QUERY SET command line to find whether you have full, partial, or none of the assist feature available. For a complete description of the Class G QUERY and SET commands, see the *VM/SP HPO CP Command Reference.*

Restricted Use of the Virtual Machine Assist Feature

Certain interrupts must be handled by CP. Consequently, the assist feature is not available under certain circumstances. CP automatically turns off the assist feature in a virtual machine if it:

- Has an instruction address stop set (ADSTOP).
- Traces SVC and program interrupts.

Since an address stop is recognized by an SVC interrupt, CP must handle SVC interrupts while address stops are set. Whenever you issue the ADSTOP command, CP automatically turns off the SVC handling portion of the assist feature for your virtual machine. The assist feature is turned on again after the instruction is encountered and the address stop removed. If you issue the QUERY SET command line while an address stop is in effect, the response will indicate that the SVC handling portion of the assist feature is off.

Whenever a virtual machine issues a TRACE command with the SVC, PRIV, BRANCH, INSTRUCT, or ALL operands, the virtual assist feature is automatically turned off for that virtual machine. The assist feature is turned on again when the tracing is completed. If the QUERY SET command line is issued while SVCs or program interrupts are being traced, the response will indicate the assist feature is off.

Extended Control-Program Support (ECPS) for VM/370

Extended Control-Program Support for VM/370 (ECPS:VM/370) improves the performance of the processor when executing VM/SP HPO beyond the improvement attained by the virtual machine assist feature described above. ECPS:VM/370 consists of three parts: CP assist, expanded virtual machine assist, and virtual interval timer assist.

CP Assist

The control program assist part of ECPS assists various CP routines that are frequently used. Because these routines are assisted by the hardware without involving VM/SP HPO, performance is improved.

Expanded Virtual Machine Assist

Expanded virtual machine assist handles the processing of additional instructions not handled by the virtual machine assist feature.

Virtual Interval Timer Assist

Virtual interval timer assist provides hardware updating of the virtual interval timer at virtual location X'50'. This results in an update frequency of approximately 300 times per second, the same as for the real interval timer. Procedures that use the virtual interval timer for job accounting, performance measurements, and the like, will therefore generate more accurate and repeatable time data than they would if the virtual timer was being updated by CP routines. Timer updating occurs only while the virtual machine is in control of the real processor.

Using the Extended Control-Program Support:VM/370

Extended Control-Program Support (ECPS) is controlled at two levels: the system and the virtual machine.

At the system level, ECPS is selectively enabled when the system is loaded. Those parts of the assist microcode installed on your processor that are inconsistent with the software logic are disabled. The class A command:

```
SET CPASSIST OFF
```

will disable both CP assist and expanded virtual machine assist. The class A command:

```
SET SASSIST OFF
```

disables only the expanded virtual machine assist part of ECPS as well as the virtual machine assist. CP assist is the only part of ECPS that is truly independent.

At the virtual machine level, whenever ECPS is enabled on the system, both expanded virtual machine assist and virtual interval timer assist are automatically enabled when you log on. If you issue the class G command:

```
SET ASSIST OFF
```

both assists as well as the existing virtual machine assist are disabled. If you issue:

```
SET ASSIST NOTMR
```

only the virtual interval timer assist is disabled. If CP assist is disabled for the system, the class A command:

```
SET SASSIST ON
```

will enable the virtual machine assist. You can then enable virtual machine assist and virtual interval timer assist for your virtual machine by issuing the class G command:

```
SET ASSIST ON TMR
```

Restricted Use of ECPS

The restrictions on the use of ECPS are the same as those described for the virtual machine assist feature with two additions. Whenever a virtual machine traces external interrupts, the virtual interval timer assist is automatically disabled. When external interrupt tracing is completed, virtual interval timer assist is reenabled. Also, parts of ECPS are disabled when the FRET trap logic is present.

Preferred Machine Assist Feature

Preferred machine assist is a machine feature that improves the performance of the MVS/System Product V=R virtual machine. It allows the MVS/SP V=R virtual machine to operate in real supervisor state. In real supervisor state, the MVS/SP virtual machine uses dedicated channels to directly control its I/O operations. This support eliminates simulation of most I/O operations by CP. I/O operations from the preferred virtual machine to nondedicated channels (controlled by CP), however, are routed to CP for processing. The preferred machine assist support also allows the MVS/SP Release 1.3 or later V=R virtual machine to utilize greater than 16Mb of real storage.

When a processor is running with greater than 16 megabytes of storage online, the preferred machine assist guest can use some, all, or none of the storage above the 16 Mb line, depending on the storage configuration specified by the system programmer at system generation time. See the *VM/SP HPO Planning Guide and Reference* for more information.

Hardware and software requirements for using preferred machine assist are not presented here. For more information about these requirements, see the *VM/SP HPO Planning Guide and Reference.*

Preferred Machine Assist Control Switch Assist

Control switch assist offers a variant of preferred machine assist. It is invoked by entering the PMAV parameter on the IPL command. Under control switch assist, certain CP Diagnose and IUCV commands are supported. They are handled as on any nonpreferred machine assist machine.

In addition, with control switch assist installed, a control switch occurs for a preferred machine assist virtual machine (in this case regardless of whether the PMAV parameter has been entered on the IPL command) when that machine becomes enabled for I/O interrupts (from a guest-owned device on a CP-owned channel). This control switch allows CP to gain control and reflect the I/O interrupt to the preferred machine assist virtual machine.

Control switch assist is not available on the 3033 processor. For more information about control switch assist and its requirements, see the *VM/SP HPO Planning Guide and Reference.*

System Initialization

The system initialization process establishes the preferred machine assist environment. Prefixing starts even when the system is generated as a uniprocessor.

CP sets up the preferred machine assist environment as follows:

Module DMKCPI determines whether the system can operate in preferred machine assist mode (it checks for DMKPMA in the load list). If present, DMKPMA checks for module DMKSLC in the load list and verifies the existence of the preferred machine assist feature on the processor. If these requirements are met, DMKPMA initializes fields in the PSA and other preferred machine assist-related areas.

DMKPMA uses the routine at entry point DMKPMAI1 to initialize the PMAAVAIL field in the PSA. If PMAV is specified on the IPL command, and the control switch assist is installed, then the VPMAAVAI field is also set on in the PSA. DMKPMAI1 also examines the contents of the Real Channel Index Table to determine the channels that are not known to CP. (Any channel that was not defined to CP at system generation is assumed to be dedicated to the preferred machine assist virtual machine.) DMKPMA builds two channel masks— one for the system and the other for the preferred virtual machine. Preferred machine assist uses the masks to route I/O operations either directly to devices on dedicated channels or to CP, for processing certain privileged operations when requested by the preferred machine assist virtual machine, and for routing I/O interrupts.

DMKPMA initializes at entry point DMKPMAI2 the fields used by routines that process privileged operations. DMKPMAI2 also contains routines that give the preferred virtual machine control of absolute page 0 and initialize the prefix registers for the processor. Control returns to DMKCPI to complete system initialization.

Preferred Virtual Machine Initialization

When the virtual machine user issues the IPL command with the preferred machine assist or preferred machine assist V option, CP activates preferred machine assist if the following are true:

- Preferred machine assist (with or without control switch assist) hardware is present.
- CP is AP or MP, the V=R user has affinity set.
- The MVS/SP virtual machine is operating V=R. CP ensures that the contents of register 11 are the same as the AVMREAL field in the PSA.
- The directory option PMA is in the directory entry for the virtual machine. If specified, PMA must immediately follow the V=R directory option. To determine whether this requirement is met, CP examines the VMPMENAB field in the VMBLOK for the virtual machine.
- The preferred virtual machine is operating in extended control mode. To determine whether this requirement is met, CP examines the VMV370R field in the VMBLOK for the virtual machine.
- Minidisks used by the preferred virtual machine are full extent virtual disks.
- To ensure data integrity, devices dedicated to the preferred virtual machine (including full extent virtual disks) must meet the following requirements:
  - Virtual and real addresses must be the same or there should be no RDEVBLOK for the device specified by the virtual address
  - Virtual channel addresses must correspond to channels controlled by CP.
- Devices that are not dedicated to the preferred virtual machine (excluding full extent virtual disks) must meet the following requirements:
  - There should be no RDEVBLOK for the device specified by the virtual address
  - Virtual channel addresses must correspond to channels controlled by CP.

CP calls module DMKPMA (at entry point DMKPMACD) to verify virtual device requirements. If the preferred virtual machine uses a device that violates a virtual device requirement, an error message results and preferred machine assist is not activated.

At virtual machine initialization, CP examines the IPL command and options. Module DMKCFG processes the IPL command at entry point DMKCFGIP. If the IPL command with the preferred machine assist option is correct and all other requirements for using preferred machine assist are met, DMKCFG builds parameters in page 0 for the virtual machine. These parameters allow the system and the IPL simulator to properly initialize and execute the preferred virtual machine. DMKCFG places the following information in page 0:

Location	Contents
00	Virtual IPL cylinder or block number
08	Address of the IPL device
0A	VMMLEVEL field from VMBLOK for the virtual machine
0C	Flag bits X'80' – STOP requested X'40' – ATTN requested X'20' – Preferred machine assist requested X'10' – Preferred machine assist with control switch assist requested
0E	VDEVTYPC for IPL device
0F	VDEVTPE for IPL device
10	Virtual console address

After initializing page 0, DMKCFG performs the following for the preferred virtual machine:

- Sets on NOTRANS (by initializing to 1 the field VMNOTRAN in VMPSTAT).
- Indicates preferred machine assist is active in VMBLOK (using preferred machine assistON field in VMMCR6 and VMPMUSER).
- Disables virtual interval timer, real timer, Virtual Machine Assist, CP assist, System/370 Extended Facility, and the dual address space facility (cross memory).
- Initializes to 1 the PMAMODE field in PSA. (This field is initialized in both PSAs for AP/MP systems.)
- Initializes to 0 the real address of virtual page 0 and the pointer to VMSEGTBL. CP relocates the page at DMKSLC-4096 to absolute page 0, allowing the preferred virtual machine to control absolute page 0.

DMKCFG passes control to DMKVMI to IPL the virtual machine. DMKVMI saves the parameters and passes control to the preferred virtual machine.

Dual Address Space Assist

MVS/System Product cross-memory services increase the efficiency of communication between address spaces. Using cross-memory, a program operating in the primary address space can pass control directly to another program operating in the secondary address space. Thus, an MVS/SP virtual machine can use cross-memory services for data movement, data access, and program sharing.

The Dual Address Space Assist is a processor microcode enhancement that allows MVS to use XMEM services while running in a virtual machine. Dual Address Space Assist extends the capability of programs to communicate between address spaces through enhanced data movement and program calling procedures. Under VM/SP HPO, MVS/SP cross-memory services are improved for 3033 processors equipped with the 3033 Extensions feature (an architectural extension to the processor). If this assist is installed on the processor (or both processors of an AP or MP system), you activate it for the system by issuing the CP command SET S370E ON XMEM. Activate the assist for the MVS/SP virtual machine by issuing the CP command SET 370E ON XMEM. If the directory option XMEM is entered in the directory entry for the MVS/SP virtual machine, the cross-memory assist is available when the virtual machine logs on.

MVS Page Fault Assist

MVS Page Fault Assist is an MVS-oriented performance enhancement in the form of hardware microcode. It is a modification to the existing hardware-detected Page Translation Exception process that reduces the amount of time required to process the interrupt. When the hardware determines that a page translation exception interrupt should be generated, it attempts to resolve the page exception without generating an interrupt. Page Fault Assist attempts to resolve first-reference page exceptions, which occur when a program makes a reference to a page that has not been referenced since it was allocated or released from storage. It attempts to resolve it by allocating a 4K real frame cleared to zeros. If the page exception is not a first reference, Page Fault Assist generates a page exception.

MVS Page Fault Assist works in conjunction with the Dual Address Space Assist. MVS Page Fault Assist is supported for MVS/System Product Release 3 systems on 3033 processors equipped with the 3033 Extensions feature. The STBYPASS VR option activates MVS Page Fault Assist for the virtual machine.

Virtual Machine Communication Facility

The Virtual Machine Communication Facility (VMCF) allows any logged-on user to transfer messages, control data, data files, or combinations of all three to another virtual machine running under the same VM/SP HPO system. Information is transferred directly from one virtual storage to the other virtual storage with CP buffering the information. Only one data page frame must be locked at any one time. The amount of data that can be transferred is limited only by the virtual storage sizes of the virtual machines involved.

VMCF contains five data movement and seven control functions and is invoked by a virtual machine via the DIAGNOSE interface (code X'0068'). A special external interrupt code, X'4001', notifies a virtual machine that a VMCF communication is pending. A virtual machine can have a maximum of 50 messages active at any one time. The number of messages is an equate in the DMKVMC module and can be changed to accommodate different storage sizes.

VMCF Diagnose Interface

When a virtual machine issues a DIAGNOSE instruction with a function code of X'0068', the Rx register contains the virtual address, doubleword-aligned, of a 40-byte parameter list. This parameter list (VMCPARM) contains a hexadecimal code to identify the specific VMCF subfunction. It also contains the data addresses, data lengths, and control information that are required to execute the particular subfunction.

The DIAGNOSE instruction, a privileged operation, is processed by DMKPRV, which passes control to DMKHVC, the DIAGNOSE interface module. DMKHVC, in turn, validates the function code and, if the code is X'0068', turns control over to DMKVMC, the VMCF module. DMKVMC validates the VMCPARM address and length, the subfunction code, and passes control to the appropriate subroutine. The VMCF subfunctions and their codes are as follows:

Code	Subfunction
X'0000'	Allow virtual machine communication
X'0001'	Disallow virtual machine communication
X'0002'	Initiate a SEND request
X'0003'	Initiate a SEND/RECV request
X'0004'	Initiate a SENDX request
X'0005'	Accept data from a SEND or SEND/RECV request
X'0006'	Cancel specific request you initiated
X'0007'	Reply to a SEND/RECV request
X'0008'	Quiesce incoming communications
X'0009'	Resume accepting communications
X'000A'	Notify a user that you are ready for communications
X'000B'	Reject a specific incoming communication

Special VMCF External Interrupt

Whenever a source virtual machine uses VMCF to correspond with another virtual machine (sink), the sink is notified of the pending communication via a special external interrupt (code X'4001'). When this interrupt is unstacked and processed, a copy of the information in the source's parameter list is passed to the sink in an external interrupt buffer. The buffer is defined when a user allows virtual machine communication. The contents are referred to as the external interrupt message header.

When certain transactions (SEND, SEND/RECV, SENDX) have been completed, a final response external interrupt is passed back to the source. The message header associated with this interrupt contains residual counts pertaining to the transferred data and data transfer return codes.

VMCF Control Blocks and Data Areas

Figure 22 shows the relationship between the various VMCF control blocks and data areas. When a virtual machine allows virtual machine communication, VMCF generates a master VMCBLOK and places it at the head of a queue pointed to by the VMCPNT field of the user's VMBLOK. Two fields in this master VMCBLOK define the address (VMCVADA) and length (VMCLENA) of the user's external interrupt buffer. The length must include the maximum size of any potential SENDX data in addition to the 40 bytes for the external interrupt message header.

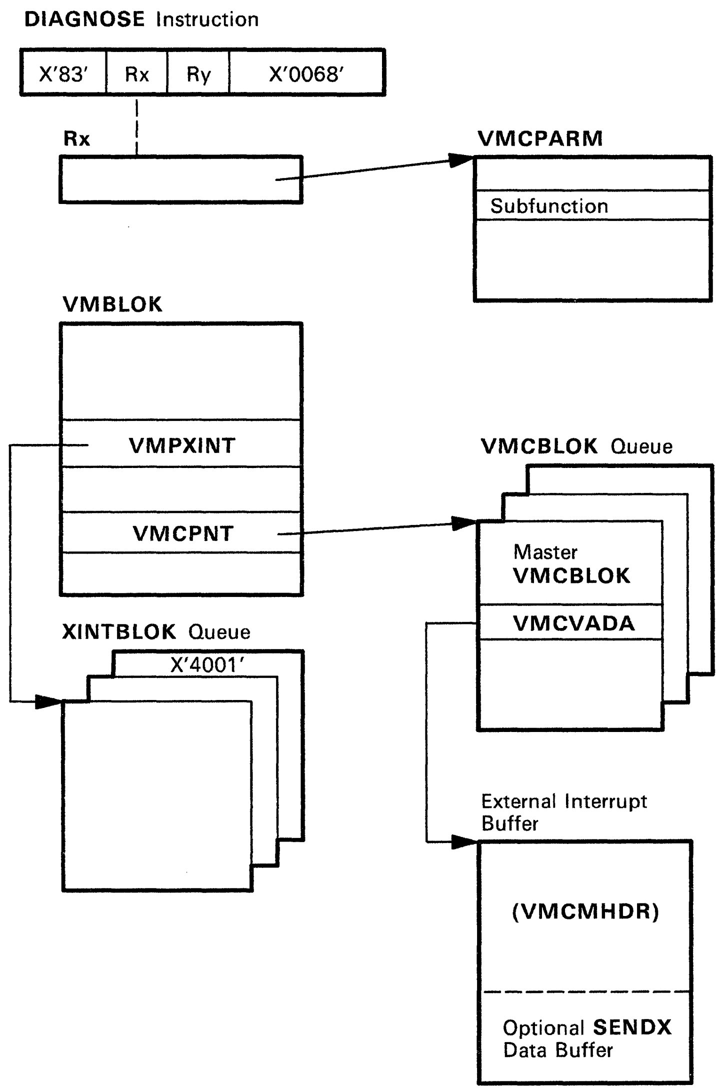


Figure 22. VMCF Control Block Relationships

When a source virtual machine executes a VMCF subfunction, a VMCBLOK is built, initialized with data from the parameter list (VMCPARM), and stacked on the VMCBLOK queue pointed to by the VMCPNT field in the sink's VMBLOK. If an XINTBLOK for a X'4001' external interrupt has not already been stacked for the sink machine, DMKVMC builds one and stacks it on the XINTBLOK queue pointed to by the VMPXINT field in the sink's VMBLOK. VMCF external interrupts are assigned a sort code of X'7FFFFFFF', giving them the lowest priority in the external interrupt queue. Each virtual machine clears its own VMCF control blocks.

Special Messages Facility

The Special Messages Facility allows a user on one virtual machine to send special messages to another virtual machine via the SMSG command. The Special Messages Facility may be used with the Virtual Machine Communications Facility (VMCF) or the Inter-User Communications Facility (IUCV) to send the messages. In the Special Message environment, CP acts as a source machine with the receiver of the special message being the sink. This relieves the burden from the issuer of SMSG of having to perform authorization and other setup necessary for sending messages to the receiving virtual machine. Authorization is performed by CP.

The issuer of SMSG is responsible for sending message text that is meaningful to the receiving virtual machine. The format and handling of special messages is entirely up to the receiving machine, which may be one designed by the installation or prepared by others.

Before the receiving virtual machine can accept special messages, it must be running with the Special Message flag ON, and it must have issued AUTHORIZE (via DIAGNOSE X'68') with CP. The authorization includes supplying the External Interrupt Buffer address and size. To ensure receiving the entire message, the receiving virtual machine should specify the size as 280 bytes (room for a 40-byte header and a 240-byte message buffer).

Note: A 'MSG TOO LARGE' condition may occur if you issue an SMSG command on a 3279 or 3278 Model 5 terminal to send a message that is longer than the message length the receiving virtual machine has specified.

Set SMSG ON by setting on the SMSG flag in the VMCF parameter list when issuing an AUTHORIZE. You may also issue the CP command SET SMSG ON. Either method sets the Special Message flag on in the VMBLOK. When this is done, any other virtual machine can issue the SMSG command to the userid of the receiving virtual machine.

Before the receiving virtual machine can receive special messages via IUCV, it must do the following:

- Enable itself to receive external interrupts
- Set bit 30 of control register 0 to a value of one
- Issue the IUCV DECLARE BUFFER function

- Issue the IUCV CONNECT function to the CP Message System Service
- Turn on the special message flag by issuing the class G command SET SMSG IUCV.

If the receiving virtual machine chooses not to accept special messages at any time, it can merely issue SET SMSG OFF. CP would then inform any machine issuing the SMSG command that the virtual machine is not receiving special messages. When it is ready to resume accepting special messages, the virtual machine need only to issue SET SMSG ON.

Figure 23 shows the processing when an SMSG command is issued.

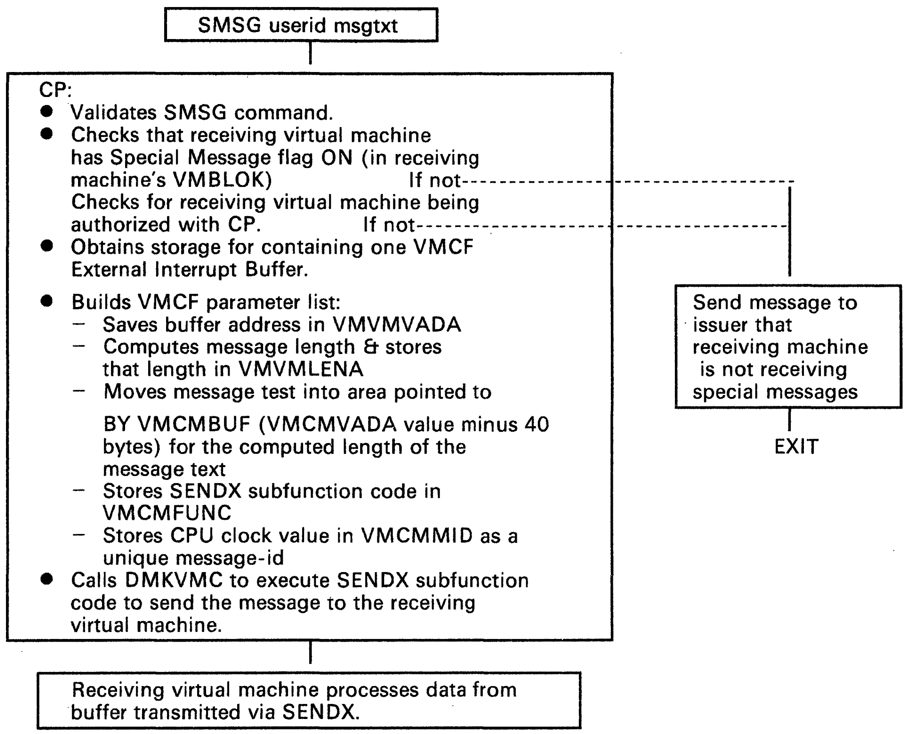


Figure 23. SMSG Command Processing

Inter-User Communications Vehicle

There are 17 IUCV functions with which you can initiate, interrogate, receive, reply to, and terminate individual communications. You can interrogate, receive, or terminate any communication over a particular path. You can interrogate and/or terminate any communication over any path. The IUCV functions are:

ACCEPT	QUIESCE	SEND
CONNECT	RECEIVE	SET CONTROL MASK
DECLARE BUFFER	REJECT	SET MASK
DESCRIBE	REPLY	SEVER
PURGE	RESUME	TEST COMPLETION
QUERY	RETRIEVE BUFFER	TEST MESSAGE

ACCEPT Function

The target communicator uses the ACCEPT function to respond to a connect request from a virtual machine or from a CP system service. If the target communicator chooses not to complete the connection, it invokes the SEVER function. If the ACCEPT function initiator does not declare a buffer, the system generates a specification exception.

The ACCEPT function initiator receives a return code if one of the following occurs:

- The path specified is not a pending connection.
- The path has been severed by the originator of the CONNECT function.
- An IUCV ACCEPT was issued in response to an APPC/VM CONNECT, or an APPC/VM ACCEPT was issued in response to an IUCV CONNECT.

Both the initiator and the originator have their path descriptor entries set to valid, and an external interrupt is built. A request for QUIESCE mode when the originator invokes CONNECT results in an indicator's being set in both the path descriptor and the external interrupt. The QUIESCE option is ignored for APPC/VM CONNECTs and ACCEPTs.

When the ACCEPT function is invoked from CP system code, the message limit and priority setup is based on the parameter list values. If the message limit is not specified, a default value is used. When an APPC/VM ACCEPT is invoked, the message limit for the path is one.

When the ACCEPT function is not invoked from CP system code, a directory check is made to determine whether or not the target communicator entry exists. The directory check search sequence is:

1. The initiator's IUCV entries to find an entry for the source communicator.
2. The initiator's IUCV entries to find an entry for ALL.

If neither of these authorizations exists, then communication is not allowed.

For the message limit, the lower value of the parameter list limit or the directory limit is used. If you specify neither, the default value prevails. The established message limit returns to the initiator in the parameter list and then passes to the source communicator in the connection complete external interrupt.

To receive data in the parameter list as well as in a buffer, the communicator must specify the PARMDATA=YES option. Parameter list data is limited to two fullwords, or eight bytes, of data. On APPC/VM paths, however, parameter list data is *not* permitted. The PARMDATA=YES option is ignored for APPC/VM ACCEPTs.

If the originator of the IUCV CONNECT or APPC/VM CONNECT is a virtual machine, and WAIT=NO was specified, then an external interrupt is stacked and the initiator receives a normal return code. If the originator of the APPC/VM CONNECT specified WAIT=YES, then the CPEXBLOK created during CONNECT processing is stacked to complete the originator's CONNECT and the initiator receives a normal return code.

If the originator is CP system code, the pending connection is located and dequeued from the CP pending connection chain and a CPEXBLOK is stacked to indicate to the originator that the connection is complete. Finally, the initiator receives a normal return code.

CONNECT Function

To start communication between virtual machines or between a virtual machine and a CP system, use the CONNECT function. During the CONNECT process:

- If a buffer is not declared, a specification exception is generated.
- If the source communicator is a CP system service, directory checking is skipped.
- If the target communicator is not logged on or has not declared a buffer, a return code is sent to the initiator.
- If the CONNECT function is invoked from CP system code, the message limit and priority setup is based on the parameter list values. If the message limit is not specified, a default value is used.
- If the target communicator is a CP system service (where the ID starts with an asterisk), the name is verified by looking it up in a table. If not found, a return code is sent to the initiator.
- If directory checking is needed, the directory is checked to verify that this connection is authorized. The directory check search sequence is:
  - The initiator's IUCV entries to find an entry for the target communicator.
  - The initiator's IUCV entries to find an entry for ALL.
- When directory authorization is not found, a return code is sent to the initiator.
- To receive data in the parameter list as well as in a buffer, the communicator must specify the PARMDATA=YES option. Parameter list data is limited to two fullwords, or eight bytes, of data. On APPC/VM paths, however, parameter list data is *not* permitted. The PARMDATA=YES option is ignored for APPC/VM CONNECTs.
- When an APPC/VM CONNECT is invoked, the message limit for the path is one.

The message limit is specified either in the IUCV directory control statement, which provides connection authorization, or in the parameter list, or in both. When a message limit is specified in both, use the lower one of the two. The established message limit returns to the initiator in the parameter list and then passes to the target communicator in the pending connection external interrupt.

In the directory entry, priority authorization for IUCV CONNECTs is allowed only if specified. For APPC/VM CONNECTs, priority is ignored.

If either the initiator or the target communicator exceeds its maximum connection limit, the initiator receives a return code.

When the target communicator is a virtual machine, a pending connection external interrupt is stacked for it. When the target communicator is a CP system service, the connect entry point for that service is invoked. When the originator is CP system code, a pending connection is added to the CP pending connection chain. When an APPC/VM CONNECT with WAIT=YES is specified, a CPEXBLOK is obtained and chained off the initiator's IUCVBLOK. IUCV then exits to the dispatcher to wait for the CPEXBLOK to be stacked. After the CPEXBLOK is stacked, the initiator receives a message that the CONNECT is complete.

DECLARE BUFFER Function

To specify the buffer address where the virtual machine external interrupt should be stored, use the DECLARE BUFFER function.

It is unnecessary for CP system code to issue the DECLARE BUFFER function, because CP does not receive external interrupts.

If the initiator already has a buffer, the initiator receives a return code.

Based on the maximum number of connections, the system builds an IUCVBLOK and a communication control table (CCT) for the initiator. The directory contains the maximum number of connections permitted for each user; if none appears, the default is four. When errors are encountered reading the directory, the initiator receives a return code.

Notes:

1. *You can reduce the overhead involved in reflecting IUCV external interrupts to the virtual machine by declaring the buffer (with the DECLARE BUFFER function) so that it fits entirely within one page. You can reduce the overhead even further by declaring the buffer so that it fits entirely within page zero of the virtual machine.*

2. *The IUCV external interrupt submask bits are set on when executing the DECLARE BUFFER function. You must invoke the SET MASK function to change any of these bit settings.*

 LY20-0897-7

DESCRIBE Function

The target communicator uses the DESCRIBE function to determine the presence of any messages not previously described or reflected in a message pending IUCV external interrupt. The parameter list will receive the pertinent information about the message.

The search routine for the SEND queue uses a first-in first-out (FIFO) approach. The first undescribed and unreflected MSGBLOK found is selected. APPC/VM MSGBLOKs are described only if the corresponding path is in RECEIVE state. Because of their position in the SEND queue, priority MSGBLOKs are scanned first. The DESCRIBE function causes the following MSGBLOK information to be stored:

- Path ID
- Target message class
- Message ID
- Message flags
- Length of message
- Length of answer area.

Note: The target message class, message ID, and length of answer area are not *stored for APPC/VM messages.*

This information eventually allows the target communicator to accept the MSGBLOK data through use of the RECEIVE function.

The DESCRIBE function sets the condition code to indicate:

- No undescribed MSGBLOK found
- MSGBLOK found; description stored.

If the message being described contains the message data in the parameter list, or is an APPC/VM SENDREQ MSGBLOK, the message is considered to be received and the MSGBLOK is removed from the SEND queue.

Note that IUCV describes each MSGBLOK once and the target communicator is responsible for removing those MSGBLOK(s) from the SEND queue. Use of either the RECEIVE function or REJECT function removes the MSGBLOK(s).

The DESCRIBE function clears the pending message external interrupt for each MSGBLOK.

Note: CP system code (aside from IUCV support) cannot use the DESCRIBE function.

PURGE Function

The source communicator uses the PURGE function to terminate a single message. The message is destroyed immediately under the following circumstances:

- If it has not been described to the target communicator, which means the target communicator is completely unaware that the message was ever sent.
- If it is on the source communicator's REPLY queue.

However, the message, when previously described to the target communicator, is marked as purged and the target communicator receives a return code in the parameter list when next invoking either the RECEIVE function or the REPLY function, whereupon the message is destroyed. Note that the PURGE function moves no data.

The PURGE function parameter list describes the to-be-purged message; this list includes:

- A path ID. (The value of the path ID determines which target communicator queues are to be searched.)
- A message ID.

When both of these are specified, then the source message class must also be specified. However, when the message ID is not specified, then the message class is optional.

Note: The parameter list flags indicate which fields are to be used in locating the message.

The search for the message starts with the source communicator's REPLY queue, then the target communicator's RECEIVE queue, and then the target communicator's SEND queue. The first MSGBLOK that matches the message specification terminates the search. When a message is purged, the message ID, the path ID, the source message class, the message tag, the message flags, and the audit trail are all stored in the parameter list.

A condition code reflects the completion status of the PURGE function. Detection of certain error conditions stores a return code, which indicates the type of error found.

QUERY Function

Use the QUERY function to extract IUCV information about the virtual machine. The IUCV external interrupt buffer size is returned in general register 0. General register 1 holds the data for the maximum number of connections outstanding for the particular virtual machine.

Notes:

1. *The QUERY function does not take a parameter list.*
2. *If errors are encountered, a condition code is set.*
3. *CP system code cannot invoke the QUERY function.*

QUIESCE Function

Use the QUIESCE function when you wish to suspend temporarily the ability of the communicating partner to send messages.

When the request is to quiesce all paths, each path quiesces individually and the initiator receives a return code. For each path on a QUIESCE ALL or for the single specified path (if the path is marked severed or if the path is already quiesced), the initiator receives a normal return code. Otherwise, the system builds an external interrupt based on the path being quiesced and marks the path as quiesced.

When the communicating partner is a CP system service, the quiesce entry point for the service is located and external interrupt data is passed to it via a CALL linkage. The initiator receives a return code upon return of control.

If a virtual machine is the communicating partner, the external interrupt for the communicating partner is stacked and the initiator receives a return code.

RECEIVE Function

The target communicator uses the RECEIVE function to accept messages.

When the complete message moves from the send area to the specified receive area, the MSGBLOK for the designated message moves from the SEND queue to the RECEIVE queue. If the receive area cannot contain the message, the MSGBLOK stays in the SEND queue and the length of the remaining data is stored in the parameter list. When the RECEIVE function is again activated, the remainder of the message becomes available.

RECEIVE function input specifies which message is to be processed, identified by:

- Message ID
- Path ID
- Target message class.

Notes:

1. *The message ID and target message class are not valid for APPC/VM RECEIVEs. If the message ID is not specified, then any combination of path ID and target message class can be specified. The flag fields of the RECEIVE parameter list indicate the type of message description to be used. If no parameter list search fields are specified (for IUCV RECEIVEs), then the first message that has not been partially received is presented. If message data is contained in the parameter list and a REPLY is expected, the MSGBLOK is moved from the SEND queue to the RECEIVE queue. On a one-way (no REPLY) message, the MSGBLOK is destroyed.*

2. *After a partially received message, the message must be completely specified in order for the remainder of the message to be received.*

The target communicator obtains the required message description in one of the following ways:

- By use of the DESCRIBE function.
- Via presentation by an external interrupt.

Note: A different linkage is used for a CP system service because neither the DESCRIBE function nor an external interrupt is available.

The RECEIVE function input identifies a receive area to which the message goes if BUFLIST=YES is not specified. A beginning address and a length describes the receive area. The address must be real to the virtual machine, and there is no alignment requirement for the beginning address. Data movement terminates when the message length exceeds the receive area length or the message ends.

Note: If length specification is 0 for either data area, there is no data movement.

If BUFLIST=YES is specified, the BUFFER= parameter of the IUCV macro instruction provides the address of a list of addresses and lengths of discontiguous buffers to contain the message text. Also, the value specified with the BUFLEN= parameter is the total of the individual buffer lengths in the list pointed to be BUFFER=.

Completion of the RECEIVE function results in update of the parameter list receive area. If BUFLIST=NO, the address is set to the originally designated length plus the number of bytes moved. The updated length is the residual count when the return code indicates that the buffer was too short, or the return code indicates that it is the remaining length of the buffer.

If BUFLIST=YES is specified, the address in the parameter list points to the entry in the buffer list to continue processing. The buffer list is updated throughout IUCV processing. As data is moved, the address in the list entry is incremented by the length moved, and the length in the list entry is decremented by that length. When the length is zero, the list pointer is incremented to indicate the next entry to process. The total length specified in the parameter list is also decremented with each move. It reflects the amount of data to be received.

The MSGBLOK moves directly to the REPLY queue when either the send area or the receive area has addressing exceptions and/or protection exceptions. When this happens, the target communicator receives a return code to that effect while the audit trail notifies the source communicator.

A condition code is set to report RECEIVE function completion status. If an error condition occurs, its detection activates the setting of a return code to indicate the type of error.

When a priority MSGBLOK moves to the RECEIVE queue, the concept of priority disappears. The REPLY function can reintroduce priority.

REJECT Function

To reject a single message from the source communicator, the target communicator executes the REJECT function. The MSGBLOK for the designated message moves from the target communicator's SEND queue or RECEIVE queue to the source communicator's REPLY queue. Use of the REJECT function moves no data.

REJECT function input specifies which message is to be rejected. The information is identified by:

- Message ID
- Path ID
- Target message class.

If the message ID is not specified, then any path ID and target message class combination are valid. The flag field of the REJECT parameter list indicates the type of message description to be used. If flags are not specified, the REJECT function is terminated with a specification exception.

The program searches the target communicator's queues for the designated message: first, the RECEIVE queue; then, the SEND queue. The first MSGBLOK that matches the designated message moves to the source communicator's REPLY queue. Rejection of the designated message is via setting a condition code.

The REJECT function parameter list stores the message ID, the path ID, and the target message class on completion.

To indicate message rejection, the audit trail is updated.

A condition code is set to report REJECT function completion status. If certain types of errors occur, storing of a return code indicates it.

REPLY Function

The target communicator uses the REPLY function to respond to a message. The MSGBLOK for the designated message moves from the target communicator's RECEIVE queue to the source communicator's REPLY queue. Data moves from the specified reply area to the source communicator's answer area.

It is assumed that REPLY function input has a complete description of the designated message requiring a reply. Partial descriptions of a message are not supported. If insufficient information is supplied the parameter list, locating the designated message will be impossible. The message description comprises:

- Message ID
- Path ID
- Target message class.

If the designated message is not found, a parameter list return code is set.

The target communicator turns on the parameter list flag to specify that the reply message has priority. The reply MSGBLOK thus precedes any nonpriority MSGBLOK(s) in the queue, immediately following any earlier-designated priority MSGBLOK(s).

Input to the REPLY function identifies a reply area if ANLIST is not specified. The description includes a beginning address and a length. The address must be real to the virtual machine although no alignment requirement is made for the beginning address. Data moves between the target communicator's reply area and the source communicator's answer area terminate when either area length is exhausted. A length of 0 for either area prevents data transfer. On completion of the REPLY function, any length mismatch results in an error condition being posted.

If ANSLIST=YES is specified, the ANSBUF= parameter of the IUCV macro instruction provides the address of a list of addresses and lengths of discontiguous buffers to contain the message reply text. Also, the value specified with the BUFLEN= parameter is the total of the individual buffer lengths in the list pointed to be ANSBUF=.

When addressing exceptions and protection exceptions occur while accessing either the reply area or the answer area, the MSGBLOK moves to the REPLY queue, a return code is sent to the target communicator, and the audit trail notifies the source communicator.

If ANSLIST=NO, completion of the REPLY function updates the reply area description in the parameter list and sets the address at the original length *plus* the number of bytes moved. Note that the updated length is the residual count if the return code indicates the buffer was too short or is the remaining length of the buffer on a normal return code.

If ANSLIST=YES is specified, the address in the parameter list points to the entry in the answer list to continue processing. As data is moved, the address in the list entry is incremented by the length moved, and the length in the list entry is decremented by that length. When the length is zero, the list pointer is incremented to indicate the next entry to process. The total length specified in the parameter list is also decremented with each move. It reflects the amount of data in the reply.

The REPLY function cannot execute if a message is sent by a one-way SEND function. Such a message never resides on the target communicator's RECEIVE queue. Thus, a "no message found" condition results.

A condition code reflects the REPLY function completion status. If an error condition occurs, its detection activates storing of a return code.

RESUME Function

Use the RESUME function to restore IUCV communications after you invoke the QUIESCE function.

If your request is to resume all paths, each path resumes individually and the initiator receives a return code.

After a RESUME function is issued for a single specified path or for ALL paths (if the path is marked severed or is not quiesced), a normal return code is returned to the initiator.

If the communicating partner is not CP, the external interrupt is stacked for the virtual machine and the initiator receives a return code.

If the communicating partner is a CP system service, the program locates the RESUME entry point for the service and the external interrupt data passes to it via a CALL linkage.

When the CALL linkage processing is completed, the initiator receives a return code.

RETRIEVE BUFFER Function

Use of the RETRIEVE BUFFER function notifies IUCV that the virtual machine no longer needs IUCV.

The program generates a SEVER ALL to terminate all messages on all paths for the designated communicator. Any control blocks built for this communicator at DECLARE BUFFER time are dismantled and released.

Note: CP system code, outside of IUCV support, cannot use the RETRIEVE BUFFER function.

SEND Function

The SEND function initiates communication by creating a MSGBLOK and enqueuing it on the target communicator's SENDQ.

The input to the SEND function must completely describe the message being sent. It must specify the source communicator's path ID and the source communicator's and target communicator's message classes. Also required when invoked from a virtual machine is the message tag that is presented to the source communicator upon completion of the message. The MSGTAG field is used by IUCV for CP-initiated messages and is not available to a CP service.

Note: The message class and message tag are not valid on APPC/VM SENDs.

If BUFLIST=YES and ANSLIST=YES are not specified, the user can specify two data areas that are used to move data between the source communicator and the target communicator. The send area contains the data to be moved from the source communicator to the target communicator. The answer area, for IUCV SENDs, is the area into which the target communicator's REPLY data is moved. For APPC/VM SENDs, the target communicator's SENDDATA data is moved into the answer area. Each area is defined by a beginning address and a length. Each address must be real to the virtual machine. Either data area can be anywhere within the source address space. There is no boundary alignment requirement on the beginning addresses.

The user may choose to send the data in the parameter list and not specify a send area, or he may choose to have the REPLY data returned in the parameter list and not specify an answer area. When using the parameter list data option, the user is limited to two fullwords, or eight bytes. Parameter list data is not valid on APPC/VM SENDs.

If BUFLIST=YES is specified, the BUFFER= parameter of the IUCV macro instruction provides the address of a list of addresses and lengths of discontiguous buffers to contain the message text. Also, the value specified with the BUFLEN= parameter is the total of the individual buffer lengths in the list pointed to be BUFFER=.

If ANSLIST=YES is specified, the ANSBUF= parameter of the IUCV macro instruction provides the address of a list of addresses and lengths of discontiguous buffers to contain the message text. Also, the value specified with the BUFLEN= parameter is the total of the individual buffer lengths in the list pointed to be ANSBUF=.

The IUCV SEND function does not move any data. The target communicator invokes the RECEIVE and/or REPLY functions to move data. Because of this, a description of the send area and the answer area is stored in the MSGBLOK for use during either function. The description consists of the beginning buffer or list addresses, the total length of each area, and the PSW key to be used for protection checking during access to each area. The APPC/VM SEND function moves the data if a receive or answer area has been previously defined by the target for the path.

Data areas are not validity-checked during the SEND operation, unless the data is actually moved. The check occurs when the areas are used. Access exceptions in the source address space are recognized and reported during processing of the RECEIVE function and/or REPLY function.

Using a parameter list flag field, you can optionally alter the SEND function to either a priority message or a one-way message. The SEND function with the priority flag set enqueues the MSGBLOK on the target communicator's SEND queue preceding all nonpriority MSGBLOK(s) and following all earlier priority MSGBLOK(S). The SEND function with the one-way flag set designates the MSGBLOK as one that does not allow a reply. When the target communicator receives a one-way message, the MSGBLOK skips the target communicator's RECEIVE queue, and is immediately placed on the source communicator's REPLY queue.

A condition code is set to report SEND function completion status. If an error condition occurs, its detection activates the setting of a condition code as well as the storing of a return code to indicate which error was detected.

SET CONTROL MASK Function

The SETCMASK function enables or disables external interrupts for the five types of IUCV control interrupts:

- Connection Pending
- Connection Complete
- Path Severed
- Path Quiesced
- Path Resumed.

Specify all mask bits in the parameter list. All mask bits are used and override all previous mask specifications.

Before the Control Mask bits are interrogated, a virtual machine must first be enabled for IUCV control type external interrupts by using the SETMASK function.

Note: The SETCMASK function cannot be used from CP system code.

SET MASK Function

The SET MASK function enables or disables external interrupts for priority messages and nonpriority messages, priority replies and nonpriority replies, and IUCV controls. Specify all mask bits in the parameter list. All mask bits are used and override any and all previous mask specifications. Use of the mask is in addition to the global external interrupt mask in the PSW.

Note: The SET MASK function cannot be used from CP system code.

SEVER Function

Use the SEVER function to terminate IUCV communications capabilities.

If the path is complete, both communicators must invoke a SEVER function to the path. After one communicator's invocation, all messages still on the path terminate and the communicating partner receives a sever external interrupt. The communicating partner then, if desired, can dequeue the terminated message(s). When finished, the communicating partner invokes the SEVER function.

For the SEVER ALL, each message on either the SEND queue or the RECEIVE queue is designated as severed in the audit trail. Then the REJECT function is invoked to terminate each message. The communication control table (CCT), part of the IUCVBLOK, contains a list of the valid path ID(s). Once message termination is complete, each valid path from 0 through the highest valid path in the CCT also terminates. The process completes with the release of all the space used by the identification control blocks. All designated paths are masked as invalid.

Path termination for a valid path proceeds as follows:

- If the path is invalid or out of range, the initiator receives a return code.

- If the path is valid, the QUIESCE function is invoked to prevent further communication.

- If the path is marked as severed, the entry is set to available and the initiator receives a return code. Any messages in the REPLY queue for the path are dequeued and the space returned to storage.

- For each message on either the SEND queue and/or the RECEIVE queue for this path, a REJECT is issued.

- Each message generated by the initiator for this path terminates with a PURGE operation.

If the communicating partner is a CP system service, the program locates the sever entry point for the service and the external interrupt data passes to it via a CALL linkage.

When the CALL linkage processing completes, the initiator receives a return code.

If the communicating partner is not CP, the external interrupt is stacked for the virtual machine and the initiator receives a return code.

Path termination for an invalid path proceeds as follows:

- If the path is severed because the SEVER function has been invoked from the target communicator, the path is set to available and the initiator receives a normal return code.

- If the communicating partner has received the pending connection interrupt, the blocks are marked as severed and the communicating partner must also invoke the SEVER function to fully dismantle the control blocks.

- If the communicating partner has not received the pending connection interrupt, the path terminates without intervention from the communicating partner.

Note: *CP system code (aside from IUCV support) cannot use the SEVER ALL function.*

TEST COMPLETION Function

The source communicator executes the TEST COMPLETION function to complete a communication. The procedure includes:

- Dequeuing the MSGBLOK of the completed message from the source communicator's REPLY queue.

- Destroying the dequeued MSGBLOK.

Data is not moved into the answer buffer by the TEST COMPLETION function. However, if the REPLY function is used with the DATA=PRMMSG option, the eight bytes of data appear in the TEST COMPLETION parameter list.

TEST COMPLETION function specifies which message is to be processed; the message is identified by:

- Message ID
- Path ID
- Source message class.

Note: *The message ID and message class are not recognized for APPC/VM messages.*

If the message ID is not specified, then any path ID and source message class combination is valid for the TEST COMPLETION function. The flag field of the TEST COMPLETION parameter list indicates the type of message description to be used. If no parameter list search fields are specified, the first REPLY queue message is presented. If the specified message is found, then along with the setting of a normal condition code, the following parameter list fields are stored:

- Message ID
- Path ID
- Flags
- Audit trail
- Message tag
- Source message class.

Notes:

1. *The message ID, message tag, and source message class are not stored for APPC/VM messages. If the specified message is not found, just a condition code is set. Note that the TEST COMPLETION function clears the pending message complete external interrupt for the REPLY queue message.*

2. *CP system code (aside from IUCV support) cannot use the TEST COMPLETION function.*

TEST MESSAGE Function

The virtual machine communicator invokes the TEST MESSAGE function to determine whether or not messages or replies are pending. The virtual machine enters a wait state when neither message or replies are pending.

The virtual machine communicator uses the TEST MESSAGE function to poll for IUCV current messages and/or replies as well as to wait for future messages and/or replies. An APPC/VM message pending on a path which is not in RECEIVE state is ignored by the TEST MESSAGE function.

Use of the TEST MESSAGE function allows an instruction stream to poll for a message and, simultaneously, to enter a wait state. If, during a wait state, an IUCV message and/or reply pending occurs, the virtual machine communicator reinvokes the TEST MESSAGE function to continue processing and a proper condition code is sent to the initiator.

Because IUCV messages are also presented as external interrupts, the TEST MESSAGE function introduces the anomaly of identifying IUCV messages simultaneously through the external interrupts and through the TEST MESSAGE function's polling capability. If the PSW external interrupt mask bit or the SET MASK function are disabled for IUCV messages and replies, the TEST MESSAGE function may be used to poll, and the DESCRIBE, RECEIVE, and TEST COMPLETION functions used to receive all information about IUCV messages and replies. This polling reduces external interrupt handling overhead.

The condition code setting indicates that there is at least one message and/or reply in the SEND queue or the REPLY queue. Status information is forthcoming when you execute the DESCRIBE function and the TEST COMPLETION function. Note that the condition code setting indicates the TEST MESSAGE function completion status.

Note: *CP system code (aside from IUCV support) cannot use the TEST MESSAGE function.*

Return codes applicable to IUCV are described in *VM/SP HPO System Facilities for Programming*.

IUCV Restrictions

The following areas of IUCV are limited:

- The use of IUCV is supported for a second-level CP system. The IUCV functions are not simulated, but are reflected to the second-level system.
- Each virtual machine is limited to less than 65,536 outstanding connections at one time.
- IUCV does not recognize anything smaller than a virtual machine. If two communicators choose to establish multiple communication paths, it is the responsibility of these communicators to manage these paths.
- A CP system service cannot establish communication with itself.
- The sum total of all CP system service connections cannot be greater than 4096.

IUCV Trace Table Entries

IUCV support generates a trace table entry for each IUCV function. There is one trace table entry type for IUCV entries (X′15′) with a subtype field to indicate exactly which IUCV function was invoked. All uses of IUCV, except for the three functions listed above, whether invoked from a virtual machine or from CP system code, are recorded in the CP trace table. The address portion of the old PSW is recorded as part of the entry. A bit in the flag byte indicates whether this address is to be interpreted as a real address (when invoked from CP) or a virtual machine address (when invoked from a virtual machine). For virtual machine addresses, the address of the associated VMBLOK can be obtained from preceding trace table entries.

The IUCV trace facilities can be suppressed at assembly time by setting &TRACE (9) to 0 or at execution time by setting the X′80′ bit to 0 in the TRACFLG3 field of the PSA.

The trace table entries for IUCV are built in two sections. The first part, consisting of IUCV subtype, the address of the USER who invoked the IUCV function, and the bit indicating whether the USER's address is real or virtual (guest real), is built only after preliminary tests are successful. These tests check to see if:

- The parameter list is valid.
- The parameter list is on a doubleword boundary.

When the first part of the trace table entry is built, a bit is set in the flags byte indicating that it is a partial trace table entry. If the function should terminate with a return code, this partial entry is not updated in the trace table. If the function completes without error, the rest of the trace table entry is filled in and the 'partial entry' bit is reset.

For IUCV functions that invoke other functions, the secondary functions are also recorded as having been invoked from CP. Examples of these secondary functions are:

- Retrieve Buffer generates a Sever ALL.
- Sever generates a Reject for each incoming outstanding message and a purge for each outgoing outstanding message.
- A Connect to a CP system service causes control to go to that service and will usually invoke the Accept function.
- The IUCV support invokes the Test Completion function to dequeue messages intended for the CP system.

IUCV External Interrupts

Prior to establishing any connections, the virtual machine must invoke the Declare Buffer function to indicate to IUCV where data associated with an external interrupt is to be stored.

There is one external interrupt type for external interrupts generated by IUCV. This external interrupt type is X'4000'. When an IUCV external interrupt is reflected to the virtual machine, the interrupt code is stored for the virtual machine and an 'External Interrupt Buffer' is stored at the address specified in the DECLARE BUFFER function. One field of this buffer, IPTYPE, is an external interrupts subtype to indicate exactly why the external interrupt occurred. The possible codes and their meanings are as follows:

- 01 - IUCV Connection Pending
- 81 - APPC/VM Connection Pending
- 02 - IUCV Connection Complete
- 82 - APCC/VM Connection Complete
- 03 - IUCV Path Severed
- 83 - APCC/VM Path Severed
- 04 - Path Quiesced
- 05 - Path Resumed
- 06 - Incoming Priority Reply
- 07 - Incoming Nonpriority Reply
- 87 - Function Complete
- 08 - Incoming Priority Message
- 88 - SENDREQ Interrupt
- 09 - IUCV Incoming Nonpriority Message
- 89 - APCC/VM Incoming Message.

While the Connect, Accept, Sever, Quiesce, and Resume functions always cause a pending external interrupt to be queued for the target virtual machine or passed to a CP service, incoming messages and incoming replies can be fielded by the target virtual machine as either external interrupts or by the satisfaction of the Describe or Test Completion functions.

When a virtual machine executes a Send, a pending external interrupt of subtype 08, 09, 88, or 89 is queued for the target virtual machine. If the target virtual machine is both enabled for external interrupts (bit 7 in the virtual PSW is set to 1) and enabled for messages (via the Set Mask function), then the external interrupt will be reflected. (In the case of APPC/VM incoming message interrupts, the corresponding path must be in RECEIVE state before the interrupt is presented.) If either condition is not met, the external interrupt will remain queued. If the target virtual machine is not enabled but instead executes the Describe function, the information about the pending message will be returned in the parameter list and the pending external interrupt for that particular message will be cleared.

The condition of being enabled for IUCV messages and issuing a Describe will cause unpredictable results. In a similar manner, the condition of being enabled for IUCV replies and issuing a Test Completion will cause unpredictable results. Although it is unpredictable as to whether the external interrupt is presented or the IUCV function (Describe or Test Completion) is satisfied, it is never the case that both the external interrupt and the IUCV function completion will occur for the same message/reply.

All IUCV external interrupts are controlled by the external mask bit in the virtual PSW (bit 7) and the submask bit in control register zero (bit 30).

There are separate additional mask bits for IUCV external interrupts that can be enabled and disabled by the Set Mask function. Five mask bits are defined for use by the Set Mask function. These mask bits are used to separately mask incoming priority messages, incoming nonpriority messages, incoming priority replies, incoming nonpriority replies, incoming APPC/VM messages, incoming APPC/VM SENDREQ interrupts, APPC/VM function complete interrupts, and IUCV control interrupts of subtypes 01, 02, 03, 04, and 05. When both the external interrupt mask and the appropriate Set Mask bits are enabled, the external interrupt can occur.

The SETCMASK function lets you set masks for the individual IUCV control interrupts. The IPCMASK field specifies which of the five types of IUCV control interrupts for the virtual machine are enabled. These interrupts are:

- Connection Pending
- Connection Complete
- Path Severed
- Path Quiesced
- Path Resumed.

The SETMASK function is interrogated before the SETCMASK function mask. If you use the SETMASK function to specify that all control interrupts are disabled, then the SETCMASK settings are not interrogated. If you specify that all control interrupts are enabled by using the SETMASK function, then the SETCMASK settings are interrogated to determine how to handle the individual types of control interrupts.

After IUCV initialization and until you issue the SETMASK and SETCMASK functions, all IUCV submask bits are on, enabling all IUCV external interrupts.

When the virtual machine has completed all communications, the virtual machine may invoke the Retrieve Buffer function to cause IUCV to stop using the external interrupt buffer and prevent further IUCV communication.

External Interrupts are not reflected to CP system code. For communications to CP services, external interrupts are replaced with one of two possible linkages depending on whether the function was initiated outside CP or from within CP.

The order of reflection for IUCV external interrupts is as follows:

1. Control interrupts (Subtype X'01', X'81', X'02', X'82', X'03', X'83', X'04', X'05') in FIFO (First In, First Out) order
2. Priority Replies (Subtype X'06')
3. Nonpriority Replies (Subtype X'07')
4. Priority Messages (Subtype X'08') and SENDREQs (Subtype X'88')
5. Nonpriority Messages (Subtype X'09')
6. APPC/VM Messages (Subtype X'89').

IUCV Control Blocks and Data Areas

Figure 24 shows the relationships between the various IUCV control blocks and data areas. IUCV identifies and describes a communicator with an IUCVBLOK. The Communication Control Table (CCT), part of the IUCVBLOK, contains a Path Description Entry (PDENT) for each path defined for the communicator. There is a PDENT in the source CCT and a different PDENT in the target CCT for each path defined. Each of these PDENTs is identified, or named, by a Path Description Identifier (PDID or Path ID). There is PDID for the source communicator's view of a path (its PDENT for the path) and another PDID for the target communicator's view of the same path (its PDENT). At the interface to IUCV, there is no relationship assumed between the two PDID values. A particular communicator can address a path only by that communicator's PDID.

Messages are represented by Message Blocks (MSGBLOKs). A MSGBLOK is created when a communication is initiated and is destroyed when a communication is terminated. A message, and its representation as a MSGBLOK, is fully identified by three values. These values are the PDID, the Message Class, and the Message ID. The source and target communicators each have their own description of a particular message.

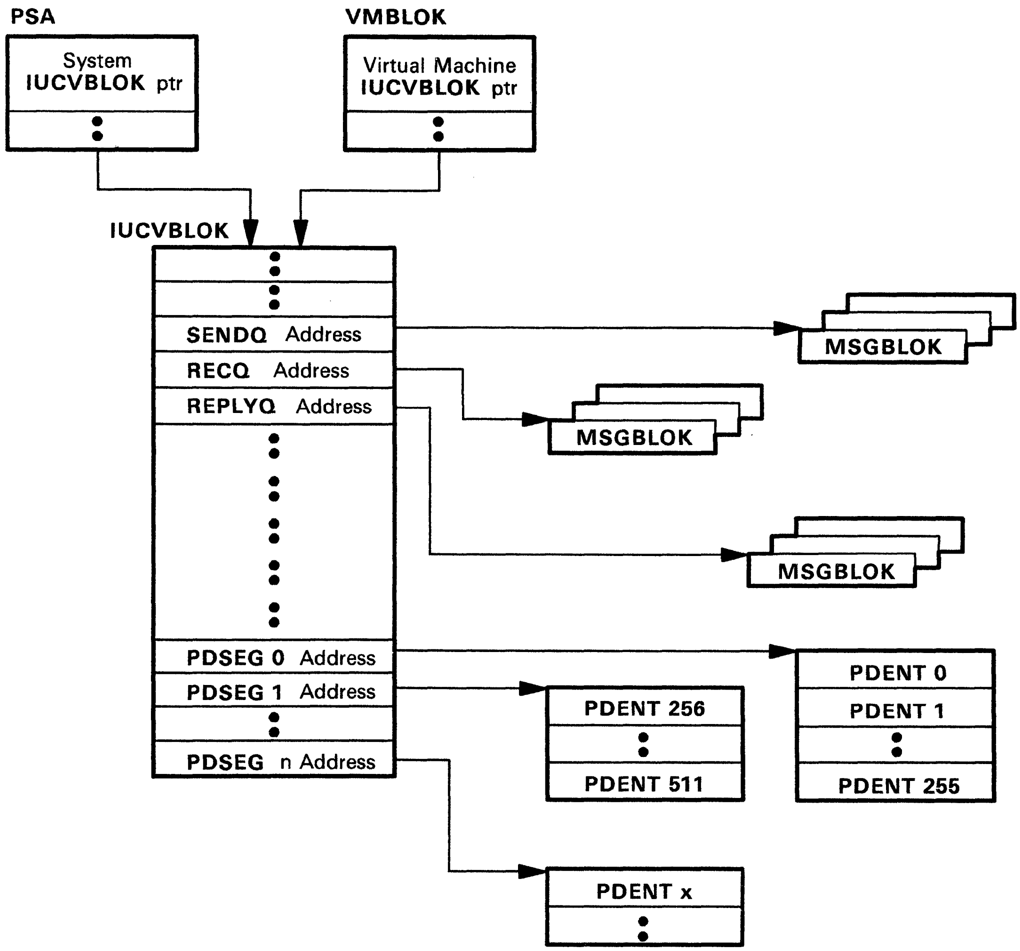


Figure 24. IUCV Control Block Relationships

A MSGBLOK represents an active communication and is chained onto one of the queues anchored in a CCT. A CCT contains 3 queues. These queues keep track of communication status. Queued MSGBLOKs can be handled FIFO, enqueued by priority, and dequeued by value.

The Send Queue (SENDQ) is defined for target communicators. It is the queue of MSGBLOKs that have been created by the source communicator but not yet accepted by the target communicator. The Receive Queue (RECQ) is also defined for target communicators. It is the queue of MSGBLOKs that have been accepted by the target communicator, but to which the target communicator has not yet replied. The Reply Queue (REPLYQ) is defined for source communicators. It contains those MSGBLOKs that have been replied to by a target communicator, but which have not yet been terminated.

A normal communication uses a MSGBLOK in a predefined manner. The MSGBLOK moves in sequence from a SENDQ, to an RECQ, and finally to a REPLYQ.

VM/VS Handshaking

The VM/VS Handshaking feature provides a communication path between CP and virtual machine operating systems that makes each system control program aware of certain capabilities or requirements of the other.

The following is a discussion of VM/VS Handshaking as it relates to OS/VS1. Functions of VM/VS Handshaking incorporated in the control program are available and applicable to any operating system that can be system generated to use this function.

VM/VS Handshaking for OS/VS1 performs the following functions:

- Closes CP spool files when the VS1 job output from its DSO, terminator, and output writer is complete
- Processes VS1 pseudo page faults
- Provides an optional nonpaging mode for VS1 when it is run in the VM/SP HPO environment.

When a VS1 virtual machine with the handshaking feature is loaded (via IPL), its initialization routines determine whether the handshaking feature should be enabled. First, VS1 determines if it is running under the control of VM by issuing a STIDP (Store Processor ID) instruction. STIDP returns a version code; a version code of X'FF' indicates VS1 is running with VM/SP HPO. If VS1 finds a version code of X'FF', it then issues a DIAGNOSE (X'00') instruction to store the VM extended-identification code. If an extended-identification code is returned to VS1, VS1 knows handshaking is supported. At this time or any time after IPL, the operator of the VS1 virtual machine can issue the CP SET PAGEX ON command to enable the pseudo page fault handling portion of handshaking. If the VS1 virtual machine is in the nonpaging mode and, if the pseudo page fault handling is active, full handshaking support is available.

Because the VS1 system does no paging, any ISAM programs run under VS1 are treated as though they are running in an ADDRSPC=REAL partition. Therefore, the ISAM option is required for the VS1 machine to successfully execute the ISAM program.

Closing CP Spool Files

If the handshaking feature is active, VS1 closes the CP spool files when its job output from the DSO, terminator, and output writer is complete. Once the spool files are closed, CP processes them and they are sent to the real printer or punch. During its job termination processing, VS1 issues a DIAGNOSE (X'08') instruction to pass the CP CLOSE command to CP for each spool file.

Pseudo Page Faults

A page fault is a program interruption that occurs when a page marked "not in storage" is referred to by an instruction with an active page. The virtual machine referring to the page is placed in a wait state while the page is brought into real storage. Without the handshaking feature, the entire VS1 virtual machine is placed in page wait until the needed page is available.

However, with the handshaking feature, a multiprogramming (or multitasking) VS1 virtual machine can dispatch one task while waiting for a page request to be answered for another task. CP passes a pseudo page fault (program interrupt X'14') to VS1. When VS1 recognizes the pseudo page fault, it places only the task waiting for the page in page wait and can dispatch another tasks.

When a page fault occurs for a VS1 virtual machine, CP checks that the pseudo page fault portion of handshaking is active and that the VS1 virtual machine is in EC mode and enabled for I/O interruptions. Then, CP reflects the page fault to VS1 by:

- Storing the virtual machine address that caused the page fault at location X'90' (the translation exception address)
- Indicating a program interruption (interrupt code X'14') to VS1
- Removing the VS1 virtual machine from page wait and execution wait.

When VS1 recognizes program interruption code X'14', it places the associated task in wait state. VS1 can then dispatch other tasks.

When the requested page becomes available in real storage, CP indicates the same program interruption to VS1, except that the leftmost bit in the translation exception address field is set on to indicate completion. VS1 removes the task from page wait; the task is then eligible to be dispatched.

VS1 Nonpaging Mode

When VS1 runs under VM, it executes in nonpaging mode if:

- Its virtual storage size is equal to the size of the virtual machine.
- Its virtual machine size is at least 1024K bytes and no more than 4096K bytes. For VS1 Release 6, the maximum size is 16,384K bytes.
- The VM/VS Handshaking feature is available.

When VS1 executes in nonpaging mode, it uses fewer privileged instructions and avoids duplicate paging. The VS1 Nucleus Initialization Program (NIP) fixes all VS1 pages to avoid the duplicate paging.

Note: The working set size may be larger for a VS1 virtual machine in nonpaging mode than for one in paging mode.

Miscellaneous Enhancements

A VS1 virtual machine with the handshaking feature avoids many of the instructions or procedures that would duplicate the function that VM provides. For example, VS1 avoids:

- ISK (Insert Storage Key) instructions and uses a key table
- ENABLE/DISABLE sequences in the VS1 I/O Supervisor (IOS)
- TCH (Test Channel) instructions preceding SIO (Start I/O) instructions.

CP Interrupt Handling

Interrupt processing occurs within the CP environment. More than 30 modules control the process of interrupting events brought about by CP or virtual machine activity. Each module handles a particular I/O device or class or a function of CP (for example, timers, paging, SVCs). For an overview of interrupt handling, see Figure 25.

Program Interrupt

Program interrupts occur in two states. If the CPU is in the supervisor state, the interrupt usually indicates a system failure in the CP nucleus and causes a system abnormal termination. If the CPU is in the problem state, a virtual machine is in execution. If the program interrupt indicates that the Dynamic Address Translation (DAT) feature has an exception, a virtual machine issued a privileged instruction, or a protection exception occurred for a shared segment system, CP takes control and performs any required processing to satisfy the exception. Usually, the interrupt is not apparent to the virtual machine. Most other program interrupts result from virtual machine processing and are reflected to the virtual machine for handling.

When a program interrupt occurs, the program interrupt handler (DMKPRG) is entered. Program interrupts can result from:

- Normal paging requests
- A paging request by a virtual machine in EC mode (virtual relocate mode)
- Privileged instructions
- PER events
- Program errors
- Monitor calls.

For information about paging requests, see "Allocation Management" in this section.

Type	Module
SVC	DMKSVCIN
External	DMKPSAEX
Machine Check	DMKMCHIN
I/O	DMKIOSIN
Program Check	DMKPRGIN

Interrupt Handler Modules

Interrupt From	Action/Module
Unknown channel	Ignored - DMKDSPCH
Unsolicited device end	Build IOBLOK
and for:	
Console	DMKCNSIN
3270s on BSC lines	DMKRGA or DMKRGB
Local 3270, 3158, and 3066 consoles	DMKGRF
Unit record, real spooling	DMKRSPEX
Solicited device end	DMKSTKIO
Channel error	DMKCCHNT
Monitor tape I/O operation	DMKMONIO
Dedicated device error - DASD	DMKDASER, DMKDADER or DMKDAUER
Dedicated device error - Tape	DMKTAPER
3270 BSC line and channel errors	DMKBSC
Recoverable errors	DMKSTKIO
Unrecoverable errors	DMKIOERR

I/O Interrupt Handler (DMKIOT) Actions

Reason for Program Check	Module
Normal paging	DMKPTRAN
Paging - virtual machine in EC mode	DMKVAT
Supervisor State	DMKDMP
Vector Facility	DMKPRVLG
Privileged instruction	DMKPRVLG or DMKFPS
DIAGNOSE	DMKHVC
Timers	DMKTMR
Virtual Machine I/O	DMKVSIEX
console	DMKVCNEX
unit record, virtual spooling	DMKVSPEX

Program Check Interrupt Handler (DMKPRG) Actions

Figure 25. Overview of Interrupt Handling

Privileged Instructions

If a program interrupt is caused by the virtual machine issuing a privileged instruction when it is running in supervisor state, DMKPRVLG or DMKFPS obtains the address of the privileged instruction and determines the type of operation requested. If the virtual machine was running in problem state, the interrupt is reflected back to the virtual machine.

For I/O privileged instructions, DMKPRVLG transfers control to the virtual I/O executive program (DMKVSIEX).

Missing Interrupt Handler

A missing interrupt condition exists when an I/O device fails to return an interrupt to the control program after a specified time interval. The missing interrupt handler automatically monitors system I/O activity to detect this condition and to attempt to correct it.

To detect missing interrupts, at system initialization module DMKCPI schedules TRQBLOKs to enter module DMKDID. DMKDID tests the RDEVBUZY and RDEVSCHD flag bits in each RDEVBLOK. It sets RDEVMID to one when it scans the RDEVBLOKs and finds RDEVBUZY or RDEVSCHD on, indicating that a device interrupt is pending for this interval.

Then, the first-level interrupt handler, DMKIOT, resets the flags when the device returns an interrupt. If both flags are on when DMKDID again receives control, a missing interrupt condition exists. CP attempts to correct the condition.

If MIH is set on, CP attempts to simulate the interrupt. If CP cannot simulate the interrupt, the user receives an interface control check (IFCC). CP then writes a record to the recording area, sends a message to inform the system operator of the missing interrupt, and tells him whether or not the missing interrupt was cleared.

If MIH is set off, CP pursues no corrective action, but sends a message to the operator informing him that a missing interrupt was detected. CP also writes a record to the error recording area.

The user can set MIH on by specifying an option in the directory, or by issuing the class G SET command.

If you enable tracing activity, CP traces the missing interrupt and records it as trace table entry X'19'. See the *Virtual Machine Diagnosis Guide* for more information about tracing.

Interrupt timing varies widely among devices. Certain devices are more critical than others. To allow greater flexibility for monitoring I/O activity, CP specifies a different time interval for each class of device. You can change the IBM supplied defaults (provided in module DMKSYS) in two ways. You can change the values supplied in the SYSMIH macro statement in DMKSYS and reassemble DMKSYS. Or you can use the SET MITIME command.

The default intervals and the devices monitored are:

CLASDASD and CLASFBA	15 seconds
CLASGRAF	30 seconds (except TYP1053 and TYP328X)
CLASTAPE	10 minutes
CLASURI and CLASURO	1 minute (except TYP3800 and TYP3289E)
Miscellaneous devices	12 minutes

Note: Miscellaneous devices include MSS devices, TYP1053X and TYP328X graphic devices, and TYP3800 and TYP3289E unit record output devices.

The SET MITIME command dynamically changes the intervals specified in DMKSYS. Time intervals used in the SET MITIME command remain in effect until you issue another SET MITIME command, or until the system is reloaded (through IPL). SET MITIME is described in the *VM/SP HPO Operator's Guide*. See the *VM/SP HPO Planning Guide and Reference* for a description of the SYSMIH macro statement. For the method of operation refer to the topic "Monitoring I/O Operations" later in this publication.

I/O Interrupts

I/O interrupts from completed I/O operations initiate various completion routines and the scheduling of further I/O requests. The I/O interrupt handling routine also gathers device sense information.

Machine Check Interrupts

When a machine check occurs, CP Recovery Management Support (RMS) gains control to save data associated with the failure for FE maintenance. RMS analyzes the failure and determines the extent of damage.

Damage assessment results in one or more of the following actions being taken:

- System termination
- In attached processor or multiprocessor configurations, a processor or an attached processor is varied offline (system converts to uniprocessor mode)
- A Vector Facility is disabled
- One or more channels are marked offline
- Virtual user running at the time of error is terminated
- Refreshing of damaged information with no effect on system configuration
- Refreshing of damaged information with the defective storage page removed from further system use
- Error recording only for certain soft machine checks.

The system operator is informed of all actions taken by the RMS routines. When a machine check occurs during startup (before the system is set up well enough to permit RMS to operate successfully), the processor goes into a disabled wait state and places a completion code of X'00B' in the leftmost bytes of the current PSW.

SVC Interrupts

When an SVC interrupt occurs, the SVC interrupt routine (DMKSVCIN) is entered. If the machine is in the problem state, DMKSVC branches to DMKSVDIN. DMKSVDIN takes the following action:

- If the interruption was the result of an ADSTOP (SVC code X'B3'), the message ADSTOP AT XXXXX is sent to the user's terminal, the overlaid instruction is replaced, and the virtual machine is placed in console function mode (CP mode) via DMKCFMBK.
- If the interruption was the result of an error recording interface (SVC 76), DMKSVD checks for valid parameters and passes control to DMKVER to convert virtual device addresses in the error record to real device addresses. The actual recording is accomplished in DMKIOE and DMKIOF. If recording is not possible, the interrupt is reflected back to the virtual machine.
- If the virtual machine's page 0 was not in real storage, then all general and floating-point registers are saved, the user's VMBLOK is flagged as being in an instruction wait, and control is transferred (via GOTO) to DMKPRGRF to reflect the interruption to the virtual machine.
- If the virtual machine's page 0 is in main storage, an appropriate SVC old PSW is stored in the user's page 0 and the interruption is reflected to the virtual machine, bypassing unnecessary register saving (fast reflection). If the new virtual PSW indicates a mode or enablement change, all registers are saved in the VMBLOK and control is transferred to DMKDSPB for PSW validation.

If the machine is in the supervisor state, then DMKSVC determines the SVC interruption code and branches to the appropriate SVC interruption handler.

SVC 0
Impossible condition or terminal error. The SVCDIE routine initiates an abnormal termination by using the DMKDMPDK routine.

SVC 4
Reserved for IBM use.

SVC 8
A link request that transfers control from the calling routine to the routine specified by register 15. The SVCLINK routine sets up a new save area, and then saves the caller's base register in register 12 and save area address in register 13, and the return address (from the SVCOPSW) in the new save area. If the called routine is within the resident CP nucleus, SVCLINK places its address in register 12 and branches directly to the called routine. If the called routine is in a pageable module, a TRANS macro is performed for register 12 to ensure that the page containing the called routine is in storage. Upon return from the TRANS execution, the real address of the pageable routine is placed in register 12 and SVCLINK branches to the called routine. The real storage location of DMKCPE is the end of the resident CP nucleus. Any modules loaded at a higher real storage address

are defined as pageable modules. If bit zero of register 15 is on when DMKSVC is entered, then the caller has requested AFFINITY. DMKSVC turns on a bit in the save area passed to the caller to indicate that control is to be returned to the caller on the same processor on which it was running before issuing the SVC. It is not ensured that control will be retained by the initiating processor throughout the called operation, but only that final return will occur on the initiating processor.

SVC 12 (X'0C')
A return request that transfers control from the called routine to the calling routine. The SVCRET routine is invoked. If the routine that issued the SVC 12 is pageable, then DMKPTRUL is called to unlock the page. SVCRET then restores registers 12 and 13 (addressability and save area address saved by SVCLINK), places the user's return address (also saved in this area) back into the SVCOPSW, and returns control to the calling routine by loading the SVCOPSW.

SVC 16 (X'10')
Releases current save area from the active chain (removes linkage pointers to the calling routine). The SVCRLSE routine releases the current save area by placing the address of the next higher save area in register 13 and returns control to the current routine by loading the SVCOPSW. This SVC is used by second level interrupt handlers to bypass returning the first-level handler under specific circumstances. The base address field (register 12) in the save area being released is examined to determine if the bypassed routine is in a pageable module. If so, DMKPTRUL is called to unlock the page.

SVC 20 (X'14')
Obtain a new save area. The SVCGET routine places the address of the next available save area in register 13 and the address of the previous save area in the save area pointer field of the current save area.

SVC 24 (X'18')
In attached processor mode, SVC 24 causes the instructions following the SVC to be executed by the main processor. This SVC is used only via the SWITCH macro to force processing to continue on the main processor (the processor capable of performing I/O). If the SWITCH macro determines that the code is currently running on the main processor then the SVC is not issued.

In multiprocessor mode, the SWITCH macro is coded with the 'PROC' operand. SVC 24 causes the instructions following the SVC to be executed on the processor specified by the "PROC = " operand.

Save areas are initially set up by DMKCPI for use by the SVC linkage handlers. There is one list of save areas per processor. The number of save areas to be pre-allocated is determined by DMKCPI based upon the model number of the processor and upon whether or not the system was AP or MP generated.

External Interrupts

Timer Interrupt

If DMKEXTIN is entered because of a timer interruption, the state of the machine must be determined. If the virtual machine was in wait state, control is transferred to DMKDSPCH, and the virtual machine stays idle until another interruption occurs. If the virtual machine is in problem state, the address of the current user's VMBLOK is obtained from RUNUSER. The user's current PSW (VMPSW) is updated from the external interruption old PSW, the address of the current VMBLOK is placed in register 11, and control is transferred to DMKDSPCH. For additional information about timers, see "Virtual Timer Maintenance."

External Interrupts

If DMKEXTIN is entered because the operator pressed the console interrupt button (INTERRUPT), a CPEXBLOK is stacked to do the following:

- If Multiprocessor mode (MP mode), continue processing on the processor which has an I/O path to the operator's console.
- Reference the current system operator's VMBLOK (DMKSYSOP).
- Disconnect this virtual machine.

The operator can now log on from another terminal. Pressing the console interrupt button activates an alternate operator's console.

Note: If this interrupt comes from the attached processor, it is ignored.

For a description of the processing of the external interruption command, refer to module DMKCPB in the Entry Point Directory.

For a discussion of external interrupts that occur in attached processor or multiprocessor mode, see "Multiprocessor External Interrupts" under "Machine Check Handler – Attached Processor and Multiprocessor Applications."

Extended Virtual External Interrupts

To reflect external interrupts to a virtual machine, DMKDSPE queues an XINTBLOK on a chain pointed to by VMPXINT in the VMBLOK. The XINTBLOKs are chained sequentially by the XINTSORT field that contains the collating number of the pending interruption. If more than one interruption has the same collating number, the interruption codes are ORed together in the XINTCODE field for possible simultaneous reflection.

When a virtual machine is enabled for external interrupts, the XINTBLOK queue for that machine is searched for an eligible block. An XINTBLOK is eligible for reflection if one or more bits of the XINTMASK field match the bits in the rightmost halfword of control register 0. If the interruption was an interruption such as CPU timer or clock comparator, the block is left chained because reflection does not reset these interrupts. If the reflected interruption(s) does not represent all those coded in the XINTMASK field, the block is left chained and only the interrupts that were reflected are reset. In all other conditions, the XINTBLOK is unchained and returned to free storage.

A special external interrupt, code X'4001' notifies a virtual machine of a pending Virtual Machine Communication Facility request. The XINTBLOK for this interrupt is set up with an XINTSORT field of X'7FFFFFFF', the lowest priority.

System Support

Free Storage Management

During its execution, CP occasionally requires small blocks of storage that are used for the duration of a task. CP obtains this storage from the free storage area. The free storage area is divided into various size subpools for all requests that are smaller than or equal to 1,024 doublewords. Requests for 128 doublewords or fewer are rounded to 2 doubleword boundaries (the subpool width is 2 doublewords). Requests above 128 doublewords and fewer than 1024 doublewords are rounded to 32 doubleword boundaries (subpool width is 32 doublewords).

The requester informs the free storage manager of the size of the block required. Two stacks are maintained for each logical subpool so that extended items are isolated from nonextended items and only used if the nonextended stack is empty. All nonextended items are time stamped when they are pushed onto the subpool stack.

When a user logs off, for example, (DMKFRTRS is called) all extended items are returned to the free list, but only those nonextended items that haven't been used for some time interval (an algorithm parameter) are returned to the free list. This saves storage on large systems.

If the request for free storage cannot be fulfilled, the free storage manager requests the temporary use of a page of storage from the dynamic paging area. If a page is obtained, the page is chained to the free storage area and used for that purpose until it is no longer needed and subsequently returned to the dynamic paging area.

If the request for a page cannot be fulfilled, the requester waits until free storage becomes available.

Storage Protection

CP provides both fetch and store protection for real storage. The contents of real storage are protected from destruction or misuse caused by erroneous or unauthorized storing of fetching by the program. Storage is protected from improper storing or from both improper storing and fetching, but not from improper fetching alone.

When the processor accesses storage, and protection applies, the protection key of the current PSW is used as the comparand. The protection key of the processor is bit positions 8-11 of the PSW.

If the processor access is prohibited because of a protection violation, the operation is suppressed or terminated, and a program interruption for a protection exception takes place.

When the reference is made to a channel, and protection applies, the protection key associated with the I/O operation is used as the comparand. The protection key for an I/O operation is in bit positions 0-3 of the CAW and is recorded in bit positions 0-3 of the CSW stored as a result of an I/O operation. If channel access is prohibited, the CSW stored as a result of the operation indicates a protection-check condition.

When a storage access is prohibited because of a store protection violation, the contents of the protected location remain unchanged. If a fetch protection violation occurs, the protected information is not loaded into an addressable register, moved to another storage location, or provided to an I/O device.

To use fetch protection, a virtual machine must execute the set storage key (SSK) or set storage key extended (SSKE) instruction. The fetch protection bit in the storage key refers to the data area that is to be protected. CP subsequently:

1. Checks for a fetch protection violation when handling privileged and nonprivileged instructions.

2. Saves and restores the fetch protection bit (in the virtual storage key) when writing and recovering virtual machine pages from the paging device.

3. Checks for a fetch protection violation on a write CCW (except for spooling or console devices).

A special case of storage protection occurs when the CMS nucleus resides in a protected shared segment. The CMS nucleus may be protected and still be shared by many CMS users. After a virtual machine has used a protected shared segment, CP checks the pages of the segment for changes. If a virtual machine user has changed a page in a protected shared segment using an ADSTOP, STORE, or TRACE command, that virtual machine receives a private copy of the segment and virtual machine execution continues. If a virtual machine user has changed a shared page by any other means, CP issues an error message and places the virtual machine in console function mode.

If the segment protection feature is active, the virtual machine user receives program check PROG04 when attempting to alter a shared segment page.

Storage Validation

At system load, the loader (DMKLD00E) uses the TB instruction as it relocates itself to the high-end of storage and as it loads the system modules into storage. The system nucleus must reside in contiguous storage. If an unusable or non-addressable frame is detected within the area reserved for the nucleus, the system load is terminated with a disabled wait state code X'AAAAAA'. There is one exception. Non-addressable frames and frames having errors encountered in the virtual=real area do not cause a disabled wait state at system load. Instead, informational messages are sent to the system operator. This presents a special consideration for virtual machine operating systems running V=R. The V=R guest should use appropriate storage techniques to validate V=R storage and avoid machine errors in real storage.

The system cannot be initialized on any supported processor when an unusable or non-addressable frame is found either at location 0 of main storage or from DMKSLC through the frame in main storage where DMKSAV ends. System initialization routines DMKCKP, DMKSAV, and DMKSTA issue the TB instruction to determine the status of every frame of real storage. System initialization will fail with unpredictable results if DMKCKP or DMKSAV cannot be loaded at their expected real storage locations. If a non-addressable frame or a frame containing errors is detected within the area reserved for the nucleus (excluding the V=R area), system initialization is terminated with a disabled wait state code X'14'. Storage frames reserved for the V=R area are not validated at system initialization. V=R area frames are validated only at system load time as described above. Non-addressable and invalid frames encountered outside the area occupied by the system modules are identified to the system operator by a series of informational messages.

Executing the Pageable Control Program

Calls to pageable routines are recognized at execution time by the SVC 8 linkage manager in DMKSVC. For every SVC 8, the called address (in the caller's GPR15) is tested to see if it is within the resident nucleus. If it is less than DMKCPEND and greater than DMKSLC, the called routine's base address is placed in GPR12 and control is passed to the called routine in the normal way. However, if the called address is above DMKCPEND or below DMKSLC, the linkage manager issues a TRANS macro, requesting the paging manager to locate and, if necessary, page-in the called routine. The TRANS is issued with LOCK option. Thus, the lock count associated with the called routine's real page indicates the responsibility count of the module.

- When the module is called, the count is incremented.
- When the routine exits via SVC 12, the count is decremented.

When the count reaches zero, the pageable routine is unlocked and is eligible to be paged out of the system. However, because all CP pageable modules are reenterable, the page is never swapped out, but when the page is stolen, it is placed directly on the free page list.

Because unlocked pageable routines participate in the paging process in a manner similar to user virtual storage pages, the least recently used approximation used by page selection tends to make highly used control program routines, even when not locked, remain resident. The called routine is locked into real storage until it exits. Thus, it can request asynchronously scheduled function, such as I/O or timer interrupts, as long as it dynamically establishes the interruption return address for the requested operation and does not give up control via an EXIT macro prior to receiving the requested interruption.

Addressability for the module, while it is executing, is guaranteed because the CALL linkage loads the real address of the paged module into GPR12 (the module base register) prior to passing control. If all addressing is done in a base/displacement form, the fact that the module is executing at an address different from that at which it was loaded is not apparent. Although part of CP is pageable, it never runs in relocate mode. Thus, the processor is not degraded by the DAT feature being active, and no problems occur because of handling disabled page faults.

System Support Modules

The system support modules provide CP with several common functions for data conversion and control block scanning and verification. Most of the routines are linked to via the BALR option of the CALL macro, and make use of the BALRSAVE and TEMPSAVE work areas in DMKPSA. Two exceptions are the virtual and real I/O control block scan routines DMKSCNVU and DMKSCNRU. These routines do not alter the contents of the BALRSAVE area, and hence may be called by another low-level BALR routine.

Control Register Usage

Every IBM System/370 processor provides the program with 16 logical control registers (logical registers since the number that are active depends on the features installed in the machine at any one time) that are addressable for loading and storing from basic control (BC) mode. CP provides only a single control register, control register zero, for normal virtual machines, and for processing systems that do not require the full set of registers (for example, CMS, or VSE).

Any user whose virtual machine operating system requires the use of control registers other than control register zero, can request the full set of 16 registers by specifying the ECMODE option in the directory entry for his virtual machine.

A virtual machine, which utilizes any System/370 features that use the control registers, requires the ECMODE option. Some of these features are expanded timer support of the System/370 CPU timer, clock comparator, the virtual relocate mode and its instructions, RRB, LRA, PTLB, virtual monitor calls, virtual Program Event Recording (PER), etc.

Restrictions and Conventions for Pageable CP Modules

Pageable CP modules must observe the following restrictions and conventions when they are designed and coded:

- The module must be entered by the standard SVC 8 CALL linkage. Modules entered by BALR or GOTO cannot be pageable. The module must return to its caller by SVC also.
- The module cannot contain any A- or V-type address constants that point to locations within itself or within other pageable modules, and it cannot contain any CCWs that contain data addresses within themselves. The only exceptions are address constant literals generated as the result of calls to other modules (because these addresses are dynamically relocated at execution time, they must be resolved by the loader to the loaded address of the called module) and a pageable module that locks itself into storage. In practice, this restriction means that data or instructions within the pageable routine must be referenced via base/displacement addressing, and the address in register 15 for a CALL may not be generated by a LOAD ADDRESS instruction.
- The pageable module must be no more than 4096 bytes in length. (The one exception is DMKSTP.)

If the three above design and coding restrictions are adhered to, the CP module can be added to the existing pageable nucleus modules by utilizing the service routine, VMFLOAD, which is described in *VM/SP HPO Service Routines Program Logic*. Additional information can be found in the *VM/SP HPO Installation Guide*.

Figure 26 lists all the executable resident and pageable modules.

Executable Resident Modules

DMKACR	DMKGRE	DMKPSA	DMKTTX
DMKACS	DMKGRF	DMKPTR	DMKTTY
DMKBSC	DMKGRG	DMKPTS	DMKUNT
DMKCCD	DMKGRH	DMKPTT	DMKVAT
DMKCCF	DMKGRI	DMKQCN	DMKVAU
DMKCCH	DMKGRT	DMKQCO	DMKVCN
DMKCCO	DMKHVC	DMKQCP	DMKVCP
DMKCCS	DMKIOC	DMKQCQ	DMKVCQ
DMKCCT	DMKIOE	DMKQVM	DMKVCR
DMKCCW	DMKIOF	DMKRET	DMKVCS
DMKCFM	DMKIOJ	DMKRGA	DMKVCT
DMKCNS	DMKIOQ	DMKRGB	DMKVCU
DMKCNT	DMKIOS	DMKRGC	DMKVCV
DMKCPX	DMKIOT	DMKRGD	DMKVCW
DMKCSC	DMKIUA	DMKRGE	DMKVCX
DMKCVT	DMKIUB	DMKRNH	DMKVFR
DMKCVU	DMKIUE	DMKRPA	DMKVIO
DMKDAD	DMKIUN	DMKRSP	DMKVMA
DMKDAS	DMKIUS	DMKRST	DMKVRR
DMKDAU	DMKLOC	DMKSCH	DMKVRS
DMKDEX	DMKLOK	DMKSCN	DMKVSC
DMKDGD	DMKMCH	DMKSCO	DMKVSD
DMKDGF	DMKMCT	DMKSEL	DMKVSE
DMKDID	DMKMHC	DMKSPK	DMKVSF
DMKDMP	DMKMPO	DMKSSS	DMKVSG
DMKDMQ	DMKMSW	DMKSST	DMKVSI
DMKDRD	DMKOPR	DMKSSU	DMKVSJ
DMKDSB	DMKPAG	DMKSSV	DMKVSP
DMKDSP	DMKPAH	DMKSTK	DMKVSQ
DMKENT	DMKPGS	DMKSVC	DMKVSR
DMKEXT	DMKPGT	DMKSVD	DMKVST
DMKFPS	DMKPGU	DMKSWA	DMKVSU
DMKFRE	DMKPMA	DMKTBN	DMKVSV
DMKFRT	DMKPRG	DMKTMR	DMKVSW
DMKGRC	DMKPRV	DMKTRK	DMKVSX
DMKGRD	DMKPRW	DMKTRQ	DMKWAI

Figure 26 (Part 1 of 2). Executable Modules

Executable Pageable Modules

DMKACO	DMKCLK	DMKEPS	DMKOPE	DMKTRP
DMKALG	DMKCMD	DMKERM	DMKPEI	DMKTRT
DMKALO	DMKCPB	DMKERP	DMKPEL	DMKTRU
DMKAPI	DMKCPI	DMKGIO	DMKPEN	DMKTRX
DMKAPS	DMKCPJ	DMKGRA	DMKPEQ	DMKUCC
DMKAPT	DMKCPM	DMKHPS	DMKPER	DMKUDR
DMKAPU	DMKCPN	DMKHPT	DMKPET	DMKUDU
DMKAPV	DMKCPO	DMKHPU	DMKPGM	DMKURS
DMKAPW	DMKCPP	DMKHVD	DMKPST	DMKUSO
DMKAPX	DMKCPS	DMKHVE	DMKREI	DMKUSP
DMKAPY	DMKCPT	DMKHVF	DMKRPD	DMKUSQ
DMKAPZ	DMKCPU	DMKIDR	DMKRPI	DMKVBM
DMKATS	DMKCPV	DMKIDU	DMKRPW	DMKVCA
DMKBIO	DMKCPW	DMKIOG	DMKRSE	DMKVCB
DMKBLD	DMKCPY	DMKIOH	DMKRSF	DMKVCH
DMKCAC	DMKCPZ	DMKISM	DMKRSQ	DMKVDA
DMKCAO	DMKCQC	DMKIUC	DMKSAV	DMKVDB
DMKCDB	DMKCQG	DMKIUG	DMKSBL	DMKVDC
DMKCDM	DMKCQH	DMKIUJ	DMKSEG	DMKVDD
DMKCDS	DMKCQI	DMKIUL	DMKSEP	DMKVDE
DMKCFC	DMKCQP	DMKIUP	DMKSEV	DMKVDF
DMKCFD	DMKCQQ	DMKJRL	DMKSFB	DMKVDG
DMKCFF	DMKCQR	DMKLNK	DMKSIX	DMKVDH
DMKCFG	DMKCQS	DMKLNM	DMKSNC	DMKVDR
DMKCFH	DMKCQT	DMKLOG	DMKSND	DMKVDS
DMKCFJ	DMKCQU	DMKLOH	DMKSPC	DMKVDT
DMKCFO	DMKCQY	DMKLOJ	DMKSPL	DMKVER
DMKCFP	DMKCRM	DMKLOM	DMKSPM	DMKVFC
DMKCFQ	DMKCSB	DMKMCC	DMKSPS	DMKVFD
DMKCFR	DMKCSF	DMKMCD	DMKSPT	DMKVFE
DMKCFS	DMKCSO	DMKMCI	DMKSRM	DMKVFI
DMKCFT	DMKCSP	DMKMHV	DMKSTA	DMKVFS
DMKCFU	DMKCSQ	DMKMIA	DMKSTP	DMKVMC
DMKCFV	DMKCSR	DMKMID	DMKSTR	DMKVMD
DMKCFW	DMKCST	DMKMNI	DMKSWM	DMKVME
DMKCFY	DMKCSU	DMKMNJ	DMKTAP	DMKVMG
DMKCKD	DMKCSV	DMKMNL	DMKTAQ	DMKVMI
DMKCKF	DMKCSW	DMKMNT	DMKTCS	DMKWRM
DMKCKH	DMKCSX	DMKMON	DMKTCT	DMKWRN
DMKCKM	DMKCSY	DMKMOO	DMKTDK	DMKXAB
DMKCKN	DMKDEF	DMKMSG	DMKTHI	DMKXAD
DMKCKP	DMKDEG	DMKNEA	DMKTOD	DMKXST
DMKCKR	DMKDEI	DMKNEM	DMKTPE	DMKZTD
DMKCKS	DMKDIA	DMKNES	DMKTRA	
DMKCKT	DMKDIB	DMKNET	DMKTRC	
DMKCKV	DMKDIF	DMKNLD	DMKTRD	
DMKCKW	DMKEIG	DMKNLE	DMKTRM	

Figure 26 (Part 2 of 2). Executable Modules

Data Area Modules

In addition to the executable resident and pageable modules (see Figure 26), there are certain modules that only contain data areas and do not contain executable code. These modules are:

Resident Module	Contents
DMKCPE	Defines the end of the CP nucleus
DMKPXA	Extension to the PSA
DMKPXB	Extension to the PSA
DMKRIO	I/O device blocks
DMKSEQ	Printer separator logo
DMKSLC	Data to define the end of the CP nucleus
DMKSTD	Starting address of STDATA table
DMKSYS	System constants
DMKTBL	Terminal translate table
DMKTTZ	CCWs and data pointed to by certain CCWs for TTY terminals

Pageable Module	Contents
DMKBOX	Output separator table
DMKBTS	Bootstrap routines for 3705
DMKEMR	Response message data module
DMKFCB	3211-type Forms Control Buffer (FCB) load tables
DMKMES	CP message data module
DMKPIA	3289 Model 4 Font Offset Buffer (FOB) load tables
DMKPIB	3262 Universal Character Set Buffer (UCSB) load tables
DMKSNT	System name table (may exceed one page)
DMKSYM	System symbol table
DMKTBM	Terminal translate tables
DMKTBN	Terminal translate tables for APL/ASCII for TTY terminals.
DMKUCB	3211 Universal Character Set Buffer (UCSB) load tables
DMKUCS	1403 Universal Character Set (UCS) load tables
DMKVCC	3203 Universal Character Set Buffer (UCSB) load tables

Virtual Timer Maintenance

The System/370 with EC mode provides the system user (both real and virtual) with four timing facilities. They are:

- The interval timer at main storage location X'50'
- The time-of-day clock
- The time-of-day clock comparator
- The CPU timer.

Real Timing Facilities

Before describing how CP maintains these timers for virtual machines, it is necessary to review how CP uses the timing facilities of the real machine.

1. The location X'50' interval timer is used only for time-slicing. The value placed in the timer is the maximum length of time that the dispatched virtual machine is allowed to execute.

   Because the BLIP function of CMS uses the interval timer (location X'50'), the use of STIMER can cause extra blips at the user's terminal. To avoid extra blips, issue the CMS command SET BLIP OFF.

2. The time-of-day clock is used as a time stamp for messages and enables the scheduler to compute elapsed in-queue time for the dispatching priority calculation.

3. The time-of-day clock comparator facility is used by CP to schedule timer-driven events for both control program functions and for virtual machines. A stack of comparator requests is maintained and as clock comparator interrupts occur, the timer request blocks are stacked for the dispatcher via calls to DMKSTKIO.

4. The CPU timer facility performs three functions:

   - Accumulates CP overhead
   - Detects in-queue time slice end
   - Simulates virtual CPU timer.

   The accumulation of CP overhead is accomplished as follows. The VMTTIME field in the VMBLOK contains the total CP overhead incurred by the virtual machine—it is initialized to the maximum positive number in a doubleword, X'7FFFFFFF FFFFFFFF'. Whenever CP performs a service for a virtual machine, GR 11 is loaded with the address of the VMBLOK and the current value in VMTTIME is placed in the CPU timer. When CP is finished with the service for that virtual machine the CPU timer, which has been decremented by the amount of processor time used, is stored back into VMTTIME. GR 11 is then loaded with a new VMBLOK pointer and the CPU timer is set from the new VMTTIME field. The amount of CP overhead for a given virtual machine at any point in time is the difference between the maximum integer and the current value in the VMTTIME field.

   Since VMTTIME only accounts for supervisor state overhead, detection of in-queue time slice end is performed by the processor timer when the virtual machine is dispatched in the problem state. The VMTMOUTQ field in the VMBLOK is initialized to the amount of problem state time that the virtual machine is allowed to accumulate before being dropped from a queue. This initial value is set by the scheduler (DMKSCH) when the virtual machine is added to a queue and its value depends on the queue entered (interactive or noninteractive) and on the processor model. For example, the initial value of VMTMOUTQ for a user entering Q1 (interactive) on a Model 145 is 300 milliseconds, while for the same user entering Q2 (noninteractive) it is 2 seconds. Each time the user is dispatched, the value in VMTMOUTQ is entered into the CPU timer. Whenever the user is interrupted, the decremented processor timer is stored into VMTMOUTQ prior to being set from the new VMTTIME. When the problem state time slice has been exhausted, a CPU timer interrupt occurs, the VMQSEND flag bit is set in the VMBLOK, and the scheduler drops the user from the queue. At each queue drop, the problem time used in-queue (the difference between VMTMOUTQ and the initial value) is added to the total problem time field (VMVTIME) in the VMBLOK.

Virtual CPU timer simulation is handled for EC mode virtual machines if the value in the virtual processor timer is less than that in VMTMOUTQ. In this case, the VMBLOK is flagged as "tracking processor timer" and a processor timer interrupt is interpreted as a virtual timer interrupt rather than as an in-queue time slice end.

Virtual Timing Facilities

Virtual location X'50' timers are updated by the elapsed processor time each time the dispatcher has been entered after a running user has been interrupted. The size of the update is the difference between the value of the timer at dispatch (saved in QUANTUM at location X'54') and the value of the timer at the time of the interruption (saved in QUANTUMR at location X'4C').

Virtual clock comparator requests are handled by the virtual timer maintenance routine, DMKTMR. They are inserted into the general comparator request stack and the virtual machine is posted when the interruption occurs.

Virtual clock comparator requests to set the virtual processor timer place the new value into the ECBLOK. Requests to store the new value update the ECBLOK field with the virtual processor time used since the last entry to dispatch and pass the value to the user. Requests to set the time-of-day clock are ignored.

A real interval timer or processor timer is one that runs when the virtual machine is executing or is in a self-imposed wait state (that is, the wait bit is on in the virtual PSW). A real timer does not run if the virtual machine is in a CP pseudo wait state (for example, page wait or I/O wait) or if the virtual machine can be run but is not being dispatched because of other user interaction. Real timers provide accurate interrupts to programs that depend on measurement of elapsed processor and/or wait time. They do not accurately measure wall time—the TOD clock must be used for this function.

An EC mode virtual machine with the real timer option has both a real interval timer and a real processor timer. Real timer requests for waiting machines are maintained in the clock comparator stack. CPU timer requests are added to TOD clock value at the time that they are issued. Interval timer requests must have their units converted. In addition, if the virtual CPU timer contains a large negative value, then a real timer request is scheduled to occur when the virtual CPU timer becomes positive, so that the pending timer interruption can be unflagged. Comparator requests for real timer interruptions are inserted into the stack whenever a virtual machine enters a self-imposed wait. They are removed either when the virtual machine resumes execution or when it is forced (or places itself) into a pseudo wait.

I/O Management

I/O Supervisor

The module DMKIOS handles the I/O requirements of all system devices except the following terminals: 1052, 3210, 3215, 2150, 2741, and compatible teletypewriter devices. Scheduling and interruption handling for these devices is essentially a synchronous process and does not require the queuing and restart services of DMKIOS. This is handled by the module DMKCNS. For handling the I/O requirements of 3270 remote equipment, refer to “Programming for 3270 Remote Terminals - an Introduction” in this section.

Real I/O Control Blocks

To schedule I/O requests and control the activity of the I/O devices of the system, I/O control uses several types of control blocks. These blocks are separated into two basic types.

- Static blocks that describe the components of the I/O system.
- The dynamic blocks that represent active and pending requests for I/O operations.

The I/O devices of the real system are described by one control block for each channel, control unit, and device available to the control program. For multiprocessor generated systems, two sets of real channel blocks are created. Units present but not represented by control blocks are not available for either user-initiated or CP-initiated operations.

Because all virtual machines are run in the problem state, any attempt to issue a SIO instruction results in a program interruption that indicates a privileged operation exception. This interruption is handled by CP's first level program interrupt handler, DMKPRGIN. It determines if the virtual machine was in virtual supervisor state (problem state bit in the virtual PSW is zero). If so, the instruction causing the interruption is saved in the VMBLOK for the virtual machine and control is transferred to the privileged instruction simulator, DMKPRVLG, via a GOTO.

DMKPRVLG determines if the privileged operation affects the virtual I/O configuration. DMKPRVLG simulates non-I/O privileged instructions (such as LPSW). If the instruction's operation code is from X'9C' to X'9F', control is transferred to DMKVSIEX.

After clearing the condition code in the user's VMBLOK, DMKSCNVU is then called to locate the virtual I/O blocks representing the I/O components (channel, control unit and device) addressed by the instruction.

DMKVSIEX then branches to handle the request based on the operation requested.

In attached processor systems and multiprocessor systems, the I/O control blocks are protected by the I/O lock, a global spin lock.

Virtual I/O Requests

The virtual I/O interface maintained by CP provides to the software operating in the user's virtual machine, the condition codes, CSW status information, and interruptions necessary to make it appear to the user's virtual machine that it is in fact running on a real System/370. The virtual I/O interface consists of:

- A virtual I/O configuration for each active virtual machine that consists of a set of I/O control blocks that are maintained in the Control Program's free storage. This configuration is built at logon time from information contained in the user's directory file, and can be changed by the user or the system operator.
- A set of routines that maintain the status of the virtual I/O configuration.
- Other system routines that simulate or translate the channel programs provided by the user to initiate I/O on units in the real system's configuration.

Virtual SIO

With a SIO, the condition code returned from DMKSCNVU is tested to verify that all addressed components were located. If they were not, then a condition code of 3 (unit not available) is placed in the PSW and control returns to the dispatcher. Otherwise, the addresses of the appropriate virtual I/O control blocks are saved, and DMKVSIEX tests the status of the addressed I/O units by scanning the VCHBLOKs, VCUBLOKS, and VDEVBLOKs to locate the block that contains the status of the addressed subchannel. The subchannel status is indicated in:

- The VCHBLOK for a selector or block multiplexer channel.
- The VCUBLOK for a shared selector subchannel on a byte multiplexer channel.
- The VDEVBLOK for a nonshared subchannel on a byte multiplexer channel.

When the block containing the status is found, the status is tested. If the subchannel is busy or has an interruption pending, condition code 2 is placed in the virtual PSW. Otherwise, the subchannel is available and the device and the control unit are tested for interruption pending or busy. If either is found, condition code 1 is placed in the virtual PSW and the proper CSW status is stored in the virtual machine's page zero. If all components in the subchannel path are free, DMKVSIEX proceeds to simulate the SIO by locating and loading the contents of the virtual machine's CAW from virtual location X'48' and testing the device type of the unit addressed.

The device type is in the VDEVBLOK. If the device class code indicates a terminal or console, control is passed to the module DMKVCNEX with a GOTO. DMKVCNEX interprets and simulates the entire channel program, moving the necessary data to or from virtual storage and reflecting the proper interruptions and status bytes. When DMKVCNEX has finished, it passes control directly to the dispatcher, DMKDSPCH.

If the referenced device is a spooled unit record device, DMKVSIEX passes control to DMKVSPEX for additional processing. When control returns to DMKVSIEX, it passes control to DMKDSPCH.

If the device is not a terminal or a spooling device, the SIO is translated and executed directly on the real system's I/O device. DMKVSIEX calls DMKFREE to obtain free storage and then it constructs an IOBLOK in the storage obtained. The IOBLOK serves as an identifier of the I/O task to be performed. It contains a pointer to the channel program to be executed and the address of the routine that is to handle any interruptions associated with the operation.

DMKVSIEX stores the contents of the user's CAW in IOBCAW and sets the interruption return address (IOBIRA) to be the same as the virtual interruption return address (DMKVIOIN) in DMKVIO. The CCW translation routine (DMKCCWTR) is then called to locate and bring into real main storage all user pages associated with the channel program, including those containing data and CCWs. The following occurs:

- The CCWs are translated.
- A corresponding real channel program is constructed.
- The data pages are locked into real storage.
- DMKCCWTR returns control to DMKVSIEX. DMKVSIEX places the user in a pseudo wait state, IOWAIT, and calls the real I/O scheduler DMKIOSQV to schedule the I/O on the real configuration.

DMKIOSQV queues the request for operation on the real channel, control unit, and device corresponding to the address used by the virtual machine. When the real SIO is issued, DMKIOS takes the user out of IOWAIT and reflects the condition code for the SIO if it is zero. If it is not zero, the operation is further analyzed by DMKVIOIN. In any case, DMKIOSQV returns control to DMKVSIEX, which passes control to DMKDSPCH.

Other Privileged I/O Instructions

Other privileged I/O instructions are handled directly by DMKVSIEX. DMKVSIEX scans the virtual channel, control unit, and device blocks in the same manner as for a SIO and reflects the proper status and condition to the virtual machine. In some cases (TIO), the status of the addressed devices is altered after the status is presented.

If the operation active on the virtual device is actually in progress in the real equipment, the simulation of a HIO or HDV is somewhat more involved, since it requires the actual execution of the instruction. In this case, the active operation is halted and the resultant condition code/status is returned to the user.

Virtual Channel-to-Channel Support (CTCA and 3088)

Virtual channel-to-channel support simulates data transfer and control communication between two channel-to-channel devices, either on two distinct processors or two channels on a single processor. Data transfer is accomplished via synchronized complementary I/O commands (for example, read/write, write/read) issued to both parts of the CTC device. Each part of the CTC device is identical and the operation of the unit is completely symmetrical.

The VM/SP HPO control program support for virtual CTC devices (channel-to-channel adapter and 3088) includes all status data, sense data, and interrupt logic necessary to simulate the operation of the real CTC device. Data transfer, command byte exchange, sense data, and status data presentation for the virtual CTC device is accomplished via storage-to-storage operations (like MVCL). No real I/O operations (excluding paging I/O) or I/O interrupts are involved. Unit errors or control errors cannot occur.

Virtual Selector Channel I/O Requests

The CCW translator, DMKCCWTR, is called by the virtual machine I/O executive program (DMKVSIEX) when an I/O task block has been created and a list of virtual CCWs associated with a user's SIO request must be translated into real CCWs.

When the I/O operation from a self-modifying channel program is completed, DMKUNTIS is called by DMKIOS. When retranslation of OS ISAM CCWs is required, the self-modifying channel program checking portion of DMKCCWTR calls DMKISMTR.

DMKCCWTR operates in two phases:

- A scan and a translate phase
- A TIC-scan phase.

A self-modifying channel program checking function is also included.

The scan and translate phase analyzes the virtual CCW list. Some channel commands require additional doublewords for control information (for example, seek addresses). Additional control words are also allocated (in pairs) if the data area specified by a virtual CCW crosses 4096-byte page boundaries, or if the virtual CCW includes an IDA (indirect data address) flag.

Space is obtained from DMKFREE for the real CCW list, and the translation phase then translates the virtual CCW list into a real CCW list. TIC commands that cannot be immediately translated are flagged for later processing by the TIC-scan phase. A READ or WRITE command that specifies that data cross 4096-byte boundaries is revised to include an IDA flag that points to an indirect data address list (IDAL) and a pair of words for each 4096-byte page, in which each word handles a data transfer of 2048 bytes (or less). The real CCW is flagged as having a CP-generated IDA. DMKPTRAN is called (via the TRANS macro) to lock each 4096-byte page.

If the real CCW string does not fit in the allocated free storage block, a new block is obtained. The old block is transferred and adjusted before being released. The translation continues with the new block. The process is repeated, as needed, to contain the real CCW string.

Virtual CCWs having an IDA flag set are converted to user translated addresses for each IDAW (indirect data address word) in the virtual IDAL. DMKPTRAN is called for each IDAW. The CCW is flagged as having a user (but not CP) generated IDA.

The TIC-scan phase scans the real CCW list for flagged (untranslated) TIC commands and creates a new virtual CCW list for the untranslated commands. Scan-translate phase processing is then repeated. When all virtual CCWs are translated, the virtual CAW in the IOBLOK task block is replaced by the real CAW (that is, a pointer to the real CCW list created by DMKCCWTR), and DMKCCWTR returns control to DMKVIOEX. The user protection key is saved.

OS ISAM Handling by DMKISMTR

Because many of the OS PCP, MFT, and MVT ISAM channel programs are self-modifying, special handling is required by CP to allow virtual machines to use this access method. The particular CCWs that require special handling have the following general format:

	0 2 4 6
A	READDATA C+7 10 bytes
B	TIC to E
C	
D	
E	SEEK: SEEK head on D
F	SEARCH on D+2

The CCW at A reads 10 bytes of data. The tenth byte forms the command code of the CCW at E. In addition, the data read in makes up the seek and search arguments for the CCWs at E and F. After the CCW string is translated by CP, it usually is in the following format:

	0 2 4 6 8
1	READDATA C+7 10 bytes
2	TIC to 3
3	SEEK: SEEK head on 6
4	SEARCH on D+2
5	etc.
6	ISAM word

To accomplish an efficient and non-timing-dependent translated operation for OS ISAM, the virtual CCW string is modified in the following manner.

DMKISMTR is called by DMKCCWTR if, during normal translation, a CCW of the type at 1 is encountered. The scan program locates the TIC at 2 by searching the translated CCW strings. The TIC at 2 locates the SEEK at 3.

The virtual address of the virtual SEEK CCW at E is located from the RCWTASK header. Seven doublewords of free storage are obtained and the address of the block is saved in the ISAM control word at 5. The seven doublewords are used to save the following information from the translated CCW strings:

7	Address of Read at 1	Address of TIC at 2
8	Unused	Unused
9	Data area for READ at 1	
10	SEEK HEAD on 9	
11	TIC to 4	
12	Image of READ CCW at 1	
13	Image of TIC CCW at 2	

The translated read CCW (at 1) is moved to the save block at 12. The TIC CCW (at 2) is moved to the save block at 13, and the addresses of 1 and 2 are saved at 7. The read CCW at 1 is modified to point to a 10-byte data area at 8+7 in the save block. The seek head CCW at 3 is copied into the save block at 10, and the seek address is modified to point to the data area at 9. At 11, a TIC CCW is built to rejoin the translated CCW string at 4. The search at 4 (or any subsequent search referencing D+2) is modified to point to 9+2. The completed CCW string has the following format:

1	Readdata 8+7	10 Bytes
2	TIC to 10	
3	Unused	
4	Search on 9 + 2	
5	Etc.	
6		ISAM word
7		
8	Unused	
9	Data Area for Readdata	
10	Seek Head on 9	
11	TIC to 4	

The interruption return address in the IOBLOK is set to DMKUNTIS. DMKUNTIS restores the CCWs to their original format from the seven doubleword extensions, moves the 10 bytes of data from 8+7 into virtual storage (at C+7), and releases the block. Normal I/O handling is resumed by DMKVIO and DMKUNT.

I/O Component States

The I/O components represented by the control blocks described in "Real I/O Control Blocks" are in one of four states and the state is indicated by the flag bits in the block status byte. If the component is not disabled, it is either busy, scheduled, or available.

If the disabled bit is on, the component has been taken offline by the operator or the system and is at least temporarily unavailable. A request to use a disabled component causes the IOBLOK to be stacked with an indication of condition code 3 on the SIO and the real SIO is not performed.

An I/O unit is busy if it is transferring data (in the case of a channel or control unit), or if it is in physical motion (in the case of a device). If an I/O unit is busy, the IOBLOK for the request is queued from the control block representing that I/O unit.

An I/O unit is scheduled if it is not busy but will become busy after a higher-level component in the subchannel path becomes available and an operation is started. For example, if a request is made to read from a tape drive and the drive and control unit are available, but the channel is busy, the IOBLOK for that request is queued from the RCHBLOK for the busy channel and the RCUBLOK and RDEVBLOK of the drive and control unit are marked as scheduled. Future requests to that drive are queued from the RDEVBLOK for the scheduled device. When the channel completes the operation, the next pending operation is dequeued and started. The scheduled control unit and device are then marked as busy.

The IOBLOKs for various I/O requests indicate the status of that request by a combination of the status bits in the IOBLOK and the queue in which the block resides. In general, an IOBLOK is queued from the control block of the highest level I/O unit (taken from device up to channel) in the subchannel path that is not available. Once the I/O operation is started, the IOBLOK is chained from the active IOBLOK pointer (RDEVAIOB) in the real device control block. Flags in the IOBLOK status fields may also indicate that a unit check has occurred, that a sense is in progress, or that a fatal I/O error (unrecoverable) has been recognized by error recovery procedures. After I/O control releases control of the IOBLOK, it is stacked on the queue of IOBLOKS and CPEXBLOKs anchored at DMKDSPRQ in the dispatcher and control is passed to the second-level interruption handler whose address is stored in IOBIRA.

I/O Interrupts

I/O interrupts are either synchronous or asynchronous. Asynchronous interrupts indicate the change in status of an I/O unit from the not-ready to ready state or busy to not-busy state. In either case, if the affected component has any pending requests queued from its control block, they are restarted, and whether or not the given interrupt is processed any further depends upon the status of the interrupting component. Channel-available and control-unit-end interrupts restart the interrupting component. An asynchronous device end is passed to the user if the device is dedicated. Otherwise, the device is restarted.

An interrupt is considered to be synchronous if the interrupting device has a nonzero pointer to an active IOBLOK. In this case, the following processing occurs:

- If a unit check has occurred, a sense is scheduled, and when the sense is completed, the appropriate ERP is called.
- If an ERP is currently in control of the task (indicated by a flag in the IOBLOK), return the IOBLOK to the appropriate ERP.
- If the operation is incomplete (for example, channel end is received without device end), the IOBLOK is copied and the copy is stacked but the original IOBLOK remains attached to RDEVAIOB to receive the final interrupt. The control unit and channel are then restarted.
- If the operation is complete (that is, the device is available), the IOBLOK is detached from the device and stacked, and the device, control unit and channel are restarted.

The restart operation usually dequeues the next IOBLOK that is queued to the restarted component and queues it to the next higher component in the subchannel path. When the channel level is reached, a SIO is issued and exit is taken to the dispatcher after handling any nonzero condition codes as previously described.

Virtual I/O Interruptions

When an I/O interruption is received, the IOBLOK is stacked for dispatching and control is passed to the address specified in the IOBIRA (interrupt return address) field. For operations requested by DMKVIOEX, the return address is DMKVIOIN (virtual interrupt return address). When DMKVIOIN receives control from the dispatcher, it loads the virtual address of the unit with which the interruption is associated from the IOBLOK and calls DMKSCNVU to locate the virtual device control blocks. DMKVIOIN then tests the IOBLOK status field to determine the cause for the interruption. If the block has been unstacked because of an interruption, the field is zero. If the operation was not started, it contains the condition code from the real SIO.

Note: The VIRA should not see a real condition code 2 as the result of a SIO, since channel-busy conditions are detected and reflected before any real I/O operation is attempted.

A condition code of 3 is reflected to the virtual machine and exit is taken to the dispatcher. For a condition code of 1, the CSW status field in the IOBLOK is examined to determine the cause for the CSW stored condition. The status is reflected to the virtual machine and various components of the virtual configuration may be freed, if the status so indicates. For example, if the CSW status indicated both channel end and device end, the operation was immediate and has completed. Thus, the CCW string (real) may be released and all virtual components marked available.

The CSW status returned for a virtual interruption must be tested in the same manner, with the additional requirement that the status be saved in the affected virtual I/O control blocks and that the CSW be saved in the VDEVCSW field for the device causing the interruption. If the unit check bit is on in the status field, the sense information saved in the associated IOERBLOK (pointed to by the IOBLOK) must be retained so that a sense initiated by the virtual machine receives the proper information.

In any case, when an interruption is received for a virtual device, a bit in the interruption mask, VCUDVINT, for the device's control unit is set to 1. The bit that is set is the one corresponding to the relative address of the interrupting device on the control unit. For example, if device 235 interrupts, the fifth bit in the VCUDVINT mask in the VCUBLOK for control unit 30 on channel 2 is flagged. Similarly, the bit in the VCHCUINT in the affected VCHBLOK is also set. In this case, bit 3 in VCHBLOK for channel 2. If the interruption is a channel class interrupt (PCI or CE), the address of the interrupting unit (235) is stored in the VCHCEDEV field in the VCHBLOK. The final interruption flag is set in the VMPEND field in the VMBLOK for the interrupted virtual machine. The bit set corresponds to the address of the interrupting channel. The next time, the virtual machine is dispatched and becomes enabled for I/O.

Monitoring I/O Activity

The missing interrupt handler, module DMKDID, receives control from a timer interrupt (TRQBLOK). The timer is reset, and DMKDID scans all of the real device blocks. If the RDEVBUZY or RDEVSCHD indicator flag is on, active I/O or device busy conditions exist. A check is made to make certain that the RDEVBUZY flag and the timer interrupt are for the same class of device. If the class is valid, the RDEVMID bit indicator-flag is turned on. If the class is not valid, the scan continues.

RDEVMID on means that the device is active for the time interval and a device interrupt is pending. When the device causes an interrupt, the indicator flags are reset by the first level interrupt handler, DMKIOT. If both flags (RDEVMID and RDEVBUZY) are on at the end of the *next* interval, a missing interrupt condition exists.

When a missing interrupt condition is detected, a CPEXBLOK is set up to give control to module DMKDID. The action taken depends on whether the request was queued or active. If the request was queued, a control unit end or device end is simulated. If the request was active, an interface control check is simulated.

If a virtual machine initiated the I/O operation, the interface control check is returned to the virtual machine. If CP started the I/O operation, the interface control check is handled by CP's ERPs.

Before either action is taken, a ten second timer is scheduled to return control to module DMKDID. When control is received from the timer, the RDEVMID flag is examined. If it is off (indicating that some I/O was completed), DMKDID sends a message to the system operator and a record is written to the error recording area indicating that the condition is corrected. If RDEVMID is on, the operator message and error record indicate the condition was not cleared.

Scheduling I/O Requests

A task that requests an I/O operation must specify the device on which the operation is to take place and must provide an IOBLOK that describes the operation. Upon entry to DMKIOS, register 10 must point to the IOBLOK. The IOBLOK must contain at least a pointer to the channel program to be started in IOBCAW and the address to which the dispatcher is to pass control in IOBIRA. In addition, the flags and status fields should be set to zero. If the operation is a control program function such as for spooling or paging, the entry point DMKIOSQR is called. If the requester is the virtual I/O executive (DMKVIOEX) attempting to start a virtual machine operation, the entry point DMKIOSQV is called and some additional housekeeping is done. In either case, an attempt is made to find an available subchannel path from the device to its control unit and channel. If an I/O unit in the path is busy or scheduled, the IOBLOK for the request is queued to the control block of the I/O unit.

Requests are usually queued first-in-first-out (FIFO), except those requests:

- To fixed-head DASD primary paging areas that are queued first
- To movable-head DASDs that are queued in order of seek address
- That release the affected component after initiation (SEEKS and other control commands) which are queued last-in-first-out (LIFO) from the control block.

Whether or not the operation has been successfully started, the caller requesting the I/O operation receives control from DMKIOS. If a free path to the device is found, the unit address is constructed and an SIO is issued. If the resulting condition code is zero, control is returned to the caller. Otherwise, the code is stored in the requester's IOBLOK along with any pertinent CSW status, the IOBLOK is stacked, any components that become available are restarted, and control is returned to the caller.

In a multiprocessor environment, both processors have I/O capability. If either processor receives an I/O request, that processor attempts to initiate I/O operations.

At system generation time, when a channel path to a device is defined on one processor, an alternate logical path is automatically defined for the other processor. Thus, both processors *can* have access to any I/O device in the MP configuration.

If either processor receives an I/O request, that processor attempts to initiate the I/O operation on one of its own paths to the required device. If none of the online paths to the required device are available from the executing processor, that processor queues the I/O request on all busy and scheduled paths to the device, both its own and those of the other processor. If there is no online path from the executing processor, that processor queues the I/O request on the first online and available path for the second processor, as well as on all busy or scheduled paths for that processor.

While it is not required that both processors have access to all I/O devices, heavily used devices should be accessible by both processors to provide efficient system operation and to increase the possibility of system recovery following a processor or channel failure.

The I/O lock serializes access to the control blocks that represent I/O devices.

Alternate Path Scheduling

Alternate path I/O scheduling is performed according to the following scheme:

DMKIOQ, called by DMKIOS, searches for an available path beginning with the primary path to the device. If an available path to the device exists, the I/O request is started immediately on the first available path to the device.

If the device is busy or scheduled, the IOBLOK is queued off the RDEVBLOK. No alternate path processing is performed at the device level.

If the device is not busy, not scheduled, nor offline, an IOBLOK for this I/O request is promoted upward to the RCUBLOK.

If the RCUBLOK is marked busy, the IOBLOK is queued on the RCUBLOK and a search is made for an alternate control unit path. If the RCUBLOK is marked scheduled and the present request will not release the control unit (as in TAPE FSF and TAPE BSF), then the IOBLOK is queued off the RCUBLOK and a search is made for an alternate control unit path. If the RCUBLOK is marked scheduled and the present request will release the control unit, the search continues for a channel path. If the RCUBLOK is not marked scheduled or busy but other I/O requests are queued on the RCUBLOK, the check is again made to see if the present request will release the control unit. If the present request will not release the control unit, the request is queued and a search is made for an alternate control unit path. Otherwise, the search continues for a channel path.

The RCUBLOK busy and scheduled indicators are turned on only for shared control units. The busy and scheduled indicators are turned on in the RCUBLOK for tape control units. The nonshared DASD RCUBLOKs never have the busy and scheduled indicators turned on. For this reason, alternate control unit path selection rarely takes place for nonshared control units. The one exception occurs when the channel path through the first control appears busy (because a real-channel-busy condition was encountered). If an alternate path exists through a second control unit, the control blocks associated with the second control unit path are examined. Finding an available channel path is the final step before issuing the SIO. If the RCHBLOK is marked busy, a search is made for an alternate channel path. If the RCHBLOK has other requests queued on the RCHBLOK, a search for an alternate channel path is made. VM/370 never marks a byte multiplexer RCHBLOK busy. The only time a byte multiplexer is marked busy is after a condition code 2 has been encountered. The I/O load on byte multiplexer channels must be sufficient to cause channel-busy conditions before path selection on an alternate channel can take place.

If a busy or scheduled path is encountered, an IOBLOK is queued to the real block (RCUBLOK or RCHBLOK) and the search continues for an available path. If more than one busy path is encountered, multiple IOBLOKs are queued for the same I/O request. This is accomplished by creating mini IOBLOKs for each busy/scheduled path after the first. The primary IOBLOK is queued off the first busy path encountered. The mini IOBLOK is 16 bytes in length and consists of the first two doublewords of the IOBLOK, which is the same as the current IOBLOK structure. The IOBLOK and associated mini IOBLOKs are chained in a single-threaded queue by means of the IOBLINK field. The active IOBLOK pointer is not stored in the IOBLINK field until just prior to the SIO. Zeros are stored in IOBLINK at entry to DMKIOSQR to indicate no mini IOBLOKs have been queued as yet. See Figure 27 for an example of mini IOBLOK queuing.

The last two words of the mini IOBLOK (IOBFPNT and IOBBPNT) are used as the double-threaded queue pointers for the RCUBLOK/RCHBLOK from which it is queued. A flag is set in the mini IOBLOK to identify it as a mini IOBLOK.

Figure 28 shows a sample control block structure when mini IOBLOKs are queued.

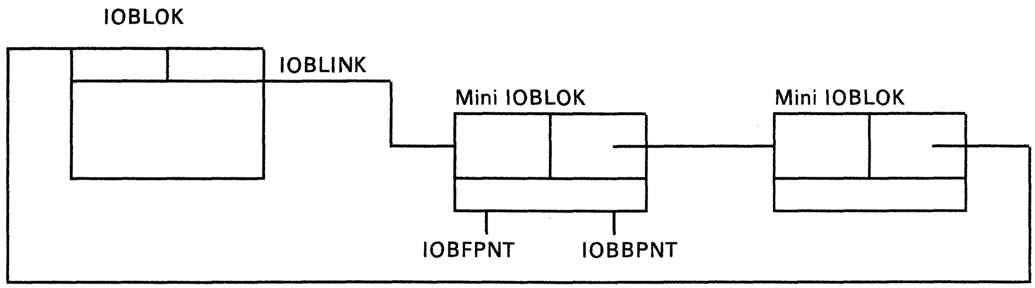


Figure 27. Mini IOBLOK Queuing

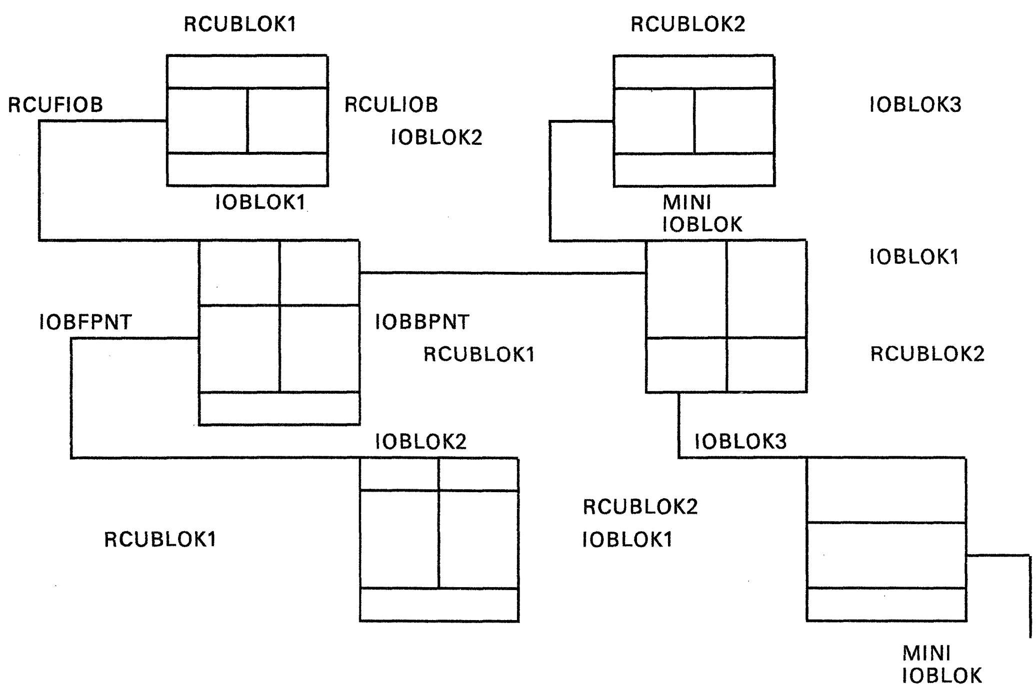


Figure 28. Control Block Structure for Alternate Path Request

Prior to starting an I/O operation associated with the request, a check is made to see if the IOBLOK is a mini IOBLOK and whether mini IOBLOKs are queued off this IOBLOK. All mini IOBLOKs associated with this request are dequeued from their respective queues by running the IOBLINK chain. The storage for the blocks is released. If the active IOBLOK is a mini IOBLOK, the IOBRADD from the mini IOBLOK is moved to the primary IOBLOK and the I/O started using the primary IOBLOK.

Reserve/Release

Reserve/release is supported for shared DASD as though each virtual machine has a separate channel path to a shared device. Reserve/release support prevents the occurrence of a channel lockout situation. This is accomplished by changing reserve CCWs to sense CCWs when a reserve is issued to a device that has alternate paths defined to it. In an MP configuration, an alternate path is defined if there is a channel path from each processor. This means that whenever alternate paths are defined to a device, the real reserve does not execute on the hardware.

Note: In an MP configuration, an alternate path is defined when there is a channel path from each processor.

When a reserve operation is changed into a sense before being executed on the hardware, the access to the device remains **unrestricted**. This can lead to an integrity exposure on the shared device. For devices being shared among virtual machines active on the same processor, reserve/release support is implemented on a virtual basis allowing the reserve/release operation codes to be simulated on a virtual basis for minidisks, including full-extent minidisks. When a reserve is issued against a minidisk, the reserve is accomplished by a locking mechanism. The status of the minidisk is maintained in the VRRBLOK that is chained from the VDEVBLOK.

The following matrix identifies how the reserve operation code is handled in the various situations.

	Defined Alternate Paths to Device[3]	Will Reserve/ Release Execute on the Hardware	Virtual Reserve/ Release Requested for Minidisks	RESERVE[1] —or— SENSE[2]
Dedicated	NO	N/A	N/A	RESERVE
DASD or Tape	YES	N/A	N/A	SENSE
Minidisk	NO	NO	NO	RESERVE
	NO	NO	YES	SENSE
	NO	YES	NO	RESERVE
	NO	YES	YES	RESERVE
	YES	N/A	N/A	SENSE

[1]The 'RESERVE' keyword in the chart indicates that the real reserve is allowed to execute on the hardware.
[2]The SENSE keyword indicates that the reserve CCW is changed to a sense CCW. Virtual Reserve/Release is requested by means of a new option on the MDISK directory control statement.
[3]In an MP system, an alternate path is defined when each processor has a channel path to the device.

DMKVIO

DMKVIO performs the following steps when virtual reserve/release processing is requested:

1. DMKVSI calls DMKCCW to perform CCW translation. For DASD devices, DMKCCW checks if the virtual reserve/release feature bit is on in the VDEVBLOK. If virtual reserve/release processing has been requested and if the device is not reserved by anyone or it is reserved by this user, processing continues normally. If the device is reserved by another user, DMKCCW calls DMKUNTFR to restore the CCWs to their original state and returns to the caller, unless sense bytes have been transferred to the user's storage in which case CP enqueues on the minidisk and waits until it is no longer reserved at which time the I/O can proceed. If the I/O request can continue and the CCW chain contains a reserve command, the VDEVBLOK and the VRRBLOK are flagged as reserved. If the CCW chain also contains a release, the IOBLOK is flagged to indicate to DMKUNTFR to release the virtual disk. Control returns to DMKVSI.
2. DMKVSI reflects a device-busy condition to the virtual machine if the minidisk is currently reserved by another user.
3. DMKUNT reflects a device end interrupt to all virtual machine users who previously received a busy condition, when the device is released.

DMKRIO

The DMKRIO module implicitly defines alternate channels for all control units in a multiprocessor environment. Reserve operations are changed into sense operations for shared DASDs, and to ensure proper integrity of data in this case, following are some recommendations for several situations:

1. Two or more virtual machines share DASD devices on a single processor.

   Use virtual reserve/release and define minidisks with the access mode "V" for all shared volumes. All virtual machines willing to share these volumes should have a link established to them.
2. One virtual machine sharing DASD devices with one or more real processors.

   Only one path to the shared DASD devices should be on-line on the VM/SP HPO side. Dedicate the shared devices to the virtual machine. Or define the shared volumes as *full pack* minidisks, making sure that every real unit on which a shared volume is mounted has the Reserve/Release Feature available.
3. Two or more virtual machines sharing DASD between themselves and one or more real processors.

This is a combination of the two situations described above.

 LY20-0897-7

Ensure that only one path to the shared DASD devices is on-line on the VM/SP HPO side. Define all shared volumes as *full pack* minidisks for VM/SP HPO with access mode "V." All virtual machines willing to share those volumes should have a link established to them. Ensure that every real unit on which a shared volume is mounted has the Reserve/Release Feature available.

Ordered Seek Queuing: Requests to start I/O on system devices are normally handled first in first out. However, requests to movable-head DASD devices are queued on the device in ascending order by seek address. This ordered seek queuing is performed to minimize intercylinder seek times and to improve the overall throughput of the I/O system.

CP assumes that very few virtual machines perform chained SEEKs. Therefore, the first logical address represents the position of the arm upon completion of the I/O operation. Ordered SEEK queuing is based on the relocated real cylinder. DMKIOQ, which handles IOBLOK queuing, uses the cylinder location supplied in IOBCYL for ordered SEEK queuing. This field is initialized by the calling CP routine for paging and spooling or by the CCW translator for virtual I/O. The CCW translator, DMKCCW, supplies the IOBCYL value in the following manner:

- Reads the IPL record, relocates to virtual cylinder 0
- Recalibrates, issues a real calibrate, and then a SEEK to virtual cylinder 0
- Issues a channel SEEK, relocates to the virtual cylinder
- For FBA devices, converts the block number being located to the corresponding cylinder number.

The IOBLOK queuing subroutine of DMKIOQ recognizes that a request is being queued on a movable-head DASD by means of the device class and type fields of RDEVBLOK. Instead of adding the IOBLOK to the end of the queue on the RDEVBLOK, the queuing routine sorts the block into the queue based on the cylinder number for the request. The cylinder number for any request to DASD is recorded in the IOBCYL field. The queue of IOBLOKs on a real device block is sorted in ascending order by SEEK address, unless the entire device is dedicated to a given user. In this case, DMKIOQ does not automatically schedule the device, and no more than one request can be outstanding at any one time.

When an outstanding I/O request for a device has completed, DMKIOS attempts to restart the device by dequeuing and starting the next IOBLOK queued on the device. For non-DASD, this is the first IOBLOK queued. Fixed-head DASD paging requests are dequeued ahead of moveable-head requests. For movable-head DASD, the queued requests are dequeued in either ascending or descending order, depending upon the current position (recorded in RDEVCYL) and the direction of motion of the arm. If the arm is seeking up (that is, toward the higher cylinder numbers), the queue of IOBLOKs is scanned from the first block toward the last until an IOBLOK is found with an IOBCYL value equal to or greater than the value in RDEVCYL, or until the end of the queue is reached. At this point, the device is flagged as seeking down and the queue is scanned from last to first until an IOBLOK with an IOBCYL value equal to or less than RDEVCYL is found. When the IOBLOK is found, it is dequeued and started. The direction of motion is indicated by an RDEVFLAG bit and the next request is dequeued downward until the head of the queue is reached.

Because the queue itself is a two-way chained list, no special handling for null or unity set lists is required, and the ordered seek algorithm returns to first-in-first-out queuing.

Dedicated Channel Support: One of the facilities of the control program allows a virtual machine to control one or more channels on a dedicated basis. The channels are attached to the virtual machine by using the privileged ATTACH CHANNEL command. A virtual machine can have one or more dedicated channels. In addition, channels can be split between virtual machines but a dedicated channel cannot be shared between two virtual machines. For instance, channel 1 could be dedicated to virtual machine A, and channel 2 could be dedicated to virtual machine B, or both could be dedicated to virtual machine A or B.

With a dedicated channel, all virtual machine device addresses must be identical to the real machine device addresses. For instance, virtual device 130 must be real device 130, and virtual device 132 must be real device 132. With dedicated channels, CP does not perform any virtual device address mapping.

CP error recording and channel recovery procedures are still in effect for dedicated channels. The dedicated channel support can be used in conjunction with the virtual=real feature for any virtual machine that is occupying the virtual=real storage space.

Virtual Console Simulation

DMKVCN receives control from the virtual machine I/O executive, DMKVSI. When control is received, the device is available with no interruptions pending. A console control block, VCONCTL, that is obtained from storage and chained from the virtual device control block, VDEVBLOK, by DMKLOG is accessed for use during the interpretation of the virtual console I/O sequence. The user's CAW is examined for validity. If it is valid, the TRANS macro is issued to fetch the first user CCW. This CCW is moved to the VCONCTL block for analysis.

The CCW is analyzed to determine if it is a read, a write, a control, a sense, a TIC, or an invalid operation. Based upon the analysis, the appropriate processing routine in DMKVCN is invoked.

The Read Simulation Routine: Obtains a buffer for input data from free storage. The location of the buffer is set in the VCONCTL block. The DMKQCNRD routine is called to schedule and perform an actual read to the corresponding real device representing the user's virtual console. If SET LINEDIT ON is specified, the buffer data is edited and translated to EBCDIC. When the read is completed, the data is moved to the specified user address obtained from the address portion of the virtual CCW. If command chaining is specified, processing returns to fetch and analyze the next CCW. If command chaining is not specified, the virtual CSW is constructed in the VDEVBLOK and an interrupt is flagged as pending in the VMBLOK.

The Write Simulation Routine: Obtains a buffer for the construction of the output message from free storage. The virtual machine data is located from the virtual CCW address in the VCONCTL block and moved to the data buffer. The DMKQCNWT routine is called to write the data in the buffer and provide the necessary length, translation, and format functions. Control is received at the DMKVCN module upon completion of the writing. At this point, the virtual CCW is re-examined. If command chaining is specified, processing continues to fetch and analyze the next CCW. If command chaining is not specified, the virtual CSW is constructed in the VDEVBLOK and an interruption is flagged as pending in the VMBLOK.

The Control Simulation Routine: Is used for the NOP and ALARM operations. A NOP operation requires no data transfer or I/O operation. An ALARM operation has no equivalent on low-speed teleprocessing equipment. Thus, a message indicating the ALARM operation is constructed. DMKQCNWT is called to output the constructed message. If the command is chained, processing continues (for NOP or ALARM) to fetch the next CCW and analyze it. If command chaining is not specified and this is not the first CCW, a virtual CSW is constructed in the VDEVBLOK and an interruption is flagged as pending in the VMBLOK. If this is the first (and only) CCW, then a condition code of 1 is presented with channel end and device end in the virtual CSW.

A Virtual Sense Operation: Is similar to a control operation, because no actual I/O operation is performed. However, there is data transfer. The sense data from the VDEVBLOK is moved to the virtual storage location specified in the virtual CCW address. If the command is chained, processing continues to fetch the next CCW and analyze it. Otherwise, an interruption is flagged as pending in the VMBLOK.

A Virtual TIC Operation: Fetches the virtual CCW addressed by the TIC address and analyzes the fetched CCW. If the fetched CCW is itself a TIC, or if the TIC is the first CCW, a channel program check condition is reflected to the virtual machine as an interruption or as a CSW-stored condition, respectively.

Invalid Operation: Any other operation is considered invalid. Command reject status is posted in the virtual sense byte and the operation is terminated with unit check status presented in the virtual CSW.

Remote 3270 Programming

For a basic understanding of CP processing of data relating to 3270 devices on binary synchronous lines, the information and terminology contained in *IBM 3270 Information Display System Component Description,* and *General Information - Binary Synchronous Communications* is required.

A digest of some of this essential information as it applies to VM/SP HPO follows:

- Text messages to and from remote terminals and printers can only be achieved when the bisync line is in text mode.
- Text messages from a remote device can be the result of a general poll or specific poll operation to the related device or devices on the bisync line. This polling communication interface is accomplished by each line-connected control unit having unique specific poll and general poll recognition circuitry and by the CP terminal list of valid bisync lines and 3270 remote control unit addresses. This list, the terminal list, is generated by system generation procedures employing TERMINAL and CLUSTER macros. For more details about terminal list generation, see the *VM/SP HPO Planning Guide and Reference.*
- Reliability and dependability of line operation is achieved by the use of: a double addressing scheme, control characters with a rigid message protocol, and complex redundancy-check characters appended to transmission messages. Examples of these techniques are shown in the formats that follow.
- Every message (text or control) that is issued by CP may or may not be responded to by the remote station or control unit. The type of response (or absence of response) that CP receives depends on the receptiveness of that device or control unit to the previously sent message (is the device ready and enabled and accurately addressed) and the content and correctness of the message (no line errors).
- To establish the relationship of the line of terminal response to a particular line or device write or read operation, CP employs an operation "tracking" facility (TP op code) imbedded in the issued CCWs. The function performed by the CP op code is described in the following CCW formats.

Format of the 3270 Remote CCW

Operation Code 1 byte	Address Field 3 bytes	Flags 1 byte	TP Op Code 1 byte	Count 2 bytes
0 7	8 31	32 39	40 47	48 63

where:

Operation Code
contains the hexadecimal value of the type of operation performed by the command.

Valid operation codes are:

X'01'	WRITE
X'02'	READ
X'03'	NO-OP
X'09'	POLL
X'23'	SET MODE
X'27'	ENABLE
X'2F'	DISABLE

Address Field
Depending on CCW usage, this field may address:

Data
Output data contained in the CONTASK.

Area
The address of the data area (read buffer) located in the BSCBLOK at BSCREAD.

Control Characters
A data-link control character such as EOT or ENQ that is defined in DMKRGA or DMKRGB.

Response
(BSCRESP). The address location of the response buffer in the BSCBLOK.

Addressing Characters
Data characters that indicate which device is to send or receive data. The entry for WRITE operation is at location BSCSEL. The entry for the READ operation is at location BSCPOLL.

Note: To see how the key words DATA, Control Characters, Response and addressing Characters are used, refer to the CCW sequences described in "I/O Program Routines for BSC Lines and 3270 Remote Devices" in this section.

Flags
The flag bits turned on in the CCW: CC (channel commands), CD (chained data), SILI (suppress incorrect length indication), skip (suppress data transfer to main storage).

TP Op Code
An imbedded teleprocessing operation code in the CCWs used in bisync line communications. This code is inspected by the secondary interruption handler, DMKRGAIN, when channel end and device end are received. The code is also used by the error processing module, DMKBSC. The code indicates the function being performed by the associated command. For use of the TP op codes, refer to the formatted CCWs that follow.

Count
Refers to the byte length of the CCW READ or WRITE operation.

I/O Programs for Bisynchronous Lines and Remote 3270s

Before data communication to remote 3270 equipment can take place, the remote teleprocessing line, the control unit and the device(s) must be enabled for communication. This occurs when a special sequence of channel commands is executed. Disabling a line occurs in a similar manner. The following is the format of the CCWs used in the enabling/disabling operation.

Enable a Line

Operation	Command Code	Address	Flags	TP Op Code	Count
Disable Line	X'2F'	0	CC, SILI	01	1
Set Mode	X'23'	Control Character	CC, SILI	01	1
Enable Line	X'27'	0	SILI	01	1

Disable a Line

Operation	Command Code	Address	Flags	TP Op Code	Count
Disable Line	X'2F'	0	SILI	01	1

After a line is enabled, communication can then be directed to a particular resource. The sequence of events is as follows:

Send a data link control character on the line that places the control unit in control mode. This mode makes the control unit receptive to the specific address indicated by the second CCW. The third CCW is a read CCW that is needed for the acknowledgement response from the addressed control unit. Normally, in response, CP transmits a block of data to that device with a write text CCW. Acknowledgement of receipt of this data is contained by the read response (text) CCW. The format of the CCW select and write text operation follows.

Write Select

Operation	Command Code	Address	Flags	TP Op Code	Count
Write an EOT	01	Control Character	CC, SILI	02	1
Write addressing char.	01	List Address Character	CC, SILI	03	5
Read Response	02	Response	SILI	05	2

When the control unit recognizes the addressing characters, it transmits an acknowledgement and enters text mode. CP then writes one or more screen lines in a Write Text operation. Multiple lines are written if more than one CONTASK is queued or if the screen must be updated.

Write Text (One Line)

Operation	Command Code	Address	Flags	TP Op Code	Count
Write text	01	Data	CD, SILI	03	vari-able
Write ETX	01	Control Character	CC, SILI	03	1
Read Response	02	Response	SILI	11	2

Write Text (Multiple Lines)

	Operation	Command Code	Address	Flags	TP Op Code	Count
(1)	Write text	01	Data	CD, SILI	03	vari-able
	TIC	08	To next write*	CD, SILI	03	1

* Format (1) or (2)

(2)	Write text	01	Data	CD, SILI	03	vari-able
	Write ETX	01	Control Character	CC, SILI	03	1
	Read response	02	Response	SILI	11	2

In situations where the line is found to be in text mode, CP can issue a write reset sequence to put the binary synchronous line in control mode. The following format illustrates the write reset CCW.

Write Reset

Operation	Command Code	Address	Flags	TP Op Code	Count
Write EOT	01	Control Character	SILI	09	1

In situations where the expected response from a remote station was not received or was invalid, the channel program may request the remote station to retransmit the response. The following write ENQ format shows this sequence. The remote station, upon receipt of the ENQ message, responds by transmitting the expected or valid response to the response area indicated by the original read response CCW.

Write ENQ

Operation	Command Code	Address	Flags	TP Op Code	Count
Write ENQ	01		CC, SILI	06	1
TIC	02	Original Read Response CCW	SILI	0	1

Read operations occur following a general poll or a specific poll for text messages. In a general poll sequence, CP transmits the general poll characters to the attached control unit on the bisync line. The control unit recognizes the polling request, and transmits any pending data, causing the second (NOP) CCW to be skipped. The last CCW provides the read buffer and the count necessary for the incoming data block from the first remote station on the list that had a message queued for transmission. If, however, there is no pending data, then the channel program ends with the third CCW. The following read initial format shows the initial read CCW sequence.

Read Initial

Operation	Command Code	Address	Flags	TP Op Code	Count
Write EOT	01	Control Character	CC, SILI	02	1
Poll	09	Control Character	CC, SILI	03	7
I/O No operation	03	0	SILI	07	1
Read Text	02	Area	SILI	10	263

After CP receives a message from a remote station, it transmits any outbound data, then repeats the general poll. When a poll completes with no inbound data, CP starts the poll delay, then allows the poll delay interval to expire before starting another poll to the line (assuming CP has no higher line priority tasks to process). If, in the process of receiving messages from remote stations, CP receives a message block that is invalid or its beginning or ending bisync control characters are not recognized, CP can elect to send a negative response back to the remote station. This negative response, the NAK control character, causes the remote station to retransmit the previous message to CP. This incoming message is processed by the second CCW of the read repeat sequence as shown in the format below.

Read Repeat

Operation	Command Code	Address	Flags	TP Op Code	Count
Write NAK	01	Control Character	CC, SILI	06	1
TIC	08	Original Read Text CCW	SILI	00	1

Once an error-free message is received from the remote station, CP proceeds with one of two actions:

1. If there is data to be transmitted, CP sends it immediately, using the Write Text format. This is known as binary synchronous "limited conversational mode."
2. If there is no data to be transmitted to the remote device that generated the message, CP sends an RVI control character to suspend the transmission of additional data. The remote 3270 responds to the RVI with EOT instead of sending another message.

Read Interruption

Operation	Command Code	Address	Flags	TP Op Code	Count
Write RVI	01	Control Character	CC, ;SILI	06	2
Read Response	02	Response	SILI	11	2

Data Formats – Remote 3270s

CP, in conjunction with remote 3270 support, uses the following formats for its text messages. For a detailed explanation of the abbreviations used, see the *IBM 3270 Information Display System Component Description*.

Write Text Data Message Format

Display commands use this message format for the placement or erasure of data anywhere on the display screen. The display commands that implement this function are: WRITE (X'F1'), ERASE/WRITE (X'F5'), ERASE WRITE ALTERNATE (X'X7E'), and COPY (X'F7').

Write Data Stream

STX	ESC	CMD	WCC	SBA	Buffer Address	Orders & Text	SBA	Buffer Address		ETX
1	1	1	1	1	2	variable	1	2		1

Write Structured Field Data Stream

CP uses the Write Structured Field data stream when the display is enabled by the system operator using the READ PARTITION (QUERY) function. CP determines the device features of 3278 and 3279 display stations. This data stream is not used if the display station type is 3277.

DLE	STX	ESC	WSF CMD	length	RD. PART.	X'FF'	QUERY
1	1	1	1	2	1	1	1

If the remote control unit supports the 3270 extended data stream, then CP transmits all text messages in binary synchronous *transparent text* mode. Transparent text mode is used so that all 256 possible character codes may be transmitted. These character codes are required for certain 3270 extended functions such as color and extended highlight features.

The format of the Write Data Stream in transparent text mode is similar to the normal format except that the leading STX and trailing ETX are modified:

DLE	STX	ESC	CMD	WCC	SBA	Buffer Address	Orders & Text	SBA	Buffer Address	DLE	ETX
1	1	1	1	1	1	2	variable	1	2	1	1

The DLE-STX header indicates the start of a transparent text block and the trailing DLE-ETX indicates the end of the block. The DLE-ETX is written by a special *command-chained* CCW to differentiate it from the data contained in the text.

The transmission control unit (TCU) hardware provides special support for transparent text mode. The TCU checks each text character and inserts a DLE character before any DLE text character, thus turning a single DLE into a double DLE. These DLEs are automatically removed by the remote control unit. See *General Information-Binary Synchronous Communications* for details.

Write Text Messages for the Copy Command

The COPY command is limited to compatible printers located on the same control unit. Action starts by pressing a PF key designated for the COPY function. CP responds by sending a message to the control unit that contains both the designated printer and the displays station that requested the action and directs the control unit to print the designated display buffer to the printer specified.

The format of the COPY messages follows.

3271 Copy Data Stream

STX	ESC	CMD X'F7'	CCC	From Address	ETX

3275 Copy Data Stream

STX	ESC	CMD X'F1'	WCC	SBA Adr	Buff (4040)	IC	ETX

Read Text and Read Header Message Formats

The following is representative of input message formats. The format of a CP-generated read operation follows.

Read Text Data Stream

Index Byte	STX	CU Adr	Dev Adr	AID	Cursor Addr	SBA	Buff Addr	Text	SBA	Buff Addr	Text		ETX

The Read Text Data Stream is in the *transparent text* format if there are binary synchronous control codes in the data.

Index Byte	DLE	STX	CU Adr	Dev Adr	AID	Cursor Addr	SBA	Buff Adr	Text	SBA	Buff Addr	Text	DLE	ETX

Error Status Data Stream

Another form of input message is the error status message. Error status is processed by the DMKRGA module. The characters, %R, following the SOH signify that this message contains sense and status data. The format of this message follows.

Index Byte	SOH	%	R	STX	CU ADR	Dev Adr	Sense/ Status Bytes	ETX

Test Request Data Stream

The test request message, upon receipt from display terminals, is ignored by CP. The input inhibit mode that the display terminal enters upon pressing the test request key can be reset only if the terminal user presses the RESET key. The characters, %/, following SOH indicate the test request function. The format of this message follows.

Index Byte	SOH	%	/	STX	Text	ETX

Allocation Management

Real storage space above the nucleus and below the 16 Mb line is made up of the dynamic paging area, the free storage area, and the trace table area. The dynamic paging area consists of page frames that are allocated to virtual machines and CP to satisfy paging requests. The free storage area consists of page frames that are allocated to virtual machines and CP for working storage. The trace table area consists of page frames into which trace information is placed.

Real storage space above the 16 Mb line consists of a dynamic paging area for virtual machines only, and can include a V=R area for a PMA guest, depending on the way the system programmer generated the system. The page frames in the greater than 16 Mb dynamic paging area are used to satisfy virtual machine page faults, but are not used for pages that CP wants to directly address. CP must move a page referenced in this area to the less than 16 Mb dynamic paging area before CP can use the page.

The size of the free storage area, the trace table area, the two dynamic paging areas, and the V=R area for the PMA guest are defined by the system programmer at system generation time.

Normal Paging Requests

If a program interruption is caused by a normal paging request (not from a virtual machine that is running in EC mode with translation on), DMKPRGIN determines whether a segment or page translation error has occurred. If one of these errors occurred, an invalid address interruption code is set, and the interruption is reflected to the virtual machine supervisor. If a segment or page translation error has not occurred, the virtual machine's current PSW is updated from the program old PSW (PROPSW), the address of the current VMBLOK is placed in register 11, and DMKPTRAN is called to obtain the required page. When the paging operation is completed, control is returned to DMKDSPCH.

Virtual Storage Management

When operating in the CP relocate environment, each virtual machine's virtual storage space is described by two sets of tables.

- One set, the segment and page tables, describes the location and availability of any of the virtual machine's virtual pages that may be resident in real storage. Locations in these tables are indexable by virtual address, and the entries contain index values that reference corresponding real storage addresses. In addition, each table entry contains an indication of whether the corresponding virtual page is available to the user in real storage. These tables are referenced directly by the DAT feature when the virtual machine's program is running.

- The second set of tables, called swap tables, is a map of the locations of the virtual machine's pages on the DASD devices that comprise the system's paging or auxiliary storage. The DASD addresses in these tables can either represent where the page is on DASD or zero, indicating that the given page has not yet been referenced and, thus, has a value of binary zeros.

The swap tables are arranged in a format indexable by virtual storage address. In addition to containing the address of a page, each entry contains flags and status bytes that indicate such information as:

- The storage protection keys to be assigned to the page when it is made resident.

- Whether the page is currently on its way into or out of the system (in transit), etc.

These tables are not referenced directly by the hardware as are the page and segment tables, but are used by paging management to locate user pages that are needed to execute a program.

A demand for a page on DASD can be made implicitly by a virtual machine or explicitly by CP.

- An implicit demand for a page is made when a program attempts to reference a page that is not available in real main storage. This attempt causes a program interruption with the interruption code indicating a page or segment exception. Upon recognition of this condition, control is passed to the paging manager to obtain a page frame of real main storage and to bring in the desired page.
- An explicit demand for a page can be made by CP (for example, in the course of translating a user's channel program). If, in the process of translation, CP encounters a CCW that addresses a page that is not resident in real storage, a call is made to the paging manager to make the referenced page resident.

While the requested page is being fetched, the requesting virtual machine is unable to continue execution. It may be possible, however, to run other tasks in the system, and CP runs these while the needed page is being paged in. When the requested page is resident, the virtual machine can be run and is dispatched in its turn.

In addition to demanding pages, virtual machines implicitly or explicitly release page frames of their virtual storage space. Part of the space may be explicitly released from both real and virtual storage via a DIAGNOSE instruction that indicates to the control program those page frames that are to be released. An entire virtual storage is released when a user loads (via IPL) a new operating system or logs off from the system.

CP also has virtual storage associated with it. This space contains CP (some parts of which need not always be resident in real storage), and virtual storage buffers for spooling and system directory operations. Although CP makes use of virtual storage space for its execution, it does not run in relocate mode. Thus, nonresident modules must be completely relocatable.

To improve performance by eliminating unnecessary LCTL and LRA instructions, CP keeps track of where each virtual machine's page zero resides. CP does this by checking an in-storage pointer in the VMBLOK—the pointer contains the address of the virtual machine's page zero if the page is resident. If it is not resident, CP issues a TRANS macro, which checks for page residency and demands a page-in if the page is not in real storage. If the page is resident, CP bypasses issuing the TRANS macro.

Virtual Buffer Manager

CP can dynamically increase the amount of virtual storage it can use by using a virtual buffer manager. This virtual buffer manager also allows CP to allocate and deallocate this virtual storage.

The virtual buffer manager maintains a chain of nondispatchable VMBLOKs, similar to the SYSTEM VMBLOK. CP modules can invoke the virtual buffer manager to create these VMBLOKs dynamically. These VMBLOKs, which have userid defined by the calling CP module, are maintained on a chain that is anchored off the SYSTEM VMBLOK and not associated with the user VMBLOK chain.

As with the SYSTEM VMBLOK, each of these VMBLOKs is associated with segment, page, and swap tables for virtual storage. CP maintains an allocation table so it can control the allocation and deallocation of the virtual storage associated with the VMBLOK. The system function using the virtual buffer manager must be sensitive to the VMBLOK associated with the virtual storage it uses. The paging operation must be done using the specific VMBLOK associated with the vitrual storage obtained to get access to the correct segment, page, and swap tables needed to access that storage.

Real Storage Management

Real storage management allocates the system's page frames of real storage to satisfy the demands for virtual pages made by the system's virtual machines. Efficiency of allocation involves a trade-off—the paging manager uses only enough processor time to ensure that:

- The set of virtual storage pages that are resident represent those pages that are most likely to be used.
- A sufficient number of cycles is available to execute virtual machine programs.

Inefficiency in the first area causes a condition known as thrashing, which means that frequently used pages are not allowed to remain resident long enough for useful work to be performed by or on them. Thrashing could be aggravated by the paging manager's page frame selection algorithm or by a scheduler that attempts to run more tasks than the system can handle (the sum of their storage requirements exceeds the real paging space available in the system). Thus, the paging manager must keep statistics on system and virtual machine paging activity and make these statistics available to the scheduler to detect and prevent a potential thrashing condition.

Inefficiency in the second area causes an unacceptable ratio of CP overhead to virtual machine program time, and in extreme cases may cause CP to use excessive processor time. To understand how allocation is determined by CP, the way in which the inventory of real storage page frames is described to the system must be understood.

Each page frame (4096-byte block) of real storage in the system is in one of two basic states: nonpageable or pageable. A nonpageable page must remain resident in real storage for some period of time. Thus, the page frame cannot be taken from its current owner to be given to someone else. Pages can be either permanently or temporarily nonpageable, depending upon their use.

Temporary locks usually occur when an I/O operation has been initiated that is moving data either to or from the page, and the page must be kept in real storage until the operation has completed.

A page can also be temporarily nonpageable if it contains an active nonresident CP routine.

In addition, a page can be nonpageable through use of the LOCK command. Pages locked this way are permanently resident until they are explicitly unlocked by the UNLOCK command. Pages that are usually considered permanently nonpageable are those that contain the resident portion of CP and those that contain the system's free storage area in which control blocks, I/O buffers, and the like, are built.

The data area that page management routines use to control and allocate real storage is the CORTABLE. Each page frame of real storage has a corresponding entry in the CORTABLE, and because the table entries are fixed in length and contiguous, the entry for any given real page frame may be located directly by indexing into the table. Each entry contains pointers that indicate both the status and ownership of the real page that it represents. Some pointers link page table and swap table entries to the real page (and thus establish ownership), while others link the entry into one of several lists that real storage management uses to indicate the page frame's status and availability for paging. A given CORTABLE entry may appear on one of three lists. The lists are known as the free list (FREELIST), the flush list (FLUSHLST), and the swap list and they represent various levels of page frame availability.

The free list contains page frames that are immediately available for assignment to a requesting virtual machine. The virtual storage pages for which they were last used have either been released by their owners or they have been paged out to auxiliary storage. Requests for real storage are always satisfied from the free list. If the list has been depleted, the requestor waits until a new page frame becomes available as the result of a virtual storage release or a page out.

The flush list gains pages when special conditions occur during physical and logical swapping. See "Flush List Management" on page 145 for more information. The flush list is one of the places that the page frame selection routine looks to find a page frame to page out or to assign to the free list for a virtual machine that requires real storage space.

A swap list contains pages that have been logically but not physically swapped out. There is one swap list for each user. Swap lists are another place the page selection routine looks to find pages that can be written to DASD.

Requests for Real Storage Page Frames

Requests for real storage fall into two general categories: those that are requesting space for virtual storage, and those (such as requests for CP work space) that need page frames for their own use. The former, more general case is discussed first, because the latter case is a subset of the first.

The main page manager routine, DMKPTRAN, maps a request for a specific virtual storage address into a page frame of real storage. This requires that the virtual page be read in and the necessary tables be updated to show the proper status of the page frame.

DMKPTRAN requires that the caller supply the virtual address to be translated, the VMBLOK address, and any options that apply to the page to be located. Most calls are made via the TRANS macro, which sets up the necessary parameters, determines whether or not the required page is resident, and calls DMKPTRAN if it is not.

When DMKPTRAN receives control, it uses the LRA instruction to see if the requested page is marked valid. If the page is valid, the routine locks the page if requested and exits to the caller. If the page is nonvalid, it may still be in real storage. Nonvalidity means that:

- The page table or segment table entry associated with the page has been flagged as invalid or
- The page table or segment table length has been exceeded.

Note: Pages above the 16 Mb line are exceptions. They are considered nonresident by CP although an LRA on them returns a condition code of zero.

If this is not the first reference to the page, it may be in transit or exist in one of the following areas:

- The less than 16 Mb free list
- The greater than 16 Mb free list
- The flush list
- The swap list
- A paging area
- A swap area.

Note: Swapping, in general, is the paging in or out of groups of pages. Logical swapping is the creation of lists of pages that can be written to DASD. Physical swapping provides for better DASD use by writing and reading pages in groups. Real storage management initiates swapping, but the I/O operations are done by the auxiliary storage manager.

When the referenced page is on a swap list, it means that the page has been logically but not physically swapped out. The routine RECLWSPG in DMKPTR reclaims the page without an I/O operation by removing it from the swap list and updating its page table entry, core table entry, and swap table entry.

When the referenced page is in a swap set that has been written to a swap area, the routine SWAPFLT swaps in the entire swap set. This situation is indicated by a swap fault, which is an address translation exception that occurs when CP or a virtual machine references a page that has been physically swapped out. Pages referenced by CP are always swapped in below the 16 Mb line. Other pages in the swap set, except page zero, can be swapped in anywhere.

If a page exists on the free list or on the flush list, DMKPTR reclaims it by the following process: If the LRA indicates that the page is invalid, it is still possible that the required page is resident. This occurs if the page frame has been placed on the FREELIST but has not yet been assigned to another virtual machine. When the page out routine removes a page frame from a virtual machine, the invalid bit is set in the corresponding page table entry. However, the real main storage index for the page frame is left unchanged. The page table entry is set to zero only when the corresponding page is actually assigned to another virtual machine. Thus, if DMKPTRAN finds the page to be invalid, a further test is made on the page table entry to see if the page can be reclaimed. If the entry is not zero (aside from the unavailable bit), the CORTABLE entry for the page frame is removed from the free or flush list and the page frame is returned to the calling virtual machine.

If the page table entry corresponding to the requested virtual page is zero, the required page is not in real storage and must be paged in. However, it is possible that the page is already on its way into main storage (intransit). This condition is indicated by a flag in the SWPTABLE entry for the virtual page. The DMKPAGIO routine maintains a queue of CPEXBLOKs to be dispatched when the pending page I/O is complete. The CPEXBLOK for the page in transit is located and a new CPEXBLOK, representing the current request, is chained to it.

Before exiting to wait for the paging operation to complete, DMKPTRAN checks to see if the deferred return (DEFER option) has been specified. If it has not, DMKPTRAN returns to the caller. If the DEFER option has been requested, DMKPTRAN exits to the dispatcher to wait for page I/O completion. When the requested page has been read into real storage, the list of CPEXBLOKs are unstacked first in first out to satisfy all requests for the page that arrived while it was in transit.

If a page is not in transit, a page frame of real storage must be allocated to fill the request. Before the allocation routine is called, a test is made to see if the caller wishes the return to his routine or to be delayed until after the requested page is available. If the DEFER option is not requested, DMKPTRAN returns to the caller after first building and stacking a CPEXBLOK that allows processing of the page request to be continued the next time the dispatcher (DMKDSPCH) is entered.

DMKPTRAN next calls the FREELIST manager (DMKPTRFR) to obtain the address of the next available CORTABLE entry. DMKPTRFR maintains a list of the CORTABLE entries for those page frames that are immediately available for assignment. As DMKPTRFR releases these page frames, a check is made to see if the number of entries on the FREELIST has fallen below a dynamically maintained minimum value. If it has, DMKSELCT is called to find page frames for placement on the free list.

Once a page frame has been assigned, DMKPTRAN checks to see if a page-in is required. It usually is, and the DASD address of the virtual storage page must be obtained from the user's swap table entry and the I/O operation scheduled. However, if the page frame has not yet been referenced (as indicated by a DASD address of zero), the real main storage page frame is set to zero, and no page-in is required. After the page-in operation has been queued, DMKPTRAN exits to the paging I/O scheduler (DMKPAGIO), which initiates the paging operation and exits to the dispatcher (DMKDSPCH) to await the interruption.

Some requests for main storage page frames are handled differently from general virtual-to-real storage mapping. In particular, it may be necessary for CP to obtain additional free storage for control blocks, I/O lists, buffers, etc. This is handled by the free storage manager, which makes a direct call to DMKPTRFR to obtain the needed storage. Usually, this storage is immediately available (due to free list replenishment). However, if the FREELIST is exhausted, the request for free storage is recognized as a high-priority call and queued first on the list of those waiting for free page frames.

Free List Replenishment

The free list contains page frames to satisfy virtual machine requests for real storage. DMKSELCT replenishes the free list in such a way that enough page frames are available to satisfy a certain number of swap in operations. The threshold value for free list replenishment can be set by the class A and class E SET SRM MINNUMSS command, and is equal to:

```
(SS * M)+1+Q1+Q2
```

where:

SS is the size of a swap set.

M is the value specified by the SET SRM MINNUMSS command, or a default value of six. This value is the number of swap in operations that can be started without having to defer the request until pages become free.

Q1 is the number of Q1 virtual machines.

Q2 is the number of Q2 virtual machines.

The free list is replenished directly when users release virtual storage space. DMKPGSPO, the page release routine, calls DMKPTTFT to place released frames directly on the free list. However, the free list frequently has to be replenished in other ways.

When the free list has to be replenished, DMKSEL obtains page frames in the following order:

1. Takes pages from the flush list.
2. Performs a core table scan.

Flush List Management

Pages are placed on the flush list when any of the following conditions are met:

- An I/O error occurs during physical swap-out, and no more swap space is available.
- When a virtual machine is dropped from queue, there are no valid swappable pages, and some pages are logically swapped for that virtual machine. These logically swapped pages are moved to the flush list.
- During core table scan, the number of resident pages becomes zero and some pages are logically swapped for that virtual machine.

DMKSELCT takes pages from the top of the flush list first.

Core Table Scan

DMKSEL performs a four-pass core table scan to replenish the free list. These passes are organized into two phases with two passes in each:

- Phase 1, Pass 1—Pages that have not been physically referenced since DMKSEL last examined them are selected for free list replenishment. This excludes reserved pages, shared pages owned by another processor, and pages owned by a virtual machine locked by another processor.
- Phase 1, Pass 2—Any of the pages not selected in the first pass because they were recently referenced are now selected.
- Phase 2, Pass 1—This pass is similar to Pass 1 of Phase 1. Now, however, the exclusions in Pass 1 of Phase 1 do not apply.
- Phase 2, Pass 2—As in Pass 2 of Phase 1, all pages are now eligible.

Note: The above sequence occurs in a general *core table scan. In a* preferred *core table scan, in which a page below the 16Mb line is exclusively sought, the following distinction is made: During Phase 1 (both passes), logically swapped pages below the 16Mb line are moved* *above the line IF free pages are available there. In the case of a logically swapped page, the copy above the line remains logically swapped—the copy below the line is moved to the free list.*

Once a page has been selected for free list replenishment, DMKSEL decides whether to page out or logically swap out that page. If the page was referenced at least once while resident, it is logically swapped out. If it was never referenced while resident, it is paged out, UNLESS that page can be used to satisfy a minimum working set. If this is the case, that page is logically swapped out. See *VM/SP HPO CP for System Programming* for a discussion of the minimum working set and the SET MINWS command.

When enough pages have been logically swapped out to form a complete swap set, that set is physically swapped out. Physical swap-out will also occur if the user has no more resident swappable pages or if the core table scan is protecting reserved pages.

Physical Swap-Out

DMKSWAPO is called by DMKSEL to initiate a physical swap-out. First, DMKSWAPO creates a swap set block (SSBLOK) for the swap set. A swap set block contains the number of pages in the swap set and the Paging Storage or DASD address where the pages are to be written.

After DMKSWAPO initializes the SSBLOK, it calls DMKPGTSW to allocate Paging Storage or DASD space for the swap set. If DMKPGTSW can successfully allocate space in a swap area (area defined by SYSPAG TYPE=SW), DMKPGUPP is called to deallocate the page area slots. This prevents the same page from existing in different Paging Storage or DASD areas at the same time. The swap set is moved to Paging Storage, if possible. Otherwise, all pages in the swap set are written to the DASD swap area with one SIO.

At the end of physical swap-out, DMKSWAPO places the page frames formerly occupied by the written-out pages on the free list.

If DMKPGTSW cannot allocate space in a swap area, return is made to DMKSELCT, which writes only changed pages to Paging Storage or DASD, and writes them to a paging area instead of a swap area. The number of SIOs required for writing these pages to DASD will generally be greater than one.

Swap Fault

A swap fault is an address translation exception that occurs when CP or a virtual machine references a page which is physically swapped out to a swap area. When a swap fault occurs, DMKPTRAN physically swaps in the entire swap set in which the page resides.

When a page in a swap area is referenced by a virtual machine, the page is placed wherever page frames are available, either above or below the 16 Mb line. If the page was referenced by CP, however, or if it is virtual page zero, then it must be swapped into the less than 16 Mb area. The other pages in the swap set can be placed anywhere.

During a physical swap-in, all the pages in the swap set are flagged as in transit to prevent other I/O operations from being started if CP makes subsequent references to pages in the swap set.

When DMKPTRAN physically swaps in pages, it sets the change and recompute flags in the page and swap tables to show that these pages are to be paged out again and will require new DASD slots. Their old slots are deallocated by DMKPGUSW when the pages are physically swapped in.

For a deferred request (DEFER option of the TRANS macro specified with the call to DMKPTRAN), the virtual machine is placed in a swap wait state and does not execute until the physical swap-in is completed.

Prepaging

Prepaging is a physical swap-in that occurs when a virtual machine is added to queue. Prepaging tries to reduce the number of page faults and their subsequent delays by performing in advance some paging that may be required by the virtual machine. DMKSWAPI reads in from auxiliary storage the number of swap sets necessary to meet the prepaging requirement. The system programmer can use the SET SRM PREPAGE command to specify the number of swap sets of resident pages that DMKSWAPI should achieve via prepaging. When the SET SRM PREPAGE command has not been issued for a virtual machine, DMKSWAPI achieves a default value of two swap sets for Q1 virtual machines and does not swap in any swap sets for Q2 virtual machines.

Paging Statistics

The real storage manager, DMKPTR, accumulates paging statistics that the scheduler, DMKSCH, uses to anticipate user storage requirements. It keeps these statistics individually for each virtual machine in its VMBLOK. Cumulative statistics reflecting total paging activity for the system are kept in DMKPSA.

Among the kinds of page counts kept in each VMBLOK are the number of page-reads and page-writes for each virtual machine, and the number of times a virtual machine enters page wait when a page frame has been stolen from it. The VMBLOK also keeps a running total of the number of pages a virtual machine has resident at each page-read.

On systems running with more than 16 Mb of storage online, the VMBLOK contains two additional fullwords that reflect the number of times a page is moved from the greater than 16 Mb dynamic paging area to the less than 16 Mb dynamic paging area, and vice-versa.

Virtual=Real Option: The virtual=real option involves the mapping in a one-to-one correspondence of a virtual machine storage area with an equivalent real storage area. For instance, virtual page 1 is in real page frame 1 and virtual page 20 is in real page frame 20. Virtual page 0 is relocated at the end of the virtual storage space because it cannot occupy real page frame 0.

Note: There are times, such as when preferred machine assist is active, when the V=R guest gets absolute page 0.

The CP nucleus is altered at system generation to support the virtual=real option. Virtual machines with virtual=real (specially identified in the directory) can then logon and use the space reserved for this option. That space can be used by only one virtual machine at a time. Two virtual machines with the virtual=real capability cannot occupy the same space at the same time.

The reliability and availability of an operating system running virtual=real on a 308x, 3090, or 4381 processor can be enhanced when the TEST BLOCK instruction (TB) is used to validate the V=R storage area.

The virtual=real option allows the virtual machine to bypass the control program's CCW translation. This is possible because I/O from a virtual machine occupying a virtual=real space contains a list of CCWs whose data addresses reflect the real storage addresses. The restriction in this situation is that the virtual machine does not perform I/O into page frame 0 because this would perform a data transfer into real page frame 0. At the same time, it is assumed, and cannot be checked, that the virtual machine also does not attempt to do I/O beyond the bounds of its virtual addressing space. To do so would cause the destruction of either the CP nucleus, which resides beyond the virtual machine space, or another user's page.

Virtual 270X lines and sense operations from the virtual machine do not use the virtual=real function. These invoke CCW translation for the virtual enable/disable lines and the transfer of the sense bytes.

If the real I/O device is an MSS 3330V, then CCW translation is not bypassed since CP must still be able to recognize an MSS cylinder fault. See Appendix B for details.

The bypassing of CCW translation for the virtual machine occupying the virtual=real space is only invoked after the virtual machine has executed the SET NOTRANS ON command. This command can only be issued by the virtual machine occupying the virtual=real space. The command initiates the bypass of CCW translation. This option is automatically turned off if the virtual machine performs an explicit reset or an implied reset by performing a virtual IPL. During virtual machine IPL, I/O must be performed into page frame 0. For this reason, normal virtual IPL simulation assumes CCW translation in effect to accomplish the full simulation. Once the IPL sequence has completed, CCW translation can be bypassed by issuing the SET NOTRANS ON command.

When the virtual machine demands a page frame through normal use of CP's page tables, the paging routine recognizes the virtual=real capability. It then assigns the virtual page to the equivalent real page frame and does not perform a paging operation, because all these pages are resident and are never swapped out.

Note: The virtual machine running with virtual=real is still run in System/370 relocate mode.

Shadow table bypass, invoked by the SET STBYPASS command, allows CP to eliminate the shadow tables for an operating system running in the V=R area. When CP runs a V=R user, the shadow table for the V=R user is identical to the virtual system's own page and segment tables, with the exception of page zero. CP relocates the virtual machine's page zero (via the shadow table) to the highest real address within the V=R area. When STBYPASS is turned on, CP modifies the virtual operating system's page table to relocate virtual page zero to the highest real address. It is then possible to dispatch the virtual machine with control register 1 pointing to the virtual page and segment tables.

The UNLOCK command has a VIRT=REAL operand that essentially releases the virtual=real area for normal system paging use. Once the area has been released, it can only be reclaimed for additional virtual=real operations only by an IPL of the system. The size of the virtual=real area is an installation specification that is part of the special nucleus generation procedure that is outlined in the *VM/SP HPO Installation Guide*. The size of the area must be large enough to contain the entire addressing space of whatever V=R machine wishes to occupy that space. A V=R machine can use a smaller space than is provided but cannot use a larger space without regenerating the CP nucleus.

Extended Storage Support

Extended storage support is a software feature that allows CP to utilize greater than 16 Mb of real storage. It can be used only on processors that have the hardware for greater than 16 Mb of real storage and 4K storage keys. This support will only be active if greater than 16 Mb of storage is configured and on-line. Extended storage support improves system performance because virtual machines can use the storage above the 16 Mb line as additional dynamic paging area, freeing CP from contending as often with virtual machines for page frames below the 16 Mb line. Thus, CP has more work space available when it needs it.

Real storage above the 16 Mb line is known as the greater than 16 Mb area. The greater than 16 Mb area below the RMSIZE value specified by the system programmer is a dynamic paging area used to satisfy virtual machine page faults, but is not used for CP pageable pages, virtual page 0, or virtual machine pages that CP wants to directly address. Before CP can work with a virtual machine page resident in the greater than 16 Mb area, that page is moved to the less than 16 Mb area. CP needs to work with virtual machine pages for such functions as privileged operation simulation, channel program translation, interrupt reflection, and console functions.

Extended Real Storage Management

DMKPTR manages real storage somewhat differently when more than 16 megabytes of real storage is online.

- It manages two FREELISTs instead of one
- It uses DMKPTTPM to move pages referenced by CP to below the 16 Mb line
- It includes page frames above the 16 Mb line when it is managing CORTABLE entries in the page selection/replenishment functions.

The only modules that can use the greater than 16 Mb area are DMKPTR, DMKPTS, DMKPTT, and DMKSEL. When CP wants to work with a page that resides in the greater than 16 Mb area, DMKPTRFR allocates a page frame in the less than 16 Mb area (as if it were satisfying a page fault) and DMKPTTPM then moves the page to that frame.

Under extended storage support, DMKPTRAN is responsible for distinguishing virtual machine page fault calls from other calls. DMKPTRAN is called when an LRA instruction indicates to CP that the page CP wants to reference is not resident or is in the greater than 16 Mb area. Anytime the LRA instruction returns a condition code of 0, indicating that the page is resident, DMKPTRAN checks to see if the VFAULT option is specified in the caller's register 2.

If VFAULT is specified, DMKPTRAN knows it is being asked to satisfy a virtual page fault. DMKPTRAN sets the real translated address into the caller's register 2, makes the appropriate page table entry valid, sets a condition code of 0, and returns to the caller. If the VFAULT option is not specified when calling DMKPTRAN, then DMKPTRAN calls DMKPTRFR to allocate a page frame in the less than 16 Mb area, and either calls DMKPTTPM to move the page to below the 16 Mb line or pages it in from DASD.

Note: *If an IX/370 virtual machine page zero resides above the 16Mb line, DMKVAURN will call DMKPTRAN to bring the page below the line. (In this case, handshaking in DMKSVC will take place via the long path.)*

When DMKPTRFR is called to allocate a page frame anywhere in storage, it compares the number of pages currently available in the less than 16 Mb area with the number of pages currently available in the greater than 16 Mb area. DMKPTRFR will take a page frame from whichever area has more page frames on its FREELIST.

DMKPTTFT is called to return a page frame of real storage to a FREELIST. It checks to see if the page frame to be returned resides above or below the 16 Mb line. For a less than 16 Mb page frame, DMKPTTFT will check the less than 16 Mb deferred request queue for a page request. If there are no page requests there, DMKPTTFT will satisfy a request on the greater than 16 Mb deferred request queue, if one exists. If there are no deferred requests, DMKPTTFT will queue the free page frame off the less than 16 Mb FREELIST. If the page frame being returned resides in the greater than 16 Mb area, DMKPTTFT will satisfy requests on the greater than 16 Mb deferred request queue only, or return the free page frame to the greater than 16 Mb FREELIST.

Page Interchange Processing

DMKPTT contains the subroutines necessary to move pages between the greater than 16 Mb area and the less than 16 Mb area. DMKPTTAL is first called at IPL time by DMKSEG if extended storage support is generated by the system programmer. At that time, DMKPTTAL allocates two virtual address slots in the system address space. These slots are needed later when CP wants to reference a virtual machine page that is stored in the greater than 16 Mb area, and needs to move the page.

DMKPTTPM is the subroutine that DMKPTR calls to move a page from one page frame to another. It uses the virtual address slots allocated at IPL time to store the old and new address of the page that is being moved. DMKPTTPM then moves the page using the move long instruction (MVCL) after DAT is turned on. The old and new address of the page are the operands of the instruction.

DASD and Paging Storage Management

Any virtual machine's virtual storage pages that have been referenced but are not resident in real storage must be kept in slots on the DASD devices or Paging Storage. DASD page space is assigned only when the page is selected for a page-out. Certain DASD pages may also be marked read-only. Thus, the DASD address slot initially associated with the page should be considered to be the source of the page only. If the page is changed after it has been read into real storage, a new slot must be obtained when it is paged out. Examples of read-only pages are those which contain portions of pageable saved systems and pages which are part of a system spool file. Slots can be reassigned when DMKPTRAN finds that it must move a page out to a movable-head DASD device. In this case, the old slot is released and the new slot is obtained.

CP-owned DASD volumes are used by CP for swapping, paging, spooling, page migration, and dump spool file allocation. Paging Storage areas can be used by CP for swapping or primary paging. The system programmer uses the SYSPAG macro to define how the CP-owned DASDs and Paging Storage are to be allocated and used. The number and order of the SYSPAG macros, which are included and assembled in DMKSYS, establish an allocation hierarchy which tells CP how to use specified areas of DASD and Paging Storage.

By specifying ORDER=SYSTEM (the default) on the SYSPAG macro, a routine will scan the ALOCBLOK chain and attempt to improve upon the ordering by spreading out the ALOCBLOKs over the control units. (This will be done only if at least four ALOCBLOKs are defined.) If ORDER=USER is specified (and if at least three ALOCBLOKs are defined), the areas will be ordered to match the order in which you specified them on the SYSPAG macro.

The types of areas that can be defined on the SYSPAG macro for CP-owned volumes are:

- SW An area defined for swapping
- PP An area defined for primary paging
- PG An area defined for general paging
- PM An area defined for page migration
- PS An area defined for spooling
- DU An area defined for dump spool files

Paging Storage areas may be used only for types SW (and then only on the first TYPE=SW level) and PP.

The 3880 Model 13 or Model 23 storage subsystem should be used only for ***nonpaging*** applications. Nonpaging refers to any data on DASD that is not defined as paging, swapping, spooling, or dump area. (Actually, these subsystems may be used for paging applications, but only if the caching function is turned off via the CACHE command.) See *VM/SP HPO CP for System Programming* for more information on this subsystem. See the *VM/SP HPO Planning Guide and Reference* for complete information on the format, restrictions, and recommendations for the usage of the SYSPAG macro.

Note: You can alternately use the SYSXSTOR macro to define Paging Storage on the 3090 processor. See VM/SP HPO CP for System Programming *or the* VM/SP HPO Planning Guide and Reference *for more information on this support.*

The SYSPAG and SYSXSTOR macros generate control blocks called SYSPLIST blocks. See Figure 29 for the SYSPLIST/ALOCBLOK Generation and Relationship. At least one SYSPLIST block is built for each SYSPAG macro. CP uses these SYSPLIST control blocks as anchor points, from which it chains ALOCBLOKs during CP initialization. Each ALOCBLOK (DASD cylinder or Paging Storage allocation block) describes a group of contiguous cylinders on a CP-owned volume or a group of contiguous Paging Storage increments.

Each such group is used for the specified type of pages. Therefore, each ALOCBLOK appears on only one allocation chain, although more than one ALOCBLOK may exist for each CP-owned volume and for Paging Storage.

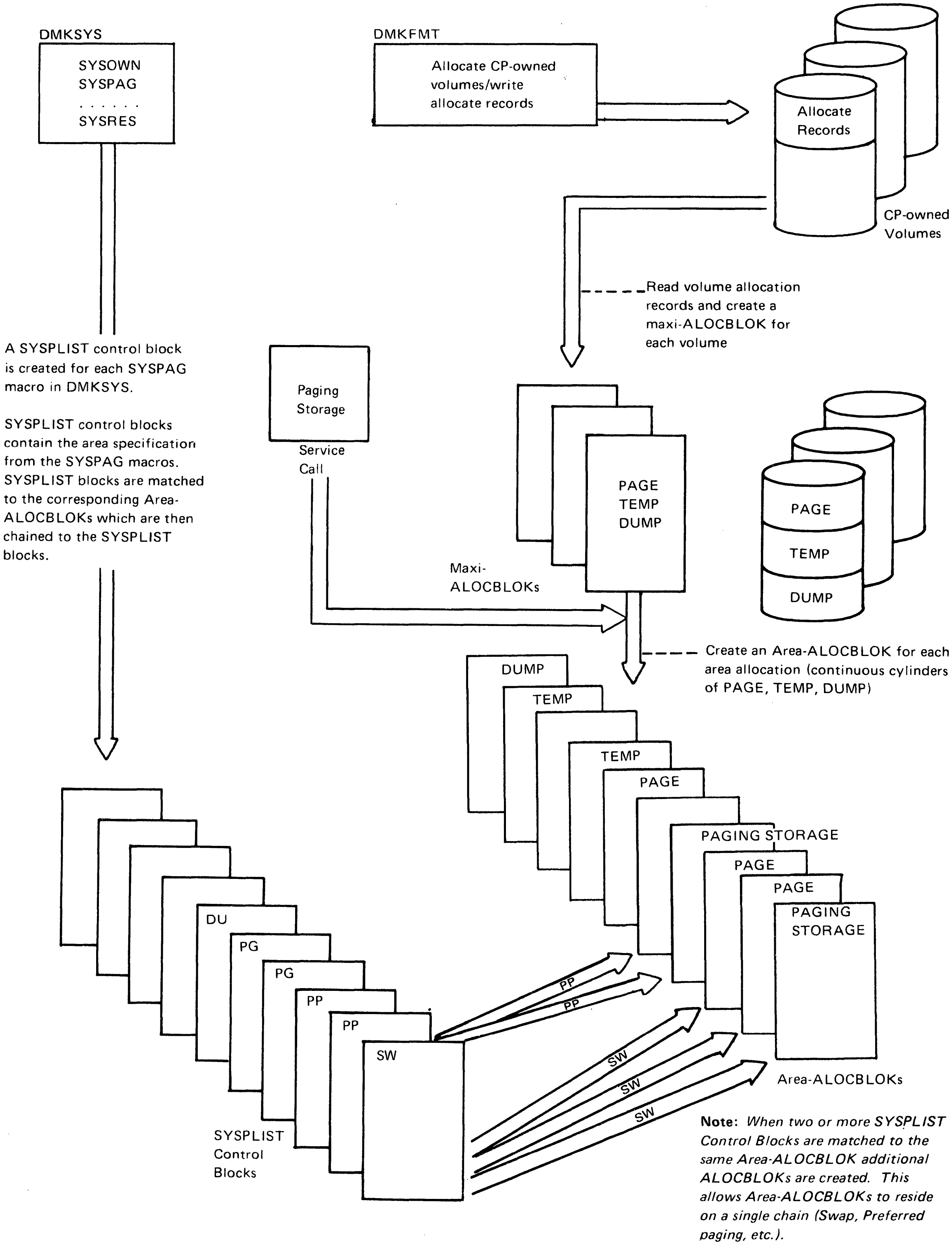


Figure 29. SYSPLIST/ALOCBLOK Generation and Relationship

Creation of ALOCBLOK Structure

DMKVDG is responsible for building ALOCBLOKs and chaining them to the appropriate anchors. DMKVDGAL is called by two modules:

- DMKIDU, when a CP-owned volume is online at initialization.
- DMKVDA, when a CP-owned volume is attached to the system.

When DMKVDGAL is called by DMKIDU, it receives the address of the maxi-ALOCBLOK for the volume. The maxi-ALOCBLOK describes all the space on the volume as specified in DMKFMT. DMKVDGAL then builds area-ALOCBLOKs, which define contiguous cylinders of the same allocation type (e.g., PAGE, TEMP). Lastly, DMKVDGAL matches the area-ALOCBLOKs to the SYSPAG macro specifications, and makes the necessary adjustments to create the final ALOCBLOKs that are chained to the appropriate SYSPLIST anchors.

Note: Before DMKVDGAL creates area-ALOCBLOKs for FBA volumes, it rounds the starting page number for each area of the volume up to the next pseudo cylinder value. Failure to match pseudo cylinder boundaries with allocation boundaries will result in the loss of use of one access position. It rounds the ending page number for each area down to the previous pseudo cylinder boundary. A ***pseudo cylinder*** *is one access position on a physical FBA device.*

When DMKVDGAL is called by DMKVDA, it must first build a maxi-ALOCBLOK before creating area-ALOCBLOKs and completing the process just described.

Area-ALOCBLOKs that are created for TDISK DASD space are not chained off a SYSPLIST anchor, but are chained by device type from anchors defined in DMKPGT.

The RECBLOKs (DASD page allocation blocks) for a given ALOCBLOK will appear on one of two chains. DMKVDGAL rechains existing RECBLOKs for recovered spool files to the appropriate area-ALOCBLOKs, in ascending order from low to high cylinder, from the field ALOCRECS. It chains RECBLOKs for PAGE and SWAP area cylinders in the same way but from the field ALOCPAGE, after it preallocates and initializes them.

Page RECBLOKs for TYPE=SW, PP, and PG are obtained as a group and are contiguous to allow quick allocation of a particular type for deallocation. The paging RECBLOKs for cylinders of these types that contain spool files will be marked "full," and will not be used for page allocation, even if the corresponding spool RECBLOKs are completely deallocated and FRETted.

Paging Storage Allocation

DMKPST builds the ALOCBLOKs for Paging Storage areas defined by the SYSPAG macro. It is called by DMKIDU during initialization, immediately after DMKIDU calls DMKVDG.

Finally, DMKIDU calls DMKVDH to order the ALOCBLOK chains. If ORDER=USER was specified (and at least three ALOCBLOKs are defined), the ALOCBLOKs are ordered according to the order specified on the SYSPAG macro.

If the SYSPAG macro was coded with the ORDER=SYSTEM option (the default), and if at least four ALOCBLOKs are defined, DMKVDH will scan the ALOCBLOK chain and attempt to improve upon the ordering. The goal will be to spread out the ALOCBLOKs over the control units as much as possible in an attempt to avoid excessive I/O contention.

If you use the SYSXSTOR macro to define Paging Storage, DMKPST calls DMKXST to build the ALOCBLOKs and RECBLOKs.

Pages are moved to and from main and Paging Storage via the Real Storage Manager (RSM). As in DASD allocation, the Auxiliary Storage Manager (ASM) controls the ALOCBLOK and RECBLOK structures, which map the pages of Paging Storage. A page, the smallest unit of allocation, is 4K bytes in length. For TYPE=PP, each bit in the RECMAP represents a single page. For TYPE=SW, each bit in the RECMAP represents a swap set which contains multiple pages.

For allocation, the ALOCBLOKs are chained to the SYSPLISTs. When the ASM determines that a page is being allocated in Paging Storage, it places an X'FF' in the device code byte (fourth byte) of the slot address, to indicate that the page is in Paging Storage.

There is no RDEVBLOK for Paging Storage. Instead, ALOCBLOKs are chained to a special anchor (APSTALOC) in the PSA. When the ASM determines that a page in Paging Storage is being de-allocated, (via the X'FF' in the device code byte), this anchor is used as the starting point to find the correct ALOCBLOK and RECBLOK.

Slot Allocation

DMKPGT is called to supply the address of an available slot or slots when they are required. When CP needs DASD space for system paging, system spooling, or page migration, DMKPGT allocates one DASD slot. When CP needs DASD space for swapping or dumps, DMKPGT allocates contiguous DASD slots.

DMKPGT maintains the chains of ALOCBLOKs normally created during CP initialization. It has several entry points, which correspond to the type of DASD space the caller requires. When DMKPGT cannot allocate space from the requested area, it allocates space from a lower level, unless the request is for swap space. Unfulfilled swap requests cause a non-zero return code to be returned to the caller. Dump requests that cannot be satisfied from TYPE=DU areas will be satisfied from TYPE=PS areas.

DMKPGT uses an algorithm called **N-select** to select multiple (N) slots on the same cylinder of the same device before proceeding to the next defined paging area as described by the next ALOCBLOK on a chain. This enables CP to write multiple pages to the same device with a single SIO.

When DMKPGT receives a request for DASD space, it examines the "current" ALOCBLOK, the one that it used to satisfy the last request from the given area. The current ALOCBLOK is pointed to by the anchor in the SYSPLIST block and contains a count of consecutive allocations to the given area. This count is the N-select count. Each time DMKPGT allocates DASD space using the current ALOCBLOK, it decreases the N-select count until the cylinder is fully allocated or the area has been used for "N" consecutive allocations. Then the N-select count goes to zero. (The N-select count is set to zero when paging I/O is started to this area.) The initial value of "N" for a given ALOCBLOK is determined at system initialization and depends on the device type and the use for a particular DASD area as defined by the SYSPAG macro.

When the count for the current ALOCBLOK is zero, DMKPGT uses the next non-full ALOCBLOK on a particular SYSPLIST level. When all the ALOCBLOKs on a particular SYSPLIST level are full, DMKPGT goes to the next (lower) SYSPLIST level (except, as mentioned, for dump requests).

Note: Spooling (PS) and page migration (PM) areas have an N-select value of 1. Consecutive requests for these slots are not related to any one user, and the requests are not grouped by the caller, so there is no reason to group these pages together on DASD. Swap (SW) areas also have an N-select value of 1, where one set of swap set pages will be allocated per request, rather than one page. Allocation for swap sets tends to be on different devices to enable more than one swap set to be in transit at the same time.

Cylinder Allocation

DMKPGT controls the swapping, paging, and spooling I/O load of the system by allocating cylinders evenly across all available channels and devices. In order for a device to be considered available for the allocation of paging and spooling space:

- Its volume serial number must appear in the system's owned list
- For CKD DASD, it must have at least one cylinder of PAGE or TEMP allocated space marked as available in the cylinder allocation block which is located on cylinder 0, head 0, record 4
- For FBA DASD, it must have at least one page of PAGE or TEMP space marked as available in the allocation extent map located in blocks 3 and 4
- It must not be an MSS 3330V volume.

DMKPGT allocates DASD slots by using the chains of ALOCBLOKs and RECBLOKS that are created at system initialization. It tries to select sparsely populated cylinders so that it will frequently be able to allocate many slots from the same cylinder. When the current cylinder is full, DMKPGT uses an algorithm called **the moving cursor**.

The moving cursor is actually a pointer which steadily moves through the chain of RECBLOKs. When a RECBLOK indicates that the last cylinder in the extent, as defined by the ALOCBLOK, is full, the cursor changes direction and moves back through the chain of RECBLOKs. Every time the moving cursor reaches the end of an extent and changes direction, it will usually skip over the first few cylinders it examines because they are the ones that were most recently allocated and are probably full, or nearly so. As the moving cursor progresses, however, it is more and more likely to find cylinders that have much unallocated space. This algorithm thus helps DMKPGT cut down on the time spent seeking an available cylinder.

In some cases, DMKPGT uses a modified moving cursor. The modified moving cursor is used for cylinder allocation for the 3880-11/3350 or the 3880-21/3350, and for page migration and spooling areas (TYPE=PM or TYPE=PS). The 3880-11 and 3880-21 uses a storage-like cache and works best by using recently deallocated slots instead of allocating from another cylinder. Page migration and spooling areas, as mentioned earlier, are not requested for contiguous slots, so there is no reason to have the moving cursor waste time examining full cylinders when it reverses direction.

The modified moving cursor starts at the first non-full cylinder of the extent. It moves in only one direction and starts over frequently. The cursor is thereby reset and enables DMKPGT to ensure that deallocated DASD slots will be reused in a minimum amount of time. DMKPGU, the slot deallocation routine, resets the cursor if the deallocated slot is on a lower-numbered cylinder.

If paging and swapping areas (TYPE=PP/PG and TYPE=SW) are allocated on the same 3880-11/3350 or 3880-21/3350, a reverse modified moving cursor algorithm is used for the first area and a modified moving cursor algorithm is used for the second area.

Page/Swapset Allocation Summary

Syspag Type	*Moving Cursor*	*"N" Select (Fixed #'s)*
Type = PP, SW		
All devices	Yes	1 Swapset
Type = PP, PG		
3880 Models 11, 21	Modified*	1 Page
3380	Yes	10 Pages
3330, 3350, 3375	Yes	8 Pages
3310, 3370	Yes	3 "Pages"
2305, 3340	Yes	3 Pages
Type = PM, PS, DU		
All devices	Modified	1 Page

- Devices supported for Swapping:
  3330, 3350, 3375, 3380, 2305.

- Extended CKD devices are not supported for swapping for speed matching buffers.

Paging I/O

DMKPAGIO handles all I/O requests for virtual storage and spooling pages. It builds the necessary IOBLOKs and channel programs, expands compressed slot addresses, and maintains two I/O request queues of CPEXBLOKs (one dummy queue and one in-transit queue) for pages to be moved. When possible, DMKPAGIO chains requests together to move several pages with one SIO.

DMKPAGIO maintains two stacks of preformatted paging IOBLOKs. One stack contains preformatted IOBLOKs for extended CKD devices, and is anchored from DMKPAGEX. The other stack contains preformatted IOBLOKs for regular FBA/CKD devices and is anchored from DMKPAGSK. When I/O operations complete, their IOBLOKs are added to a list of available blocks. When DMKPAGIO needs new IOBLOKs, it takes them from the appropriate list. When either list is empty, DMKPAGIO calls DMKFREE as needed to get storage for a new block.

DMKPAG also maintains a stack of IOBLOK extensions anchored from DMKPAGXS. These are used for swap requests, and are pointed to by the IOBLOK for the swap request.

---

* Reverse modified moving cursor is used if paging area (TYPE = PP or PG) and swapping area (TYPE = SW) are allocated on same 3880-11/3350 or 3880-21/3350.

DMKPAGIO is entered by a GOTO from:

- DMKPTR, to read virtual storage pages or swap sets
- DMKSEL, to write virtual storage pages
- DMKSWA, to write swap sets
- DMKRPA, to read and write virtual storage spool buffers.

When DMKPAGIO is called for paging (not swapping), it receives the CORTABLE entry for the page to be moved, the address of a swap table entry for the slot, a read or write operation code, and the address of a CPEXBLOK that will be stacked for dispatching after the I/O for the page has completed. When DMKPAGIO is called for swapping, it receives a swap set block (SSBLOK) address instead of a CORTABLE entry.

DMKPAGIO indexes into the system OWNDLIST by using the device code that is part of the page address. It finds a device to which to direct the I/O request. Then it looks for a preformatted IOBLOK on the proper stack.

As DMKPAGIO creates IOBLOKS, it slot-sorts them and queues them to dummy RDEVBLOKs that were created at system initialization. The pointer to the dummy RDEVBLOKs is stored at DMKPAGRD. This enables DMKPAGIO to chain the IOBLOKs for several pages together, so they can be paged with one SIO. When the I/O request queues are empty, DMKPAGIO unstacks the IOBLOKS, unstacks the corresponding CPEXBLOKs from the dummy queue and adds them to the in-transit queue, and calls DMKIOS to start the I/O operation.

Note: Swap sets are already grouped together, and their requests are started immediately. I/O done on a 3880-11 is also started immediately.

DMKPAGIO also periodically calculates system paging overhead. After 200 pages have been moved (read or written), the elapsed time for the 200 page moves is computed, and the paging rate is calculated in page moves per second. The recent paging load, expressed as the percentage of time that more than one half of the system's pages were idle due to page-wait, is averaged with the previous load and re-projected as the expected load for the next interval.

Preferred System Paging

Preferred system paging is high speed paging support that is carried out by the 3880 storage control system. The 3880 Model 11 storage control system consists of two levels of storage (an electronic storage array and variable numbers of 3350 DASDs), and two paging storage directors, only one of which can be used to access the storage array.

The 3880 Model 21 storage control system also consists of two levels of storage, (a cache and variable numbers of 3350 DASDs), and two paging storage directors, both of which can be used to access the cache.

Page I/O Request Queuing Algorithm

The ordering of page I/O requests that are chained together for initiation with one SIO is done on a priority ordering basis. The priority is:

1. In-queue requests
2. Not-in-queue requests
3. Reads
4. Writes
5. Q1 requests
6. Q2 requests.

Page Migration and Swap Table Migration

Both page and swap table migration are invoked through the performance monitoring routine (DMKSTP) or, for page migration only, via the migrate command. It has the responsibility of determining whether conditions warrant invoking page migration and/or swap table migration.

Page Migration

The purpose of page migration is to promote efficient usage of the preferred devices in a storage hierarchy. The specific goal is to keep the primary paging (PP) area available for truly active pages, and also the general paging (PG) area if necessary. Swap sets in the swapping (SW) area (Paging Storage only) are also migrated. The page migration routine (DMKPGM) is invoked via a CP request block by DMKSTP. It operates according to Figure 30 on page 161.

Note: Swap sets in Paging Storage are migrated to the next TYPE=SW level, if one exists. Otherwise, they are broken apart and each page is migrated separately to the first available non-Paging Storage *level of TYPE=PP.*

DMKSTP invokes page migration if the percentage of occupied storage on the PP, PG or SW level reaches the value set by the SET SRM PGFULL command. (The default value is 99 percent.)

To migrate pages, DMKPGM scans each virtual machine's segment and page tables. If a page is selected for migration, DMKPGM checks the level on which the page is located. If the page is on a PP or PG level, DMKPGM checks to see if enough pages of that type have been migrated. If so, the page is not migrated. Otherwise, the address of the first SYSPLIST of the next level type is saved for use when calling DMKPGTPM for a DASD slot for migration.

You can monitor page migration performance with the QUERY SRM PGMACT command. For more information, see *VM/SP HPO CP for System Programming*.

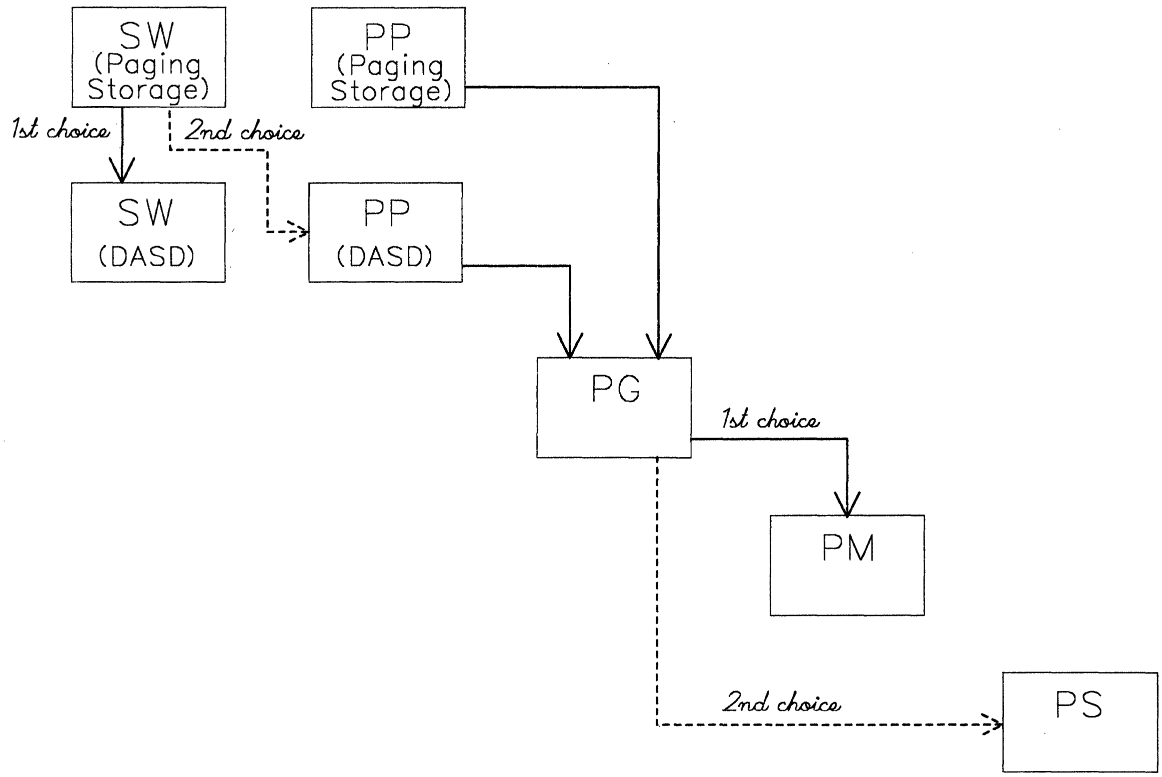


Figure 30. Page Migration Hierarchy

Swap Table Migration

DMKSTP invokes swap table migration if the following condition is met: the space occupied by page tables and swap tables is equivalent to at least 12% of nonextended free storage.

DMKSTR scans the VMBLOK chain for virtual machines with swap tables that are eligible for migration. DMKSTR migrates the swap table, setting bits to mark where the migratable pages in the segment reside on DASD (TYPE=PP or PG). This information is used by DMKPGM to determine if a swap table should be restored, in order to migrate more of the pages in that segment.

The swap table is brought back into storage whenever a reference is made to that segment. The page translation routine (DMKPTR) always calls the swap table migration routine (DMKSTR) whenever a reference is made to a segment table entry with a zero page table pointer. If appropriate, DMKSTR calls DMKBLD to have a page table and a swap table built for that segment. If the swap table has been migrated, the 4K buffer page is brought into storage and the swap table information is copied into the swap table.

Virtual Storage Paging Error Recovery

Errors encountered during virtual storage (as opposed to spooling) paging operations can generally be classified as either soft or hard errors. Soft errors allow the system to continue operation without delay or degradation. Hard errors can cause noticeable effects such as the abnormal termination of user tasks (abend) and response degradation. Errors that are successfully retried or corrected are known only to the I/O supervisor and the I/O error retry and recording routines—they appear to the second level interruption handlers (such as WAITPAGE) as if the original operation completed normally.

Soft Error Recovery: An I/O error that occurs on a page swap-out is considered to be a soft error. DMKPTRAN calls DMKPGTPG to assign a different DASD page slot and the page is re-queued for output. The slot that caused the error is not de-allocated, and thus is not assigned to another virtual machine. All other uncorrectable paging errors are hard because they more drastically affect system performance.

Hard Error Recovery: Hard paging errors occur on either I/O errors for page reads or upon exhausting the system's spooling and paging space. Recovery attempted on hard errors depends upon the nature of the task for which the read was being done. If the operation was an attempt to place a page of a virtual machine's virtual storage into real storage, the operation of that particular virtual machine is terminated by setting the page frame in error to zero and placing the virtual machine in console function mode. The user and operator are informed of the condition, and the page frame causing the error is not de-allocated, thereby ensuring that it is not allocated to another user.

The control program functions that call DMKPTRAN (such as spooling, pageable control program calls, and system directory management) have the option of requesting that unrecoverable errors be returned to the caller. In this case, the CP task may attempt some recovery to keep the entire system from terminating (abend). In general, every attempt is made to at least allow the operator to bring the system to orderly shutdown if continued operation is impossible.

Proper installation planning should make the occurrence of a space exhaustion error an exception. An unusually heavy user load and a backed-up spooling file could cause this to happen. The operator is warned when 90% of the temporary (paging/spooling) space in the system is exhausted. He should take immediate steps to alleviate the shortage. Possible remedies that exist include preventing more users from logging on and requesting users to stop output spooling operations. More drastic measures might include the purging of low-priority spool files. If the system's paging space is completely exhausted, the operation of virtual machines progressively slows as more and more users have paging requests that cannot be satisfied and operator intervention is required.

Virtual Relocation

CP provides the virtual machine the capability of using the DAT feature of the real System/370. Programming simulation and hardware features are combined to allow usage of all of the available features in the real hardware, (that is, 2K or 4K pages, 64K or 1M segments).

Note: *When greater than 16 megabytes of storage are configured and online, any virtual machine running with DAT turned on must use 4K page size. Virtual machines that have 2K DAT tables do not have enough bits to store the page frame address in the page frame table entry for pages resident above the 16 Mb line—this is because of the S/370 architecture for extended addressing. A virtual machine that attempts to use DAT with 2K page size will receive a program check.*

For clarification, some term definitions follow:

First-level storage: The physical storage of the real CPU, in which CP resides.

Second-level storage: The virtual storage available to any virtual machine, maintained by CP.

Third-level storage: The virtual storage space defined by the system operating in second-level storage, under control of page and segment tables which reside in second-level storage.

Page and segment tables: Logical mapping between first-level and second-level storage.

Virtual page and segment tables: Logical mapping between second-level and third-level storage.

Shadow page and segment tables: Logical mapping between first-level storage and third-level storage.

A standard, nonrelocating virtual machine in CP is provided with a single control register, control register zero, that can be used for:

- Extended masking of external interruptions
- Special interruption traps for SSM
- Enabling of virtual block multiplexing.

A virtual machine that is allowed to use the extended control feature of System/370 is provided with a full complement of 16 control registers, allowing virtual monitor calls, PER, extended channel masking, and dynamic address translation.

An extension to the normal virtual-machine VMBLOK is built at the time that an extended control virtual machine logs onto CP. This ECBLOK contains the 16 virtual control registers, 2 shadow control registers, and several words of information for maintenance of the shadow tables, virtual CPU timer, virtual TOD clock comparator, and virtual PER event data. The majority of the processing for virtual address translation is performed by the module DMKVAT, with additional routines in DMKPRG, DMKPRV, DMKDSP, DMKCDB, DMKLOG, DMKUSO, and DMKPTR. The simulation of the relocation-control instructions (that is, LCTL, STCTL, PTLB, RRB, and LRA) is performed by DMKPRV or DMKFPS. These instructions, with the exception of LCTL and STCTL, are not available to virtual machines which are not allowed the extended control mode.

When an extended-control virtual machine is first active, it has only the real page and segment tables provided for it by CP and operates entirely in second-level storage. DMKPRV examines each PSW loaded via LPSW to determine when the virtual machine enters or leaves extended control or translate mode, setting the appropriate flag bits in the VMBLOK. Flag bits are also set whenever the virtual machine modifies control registers 0 or 1, the registers that control the dynamic address translation feature. DMKDSP also examines PSWs that are loaded as the result of interruptions to determine any changes in the virtual machine's operating mode. The virtual machine can load or store any of the control registers, enter or leave extended control mode, take interruptions, etc., without invoking the address translation feature.

If the virtual machine, already in extended control mode, turns on the translate bit in the EC mode PSW, then the DMKVATMD routine is called to examine the virtual control registers and build the required shadow tables. (Shadow tables are required because the real DAT hardware is capable of only a first-level storage mapping.) DMKVATMD examines virtual control registers 0 and 1 to determine if they contain valid information for use in constructing the shadow tables. Control register zero specifies the size of the page and segment the virtual machine is using in the virtual page and segment tables. The shadow tables constructed by DMKVATMD are always in the same format as the virtual tables.

The shadow segment table is constructed in first-level storage and initialized to indicate that all segments are unavailable. Flags are maintained in the VMBLOK to indicate that the shadow tables exist. DMKVATMD also constructs the shadow control registers 0 and 1. Shadow control register 0 contains the external interruption mask bits used by CP, mixed with the hardware controls and enabling bits from virtual control register 0. Shadow control register 1 contains the segment table origin address of the shadow segment table.

When the virtual machine is operating in virtual translate mode, CP loads the shadow control registers into the real control registers and dispatches the user. The immediate result of attempting to execute an instruction is a segment exception, intercepted by DMKPRG and passed to DMKVATSX. DMKVATSX examines the virtual segment table in second-level storage.

If the virtual segment is not available, the segment exception interruption is reflected to the virtual machine. If the virtual segment is marked available, then DMKVATSX:

- Allocates one full segment of shadow page table, in the format specified by virtual control register 0.
- Sets all of the page table entries to indicate page not in storage.
- Marks the segment available in the shadow segment table.
- Redispatches the virtual machine via DMKDSP.

Once again, the immediate result is an interruption, which is a paging exception and control is passed to DMKVATPX. DMKVATPX references the virtual page table in second-level storage to determine if the virtual page is available. If the virtual page is not available, the paging interruption is reflected to the virtual machine. However, if the virtual page is marked in storage, the virtual page table entry determines which page of second-level storage is being referenced by the third-level storage address provided. DMKVATPX next determines if that page of second-level storage is resident in first-level storage at that time. If so, the appropriate entry in the shadow page table is filled in and marked in storage. If not, the required page is brought into first-level storage via DMKPTRAN and the shadow page table filled in as above.

As the virtual machine continues execution, more shadow tables are filled in or allocated as the third-level storage locations are referenced. Whenever a new segment is referenced, another segment of shadow page tables is allocated. Whenever a new page is referenced, the appropriate shadow page table entry is validated, etc. No changes are made in the shadow tables if the virtual machine leaves translate mode (usually via an interruption), unless it also leaves extended control mode. Dropping out of EC mode is the signal for CP to release all of the shadow page and segment tables and the copy of the virtual segment table.

There are some situations that require invalidating the shadow tables constructed by CP or even releasing and reallocating them. Whenever DMKPTR pages out a page that belongs to a virtual relocating machine, DMKVATSI is called to selectively invalidate the shadow page tables. If the stolen page is below the high-water mark, the shadow page table entry for the stolen page is invalidated. If the stolen page is above the high-water mark and virtual machine assist is on, a bit is set in the VMBLOK to indicate that all of the shadow page tables above the high-water mark must be invalidated. The actual invalidation is handled by DMKVATAB, called from DMKDSP when the virtual machine is about to be dispatched. If the stolen page is above the high-water mark and virtual machine assist is off, the shadow page tables are scanned to selectively invalidate shadow page table entries that map to the real page being stolen.

The other situations which cause shadow table invalidation arise from the simulation of privileged instructions in DMKPRV or DMKFPS. Flags are set in the VMBLOK whenever the virtual machine loads either control register 0 or 1, and DMKPRV calls DMKVATAB to perform whatever maintenance is required. When control register 1 is loaded by the virtual machine, DMKVATAB scans the chain of STOBLOKs to see if shadow tables are already allocated for the value in virtual control register 1. If a matching STOBLOK is found, it is requeued as the first in the STOBLOK chain and the virtual machine can be redispatched. If a matching STOBLOK is not found, and the number of STOBLOKs is equal to the maximum STOBLOK count, the last STOBLOK in the chain is reused by invalidating the entire shadow table and then queuing it first on the STOBLOK chain. If the number of STOBLOKs is less than the maximum STOBLOK count, a new STOBLOK is acquired and initialized, and placed first in the STOBLOK chain. When control register 0 is loaded, DMKVATAB examines the relocation-architecture control bits to determine if they have changed, (such that the format of the virtual page and segment tables no longer matches that of the shadow tables). If the format has not changed, the shadow tables are left intact—otherwise, all of the shadow tables must be returned to free storage and another set, in the new format, must be allocated and initialized. The same actions can result from modifying the control registers via the CP console functions, in which case DMKVATAB is called from DMKCDB. The privileged operation, PTLB, causes the shadow page tables to be invalidated above the high-water mark because the shadow tables are the logical equivalent of the translation look-aside buffer.

The privileged instruction LRA is simulated via DMKVATLA, which searches the virtual page and segment tables to translate a third-level storage address to a second-level storage address, returning a condition code indicator to DMKPRV, or forcing an interruption if the tables are incorrectly formatted.

Most error situations that occur in the virtual machine are handled by means of the extended program interruptions associated with the real address translation hardware. Whenever a virtual relocating machine loads control registers 0 or 1 with an invalid value, DMKVAT releases all of the shadow tables exactly as if the hardware controls had changed. The shadow control registers are set valid, with the shadow segment table re-allocated at a minimum size and all segments marked unavailable. Flag bits are set in the VMBLOK to indicate that the shadow tables are artificially valid, and DMKVATSX reflects a translation specification exception to the virtual machine as soon as it is dispatched. While it is possible for the virtual machine to enter an interruption loop (if the new PSW is also a translate mode PSW), the cited process prevents the occurrence of a disabled loop within CP, which would result if the virtual machine is never dispatched.

Extended Storage Key Support

DMKPRV checks the status of the reference and change bits in the virtual storage keys, which involve the privileged instructions ISK, ISKE, RRB, RRBE, SSK, and SSKE.

For SSK, CP increases simulation counter DMKPRVEK, sets general purpose register 0 to X'0C' and branches to subroutine CKCR0B7 to simulate the instruction.

For ISK, CP increases the simulation counter DMKPRVIK and sets to X'04' general purpose register 0. DMKPRV uses BAL to invoke subroutine CKCR0B7.

For RRB, CP increases simulation counter DMKPRVRR and sets to X'08' general purpose register 0. DMKPRV uses BAL to invoke subroutine CKCR0B7.

Subroutine CKCR0B7 examines the XKEYMODE field in the PSA to determine the type of real storage frames that are installed on the processor. If XKEYMODE indicates that the storage for this virtual machine consists entirely of frames that are protected by single storage protection keys, CP does the following:

- Examines bit 7 of control register 0 (VMVCR0 if in BC mode, or EXTCR0 if in EC mode)
- If control register 0 bit 7 is B'1', control passes to DMKPRWXK to simulate the proper extended key operation for the virtual machine
- If control register 0 bit 7 is B'0', CP returns a special operation exception to the virtual machine.

If XKEYMODE indicates that the storage for this virtual machine consists entirely of frames that are protected by two storage protection keys, control returns to DMKPRV to simulate the key operation.

The extended key instructions ISKE, SSKE, and RRBE are used for manipulating the storage keys of single key real storage frames. CP simulates the extended key instructions only when they are installed on the processor. To use the extended key instructions, the virtual machine must be operating in extended control mode or an operation exception results.

Module DMKPRV validates the ISKE, SSKE, and RRBE instructions and increases the simulation counters DMKPRVXI, DMKPRVXR, and DMKPRVXS respectively, to keep track of the number of times that the instructions are simulated. Control passes to DMKPRW at entry point DMKPRWXK for simulation of the instruction.

Before simulating an extended key operation, CP examines the XKEYMODE field in the PSA. XKEYMODE contains the value B'1' if one or more single key real storage frames were found during system initialization. CP simulates extended key operations for guest virtual machines only when the XKEYMODE field is set to B'1'.

If the processor has single-key storage frames and virtual machine assist can process the extended key instructions, it does so, provided the assist is there and enabled. If the assist is not there, disabled, or cannot handle the instruction, it is passed to CP for simulation.

Virtual machine assist processes extended key instructions for processors that do not have single key real storage frames. That is, if the extended key instructions are installed on the processor, and if virtual machine assist can process the instruction, the presence of single key real storage frames is irrelevant. If single processor mode is active, virtual machine assist will not simulate ISK, ISKE, SSK, or SSKE instructions.

Free Storage Management

DMKFRE and DMKFRT are responsible for the management of free storage. CP uses them to obtain free storage for I/O tasks, CCW strings, I/O buffers, and almost all other related applications. However, CP does not use them for real channel control blocks, real control unit blocks, real device blocks, or the CORTABLE.

The way storage is allocated depends on the amount of storage requested. Block sizes of 1024 doublewords or fewer are grouped into 92 subpool sizes. The subpool sizes range from 2 doublewords to 1024 doublewords.

Requests for 128 doublewords or fewer are rounded to the next highest boundary that is a multiple of two (2,4,6,...128). For example, DMKFREE would try to honor a request for seven doublewords by searching for a subpool size of eight doublewords. Requests for greater than 128 doublewords are rounded to the next highest boundary that is a multiple of 32 (160,192,224,...) to a maximum subpool size of 1024 doublewords. Block sizes of 1024 doublewords or fewer are handled by LIFO (push-down stack) logic. Block sizes of greater than 1024 doublewords are strung off a chained list.

When subpools are exhausted, small blocks are generally obtained from the first larger block at the end of available free storage. Large blocks, on the other hand, are obtained from the high-numbered end of the last larger block. This procedure tends to keep the volatile small subpool blocks separated from the large blocks, some of which stay in storage for much longer periods of time. Thus, undue fragmenting of available storage is avoided.

DMKFRE initially starts without any subpool blocks. They are obtained from DMKFREE and returned to DMKFRET on a demand basis.

The various cases of calls to DMKFREE for obtaining free storage, or to DMKFRET for returning it, for subpool sizes and large sizes, are handled as follows.

Calling DMKFREE for a Subpool

The subpool algorithm maintains two entire sets of subpools. Each processor maintains its own subpool table when it requires subpool-sized blocks.

Subpool Available: When there is a call to DMKFREE for a subpool, the appropriate subpool for the requested size is checked to see if there is at least one suitable block available. If one is available, the first one found is detached from the chain, the chain is pointed to the next subpool block of the same size (if any), and the detached block is returned to the caller.

Subpool Not Available: In an AP or MP configured system, if "NUMSTEAL + MINLEAVE" (2+3) blocks exist on the other processor's subpool of the the desired size, then NUMSTEAL (2) blocks will be stolen. One is added to this processor's subpool and the other is used to satisfy the caller's request. If no block of suitable size is available when a call to DMKFREE is made, the way the request is then handled depends on the size of the request. Requests for more than 28 doublewords are handled in the same manner as requests for greater than 1024 doublewords (see "Calling DMKFREE for a Large Block").

If the request is for 28 doublewords or less, the chained list of free storage is searched for a block of equal or larger size. The first block of equal or larger storage outside the dynamic paging area is used to satisfy the call, with a block of equal size taking priority. If an equal block is found, it is detached from the chain and returned to the caller. If a larger block must be used, the low-numbered end is split off and returned to the caller.

If there is still no block large enough to satisfy the request, then DMKPTRFR is called to obtain another page from the dynamic paging area. This page is merged into the chain of free storage, and the above process is repeated until DMKFREE obtains the needed block.

Calling DMKFREE for a Large Block

If there is a call to DMKFREE for a block larger than 1024 doublewords, or if no subpool is available for a request of greater than 28 doublewords, the chained list of free storage is searched for a block of equal or larger size outside the dynamic paging area, with a block of equal size taking priority. If such an equal size block is found, it is detached from the chain and given to the caller. If at least one such larger block is found, the desired block size is split off the high-numbered end of the last larger block found and given to the caller.

If a block of suitable size is not found outside the dynamic paging area, a block within the dynamic paging area may be used to satisfy the request. If no such block exists, DMKPTRFR is called to obtain another page of storage from the dynamic paging area. This page is merged into the chain of free storage. The above process is repeated (as necessary) until DMKFREE obtains the needed block.

Calling DMKFREE for a Prime Storage Block

A special call can be made to DMKFREEP for a 16 doubleword prime storage block (equivalent in size to one cache-line). There exists one prime list for each processor in the AP or MP configured system. Typically, these blocks will be used frequently for short periods of time (for example, CPEXBLOKs, IOBLOKs, SVC SAVEAREAs).

Calling DMKFREE for an Align Pool Block

A special call can be made to DMKFREEA for a block size of 16, 32, 48, 64, 80, 96, 112, or 128 doublewords (cache-aligned). There exists only one align pool for use by both processors in the AP or MP configured system. These blocks are used for long periods of time and are referenced frequently (for example, VMBLOKs).

Calling DMKFRET for a Subpool

DMKFRET processes the CP Assist Fret instruction (E601) to return free storage. If the microcode cannot return the block to an appropriate subpool, control is passed to DMKFRTT to handle the request.

If a subpool sized block is given back via a call to DMKFRET, the block is attached to the appropriate subpool chain on a LIFO (push-down stack) basis, and return is made to the caller. If the block was in a page within the dynamic paging area, it is also placed in a subpool chain.

(Subpool storage is returned, however, via a call to DMKFRTRS, to the regular free storage chain once every hour or when a user logs off. DMKFRTRS then calls DMKFRTSN to search the free storage chain for page frame sized blocks needing to be returned to the dynamic paging area.)

Calling DMKFRET for a Large Block

If a block larger than 1024 doublewords is returned via DMKFRET, control is passed to DMKFRTT where the block is merged appropriately into the regular free storage chain. Then a check is made to see if the area given back (after all merging has been done) is a page frame within the dynamic paging area. If so, DMKPTTFT is called to return the block to the dynamic paging area for subsequent use. If the block is returned by specific modules known to use large blocks frequently for very short periods of time, DMKPTTFT is purposely not called, to avoid continuous extending and unextending of free storage over very short time intervals.

Calling DMKFRET for a Prime Storage Block

When a prime storage block is returned via DMKFRET, it is attached to the appropriate processor's prime list on a LIFO (push-down stack) basis.

Calling DMKFRET for an Align Pool Block

When an align pool block is returned via DMKFRET, it is inserted in the appropriate list in the align pool (32 doubleword blocks are attached to the 32 doubleword list, etc.). Blocks are chained together in ascending order according to starting address.

Align Pool Storage Collector

DMKFRTRA is the align pool storage collector. It tries to return pages to the dynamic paging area when they are no longer needed in an aligned storage pool. DMKFRTRA looks for such pages whenever a user logs off and every hour after system IPL. It is invoked as part of a call to DMKFRTRS, the regular subpool collector.

DMKFRTRA scans the subpools looking for contiguous cache-aligned blocks. When it finds enough such blocks to make up one or more complete pages, it unchains these blocks from the appropriate list. It then calls DMKPTTFT to return the page or pages to the dynamic paging area.

Free Storage Page Frame Allocation

The number of page frames allocated to free storage depends upon:

1. The real machine storage size

2. The RMSIZE operand specified in the SYSCOR macro at system generation time

3. The PRIME operand in the SYSCOR macro

4. The FREE operand in the SYSCOR macro

5. The number of unusable or inaccessible storage frames detected during CP initialization (308x processors only).

The storage size used by VM/SP HPO is the smaller of the real machine storage size and the RMSIZE value.

If the FREE operand was not included in the SYSCOR macro statement for DMKSYS, the default amount of free storage DMKSTA allocates at IPL time is three page frames for the first 256K of real storage (not including a V=R area, if any), and one page frame for each additional 64K below the 16 Mb line. If more than 16 Mb of storage is online, then one additional page frame will be allocated for each 256K above the 16 Mb line.

If the PRIME operand was not included in the SYSCOR macro statement for DMKSYS, the default amount of prime storage that DMKSTA allocates at IPL time is 10% of free storage.

DMKSTA will allocate an additional 25% of free storage in AP and MP environments.

If the FREE operand was included in the SYSCOR macro statement for DMKSYS, that value is the number of fixed free storage page frames allocated at IPL time.

When the values for free storage and trace area storage are too large, they are decreased proportionally to guarantee that a predetermined number of DPA pages exist. Prime storage is also adjusted by this percentage.

CP FRET Trap

The CP FRET Trap detects the release of areas of free storage that were not assigned, previously released, or outside the boundaries of the storage given. Based on the value of the option &FRETRAP, the trap code is conditionally assembled in modules DMKCPI, DMKFRE, DMKFRT, and DMKSVC. &FRETRAP can be found in OPTIONS COPY and has a default value of 0 for normal operations without the trap. The trap may be installed at system generation time. Refer to the *VM/SP HPO Installation Guide* for the installation instructions.

When the trap is installed, DMKCPI disables CP Assists FREEP, FREE, FRET, DSP1, DSP2, DSP3, DSP4, UNTFR, and Diagnose '18' during system initialization. DMKFRE adds two doublewords to each free storage request, creating a trap extension area. The extension area contains:

- The status of the request. The status consists of the tag ALLO when the storage is allocated by DMKFRE or the tag FRET when the storage is released by DMKFRT.
- The saved size, in doublewords, of the requested free storage area.
- The starting address of the assigned free storage block.
- The return address of the module requesting the storage.
- The last three bytes of the calling module's name, if it is pageable.

For the format of the extension area, refer to the FREEXT control block in the *VM/SP HPO Data Areas and Control Block Logic Volume 1 (CP).*

DMKFRT checks each request to release free storage for an ALLO tag. It checks at the address calculated by adding the size in bytes of the storage block being released to the address of the block. If the ALLO tag is found, the size of the free storage block being released is checked against the saved size in the extension area. If the sizes are equal, the ALLO tag is changed to FRET. The requested free storage block, including the extension area, is then released.

For prime storage requests, the extension area contains:

- The last three bytes of the calling module's name (if it is pageable)
- The low order byte of the size request
- The return address of module requesting the storage
- The saved size (in doublewords and bytes).

Trap Error Detection

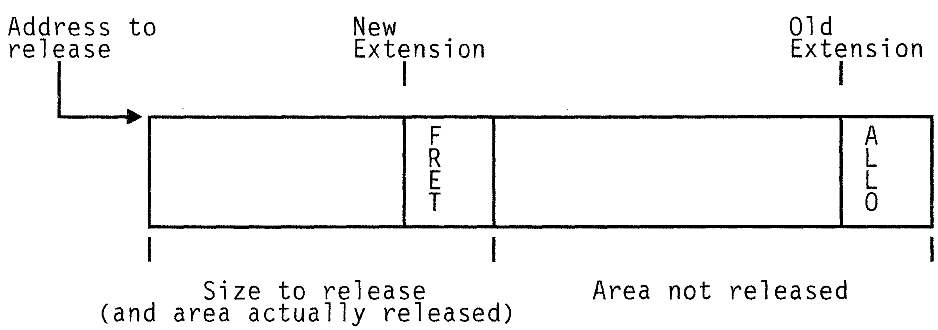
When storage is released, for systems generated in uniprocessor mode (or in AP/MP mode and the second processor is not operating), the trap code in DMKFRT detects three types of errors. If the extension area cannot be located, the system abends with code FRT013. If the tag in the extension area is FRET instead of ALLO, indicating that the storage has already been released, the system abends with code FRT016. If the size of the free storage block being released does not match the saved size in the extension area, the system abends with code FRT015.

With the prime storage extension, two other types of errors may be detected. If the prime block is not marked prime, the system abends with either code FRT20 or SVC06. If the prime block is not on a cache boundary, the system abends with code FRT21.

AP/MP Differences

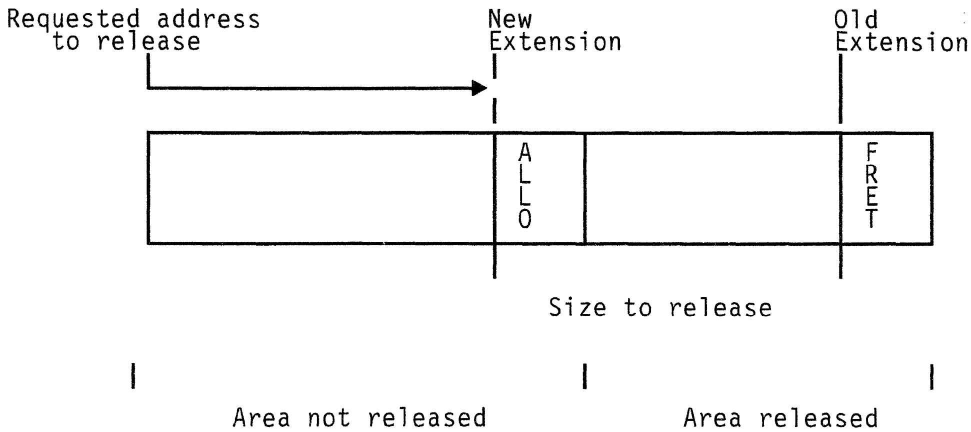
For systems generated in AP/MP mode, and the second processor is operating, the area released can be a split of the storage area given. Either the front or the rear portion of the free storage area may be released. Storage splits are allowed due to the way CP handles page and swap tables for shared systems in the AP/MP environment. With one call to DMKFREE, two sets of contiguous page and swap tables are obtained (one for each processor). Under some conditions, the page and swap tables for only one processor are released. The trap code allows only “valid” splits. A split is considered “valid” if both parts of the split free storage area are each the size of a contiguous page and swap table, and the saved size in the extension area is the size of two page and swap tables.

Valid Split: Release of Front Portion



When the front portion of the free storage block is being released, the extension area is updated: the saved size is adjusted to the size of a page and swap table, and the new address of the area not being released is stored. The information in the extension area is copied to create a new extension area. The new extension area is constructed within the front portion being released, using the last two doublewords. The new extension area is updated with the address of the front portion and the FRET tag. The front portion of the storage area is returned to the DMKFRELS chain.

Valid Split: Release of Rear Portion



When the rear portion of the free storage block is being released, the extension area is copied to create a new extension. The new extension area is constructed at the end of the front portion that was not requested to be released. The saved size, in both the old and the new extension areas, is adjusted to the size of a page and swap table. The old extension is updated with the new address of the rear portion actually released, information about the module releasing the storage, and the FRET tag. The new address of the rear portion is calculated by adding the requested address to release to the size in bytes of the new extension area. At this new address, the rear portion of the free storage block is returned to the DMKFRELS chain (including the old extension).

Trap Error Detection in AP/MP Mode

For systems generated in AP/MP mode when the second processor is operating, the trap code takes into account the possibility of storage splits while checking for errors. The trap code checks for the ALLO tag at the address calculated by adding the address of the free storage block to the size in bytes of the returned free storage area. If at this address a FRET tag is found, the system abends with code FRT016. If neither the ALLO or FRET tag is found, the trap code checks for an ALLO tag at that address plus the size in bytes of a page and swap table. The system abends with code FRT013 if the ALLO tag is still not found. If the ALLO tag is found at one of the two addresses checked, but the size of the free storage area to be released does not match the saved size in the extension area, then the trap checks for a "valid" split. Abend FRT015 occurs if the split is not "valid."

CP Initialization

System initialization starts when the operator selects the DASD address of the CP system residence volume (SYSRES) and IPL's the sytem. The System/370 hardware reads 24 bytes from the SYSRES (record 1 of cylinder 0) into location 0 of main storage. This record consists of an initial PSW and a channel program. The channel program reads the module DMKCKP from the SYSRES (record 2 of cylinder 0) into location X'1000' of main storage. (The load point for a V=R system is the V=R size plus X'2000', or

DMKSLC plus X'1000'.) The hardware then loads the initial PSW from location 0 of main storage. This PSW causes execution to begin at the entry point DMKCKPT.

A VM/SP HPO system cannot be initialized on any supported processor when an unusable or nonaddressable frame is found either at location 0 of main storage or from DMKSLC through the frame in main storage where DMKSAV ends. The system initialization process will fail with unpredictable results if DMKCKP or DMKSAV cannot be loaded at their expected real storage locations.

DMKCKP will load DMKSAV from the SYSRES and then pass control to DMKSAVRS. DMKSAVRS loads the CP nucleus from the nucleus area of the SYSRES into main storage starting at page 0 and ending with the pageable module immediately preceding DMKSAV in the CP load list. Next DMKSAV passes control to DMKCPINT.

DMKCPINT performs the main initialization function. This includes:

- Calling DMKSTANT to initialize main storage
- Calling DMKMNTIO to mount the I/O devices that were defined in DMKRIO and are available
- Calling DMKSEGCP to set up CP's segment page, core, and swap tables
- Building a LANGBLOK for the installation default language
- Calling DMKOPERC to locate the operator's console
- Calling DMKTODIN to initialize the time-of-day clock
- Calling DMKCPJNT to continue system initialization
- Calling DMKUDROV to read any command class override records stored on the primary override cylinders.

When a processor is running with more than 16 megabytes of online storage, DMKSTA allocates all page frames above the 16 Mb line as a dynamic paging area anchored by DMKPTRF2, and allocates a separate dynamic paging area below the 16 Mb line that is anchored by DMKPTRF1.

For each valid CP-owned DASD volume which is online at initialization, DMKALO builds a maxi-ALOCBLOK. (DMKCPI calls DMKMNT to mount and initialize CP-owned devices. DMKMNT then calls DMKALO to build the maxi-ALOCBLOKs.) Each maxi-ALOCBLOK contains the DMKFMT allocation information from the allocation record on the DASD volume, and describes all the space on that particular volume. DMKIDU then calls DMKVDG to finish creating an ALOCBLOK chaining structure for DASD allocation.

DMKCPI then calls DMKPSTIN to initialize Paging Storage. The READ SCP INFORMATION Service Call is issued to obtain the largest valid Paging Storage increment number, as well as the number of Paging Storage pages in each increment. For each Paging Storage area specified on the SYSPAG macros, RECBLOKs and an ALOCBLOK are then permanently allocated from free storage to map the pages in the specified increments.

The installation may issue the QUERY PSTOR command to display the amount of Paging Storage available for swapping and primary paging, and the percentages of these amounts that are not usable. See the *VM/SP HPO Operator's Guide* for more information on the QUERY PSTOR command.

After DMKCPI calls DMKPST to initialize use of Paging Storage, DMKPST calls DMKVFI to initialize use of the Vector Facility.

You can use the QUERY VECTOR command to obtain information about the availability and current use of the Vector Facility. See the *VM/SP HPO Operator's Guide* or *VM/SP HPO CP for System Programming* for more information on this command.

When CP initialization occurs on a processor that has more than 16 megabytes of online storage, DMKSEG calls DMKPTSAL. DMKPTSAL allocates two system virtual address slots that CP later uses for page interchanges necessary for CP to accommodate real storage above the 16 Mb line.

Some models of the 308x, 3090, and 3033 processors are equipped with real storage frames that are protected by one storage protection key per frame, rather than two storage protection keys. At system initialization, CP assumes that all storage frames are protected by two storage keys. CP initializes the storage protection keys by issuing SSK instructions.

If during real storage initialization a special operation exception occurs, DMKSTA initializes to one the field XKEYMODE in the PSA and initializes to B'1' bit 7 of control register 0. CP initializes these areas to indicate the presence of one or more single key storage frames. CP then examines the level of the virtual machine assist feature to determine whether virtual machine assist is capable of simulating the extended key instructions (ISKE, RRBE, and SSKE). The correct level of virtual machine assist must be present on both processors of an AP or MP configured system. If virtual machine assist can process the extended key instructions, DMKSTA initializes to B'1' XKEYASST in the PSA. Control is returned to DMKCPI.

DMKCPJNT calls DMKWRMST to perform a WARM, CKPT, FORCE or COLD start, DMKIDUMP to locate DASD space for system dumps, and DMKOPELO to LOGON the system operator. When control is returned from DMKCPJ, DMKCPI goes to the dispatcher, DMKDSPCH, to begin dispatching virtual machines.

See Figure 4 on page 26 for a more detailed description of CP initialization.

Warm Start

Following an orderly system shutdown where warm start data has been saved by the checkpoint programs (DMKCKP, DMKCKD, DMKCKF, DMKCKH, DMKCKM, DMKERP, DMKCKW, and DMKCKN) it is possible to start the system with a warm start. DMKWRMST is called by DMKCPJ during system initialization to perform a warm start. DMKWRMST reconstructs the system log message, the saved accounting ACNTBLOKs, the saved printer, punch, open reader, and delete SPOOL file blocks (SFBLOKs), and the Reader Hash Table (plus extension pages), the saved ALOCBLOK and RECBLOK data, and the saved SPOOL hold queue blocks (SHQBLOKs). In addition, if this is an automatic system re-IPL following a dump to DASD, DMKWRMST reenables the terminals that were enabled when the system abended. Control passes back to DMKCPJ (which calls

DMKIDU to reinitialize the paging function). Control then goes to DMKWRMSY to reconstruct the closed reader SFBLOKs (in SYSSPOOL's virtual storage) and the SPUBLOKs (in SYSSPOOL's virtual storage). DMKWRM then calls DMKWRNWM to initialize the checkpoint area cylinders and also to checkpoint any SHQBLOKs or open SFBLOKs. Finally DMKWRMST invalidates the first record of the warm start area so that another warm start cannot occur until the system undergoes an orderly shutdown.

Checkpoint Start

If the system is unable to perform a warm start because of I/O errors or because no warm start data was saved from the last session, then a checkpoint (CKPT) start can be attempted. This option attempts to initialize the system using the information that has been dynamically checkpointed during system operation and stored on the checkpoint area. DMKWRMST is called by DMKCPJ for a CKPT start. DMKWRM calls DMKCKVWM to handle the CKPT start. DMKCKVWM restores the checkpointed real device information for system printers, punches, and readers, and reconstructs the SPOOL hold queue blocks (SHQBLOKs). DMKCKVWM also reconstructs the allocation maps for the SPOOL file blocks (SFBLOKs) which were dynamically checkpointed during system operation. This is done via a multitasking approach, in which a task is built to allocate each spool file. DMKWRMST then invalidates the first record of the warm start area so that a warm start cannot occur until the system undergoes an orderly shutdown. DMKCKRSY rebuilds SYSSPOOL's virtual storage and restores all the spool file chains (printer, punch, and reader).

The system log message is not reconstructed as for a WARM start. Also, since ALOCBLOK and RECBLOK information is not saved on the checkpoint area, the pages used by the reconstructed SPOOL files must be allocated by following the SPOOL file links (SPLINKs) for each reconstructed SPOOL file. Because of this, a checkpoint start takes longer than a warm start. For a CKPT start, DMKCKVWM loads a disabled wait state PSW (code 00E) if an invalid checkpoint record is encountered.

Force Start

If the system is unable to perform a checkpoint start because of I/O errors or invalid data on the checkpoint area, then a FORCE start can be attempted. A FORCE start works in the same way as a CKPT start, except DMKCKVWM truncates or erases any data that cannot be read.

Cold Start Overview

A COLD start is usually performed only on the initial loading operation of a new version of the VM/SP HPO system or when a hardware error prevented valid system checkpointing and shutdown. DMKWRMST is called by DMKCPJ during system initialization to perform a COLD start. No data is recovered from the checkpoint or warm start area. DMKWRMST calls DMKCKSIN to initialize the checkpoint area. Next DMKWRMST invalidates the first record of the warm start area so that a warm start cannot occur until the system undergoes an orderly shutdown.

Shutdown Start

A SHUTDOWN start is performed in order to halt the initialization process. For example, if vital CP owned volumes are not mounted, then it might be desirable to mount those devices and then re-IPL without first doing a WARM, CHECKPOINT, FORCE or COLD start. For a SHUTDOWN start no processing of system warm start or checkpoint data is done. A disabled wait state PSW (code 006) is loaded.

System Shutdown

When the operator or other authorized user issues the SHUTDOWN command without the REIPL option, DMKCPSSH moves "CPCP" into the CPID field of the PSA. "CPCP" tells the checkpoint program that a system shutdown is to be performed.

When the operator or other authorized user issues the SHUTDOWN command with the REIPL option, DMKCPSSH moves "WARM" into the CPID field of the PSA. "WARM" tells the checkpoint program that a system shutdown is to be performed and then an automatic warm start is to be performed.

DMKCPSSH then goes to DMKDMPRS. DMKDMPRS issues an IPL CCW to read the IPL sequence and, subsequently, DMKCKP from the SYSRES just as during system initialization. DMKDMPRS then loads the initial PSW from location 0 of main storage and DMKCKPT gets control.

DMKCKPT reads in the rest of the checkpoint programs (DMKCKD, DMKCKF, DMKCKH, DMKCKM, DMKERP, DMKCKW, and DMKCKN). DMKCKPT calls DMKCKDEV to drain pending I/O interrupts and issue an HIO to all available devices. DMKCKPT calls DMKCKFIL to save the addresses of enabled terminals, the status of the system operator, device and user accounting cards, the system log message, printer, punch, open reader, and delete SPOOL file blocks (SFBLOKs). DMKCKFIL saves this information on the warm start area for recovery by DMKWRM during a warm start. DMKCKWSP is called to save the closed reader SFBLOKs and SPUBLOKs (in SYSSPOOL's virtual storage) and DMKCKWHT is called to save the Reader Hash Table (plus extension pages), open CPTRAP and monitor SPOOL files, the allocation RECBLOKs, and the SPOOL hold queue blocks (SHQBLOKs). DMKCKP calls DMKCKMSV to save virtual machines which were enabled for VMSAVE. DMKCKMSV saves the virtual machines on the CP owned DASD as specified in the system name table, DMKSNT. If CPID = "CPCP" (i.e. SHUTDOWN was requested), then DMKCKPT loads a disabled wait state PSW (code 008) and the system shutdown is complete. If CPID = "WARM" (i.e. SHUTDOWN REIPL was requested), then DMKCKPT will load DMKSAV, set CPID to "WARM" and transfer control to DMKSAVRS in the same way that a normal system initialization is performed. DMKCPINT will get control from DMKSAV and perform the same functions as for normal system initialization, but since CPID is "WARM", the operator will not be requested to change the time of day clock, nor choose the type of start to perform. In fact, it is not necessary for an operator to be present during this automatic re-IPL. The system will automatically perform a WARM start (see Figure 5 on page 27).

Dumping the System and Automatic Re-IPL

When a system abend occurs or when the system restarts, the module DMKDMPDK dumps all of main storage or just the CP portion of main storage (plus SYSSPOOL's virtual pages that are resident at the time of the dump) to the indicated dump device. After the dump is completed, DMKDMPRS issues an IPL CCW to read the IPL sequence and, subsequently, DMKCKP from the SYSRES just as during system shutdown. DMKDMPRS then loads the initial PSW from location 0 of main storage and DMKCKPT gets control. If the dump was to a printer or to a tape, then DMKDMP leaves CPID set to "CPCP" and DMKCKPT will conduct an orderly system shutdown just as if the SHUTDOWN command had been issued. If the dump was to a DASD, then DMKDMP sets CPID to "WARM" before loading DMKCKP. DMKCKPT will still perform an orderly system shutdown and then an automatic warm start will be performed just as if the SHUTDOWN REIPL command had been issued.

Initialization and Termination

Attaching a Virtual Machine to the System

Note: The process described here ignores VCNA SNA terminal attachment support.

After CP has been initialized, DMKCPVEN enables the communication lines in response to the ENABLE command. Then an individual virtual machine is attached to the system, using the following steps:

1. *Terminal Identification*

   When the CP receives the initial interrupt from a terminal on an enabled line (normally initiated by a user dialing in on a data-set), the DMKCNSIN routine is entered. DMKCNSIN determines the terminal device type, stores this information in the terminal device block, writes the online message and puts the terminal line in a state to receive an attention interruption.

2. *Attention from User*

   After the online message has been displayed at the user's terminal, and he has pressed the ATTENTION key, DMKCNSIN (the console interruption routine) calls DMKBLDVM to build a skeleton VMBLOK for the user. At this time, the userid is LOGONxxx, where xxx is the terminal real device address, and a flag is set to indicate that the user has not yet completed the logon process.

   Then DMKCNSIN calls DMKCFMBK, which types a single blank at the terminal, and issues a read to the terminal for the user to enter his first command (normally LOGON or DIAL).

3. *First Command from User*

After the first command has been entered by the user, DMKCNSIN further determines the type of terminal. If the terminal is a 2741, DMKTRMID is called to identify it as either a 2741P (PTTC/EBCD) or a 2741C (Correspondence) terminal. If successful, the correct device type and translate tables for input and output are set. If not, flags are set to indicate that the terminal is not yet identified.

Then control is returned to DMKCFMBK, which determines if the first command is valid (for example, LOGON, MSG, or DIAL). If the first command is not valid, a restart message is given, and the read to the terminal occurs again for the first command. If the first command was LOGON (or its abbreviation), DMKLOGON is called to complete the process of attaching the virtual machine to the system.

The operations performed by DMKLOGON include the following:

- Obtains the userid from the command line, and checks for a possible password and other optional operands.
- Checks the userid and password against entries in CP's directory of users.
- Ensures that the user is not logged on at another terminal (an error condition), or reconnects the user if he was running in disconnect mode.
- Obtains pertinent information on the user's virtual machine from the user machine block portion of the directory.
- Stores the correct userid (replacing the LOGONxxx userid used until now), virtual storage size, and other vital information in the virtual machine's VMBLOK.
- Allocates and initializes segment, page, and swap tables (necessary for handling of the virtual machine's virtual storage).
- Obtains free storage for the virtual machine's swap control block (SCBLOK).
- Schedules MSS volume mounts for any required MSS volumes if the MSS is available and the volume is not already mounted.
- Allocates an extended VMBLOK (ECBLOK) if the user's virtual machine has the ability to run in the extended control mode.
- Allocates and initializes virtual device blocks, control unit blocks, and channel blocks, using information from the user device blocks portion of the directory.
- Establishes links (as feasible) to all DASD included in the directory, the accessibility of any disk being determined by the user access mode in the directory, and whether any other users are presently linked to the disk, in read mode and/or write mode.
- Initializes all other virtual device blocks as appropriate, such as reader, punch, printer, and terminal.

- Maps all virtual devices to real devices.
- Performs appropriate accounting.
- Informs the user of the date and time of the most recent revision to the system log message (LOGMSG), and of the presence of any outstanding spooled files in his virtual reader, printer, or punch.
- Sends a ready message to the user with the date and time (and weekday), and a message to the system operator indicating that the user has logged on.

If the virtual machine has a device address or a named system in the directory and the initialization was not suppressed via an option on the LOGON command line, then that device or named system is then loaded (via IPL) at the conclusion of the logon process. Otherwise, when the logon functions are complete, the user's terminal is placed in CP read mode ready for the entry of his first desired command.

Under the latter condition of no automatic IPL, the user can IPL an alternate nucleus by using the STOP option in the IPL command. This option causes the normal IPL procedure to halt execution prior to loading the initial PSW, and issues a DIAGNOSE code 8 that places the user's terminal in CP read mode. A hexadecimal character entered in location X'08' changes the nucleus name. A hexadecimal character entered in location X'09' changes the apparent storage size. The BEGIN command allows the IPL procedure to continue.

System Reconfiguration

308x Processor Complex

The 308x Processor Complex uses the 3082 Processor Controller to coordinate central communications for the processor complex. The monitoring and service support facility (MSSF) is a hardware component of the processor controller. MSSF supplies I/O configuration and storage information for the 308x Processor Complex and executes commands that modify the real system configuration.

CP uses the MSSFCALL instruction to communicate with the MSSF. MSSFCALL is a diagnose instruction with a function code of X'80'. A command request to the MSSF requires a hardware call control block (HCBLOK). CP modules use the HCBLOK to issue an MSSFCALL instruction. The HCBLOK is a 48 byte storage area that is used to order requests to the MSSF and provide status of the requests to the caller.

After executing a command, MSSF uses the MSSF data block (MSFBLOK) to return information to the requester. MSFBLOK is a 2K byte storage area that MSSF uses to return header information and other command dependent information. The maximum length of the MSFBLOK is 2K. However, CP always obtains a page of storage to ensure that the control block is on a 2K storage boundary. The page of storage is locked to prevent page faults during processing of the request.

MSSF provides support for VARY PROCESSOR and SCPINFO commands. If the VLOG option of the VARY OFFLINE PROC command is used, the processor is logically switched offline—MSSF is not called for this process. For real MSSF processing, the SCPINFO command must originate from a V=R virtual machine.

Note: *If the preferred machine assist guest issues a command to vary the channel offline, the hardware receives the instruction directly and the channel is varied offline. VM has no way to intercept the instruction when the preferred machine assist guest is running.*

3090 Processor Complex

The 3090 Processor Complex uses the Service Processor to obtain I/O configuration and storage information and to execute commands that modify the real system configuration. VM/SP HPO uses the Service Call together with the Service Call Control Block and the HCBLOK when invoking the Service Processor. The interface for MSSF support also processes Service Call requests, (they are mutually exclusive).

Real MSSF or Service Call Processing – VARY PROCESSOR

Module DMKCPU receives control to execute a VARY PROC command. If the VARY PROCESSOR command is issued on a 308x or 3090 processor (CPUID = 308x or 3090 in PSA), DMKCPU calls DMKPTR to obtain a page of storage for the HCBLOK and the MSFBLOK or SCCBLOK—the MSFBLOK or SCCBLOK uses the first 2K of storage while the HCBLOK uses the remaining 2K. DMKCPU sets the HCBLOK fields to contain the address of the MSFBLOK or SCCBLOK, the command, and the return address for MSSF or Service Call interruptions. DMKCPU passes control to DMKMHC (at entry point MDKMHCCP) to schedule the MSSF or Service Call request.

Note: *When VM/SP HPO is running second level as a virtual MP guest on VM/XA, MSSF and Service calls are bypassed.*

Real MSSF or Service Call Processing – SCPINFO and IOCP

Module DMKHVC receives control to validate the diagnose instruction produced by an IOCP or SCPINFO command. DMKHVC calls DMKMHV to process the diagnose instruction. DMKPRV validates the Service Call instruction (SERVC) and calls DMKMHV to simulate the privileged operation. DMKMHV examines the MSSF external interruption pending field (VMMSSFXP) in the virtual machine's VMBLOK to determine whether the virtual machine has an outstanding request. If an MSSF or Service Call external interruption is pending, the virtual machine's PSW (VMPSW) is set to condition code two (busy), and control returns to the caller. If a request is not pending, DMKMHV does the following:

- Sets VMPSW to indicate condition code zero
- Sets VMMSSFXP in the VMBLOK for the virtual machine

 LY20-0897-7

- Builds an HCBLOK
- Sets the HCVMEEQ field to indicate a virtual machine generated MSSFCALL or Service Call operation
- Passes control to DMKMHC (at entry point DMKMHCVM) to schedule the request.

If DMKMHV is processing the IOCP command, before passing control to DMKMHC, DMKMHV verifies that the processor is a 308x or 3090 and examines the virtual machine user's privilege class. The virtual machine user must have privilege class C or E to read from an input/output configuration data set (IOCDS) and privilege class C in order to write to an IOCDS. If the user does not have the appropriate privilege class for a MSSFCALL, DMKMHV sets an invalid command response code and returns control to DMKHVC. If the user does not have the appropriate class for a Service Call request, an invalid function code is set by DMKMHV and control returns to DMKPRV.

For more information on the IOCP command, see the *VM/SP HPO Planning Guide and Reference* and the *IOCP User's Guide and Reference* for either the 308x Processor Complex (GC28-1027) or the 3090 Processor Complex (GC38-0039).

For an IOCP write request, DMKMHV limits access to an IOCDS to one user at a time. DMKMHV obtains a lock to serialize access to an IOCDS. Because DMKMHV is pageable, the IOCP write lock (ICPWLOK) is defined in module DMKMHC at entry point DMKMHCLK. DMKMHV uses an external reference to DMKMHC to examine the lock.

- If the lock is free, DMKMHV ensures that the IOCDS is open and sets the lock for the virtual machine user.
- If the lock byte is not set up properly, an error response code is returned.
- If the lock is held, DMKMHV determines whether the lock is held by this user. If the lock is not held by this user, DMKMHV sets condition code two (busy) and returns control to the caller.
- If DMKMHV determines that the lock is held by this virtual machine user, DMKMHV prepares to perform a real MSSF or Service Call open for the IOCDS.

After obtaining the lock, DMKMHV builds an HCBLOK to contain the address of the data block and command word, and calls DMKMHC to schedule the request. The lock is not needed to process an IOCP read request. DMKMHCVM is the entry point for MSSFCALL or Service Call instructions generated by a virtual machine.

Scheduling and Executing the Real MSSF or Service Call Request

Module DMKMHC schedules all CP and virtual machine requests for MSSF or Service Call operations. DMKMHC contains several entry points. DMKMHCCP is the entry point for CP generated MSSFCALL or Service Call operations while DMKMHCVM is the entry point for virtual machine MSSFCALL or Service Call requests. Though the entry points are different for the operations, processing is similar. On entry, DMKMHC obtains the address of the HCBLOK and performs the following functions:

- Verifies that the processor-id is that of a 308x or 3090 processor.
- Sets the HCBLOK pointer, flag bytes, and priority fields to zeros.
- Sets the HCMSFSYS field to indicate a CP or virtual machine generated MSSFCALL or Service Call operation.
- Saves the system VMBLOK address in the HCUSER field of the HCBLOK.
- Adds the address of the HCBLOK to the internal control block pointer chain (HCANCHOR).
- Examines the active pointer (HCACTIVE) to determine whether MSSF or Service Processor is busy with a previous request.

HCACTIVE contains the address of the HCBLOK that MSSF or Service Processor is currently processing. The HCACTIVE field is zero if the MSSF or Service Processor is available. If the MSSF or Service Processor is available, DMKMHC passes control to subroutine MSFCALL (using BALR) to issue the diagnose instruction for a 308x processor or the Service Call instruction for a 3090 processor. Subroutine MSFCALL performs the following:

- Obtains the address of the HCBLOK from the queue
- Issues the MSSFCALL for a 308x processor or issues the Service Call for a 3090 processor
- Traces the or Service Call (if internal tracing is active)
- Validates the condition code returned by MSSF or the Service Processor.

If MSSF or the Service Processor returns condition code zero, the address of the HCBLOK is placed in HCACTIVE and HCGLAG is set to indicate that MSSF is processing the diagnose instruction or the Service Processor is processing the Service Call instruction. The HCBLOK is deleted from the queue of pending requests and control is returned to the caller.

The MSSFCALL or Service Call instruction and response are traced if the internal trace facility is active. For further information on the format of the trace table entries see the *Virtual Machine Diagnosis Guide*.

Processing the MSSF or Service Processor Interrupt

MSSF and the Service Processor use an external interruption to signal CP that processing is complete. DMKEXT passes control to DMKMHC. DNMMHC processes external interruptions resulting from the MSSFCALL or Service Call at entry point DMKMHCIN.

At entry point DMKMHCIN processing is as follows:

- Obtains the global system lock
- Verifies that the interruption from DMKPSA is associated with the HCBLOK address in the HCACTIVE field
- Examines the interruption to determine whether CP or a virtual machine initiated the request.

For a CP request, DMKMHC obtains a CPEXBLOK and initializes it to contain the interruption return address (DMKCPUMI) and the address of the MSFBLOK or SCCBLOK. DMKMHC places the CPEXBLOK on the dispatcher queue and control returns to the caller, which examines the completion code and issues appropriate messages.

For a virtual machine request (HCVMREQ=1), processing is as follows:

- Resets the interruption pending field (VMMSSFXP) in the virtual machine's VMBLOK
- Resets information on Service Call facilities, Paging Storage for the 3090, and storage information for both the 3090 and 308x
- Obtains an XINTBLOK for the virtual machine
- Determines whether the virtual machine is logging off or performing a system reset
- Releases the HCBLOK
- Places the XINTBLOK on the virtual machine's external interruption queue
- Exits to the dispatcher to return control to the caller.

DMKMHC compares the HCACTIVE field with the address of the data block. If the addresses do not match, ABEND MHC02 is generated. If they match, DMKMHC uses zeros to clear HCACTIVE and examines the queue or pending MSSF requests. If another request is in the queue, DMKMHC sets HCACTIVE to the addresss of the new HCBLOK, and the new request is executed.

When a virtual machine logs off or performs a system reset, DMKCFP receives control. DMKCFP (at entry point DMKCFPRR), calls DMKMHC to determine whether the virtual machine has an outstanding MSSF request. At entry point DMKMHCRE, processing is as follows:

- Clears VMMSSFXP in the virtual machine's VMBLOK
- Examines the queue of pending requests
- Deletes the HCBLOK from the queue if the request is not active
- Releases storage for the MSFBLOK or the SCCBLOK and the HCBLOK
- Examines the IOCP write lock
- Clears the lock if it sets for this virtual machine
- Returns control to DMKCFP through the dispatcher.

If the virtual machine has an active operation, DMKMHC waits for the external interruption before releasing the MSFBLOK or SCCBLOK and HCBLOK.

Virtual MSSF or Service Call Processing – SCPINFO

On a 308x processor, module DMKHVC receives control to validate the diagnose instruction produced by a virtual machine SCPINFO command. DMKHVC calls DMKMHV to process the diagnose instruction. DMKMHV simulates the MSSFCALL instruction by setting the virtual machine's PSW to contain a condition code and scheduling external interruptions in the same sequence as would be presented on the real processor.

On a 3090 processor, module DMKPRV receives control to validate the Service Call instruction. DMKPRV calls DMKMHV to simulate a virtual Service Call.

DMKMHV examines the XINTBLOKs for the virtual machine to determine whether the virtual machine has an outstanding request. If an external interruption is pending, the virtual machine's PSW (VMPSW) is set to condition code two, and control returns to the caller. If there is no request pending for the virtual machine, VMPSW is set to condition code one. DMKMHV then verifies the MSFBLOK or SCCBLOK. If there are no violations, the MSFBLOK or SCCBLOK is set up to contain predefined response codes.

After setting the MSFBLOK or SCCBLOK, DMKMHV obtains an XINTBLOK for the virtual machine and sets it to the address of the MSFBLOK or SCCBLOK. The XINTBLOK is placed on the virtual machine's external interruption queue and control returns to the caller through the dispatcher.

If an error condition exists for violation of boundary, block length, or command, the data block response field is set to the appropriate error response code. Control returns to DMKHVC or DMKPRV.

Figure 31 shows a real MSSFCALL control block structure. Figure 32 shows a virtual MSSFCALL control block structure.

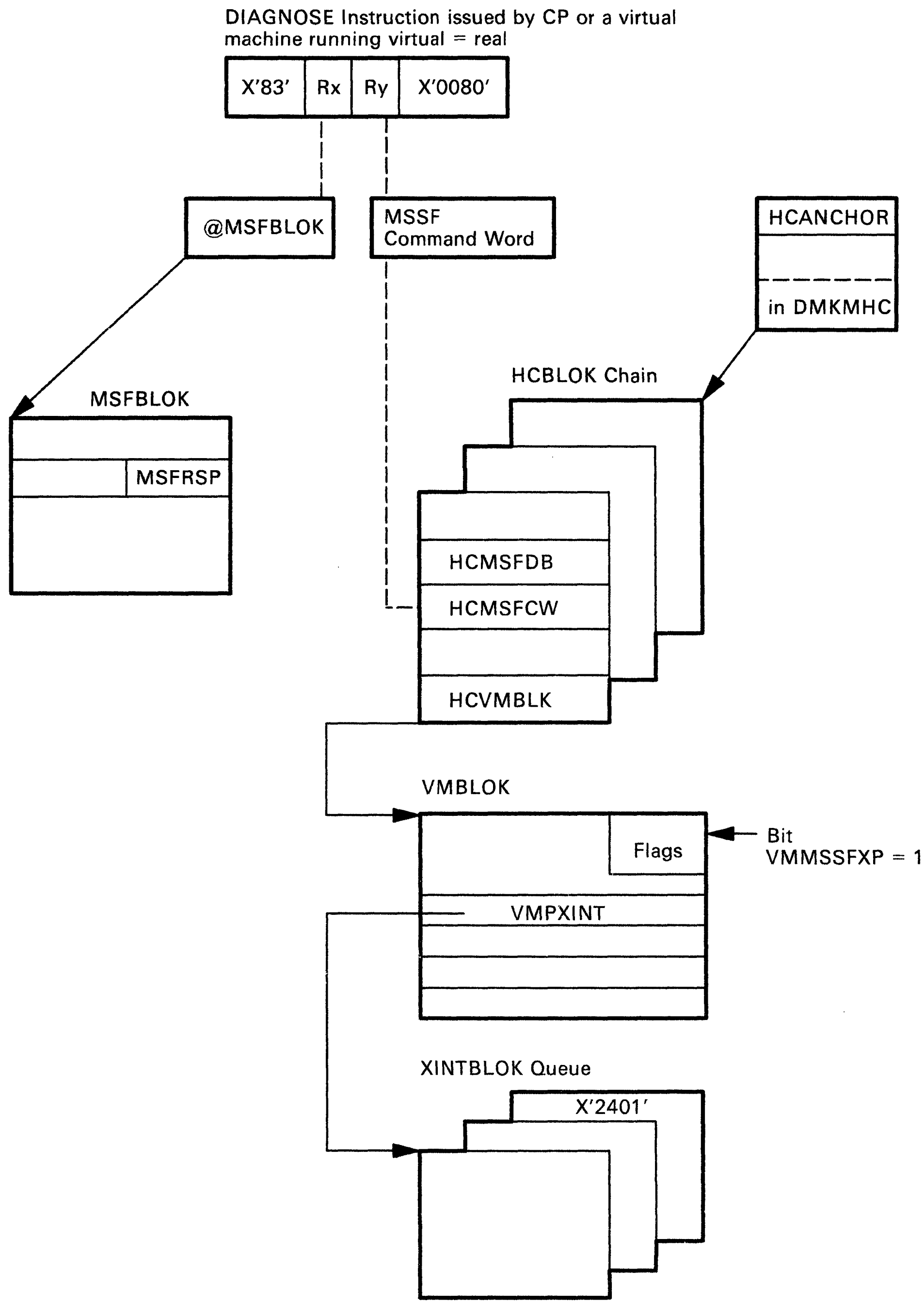


Figure 31. Real MSSFCALL Control Block Structure

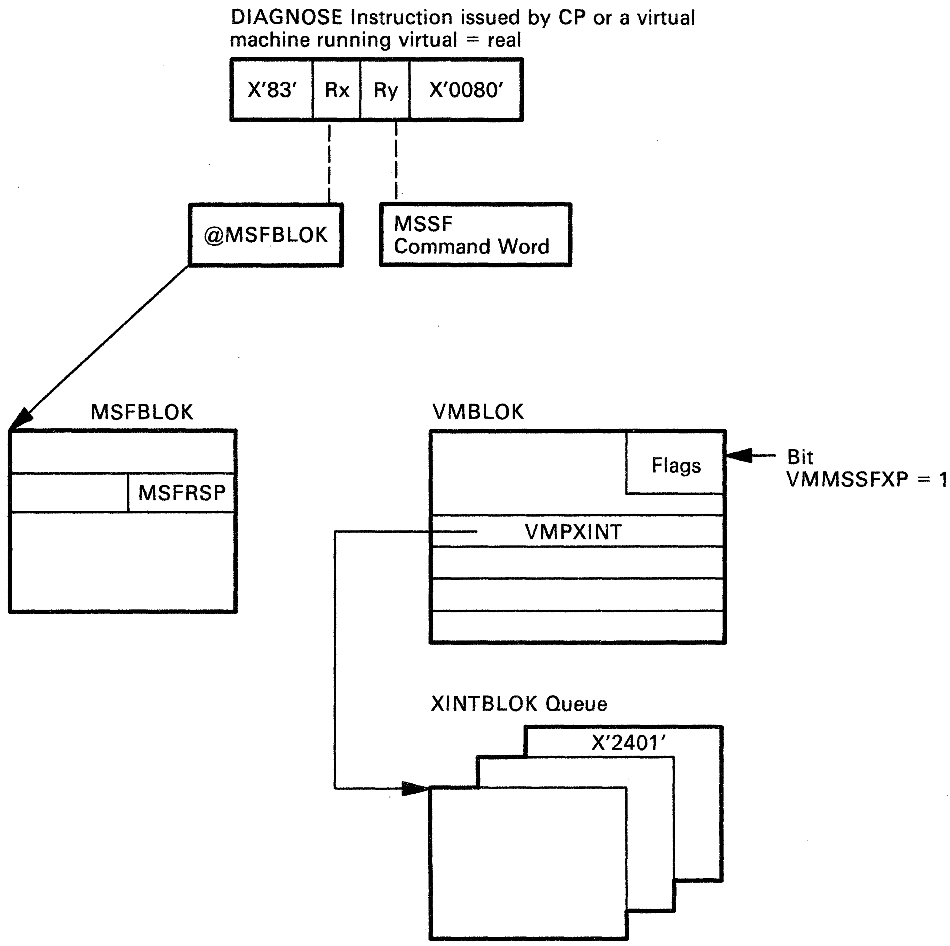


Figure 32. Virtual MSSFCALL Control Block Structure

I/O Reconfiguration

Three commands alter the I/O configuration of a user's virtual machine after he has logged on. Two are user commands, while the third is a system operator command, because it affects the status of real devices attached to the system. The ATTACH and DETACH commands are contained in DMKVDA, DMKVDC, DMKVDD, and DMKVDE and the DEFINE command in DMKDEF. The system command scanner (DMKCFM) calls both pageable modules after their format and privilege classes have been validated. These commands access the same control-block building subroutines in the module DMKVDS that DMKLOG, the LOGON processor, uses.

Note: *The authorized user will be able to dynamically redefine the virtual machine's device configuration during the terminal session (using the ATTACH, DETACH, and DEFINE commands), but only if there is a sufficient amount of available contiguous space. See "Virtual Machine I/O Management" on page 8 to determine the maximum device limit.*

Attaching a Real Device: The system operator can dedicate any real device to a single virtual machine by issuing the ATTACH command. The device attached is available only to the given virtual machine, and all I/O requests to it are handled by CCW translation. If the device is a DASD, cylinder relocation does not occur when SEEK addresses or home addresses are referenced. The I/O supervisor does not queue operations on the device, nor does it automatically restart it or do ordered seek queuing. Nonsharable devices such as tape drives must be attached to a virtual machine to be accessed by the virtual machine. A virtual machine can also have a dedicated card reader/punch or printer. However, this is usually not necessary because of the unit record spooling facilities of CP. Unit record input or output on a dedicated (attached) device is not spooled by CP. The unit attached may be given a virtual address different from its real address. However, the virtual machine may not already have a virtual device at the attached address. A real device cannot be attached (1) if it is currently dedicated to another virtual machine, (2) if it contains minidisks that are in use by other virtual machines, or (3) if it is a system-owned volume that is in use for spooling or paging.

The system operator can dedicate a remote 3270 Information Display System Printer (3284, 3286, 3287, 3288, or 3289) to a single virtual machine by issuing the NETWORK ATTACH command. The printer attached is available only to the given virtual machine, and all I/O requests to it are handled by Start I/O.

Defining a Virtual Device: A system user can define a new virtual device with the DEFINE command that does not require the dedication of a corresponding real device. Devices that can be defined are consoles, spooled readers, punches and printers, dialable TP lines, virtual channel-to-channel devices, pseudo timers, and temporary disks. With the DEFINE command, the user can change any existing virtual device address whether it corresponds to a shared or dedicated real device or no real device unit.

Care must be taken when using the DEFINE statement to create (or move) a virtual device. CP checks for a potential subchannel protocol conflict on the virtual control unit that would support the new device. If a conflict is detected, the operation is not performed. Instead, CP sends an error message to the user who issued the command.

The DEFINE command can also describe the virtual machine channel mode of operation, that is, either selector or block multiplexer. The default mode, selector channel mode, reflects a channel busy to any SIO operation attempted on the same channel path that has not completed the previous channel SIO operation. Block multiplexer mode allows the successful initiation of different devices on the same channel path. Channel 0, a byte-multiplexer channel, is unaffected by the DEFINE command. Use of the DEFINE command with the CHANNELS operand generates a virtual machine reset. Therefore, it should be invoked prior to the virtual machine IPL operation.

Note: The channel mode selected has no bearing on the types of channels that are attached to the real system.

Temporary disks are dynamically obtained cylinders or blocks of DASD storage space. They are available to the user for as long as they are part of his virtual machine configuration, but the data on them is destroyed after the user detaches the area. For all other purposes, however, they appear to be a standard disk.

Detaching a Virtual Device: A virtual device can be removed from a virtual machine configuration prior to logging off with the DETACH command. A user can detach any of his own devices, and the system operator can detach a real device from a virtual machine. If the operator detaches the device, the user is informed of the operator's action. A real device can be detached only if it is dedicated to a single virtual machine or is attached to the system and is not in use when the DETACH is issued.

The system operator can detach a remote 3270 Information Display System Printer (3284, 3286, 3287, 3288, or 3289) from a virtual machine with the NETWORK DETACH command. If the operator detaches a 3270 Information Display System Printer, the user is informed of the operator's action. A 3270 Information Display System Printer can be detached only if it is dedicated to a single virtual machine or is attached to the system and is not in use when the NETWORK DETACH command is issued.

Disconnecting a Terminal or Virtual Machine

A user may permanently or temporarily disconnect his terminal or virtual machine from the system by a console command, or the terminal or virtual machine may be forcibly disconnected by the operator. The system can also log off the virtual machine. Reconfiguration routines that handle the termination process are in the pageable module, DMKUSO.

Permanent Disconnect: The user may voluntarily remove his virtual machine from the system via the LOGOFF command. This command terminates all virtual machine operation, releases all storage occupied by control blocks and virtual storage pages, and disconnects the teleprocessing line connection to the user's terminal. If the user specifies the HOLD option with LOGOFF, all of the above occurs, except that the teleprocessing line remains enabled. This option is especially useful for dialed connections that are reused immediately by another user.

The virtual machine can be forced off the system by the system operator via the FORCE command. This has the same effect as a user-initiated logoff, except that the user is informed that the operator has logged off his machine. A virtual machine may also be logged off the system if the virtual machine is running disconnected (without an active terminal) and the virtual machine attempts a terminal read or enters a disabled wait state.

The DMKUSOLG and DMKUSOFF subroutines process the LOGOFF command. DMKDSP calls DMKUSOFF to force the logoff of a virtual machine.

Temporary Disconnect: A user may temporarily disconnect his terminal from his virtual machine by using the DISCONN command, while allowing the virtual machine to continue to run. This command flags the virtual machine as being disconnected and releases the user's terminal and teleprocessing line. If the HOLD option was specified in the DISCONN command, CP allows the line to remain enabled, and another user can use the terminal to log on. The disconnected virtual machine continues to be dispatched until it either attempts to execute a terminal read to the disconnected console or it enters a disabled wait state. At this time, the dispatcher (DMKDSP) calls the routine DMKUSOFF directly to force the machine out of the system. While the machine is disconnected from its virtual console (real terminal) any terminal output is lost. In addition, CP may apply a disconnected penalty to the machines scheduling priority, to bias the system in favor of interactive users.

A user's virtual machine may also be disconnected by the system. If the disconnected user logs on to the system while the disconnected machine is still running, it is reconnected and can continue to interact with the system in the usual manner.

The DMKUSO subroutine processes the DISCONN command.

Console Functions

DMKCFM analyzes CP commands and passes control to the appropriate routine to handle the command. DMKCFM can be entered by the Attention key (or equivalent) at the user's terminal or directly from a virtual machine.

When a console interruption occurs by the Attention key at the user's terminal, DMKIOSIN calls DMKCNSIN to handle the unsolicited interruption, then DMKCNSIN calls DMKCFMBK.

DMKCFMBK first calls DMKFREE to obtain storage for an 18-doubleword input buffer. Next, DMKQCNWT is called to send the CP message to the terminal to inform the user that he has entered console function mode. DMKQCORD is then called to read the command line entered at the console.

DMKCFMEN is the entry point for commands coming directly from the virtual machine. DMKPRGIN enters at DMKCFMEN here when a DIAGNOSE instruction with a code of 8 is detected. The address of an 18-doubleword input buffer is passed in register 1. Therefore, a read to the terminal is not needed.

After either the read to the terminal or entry from the virtual machine, DMKSCNFD is called to find the command type. On return from DMKSCNFD, register 1 points to the start of the command and register 0 contains the length of the command. DMKCFM then calls DMKCFCMD to scan and validate the command. DMKCFCMD contains a command table which it searches until it finds a command name match. It compares the user classes in the field CTCLASS. If the user is authorized to issue the command, DMKCFCMD will pass the appropriate module entry point from the command table to DMKCFM. The command is then processed. For the commands ATTACH, DETACH, INDICATE, NETWORK, QUERY, and SET, DMKCFCMD uses subcommand tables in DMKCMD to locate the appropriate module entry points.

If DMKCFCMD cannot find a command name match, the user receives a message informing him that the command or subcommand table search has failed.

After the command has been processed, control is returned to DMKCFM. There are three possible returns. (1) On a normal return, the input buffer is scanned to see if there are any more commands. If none exist, DMKCFM returns to the virtual machine (if entered via DIAGNOSE) or calls DMKQCNRD to read the next command from the terminal. (2) On a return plus 4, the VMCFWAIT bit is turned off to allow the virtual machine to run. DMKFRET is called to return the input buffer storage. Then control returns to either the virtual machine, if entered via a DIAGNOSE, or to DMKDSPCH if entered via the Attention key. (3) On a return plus 8, the operation is the same as plus 4 except that the VMCFWAIT bit is left on.

Dispatching and Scheduling

The dispatcher, DMKDSP, gives the processor resource to a virtual machine. It is called after an interrupt handler has finished processing, and after each stacked CPEXBLOK, I/O request, deferred input task, and external interrupt has been taken care of. The dispatcher updates the virtual timers and CPU timer of the virtual machine just serviced, and also reflects any pending interrupts for that virtual machine. Then the dispatcher either stacks the next available stacked request, or places the highest priority dispatchable user into execution.

The scheduler, DMKSCH, decides how often and for how long virtual machines will be given processor resources. The scheduler keeps statistics to make dispatching decisions, and maintains queues for runnable, dispatchable, and eligible users.

The auxiliary routine that assists the dispatcher and scheduler is the request stack maintenance routine, DMKSTK. Other auxiliary routines: DMKTRQ, which handles timer requests, and DMKSTP, which periodically recalculates feedback variables. (Also, DMKSTP uses data areas in DMKSTD.)

To make decisions on dispatching and scheduling, the control program places all virtual machines into various categories, and recognizes user machines as being in one of several states. The virtual machine categories either interactive or noninteractive virtual machine, are defined in the following way:

- An interactive virtual machine is one whose use of the system is punctuated by long idle waits, and does not have long processor execution times. A virtual machine becomes eligible to enter interactive status whenever it remains idle for more than 300 milliseconds. It remains interactive until it uses one Q1 time slice and one Q2 time slice. In addition, the scheduler identifies virtual machines that use CPU in frequent, short bursts. If the average length of these bursts is less than one fourth of the Q1 time slice, the scheduler forces that virtual machine into Q1 and treats it as if it were interactive.

- A non-interactive virtual machine is one that has violated an interactive criterion: it has used more than one Q1 time slice and one Q2 time slice without entering a long idle wait (remaining idle for more than 300 milliseconds.). CP schedules interactive users ahead of non-interactive users. Non-interactive users are subdivided into several classes. Normal non-interactive virtual machines are scheduled by a priority scheme described below. A virtual machine is allowed to execute for a specified time period and then it is placed in a list of those machines that are waiting.

Note: You can alter the queue scheme by using the NOQ2 or NOQ3 option on the SET QDROP OFF command. Specifying NOQ3 will force a virtual machine to be kept in Q1 or Q2. Specifying NOQ2 will force a virtual machine to be kept only in Q1. Furthermore, virtual machines are categorized on a sliding scale by their levels of resource utilization. The actual scheduling sequence depends on both a virtual machine's interactive/noninteractive characteristics and its current resource utilization level.

To give preference to certain classes of virtual machines, a priority scheduling scheme allows virtual machines to be scheduled with a priority class. The priority is a number assigned by the directory. However, the number may be altered by the system operator. This priority is referred to as the "user priority" to distinguish it from the priority used by the scheduler for ordering virtual machines for service.

Run List

The run list is a list of all virtual machines, runnable and nonrunnable, that are competing for processor resources. The scheduler places a virtual machine in the run list if its projected working set size is less than or equal to the number of page frames available for allocation, AND if processor time is available. If processor time is not available, the scheduler decides if the virtual machine should go on the run list anyway by checking to see if the virtual machine's deadline priority is earlier (better) than that of the last virtual machine already on the run list. If this is true, then the virtual machine is placed on the run list.

Note: The INDICATE POSITION and INDICATE QUEUES commands show you the specific list on which a user is waiting. If the user is on the eligible list, the response will show you that the user is either interactive or noninteractive, and that the user is waiting for either storage or processor time. See VM/SP HPO CP for System Programming for more information on these commands.

The interactive buffer is a pool of page frames which is set aside for the exclusive use of interactive virtual machines. The size of the interactive buffer can be set with the class E SET SRM IBUFF command. Pages in the interactive buffer can only be used by interactive virtual machines (Q1 or first Q2 time slice).

If space is not available for a virtual machine, the scheduler:

1. Checks the minimum MPL levels
2. Attempts to preempt virtual machines of lower priority
3. Blocks addition of this virtual machine and others of the same class to the run list.

A virtual machine that completes its time slice is dropped from queue and placed in the eligible list according to its deadline priority. A virtual machine in queue that enters a long idle wait is removed from queue if it remains idle for at least 300 milliseconds.

Dispatch List

The dispatch lists, or true run lists (TRLs), are subsets of the run list, anchored in the PSA extensions as doubly-linked lists of VMBLOKs of the virtual machines that are currently dispatchable. One dispatch list exists for each processor on an AP/MP system. The dispatcher scans the processor's dispatch list when it wants to give processor time to a user. The dispatch list is maintained by the scheduler, who considers a user to be dispatchable when the VMRUN bit in the VMDSTAT field of a VMBLOK is on. The VMRUN bit is a summary of the bits in the VMRSTAT field, each of which indicates various conditions that keep a runnable user from being dispatchable. If any bit in VMRSTAT is on, the VMRUN bit is off.

The scheduler maintains the dispatch list by adding and removing the VMBLOKs of users as their current status dictates. The four basic actions the scheduler must perform to do this are:

1. Adding a VMBLOK already in the run list to a dispatch list when the user becomes dispatchable
2. Adding a VMBLOK to a dispatch list when a user, who is currently dispatchable, is added to the run list
3. Removing a VMBLOK from a dispatch list when the user is no longer dispatchable
4. Removing a VMBLOK from a dispatch list when the user is dropped from the run list.

To add a VMBLOK to a dispatch list AND to the run list, the scheduler follows this procedure:

1. Call DMKLOK to lock the run list
2. Add user to the run list
3. Call DMKLOK to unlock the run list
4. Call DMKLOK to lock the dispatch list

5. Add user to the dispatch list by using the anchor point of the dispatch list (TRLANCHR) and the priority of the user (VMTPRIOR) to scan the dispatch list for the proper place to insert the new VMBLOK. The forward pointer (VMFTRL) of the new VMBLOK will be a nonzero value to indicate that the VMBLOK is in the dispatch list.

6. Call DMKLOK to unlock the dispatch list.

To add a VMBLOK to a dispatch list, when it is already in the run list, the scheduler follows this procedure:

1. Call DMKLOK to lock the dispatch list

2. Add user to the dispatch list as in step 5 above

3. Call DMKLOK to unlock the dispatch list.

To drop a VMBLOK from a dispatch list AND from the run list, the scheduler follows this procedure:

1. Call DMKLOK to lock the run list

2. Drop user from the run list

3. Call DMKLOK to unlock the run list

4. Call DMKLOK to lock the dispatch list

5. Drop user from the dispatch list by linking the VMBLOKs of the users that come before and after the VMBLOK that is about to be deleted. It does this by updating their backward and forward pointers (VMBTRL and VMFTRL). The scheduler then sets to zero the deleted VMBLOK's forward pointer, to mark it as no longer in the dispatch list.

6. Call DMKLOK to unlock the dispatch list.

To drop a VMBLOK from a dispatch list but NOT from the the run list, the scheduler follows this procedure:

1. Call DMKLOK to lock the dispatch list

2. Drop user from the dispatch list as in step 5 above

3. Call DMKLOK to unlock the dispatch list.

Occasionally a dispatch list maintenance problem will occur. The scheduler will realize this when it attempts to add a VMBLOK to a dispatch list, only to find that it is already there, or when it attempts to remove a VMBLOK from a dispatch list when the VMBLOK is not there. In either of these instances, CP will abend.

When the dispatcher wants to put a virtual machine into execution, it uses the anchor of the processor's dispatch list (TRLANCHR) and the dispatch list forward pointer (VMFTRL) to find the address of a VMBLOK reflecting a dispatchable user.

Note: *When a system is running in AP/MP mode, the dispatcher will call the scheduler as needed to update the dispatch list after a CPU timer interrupt. This occurs whenever the system lock is needed for the scheduler to update the current VMRSTAT.*

Dispatch Request Queues

The dispatch request queues contain pointers to CPEXBLOKs, IOBLOKs, and TRQBLOKs. CPEXBLOKs are control blocks that designate CP tasks to run. IOBLOKs contain information on imminent I/O operations, and TRQBLOKs are used to manage system timing facilities. The scheduler gives priority to these blocks over VMBLOKs.

For attached processor and multiprocessor systems, the scheduler maintains two dispatch request queues and two dispatch lists (one for each processor). When the scheduler adds a virtual machine to a dispatch list, the virtual machine is queued for the local processor unless:

- The other processor has fewer virtual machines on its dispatch list, or
- A virtual machine has affinity specified for the other processor, or
- A virtual machine's virtual Vector Facility status is in the real Vector Facility on the other processor.

It stays assigned to that processor until it is stolen, becomes nondispatchable, or drops from queue.

Active Wait

In MP systems work can become available without an accompanying interrupt. When one processor becomes idle, it goes into an active wait. During active wait, the otherwise idle processor scans certain queues looking for work. The first queue it checks is the local DSPRQ. If the dispatcher request queue is empty or the system lock is unavailable, the header of the local dispatch list is examined. If this queue is empty or a runnable user is not found, the dispatch list of the other processor is examined. If the idle processor finds more than one runnable virtual machine on that dispatch list, it dequeues the second virtual machine, adds it to its local dispatch list, and dispatches it. If it does not find more than one VMBLOK, it will return to the dispatcher (if trying to steal) or check the local dispatch list again.

Note: *When VM/SP HPO is running second level as a virtual MP guest on VM/XA SF, active wait will load an enabled wait PSW if it has no work on its queues. It will issue DIAGNOSE X'44' if it has work that cannot be unstacked. When work is stacked for a waiting processor, that processor will be signaled.*

Queue Drop Elimination

When a virtual machine is not dispatchable (a bit is on in VMRSTAT) and in queue (VMINQ is on), the scheduler must decide whether to drop the virtual machine from queue. When the virtual machine enters a potential long wait, the scheduler calls DMKFREXX for a TRQBLOK to create a 300-millisecond queue drop delay. The TRQBLOK for the queue drop delay is anchored in the field VMDRPTRQ in the VMBLOK.

After the scheduler invokes a 300-millisecond delay, it returns to the caller because it can do no more useful work until after the virtual machine is dropped from queue. If the virtual machine becomes runnable before the 300-millisecond delay is completed, the delay is canceled and the virtual machine continues in its current queue stay. If the delay expires, the virtual machine is dropped from the run list.

When the scheduler is called on behalf of a virtual machine, and the virtual machine is dispatchable, the scheduler ignores the call. This prevents a noninteractive virtual machine from periodically becoming interactive and hindering response times for truly interactive machines.

Virtual Machine Run Lists and States

To efficiently manage the large inventory of potential virtual machines that are logged on to the system, CP defines several states that a virtual machine may occupy. The scheduler can move a virtual machine from one state to another. However, a virtual machine may exist in only one state at any given instant. CP can then make scheduling and dispatching decisions by looking only at the subset of virtual machines that are in the appropriate state. To do this search, it also maintains lists of virtual machines in certain executable states.

A user's virtual machine may be in one of the following states:

State	Meaning
1	Interactive and dispatchable (in run list, in dispatch list)
2	Interactive and not dispatchable (in run list, not in dispatch list)
3a	Interactive and runnable, but no available real storage (waiting, in eligible list)
3b	Interactive and runnable, but no available real storage or processor time (waiting, in eligible list)
4	In wait state with queue-drop delay timer active (in run list)
5	Noninteractive and dispatchable (in run list, in dispatch list)
6	Noninteractive and not dispatchable (in run list, not in dispatch list)
7a	Noninteractive and runnable but no available real storage (waiting, in eligible list)
7b	Noninteractive and runnable but no available real storage or processor time (waiting, in eligible list)
8	In long idle wait (idle for more than 300 ms.)

Entries on the run list and the dispatch list are the VMBLOKs for those virtual machines in states 1 and 5, and represent the virtual machines that can be run at any given time. (States 2, 4 and 6 remain in the run list even though they are not dispatchable.) The dispatch list is sorted by deadline priority, which is the time of day (TOD) by which the virtual machine should have used its allocated processor time and been dropped from queue. A task is defined as one processor time slice or execution between long idle waits (movement from state 8 to state 1), whichever is less. It is re-projected for a virtual machine each time it is dropped from a queue.

The deadline priority is recalculated each time a virtual machine is dropped from queue and represents the next expected time that the virtual machine should again be dropped from queue.

The eligible list contains virtual machines in states 3 and 7. These virtual machines are potentially executable, but they are not allowed to compete for the processor because of the current demand for main (real) storage. The list is ordered by deadline priority, which is determined at the time the virtual machine was previously dropped from queue, as follows:

1. Paging ratio - The virtual machine's projected working set size, calculated the last time it was dropped from a queue, is expressed as a ratio to the overall system average working set size.

2. Processor resource ratio - The virtual machine's processor resource utilization the last time it was dropped from a queue, is expressed as a ratio to the overall system average processor resource utilization.

3. Average queue delay - The sum of the average time spent in Q1, the average time in E1, the average time in Q2, and the average time in E2.

4. Paging bias - The total eligible list time divided by the total list time (sum of the total eligible and dispatch list times) for all virtual machines.

5. User bias ratio - A user bias ratio is formed by multiplying the paging ratio by the paging bias and adding that to the result of multiplying the processor resource ratio by 100 minus the paging bias. The resulting sum is divided by 100 to arrive at a user bias. In effect, this is a weighted average of the paging and processor resource ratios, where the weights are determined by the paging bias. The higher (lower) the real storage contention, the larger (smaller) the weight given to the paging ratio.

6. Prioritized average queue delay - The average queue delay, which is the sum of the average times a virtual machine spends on the eligible list and on the dispatch list, is multiplied by the function (64**(UP/64))/64, where UP is the user priority. For those priorities equal to 64, the function is 1, for those greater than 64, the function is greater than 1, and for those less than 64, the function is less than 1.

7. Virtual machine queue delay factor - The user bias ratio is multiplied by the user-prioritized average queue delay to arrive at the virtual machine's queue delay factor. The queue delay factor is stored in the VMBLOK in the VMQPRIOR field. The units of the queue delay factor are equal to the high-order word of the time-of-day clock multiplied by 32 (approximately 1/32 of a second).

8. Q3 - If a given virtual machine is (a) a Q2 virtual machine, and (b) it is dropping from Q2 because it has used up its allocated processor in-queue time, and (c) it has dropped from Q2 at least 6 consecutive times because of using all allocated processor resources (without entering a true long idle wait), then the virtual machine is marked as Q3 and the queue delay factor is multiplied by 8. A Q3 virtual machine is treated as a normal Q2 virtual machine, except for differences in deadline priority calculation and the amount of processor time the virtual machine is allowed in queue.

9. Virtual machine deadline priority - The virtual machine's queue delay factor from Step 6 is added to the time of day when the virtual machine drops from queue. This value becomes the virtual machine's deadline priority and represents the expected time of day that the virtual machine should be next dropping from queue. The deadline priority is stored in the VMBLOK in the VMEPRIOR and VMTPRIOR fields.

10. Q1 deadline priority - If the virtual machine is being added to the Q1 eligible list, then the deadline priority is adjusted to account for the shorter processor time allowed in Q1. The queue delay factor (from VMQPRIOR) is subtracted from the deadline priority (from VMEPRIOR and VMTPRIOR). The queue delay factor is divided by 8 (shifted right 3) and stored back in VMQPRIOR as the new queue delay factor. In addition, if the virtual machine is currently using less than its allocated processor resource, the queue delay factor is shifted right additionally the number of places determined by the interactive bias. In the standard system the interactive bias is set to 2. The new queue delay factor is then added back into the result of the subtraction to form the Q1 deadline priority, which is then stored in VMEPRIOR and VMTPRIOR.

The VMBLOK is then sorted into the eligible list in ascending value of VMTPRIOR. The units used in the deadline priority calculation are obtained by taking the high-order word of the time-of-day clock and shifting it left 5 bit positions. The low-order bit position is approximately equal to 1/32 of a second. To handle the effects of losing the high 5 bits of the time-of-day clock, the current time-of-day value first has the time of day at system IPL subtracted from it before shifting. The effects of the individual virtual machine's resource utilization and user priority are illustrated by the following examples.

Example 1
Assume that two virtual machines are to be added to the eligible list for Q2. The current paging bias percentage is 0, both user priorities are 64, and the current average queue delay is 10 seconds. Virtual machine A has a current resource utilization (user bias) of 1/2 the system average. Virtual machine B has a resource utilization equal to the system average. The queue delay factors are obtained as follows:

Step 1:

Virtual Machine	Paging Ratio		Processor Ratio		Combined Ratio
A	0 x ?	+	(100-0) x 1/2	=	50/100 = 1/2
B	0 x ?	+	(100-0) x 1	=	100/100 = 1

Step 2:

Virtual Machine	Average queue delay		Virtual Machine Priority ratio		Queue delay
A	10	x	1	=	10
B	10	x	1	=	10

Step 3:

Virtual Virtual Machine	Combined ratio		Queue delay		Machine delay
A	1/2	x	10	=	5
B	1	x	10	=	10

If both virtual machines A and B dropped from queue at time-of-day 0, then virtual machine A has a deadline priority of 5 seconds and virtual machine B has a deadline priority of 10 seconds. Assuming virtual machine B continues to run, it should drop from queue once every 10 seconds, which is the system average. By cycling through queue at the average system rate, virtual machine B should maintain its processor resource utilization at the system average level (which will maintain its virtual machine delay at 10 seconds). Virtual machine A, whose current processor resource is 1/2 the system average, is allowed to cycle through queue in 1/2 the normal time. If virtual machine A continues to run at this rate, its processor resource utilization is accumulating at twice the system average. As virtual machine A's processor resource utilization approaches the system average, its combined ratio approaches 1. This increases its delay factor to 10 seconds, and its and virtual machine B running at the same rate.

Example 2
The conditions are the same as example 1, except both virtual machine A and B have the same processor resource utilization. Virtual machine A has a virtual machine priority of 64 and user B has a user priority of 54.

Step 1:

Virtual Machine	Paging Ratio		Processor Ratio		Combined Ratio
A	0 x ?	+	(100-0) x 1	=	100/100 = 1
B	0 x ?	+	(100-0) x 1	=	100/100 = 1

Step 2:

Virtual Machine	Average queue delay		Virtual Machine Priority ratio		Queue delay
A	10	x	1	=	10
B	10	x	1/2	=	5

Step 3:

Virtual Virtual Machine	Combined ratio		Queue delay		Machine delay
A	1	x	10	=	10
B	1	x	5	=	5

Virtual machine B continues to have queue delays calculated smaller than virtual machine A (or the system average) as long as its processor resource utilization is less than twice the system average.

 LY20-0897-7

The above examples illustrate the following general points about the deadline priority scheduling:

1. The purpose of the calculations is to effectively control each virtual machine's resource utilization rate by controlling how fast each machine is able to move through queue each time.

2. Each virtual machine's default allowed utilization is the same, namely the system "fair share" value, that is, the total system resources divided by the number of active virtual machines.

3. Deviations from a virtual machine's specified resource utilization (because of any particular advantageous or disadvantageous stay in queue) are corrected for successive stays in queue.

4. The system's response to any virtual machine's transaction is proportional to that virtual machine's current resource utilization and the total system load. Very trivial, interactive virtual machines receive the best service. Service for others degrades proportionally to their current level of utilization.

Paging Percentage Bias

The paging percentage bias is a dynamically calculated value that is proportional to the percentage of time spent in the eligible list. The upper limit of the paging percentage bias is 40 percent. The paging percentage bias represents the portion of the utilization ratio due to paging, rather than processor utilization. The maximum paging percentage bias can be queried and set with the QUERY SRM and SET SRM commands.

Interactive Bias

The interactive bias shift is set to 2. Normally when a virtual machine enters the eligible list as Q1, its priority delay factor is shifted right 3 (divided by 8). In addition, if the virtual machine's current resource utilization is less than its allocated amount (or fair share), the priority delay factor is shifted right an additional amount specified by the interactive bias (2). This allows much faster response for trivial interactive virtual machines, but still prevents the nontrivial interactive virtual machines from obtaining more than their fair share of the system resources. The interactive bias can be queried and set with the QUERY SRM and SET SRM commands.

User Priority Function

The user priority function is a nonlinear function, and changes to user priorities effect performance differently, depending on the original priority. The priority ratio is used to pro-rate a virtual machine's queue delay. The smaller the ratio (and the smaller the priority), the shorter the queue delay and therefore the better the service for that virtual machine. The following is a table of user priorities and the priority ratio calculated from the user priority function.

User Priority	Priority Ratio
0	1/64
11	1/32
21	1/16
32	1/8
43	1/4
54	1/2
64	1
75	2
85	4
96	8

A user priority of 0 provides a virtual machine with a "fair share" 64 times as large as the average (by multiplying its actual use by 1/64). A user priority of 99 provides a virtual machine with a "fair share" 1/8 of the average (by multiplying its actual use by 8). The normal priority is assumed to be 64 (which results in a multiplier of 1).

Virtual Machines in Idle Wait

Virtual machines that use the Diagnose X′58′ instruction to communicate with 327x terminals are in an idle wait state when they are waiting for full-screen output. To prevent the scheduler from dropping these virtual machines from queue and adding their page frames to the FLUSHLST, CP keeps track of virtual machines that are currently waiting for high-speed I/O by using the counter VMDVBSY in each user's VMBLOK. Whenever VMDVBSY contains a positive value, indicating that the user is in idle wait and waiting for the high-speed I/O to complete, the scheduler will start the 300 millisecond queue drop delay timer and return to the caller when it would otherwise drop the virtual machine from queue.

VMDVBSY is increased whenever a virtual machine issues a Diagnose X′58′ instruction, and decreased when the I/O service is finished. DMKGRF, which supports graphic display devices, uses the status of the bit RDEVPS located in the field RDEVSTA3 to control VMDVBSY. RDEVPS tells DMKGRF whether it is dealing with a logical device (RDEVPS on) or a local device (RDEVPS off). The VMDVBSY field is not maintained for logical devices.

Controlling Multiprogramming

To control the number of virtual machines allowed in queue, the scheduler monitors the paging activity of all virtual machines and of the total system. A decision as to whether or not to move a potential virtual machine from the eligible to the dispatch list is based upon whether or not its projected working set exceeds the system's remaining capacity. A virtual machine's projected working set is calculated at queue drop time.

Average Resident Pages - The basic mechanism used in predicting working set size is dependent upon the virtual machine's average resident pages while in queue. When a page is acquired, the current number of resident pages for the virtual machine is added to the resident page sum (VMPGRINQ). At queue drop, the resident page sum and the number of pages read are corrected to allow for initialization of the virtual machine's working set, and the corrected resident page sum is divided by the corrected number of pages read and swapped in to approximate the average number of resident pages.

Projected Working Set Size - Before using as a working set size estimate, the average resident page value is adjusted upward or downward dependent upon how well the virtual machine (and overall system) ran while it was in queue.

There is a tuning parameter in the system, which is changeable by the CP SET PAGING command, that provides a "norm" with which to compare paging performance. This value is set at a default value of 4. It is used in two ways to adjust the average resident page value for use as the working set size estimate.

1. The average resident page value is increased by 1 for each page steal in excess of 4 percent of the total number of pages read for that virtual machine while in queue.

2. The average resident page value is multiplied by the square root of the virtual machine's paging performance ratio, which is defined as follows:

At queue drop time, the estimated productive processor time per page read is calculated. The total processor time used by the virtual machine while in queue is divided by the number of pages read while in queue. From this processor time per page read value is subtracted the estimate processor time per page read. The scheduler maintains an "ideal" system-wide productive processor time per page read value. Dividing the virtual machine's calculated productive processor time per page read by the "ideal" system value gives the virtual machine's paging performance ratio for that time in-queue.

The adjustment formula for average resident pages is:

```
ARP' =  (ARP + S) * (ICPU/PCPU)  - reserved pages
```

where:

ARP'	is the adjusted average resident page value.
ARP	is the calculated average resident page value.
S	is the number of page steals in excess of 4 percent of the number of pages read.
PCPU	is the average productive processor time per page read.
ICPU	is the system's ideal productive processor time per page read.

The projected working set size estimate is set equal to the adjusted average resident pages (ARP') and stored in the VMWSPROJ field in the VMBLOK. There are four constraints on the projected working set size estimate:

1. The minimum value for the projected working set size is 2 (except for a V=R user or a user with reserved pages, in which case it may be 0).

2. The number of pages read while in queue.

3. The maximum number of pages owned plus the number of pages stolen while in queue.

4. The maximum value of a size is set by the SET SRM MAXWS command. Ordinarily this maximum is not active.

Assuming that page contention is being adequately controlled and the system resources have not been severely overcommitted, the projected working set size would ordinarily be close to the number of average resident pages. However, this estimate is sensitive to the overall system paging activity for the following reasons:

1. If there is no paging load on the system, the "ideal" productive processor time per page fault is set to its minimum value, and the projected working set size is continually underestimated.

2. As paging activity increases, the ideal productive processor time is calculated larger and causes the square root calculation to approach 1.

3. If the paging activity becomes excessive, and the system is unable to control it by reducing the multiprogramming level, the ideal productive processor time figure approaches its maximum value. This normally happens when there is insufficient real storage or paging I/O capacity to meet the interactive requirements.

In summary, the scheduler selects the subset of logged-on virtual machines that are allowed to compete for the resources of the processor. The scheduler prevents a virtual machine from being added to the active subset if the virtual machine's projected main storage requirements, added to those of the other active virtual machines, and the larger size of the interactive buffer (for noninteractive virtual machines), or the size of the largest user left in the eligible list cause the current capacity of the system to be exceeded. Once the active subset (the set of in-queue virtual machines) has been selected, the dispatcher allocates resources of the processor among them.

The list of executable virtual machines in a queue (on the dispatch list) is sorted by deadline priority (the same as for the eligible list). The only exceptions to this rule are machines identified as being compute-bound, that is, which use at least the dispatching time-slice amount of processor time without becoming nonrunnable. The dispatching time-slice is set at 50 ms for a System/370 Model 145 and is adjusted for other models based on the ratio of their speed to the Model 145 speed. In the case of a compute-bound virtual machine, the following value is added to the deadline priority for use in ordering the virtual machine on the dispatch list:

```
deadline TOD - current TOD
--------------------------
            4
```

The current TOD is taken as the time it was first determined that the virtual machine was compute-bound.

Virtual machines identified as interactive are not necessarily placed at the top of the dispatch list, although you can ordinarily expect to find them there. The virtual machine that is found at the top of the list is ordinarily one that: 1) has been waiting the longest since it was dropped from queue or 2) has the smallest current resource utilization. Normally, interactive virtual machines satisfy both of these requirements.

Projected Working Set Size Estimate Feedback Control

A performance monitoring routine (DMKSTP) is invoked at intervals between 15 and 60 seconds (the interval depends on the processor speed and the multiprogramming level). This routine calculates among other things: a) the smoothed activity values used in the INDICATE LOAD command response, b) the average queue delay, c) the average processor resource utilization, and d) the "ideal" processor per page read.

Part of the response to the INDICATE LOAD command is a value that estimates the paging load on the system due to page contention, expressed as a percent of total system resources. The difference between this percent number and the SET PAGING value is divided by the number of pages read during the interval. This percent (positive or negative) is multiplied by the measurement interval and the resulting product is added to the "ideal" processor time per page fault.

The "ideal" processor time per page read should control page contention by controlling the working set size estimates. Page contention can be reduced by reducing the level of multiprogramming. High levels of page contention (with respect to the SET PAGING value) cause the "ideal" processor time per page read to increase. Eventually, increases in this "ideal" value cause all working set size estimates to increase, which should lead to decreases in the average multiprogramming level. The reverse situation, where page contention is lower than the SET PAGING value, causes a reduction in the "ideal" value, leading to smaller working set size estimates and possibly higher levels of multiprogramming.

Dispatcher Fast Path

The dispatcher fast path is software support that allows certain users running in the appropriate real and virtual machine states to take a shortcut through the code in DMKDSP. It eliminates unnecessary interrogation of users by the dispatcher before they are run, thus reducing the average time a user spends in DMKDSP. A user can take the fast path when the dispatcher determines, from an inspection of summary bits, that the user belongs to a class of virtual machines that do not need much checking before they are run.

The flags that indicate if a user is in the appropriate real machine environment to take the fast path are located in the byte DPSTAT in the PSA. If virtual machine assist is on the machine, this byte is initialized at IPL time to X'0F' plus the bits that reflect the actual machine state. If virtual machine assist is not on the machine, the state of the low order four bits is unpredictable, but since the DPMICON bit will be zero, the dispatcher fast path will not be activated anyway. The dispatcher fast path requires that each of the four bits are turned on. This indicates that the user is in a physical AP/MP environment with virtual machine assist on and the segment protection feature available, and that the TLB has not been used by a STBYPASS VR user since the last PTLB (purge translation lookaside buffer) instruction.

Bits Defined in DPSTAT:

DPMICON	X'80'	Indicates virtual machine assist is on.
DPAPUOP	X'40'	Indicates system is in AP/MP mode.
DPSEGPRT	X'20'	Indicates system has segment protect feature.
DPOKTLB	X'10'	Indicates TLB is or will be OK when a virtual machine is dispatched. It is turned off when a STBYPASS VR user is dispatched and turned back on when a non-STBYPASS VR user is dispatched.

The flags that tell the dispatcher if a user is in the appropriate virtual machine state to take the fast path are located in a byte in the VMBLOK called VMPATH. This byte contains three bits that reflect the user state, the interrupt history, and the locking history of the virtual machine. A user can take the fast path when all seven bits in DPSTAT and VMPATH are on. As the table below shows, a virtual machine is eligible for the fast path if it is in BC mode and if it is resuming execution after a simple interrupt such as an I/O interrupt, a disk diagnose, or a page fault.

Bits Defined in VMPATH:

VMVBCM	X'04'	Turned on to indicate a simple BC mode machine.
VMIHIST	X'02'	Turned on when the virtual machine is in a simple interrupt that leaves it eligible for the fast path. Turned off when the VMBLOK is locked more than once before the event that caused the interrupt has been completed.
VMLKHIST	X'01'	Turned on when the virtual machine has been locked. Turned off when an event which makes the virtual machine eligible for the fast path (such as a paging operation) has completed.

One action the dispatcher must perform to prepare a virtual machine to run is to acquire the VMBLOK lock. The logic for interfacing with the lock manager, DMKLOK, is contained in the subroutine LOCKVM.

When the lock manager is called, if the VMBLOK lock is obtained, DMKLOK maintains the VMLKHIST and VMIHIST bits as follows:

- If VMLKHIST is already on, turn VMIHIST off
- If VMLKHIST is off, turn VMLKHIST on.

Since both VMIHIST and VMLKHIST must be on in order for the virtual machine to take advantage of the dispatcher fast path, the virtual machine will be disqualified from the dispatcher fast path for its next dispatch if DMKLOK turns off the VMIHIST bit.

Every virtual machine has its VMIHIST and VMLKHIST set to B'00' at dispatch time, regardless of the previous settings.

When the subroutine LOCKVM is called and general register 1 is equal to LASTUSER, the lock manager is not called. Therefore, code equivalent to the maintenance of the history bits in the lock manager is included in the subroutine LOCKVM.

Another way the dispatcher fast path saves time spent in DMKDSP is by preserving the interrupt old PSWs in a field called VMECPSW after a program interrupt or an I/O interrupt. This support also saves real control register 6 in a field called VMC6SAVE after a program interrupt or an I/O interrupt.

Thus, if the virtual machine qualifies for the dispatcher fast path after it was interrupted by DIAGNOSE X'18', a page fault, or an I/O interrupt, the dispatcher does not have to reconstruct the appropriate PSW and control register 6 in order to run the guest. The correct data is available in VMECPSW and VMC6SAVE except for reflection of the correct condition code from the DIAGNOSE X'18' completion.

Fast Redispatch

DMKDSP also provides a fast dispatch path for virtual machines that have issued specific privileged instructions that are not handled by virtual machine assist.

These virtual machines can be dispatched very rapidly because the virtual machine's program old PSW needs very little reconstruction to redispatch the virtual machine, hence use of full PSW reconstruction path is not required. The decision for using the fast dispatch path (DMKDSPA) is accomplished by the module that handles privileged operation, DMKPRV or DMKVIO. A fast redispatch path is also available after I/O interrupts. If DMKDSP can determine that the I/O interrupt processing had no effect on the running virtual machine's status and it caused no higher-priority virtual machine to become runnable, then the virtual PSW stored at the I/O old PSW location will be used to redispatch the virtual machine.

Enable Window

The CP supervisor runs disabled for all I/O and external interrupts. The dispatcher, in order to alleviate part of this problem, will temporarily enable for interrupts and then disable. There are three occasions when the dispatcher enables for interruptions (enable windows):

1. When an enabled wait state is entered.

2. When an enabled problem state is entered to run a virtual machine.

3. When another part of the supervisor is to be entered via the unstacking of an CPEXBLOK.

On occasion 3, if the dispatcher finds a CP request block to unstack, it first enables then disables for I/O and external interruptions before unstacking the request.

Favored Execution Options

When the resources of the processor (and real storage) are being allocated, the dispatching and scheduling functions are implemented in such a manner that options exist which allow an installation to designate that certain virtual machines are to receive preferential treatment.

The favored execution options allow an installation to modify the algorithms described above and force the system to devote more of (or a specific portion of) its resources to a given virtual machine than would ordinarily be the case. The options provided are:

1. The favored execution option.

2. The favored execution percentage.

The favored execution option means that the virtual machine so designated is never to be dropped from the active (in-queue) subset by the scheduler as long as it remains runnable. When the virtual machine is executable, it is to be placed in the dispatchable list at its normal priority position. However, any active virtual machine represents either an explicit or implicit commitment of main storage. An explicit storage commitment can be specified by either the virtual=real option or the reserved page option. An implicit commitment exists if neither of these options are specified, and the scheduler recomputes the virtual machine's projected working set at what it would normally have been at queue-drop time. Multiple virtual machines can have the basic favored execution option set. However, if the combined main storage requirements of the favored virtual machines and the other runnable virtual machines exceed the system's capacity, performance can suffer due to thrashing.

The basic favored execution option removes the primary source of elapsed time stretch-out in a loaded time-sharing environment. However, if the favored task is highly compute-bound and must compete for the processor with many other tasks of the same type, an installation can define the processor allocation to be made. In this case, the favored execution percentage option can be selected for the virtual machine. This option specifies that the selected virtual machines are to receive a given minimum percentage of the total processor time, if it can use it. It modifies the deadline priority to specify a fixed interval between queue drops instead of one proportional to system load, number of active virtual machines, current resource utilization, etc. The deadline priority is modified in the following way:

1. The in-queue time slice is divided by the requested percentage to arrive at the fixed stretch-out interval.

2. The calculated interval is added to the TOD at queue drop to arrive at the next expected queue drop TOD (or deadline).

These options can impact the response time of other virtual machines. Both favored options can be applied to any number of virtual machines, either in conjunction with one another or separately.

Dispatching and Scheduling Support Routines

Most of the routines in the CP nucleus are reenterable and multiple control program or virtual machine tasks can make use of one routine at the same time. However, there are certain areas where requests for a resource must be serialized (as in paging) or delayed while previous requests are serviced (as in requests to schedule I/O).

The CP Request Stack

The routine handling the request obtains a CPEXBLOK from free storage and stores the caller's registers in it; when the requested resource is free, the CPEXBLOK is stacked for the dispatcher via a call to the request stack manager (DMKSTK). The dispatcher unstacks the block and exits to the requesting routine the next time it is entered. I/O requests are stacked in the same manner, except that the stacking vehicle is the IOBLOK, and return is passed to the address specified in the interrupt return address (IOBIRA). In either case, it should be noted that the dispatcher always unstacks and gives control to any stacked IOBLOKs and CPEXBLOKs prior to dispatching a user. This guarantees that CP information needed by a virtual machine (such as page availability) is always as up to date as possible.

CP Spooling

The spooling support in CP performs three functions.

- Simulates the operation of the virtual unit record devices that are attached to each user's virtual machine configuration. The simulation is done in such a way that it appears to the program in the virtual machine that it is controlling a real unit record device. This support involves the interception and interpretation of virtual machine SIOs, the movement of data to and from the virtual machine's virtual storage space, and the reflection of the necessary interruption codes and ending conditions in PSWs, CSWs, and sense bytes. This support is provided by the virtual spooling executive.

- Operates the real unit record equipment, attached to the system, that transcribes virtual machine output spool files to the real printer or punch and input from the real card reader to DASD storage. This function is provided by the real spooling executive.

- Provides an interface among the virtual machines, the system operator, and the spooling system so that the location, format, priority and utilization of the systems spooling data and resources can be controlled.

Spool Data and File Format

Data Format

The buffers that collect and write spool data are all one page (4096 bytes) in length, and contain the data to be transcribed and all CCWs necessary for operating the unit record devices that perform the transcription. The data is provided in the exact format required with no compression except that trailing blanks are suppressed. A 2-byte field at the end of the data CCW contains the length of the original data (including the truncated blanks). The first two doublewords of each buffer contain linkage information described below, followed by the data and CCWs, except for the first spool buffer which contains 3800-related information.

Spool files created on virtual spooled 3211, 3203, 3262, or 3289E, or 4245 printers may contain the following CCWs:

Load FCB (X'63')
Fold (X'43')
Unfold (X'23').

Spool files created on a virtual 3800 may contain the following CCW's:

Load Forms Control Buffer (X'63')
Load Translate Table (X'83')
Load WCGM (X'53')
Load Copy Number (X'23')
Load FOSC (X'43')
Load Graphic Char. Mod. (X'25')
Load Copy Modification (X'35')
Initialize Printer (X'37')
Clear Printer (X'87')
Select Translate Table 0 (X'47')
Select Translate Table 1 (X'57')
Select Translate Table 2 (X'67')
Select Translate Table 3 (X'77')
End of Transmission (X'07')
Mark Form (X'17').

In addition, since the data associated with the Load Graphic Modification (X'25') and Load Copy Modification (X'35') CCW's may be longer than 4080 bytes. This data may appear in successive DASD buffers. This is accomplished by setting the "Data Chaining" bit in the CCW associated with each section of data except the last one.

Each spool logical record (card or print line) is stored as one CCW that moves data (READ or WRITE), a TIC to the following CCW, and the full data record. Space is left at the end of each buffer so that a SENSE command can be inserted to force concurrent channel end and device end. For card punch channel programs there is an additional back chain field that points to the card previously punched so that error recovery for punch equipment checks can back up one card. The only exception to the format of READ/WRITE-TIC-Data is in buffers of files directed to the printer. In this case, immediate operation code CCWs (skips and spaces) are followed by the next CCW.

File Format

In addition to the data and CCWs contained in each spool buffer, the first two doublewords contain forward and backward links to the next and previous buffers in the file. This two-way linkage allows the file to be backspaced or restarted from any point at any time. The exception to this are spool files that contain 3800-related CCW's. These files can only be restarted from the beginning of the file. Also, it means that if I/O errors are encountered while reading one buffer, the file is put in system hold status. If purged, all buffers except those in error are released. The two-way chain allows this control of the file while preventing fragmentation by allowing pages to be assigned and released individually regardless of their ownership.

The first spool buffer of an output spool file contains a special data record called the tag record. This record immediately follows the two doublewords containing the forward and backward buffer linkage pointers. The tag record allows VM/SP HPO users to specify information to be associated with spool files that they generate. The information is entered via the CP TAG command, although the tag record is not considered a spool file data record and is not printed or punched as part of the spool file. However, the contents may be interrogated via the CP TAG QUERY command. There are two fields at the end of the buffer to identify the number of buffers in the file and the size of the largest data written in the file.

The format of the tag record is an NOP CCW, followed by a TIC to the next CCW and a 136-byte data field. Any blanks in the NOP CCW will not be truncated. Therefore, the 2-byte length field at the end of the CCW will always contain X'0088' (decimal 136). To differentiate the tag record from an immediate NOP CCW (no TIC-data sequence) independently of the command code, the "skip" bit (bit 35) in the CCW has the following convention:

Bit 35	= 0 for NOP CCW, TIC, data (tag record)
	= 1 for NOP CCW (immediate NOP command)

Each spool file in the system is controlled by a spool file control block (SFBLOK). Printer and punch SFBLOKS are resident in free storage. Closed reader SFBLOKs are found in a virtual address space, SYSSPOOL, created specifically for them. While the file is open, these blocks are chained from the devices (either real or virtual) that are processing the file. When a printer or punch file is closed, it is chained from the device type file anchor (DMKRSPPR for printer files, DMKRSPPU for punch files). When a reader file is closed, it is moved from real storage to a user-unique spool file chain in SYSSPOOL's virtual storage, anchored in the Reader Hash Table. This table contains pointers to a user's first reader file. When a user is logged on, the VMBLOK points to the user's Reader Hash Table entry. Each SFBLOK contains information about the file that describes its owner and originator (these can be different for transferred files), the filename and filetype, and the class and number of copies for output files. All of these attributes can be examined and most can be changed by the file's owner or the system operator. The SFBLOK also contains information such as the starting and ending buffer addresses for the file, the starting address and the length of external attributes buffer (if it exists), the record size, certain file status flags, destination, and so on.

Spool Buffer Management

Real/Virtual Storage Management

Buffers that temporarily store spool data on its way between DASD secondary storage and the user's machine are allocated from a pool of virtual storage space that belongs to CP. The size of this pool varies with the real storage available to VM/SP HPO (the storage specified at system generation or actual real storage, whichever is less). Allocation is as follows:

Storage Size Available	Virtual Buffers Allocated
384K to 655,360 bytes	128
655,361 bytes to 1.1 megabytes	320
1.1 megabytes to 3 megabytes	640
over 3 megabytes	1280

Virtual storage buffers are allocated in 1-page increments by DMKPGT at the time the spool file is opened for either input or output. If no virtual storage space is available, the virtual machine is terminated with an abend. This places limits on the number of concurrent spooling operations permitted by the system because spooling operates as a high-priority task.

Real storage is not allocated for a spooling buffer until a virtual machine actually issues a SIO that attempts to transfer data between the buffer and the user's virtual storage space. At this time, a page of real storage is allocated to the buffer via the real storage paging manager. The buffer is locked in main storage (that is, is unavailable to be paged out) only for the amount of time necessary to transfer the data. After the data transfer is complete, the buffer is treated as a normal page of virtual storage, and can be selected to be paged out. This ensures that low-usage spool files do not have buffers in real storage, while the buffers for high-usage files should remain resident. (Two spool file buffers are maintained for printing on a real 3800 printer.)

DASD Space Allocation

While a spool buffer is inactive, it resides in real storage or on the paging device. After it has been filled with data from the virtual machine or a real input reader, it is written to a page of secondary DASD storage. The allocation of pages on the spooling disk(s) is managed by DMKPGT, which handles requests for both pages of virtual storage and semipermanent spool file residence. DMKPGT maintains a separate allocation block chain for spooling pages. Each block contains control information and a bit map that allocates pages on a single cylinder. If none of the cylinders allocated have any available pages, DMKPGT enters its cylinder allocation routine. See the section "Cylinder Allocation" for more information.

Paging Device Support: All actual I/O for the page buffers on any device is controlled by the paging I/O executive DMKPAGIO.

Virtual Spooling Manager (DMKVSP)

The two functions of the virtual spooling manager are (1) to simulate the operation of all spooled unit-record devices attached to the user's virtual machine, and (2) to read and write the spool files associated with those devices. The following virtual devices are supported for spooling, with the exceptions noted:

- IBM 2540 Card Reader/Punch, except for punch feed read and column binary
- IBM 3203 Printer Model 4 and Model 5 (132 positions)
- IBM 3262 Printer Models 1, 5 (in 3262-1 Emulator Mode), and 11 (132 positions)
- IBM 1403 Printer Models 2 and N1 (132 positions)
- IBM 3211 Printer (150 print positions)
- IBM 3505 Card Reader (except for mark senses reading)
- IBM 3525 Punch (except for the card read, print, and data protect features)
- IBM 3800 Printing Subsystem (204 print positions)
- IBM 4245 Printer Model 1 (132 positions)
- IBM 4248 Printer Model 1 (168 positions).

The following consoles are supported for spooling when entered into the directory as the virtual system console:

- IBM 1052 Printer-Keyboard Model 7 (via the 2150 Console)
- IBM 3210 Console Printer-Keyboard Models 1 and 2
- IBM 3215 Console Printer-Keyboard Model 1.

All virtual printers, except 3800, must have the universal character set feature. No checking is done on the spooled printer data. There is, however, an exception to this. On a virtual 3800 all Load Checks and Data Checks are reflected to the virtual machine unless

- Explicitly disabled via the NODATCK parameter of the DEFINE command or
- Explicitly disabled via a 'Block Data Check' CCW or if
- A 'Select Translate' CCW has associated with it a specified character arrangement table in the spool command.

However, any UCS buffer commands issued by the virtual machine (load UCS buffer, block data checks, etc.) are ignored. It is up to the user and the installation to ensure that the output is directed to the proper real printer via use of the output CLASS and FORMS described below. For the 3203, 3211, 3262, 3289E, 4245, or 4248 printer, forms control buffer (FCB) commands are accepted and simulated by means of a virtual FCB maintained by the executive. For the 4248 printer, the extended FCB is also accepted. The use of the virtual FCB is the only way to simulate end-of-form conditions reflected by the detection of a channel 9 or 12 punch. The LOAD FCB command and its associated data (FCB image) are also captured in the spool file. When the file prints on a real 3203, 3211, 3262, 3289E, 4245, or 4248 printer, according to the file FCB, CFILEFCB, or DEFFCB option on the START command, the FCB image is sent to the real printer and controls channel skipping. If the file prints on a real 1403, the FCB is ignored. Since there is no provision for FCB loading, the device is carriage tape controlled.

The FOLD and UNFOLD commands are also captured in the spool file for a virtual spooled 3203, 3211, 3262, 3263, 3289E, 4245 or 4248, and sent to the real printer when the file is processed (unless it is a real 1403 printer).

For a virtual 3800, all Load and Control CCWs are included in the spool file and issued when printed on a real 3800. When that file prints on another device (such as 1403, 3211, 3262, 3289E, 4245 or 4248) these CCWs are changed to NO-OPs and therefore ignored.

If any of the unsupported unit record features are required, the real device must be attached directly to the user's virtual machine. Thus, a 3505 reader could be a spooling input reader, but attached directly to a batch virtual machine when it is necessary to read mark sense cards.

If a 3211-type printer is started with the DEFFCB option, all LOAD FCB commands imbedded in a spool file are NO-OPed (that is, they are not sent to the printer).

Note: If a spool file contains an extended FCB, it can only be printed on a real printer that supports the extended format (the 4248) or on a real printer that NO-OPs the LOADFCB command.

Output File Processing

DMKVSP receives control from the virtual I/O executive, DMKVIO, when the user's machine issues a SIO to a spooled unit record device. DMKVIO does not pass control until it has been determined that the device is available (that is, it is not busy and has no interruptions pending). DMKVSP first determines if the device is currently processing a file. If it is, processing continues. If this is the first command issued by the given device, a new output file must be opened. An open subroutine is called to build the control blocks necessary to manage the file and to obtain virtual storage and DASD buffer space. Control is then returned to DMKVSP.

Before the first record of an output spool file is written, DMKVSP writes a tag record (NOP CCW, TIC, data sequence) and initializes the 136-byte data area to blanks. It then sets the spool buffer displacement pointer to the first doubleword in the buffer beyond the tag record. DMSVSP then analyzes and interprets the channel program associated with the virtual machine's SIO. Each CCW is tested for validity of command, address, flags, alignment, protection, etc., and if the CCW is valid, the virtual machine's data is moved from his own virtual storage space to the buffer in the spooling virtual storage. When this buffer is full, it is written to a page of DASD secondary storage and a new buffer is obtained. The interpretation of the virtual machine's channel program continues until there are no more CCWs or until an error condition is detected that prohibits further processing. In either case, the device is marked as having the proper interruptions pending, a CSW is constructed, and DMKVSP exits to the main dispatcher. In contrast to nonspooled I/O, the virtual machine has remained in a pseudo-wait (IOWAIT) for the time it took to interpret the entire channel program.

The output file can be logically closed by the virtual machine either by issuing an invalid CCW command code, or by the CP CLOSE command. In either case, DMKSPL checks for tag record information and 3800-related information in the VSPXBLOK. (The VSPXBLOK, pointed to by the VDEVEXTN field of the VDEVBLOK for the output spool device, contains the tag information entered via the CP TAG command.) If tag data exists, the first spool buffer for the file is read in, the tag data is inserted in the tag record, and the buffer is rewritten to DASD storage. If no tag data exists, the tag record data field is left blank. The first buffer is read in, and the number of buffers and the size of the largest CCW are put into the first SPLINK. The device is then cleared of pending interruptions, the file chains are completed, and the file either is queued for output on a real device of the proper type (printer or punch) or, if TRANSFER is in effect, is queued for input to another virtual machine.

The 3800-related information includes:

CHARS - character arrangement table
MODIFY - copy modification name
FCB - file control block
FLASH - flash count overlay use.

This information is contained in the VSPXBLOK for a virtual printer. When the file is closed, the information is contained in the first DASD buffer.

Input File Processing

Input file processing is similar to output file processing, except for the open and close functions, and the analysis of CCW commands and the direction of data movement. Many common routines are utilized to locate and verify CCWs, obtain buffer space, and to move the spooling data.

The difference in the open function is that, instead of creating a new file, it is necessary to locate a reader file that already exists in the system. To do this, the open subroutine scans the user's reader SFBLOK chain, anchored in the Reader Hash Table. If a file is not found, a unit-check or intervention-required condition is reflected to the virtual machine; otherwise, its SFBLOK is chained to the control block for the reader and the channel program is interpreted in the same manner as for an output file.

After the input file is exhausted, a unit exception is reflected to the user machine, unless the user has requested either continuous spooling or that an EOF not be reflected. With continuous spooling, the unit exception is not reflected until the last file for that virtual machine is processed. If NOEOF is specified, the simulation terminates with a unit check or intervention-required condition (similar to what happens if the EOF button on a real reader is not pushed).

In either case, the input file is then deleted from the system, unless the user has specifically requested that his input files be saved. If the file is saved, it can be re-read any number of times.

Virtual Console Spooling

Support of virtual console I/O for both the virtual machine and VM/SP HPO is provided as an option for the spooling capabilities. This support fulfills the following requirements:

- Provides hardcopy support for CMS Batch Facility virtual machines.
- Provides hardcopy support for display devices used as system or virtual machine consoles.
- Allows disconnected virtual machines to spool virtual console output, CP commands and system resources to disk instead of losing the output.
- Improves the performance of virtual machines that currently produce a large amount of console output.

Whenever a SIO is issued to a virtual machine console, the virtual console manager (DMKVCN) determines if the spooling option is active. If it is, control is passed to DMKVSTVP (via DMKQCN and DMKQCO) to insert the data into a spool file buffer. While console spooling utilizes, basically, the same code as printer spooling, the following exceptions are made:

- A skip to channel 1 CCW is inserted after every 60 lines of output.
- The operator's virtual console spool buffer is written out after every 16 lines of output.

- The virtual spool buffer is written out to the allocated spool device when the first CCW is placed in that virtual buffer. The linkage area of the virtual spool buffer takes the form of a CLOSE file to allow checkpoint (DMKCKP) to recover the active spool file in the event of a shutdown because of system failure. If data in the virtual buffer has not yet been written to the spool device, it will not be recovered.

To maintain a pseudo closed file status for console spool files, DMKSPL now assigns spool identifications to all output spool files where they are first queued.

A virtual system reset, device reset, or IPL *does not* close the virtual console spool file. The LOGOFF, FORCE, or DETACH of virtual console commands *does* close the virtual console spool file. The SHUTDOWN command does close the operator's console spool file. If the SHUTDOWN command is issued by a Class A user other than the operator, the console spool file for both the user and operator is closed.

The inclusion of the spool file tag record in a virtual console spool file is processed by DMKVSP and DMKSPL as described for printer spool files in "Output File Processing" under "Virtual Spooling Manager." Virtual console I/O is not spooled when the TERMINAL CONMODE 3270 function is used and when virtual channel command codes X'19', X'29', and X'2A' are used by full screen applications.

Real Spooling Manager (DMKRSP)

Command chaining is used for all unit record channel programs so that the devices are running at their maximum speed with a minimum of interruptions. In addition, because of the high speed of the 3800 Printing Subsystem, a double-buffering arrangement is used to write output to it. All other real spooling devices utilize a single output buffer.

Output File Processing

Both the input and output operations of DMKRSP are interruption driven. Thus, DMKRSP does not process unless an internally or externally generated not-ready to ready device end interruption occurs. External interruptions are generated by the hardware in the normal manner, while internal, "pseudo interruptions," are generated by the software when an output file has been queued on the real printer or punch file chain, or when the operator issues a START command to a drained device.

Upon receipt of the initial device end for a printer or punch, DMKRSP searches the appropriate file chain for the SFBLOK of a file whose class, form, and destination match those of the device that was made ready. If FLASH is specified for a 3800 printer, the flash overlay name must also match. In addition, the location of 3800 Load CCWs within the spool file will determine whether it is eligible for printing on the given output device. If the device is in AUTO or SETUP mode, first preference is given to the current form. If no file with the current form exists, a file with a different form is selected, and a MOUNT REQ message is sent to the operator. On the next device end, processing will continue. When the SFBLOK is located (provided the file is not in a hold status), it is unchained from the output queue and chained to the real device block that services the file. A page of real main storage (two pages for a 3800 printer) is then obtained for use as a buffer, and the output separator routine (DMKSEP) is called to print output identifier pages. DMKTCS and DMKTCT are then called to set up the 3800 for printing that file. When DMKSEP returns control to DMKRSP, the first buffer of the file is paged into real main storage, and the CCWs in the channel program that it contains are adjusted so that their data addresses correspond to the real addresses at which the data resides. To reduce the number of channel commands that cause movement of the printer carriage, DMKRSP also performs CCW optimization. Whenever a sequence of two carriage control commands can be replaced by a single equivalent command, the first CCW is replaced by the equivalent command and the second is changed to a no-op. For example, if a "Write and Space 1 line" command is followed by an "Immediate Space 2 lines" command, DMKRSP will replace them with a single "Write and Space 3 lines" command. The real SIO supervisor (DMKIOSQR) is then called to start the channel program, and DMKRSP exits to the dispatcher (DMKDSPCH) to await the interruption.

In SETUP mode, only a page of data at a time is printed until the START command is entered again. Then the entire file is printed.

When the channel end/device end interruption for the completed buffer is unstacked to DMKRSP, the forward chain file link field locates the next buffer. This buffer is paged-in, and the process is repeated until the final buffer is processed. At this point, the number of copies requested for the file is decremented. If the number of copies is 0, processing is terminated and the file is deleted from the system; otherwise, the process is repeated as many times as necessary. For a 3800 printer, double buffering is maintained so that the second buffer is filled while the first buffer is being printed.

When file processing is completed, a scan of the appropriate output queue is again made, and if a file is found it is processed. If the queue is empty, or if a file with a matching class, form, or destination is not found, an exit is taken to DMKDSPCH to wait for another ready interruption. If a 3800 device is used, the file is placed on the 3800 delayed purge queue. If this queue reaches maximum size, the oldest file in the queue is deleted from the system.

Output file processing can be modified by either the system operator, by a spooling support command or as a result of system errors. The operator commands allow a given file to be backspaced or restarted, and the files of individual users or the whole system to be held and released for output. I/O errors also affect the spooling system, and a description of how they are processed is in the section "Spool File Error Recovery."

Input File Processing

Reader file processing is initiated by the receipt of a device end interruption from a spooling card reader. No explicit operator command is required to start the processing of an input file. When the device end is unstacked to DMKRSP, a call is made to DMKRSTIN to handle the input process. DMKRST contains an open subroutine that is called to build the necessary control blocks and to obtain the virtual, real, and DASD buffer space required for the file. A channel program to read 41 cards is built in the buffer, and DMKIOSQR is called to start the reader.

When the interruption for the first buffer is unstacked, the first card is checked for its validity as a userid card. The minimum information that this card must contain is the userid of the owner of the input file. It may appear anywhere on the card, with the restriction that it must be the first information punched. Optional information on the userid card can include a filename and type and/or the class of the virtual card reader to which the file is to be directed. If the userid is valid, the file processing continues; otherwise, the operator receives an error message and processing is terminated.

After each file buffer is read, it is written onto disk by the paging I/O routines in the same way that virtual output files are handled. When a unit exception signaling physical end of file is received from the reader, the file is closed by writing the final buffer to disk and completing and queuing the SFBLOK to the user's reader file chain. If the owner of the file is currently logged on, he is given a message indicating that a file has been read and if he has an available card reader, it is posted with a device end interruption. An available reader is one of the correct class which is ready, is not busy, has no active file, and has no pending interruptions.

As DMKRST completes each phase, it returns to DMKRSP for an exit to the dispatcher (DMKDSP).

Accounting Card Processing

Various routines in CP accumulate, format, and store account records that contain system usage information for certain users. These routines format the information into an 80-column card image record.

In addition to the records generated by CP to account for a virtual machine's use of system resources, the user may request records in order to account for the use of virtual machine resources by jobs running under his userid. In order to do so, the user must have the account option (ACCT) entered into the directory.

The user can issue a code X'004C' DIAGNOSE instruction with a pointer to either a parameter list containing user-specified "charge to" information, or a data area containing up to 70 bytes of user-specified information to be included in the accounting record. DMKHVC validates the instruction operands, builds an account buffer (ACNTBLOK), and DMKACOQU is called to put the records in spool format. For additional information about this user option, see "DIAGNOSE Interface (DMKHVC)" under "Privileged Instructions."

Spooling Commands

The spooling commands provide an interface between the user, the system operator, and the spooling system. There are three types of spooling commands:

- Those that affect virtual devices
- Those that affect real devices
- Those that affect spool files that are queued within the system.

The commands that affect virtual devices are generally available to all system users, and a user can only affect the status of devices that are attached to his own virtual machine. Commands that affect the status of the real system's spooling devices can be used by the system operator only. Commands that affect closed spool files that are awaiting processing are generally available to all users, with some additional capabilities assigned to the system operator. For example, a user may alter the characteristics only of those files that have an owner's userid that matches his own, whereas the system operator may change any spool file in the system.

File States and Attributes

Each spool file in the system has a number of attributes that are assigned to it, either explicitly or by default, at the time that it is created. These attributes and their values are as follows:

- The filename and filetype can be either a 24-character field or two 8-character fields (one each for filename and filetype). Any of these fields can be replaced by a user-supplied value.
- The spool ID is a user-unique number between 1 and 9900. It is automatically assigned when the file is created (opened). The file's owner, the device type, and the ID number are specified. Usually, the userid defaults to the identification of the user issuing the given command. (When a class G user refers to a spool ID, that user's userid is the default. Privileged users, however, must specify a userid and a spool ID when referring to a specific spool file.)
- The number of logical records (cards or print lines) in the file is an integer between 1 and 16 million. For printer files, the record count also includes any immediate operation code space or skip CCWs.
- The originating user is the identification of the file's creator, if the file has been internally transferred from the originator's printer or punch to the new owner's card reader.
- The number of copies requested for an output file is between 1 and 255. Unless altered by the user or operator, it defaults to 1.
- The device type is used by DIAGNOSE for a file transferred to a reader to determine the virtual type of output device.

- CHARS for 3800 printer.
- FCB for 3800 printer.
- MODIFY for 3800 printer.
- FLASH for 3800 printer.

In addition to those attributes, a file that is queued for real output or virtual input always has a class associated with it. A class is a single alphanumeric character from A through Z or from 0 to 9. It controls both the real or virtual device on which the file will be printed, punched, or read, and the relative priority and sequence of output on the device. While each file is assigned a single class, each real spooling output device can be assigned from one to four classes. The device then processes only files that have a class attribute that corresponds to one of its own, and processes these files in the order that its own classes are specified.

For example, if a printer is assigned the classes A, D, 2, it processes any printer file with a class of A before it searches the printer output queue for a file with class D. All class D files are printed before class 2 files.

The output class for a file is assigned at the time the file is created and is the class that is associated with the virtual device that created it. While each real spooling device can have up to four classes, each virtual spooling device can have only one. When a user logs on to the system, the class associated with a device is the one defined in his directory entry for that device. However, he can alter this class at any time by the SPOOL command. As files are created and closed by a device, they take on the device's output class.

Each file also has an operator form and a user form. These are one-to eight-characters each. The user form is defined by the user. There is a system default user form for each spool file type (printer, punch, or console). The operator form is determined from the user form by a table lookup. The table is defined by the SYSFORM macro at CP system generation. The SYSFORM macro also specifies the default form.

The operator may start a spool device in MANUAL mode for a certain form. The system will then process only spool files with that operator form. AUTO mode is also available. In this mode, all spool files are grouped by form, and each group is processed with the operator being prompted each time a new group is processed.

Each printer, punch, and console file has a destination (DEST is the keyword) assigned to it. The default is OFF, unless a DESTination is set via the SPOOL, CHANGE, or CLOSE command. The DESTination value is a 1- to 8-character alphanumeric name that your installation assigns.

Individual spool files and virtual printers and punches have one DESTination. Real printer and punch spooling devices can have as many as four DESTinations. For example, a real printer can be started such that it handles files for DESTinations OFF, PRT1, PRT2, and PRT3.

The same operator and user commands that use CLASS and FORM can be used to control DESTination.

After they are closed and are awaiting output, their class, form, and destination can be changed by a CHANGE command issued either by the file's owner or by the system operator. The system operator can alter the system-generated output class(es) of a real output device by the START command.

Output files transferred to a user's virtual reader can also be controlled by class. If the receiving user has several readers, the input to each can be limited to files of a certain class. In addition, the ORDER command allows sequencing of input files by class as well as spoolid number.

Output priorities can also be managed by altering the hold status of a file. Individual users can alter the hold status with the CHANGE command, while the system operator can change (hold or free) the files of specific individual users.

SPOOL and CHANGE commands can be used to modify the CHARS, FCB, MODIFY, and FLASH attributes of a file or a virtual printer.

Imbedded LOAD FCB commands also affect the printer selection. A spool file with an imbedded LOAD FCB command is eligible only for output to a printer that can handle the FCB length, or a printer that will NOOP the LOAD FCB command (like the 1403).

The operator may start a 3211-type printer in DEFFCB mode to remove the FCB restriction. DEFFCB causes DMKRSP to NOOP all LOAD FCB commands that are found in the spool file. A file that would normally be incompatible can then be printed.

Virtual Device Spooling Commands

These commands affect the status of a user's virtual spooling devices:

Command	**Meaning**
CLOSE	Terminates spooling operations on a specified device. It clears the device of any pending interrupt conditions, and for output files, updates the tag record, completes and queues the file for real output. Optional operands allow the user to specify a filename and filetype, and to override for the given file any standard FORM, HOLD/NOHOLD or COPY operands set into the output device by the SPOOL command.
SPOOL	Establishes the file attributes that apply to files created on, or read by, the given device. It establishes the class and form that will be in effect, whether: files are to be automatically held, input files are to be saved or purged after reading, and output files are to be directed to the real system printers and punches or are to be transferred to a user's virtual reader. The SPOOL command also specifies 3800 attributes.

Note: The SPOOL command invokes the access control verification routine. See the VM/SP HPO Planning Guide and Reference *or* VM/SP HPO CP for System Programming *for more information. See also Appendix D for further information on the access control verification modules.*

Real Device Spooling Commands

The operator can use these commands to control the activity of the real spooling devices:

Command	**Meaning**
BACKSPAC	Backspaces an active spooling device for either a specified number of pages (printers only) or to the beginning of the file (printers or punches).
DRAIN	Stops the operation of a specified output or input device after it has finished processing the file on which it is currently working. A printer must be drained prior to the issuance of the LOADBUF command. Unit record devices are normally drained prior to system shutdown.
START	Restart a device after it has been drained. Options allow the operator to specify the spooling output class, form, destination, and mode for the output device type of virtual 3800 files to be printed and output separator records. For a 3800 printer, the IMAGE CHAR, FCB, and PURGE options may also be specified.
FLUSH	Immediately halts the output on the specified device and either flushes that copy of the file from the system, or puts it into the system hold status for future processing.
REPEAT	Supplements the number of copies requested by the user for the file when it was created. The operator can specify a number from 1 to 255 that is added to the number specified by the user.
LOADBUF	Loads the universal character set buffer of the FCB of the specified printer with the specified image. If requested, the system verifies the loading by printing its contents on the affected printer.
SPACE	Forces the output on the specified printer to be single spaced, regardless of the skipping or spacing commands specified by the file's creator.

Spool File Management Commands: The spooling commands alter the attributes and status of closed spool files that are queued and awaiting processing. When a command applies to an individual file, the device type (RDR, PUN, PRT) and the spoolid number must be provided to identify the file. In most commands requiring a spoolid, the keyword CLASS, FORM, or DEST followed by a valid spool class or form or the keyword ALL is an acceptable substitute for the spoolid number. This causes the command to be executed for all files of the given class, form, or device type. The userid is the identification of the user issuing the command, except that the system operator must explicitly supply the identification of the user whose files he wishes to affect or he must specify the keyword SYSTEM, which gives access to all files (valid for CHANGE, PURGE, ORDER, and TRANSFER commands also).

Command	**Meaning**
CHANGE	Changes the filename and filetype, the number of copies, the form, the destination, or the class of the specified file. The CHANGE command also specifies 3800 attributes. Any of the above attributes of a file can be determined via the QUERY command.
HOLD	Places, via the system operator, the specified file in a hold status. The file is not printed or punched and is released by the system operator. The operator can hold any user files by device type.
FREE	Opposite of the HOLD command. Allows a file or group of files that were previously held to become available for processing. However, the user cannot reset a hold that was set by the operator with the HOLD command.
PURGE	Removes unwanted spool files from the system before they are printed or punched.
ORDER	Reorders the input files in a virtual card reader. It can order files by identification number, by class, by form, or by any combination of the three.
SPTAPE	Dumps output spool files to tape or loads output spool files from tape.
TRANSFER	Transfers a closed spool file to a different queue (reader/printer/punch), to a different user, or both simultaneously. Also optionally reclaims spool files for the file's originator.

Notes:

1. *The (user-unique) spool ID will change when a file is transferred from one user to another.*

2. *The TRANSFER command invokes the access control verification routine. See the* VM/SP HPO Planning Guide and Reference *or* VM/SP HPO CP for System Programming *for more information. See also Appendix D for further information on the access control verification modules.*

 LY20-0897-7

Spool File Error Recovery

Unit Record I/O Errors

I/O errors on real spooling unit record devices are handled by a transient routine that is called by DMKIOS after it has sensed the unit check associated with the error on a spooling device. If appropriate, a restart CAW is calculated and DMKIOS is requested to retry the operation, in some cases waiting for a device end that signals that the failing device has been made ready after manual corrective measures have been taken. If, after retrying the operation, the error is unrecoverable, DMKIOS is informed that a fatal error has occurred. DMKIOS then unstacks the interruption, flagged as a fatal error, and passes control to real spooling executive. The routines that handle unstacked interruptions in real spooling execute only module operations that have been completed correctly or those that are fatal errors. If a fatal error is unstacked, the recovery mechanism depends on the operation in progress.

For fatal reader errors, processing of the current file is terminated and any portion of the file that has been read and stored on disk is purged. The owner of the file is not informed of the presence of a fractional part of the file in the system.

For fatal printer or punch errors, the SFBLOK for the partially completed file is re-queued to the appropriate output list and processing can be resumed by another available printer or punch, or can be deferred until the failing device is repaired.

In any case, the failing device is marked logically offline, and no attempt is made by the system to use it until the operator varies it back online via the VARY command.

If an invalid load module is specified for a 3800 printer (refer to DIAGNOSE code X'74'), the file involved is held or purged, and the printer queue is searched for the next file to print. In addition, the user and operator are sent a message describing the action.

DASD Errors during Spooling

DASD I/O errors for page writes are transparent to the user. A new page for the buffer is assigned, the file linkage pointers are adjusted, and the buffer is rewritten. The failing page is not de-allocated and no subsequent request for page space is granted access to the failing page. If an unrecoverable error is encountered while reading a page, processing depends on the routine that is reading the file. If the processing is being done for a virtual reader, the user is informed of the error and a unit check/intervention required condition is reflected to the reader. If the processing is being done for a real printer or punch, the failing buffer is put into the system hold status, and processing continues with the next file. In either case, the DASD page is not de-allocated and it is not available for the use of other tasks.

DASD Spool Space Exhausted

If the space allocated for paging and spooling on the system's DASD volumes is exhausted and more is requested by a virtual spooling function, the user receives a message and a unit check intervention required condition is reflected to the virtual output device that is requesting the space, the output file is automatically closed and it is available for future processing. The user can clear the unit check and periodically retry the operation which will start when space is free or completely restart later from the beginning of the job. If the task requesting the space is the real spooling reader task, the operator receives an error message and the partially complete file is purged. Any time the spooling space is exhausted, the operator is warned by a console message and alarm. However, the system attempts to continue normal operation.

Recovery from System Failure

Should the system suffer an abnormal termination, CP attempts to perform a warm start. Spool file and device data, as well as other system information is copied from real storage to warm start cylinders on DASD storage. If any virtual machines were enabled for VMSAVE, they are stored on DASD as specified in the DMKSNT module. When the system is reinitialized, the spool data and other system data is retrieved from the warm start cylinders and operation continues.

If the warm start data in real storage was damaged by the abnormal termination, the warm start procedure recognizes the situation and notifies the operator that a warm start cannot be performed. Another recovery method would be to attempt a checkpoint start.

The spool file recovery routines (DMKCKS, DMKCKT, DMKCKV) dynamically checkpoint on DASD storage; the status of all open reader files, the status of all closed output files, real spooling device data, and system hold queue information. This information is stored on checkpoint cylinders that are allocated, along with warm start cylinders, at system generation.

When a checkpoint (CKPT) start is requested, spool file and spooling device information is retrieved from the checkpoint cylinders. Spool file blocks are chained to their appropriate printer, punch, or individual reader chains, record allocation blocks are reconstructed, spooling device status is restored, and system hold queues are chained to the proper devices. System operation then continues.

If the checkpoint start procedure encounters I/O errors or invalid DASD data on the checkpoint cylinders, the operator is notified. The FORCE option of the checkpoint start performs all the checkpoint start functions except that invalid or unreadable files are bypassed. While this is at best a partial recovery, the only other alternative is a cold (COLD) start, where all spool file data is lost.

Recovery Management Support (RMS)

The machine check handler (MCH) minimizes lost computing time caused by machine malfunction. MCH does this by attempting to correct the malfunction immediately, and by producing machine check records and messages to assist the service representatives in determining the cause of the problem.

The channel check handler (CCH) aids the I/O supervisor (DMKIOS) to recover from channel errors. CCH provides the device-dependent error recovery programs (ERPs) with the information needed to retry a channel operation that has failed.

This support is standard and model-independent on the external level (from the user's point of view there are no considerations, at system generation time, for model dependencies).

System Initialization for RMS

DMKCPJ calls DMKIOEFL to initialize the error recording at cold start and warm start. DMKIOEFL gives control to DMKIOG to initialize the MCH area. A store CPU ID (STIDP) instruction is performed to determine if VM/SP HPO is running in a virtual machine environment, or running stand-alone on the real machine. If the system is running in a virtual machine, the version code is set to X'FF' by DMKPRV. If the version code returned is X'FF', the RMS functions are not initialized beyond setting the wait bit on in the machine check new PSW (virtual). This occurs because machine check interruptions are not reflected to any virtual machine. VM/SP HPO, running on the real machine, determines whether the virtual machine should be terminated.

If the version code is not X'FF', DMKIOG determines what channels are online by performing a Store Channel ID (STIDC) instruction and saves the channel type for each channel that is online. The maximum machine check extended logout length (MCEL) indicated by the Store CPU ID (STIDP) instruction is added to the length of the MCH record header, fixed logout length and damage assessment data field. DMKIOG then calls DMKFRE to obtain the necessary storage to be allocated for the MCH record area (MCRECORD), the CP execution block (CPEXBLOK), MCHAREA, and MCEL. The address of MCHAREA is put in the PSA (AMCHAREA). Pointers to MCRECORD and the CPEXBLOK are put in MCHAREA. DMKIOG puts the address of MCEL in control register 15. DMKIOG obtains the storage for the I/O extended logout area and initializes the logout area and the ECSW to ones. The I/O extended logout pointer is saved at location 172 and control register 15 is initialized with the address of the extended logout area. The length of the CCH record and the online channel types are saved in DMKCCH. It should be noted that the ability of a processor to produce an extended logout or I/O extended logout and the length of the logouts are both model- and channel-dependent. 308x processors do not store a machine check fixed logout, machine check extended logout, or a region code. If VM/SP HPO is being initialized on a Model 165 II or 168, the 2860, 2870, and 2880 standalone channel modules are loaded and locked by the paging supervisor and the pointers are saved in DMKCCH. If VM/SP HPO is being initialized on any other model, the integrated channel support is assumed; this support is part of the channel control subroutine of DMKCCH. Before returning to DMKIOE, the VM/SP HPO error recording cylinders are initialized. DMKIOE passes control back to DMKCPJ and control register 14 is initialized with the proper mask to record machine checks.

Overview of Machine Check Handler

A machine malfunction can originate from the processor, Vector Facility, real storage, or control storage. When any of these fails to work properly, the processor attempts to correct the malfunction.

When the malfunction is corrected, the machine check handler (MCH) is notified by a machine check interruption and the processor logs out fields of information in real storage, detailing the cause and nature of the error. The model-independent data is stored in the fixed logout area and the model-dependent data is stored in the extended logout area. The machine check handler uses these fields to analyze the error, format an error record, and write the record out on the error recording cylinder of SYSRES.

If the machine fails to recover from the malfunction through its own recovery facilities, the machine check handler is notified by a machine check interruption. An interruption code, noting that the recovery attempt was unsuccessful, is inserted in the fixed logout area. The machine check handler then analyzes the data and attempts to keep the system as fully operational as possible.

Recovery from machine malfunctions can be divided into the following categories: functional recovery, system recovery, operator-initiated restart, and system repair. These levels of error recovery are discussed in their order of acceptability, functional recovery being most acceptable and system repair being least acceptable.

Functional Recovery

Functional recovery is recovery from a machine check without adverse effect on the system or the interrupted user. This type of recovery can be made by processor retry, the ECC facility, or the machine check handler. Processor retry and ECC error correcting facilities are discussed separately in this section because they are significant in the total error recovery scheme. Functional recovery by MCH is made by correcting storage protect feature (SPF) keys and intermittent errors in real storage.

System Recovery

System recovery is attempted when functional recovery is impossible. System recovery is the continuation of system operations at the expense of the interrupted user, whose virtual machine operation is terminated. System recovery can only take place if the user in question is not critical to continued system operation. An error in a system routine that is considered to be critical to system operation precludes functional recovery and would require logout and a system dump followed by reloading the system.

Operator-Initiated Restart

When the errors may have caused a loss of supervisor or system integrity, the system is put into a disabled wait state. The operator is instructed to run the standalone error recovery (SEREP) program and then manually restart the system.

System Repair

System repair is recovery that requires the services of maintenance personnel and takes place at the discretion of the operator. Usually, the operator has tried to recover by system-supported restart one or more times with no success.

System/370 Recovery Features

The operation of the Machine Check Handler depends on certain automatic recovery actions taken by the hardware and on logout information given to it by the hardware.

Processor Retry

Processor errors are automatically retried by microprogram routines. These routines save source data before it is altered by the operation. When the error is detected, a microprogram returns the processor to the beginning of the operation, or to a point where the operation was executing correctly, and the operation is repeated. After several unsuccessful retries, the error is considered permanent.

ECC Validity Checking

ECC checks the validity of data from real and control storage, automatically correcting single-bit errors. It also detects multiple-bit errors but does not correct them. Data enters and leaves storage through a storage adapter unit. This unit checks each doubleword for correct parity in each byte. If a single-bit error is detected, it is corrected. The corrected doubleword is then sent back into real or control storage and on to the processor. When a multiple-bit error is detected, a machine check interruption occurs, and the error location is placed in the fixed logout area. MCH gains control and attempts to recover from the error.

Control Registers

Two control registers are used by MCH for loading and storing control information (see Figure 33). Control register 14 contains mask bits which specify whether certain conditions can cause machine check interruptions and mask bits which control conditions under which an extended logout can occur. Control register 15 contains the address of the extended logout area.

Word	Bits	Name of Field	Associated with
14	0	Check-stop control	Mch-Chk handling
14	1	Synchronous MCEL control	Mch-Chk handling
14	2	I/O extended logout control	Chan-Chk handling
14	4	Recovery report mask	Mch-Chk handling
14	5	Degradation report mask	Mch-Chk handling
14	6	External damage report mask	Mch-Chk handling
14	7	Warning mask	Mch-Chk handling
14	8	Asynchronous MCEL control	Mch-Chk handling
14	9	Asynchronous fixed log control	Mch-Chk handling
15	8-28	MCEL address	Mch-Chk handling

Figure 33. RMS Control Register Assignments

Machine Check Handler Subroutines

VM/SP HPO Machine Check Handler module (DMKMCH) consists of the following functions:

- Initial analysis subroutine
- Main storage analysis subroutine
- Storage protect feature analysis subroutine
- Recovery facility mode switching
- Operator communication subroutine
- Virtual user termination subroutine
- Soft recording subroutine
- Buffer error subroutine
- Termination subroutine.

Initial Analysis Subroutine

The initial analysis subroutine of DMKMCH receives control by a machine check interruption. To minimize the possibility of losing logout information by recursive machine check interruptions, the machine check new PSW gives control to DMKMCH with the system disabled for further interruptions. There is always a danger that a machine malfunction may occur immediately after DMKMCH is entered and the system is disabled for interruption. Disabling all interruptions is only a temporary measure to give the initial analysis subroutine time to make the following emergency provisions:

- It disables for soft machine check interruptions. Soft recording is not enabled until the error is recorded.
- It saves the contents of the fixed and extended logout areas in the machine check record.

- It alters the machine check new PSW to point to the term subroutine. The term subroutine handles second machine check errors.

- It enables the machine for hard machine check interruption.

- If a virtual user was running when the interruption occurred, the running status (GPRs, FPRs, PSW, M.C. old PSW, CRs, etc.) is saved in the user's VMBLOK.

- It initially examines the machine check data for the following error types:

  MCIC=ZERO
  PSW invalid
  System damage
  Timing facilities damage

  The occurrence of any of these errors is considered uncorrectable by DMKMCH; the primary system operator is informed, the error is formatted and recorded, and the system enters a wait state, code 001 or 013.

- If none of the above errors are present, it checks for a channel inoperative error on a 303x processor. If such an error is detected, DMKACRCT will be called to attempt (a) to recover the failing channel(s) and (b) if the channel(s) is still not operational mark the channel(s) offline and attempt to continue system operation.

- If the instruction processing damage bit is on, it tests for the following types of malfunctions:

  - Multiple-Bit Error in Main Storage – Control is given to the main storage analysis subroutine.

  - SPF Key Error – Control is given to the SPF analysis subroutine.

  - Retry failed – If the processor was in supervisor state the error is considered uncorrectable and the system is terminated. If the processor was in problem state, the virtual machine is reset or terminated and the system continues operation.

- If processor retry or ECC was successful on a soft error, control is given to the soft recording subroutine to format the record, write it out on the error recording cylinder, and update the count of soft error occurrences.

- If external damage was reported, control is given to the soft recording subroutine to format the record and write it out on the error recording cylinder.

Main Storage Analysis Subroutine

The main storage analysis subroutine is given control when the machine check interruption was caused by a multiple-bit storage error. An initial function points the machine check new PSW to an internal subroutine to indicate a solid machine check, in case a machine check interruption occurs while exercising main storage.

Damaged storage areas associated with any portion of the CP nucleus itself cannot be refreshed; multiple-bit storage errors in CP cause the system to be terminated. An automatic restart reinitializes the system.

If the damage is not in the CP nucleus, main storage is exercised to determine if the failure is solid or intermittent. Multiple-bit ECC storage errors on a 303x processor are always treated as solid errors. If the failure is solid, the 4K page frame is marked unavailable for use by the system. If the failure is intermittent, the page frame is marked invalid. The change bits associated with the damaged page frame are checked to determine if the page had been altered, by the virtual machine. If no alteration had occurred, CP assigns a new page frame to the virtual machine and a backup copy of the page is brought into storage the next time the page is referenced. If the page had been altered VM/SP HPO resets or terminates the virtual machine, clears its virtual storage, and sends an appropriate message to the user. Normal system operation continues for all other users.

Storage Protect Feature Analysis Subroutine

The storage protect feature analysis subroutine gets control when the machine check interruption was caused by a storage protect feature error. An initial function points the machine check new PSW to an internal subroutine if a machine check interruption occurs during testing and validation. The storage protect feature analysis routine then determines if the error was associated with a failure in virtual machine storage or in the storage associated with the control program.

A storage protect feature error associated with VM/SP HPO is a potentially catastrophic failure. Namely, VM/SP HPO always runs with a PSW key of zero, which means that the storage protect feature key in main storage is not checked for an out-of-parity condition. The storage protect feature analysis subroutine exercises all 16 keys in the failing storage 2K page frame. If a storage protect feature machine check occurs in exercising the 16 keys 5 times each, the error is considered solid and the operating system is terminated with a system shutdown. If a storage protect feature machine check does not occur, the machine check is considered intermittent. The zero key is restored to the failing 2K page frame and this is transparent to the virtual machine.

If a storage protect feature machine check occurs that is associated with a virtual machine, the storage protect feature analysis subroutine exercises all 16 keys in the failing storage 2K page frame. If a storage protect feature machine check does not occur, the machine check is intermittent and the SWPTABLE for the page associated with the failing storage address is located. The storage key for the failing 2K storage page frame is retrieved from the SWPTABLE and the change and reference bits are set on in the storage key. The storage key is then stored into the affected failing storage 2K page frame. If a storage protect feature machine check occurs in exercising the 16 keys 5 times each, then the machine check is considered solid and the following actions are taken. (1) The virtual machine is selectively reset or terminated by the virtual machine termination subroutine; (2) The 4K page frame associated with the failing address is removed as an available system resource. This is accomplished by locating the CORTABLE for the defective page and altering the CORFPNT and CORPBPNT pointers to make the page unavailable to the system. The CORDISA bit in this CORTABLE is set on to identify the reason for the status of this page in a system dump.

Recovery Facility Mode Switching

The recovery facility mode switching subroutine (DMKMCIMS) allows the service representative to change the mode that processor retry and ECC recording are operating in. This subroutine receives control when a user with privilege class F issues some form of the SET command with the MODE operand. A check is initially made to determine if this is VM running under VM. If this is the case, the request is ignored and control is returned to the calling routine. For the format and usage of the SET command with the MODE operand, refer to the *VM/SP HPO Operator's Guide.*

Operator Communication Subroutine

The operator communication subroutine is invoked when the integrity of the system has degraded to a point where automatic shutdown and reload of the system has been tried and was unsuccessful, or could not be attempted due to the severity of the hardware failure. A check is first made to determine if the system operator is logged on as a user, then a check is made to determine if the system operator is disconnected. If either of these checks is not affirmative a message cannot be issued directly to the system operator. A LPSW is performed to place the processor in a disabled wait state with a recognizable wait state code in the processor instruction counter.

Virtual User Termination Subroutine

The virtual machine termination subroutine selectively resets or terminates a virtual user whose operation has been interrupted by an uncorrectable machine check. First, the machine is marked nondispatchable to prevent the damaged machine from running before reset or termination is performed. The machine check record is formatted and DMKIOEMC is called to record the error. Then the user is notified by a call to DMKQCNWT that a machine check has occurred and that his operation is terminated. The primary system operator is notified of the virtual user termination by a message issued by a call to DMKQCNWT. If the virtual machine is running in the virtual=real area, DMKUSO is called to log the virtual machine off the system and to return the storage previously allocated to the virtual machine and to clear any outstanding virtual machine I/O requests. The HOLD option of LOGOFF is invoked to allow a user on a dial facility to retain the connection and thus permit LOGON without re-establishing the line connection. However, if the virtual machine is running in the virtual area, and DMKCFM is then called to put the virtual machine in console function mode, the user must re-initialize the system to commence operation.

Soft Recording Subroutine

The soft recording subroutine performs two basic functions:

- Formats a machine check record and calls DMKIOEMC to record the error on the error recording cylinder.
- Maintains the threshold for processor retry and ECC errors and switches from recording to quiet mode when the threshold value is exceeded. To accomplish this, a counter is maintained by DMKMCH for successful processor retry and corrected ECC events.

Processor Retry Recording Mode: Recording mode (bit 4 of control register 14 set to one) is the initialized state, and normal CP operating state for processor retry errors. Recording mode may also be entered by use of the CP SET command. When 12 soft machine checks have occurred, the soft recording subroutine switches the processor from recording mode to quiet mode. For the purpose of model-independent implementation this is accomplished by setting bit 4 of control register 14 to zero. Because in quiet mode no soft machine check interruptions occur, a switch from quiet mode to recording mode can be made by issuing the SET MODE RETRY|MAIN RECORD command. While in recording mode, corrected CPU RETRY|MAIN reports are formatted and recorded on the error recording cylinders, but the primary systems operator is not informed of these occurrences.

Processor Retry Quiet Mode: Quiet mode (bit 4 of control register 14 set to 0) can be entered in one of two ways: (1) when 12 soft machine checks have occurred, or (2) when the SET MODE RETRY QUIET command is executed by a class F user. In this mode, both processor retry and ECC reporting are disabled. The processor remains in quiet mode until the next system IPL (warm start or cold start) occurs or a SET MODE RETRY|MAIN RECORD command is executed by a class F user. SET MODE MAIN is treated as invalid on 303x, 308x and 3090 processors.

ECC Recording Modes: To achieve model-independent support, RMS does not set a specific mode for ECC recording. The mode in which ECC recording is initialized depends upon the hardware design for each specific processor model. For the IBM System/370 Models 158, 168, 303x, 308x, and 3090, the hardware-initialized state (therefore the normal operational state for CP) is quiet mode. For the IBM System/370 Models 155II and 165II, the hardware initialized state (the normal operational state for CP) is record mode. An automatic restart incident due to a system failure does not reset the ECC recording mode in effect at the time of failure.

The change from record to quiet mode for ECC recording can be initiated in either of the following ways: (1) by issuing the SET MODE {MAIN|RETRY} QUIET command, or (2) automatically whenever 12 soft machine checks have occurred. For the purpose of model-independent implementation, this occurs by setting bit 4 of control register 14 to zero.

The change from quiet to record mode for ECC recording can be accomplished by use of the SET MODE MAIN RECORD command. This recording mode option is for use by maintenance personnel only. It should be noted that processor retry is placed in recording mode if it is not in that state when the SET MODE MAIN RECORD command is issued.

While in recording mode, corrected ECC reports are formatted and recorded on the error recording cylinder, but the primary systems operator is not informed of these incidents.

Buffer Error Subroutine

On processor models equipped with a high-speed buffer (155-II, 158, 165-II, 168, 303x, 308x) or a data lookaside table (DLAT) (165-II, 168, 303x, 308x, 3090) the deletion of buffer blocks because of hardware failure is reported via a degradation report machine check interruption. MCH enables itself for degradation report machine check interruptions at system initialization by setting bit 5 of control register 14 to 1. If a machine check interruption occurs that indicates high-speed buffer or DLAT damage, MCH formats the record and calls DMKIOEMC to record it on the error recording cylinder, informs the primary systems operator of the failure, and returns control to the system to continue normal operation.

Termination Subroutine

The termination subroutine is given control if a hard machine check interruption occurs while DMKMCH is in the process of handling a machine check interruption. Note that soft error reporting is disabled for the entire time that MCH is processing an error.

An analysis is performed of the machine check interruption code of the first error to determine if it was a soft error. If it was, the first error is recorded, the system status is restored and control is restored to the point where the first error occurred. If the first error was a hard error, the operator communication subroutine is given control to issue a message directly to the system operator, and to terminate CP operation.

Overview of Channel Check Handler

The channel check handler (CCH) aids the I/O supervisor in recovering from channel errors and informs the operator or service representative of the occurrence of channel errors.

CCH receives control from the I/O supervisor when a channel data check, channel control check, or interface control check occurs. CCH produces an I/O error block (IOERBLOK) for the error recovery program and a record to be written on the error recording cylinder for the system operator or service representative. The operator or service representative may obtain a copy of the record by using the CMS CPEREP command. A message about the channel error is issued to the system operator each time a record is written on the error recording cylinder.

When the I/O supervisor program detects a channel error during routine status examination following an SIO, TIO, HIO, or an I/O interruption, it passes control to the channel check handler (DMKCCH). DMKCCH analyzes the channel logout information and constructs an IOERBLOK and, if the error is a channel control or interface control check, an ECSW is constructed and placed in the IOERBLOK. The IOERBLOK provides information for the device-dependent error recovery procedures. DMKCCH also constructs a record to be recorded on the error recording cylinder. Normally, DMKCCH returns control to the I/O supervisor after constructing an IOERBLOK and a record. However, if DMKCCH determines that system integrity has been damaged (system reset or invalid unit address, etc.), then CP operation is terminated. CP termination causes DMKCCH to issue a message directly to the system operator and place the processor in a disabled wait state with a recognizable wait code in the processor instruction counter.

Normally, when DMKCCH returns control to the I/O supervisor, the error recovery program for the device which experienced the error is scheduled. When the ERP receives control, it prepares to retry the operation if analysis of the IOERBLOK indicates that retry is possible. Depending on the device type and error condition, the ERP either effects recovery or marks the event fatal and returns control to the I/O supervisor. The I/O supervisor calls the recording routine DMKIOE to record the channel error.

The primary system operator is notified of the failure, and DMKIOE returns control to the system and normal processing continues.

If the channel check is associated with an I/O event initiated by a SIO in a virtual machine, the logout is reflected to the virtual machine in one of two ways, depending upon whether the channel check occurred at SIO time or later in an interrupt. If it occurred at SIO time, then DMKVSI (or occasionally DMKVIO) calls upon DMKCCHRF to reflect the logout. If it occurred in an I/O interrupt, the dispatcher notices the channel check as it is reflecting the I/O interrupt to the virtual machine, and so, at that time, DMKDSP calls upon DMKCCHRF to reflect the logout.

Channel Control Subroutine

Control is passed to the channel control subroutine of DMKCCH after a SIO with failing status stored, or an I/O interrupt because of a channel control check, interface control check, or channel data check.

If "logout pending" is indicated in the CSW, the CP termination flag is set. The existence of real device blocks (RCHBLOK, RCUBLOK, RDEVBLOK), for the failing device address, is determined by a call to DMKSCNRU and an indicator is set if they do exist. An indicator is also set if the IOBLOK for the failing device address exists. A call to DMKFREE obtains storage space for the channel check record and the channel control subroutine builds the record. If the indicators show that the real device blocks and the

IOBLOK exist, a call to DMKFREE obtains storage space and the channel control subroutine builds the I/O error block (IOERBLOK); if these blocks do not exist, the IOERBLOK is not built. The IOERBLOK is used for two purposes:

1. The device-dependent error recording program (ERP) uses the IOERBLOK to attempt recovery on CP-initiated I/O events. If the I/O events that resulted in a channel check are associated with a virtual machine, the I/O fatal flag is set in the IOBLOK and the virtual machine is reset, cleared, and put into CP read status. The length and address of the channel check record is placed in the IOERBLOK and the IOERBLOK is chained off the IOBLOK.

2. DMKIOECC uses the IOERBLOK to record the channel check record on the error recording cylinders.

The channel control subroutine gives control to a channel-dependent error analysis routine to build or save the extended channel status word (ECSW). When the channel control subroutine regains control, eight active addresses are saved in the channel check record.

If the CP termination flag is set, the I/O extended logout data from the channel check record is restored to main storage for use by SEREP. If the system operator is both logged on as a user and connected to the system, a message is sent to him advising him of the channel error. A LPSW is then executed to place the processor in a disabled wait state with a wait state code of 002 in the processor instruction counter.

If the CP termination flag is not set, a check is made to determine if an IOERBLOK was built by the channel control subroutine.

If an IOERBLOK was not built, a CPEXBLOK is stacked to call DMKIOECC to record the channel check record on the error recording cylinders and send the system operator a message informing him of the error. If channel termination is set, DMKACRCT is called to attempt to recover the failing channel with a CLRCH operation. If the CLRCH recovers, the failing channel, or if the system can continue operation with the failing channel marked offline, DMKACR returns to DMKCCH. Otherwise a wait state X'002' is loaded.

If an IOERBLOK was built, control is returned to DMKIOS, which calls the appropriate ERP. Whether or not recovery is successful, DMKIOS eventually calls DMKIOE to record the channel check record. DMKIOE examines the status of the in CSW error in the IOERBLOK to determine if it was a channel error; if so, it finds the length and pointer to the channel check record and records the error on the error recording cylinder. If this was not a channel error, DMKIOE continues normal processing.

Individual Routines

A separate channel error analysis routine is provided for each type of channel for which DMKCCH can be used. The purpose of these routines and the channel control subroutine is to analyze the channel logout to determine the extent of damage and to create a sequence and termination code to be placed in the ECSW in the IOERBLOK. At system initialization, the correct model dependent channel recovery routine is loaded and the storage necessary to support the routine is allocated. The model-dependent error analysis subroutines and routines and their functions are as follows:

Integrated Channels (Models 135, 135-3, 138, 145, 145-3, 148, 155II, 158, 3031, 3032, 3033, 3081, 3090 and 43xx Processors)

Since all of these systems have integrated channels one common subroutine is used to handle all of these processor types. This subroutine:

- Indicates CP termination if the ECSW is not complete or the reset codes are invalid
- If the channel has been reset or if the error was an I/O interface inoperative condition on a 303x processor, indicate channel termination
- Moves the ECSW to the IOERBLOK
- Moves the hardware stored unit address and the I/O extended logout to the channel check record
- Sets the I/O extended logout area and ECSW area to ones
- Returns control to the channel control subroutine.

2860 Channel (Models 165 II and 168)

The 2860 logout area is checked to determine if a complete logout exists; if not, CP termination is necessary.

A check is made in the logout area for validity of the CSW fields and bits are set in the channel check record's ECSW field to indicate bad fields.

The channel logout is then checked and sequence codes are set based on the presence of a channel control check, or an interface control check. If a channel control check is present, the codes set are determined through parity. The count determines if parity is good and sets a resultant condition code.

The logout area is examined to ensure that the unit address has valid parity and is the same address passed by DMKIOS. If so, the unit-address-valid bit in the ECSW is set. If the unit address is not valid, the unit-address-valid bit is reset to indicate the invalid condition.

The ECSW field in the channel check record is moved to the IOERBLOK, if one exists.

After completing the ECSW the 2860 routine moves the 2860 I/O extended logout into the channel check record, set the I/O extended logout area to ones, and returns to the channel control subroutine.

2870 Channel (Models 165 II and 168)

If the channel failed to log out completely, at least part of the logout area is all ones. If a fullword of ones is found, a CP termination condition exists.

A check is made in the logout area for valid CSW fields, and bits are set in the channel check record's ECSW field to indicate bad fields.

The termination and sequence codes are set depending on the presence of an interface control check or channel control check. If a channel control check is present, the codes set are determined through parity, count, and/or data transfer checks. For the 2870, parity can be determined directly from the channel logout.

The logout area is also examined to ensure valid parity in the unit address and to ensure that the address is the same as that passed to DMKCCH by DMKIOS. If so, the unit-address-valid bit in the ECSW is set.

The third word of the logout area is also analyzed for type II errors. If a type II error is found, a CP termination condition exists.

The ECSW field in the channel check record is moved to the IOERBLOK, if one exists.

Before returning to the channel control subroutine, the 2870 routine moves the 2870 I/O extended logout into the channel check record and sets the I/O extended logout area to ones.

2880 Channel (Models 165 II and 168)

This routine analyzes 9 words of the 28-word logout.

The 2880 analysis routine handles channel data checks, interface control checks, and channel control checks.

Termination code 3 (system reset) is not set in the ECSW because the 2880 channel does not issue system reset to the devices. Retry codes of 0 to 5 are possible.

Note: *There are several catastrophic conditions under which the CP termination flag can be set, in the 2880 analysis routine. They are:*

- *The channel did not complete the logout.*
- *The CSW is not reliable.*
- *The unit address in the I/O interruption device address field is not correct.*

Only a channel check record is needed if the channel has recognized an internal error and has recovered from it without any damage. No recovery action is necessary in these cases.

If the channel address in the I/O interruption device address field does not match the channel address in the logout, a CP termination condition exists.

If the channel was doing a scan and the unit control word had a parity check a CP termination condition exists. If there was no parity check, there was no damage during the scan and only a channel check record is required.

Depending on the sequence the channel has entered, the termination and sequence codes are set; command address, unit address, and unit status validity is determined; and the sequence code is set valid. The ECSW field in the channel check record is moved into the IOERBLOK, if one exists.

Before returning to the channel control subroutine, the 2880 routine moves the I/O extended logout into the channel check record and sets the I/O extended logout area to ones.

Error Recording Interface for Virtual Machines

The error recording interface provides a means of recording errors encountered by operating systems running in a virtual machine under VM/SP HPO. An SVC 76 issued by a virtual machine is used to signal CP that error recording is required. The SVC interruption handler in DMKSVD examines general registers 0 and 1 to determine if valid parameters have been passed. If valid parameters are not found, the SVC is reflected back to the virtual machine and no recording takes place. If valid parameters are passed, a pageable routine (DMKVER) processes the error record.

DMKVER validates the record passed by the virtual machine. If invalid conditions are found, no recording takes place. Control is returned to the SVC interruption routine in DMKPSA to reflect the SVC to the virtual machine as an SVC interruption. The action taken by the virtual machine is dependent on the operating system running in the virtual machine, not VM/SP HPO. If the record is valid, it is modified by changing virtual information to real. The actual recording is accomplished by using the modules DMKIOE and DMKIOF.

Control is then returned to the instruction following the SVC 76 rather than reflecting the SVC. This eliminates the duplication of error recording in VM/SP HPO and the operating system in the virtual machine. If DMKVER determines that the recording represented a permanent I/O error, a message is sent to the primary system operator.

Error Recording and Recovery

The error recording facility is made up of six modules. Four modules (DMKIOE, DMKIOF, DMKIOC, and DMKIOJ) are resident and the other two (DMKIOG and DMKERP) are pageable.

The error recording modules record temporary errors (statistical data recording) for CP generated I/O except for DASDs with a buffered log.

The error recording routines record: unit checks, statistical data counter overflow records, selected temporary DASD errors, machine checks, channel checks, and hardware environmental counter sense data on the error recording cylinders of the system resident device in a format suitable for subsequent processing by the CPEREP command (DMSIFC). The recorder asynchronously updates the statistical data counters for supported devices. The recorder also initializes the error recording cylinders at IPL if they are in an unrecognizable format.

To prevent the loss of error information in the event of a CP abend or a SHUTDOWN command, DMKCKP calls DMKERP to record OBR, MDR, machine check, and channel check errors that are queued to be processed.

Note: You should run CPEREP when the error cylinders are 90% full. If the cylinder becomes full, errors will be lost and error records that are kept will use free storage below the 16Mb line. If CP abends with a DAS001, and CPEREP has not been run, error information for the 3880 Model 11 or Model 21 can still be retrieved using the DAS001 technique described in the "3880 Model 11 and Model 21 Storage Subsystem" section of the VM/SP HPO OLTSEP and Error Recording Guide.

When the recorder is entered from DMKIOS, it is entered at DMKIOERR. This entry is used for unit checks and channel data checks. A test is made of the failing CSW (located in the IOERBLOK) to see if the error was a channel error. If it was, control is passed to the routine for recording channel checks.

The IOERBLOK sense data, IOBLOK flags, and VMBLOK privilege class are examined to determine if the error should be recorded.

Error Record Writing

After an error record is formatted, it is added to the error recording area using DMKRPAGT and DMKRPAPT. The error recording area has page-sized records (4096 bytes). Each page contains a header (8 bytes) which signifies: the cylinder and page number of the page on a CKD DASD, or page number of the page on an FBA DASD (4 bytes), the next available space for recording within page (2 bytes), a page-in-use indicator (1 byte), and a flag byte. Each record within the page is recorded with a 4-byte prefix.

If an error record is too large to be added into a page, a new page is retrieved, updated with record, and placed back on the error recording cylinder with the paging routines.

The area to be used for error recording is specified by the installation or system programmer at system generation time. For CKD DASDs, an integral number of cylinders (from two to nine) is specified. For FBA DASDs any number of pages can be specified. Generally, fifty pages or more is an appropriate amount. Errors are recorded in the order in which they occur. If the error recording area become 90 percent full, a message is issued to the operator using DMKQCNWT to warn him of the condition. If the area becomes full, another message is issued to inform the operator and recording is stopped.

On the 303x processors, frame records are read from all accessible SRF devices and written on the error recording area during initialization if no records exist (as after a CPEREP CLEARF operation).

If a channel check error is to be recorded, the recorder is entered at DMKIOERR or DMKIOECC. The channel check handler determines the entry. A channel check error record is formatted.

A machine check enters at DMKIOEMC. Pointers are passed from the machine check handler in registers 6 and 7 to locate a buffer where the machine check record and length are saved. A machine check error record is recorded with the saved machine check logout and additional information. The machine check error record is written onto the error recording area by using the paging routines.

Hardware environmental counter records are formed using routine DMKIOEEV. This routine is scheduled by DMKIOS after control is returned from the ERP. Sense data information is stored in the IOERBLOK by the ERP. The record formed is called a nonstandard record.

Clear and Format Recording Area

DMKIOEFM is called by DMSIFC (CPEREP command) via a DIAGNOSE instruction. DMKIOEFM is invoked to reset the specified error recording cylinders (if CLEAR, CLEARF, or ZERO=Y was specified). The clear is performed by resetting each page-header, space-available field. Pointers in storage are then updated to address the first available page in the error recording area. Control is then returned to the calling routine. For details on the CPEREP command and EREP execution, refer to the *VM/SP HPO OLTSEP and Error Recording Guide* and EREP publications.

CLEARF on a 303x processor clears the area, then causes the frame records to be read from each SRF device specified at system generation on the RIOGEN macro.

Find First Recording Cylinder at IPL

DMKIOEFL is called by DMKCPI to find the first available page that can be used for error recording. The paging routines, DMKRPAPT and DMKRPAGT, are used to read the error recording pages (4096-byte records).

As each page record is read, it is examined to see if this record is the last recorded. If so, a pointer in storage is saved so recording can continue on that page record. Control is then returned to the caller. If any error recording page is in an unrecognizable format, the error recording area is automatically reformatted by CP.

DASD Error Recovery, ERP (DMKDAS or DMKDAD for CKD or DMKDAU for FBA)

Error recovery is attempted for CP-initiated I/O operations to its supported devices and for user-initiated operations to CP-supported devices that use a DIAGNOSE interface. The primary control blocks used for error recovery are the RDEVBLOK, the IOBLOK and the IOERBLOK. In addition, auxiliary storage is sometimes used for recovery channel programs and sense buffers.

The initial error is first detected by the I/O interruption handler which performs a SENSE operation if a unit check occurs. Unit check errors are then passed to an appropriate ERP. If a channel check is encountered, the channel check interruption handler determines whether or not retry is possible and passes control to an ERP through the I/O interruption handler. DASD errors are processed as described below.

Channel Errors

- I/O interface inoperative on a 303x processor is reflected to the virtual machine if the channel is dedicated. Otherwise, a call is made to DMKACR to attempt to recover the failing channel. If recovery from the channel check is not possible, a wait state X'002' occurs.
- Channel control check is treated as seek check. It is retried 10 times.
- Interface control check is treated as seek check. It is retried 10 times.
- Channel data check is treated as data check. It is retried 10 times.

Unit Check Errors

Equipment check: Retry the operation 10 times for 3330, 3340, 3375, 3380, 3350, 2305, and FBA devices. If Alternate Interface Disabled is also on, retry the operation one time. For FBA devices, if the error is also permanent, the command is not retried.

No record found (CKD devices only): Execute a READ HOME ADDRESS and check home address against seek address. If they are the same, consider the error permanent. If not, return to caller.

Seek check (CKD devices only): Retry the operation 10 times except that 3375/3380/3330/3350 seek checks are retried by hardware.

Intervention required: Issue a message to console and wait for solicited device end. This procedure is repeated once.

Bus out check: One retry of the operation.

Data checks: For the 2305/3340, retry the operation 10 times. For the 3375/3380/3330/3350/FBA, the operation is retried by hardware.

Overrun: Retry the operation 10 times. For FBA devices, the operation is retried 10 times, unless the error is also permanent.

Missing address marker: Retry the operation 10 times.

Command reject: The command is not retried.

File Protect: The command is not retried.

Chaining check: Test for command reject. If not present, retry the operation 10 times.

Environmental data present: Issue a BUFFER UNLOAD command and retry the operation.

Track condition check (CKD devices only): On CP I/O and Diagnose I/O, when a track condition check is received from a device for which CP does not provide alternate track software recovery, the condition is treated as a permanent error. CP does provide alternate track support for other devices; this support is described in the section "Alternate Track Recovery, ERP (DMKTRK)."

Check Data (FBA devices only): The command is not retried.

The error recovery routine keeps track of the number of retries in the IOBRCNT field of the IOBLOK. This count determines if a retry limit has been exceeded for a particular error. On initial entry from DMKIOS for an error condition, the count is zero. Each time a retry is attempted, the count is increased by one.

The ERP preserves the original error CSW and sense information by placing a pointer to the original IOERBLOK in the RDEVBLOK. Additional IOERBLOKs, which are received from DMKIOS on failing restart attempts, are discarded. The original IOERBLOK is thus preserved for recording purposes.

If after a specified number of retries, DMKDAS or DMKDAD (for CKD devices) or DMKDAU (for FBA devices) fails to correct the error, the operator may or may not be notified of the error. Control is returned to DMKIOS. DMKIOS is notified of the permanent error by posting the IOBLOK (IOBSTAT=IOBFATAL). The error is recorded via DMKIOS by DMKIOERR, if DMKDAS or DMKDAU and DMKIOE determine that the error warrants recording.

If the error is corrected by a restart, the temporary or transient error is not recorded. Control is returned to DMKIOS with the error flag off.

Before returning control to DMKIOS on either a permanent error or a successful recovery, the ERP frees all auxiliary storage gotten for recovery CCWs, buffers, and IOERBLOKs.

The DMKIOS interface with the ERP uses the IOBSTAT and IOBFLAG fields of the IOBLOK to determine the action required when the ERP returns to DMKIOS.

When retry is to be attempted, the ERP turns on the restart bit of the IOBFLAG field. The ERP bit of the IOBFLAG field is also turned on to indicate to DMKIOS that the ERP wants control back when the task has finished. This enables the ERP to receive control even if the retry was successful and allows the freeing of all storage gotten for CCWs and temporary buffers. The IOBRCAW is set to the recovery CCW string address.

In handling an intervention-required situation, the ERP sends a message to the operator and then waits for the device end to arrive. This is accomplished by a return to DMKIOS with the ERP bit in the IOBFLAG field set on and the IOBSTRT bit in the IOBFLAG field set off. When the device end interruption arrives, the original channel program which was interrupted is then started.

The ERP flags of the IOERBLOK are also used to indicate when special recovery is being attempted. For example, a READ HOME ADDRESS command when a no record found error occurs.

The other two indicators are self-explanatory and are explained in Figure 34.

Field			
IOBFLAG IOBERP	**IOBFLAG IOBRSTRT**	**IOBSTAT IOBFATAL**	**Action To Be performed by DMKIOS**
1	0	0	Return control when solicited device end arrives
1	1	0	Restart using IOBRCAW
0	0	1	Permanent I/O error
0	0	0	Retry successful

Figure 34. Summary of IOB Indicators

If the error is uncorrectable or intervention is required, the ERP calls DMKMSW to notify operator. The specific message is identified in the MSGPARM field of the IOERBLOK.

Alternate Track Recovery, ERP (DMKTRK)

The software alternate track recovery support described in the following paragraphs applies only to the 3340/3344 disk. For 3380, 3330, and 3350 disks no software support is needed since the hardware performs alternate track recovery. No support is needed for the 2305 drum since the CE is able to rewire the device to use spare tracks in place of defective tracks. Track condition checks from any device type are reflected back to the virtual machine.

Overview of 3340 Alternate Track Support

The 3340 alternate track support applies to CP I/O, to Diagnose I/O (thereby giving alternate track support to CMS), and to SIO executed in a virtual machine. For CP I/O and Diagnose I/O, the alternate track recovery support essentially consists of directing (seeking) an interrupted channel program to an alternate track and restarting it. Later, in some cases, the interrupted channel program is directed back to the original cylinder and restarted there. For SIO in a virtual machine, the operating system in the virtual machine provides its own error recovery when CP reflects a track condition check to the virtual machine.

On the 3340 disk, alternate tracks are assigned in the conventional alternate tracks cylinders at the high end of the real disk, not in the last cylinder of each minidisk. Therefore a virtual machine may need to seek outside of its minidisk extent. This occurs when an operating system in a virtual machine performs its own error recovery following a track condition check. So for SIO issued from a virtual machine, CP's alternate track support must permit the virtual machine to escape from the confines of its minidisk to get to the alternate tracks assigned to the defective tracks of that minidisk. Yet at the same time CP must still prevent the virtual machine from accessing other tracks that it does not own.

Since alternate tracks are assigned only in the conventional alternate tracks cylinders at the high end of the real disk, CP does not apply minidisk cylinder relocation values to a virtual machine's channel commands that reference alternate tracks. Similarly, CP does not unrelocate alternate track CCHH addresses returned by read home address, by read record zero, in sense information, or for error recording.

Alternate Track Hardware Operation and Implications

The home address record (HA) on any track contains a flag byte with two bits that are involved in alternate track assignments. One bit, when set to one, indicates that the track is defective and that the track should have (and ordinarily does have) an alternate track assigned. The count field of record zero of a track with this bit set should point to (have the CCHH address of) the assigned alternate track. The second bit in the flag byte, when set to one, indicates that the track in which it appears is an assigned alternate track. The count field of record zero of an assigned alternate track should point back to (have the CCHH address of) the flagged defective track that it is assigned to.

Before using the pointer in record zero of a flagged track to get to the corresponding alternate, it is considered good form for an operating system to check the pointers both ways to see that each points to the other. CP performs two-way checks of the pointers for seeks to an alternate track initiated by Diagnose or by SIO in a virtual machine. For its own I/O, CP uses the forward record zero pointer without performing a two-way check. Performing a two-way check would decrease performance and should not be necessary since all of the record zero pointers were checked in both directions by the Format/Allocate program (DMKFMT) when the CP-owned disk was originally formatted.

Note: The DASD Dump/Restore (DDR) program also checks the record zero pointers both ways when a tape is restored to a disk.

Except for those channel commands that deal specifically with the home address and record zero, any attempt to search or read or write on a track that is flagged as defective results in a unit check with "track condition check" indicated in the sense data.

Operations on an assigned alternate track can also result in a unit check with "track condition check" indicated in the sense data. But in this case it occurs when an attempt is made to *leave* the assigned alternate track, not when the operation is reading or writing on the track. The situations where trying to leave the alternate track results in a track condition check are:

- Any multi-track operation
- A record overflow operation.

The hardware does *not* generate a track condition check when a seek is used to leave the track. This applies to any kind of seek, including seek head.

When a channel program from a virtual machine SIO (or from a Diagnose) is allowed to access an alternate track, subsequent CCWs in the channel program must be prevented from accessing adjacent tracks in the alternate track cylinder since these may belong to other virtual machines. A channel program may attempt a transition from one track to the next by any of the following:

- Seek
- Seek head
- Multi-track search or read
- Record overflow.

The full seek causes no problem: since it specifies the cylinder as well as the track, it causes the channel program to leave the alternate track and to return to a cylinder within the minidisk extent. It is certain to go back to the minidisk because the seek address was verified when the virtual CCWs were translated to real.

The seek head is dealt with as follows. When a seek to an alternate track is encountered in a virtual channel program by CP during the CCW translation process, CP converts all seek head commands (in the real, translated CCWs) to an invalid CCW opcode (X'FF'). Then when the translated channel program is executed, it is interrupted (with a command reject) at each seek head CCW so that the track to which the channel program is seeking can be checked to see that it really belongs to the virtual machine that requested the I/O. Note that this only happens to channel programs that seek out of the minidisk to an alternate track.

The multi-track operations and record overflow operations also cause no problem, because, as explained above, these are caught by the hardware and result in a track condition check.

VM/SP HPO does not provide alternate track recovery for overflow records on a 3340/3344. Therefore, a record overflow operation that is indicated to be incomplete will cause a fatal error.

Module Function and Control Flow

DMKTRKVA: When DMKCCWTR finds a virtual machine seeking out of its minidisk extent to what should be an assigned alternate track, it has to do a check of the backward record zero pointer to verify that the alternate belongs to that minidisk. So DMKCCWTR calls DMKTRKVA, passing the CCHH address of the alternate as input, and DMKTRKVA performs CP I/O to read record zero of the alternate and then returns the pointer found in record zero to DMKCCWTR.

DMKTRKFP: This is called by both DMKUNT and DMKVIO. Its function is to handle command rejects in channel programs initiated by virtual machine SIO when the channel program was found (by DMKCCWTR) to be seeking to an alternate track outside the minidisk extent. The command rejects result because, for these channel programs, any seek head commands have been invalidated (opcode changed to X'FF') in order to trap seek heads that might switch to another minidisk's track in the alternate track cylinder.

Note: Even though DMKCCWTR may also find Diagnose I/O channel programs that seek directly to an alternate track and invalidate the seek head opcodes on these channel programs, the command rejects resulting from these channel programs are handled by DMKTRKIN, not by DMKTRKFP.

DMKTRKIN: This routine performs alternate track recovery for CP I/O and for Diagnose I/O both when the Diagnose channel program results in a track condition check and when a command reject results from a seek head whose opcode DMKCCWTR made invalid. The routine has nothing to do with alternate track recovery for SIO issued by a virtual machine. But it does share a few small subroutines with DMKTRKFP.

DMKTRKIN is called only by DMKDASER, which in turn is called only by DMKIOS. These three routines work closely together during alternate track error recovery and the control flow back and forth between these routines is controlled to a great degree by flags in the IOBLOK and the IOERBLOK.

The control blocks of major concern in this area are the RDEVBLOK, the IOBLOK, and the IOERBLOK. When an error occurs and DMKIOS makes the initial call to DMKDASER (at the time of the first error associated with this IOBLOK), an IOERBLOK containing sense data has already been created; the IOBIOER field of the IOBLOK points to it. When DMKDASER gets control, it notices that this is a first call and it moves the pointer out of IOBIOER into RDEVIOER so that this first IOERBLOK, associated with the original error, can be kept over a period of time during which attempts may be made to retry the I/O operation. During these retries, further errors may cause new IOERBLOKs, pointed to by IOBIOER, to be sent back from DMKIOS. Generally speaking, RDEVIOER continues to point to the original IOERBLOK and new IOERBLOKs are created and sent back from DMKIOS after each retry that ends with an error. Generally, the new IOERBLOK from the failed retry is discarded before the next retry. But occasionally a new IOERBLOK is used by DMKDASER or DMKTRKIN to replace the original IOERBLOK, so it is pointed to by RDEVIOER and the first original IOERBLOK is discarded before the next retry. This happens when the new error is deemed to be more severe than the original (DMKDASER gives priority to channel checks) or when the original error gets corrected by a retry, but then the channel program fails on a later CCW (DMKTRKIN does this).

Control flow back and forth between DMKIOS and DMKDASER is controlled by the setting of the flags IOBERP, IOBRSTRT, and IOBFATAL, and has been described earlier in the section "DASD Error Recovery, ERP (DMKDAS)."

The control flow back and forth between DMKDASER and DMKTRKIN is controlled by the flags IOERRDR0 and IOERALTR and by a return code that DMKTRKIN passes back in register 1. Whenever either of the two flags is set, they cause DMKDASER to call DMKTRKIN whenever DMKDASER gets control (which in this case happens after a retry), even though there is no track condition check indicated in the new IOERBLOK. The IOERRDR0 flag indicates to DMKTRKIN that the retry being returned from was used to execute a channel program to read record zero. The IOERALTR flag indicates to DMKTRKIN that the retry being returned from is a restart of a user channel program (not strictly error recovery CCWs) that had a track condition check earlier. This means that invalidated seek head opcodes can be expected.

Details of Alternate Track Recovery for CP I/O and Diagnose I/O

Once a CP I/O or Diagnose I/O channel program has to be restarted because of a track condition check, the error recovery procedure invalidates (for Diagnose I/O only) all seek head opcodes in the channel program and sets the IOERALTR flag (indicating that alternate track error recovery is in progress) before proceeding. The IOERALTR flag remains set whenever any portion of the users channel program is being retried, until the channel program either ends successfully or ends with a permanent error.

Note: The flag does not remain set continuously; there are breaks while the error recovery procedure takes time out to use its own channel program to read record zero (the channel program is passed back to IOS as a "retry"). At these times the IOERRDR0 flag is set instead of the IOERALTR flag.

During the further execution of a Diagnose Channel program, invalidated seek head opcodes may be encountered once the IOERALTR flag is turned on. CP channel programs do not use seek head. The number of these opcodes encountered may be several, or none at all, depending on the user's channel program. Also, these invalidated seek heads may be trying to seek off of an assigned alternate track (usually to the next logical track) or they may have no involvement with flagged tracks at all, again depending on the nature of the user's channel program. Whenever the channel program is stopped by an invalidated seek head, a determination is made of whether or not it is trying to get off of an alternate track. This determination is made by looking at the current cylinder number (available in sense data taken at the time of the command reject) and seeing whether or not it falls within the alternate track cylinder area at the high end of the disk. If the seek head *was not* trying to get off of an alternate track, there is no problem and the subject channel program is restarted with a seek to the current cylinder and to the track specified by the invalidated seek head. If the seek head *was* trying to get off of an alternate track, record zero of the alternate track is read first to get the cylinder number of the defective track. Then the subject channel program is restarted with a seek to the cylinder of the defective track, but to the track specified by the invalidated seek head.

Tape Error Recovery, ERP (DMKTAP and DMKTPE)

Error recovery is attempted for user-initiated tape I/O operations to CP-supported devices that use the DIAGNOSE interface. The primary control blocks used for error recovery are the RDEVBLOK, the IOBLOK, and the IOERBLOK. In addition, auxiliary storage is used for recovery channel programs (repositioning and erase).

The interruption handler, DMKIOS, performs a SENSE operation when a unit check occurs. If the tape device is a 3480, tape errors are passed to DMKTPE; for all other tape devices, tape errors are passed to DMKTAP. The sense information associated with a unit check is contained in the IOERBLOK. If a channel check is encountered, the channel check interruption handler determines if retry is possible and passes control to the ERP through the I/O interruption handler.

When an error is encountered and ERP receives control, the tape error recovery module determines if this is the first entry into the ERP for this task. The IOBRCNT (IOB error count) field of the IOB is zero on initial entry. On this first entry, the pointer to the IOERBLOK is placed in the RDEVIOER field of the RDEVBLOK. This preserves the original error CSW and sense information for recording. Thereafter, IOERBLOKS are discarded before a retry is attempted or a permanent error is passed to IOS.

The ERP looks for two other specific conditions. If the error count field is not zero, entry must be due to a recovery attempt. Thus, it may be a solicited device end to correct an intervention-required condition or a retry attempt for either tape repositioning or channel program re-execution.

The ERP keeps track of the number of retries in the IOBRCNT field of the IOBLOK to determine if a retry limit has been exceeded for a particular error. If the specified number of retries fails to correct the error, the error is recorded and DMKIOS is notified of the permanent error by turning on a status flag in the IOBLOK (IOBSTAT=IOBFATAL).

If the error is corrected, the temporary error is not recorded and control is returned to DMKIOS with error flags all off. When repositioning is required in order to attempt recovery, additional ERP flags are contained in the IOERBLOK to indicate paths for specific errors (that is, data check on write must reposition, erase, and then reissue original channel program).

All error recovery is started the same except for intervention-required errors. The IOBFLAG is turned on to indicate RESTART (IOBFLAG=IOBRSTRT), and the IOBRCAW (IOBLOK Restart CAW) is filled with the restart channel address word. In addition, an IOBFLAG flag is turned on to indicate that the ERP is in control so that control can be returned to ERP during all tape error recovery (IOBFLAG=IOBERP). In the case of an intervention required error, the ERP sends a message to the operator, and then returns to DMKIOS with indications that tell DMKIOS the ERP is waiting for a device end on this device. This is done by clearing the restart flag and returning to DMKIOS with only the IOBERP flag on.

When ERP has determined a permanent error situation or successfully recovered from an error, all auxiliary storage obtained for recovery CCWs, buffers, and IOERBLOKs is freed before a return is made to DMKIOS (see Figure 34 for a summary of the IOB indicators), also, the statistical counters for 2400, 3410, and 3420 devices are updated.

If the error is uncorrectable or operator intervention is necessary, ERP calls the message writer to write the specific message.

3270 Remote Support Error Recovery

Recovery from errors associated with binary synchronous lines, and the related channel and transmission control unit hardware is processed by DMKBSC. Recovery from errors associated with data and control processing by the remote station (the device) as defined by remote status and sense byte definition (see *IBM 3270 Information Display Component Description*) is processed by DMKRGA. Control blocks associated with these errors are the CONTASK, the RDEVBLOK, the BSCBLOK, the NICBLOK, the IOBLOK, and the IOERBLOK.

The interruption handler, DMKIOS, performs a SENSE operation upon detection of a unit check condition (IOERBLOK). The related sense data is analyzed as it relates to the previous operation (CONTASK or BSCBLOK, whichever is applicable). If a channel check is encountered by the channel check interruption handler, the channel check interruption (DMKBSC) procedures determine if recovery can be attempted. If it cannot be retried, that operation is aborted and an appropriate message is sent to the system operator.

DMKRGB sends a Write Structured Field command with a Read Partition (QUERY) data stream to all 3278 and 3279 display stations during ENABLE processing. The purpose of this operation is to test for the presence of color, extended highlighting and programmed symbol set features on the enabled displays. A returned unit check with command reject is indicative of a device(s) without these features. For other errors, ERP receives control and either DMKBSC or DMKRGA determines if this is the first entry into the ERP for this task. The IOBRCNT (IOB error count) field of the IOB is zero on initial entry. On this first entry, the pointer to the IOERBLOK is placed in the RDEVIOER field of the RDEVBLOK. This preserves the original error CSW and sense information for recording. Thereafter, IOERBLOKs are discarded before a retry is attempted or a permanent error is passed to IOS.

The ERP looks for two other specific conditions. If the error count field is not zero, entry must be due to a recovery attempt. Thus, it may be a solicited device end to correct an intervention-required condition or a retry of channel program execution.

The ERP keeps track of the number of retries in the IOBRCNT field of the IOBLOK to determine if a retry limit has been exceeded for a particular error. If the specified number of retries fails to correct the error, the error is recorded and DMKIOS is notified of the permanent error by turning on a status flag in the IOBLOK (IOBSTAT=IOBFATAL).

If an apparent fatal error occurs during a Write Text operation, DMKRGA makes an attempt to recover by reselecting the remote 3270, using the standard SELECT channel commands. If the SELECT CCW's succeed, DMKRGA restarts the Write Text operation. If the SELECT channel command fails, DMKRGA treats it as a permanent TP failure.

If the error is corrected, the temporary error is not recorded and control is returned to DMKIOS with all error flags off.

When ERP has determined a permanent error situation or successfully recovered from an error, all auxiliary storage obtained for recovery CCWs, buffers, and IOERBLOKs is freed before a return is made to DMKIOS (see Figure 34 for a summary of the IOB indicators). Also, the statistical counters for 3270 are updated.

The Attached Processor and Multiprocessor Environments

Attached processor support is requested by specifying AP=YES on the SYSCOR macro. Multiprocessor support is requested by specifying MP=YES. For a complete description of system generation considerations, see the *VM/SP HPO Installation Guide.*

CP Initialization for the Attached Processor or Multiprocessor

IBM System/370 Principles of Operation has a detailed discussion of prefixing that is necessary for understanding the initialization done for the attached processor and multiprocessor systems.

Processor Addresses

The CP initialization routine, DMKCPI, begins normal processing by storing the physical address of the initialized processor in the PSA at location absolute zero (field IPUADDR). (Prefixing has not yet been established.) The logical processor address is computed by doing a logical OR of the physical address with X'40' and is stored in the PSA in LPUADDR. The logical value is used by the CP LOCK manager to avoid using a zero value. The physical value is used for signaling between the two processors.

If AP=YES or MP=YES was coded on the SYSCOR macro, DMKCPI uses the SIGP function to see if another processor or noninitialized processor is available. If so, its physical and logical addresses are stored in the PSA in IPUADDRX and LPUADDRX, respectively. If another processor or noninitialized processor is not available, APUNONLN is set to 1. If the multi-processing option is installed, a message is sent to the operator.

PSA Setup

The top two 4K pages of storage are marked (in the CORTABLE) as being CP-owned and are used as the PSAs for the two processors. The addresses of these two pages are stored at PREFIXA and PREFIXB in the PSA at location absolute zero. DMKAPI copies the information from the PSA at location absolute zero to the new PSA locations. In the PSA designated for the attached processor, PREFIXA and PREFIXB are switched. Thus, on either processor PREFIXA always represents the current processor and PREFIXB the other processor. The values of IPUADDR, LPUADDR, IPUADDRX, and LPUADDRX are also switched so that IPUADDR and LPUADDR always contain the processor addresses of the current processor and IPUADDRX and LPUADDRX contain the other processor addresses.

DMKPXA and DMKPXB are extensions to the PSA. These extensions contain headers for the processor-local queues and associated locks and counters. DMKAPI initializes the field PSAEXT in the PSA of the processor being initialized with the PSA extension of that processor. It also initializes the field PSAEXTX in the same PSA to point to the PSA extension of the other processor. The PSA extension for the IPL processor is DMKPXA and the PSA extension for the non-IPL processor is DMKPXAB.

Locking

To provide system integrity, VM/SP HPO attached processor and multiprocessor support is designed around one global system lock, a VMBLOK local lock, and several system local locks for specifically identified queues or modules.

Note: *When VM/SP HPO is running second level as a virtual MP guest on VM/XA, DIAGNOSE X'44' will be issued when a spin lock is entered.*

Global System Lock

Much of CP runs under the global system lock. For example, all command processing requires the global system lock. Also, all code executed via an IOBLOK, TRQBLOK, or CPEXBLOK is protected by the global system lock. Certain basic system functions, however, are able to execute without the global system lock on the mainline, nonerror paths. These functions include virtual page fault processing, the simulation of virtual I/O requests and other privileged operations, and the processing of a real I/O interrupt.

If the global system lock is needed and cannot be obtained, the function must be deferred until the global system lock is available. Deferral of the function is accomplished by either stacking the VMBLOK appendage (called the deferred interrupt block) or a CPEXBLOK for later processing. The processor that could not obtain the global system lock will then use the unlock dispatcher entry to dispatch a new virtual machine.

In some situations, a function cannot be deferred even though the global system lock is not available. In these cases, a processor will spin on the global system lock until it becomes available.

To ensure system integrity along the paths that do not require the global system lock, other local locks have been defined. With the exception of the VMBLOK lock, these locks are all spin locks and are held for relatively short periods of time.

Dispatch List Locks

Each processor has its own dispatch list queue, also call the true run list (TRL), of runnable users and each queue has its own dispatch list lock. The dispatcher adds the user to the dispatch list queue of the processor for which the user has affinity. This affinity lasts until the next queue-drop or until the VMBLOK is stolen by the other processor. If a user has no affinity, the user is added to the shorter of the two TRLs. If the to TRLs are equal in length, the user is added to the current processor.

Note: If the system operator issued the SET AFFINITY command for this user, the scheduler stacks the VMBLOK for the appropriate processor.

When a VMBLOK's status changes to not runnable, the scheduler removes the VMBLOK from the dispatch list.

Dispatcher Request Queue Locks

Each processor has its own dispatcher request queue (DSPRQ) and each queue has its own lock. These locks can be set by either processor. The dispatcher request queue locks control all additions to or deletions from the IOBLOK/TRQBLOK queue or the CPEXBLOK queue (collectively called the dispatch request queue) for each processor. The dispatcher determines the dispatcher request queue for a given VMBLOK by examining the field VMSTKCPU. If this field is zero, the dispatcher selects the local processor unless CPEXPROC indicates that the block is processor specific for the other processor.

I/O Lock

The I/O lock is a spin lock that serializes access to I/O devices by serializing access to fields in the real I/O control blocks: RCHBLOK, RCUBLOK, and RDEVBLOK.

Real Storage Management Lock (RM Lock)

The real storage management lock (called the RM lock) is a spin lock that controls access to certain real storage management functions and queues.

Scheduler Run List Lock

The scheduler run list (SRL) spin lock controls all additions to and deletions from the run list.

VMBLOK Lock

Each VMBLOK contains one lock, called VMLOCK, which is used by routines that need to serialize certain virtual machine related resources. These resources include the following:

1. Any unlocked or unshared pages belonging to the virtual machine.
2. Any of the unshared translation or backing store tables defining the address space of the virtual machine.
3. Certain fields of the VMBLOK that are modified by routines that do not hold the system lock. Some of these fields are VMPSW, VMGPRS, and VMRSTAT.

The dispatcher obtains the VMBLOK lock before a virtual machine is dispatched and also before a CP request or an I/O request is unstacked. When a virtual machine is dispatched, the VMBLOK address of this virtual machine is saved in the processor's PSA in the field RUNUSER. Normally this virtual machine is also unlocked by the dispatcher when it is entered after an interrupt handler has finished processing. When RUNUSER is still locked, the PSA field LASTUSER is equal to RUNUSER. When RUNUSER is unlocked, LASTUSER is set to ASYSVM.

When a CP request or an I/O request is unstacked, the associated virtual machine is locked and the VMBLOK address is placed in register 11. When the dispatcher is entered after a CP request or an I/O request has been serviced, the virtual machine whose VMBLOK address is in register 11 is locked and will be unlocked by the dispatcher. This virtual machine may not be the same virtual machine that was locked when the CP request or the I/O request was unstacked.

A CP routine must lock another virtual machine for any of the following reasons:

1. The routine, or a routine it calls, accesses any unshared page of the virtual machine.
2. The routine, or a routine it calls, alters any field of the VMBLOK that is serialized only by the VMBLOK lock.
3. The routine, or a routine it calls, could be interrupted and an exit taken to the dispatcher.

The original VMBLOK lock must be released before gaining the new lock.

Figure 35 shows the modules that obtain the VMBLOK lock for a virtual machine other than the one requesting the service.

Module	Action
DMKAPI	Locks the virtual machine that was last dispatched.
DMKBLDVM	Locks the virtual machine just built.
DMKCFO	Locks the virtual machine being set as favored.
DMKCNS	Locks the virtual machine associated with a real device block.
DMKCPS	Locks the virtual machine whose virtual device is being reset when a real device is halted.
DMKCPP	Locks each virtual machine in order to prepare the VMBLOK for uniprocessor mode.
DMKCPV	Locks the virtual machine whose storage is being locked or unlocked, or for whom accounting is being done.
DMKCSU	Locks the virtual machine receiving transferred spool files.
DMKDIA	Locks the virtual machine of the dialed system, the virtual machine of the line being dropped (DMKDIADR), or the virtual machine that owns the channel-to-channel device being coupled.
DMKGRF	Locks the virtual machine associated with a real device block.
DMKLOG	Locks the virtual machine being reconnected or the virtual machine being autologged.
DMKMID	Locks the virtual machines receiving messages at midnight.
DMKMSG	Locks the virtual machine receiving a message.
DMKMSW	Locks the system operator.
DMKNES	Locks each virtual machine active when the NETWORK SHUTDOWN command is processed.
DMKNLD	Locks the virtual machine associated with a real device block.
DMKPAG	Locks the virtual machine associated with a queued I/O request.
DMKPTR	Locks the virtual machine from which a page will be stolen.
DMKQCN	Locks the system operator.
DMKRGA	Locks the virtual machine associated with a NICBLOK.
DMKRGB	Locks the virtual machine associated with a NICBLOK.
DMKRNH	Locks the virtual machine of the destination user for a console task or the virtual machine associated with a remote teleprocessing line.
DMKSPL	Locks the virtual machine receiving a transferred spool file or the virtual machine owning a spooled reader file.
DMKVCA	Locks the virtual machine of the coupled-to CTC device.
DMKVCH	Locks the virtual machine to which the channel is being attached, or the the virtual machine from which the channel is being detached.
DMKVDA	Locks the virtual machine involved in attaching or detaching a real device.
DMKVDD	Locks the virtual machine involved in detaching a real device.
DMKVMC	Locks the virtual machine to which the caller is communicating.

Figure 35. Modules That Obtain Additional VMBLOK Lock

There are situations when a CP routine may access a virtual machine without locking it. If the CP routine, or any routine it calls, is only altering VMBLOK fields that are serialized by the system lock, locking the virtual machine is not necessary. For example, to process the SET PRIORITY command for a virtual machine, locking the virtual machine is not necessary since the altered VMBLOK field, VMUPRIOR, is serialized by the system lock. But to process the SET FAVORED command, locking the virtual machine is necessary since some of the VMBLOK fields altered, such as VMRSTAT, are only serialized by the VMBLOK lock.

DMKLOKFR	Free Storage Lock
DMKPXA, TRLLOCK DMKPXB, TRLLOCK	Dispatch List (True Run List) Locks (one per processor)
DMKLOKTR	Timer Request Queue Lock
DMKPXA, RQLOCK DMKPXB, RQLOCK	Dispatcher Request Queue Locks (one per processor)
CPEXBLOK IOBLOK, TRQBLOK	Queue Lock Deferred execution blocks Processor related blocks

These are system spin locks that are held for very short periods of time. The control program code that runs without the global system lock must manipulate these queues and these locks to ensure system integrity along the unlocked paths.

User-Defined Locks

If you have user-defined areas that are used by more than one virtual machine and you need to serialize their use, you will need to define your own locking conventions. You can use the LOCK macro to obtain and release a PRIVATE lock. *VM/SP HPO CP for System Programming* has details on how to code the LOCK macro.

Machine Check Handler – Attached Processor and Multiprocessor Applications

A machine check interrupt is initially handled without the global system lock. DMKMCH determines if the error requires system termination, virtual machine termination, or simply recording and continuation. If the system was in a wait state or a virtual machine was in control and the system is not to be terminated, the machine check handler requests the global system lock with the defer option. If the lock can be obtained, normal DMKMCH processing continues. If the lock cannot be obtained, DMKMCH stacks a CPEXBLOK with CPMCHSE set and exits to DMKDSPRU. This CPEXBLOK causes processing to resume at DMKMCHSE with the global system lock held. Any machine checks that occur before the CPEXBLOK processing has completed are considered recursive machine checks and handled appropriately. If the control program was in control and the system is not to be terminated, the machine check handler saves status in the CPEXBLOK, sets CPMCHSE and reloads

MCOPSW. CPMCHSE is set to prevent the dispatcher from starting any new work on this processor until the machine check processing has completed.

DMKMCH passes control to DMKACRCT if channel termination has been set. DMKACR will attempt to recover the failing channel or channels by issuing CLRCH to each affected channel. If CLRCH does not restore a failing channel to an operational state, and if the system can continue operation without the failing channel, DMKACR will mark all paths through the channel offline and return to DMKMCH.

For machine checks on a Vector Facility, DMKMCH will take one of two actions: For a Vector source machine check, DMKMCH will reset the current user (if any) and place that user in console function mode. For a Vector failure machine check, (or for the 12th Vector source machine check since initialization), DMKMCH will disable the Vector Facility. If there is more than one Vector Facility connected to the complex, normal operation will continue, except for the user (if any) whose integrity is exposed because of the error. If no other Vector Facility is available, current and future users will be informed of the facility's status when they attempt to use it. A VARY-ON of the facility may allow re-access.

DMKMCH passes control to DMKMCTPT if the system is running in attached processor or multiprocessor mode and a decision has been made to terminate the system. In general, if a virtual machine was running when the machine check occurred, only that virtual machine is terminated.

DMKMCTPT determines if the system can continue and if the processor can continue. If the machine check was not a clock error or the control program was in control on either processor, the other processor is signalled to stop and store status and a wait state PSW is loaded on the failing processor. An attempt is made to issue message 610W to the operator before the main processor is stopped. If the machine check was a clock error on a non-I/O processor and the control program was not in control, the main processor is signalled via an external call to initiate automatic processor recovery with an indicator to continue processing. If the error was a clock error on an I/O processor, and (a) the system is an MP system or (b) the channel-set switching facility is installed, and if the control program was not in control, automatic processor recovery will be initiated on the nonfailing processor.

The malfunction alert interrupt handler (DMKMCTMA) receives control from the external second level interrupt handler. If the malfunction alert came from the main processor in an AP system, and if the channel-set switching facility is not installed, a X'001' wait state disabled PSW is loaded. If the malfunction alert came from an attached processor and a virtual machine was in control, an indication is set to terminate the virtual user and CPAPRPND is set for processor recovery. If the attached processor was in supervisor state, message 610W is sent to the operator and a 013 wait state PSW is loaded. If the attached processor was in a wait state, CPAPRPND is set for processor recovery. If the malfunction alert occurs in a MP system and if a virtual machine was in control on the failing processor, an indication is set to terminate the virtual machine and automatic processor recovery is initiated.

The automatic processor recovery routine (DMKMCTPR) receives control from the external SLIH or the dispatcher. If the system is to continue processing, the vary processor offline routine (DMKCPPUP) is called. DMKCPPUP examines the chain of virtual machines for affinity to the failing processor and shared segment pointers. Any shared segment pointers for the failing processor are switched to point to the recovery processor's shared segments. All the system control blocks and save areas necessary to run in attached processor or multiprocessor mode are also freed. The time from the first timer request queue element is placed into the clock comparator for the main processor.

While preserving the maintained fields in the absolute zero area, the recovery processor's prefix storage area is copied to the absolute zero area and prefixing is stopped. The APUOPER (or MPUOPER) in the PSA is turned off in the absolute zero area, and the prefix storage areas for both processors are freed. The pages and DASD slots held by the failing processor for shared segments are freed by DMKPGT and DMKPTR. A message (194I) is issued, and return is made to DMKMCTPR. For any virtual machines with affinity to the failing processor, DMKMCTPR resets the affinity for each, issues message 621I, and puts the machine in console function mode (if the virtual machine is not disconnected). If a virtual machine is to be terminated, the virtual machine is reset, messages 616I and 619I issued. Normal return causes the system to continue processing in uniprocessor mode.

The action that the machine check handler takes for a given situation is determined by the error itself, the operating environment of VM/SP HPO, and whether the system was performing a CP function or a virtual machine function—the system was not performing at all (a loaded wait state condition when the error occurred). Figure 36 clarifies the action the system takes for the given situations.

Error Condition	VM/SP HPO Processing				Virtual Machine Processing			
	Uni-Proc.	Attached Processor		Multiproc. Operation	Uni-Proc.	Attached Processor		Multiproc. Operation
		Main	Attach.	Either Processor		Main	Attach.	Either Processor
Invalid machine check interrupt code	1	1	1	1		1	1	1
Invalid PSW data	1	1	1	1	1	3	3	3
Register, Program mask instruction address invalid	1	1	1	1	1	3	3	3
System damages	1	1	1	1	1	3	3	3
TOD or CPU Clock Errors	1	1		1	1	1	1	3,4
Multibit (solid) Storage error	1	1	1	1	3,2	3,2	3,2	3,2
Multibit (intermittent) storage error	1	1	1	1	3,2	3,2	3,2	3,2
Storage Protect Key (solid) failure	1	1	1	1	3	3	3	3
Storage Protect (intermittent) failure	2	2	2	2	2	2	2	2
Malfunction alert	5	1	1	1	5	1	3,4	3,4
Channel inoperative	6	6	5	6	6	6	5	6
Vector Facility Failure (or 12th Source Error)	7	7	7	7	3	3	3	3

Legend:
1 = Load wait state PSW
2 = Refresh for retry operation
3 = Terminate the virtual machine
4 = Automatic processor recovery
5 = Not applicable
6 = Channel recovery
7 = Vector Facility disabled

Figure 36. Condition/Action Table for Uncorrectable Errors

Multiprocessor External Interrupts

For external interrupts that can occur in attached processor/multiprocessor mode (time-of-day sync check, malfunction alert, external call, and emergency signal), DMKPSAEX gives control to DMKEXTSL. DMKEXTSL does the following for each kind of interrupt:

Malfunction alert

- Call DMKMCTMA, which will either load a disabled wait state on the appropriate processor or initiate automatic processor recovery, to allow the system to run in uniprocessor mode. If a user was running at the time of the malfunction alert he is terminated.

SHUTDOWN Emergency Signal

Issued prior to shutting the system down.

- Turn off APUOPER in each PSA to indicate that the system is in uniprocessor mode.
- Load a 008 disabled wait PSW on the receiving processor.

EXTEND Emergency Signal

- Disable channel zero.
- Pass control to the dispatcher at DMKDSPRU.

EXTEND EXIT Emergency Signal

- Enable channel zero in control register 2.

QUIESCE Emergency Signal

- Give control to the dispatcher at DMKDSPRU, which will load a wait PSW that is enabled for external calls only.

SYNC Emergency Signal

Issued by DMKCLKMP when the clocks are no longer synchronized (low order synchronization).

- Give control to DMKCLKAP to synchronize the clock on the attached processor. If the set clock fails, the non-IPL'ed processor is terminated with a CLK003 abend.

CLKCHK Emergency Signal

- Give control to DMKCLKCC. If the clock on the non-IPL'ed processor is not synchronized with the IPL'ed processor (high order synchronization) or is not set, then a flag is set to cause DMKCLKMP on the main processor to synchronize the clocks. The non-IPL'ed processor is then put in a wait state enabled for external interrupts. If the clock is not working, the non-IPL'ed processor is terminated with a CLK003 abend.

APR External Call

- Give control to DMKMCTPR to allow the system to run in uniprocessor mode.

RESUME External Call

Cancels a previous QUIESCE.

- Give control to the dispatcher at DMKDSPRU.

WAKEUP External Call

"Wake-up" an idle processor.

- If the system was running a user, reload the external old PSW.
- If the system was not running a user, then try to obtain the SYSTEM lock.
- If the SYSTEM lock is obtained, give control to the dispatcher at DMKDSPCH.
- If the lock is not obtained, give control to the dispatcher at DMKDSPRU.

DISPATCH External Call

Inform the other processor of a processor related CPEXBLOK.

- Try to obtain the global system lock.
- If the system lock is obtained, go to the dispatcher at DMKDSPCH.
- If the lock is not obtained and the system was in a wait state, go to DMKDSPRU.
- If the lock was not obtained and the system was not in a wait state, reload the external old PSW.

Time-of-Day SYNC Check

- Call DMKCLKSC. DMKCLKSC signals the non-IPL processor to quiesce. It then sends message DMKCLK970W to the operator and calls DMKCLKMP. DMKCLKMP issues a SYNC emergency signal to synchronize the clocks. DMKCLKSC issues a RESUME signal to allow the quiesced processor to continue.
- If the SYSTEM lock is held, go to the dispatcher at DMKDSPCH.
- If the SYSTEM lock is not held, go to the dispatcher at DMKDSPRU.

I/O Subsystem

Mainline, nonerror processing in the I/O subsystem runs without the global system lock. Access to fields in the real I/O control blocks (RCHBLOK, RCUBLOK, and RDEVBLOK) is serialized by the I/O lock, a global spin lock. In an attached processor environment, only the main processor is capable of initiating I/O requests and receiving I/O interrupts. If the I/O subsystem running on the attached processor requires that I/O be started, a SWITCH will be issued to resume processing on the main processor.

In a multiprocessor environment, both processors have I/O capability. If either processor receives an I/O request, that processor attempts to initiate I/O operations.

At system generation time, when a channel path to a device is defined on one processor, an alternate logical path is automatically defined for the other processor. Thus, both processors *can* have access to any I/O device in the MP configuration.

If either processor receives an I/O request, that processor attempts to initiate the I/O operation on one of its own paths to the required device. If none of the online paths to the required device is available from the executing processor, that processor queues the I/O request on all busy and scheduled paths to the device, both its own and those of the other processor. If there is no online path from the executing processor, that processor queues the I/O request on the first online and available path for the second processor, as well as on all busy or scheduled paths for that processor.

While it is not required that both processors have access to all I/O devices, heavily used devices should be accessible by both processors to provide efficient system operation and to increase the possibility of system recovery following a processor or channel failure.

Shared Segment

The shared segment subfunction of VM/SP HPO (DMKATS, DMKCFG, DMKCFH, DMKPGS, and DMKVMA) runs under the global system lock on either processor. All protected shared segments are duplicated in a system that is generated for attached processor or multiprocessor mode and that is initialized on a machine with the multiprocessing feature. DMKCFG obtains sufficient storage to construct the duplicate page and swap tables in contiguous storage. The SHRTABLE SHRPAGE pointer points to the page and swap tables for the main processor, and the page and swap tables for the attached processor are at a fixed displacement from the page and swap tables for the main processor. DMKCFG initializes both sets of page and swap tables. Initially, the two swap tables point to the DASD locations specified in DMKSNT. However, as the pages are read into storage and then stolen, each shared page is allocated its own DASD slot and is pointed to by only one swap table entry.

The last user to purge a shared system causes both sets of page and swap tables to be released.

One shared page table is reserved for use by each processor. This includes both problem state and supervisor state execution on behalf of a virtual machine. To accomplish this, each time a virtual machine running a shared system is locked, a test is made to determine whether or not the virtual machine was last serviced on this processor. If it was last serviced on the other processor, all of its shared page table pointers in its segment tables are switched to this processor's shared pages.

DMKPTR is able to steal a shared page from a shared page table reserved for the processor it is running on without notifying the other processor. The virtual page could not appear in the look-aside buffer of the other processor.

The dispatcher releases the VMBLOK lock on LASTUSER following the check for pending interrupts (assuming no fast redispatch possible) unless the virtual machine was running one or more shared systems. In the latter case the VMBLOK lock is not released until the DMKVMA scan for a changed page is completed.

DMKVMA scans all protected shared segments that the virtual machine used. For every changed page that it finds, DMKVMA checks whether or not the system lock is held. If the system lock is held, the changed page is returned to CP free storage. If the system lock is not held, DMKVMA marks the page table entry as invalid, marks the swap table entry as in transit, and indicates that the core table entry is on the free and flush lists. The other virtual machines can continue to use the shared segments. The changed pages are replaced when the next reference to the changed page is made.

If the shared segment is violated, an error message (DMKVMA456) is sent to the violator, and he is placed in console function mode. The user may examine his PSW and registers to determine what caused the violation. The user enters the BEGIN command to resume execution at the point of interruption.

Segment Protection Extension

Segment protection extension is a microcode assist hardware feature that prevents virtual machine users from changing shared system segments. It is an enhancement to virtual machine assist that provides the same level of segment protection currently handled by the control program. The segment protection extension is available on the 308x, 3090, and dual 4381 processors. If the feature is not present and you desire shared-segment protection, CP continues to handle shared-segment protection.

At system initialization, DMKCPI determines whether the segment protection extension is available. DMKCPI initializes the segment protection status field (CPSEGPRT) to zero. The segment protection bit (bit 29) is turned on in the segment table entry and a store is attempted into the location in DMKCPI. If no protection exception occurs, the segment protection extension is not available. If a protection exception occurs, the segment protection extension is available, and CPSEGPRT is set to one. In both cases, the segment protection bit in the segment table entry is turned off. System initialization continues.

The first time a shared segment is used, a share table (SHRTABLE) and the appropriate number of page and swap tables are built. The status field CPSEGPRT is examined to determine whether the segment protection extension is available. If the segment protection extension is available, the SHRSGPRT field in the SHRFLAG byte of the SHRTABLE is turned on. When SHRSGPRT is on, only one set of page and swap tables are built for the segment. No VMABLOK is built for the segment. With the segment protection feature providing protection at the segment level, the scan for a shared page is eliminated. Additionally, the segment protection extension eliminates the need for the control program to maintain duplicate page tables and swap tables in attached processor or multiprocessor systems with protected shared segments.

You can operate without segment protection by specifying PROTECT=OFF in the NAMESYS macro statement. By specifying PROTECT=OFF, you do not enable the assist for the particular segment. The segment protection extension will still protect segments for which PROTECT=OFF was not specified. But this will also disable the (software) segment protection currently handled by CP and hence there will be no protection at the segment level at all.

CP Method of Operation and Program Organization

This part contains the following information:

- CP Program Organization
- Use of the Annotated Flow Diagram
- Virtual I/O Operations and Interruption Processes.

CP Program Organization

Use of the Annotated Flow Diagram

The following text sections, which describe each major CP function, are annotated flow diagrams. These diagrams, consisting of logic labels and commentary, describe the general flow and use of CP logic modules and their relationship to other modules while performing a specific function or task. The annotated flow diagrams do not contain references to error messages, abnormal termination conditions, or most control block field labels. This avoids complexity and makes the general logic of CP and its related tasks more understandable to the user. With "understandability" as the key, obtuse and complex logic that is used for obscure and seldom used functions is not described. Also the flow diagram does not indicate nor describe every entry point encountered in a function. Nor do the diagrams illustrate the innumerable times that commonly used modules are utilized. DMKFRE and DMKCVT, the obtaining and returning of free storage and the number base conversion modules are such examples. Annotated flow diagrams are arranged by function and subfunction. Titles for these functions and subfunctions also precede annotated flow text and labels. The text in the charts is prefixed by underscored and capitalized entry points and labels. Entry points are indicated by seven or eight characters; the first three characters are DMK. Labels are indicated by prefixing with a comma and the six-character module identification.

The annotated flow diagrams in this section do not reflect use of the MSS. If there is an MSS attached to your system, consult Appendix B in this volume for flow diagrams of those functions that utilize the MSS (such as logging on a virtual machine that has a minidisk defined on an MSS 3330V volume).

Note: *Annotated flow diagrams are not to be construed as trace material. The dynamics of CP operations preclude the use of the annotated flow diagrams, as they are shown in this manual, as traces of CP functions.*

CP Interrupt Processing

SVC Interrupts – Problem State

DMKSVDIN
Entry for SVC interrupts from problem state. For problem mode and ADSTOP (SVC X'B3'), the overlaid instruction is replaced.

DMKCFMBK
Console function mode is entered.

DMKSVDIN
For problem state SVC 76 (X'4C'), check for valid parameter passing.

DMKVERD, DMKVERO
Determine the operating SCP used in the virtual machine by examining passed parameters in R0 and R1.

DMKSVD, SVCVER
For invalid parameter passing, error recording is not performed.

DMKIOEVR
The SVC is reflected to the user.

DMKIOFVR
On correct parameter reflection, record the error.

DMKSVD, REFSVCB
REFSVCB is called if TRACE SVC was in effect or if the virtual machine's page zero is not in real storage. Obtains the system lock before continuing. If the system lock is not immediately available, REFSVCB defers the interrupt and exits to DMKDSPRU.

DMKTRCSV
The DMKTRC module is called if TRACE SVC was invoked.

DMKPRGRF
If tracing is not active, flag user as being in instruction wait state and reflect the SVC back to the user.

DMKSVD
If the virtual machine's page zero is in real storage, generate and store an old SVC PSW. Fetch the new SVC PSW. If there is no PSW state change, store user's new PSW in RUNPSW, restore registers, and dispatch via LPSW.

DMKSVD, REFSVCA
If there is a PSW state change, obtain the system lock before continuing. If the system lock is not immediately available, defer the interrupt and exit to DMKDSPRU.

DMKDSPB
Check the altered PSW.

SVC Interrupts – Supervisor State

DMKSVCIN
Entry for SVC interrupts from supervisor state.

DMKSVC, SVCDIE
Entry is for a system failure and is a SVC 0 or SVC 4 abend condition.

DMKDMPDK
Perform partial or full real storage dump.

DMKCKPT
Checkpoint the system.

DMKCPINT
Perform an automatic IPL if indicated.

DMKSVC, SVCLINK
Entry via SVC 8 provides linkage to a called routine in R15.

DMKPTRUL
If called routine is not resident, page it in and return control to the caller by loading the SAVERTN into the old PSW and then load the old PSW. The caller's addressability, SAVEAREA address and return address are maintained in a new SAVEAREA.

DMKSVC, SVCRET
Entry via SVC 12 return control from the called routine to the calling routine and restores addressability via R12 and R13.

DMKPTRUL
If a nonresident module, unlock page to return it to DASD.

DMKSVC, SVCRLSE
Entry via SVC 16 to release the current SAVEAREA used by SVC 8 and 12. Return to caller.

DMKSVC, SVCGET
Entry via SVC 20 to obtain a new SAVEAREA. Return to caller.

DMKSVC, SVCSWIT
Entry via SVC 24 to switch control to the main processor.

External and Clock Interrupt Reflection

DMKEXTIN
Entered via the interrupt key on system console, adjust accounting to charge for supervisor overhead. If problem mode, attention interrupt, update the virtual machine PSW from the external old PSW.

DMKPSA, EXTBUTTN
Exit to dispatcher, if there is no logged-on operator, or the operator is disconnected, or there is no active terminal. If the operator was logged on and the external interrupt key was pressed, disconnect the operator's terminal.

DMKQCOCL
Clear all console requests.

DMKSCNRD
If the device is a terminal or graphic device, issue HIO to the real device.

DMKDSPCH
Exit to the dispatcher.

DMKPSA, EXTBUTTN
For 3704/3705, convert resource identifier for the NCP terminal for the indexable entry into the NICBLOK for the associated VMBLOK, then

DMKRNHND
Reset all BTUs.

DMKDSPCH
Exit to the dispatcher.

DMKPSA, EXTEXTD
Upon location X'80' timer interrupt, indicate the user end of the time slice by storing flag in the VMBLOK's VMOSTAT.

DMKDSPCH, DMKDSPRU
If the system lock is held or is available, exit to the main entry of the dispatcher, DMKDSPCH. Otherwise, exit to DMKDSPRU.

DMKPSA, EXTTIMER
Upon processor timer interrupt, VMTLEVEL in VMBLOK as a real processor timer interrupt.

DMKTMRVT
Simulate the interrupt.

DMKDSPCH, DMKDSPRU
If the system lock is held or is available, exit to the main entry of the dispatcher, DMKDSPCH. Otherwise, exit to DMKDSPRU.

DMKPSA, EXTCKC
Upon clock comparator interrupt reflection unchain the active TRQBLOK. Call DMKSTKIO.

DMKSTKIO
Stack the block.

DMKDSPCH, DMKDSPRU
If the system lock is held or is available, exit to the main entry of the dispatcher, DMKDSPCH. Otherwise, exit to DMKDSPRU.

Missing Interrupt Processing

DMKDID
Receives control from a timer interrupt (TRQBLOK).

DMKDIDDA
Checks for DASD devices when a missing interrupt is detected.

DMKDIDGR
Checks for graphic devices, except 1053 and 328X printers, when a missing interrupt is detected.

DMKDIDTA
Checks for tape devices when a missing interrupt is detected.

DMKDIDUR
Checks for unit record devices when a missing interrupt is detected.

DMKDIDMS
Checks for miscellaneous devices when a missing interrupt is detected.

DMKDIDLF
Cleans up certain missing interrupts at LOGOFF/FORCE times.

DMKDID, SCAN
Scans RDEVBLOKs. If RDEVBUZY flag is on, the RDEVTYPC field is examined to verify that the RDEVBUZY flag and timer interrupt are for the same class of device. If the classes match, the RDEVMID bit is turned on. If not, the scan continues. The timer is reset.

DMKDID
Stacks a CPEXBLOK to regain control.

DMKDID, DELQDV
Receives control from the CPEXBLOK and calls DMKIOSHA to perform a HALT DEVICE and CLEAR I/O.

DMKDID

Simulates I/O interrupt through DMKIOTRC. Schedules a ten second timer interrupt to check RDEVMID. A message is sent to the operator and a record is written to the system log record.

Monitor Interrupt Processing

DMKMON

The VM Monitor data collection component uses both sample and trace techniques. Selected system counters are sampled by routines entered periodically via TRQBLOK. Selected events are traced upon execution via monitor call instructions embedded at strategic points in the control program.

DMKENTTI

Entered via TRQBLOK every two seconds (unless specified otherwise with the MONITOR INTERVAL command). A new TRQBLOK is immediately stacked via a call to DMKSCHST to specify return of control to the same entry point two seconds later. This subroutine is a high frequency (relative to the PERFORM, USER, DASTAP sampler) I/O status sampler. All channels are tested for a busy condition with a TCH instruction. All control units and devices are tested for a busy condition by examining the appropriate CP control blocks. The data obtained is accumulated for later sampling by the DASTAP class of data collection in a class 6 (DASTAP) code 2 (I/O status) record. The subroutine DMKENT62 performs this collection after the standard class 6 (DASTAP) code 1 record has been collected by MONCOD61 in DMKMONTI.

DMKMONMI

Entered from DMKPRG after a monitor call in a class currently enabled (as defined in CR 8 mask) has been executed by CP in supervisor state. The monitor call instruction number and code number stored by the hardware in the PSA are used to index branch tables to reach the appropriate data collection routines. As necessary, the data is stored in the monitor I/O buffers before output. Upon completion, control returns to instruction after monitor call.

Class	Code	Activity Being Monitored
1	0	Begin a console read
	1	Console output
	2	End a console read
	3	Begin sleep with time out
	4	User logon
	5	User logoff
2	2	User dropped from queue
	3	User added to queue
	4	User added to eligible list
3	0	Logical swap-out
	1	Start of physical swap-out
	2	End of physical swap-out
	3	Start of physical swap-in
	4	End of physical swap-in
	5	Logical swap-in
5	0	Beginning of instruction simulation
7	0	SIO for DASD seek
8	2	Device I/O counts and system clocks data collected

DMKMONPR
All data collection subroutines use a common buffer management subroutine to obtain sufficient space in the monitor buffers. When not enough space is available, a switch is made to the next buffer in the chain and the full buffer is scheduled for output via a CPEXBLOK. I/O is handled by DMKIOSQR if tape is in use, or by DMKMIAWO if a spool file is in use. If data collection gets ahead of buffer output and all the monitor buffers are filled, a temporary suspension occurs.

DMKMONIO
Handles normal and abnormal completion of buffer output to disk or tape. For normal completion, the buffer used for I/O is made available next buffer is already full, its output immediately scheduled.

DMKENTKC
Entered via CPEXBLOK at midnight if automatic monitoring to spool file is in effect and it is required to close out the current file and continue monitoring with a new file. DMKENT satisfies the nucleus residency requirements of CPEXBLOK entry point and acts as a stepping stone to DMKMIA. Goes to DMKDSP after successful call to DMKMIAKC.

DMKMIAKC
Sets up a request to invoke a MONITOR CLOSE command in DMKMCCCL.

DMKMCCCL
Executes MONITOR CLOSE command and calls DMKMIACC to complete processing.

DMKMIACC
Invoked by the MONITOR CLOSE command to close the spool file and chain the spool file block to the reader of the virtual machine where data reduction is to take place. Starts new spool file if appropriate.

DMKENTST
Entered via TRQBLOK due to previous determination by automatic monitoring facilities that a MONITOR START SPOOL command should be issued. This entry satisfies the need for CP nucleus residency and immediately calls the pageable DMKMIAIN.

DMKMIAIN
Builds a message buffer containing a MONITOR START SPOOL command and calls DMKMCCCL.

DMKMCCCL
Executes MONITOR START SPOOL command. DMKENTST gives control to DMKDSP after successful execution.

DMKENTET
Entered via TRQBLOK due to previous determination by automatic monitoring facilities that a MONITOR STOP command should be issued at this time. This entry satisfies the need for CP nucleus residency and immediately calls the pageable DMKMIAEN.

DMKMIAEN
Builds a message buffer containing a MONITOR STOP command and calls DMKMCCCL.

DMKMCCCL
Executes MONITOR STOP command. DMKENTET gives control to the dispatcher after successful execution.

DMKMIAST
Entered from DMKCPI when it is determined that automatic monitoring has been requested via the SYSMON macro in DMKSYS and that TRQBLOKs should be queued via calls to DMKSCHST to invoke a MONITOR START SPOOL command and a MONITOR STOP command at specified times in the future. If monitoring is required to start immediately because the start time has passed, a CPEXBLOK is built to give control to DMKENTSC, which invokes the DMKMIAIN mechanism described above.

All other DMKMCC, DMKMNI, DMKMNJ, and DMKMIA entry points are used as a result of the processing of MONITOR commands or special conditions.

DMKMOO
Formats records for the timer-driven monitor classes: PERFORM (Class 0), USER (Class 4), and DASD/Tape (DASTAP, Class 6).

Class	Code	Activity Being Monitored
0	0	Interval and swapping statistics for IPL processor
	1	APU clocks and counters
	2	System resource management data
	3	Page and swap table migration data
	4	Greater than 16 Mb page and swap table migration data
	5	DASD and Paging Storage usage by ALOCBLOK
	6	System-wide swapping data
	7	Free storage usage
4	0	Virtual machine statistics collected
	1	Shadow table maintenance
6	0	DASTAP header initialized and created
	1	Device information collected
	2	I/O utilization
	3	Channel utilization
	4	3880 Model II subsystem status
	5	3880 Model II subsystem counts

Three Class 0 monitor call codes have been reserved for special purposes. They are used without actually executing monitor calls, but are a result of MONITOR command processing. They are:

Class	Code	Function
0	97	Write header record after MONITOR START command
	98	Write trailer record after MONITOR STOP command
	99	Write suspension record when data collection resumes

Program Interrupt Processing

DMKPRGIN
For a program interrupt received while in supervisor mode (indication of CP module error) and INTRDR+1 does not indicate MONITOR CALL (X'40') exit to DMKPRG, CPERROR.

DMKPRG, CPERROR
Send abend message to the system operator.

DMKDMKPK
Dump storage and initiate loading (via IPL).

DMKPRGIN
For supervisor state and MONITOR CALL save registers in DMKPRGPR.

DMKPRGMI
Do MONITOR CALL interrupt processing (DMKMON).

DMKPRG, PRNSTAT
For paging exception X'11' and EC mode with translation on call DMKVATEX.

DMKPRG, PRNSTAT
For PER interrupt (X'80'), save PER event information (PER code and PER address) in the PERBLOK (if CP PER) or the ECBLOK (if user PER), and turn on the PER pending flag (VMPERPND).

DMKVATEX
Process the exception.

DMKPRGIM
For paging exception, X'11' and EC mode with translation off, and enabled for I/O interrupts and PAGEX on call DMKVATPF.

DMKVATPF
Process the pseudo page fault.

DMKPRG, PAGEXCP
For all other page fault conditions go to DMKPTRAN.

DMKPRG, OBSLOCK
The system lock must be obtained before DMKPTRAN is called. If the system lock is not immediately available, defer the interrupt and exit to DMKDSPRU.

DMKPTRAN
Bring in the page from the auxiliary device.

DMKDSPCH
Exit to dispatcher.

DMKPRG, PRNSTAT
For segment exception X'10' with EC mode on and translation on call DMKVATSX.

DMKPRG, PRNSTAT
For a Vector Operation Exception, X'19', pass control to DMKVFRIN.

DMKVFRIN
Process the vector operation exception, (X'19').

DMKVFSOS
Obtain a vector register save area (and a VECBLOK, if necessary).

DMKVATSX
Process the exception.

DMKPRG, PRGSIMI
For the segment exception, X'10' does not follow the above parameters; process it as an addressing exception.

DMKPRG, TRANSEX
Process X'12' translation exceptions.

DMKPRG, PROG01
For a privileged operation exception of a virtual machine in supervisor mode, examine INTPR+1. If X'02', call DMKFPS; otherwise, call DMKPRVLG.

DMKFPS
Process the exception and, if successful, dispatch the user. If unsuccessful, return to DMKPRGIN.

DMKPRVLG
Process the exception.

DMKPRV, DMKPRGSM
For virtual machines in problem mode, store the users new program PSW in VMBLOK VMPSW.

DMKPSASV
When the program interrupt occurs and the user's page 0 is not resident or the virtual machine is in EC mode, paging is performed.

DMKDSPB
Check the new PSW.

DMKPRVLG
Validate the privileged operation indicated in VMINST and perform the service.

Code	Operation
X'08'	SSK - Set storage key
X'09'	ISK - Insert storage key
X'44'	EX - Execute instruction
X'80'	SSM - Set system mask
X'82'	LPSW - Load PSW
X'9C'	SIO - Start I/O
X'9D'	TIO - Test I/O
X'9E'	HIO - Halt I/O
X'9F'	TCH - Test Channel
X'AC'	STNSM - Store, then AND system mask
X'AD'	STOSM - Store, then OR system mask
X'AE'	SIGP - Signal processor
X'B1'	LRA - Load real address
X'A649'	VRSVC - Save Changed Vector Registers
X'A6CA'	VACSV - Save Vector Activity Count
X'A6CB'	VACRS - Restore Vector Activity Count
X'B202'	STIDP - Store processor ID
X'B203'	STIDC - Store channel ID
X'B204'	SCK - Set TOD clock

Code	Operation
X'B206'	SCKC - Set TOD clock comparator
X'B207'	STCKC - Store TOD clock comparator
X'B208'	SPT - Set CPU timer
X'B209'	STPT - Store CPU timer
X'B20A'	SPKA - Set PSW key from address
X'B20B'	IPK - Insert PSW key
X'B20D'	PTLB - Purge TLB PTLB - Purge TLB
X'B210'	SPX - Set prefix
X'B211'	STPX - Store prefix STPX - Store prefix
X'B212'	STAP - Store CPU address
X'B213'	RRB - Reset reference bit
X'B220'	SERVC - Service Call
X'B229'	ISKE - Insert storage key extended
X'B22A'	RRBE - Reset reference bit extended
X'B22B'	SSKE - Set storage key extended
X'B221'	IPTE - Invalidate Page Table Entry
X'B22C'	TB - Test Block
X'E501'	TPROT - Test Protection
X'B6'	STCTL - Store control registers
X'B7'	LCTL - Load control registers
X'BA'	CS - Compare and swap
X'BB'	CDS - Compare double and swap

DMKPRV, LOCKET
The system lock must be obtained before other supervisor routines are called. If the system lock is not immediately available, defer the interrupt and exit to DMKDSPRU.

DMKHVCAL
On privileged operations of DIAGNOSE X'83' and the associated function code, perform the service.

DMKVSIEX
Execute privileged I/O operations of SIO, HIO, TIO and TCH.

DMKTMRTN
Perform privileged operations related to TOD clock, TOD clock comparator and the processor timer.

DMKPRGSM
Program interrupt is reflected back to the user on invalid instruction operands, unsupported instruction operand codes and DIAGNOSE '83' function codes that are not a multiple of 4.

DMKMHV
Process Service Call.

Virtual I/O Operations and Interruption Processes

CTC Device Operations between Two Virtual Machines

DMKVSIEX
Virtual I/O operation is reflected to DMKVCA, the channel device module, for processing.

DMKVCAST
For SIO, check if the CTC device (channel-to-channel adapter or 3088) is coupled. If not coupled, call DMKDIBSM.

DMKDIBSM
Simulate return status.

DMKVCA, VCASTART
For a coupled CTC device, analyze operations resulting in X-side (read) and Y-side (write) of the data transfer operation.

DMKVCA, VCASIOB
Detected interrupts are presented to users via stacked IOBLOKs and DMKSTKIO.

DMKVCBTS
CTC device TIO activity is determined by examining Y-side information to determine mode and activity.

DMKVCBSH
CTC device HIO and HDV is processed by determining the condition code to present and whether the Y-side should be notified.

DMKVCBRD
CTC device process results from RESET xxx or SYSTEM RESET commands. The CTC device status is reset but the CTC devices are not uncoupled.

DMKVCBRS
Uncoupling of the CTC device is achieved in the VDEVBLOK (VDEVNRDY flag) of the idle CTC device by an invoked DETACH xxx or user LOGOFF. Return to calling routine.

Scheduling I/O for CP and the Virtual Machine

DMKIOSQR
Entered via SVC. Entry point indicates a CP I/O event as indicated in the IOBLOK. For start request, increment the SIO count in the RDEVBLOK and start the device if it is available. If not (device busy or already scheduled) queue the IOBLOK and return the operation to the caller.

DMKIOSQV
Entered via SVC. Entry point indicates virtual machine initiated I/O event. Preserve VMBLOK address in R11, turn off IOBCP bit in the IOBLOK, add 1 to SIO count in the VDEVBLOK (or RDEVBLOK). Process the SIO if there is any available path to the device. If not, queue the IOBLOK and return the operation to the caller.

Standard DASD I/O Initiated via Diagnose

DMKDGDDK
Perform simple count-key-data disk I/O of a standard format. Entry is via DMKHVC code X'18'.

DMKSCNVU
Find device related to SIO cuu address.

DMKFREE
Allocate storage for IOBLOK and RCWTASK.

DMKDGDDK
Build and check the CCW string.

DMKIOSQV
Execute I/O. On completion, post condition code (and error return code in R15, if detected).

DMKDSPCH
Exit to dispatcher.

General I/O Operation Initiated via Diagnose

DMKGIOEX
Perform general I/O operation. Entry is via DMKHVC code 20.

DMKSCNVU
Find device related to SIO cuu address.

DMKFREE
Allocate storage for the IOBLOK.

DMKCCWTR
Build the read CCW list.

DMKDEXIN
Insert Define Extent into channel program if cache should be bypassed.

DMKIOSQV
Queue the I/O request for execution.

DMKGIO, DIAGRTN
On interrupt return, check status.

 LY20-0897-7

DMKUNTFR
If no problem encountered, free storage used for CCW string and IOBLOK.

DMKGIO, DIAGRTN
Reflect the condition code and return code to the user.

DMKDSPCH
Exit to dispatcher.

DMKUNTRN
On returned error condition, convert real CSW to virtual CSW and set in user's page 0.

DMKGIO, GIOEXT
Exit via SVC 12.

Virtual Machine I/O Instruction Simulation and Interrupt Reflection

I/O Instruction Simulation

DMKVSIEX
Entry from DMKPRV to simulate I/O per VMBLOK's VMINST field.

DMKVSI, VIOSIO
On detected SIO, call -

DMKSCNVU
To locate VCHBLOK, VCUBLOK, and VDEVBLOK for the cuu called per SIO instruction.

DMKVSIEX
Determine device availability and set condition code accordingly.

DMKCCWTR
Build real channel program.

DMKDEXIN
Insert Define Extent into channel program if cache should be bypassed.

DMKIOSQV
If the operation is warranted, schedule the operation.

DMKVSI, VIOTIO
For TIO, check device status, pending interrupts, and set appropriate condition codes.

DMKVSI, VIOHIO
For HIO, check for dedicated channel, CE, CU, or device busy condition, and subchannel busy and set appropriate condition codes.

DMKVSI, VIOTCH
Check for dedicated selector or busy channel and check for pending abnormal interrupt and set appropriate condition code.

Interrupt Reflection

DMKVIOIN
Entry from DMKDSP to process the reflected virtual interrupt.

DMKSCNVU
Locate the VCHBLOK, VCUBLOK, and VDEVBLOK.

DMKVIOIN
Analyze blocks and reflect condition code to user. If condition code equals 1 (cc=1), save status from the real device (if real device) and DMKUNTFR.

DMKUNTFR
Translate and store CSW in user's page 0.

DMKVIO, VIOCC1
On TIO or HIO, free the device and set CC=1.

DMKFRET
Fret storage for the IOBLOK.

DMKDSPCH
Exit to dispatcher.

Virtual Console Simulation

DMKVSIEX
Entry for virtual console activity comes from the SCP stored in the user's virtual machine. The program's generated CCWs and data are reflected to the attached terminal used by the virtual machine operator.

DMKVCNEX
Locate and move non-TIC CCWs from the users virtual storage to a VCONCTL block.

DMKVCN, GETCCW
Update CAW and CSW in respective control block.

DMKVCN, VCNRD
For read operation, build a read console buffer VCONBUF for the input to be read from the terminal.

DMKQCORD
Queue a console read request.

DMKVCNEX
Set return address in VCONCTL VCNRDRET field.

DMKVSTCP
Spool console activity if SPOOL CONSOLE START is specified.

DMKDSPCH
Exit to dispatcher. Wait for completion.

DMKVCN, VCNWR
Calculate and obtain free storage (VCONBUF) necessary for the write to console operation.

DMKVCN, VCNMDAT
Translate and bring in user's data page and move it into VCONBUF.

DMKQCNWT
Queue a console write request.

DMKDSPCH
Exit to dispatcher.

DMKVCN, VCNSNCN
ON a sense operation, set CE and DE in the virtual PSW. Reflect the PCI flag in the PSW if the PCI flag was set in the CCW. Set the IL flag if warranted. Move the sense data from the VDEVBLOK to user storage as designated by the CCW. Update VDEVBLOK's VDEVCSW to reflect status and count.

DMKVCN, VCNCC1
On completion of I/O operation, set appropriate status for command reject, not ready protection check, incorrect length, channel program check. Set appropriate CC and CSW in users page 0. Otherwise post pending interrupt status in VMBLOK, VCHBLOK, VCUBLOK, and VDEVBLOK.

DMKVCN, FLAGTEST
If command chaining, process the next CCW.

DMKDSPCH
Exit to dispatcher.

Local Graphic I/O and Interrupt Processing

DMKGRFEN
Entry for local graphic device enable and disable function (from DMKCPVEN and unstacked CPEXBLOK). Invoking CP ENABLE/DISABLE commands, start or terminate local 3270 display (and supported print devices) and certain system console activity.

DMKGRF, LOGUSER
Format and write out the logo at the screen.

DMKGRF, ATTNINT
Unsolicited attention for RDEVBLOK (enabled).

DMKBLDVM
Build LOGON VMBLOK for logon process.

DMKIOSQR
Schedule request to clear and display the logo.

DMKDSPCH
Exit to dispatcher to wait for interrupt. Successful logon per the next interrupt begins the operation of building the user's virtual machine.

DMKSCNRU
From the IOBLOK, locate the real device blocks related to the interrupt. Analyze IOBLOK CSW and condition code and the I/O operation to determine read/write sequential action. For unit error, retry 10 times (if applicable). If recovery fails, log off. For ATTN interrupts, attempt to log on the new user if unsolicited ATTN occurs. Otherwise, set up for READ CCW string.

DMKIOSQR
Issue the SIO.

DMKDSPCH
Wait for the response.

DMKGRFIN
Local 3270 display and certain system console interrupt entry from dispatcher. On response of CE and DE, go to auxiliary processing routine address in TRQBLOK extension TRQBCRT and execute the processing routines.

DMKGRF, RDATA
Process read response of data plus ENTER key.

DMKCNSED
Edit and modify length count. Move data to caller's buffer.

DMKQCNWT
Schedule rewrite to screen (unless inhibited).

DMKIOSQR
Perform start I/O.

DMKDSPCH
Exit to dispatcher.

DMKGRFIC
Entry point to process CONTASKS queue for local 3270 devices.

DMKGRHIN
Entry point to build channel programs for 3066 devices.

DMKFREE
Get storage for IOBLOK and TRQBLOK.

DMKGRF, BLDCCWS
Execute CONTASK, if appropriate. If not -

DMKDSPCH
Exit to dispatcher.

DMKGRF, RDMINT
For read return, determine function key action and write response (if appropriate) via KEYTBL.

DMKGRFTI
Entry point for processing timer interrupts.

DMKGRCUP
Generate the 3270 orders required to update the screen status, clear the input area, or clear the output area.

DMKGRAOT
Perform APL/TEXT translation for outbound data.

DMKGRHIN
Entry point to handle 3066 CCW strings.

Locate and Validate an ISAM Read Sequence

DMKISMTR
Entry from DMKCCW modules to locate and modify an ISAM CCW string. Using the IOBLOKs IOBCAW locate the RCWTASK. Check for the ISAM read CCW.

DMKISM, CHKRD
Check for the correct ISAM sequence as follows:

1. The last CCW in the RCWTASK is a TIC.
2. This RCWTASK points to the next RCWTASK with a minimum of 2 CCWs.
3. The first modified CCW is in real storage.
4. The last byte of the ISAM read overlays the operation code of the first CCW in the next RCWTASK.
5. The TIC in the RCWTASK is to the next RCWTASK's first CCW.
6. The date address of the first CCW in the next RCWTASK is the same address of the ISAM read+1 as it is in real storage.

DMKFREE
Storage obtained for seven double words save block.

DMKISM, CHKTSK2

Institute the ISAM read modification as follows:

1. Set the read to point to the save block data area.
2. Set the CP TIC to point to the modified CCW in the same block.
3. Set the modified CCW (seek head) in the save block to point to the save block data area.
4. Set the CP TIC in the save block to return to the RCWTASK following the modified (seek head) CCW.
5. Set the search CCW in the RCWTASK to point to the data area in the same block.

DOUBLEWORD SAVE BLOCK

Read Address | (2) TIC Address
Unused |
Read Data Area
| (3) Modified CCW
(4) TIC to RCWTASK
Real Read CCW
Real TIC CCW

DMKISM, CHKTSK2

Return to DMKCCW module via SVC 12.

Scheduling CP and Virtual Machine I/O Operations and Interrupt Handling

DMKIOSQR

Entry to process CP generated I/O. Flag the IOBLOK as a CP generated event. Initiate I/O if path to real device is free (available). If not, queue the IOBLOK and return to caller.

DMKIOSQV

Entry to process I/O for virtual machine I/O operations. Mark IOBLOK as not CP initiated. Save VMBLOK address. If path to the RDEVBLOK or the RCUBLOK is busy queue the IOBLOK and return to caller.

DMKIOS, IOSTATDV

If available status, start the I/O and return to caller.

SIO Operations

DMKIOS, IOBSTART
If I/O request has not been reset, save the address of the active IOBLOK and set device busy. If the device is being reset, unflag scheduled device and scheduled control unit. Stack the IOBLOK and restart the device.

DMKIOS, IOSSIO
Set the subchannel path busy and chain the active IOBLOK from the RDEVBLOK.

DMKIOS, IOSSIO
Locate caller's CAW and issue the SIO. Check SIO completion. Returned condition code sets sequel action.

cc = 0	Indicates successful start
cc = 1	CCW stored, initiate sense operation
cc = 2	Busy condition, retry or requeue IOBLOK
cc = 3	Fatal error (not operational), stack the IOBLOK and return to caller

HIO Operations

DMKIOSHA
Entry point for halting a device. If device is not active, return to caller. If IOBLOK active, reset the IOBLOK to halt the device and mark the device reset in RDEVBLOK.

DMKIOS, IOS1OKI
If the channel path is busy with a burst mode operation, stack the IOBLOK to halt the operation when the channel path becomes available. Return to caller.

Interrupt Processing

DMKIOTIN
Entry from I/O new PSW. Check old PSW. If problem mode, save processor status in the VMBLOK.

DMKSCNRN
Locate RCHBLOK, RCUBLOK, and RDEVBLOKs for interrupt unit.

DMKVIOIN
Process dedicated channel interrupt condition. If control unit end or channel available interrupt occurs, restart the operation, if interrupt does not occur stack it.

DMKIOTIN
If the IOBLOK is not active on RDEVBLOK interrupt, call IOTGTIOB to construct an IOBLOK and continue.

DMKIOT, IOSENSE
Schedule sense operation, then go to dispatcher.

DMKIOS, IOSRSTRT
For PCI or CE interrupts, copy and stack the IOBLOK.

DMKCNSIN
Process PCI or CE interrupts, if related to local graphic device or nondedicated TP line.

DMKIOS, DOSENSE
For split seek complete interrupt, rechain the seek and reschedule operations.

DMKSTKIO
Stack IOBLOK and restart any units freed by the interrupts.

DMKDSPCH, DMKDSPA
If the system lock is held or is available, exit to the main entry of the dispatcher, DMKDSPCH. Otherwise, exit to DMKDSPA to try to redispatch RUNUSER.

Terminal Console I/O Control, START/STOP, 3210, 3215, and Others

Enabling/Disabling

DMKCNSEN
Per unstacked CPEXBLOK, on enable or disable function, check current status of the current real device and set flag in RDEVFLAG. Build CONTASK and IOBLOK.

DMKIOSQR
Issue SIO for enabling or disabling function and check return.

DMKDSPCH
Exit to dispatcher.

Process CONTASK Data

DMKCNSIC
Entry from DMKQCN module. Build I/O CCW string as defined by the console device type. Also select the proper line code to interface with the device. Place in CONTASK. For output CONTASK determine the correct translation table applicable to terminal communications (DMKTBL). To append proper control character to the data stream for the particular device type, refer to the following labels:

- **DMKCNS, INCWTTY**
  Teletypewriters

- **DMKCNS, INC2741**
  2741, 3767

- **DMKCNS, INC1050**
  1050, 1051

- **DMKCNS, INC3210**
  3210, 3215

DMKCNS, INCFINS
Attempt to start I/O by halting the current operation, if the operation is a "prepare" CCW or the input is a read and the forthcoming output is a priority write CONTASK.

DMKFREE
Get storage to build IOBLOK, if needed.

DMKCNSIN
Set return address in IOBIRA.

DMKIOSQR
Start I/O. If busy condition encountered build CPEXBLOK and queue for later execution.

DMKDSPCH
Exit to dispatcher.

Start/Stop Terminal Interruption Process

DMKCNSIN, CMBREAK
For an active input task halted, RDEVFLAG=RDEVHIO to process priority output task.

DMKFREE
Build CONTASK for reverse break CCWs.

DMKCNS, CNSBREAK
Move the input CONTASK following the last priority write output CONTASK on the chain.

DMKCNS, CNSIOUC
For unit check with intervention required, assume an attention interruption and build a "prepare" CCW for the 2741.

DMKCNS, CNSLOGF
For unit check and timeout condition - logoff the virtual machine and re-enable the line.

DMKCNS, CNSRTRY
For data check and other conditions, retry the previous operation.

DMKQCOET
Process completed output CONTASK.

DMKCNSIN
Interpret interruption status and CCW residual count for input CONTASK completion.

DMKCNS, CNINCT
Validate input data and control characters and translate to EBCDIC from line code.

DMKTRMID
Attempt to identify, if applicable, the line code identification; PTTC/EBCD or correspondence.

DMKCNSED
Perform line editing of the input buffer.

DMKCNS, CNSRT41
Prepare and issue control CCWs to request status information from the terminal.

Processing the Control CONTASK Interrupt

DMKCNSIN, CNSCTAK
For control task interrupt return, examine the interrupt status according to control task function:

- **DMKCNS, CNSTAK**
  Reset control task.

- **DMKCNS, CNSCTID**
  Device identification.

- **DMKCNS, CNSCTPR**
  Attention signal.

DMKCNS, CNSCTPR
Write "Online" interpretation of response determines retry, or build new CONTASK and execute or stack or process next CONTASK.

DMKQCOET
Process completed CONTASK requests. If no tasks remain for the terminal, set IOBLOK's IOBIRA to DMKCNSIN and link the IOBLOK to the user.

DMKDSPCH
Exit to dispatcher.

Console Scheduling

DMKQCORD
SVC entry to build CONTASK for input data. Set the input buffer to zeros.

DMKFREE
Get storage to build CONTASK.

DMKQCONQ
Stack CONTASK on RDEVBLOK, if RDEVCON was zero. If not, exit to the appropriate interrupt handler per RDEVTYPC and RDEVTYPE or -

DMKDSPCH
Exit to dispatcher.

DMKQCNWT
SVC entry to build CONTASK for output data. Strip trailing blanks from output message, modify byte count and determine real device destination.

DMKFREE
Get storage to build output CONTASK.

DMKQCN, WRDSCK
Update CONTASK CCW message byte count for the message text, terminal and line control information and (if appropriate) time stamp.

DMKCVTDT
If time stamp required, get the value for CONDATA area.

DMKVSPVP
Spool console message, if VDEVFLAG=VDEVCSPL.

DMKQCN, CRSCAN1
If message data contains carriage returns, X'15', create a separate CONTASK for each line.

DMKQCN, CHKCHAIN
For local and remote 3270s that are not accessed via VCNA, compute whether the 3270 screen will be filled. If not, operate as though the NORET parameter were specified and return to the caller so that the caller has an opportunity to generate more output lines.

DMKQCONQ, WAKEUPR
On first CONTASK or priority CONTASK, enqueue on chain from RDEVBLOK in appropriate location, then call related interrupt handler.

DMKQCONQ, WAKEMUP
If NORET or DFRET specified, build and stack CPEXBLOK to alert the interrupt handler and return via EXIT SVC otherwise go to specified interrupt handler.

DMKQCPTO
Entry via SVC to disconnect and logoff a virtual machine as a result of transmission line failures. Place the virtual machine in a wait state, VMRSTAT=VMCFWAIT.

DMKSCHDL
Alter virtual machine to unrunnable state.

DMKFREE
Get storage for message for the system operator.

DMKSCNRN, DMKSCNRD, DMKCVTBH, DMKSYSNM
Fill in message variables.

DMKQCNWT
Send the user disconnect message to the operator.

DMKQCP, DSCGTRQ
Build TRQBLOK, if needed, for 15 minute delay, schedule it, and exit via SVC.

DMKQCP, DSCTLOG
After time elapse, TRQBLOK is unstacked and VMOSTAT is set to VMKILL for inevitable DMKUSOFF logoff operation.

DMKDSPCH
Exit to dispatcher.

3704/3705 Interrupt Handler

DMKRNHIC
Entry via DMKQCN or via CPEXBLOK for 3704/3705 resource initialization. Locate the NICBLOK and check resource availability.

DMKRNH, LINEBRK
For resource unavailable, set RC=12 in CONTASK save area and return task via DMKQCNET.

DMKRNH, TAGTASK
For resource available, set CONTASK values per input and output task requirements.

DMKRNH, TASKENQ
Move CONTASK from RDEVBLOK chain to NICBLOK chain.

DMKRNH, RNSTART
On 3704/3705 available condition, search NICLIST and build an IOBLOK if required.

DMKRNHIC, RNEXLST
Search the NICBLOKs for CONTASKs to be sent to 3704/3705, build and chain for output.

DMKRNH, RNCHAIN
Perform necessary function for each resource.

DMKIOSQR
Start output I/O operations.

DMKRNH, RNICHN1
Return via R7.

DMKRNHND
Entry via SVC to schedule resource control tasks.

DMKRNH, RNHNDTK
Build control CONTASK and enqueue it for execution.

DMKRNH, STKCPEX
For NORET specified, build and stack a CPEXBLOK to perform SVC exit.

DMKRNH, RNDEXIT
Attempt to start output via GOTO DMKRNHIC.

DMKRNH, RNFDISC
Entry for 3704/3705 recovery.

DMKNLDR
Load the 3704/3705, if it was not previously loaded.

DMKFRE
Get storage to build CKPBLOK (telecommunications control block), if necessary.

DMKRNH, RNSBITS
Record active line and enabled terminal flag bits.

DMKQCOET
Clear CONTASK chains.

DMKQCPTO
Force disconnect to all active users.

DMKNLEMP
DUMP the 3704/3705.

DMKNLDR
Reload the named program.

DMKRNHND
On "IPL complete" signal, reenable resources.

DMKFRET
Release the CPEXBLOK.

DMKDSPCH
Exit to dispatcher.

DMKRNHIN
Entry via IOBLOK to perform input and output interrupt processing.

DMKRNK, RNIOERR
For input process failure. Analyze the failure and if related to the 3704/3705 and not to a particular resource, either retry or dump and reload.

DMKRNH, READBUF
Interpret response codes for each BTU received and schedule necessary control operations.

DMKRNH, CMPREAD
Generate response to a read error.

DMKRNH, CMPWRITE
Generate response to a write error.

DMKRNH, CMPCONT
Generate response to a contact task error.

DMKRNH, COMDISC
Generate response to a disconnect task error.

DMKRNH, COMCNTL
Generate response to a control task error.

DMKRNH, UNSOLIT
Generate response to an unsolicited read.

DMKQCOET
Return completed CONTASKs.

DMKRNH, RNSTART
Attempt to restart the 3704/3705.

DMKDSPCH
Exit to the dispatcher.

DMKRNHIN
Entry via IOBLOK to perform input and output interrupt processing.

DMKRNH, SCHREAD
On output, examine interrupt status per IOBLOK values and if ATTN, build and start a read CCW sequence.

DMKRNH, RNIOEUC
If unit check and fatal, dump and reload the 3704/3705.

DMKRNH, RNOREAD
If pending ATTN cleared via SIO -

DMKIOSQR
Reschedule write operations.

DMKRNH, RNSLOWDN
If unit exception, set RDEVSLOW and reschedule rejected CONTASKs.

DMKQCOET
Return only CONTASKs without CONRESP or CONSPLT set. Retain others until final response is received.

DMKRNH, RNSTART
Attempt to restart the 3704/3705.

DMKDSPCH
Exit to dispatcher.

Handling Remote 3270 with Binary Synchronous Lines

Remote Display Station and Binary Synchronous Line Enabling/Disabling

DMKRGBEN
Entered when the NETWORK ENABLE/DISABLE command is issued.

DMKFREE
Get storage for the necessary CONTASK, IOBLOK, and if applicable, BSCBLOK.

DMKRGB, LINESUP
Set up required CCWs and control data in the CONTASK for tasks. These tasks include: enabling the binary synchronous line, READ PARTITION (QUERY) processing if appropriate, enabling a device, LOGO messages, screen formatting, and disable line or device (logoff).

DMKFREE
For logon function build logon VMBLOK.

DMKIOSQR
Start line I/O or device I/O, for not busy condition.

DMKRGB, RGFTASK
For busy condition, build CPEXBLOK and exit to caller.

Request Handler for 3270 I/O Events

DMKRGBIC
Entry from DMKDSP. On a not available line condition, exit to dispatch. For available line, process the associated CONTASKs by queueing the related resource from the NICBLOK.

DMKIOS, RGSTART
Process POLL SIO on a no CONTASK queued condition.

DMKIOSQR
Process selection SIO on available resources and not in control mode per NICBLOK conditions and the CONTASK CONSTAT field.

DMKDSPCH
Exit to dispatcher.

DMKGRAOT
Perform APL/TEXT translation for outbound data.

DMKGRCUP
Generate the 3270 orders required to update the screen status, clear the input area, or clear the output area.

Secondary Interruption Processor for 3270

DMKRGAIN
Entry from DMKIOS, examine line interruption condition. Discard any of the following and go to the dispatcher: nonbinary synchronous line, copied IOBLOK, unsolicited interruption, bisync line flagged not-in-use, nonterminal class device.

DMKRGA, FATALER
For IOBFATAL condition or any nonzero condition code, free all related CONTASK, IOBLOK, IOERBLOK, and BSCBLOK.

DMKRGA, DISASTA
Log off all affected users on that line.

DMKMSWR
Send message to the system operator.

DMKDSPCH
Exit to dispatcher.

DMKRGAIN
If line or terminal response did not fall in the previous category, process via TP code branch. The code is in the fifth byte of the ending CCW or IOBCSW-8.

DMKRGC
Process the input line.

DMKDSPCH
Exit to the dispatcher.

3270 Binary Synchronous Line Error Recovery

DMKBSCER
Entry via DMKIOS to process errors related to the binary synchronous line unit check and channel error conditions. On first error pass, move the IOERBLOK pointer from the IOBLOK to the RDEVBLOK, reset retry and fatal flags, set the ERP flag and call DMKFREE.

DMKFREE
Get free storage for a work area for retry CCWs.

DMKBSC, NOTFIRST
On a not first error condition, test for unrecoverable error condition. Unrecoverable errors include: program check, protection check, chaining check, equipment check, interface control check and channel control checks. If one of these, notify the system operator. Reset flags, initiate error recording and

DMKFREE
Free IOERBLOK.

DMKIOSQR
Go back to scheduler.

DMKRGA
Analyze TP code, sense data CSW residual count and retry count to determine retry or IOBFATAL flag setting.

Real Storage Allocation and Page Management

Page Requests

DMKPTRAN
Entry to translate the virtual address provided by the caller into a real storage address. DMKPTRAN is usually called via the TRANS macro.

DMKPTR, RESTART
Returns to the caller if the virtual address in R1 is beyond the range of the user's storage size.

DMKPTR, ADDROK
Uses the Load Real Address (LRA) instruction to see if the page is resident.

DMKPTR, CHKMOVE
Moves a page in the greater than 16 Mb area to the less than 16 Mb area if the caller specified the BRING option, and if the page is the user's virtual page 0 or if the caller did not specify the VFAULT option.

DMKPTR, TESTLOCK
Locks a resident page in storage if the caller so specified.

DMKPTR, GETRADD
Sets real address in R2, makes PAGTABLE entry valid, sets condition code equal to 0, and exits to the caller.

DMKPTR, INTRAN

For a page that is not resident but is in transit, if the caller did not specify the DEFER option, it stacks a CPEXBLOK for return to DMKPTRAN and exits. If the caller did specify the DEFER option, it locates the CPEXBLOK for the real page or swap set requested, and chains another CPEXBLOK with a return address of TRANRETN to the same chain.

DMKPTR, TRANRETN

When page is no longer in transit, it restores registers and returns to RESTART for processing.

DMKPTR, READPAGE

For a page that is non-resident and not in transit, if the BRING option was not supplied by the caller, it exits to the caller with a condition code of 1. Otherwise, it checks for a DEFER option as in INTRAN. Reads in a page if necessary.

DMKPTR, SELECT

Determines if free lists need to be replenished. Calls DMKSELCT if they do.

DMKSELCT

Replenishes the free lists.

DMKPTR, SWAPFLT

Entry for swap faults. If swap set is in transit or enqueued for allocation, it branches to INTRAN. Otherwise it reclaims all pages from the free list if possible, or if not, sets up to do I/O.

DMKPTR, DOIO

Reads in a page from backing storage.

DMKPTR, SWAPIO

Stacks a CPEXBLOK to set up for I/O for a swap in request.

DMKPTR, CKDEFER

When the caller specified the DEFER option, it builds a CPEXBLOK to return to the user after the page is in storage.

DMKPTR, PAGEIN

When the page is read into storage, it removes the user from the wait state and updates the lock count if required.

Obtain, Return, Lock, and Unlock a Real Page Frame

DMKPTRFR

If the caller is DMKFREE trying to extend system free storage, it returns to the caller the storage in the EXTEND buffer. Otherwise, it branches to the PAGFREE subroutine to get a page of storage from the less than 16 Mb free list.

 LY20-0897-7

DMKPTREP
Obtain a page frame from the >16Mb free list or <16Mb free list, whichever is larger.

DMKPTR, PAGFREE
This entry is for requests for a page frame from below the 16 MB line. If no page frame is available, it chains a CPEXBLOK to the proper queue of tasks waiting for pages and exits to the dispatcher. If a page frame is available, it branches to GETFREE.

DMKPTR, PAGFREE2
This entry is for requests for a page frame anywhere in storage. It is called only for virtual storage page faults, not for CP pages. It will try to get a page frame from the greater than 16 MB free list if that list has more page frames than the less than 16 Mb free list.

DMKPTR, GETFREE
Obtains a page frame from the less than 16 Mb free list by unchaining the proper CORTABLE entry, and exits to the caller.

DMKPTR, GETFREE2
Obtains a page frame from the greater than 16 Mb free list by unchaining the proper CORTABLE entry, and exits to the caller.

DMKSELFE
Entered via an unstack of a CPEXBLOK built by DMKPTRFR. Checks for available pages on the flush list. If none are available, it scans the core table for a page.

DMKPTR, SELECT
Determines whether the free lists need to be replenished. If they do, it calls DMKSELCT.

DMKSELCT
Replenishes the free lists.

DMKPTTFT
Processes pages to be returned by chaining them to the appropriate free list. It processes waiting page requests first.

DMKPTRLK
Entered to lock a page in real storage. If the page is already locked, it adds one to the lock count and exits. If the page is not locked, it sets a flag to indicate that the page is locked, sets the lock count to one, and exits to the caller.

DMKPTRUL
Entered to unlock a page from real storage. It reduces the lock count by one and exits to the caller. If the lock count is now equal to zero, it resets the bit that indicated that the page was locked.

Physical Swap-Out

DMKSWAPO
Entry to process physical swap-out. Called by DMKSEL. R2 points to the VMBLOK of the owner who has a swapset to write.

DMKSWA, NEXTSET
Builds a swap set block and initializes it.

DMKPGTSW
Obtains a swap area DASD or Paging Storage slot. On return to DMKSWA, if R1 equals zero, the swap set is paged out to a paging area. Otherwise, the swap-out is done to a swap area.

DMKSWA, SWAPOINC
When swap request is met, it exits to SWAPRET to return to the caller.

DMKSWA, SWAPRET
Returns to the caller.

DMKSWAPD
Entered after completion of a swap-out, and entered via an unstack of a CPEXBLOK. For each valid entry in the SSBLOK, it calls DMKPTRPS to get the page table and swap table addresses, resets the appropriate bits in the swap table for pages swapped out, and calls DMKPTTFT to put frames on the free list.

Logical Swap-In/Prepaging

DMKSWAPI
Called from DMKSCH when a user is added to queue, to prepage a user's pages. R1 contains the address of the SCBLOK.

DMKSWA, NEXTSSBX
Loops through the chain of SSBLOKs anchored at SCBFSSB looking for SSBLOKs that are not marked as old. If it can't find any, it branches to SWAPIWSX to perform a logical swap-in.

DMKSWA, GETVIRT
Physically prepages a swap set. Calls DMKPTRAN, specifying BRING and VFAULT options.

DMKPTRAN
Reads the swap set into storage.

DMKSWA, SWAPIWSX
Logically swaps in a user's pages.

DMKSWA, SWAPIRET
Returns to caller.

Reading/Writing a DASD or Paging Storage Page to/from Virtual Storage

Virtual Storage Management – Non-EC Mode

DMKRPAGT
Entered via SVC call to read DASD page or page from Paging Storage into storage.

DMKRPASV
Entered via SVC call to read SYSSPOOL's virtual pages into storage.

DMKPGUPR
Release DASD space that was previously occupied by this virtual storage page.

DMKRPA, RESIDENT
Remove resident page frames from the user list.

DMKPTTFT
Place these page frames on the free list.

DMKRPA, STORDASD
Update the SWPTABLE with disk address in R0.

DMKPTRAN
Bring the page into storage.

DMKRPA, EXIT
Put real storage address of the virtual page is passed back to the caller in R2.

DMKRPAPT
Entered via SVC call to write out a page to DASD storage or Paging Storage.

DMKPTRAN
Locate the page to be moved and lock it.

DMKRPAPT
Store all registers in CPEXBLOK and flag CPEXR0 as a write request.

DMKPAGIO
Write the page.

DMKRPA, IORETN
Decrease page wait count. If zero results, take user out of page wait.

DMKPTRUL
Unlock the page frame. Return to caller.

Virtual Storage Management – EC Mode

DMKVATAB
Entry via BALR when an EC mode virtual machine needs a shadow table generation and update or purge operation.

DMKVATMD
Get storage to create shadow table. Flag VMBLOK to show shadow table existence.

DMKVATBC
Free shadow page, segment and copy segment when user leaves EC mode or alters CR 0.

DMKVATSI
Entry via BALR to selectively invalidate a shadow page table entry.

DMKVAURN
Entry to perform third level to first level translations and third level translations to second level address translations. Use TRANS macro to access virtual segment and page tables to get the virtual page into real storage.

DMKVATLA
Using the TRANS macro to access the virtual segment and page tables, pass the resulting page and displacement to DMKPRVLG.

DMKVATPX
Invoked by DMKPRGIN when a paging exception is received for an EC mode virtual machine.

DMKVAT, SETUPEX
Perform set up operation and develop page table address.

DMKPTRAN
Get the page.

DMKVATPX
Update the shadow table.

DMKVATSX
Invoked by DMKPRGIN when a segment exception is received for an EC mode virtual machine.

DMKVAT, SETUPEX
Perform setup operation, then invalidate the shadow page table or if none exists, allocate a new shadow table and set it invalid.

DMKVATPF
Entered via DMKVATPG from DMKPRG to simulate pseudo page fault interrupts when a paging exception occurs with pseudo page fault interrupts enabled.

DMKPTRAN
Bring in the DASD or Paging Storage page.

DMKPRGSM
Reflect program check X'14' to the user.

DMKVAT, PAGRES
Entered when the page becomes resident in storage. Build the PGBLOK, set high order bit in the translation exception address field.

DMKDSPCH
Exit to dispatcher.

Allocation and Deallocation of DASD Slots and Paging Storage Pages

DMKPGTPG
Entry to search and allocate a page (slot) of DASD space or a page from Paging Storage for virtual storage paging. Points to DMKSYSPG for ALOCBLOKs allocated as type PG.

DMKPGTPM
Entry to search and allocate a page of DASD space for page migration. DMKPGTPM is passed a SYSPLIST which it uses to begin its search for DASD space. Within a given SYSPAG level, uses the first ALOCBLOK that has space available. Finds the RECBLOK in use, or finds the RECBLOK for the cylinder that is closest to the current head position and that has an available page. If it finds an allocated cylinder with an available page, it calculates the address of the allocated page, marks the page as allocated, increases the number of pages in use, and places the address in R1. Returns to caller.

If no allocated cylinder has an available page, finds the next available cylinder. Then constructs a new RECBLOK for the allocated cylinder, and chains it to the existing RECBLOKS before calculating address and returning to caller.

DMKPGTSG
Entry to search and allocate a page of DASD space for spool file records. (Same operation as DMKPGTPM).

DMKPGUPR/DMKPGUSD/DMKPGUSP
Entry to deallocate a DASD or Paging Storage page used for paging or spooling. If the page has not yet been allocated, it resets the SWPTABLE entry to zero and returns to the caller. Otherwise it locates the RDEVBLOK for the device on which the page has been allocated by indexing into the OWNED list with the device code supplied in the DASD address. It locates the ALOCBLOK and RECBLOK for the cylinder on which the page is allocated. It then finds the bit which represents the page and sets it to zero, and decreases the number of pages in use. If the number of pages in use is zero, it deallocates the cylinder. Otherwise, it resets the SWPTABLE entry to zero, and returns to the caller.

DMKPGUSR

Entry to release a set of DASD pages that belongs to a spool file or user that is no longer needed. Operation is essentially the same as for DMKPGUPR, but when the appropriate RDEVBLOK and RECBLOK are located, it exclusively ORs the bit mask in the dummy RECBLOK against the map in the real RECBLOK to zero out the bits for the pages being deallocated.

DMKPGTCG/DMKPGTDG/DMKPGTDT

Entry to allocate contiguous pages for 370X dump, system dump spool file, and system dump, respectively. It scans the ALOCBLOK for the device to determine if the request can be satisfied. If not, it starts over. If the request can be satisfied, it marks the chosen cylinders as allocated and returns the CCPD address to the caller in GR1. It builds and chains the required RECBLOKs.

DMKPGUDU

Entry to deallocate contiguous DASD pages that were previously allocated for a CP system dump.

DMKPGUSW

Entry to deallocate N contiguous DASD or Paging Storage pages that were used for swapped pages, where N is defined in the swapset block. Deallocation processing is as normal, except the RECBLOK allocation map uses one bit per swap set, rather than one bit per DASD slot or Paging Storage page.

DMKPGUPP

Entry to deallocate DASD or Paging Storage pages that were formerly used for pages that are now in real storage and about to be swapped out.

DMKPGUVG

DMKPGT contains an internal table, PAGETABL, in which the allocation of page frames for the CP paging VMBLOK is kept. The PAGETABL is scanned for a zero bit denoting the page frame is available. The page is marked allocated by setting the bit to one and the address of the page frame is returned to the caller in R1. If no page frames are available, a CPEXBLOK is built and queued to the deferred request chain.

DMKPGUVR

Entry to release a page of virtual storage. Check the chain of deferred requests. If there are none, reset the page bit in the PAGETBL to 0 and exit to the caller. Otherwise, give the page to the first requestor in the deferred chain and stack his CPEXBLOK for the dispatcher.

Shared Segment Storage Management

DMKATSCF
Entry via SVC from the command processor if an ADSTOP, TRACE, or STORE command is to alter a shared page. The virtual machine issuing the CP command will be unshared from the named system, that is, given a private copy.

DMKERMSG
The running virtual machine is informed of the share page violation.

DMKVMASH
Entered from DMPDSP or DMKPTR via BALR. The protected shared page tables are examined for hardware change bit being on. The resulting condition code is reflected to the caller.

DMKVMASW
Entered to switch the virtual machine from one set of page tables to the other.

Temporary Disk Storage Management

DMKTDKGT
Entry to allocate temporary disk (TDISK) space. Upon entry R1 contains the device type and R0 contains the amount of requested space. For FBA devices, DMKTDKGT first converts the block count in R0 to pseudo cylinder values, before it allocates space. DMKTDKGT tries to locate an ALOCBLOK on the allocation chain, and then tries to locate enough contiguous unallocated cylinders on the ALOCMAP. If it can't find enough available space, it places a zero in R8 and returns to the caller. If it finds enough space, it places the address of the first allocated cylinder in R1 and the RDEVBLOK pointer in R8, and returns to the caller.

DMKTDKRL
Entry to deallocate TDISK space. DMKTDKRL uses the address of the RDEVBLOK in R8 to locate the ALOCBLOK for TDISK space on the allocation chain. For FBA devices, the block counts in R0 (number of blocks) and R1 (first block number) are converted to pseudo cylinder values. If the TDISK CLEAR option is in effect, DMKCPXCK is called to build CPEXBLOKs, and DMKZTD is called to clear TDISK space. DMKTDKRL then returns to the caller. If the TDISK CLEAR option is not in effect, DMKTDKRL sets the appropriate bytes in the ALOCMAP to X'02', and returns to the caller.

Paging I/O Scheduler

DMKPAGIO
Entry to initiate page I/O activity. Uses preformatted IOBLOKs from the IOBSTACK for requests on CKD or FBA DASD, or DMKPAGEX for requests on extended count-key-data devices; fills in the CCWs with DASD operation code and values from CPEXBLOK swap table and core table. Chains the CPEXBLOK on the in-transit queue. For swap requests, it gets the IOBLOK extension from DMKPAGXS and fills it in with CCWs and the SSBLOK.

DMKPAG, GETRDEV
Finds the Paging RDEVBLOK.

DMKPAG, SLOTSORT
For a swapping request, it slot-sorts and queues IOBLOKs. For a paging request, it slot-sorts IOBLOKs and temporarily queues them to the appropriate dummy RDEVBLOK.

DMKPAG, FINDIOB
Searches paging IOBLOKs to find those seeking the same cylinder address. If it finds any, it chains the channel programs together with TICs. It does not chain 3880-11/21 requests or swap set requests together.

DMKPAG, QUEUEIO
For 3880-11/21 requests and swap set requests, schedules an immediate I/O operation.

DMKIOSQR
Starts the I/O.

DMKPAG, EXITX
Starts I/O for paging requests by unstacking CPEXBLOKs from the dummy in transit queue and placing them on the in transit queue (DMKPAGQ). It dequeues from the dummy RDEVBLOK the corresponding IOBLOK for each paging CPEXBLOK.

DMKIOSQR
Starts the I/O.

DMKDSPCH
Exits to the dispatcher.

DMKPAH, UNTRANS
Upon interrupt return, unchains the CPEXBLOK from the intransit queue.

DMKSTKCP
Stacks all deferred requests for execution.

DMKPAH, UNSTACK1
Returns IOBLOK to IOBSTACK or DMKPAGEX.
Returns IOBLOK extension to DMKPAGXS.

DMKDSPCH
Exits to dispatcher.

Release Virtual Storage Pages

DMKPGSSS
Entry to release partial virtual storage. Per R1 (address of first page to be released) and R2 (address of last page to be released) set partial entry flag.

DMKPGSPO
Entry to check for shared segments and decrement usage count. Store registers and flag full entry condition. Examine VMSHRSYS for shared segments. If so, decrease use count. On zero use count unchain the SHRTABLE from the active list.

DMKPGS, CKCLEAR
On NOCLEAR exit to caller. If not, store number of release pages in R8.

DMKPGS, PGOUT2
Locate page and swap tables for the segment to be released and index to the entry for the first page.

DMKPTRAN
Initiate paging.

DMKPGS, NEXTPAGE
When pages are to be released, it checks for the end of the segment. Otherwise, it exits to the caller.

DMKDSPCH
Exit to caller.

DMKPGSPS
Entry to release storage containing a named system passed by the caller. If register one is nonzero, search the page tables looking for a header equal to the named system. If found, release the swap and page tables and build new ones, if the address range still lies within the user's virtual storage size. If register one is zero, release and rebuild swap and segment tables for all segments above the normal virtual storage size that do not have SHRTABLE entries.

Free Storage Management

DMKFRTTR
Entry to assign storage to free storage management during CP initialization and whenever DMKPTRFR obtains a page from the dynamic paging area for use as free storage.

DMKFREE
Entry to obtain a block of storage, validate input doubleword request (R0).

DMKFRERC
Entry to obtain a block of free storage and return a condition code one if the request cannot be satisfied.

DMKFRE, CHEKSIZE
Checks the size of the request. For subpool size, see "DMKFRE, FREESUB" entry. For larger than 1024 doublewords, see "DMKFRE, FREE01" entry.

DMKFRE, FREESUB
On subpool size request, index into SUBTABLE. For correct size block found, remove block from chain and put the address of the block in R1. Return to caller.

DMKFRE, FREE02
For subpool size not found, get next large subpool size. Test for optimum X'28'. For correct size block found, remove block from chain and put the address of the block in R1. Return to caller.

DMKPTRCO
Obtains a page from the dynamic paging area for DMKFREEA.

DMKFREEA
Attempts to fill request for a cache-aligned block. If unavailable, calls DMKPTRCO for a page of storage from the less than 16 Mb free list. If again unavailable, branches to DMKFREEB to perform the normal free logic.

DMKFREEP
Attempts to fill request for a PRIME storage block. If unavailable, branches to DMKFREEB to perform the normal free logic.

DMKFRE, FREE16
If no block can be found to honor user request, call DMKPTRFR.

DMKPTRFR
Fetch a page from the dynamic paging area. Chain it to the free storage chain. Processing then continues. See entry DMKFRE, FREESUB.

DMKFRE, FREE01

For a large block request, search the large storage chain. For an equal or larger block found, remove block of request size from the chain, put the address in R1, and return to the caller. Otherwise, see "DMKFRE, FREE16" entry.

DMKFRTRS

Entry to return all subpool blocks to the free storage chain per the SUBTABLE reference, as each subpool block is released, its address and length are placed in R1 and R2 respectively. Branch and link to FRET05 to return the block to the free storage chain (DMKFRELS). Repeat action through all subpools. Return to caller.

DMKFRTRA

Scans cache-aligned subpools looking for pages to return to the dynamic paging area. It is invoked whenever DMKFRERS, the regular subpool collector, is invoked.

When an entire page of contiguous blocks of cache-aligned storage is found, it loads R7 with the address of the first block (the address of the page to be returned).

DMKFRTSN

Entry to search the large storage chain for blocks needing to be returned to the dynamic paging area.

DMKFRTT

Entry to restore block to subpool or free storage. Per R0 and R1 (number of doublewords to be released and and address of the first double word, respectively), the subpool sized block is returned to the appropriate subpool. Update the pointer in the SUBTABLE.

DMKFRTT

Entry to handle requests to return storage that cannot be handled by the CP Assist Fret instruction at DMKFRET.

DMKFRT, FRET01

Return a block of storage to free storage chain by merging into the chain storage addresses in an ascending order of sequence. Return to caller if the block is not the size of a page or more.

DMKFRT, FRET22

If the block includes a whole page frame or more and is in the dynamic paging area, branch and link to FRET22J, which calls DMKPTTFT to give the block back for paging. Call DMKFRTSN to search the large storage chain for other blocks needing to be returned to the dynamic paging area. Return to caller.

DMKFRTTE

Entry to return storage (known to have been obtained from the dynamic paging area) of subpool size or greater to the free storage chain.

Virtual Machine Initialization and Termination

Attaching a Virtual Machine to the System

DMKCNSIN
Entered via interrupt from a console or terminal (not displays) device. If appropriate, determine and store device type in the RDEVBLOK. Write the system online message. Sets up to receive attention interrupt.

DMKBLDVM
On attention interrupt, build skeleton VMBLOK for LOGONxxx.

DMKCFMBK
Send read CCWs to the terminal for LOGON or DIAL response.

DMKTRMID
On response determine translate tables to be used.

DMKCFMBK
Validate command and transfer to DMKLOGON.

DMKLOGON
LOGON command execution.

DMKRPWEP
Verify LOGON password.

DMKDIAL
Dial access linkage to multiaccess system.

DMKUDR
Via user directory access, validate user logon eligibility. On acceptance of eligibility, that is the successful completion of logon, build and allocate control blocks and linkages for the user's virtual machine.

CP Initialization and Termination Procedures

Loading the CP Nucleus on a Cold Machine

The value of CPID controls the actions of DMKCKP. For a cold machine, CPID is neither "CPCP" nor "WARM." In this case, DMKCKP simply loads DMKSAV and transfers control to DMKSAVRS.

IPL
The 24 byte IPL sequence containing the IPL PSW and a read CCW is loaded from the IPL device. The read CCW reads DMKCKP from the IPL device into location X'1000'. (The load point for a V=R system is the V=R size plus X'2000', or DMKSLC plus X'1000'.) The IPL PSW with an instruction address pointing to DMKCKPT is loaded.

DMKCKPT
Initial entry point to load the system.

DMKCKP, READREST
Load the rest of the checkpoint program (DMKCKD, DMKCKF, DMKCKH, DMKCKM, DMKERP, DMKCKW, and DMKCKN).

DMKCKP, COLD
If CPID is not equal to “CPCP” or “WARM,” then this is a COLD machine. Move “COLD” into CPID. Load both pages of DMKSAV into the storage location assigned by the loader during system generation. Go to DMKSAVRS.

DMKSAVRS
Load the nucleus up to DMKSAV. For processors with the Test Block facility, issue a Test Block to each 4K block of main storage before loading that page from the nucleus area. Go to DMKCPINT to initialize the system. See System Initialization for more details.

System Initialization

The value of CPID controls the actions of the system initialization routines. If CPID = “COLD,” then this is a normal system IPL. If CPID = “WARM,” then this is an automatic WARM start re-IPL following a system abend or SHUTDOWN REIPL.

DMKCPINT
Entry point to perform system initialization.

DMKCPI
Initialize the new PSWs, the control registers, the TOD clock comparator, the CPU timer, and the CPU ID field of the PSA.

DMKCPI, INITREAL
Check if multiprocessing hardware is present.

DMKCPI, CHKCSS
Check if the channel set switching feature is installed.

DMKCPI, APCHKADR
For systems defined as AP during system generation, check if the other processor is available.

DMKCPI, CALLSTA
Call DMKSTANT.

DMKSTANT
Determine the real storage size, initialize the CORTABLE, allocate free storage and initialize system paging tables.

DMKCPI, CLEARCPA
If the ECPS microcode assist is not present or the wrong level, then replace each microcode instruction in the system with a NO-OP.

DMKCPI, OBTNSAVE
Obtain storage for use by DMKPTR in case the system needs to extend.

DMKCPI, GTXBLOOP
For systems defined as AP during system generation, if the other processor is available, then get storage for use by the switch macro and call DMKAPIPR.

DMKAPIPR
Called only if the other processor is available. Initialize the PSAs for both processors and start prefixing on both processors.

DMKCPI
Call DMKMNTIO.

DMKMNTIO
Entry point to mount I/O devices defined during system generation which are available.

DMKMNT, DOHIO
Issue a HIO to each I/O device defined in DMKRIO. If the device is not available, then go to MOUNTDVI to get the next device. Otherwise continue.

DMKMNT, RELEASE
Issue a RELEASE CCW to each DASD and TAPE which is available.

DMKMNT, READLABL
Read the label from each DASD and TAPE which is available.

DMKMNT, BUILDRDC
Build RDCBLOKs for FBA devices which are available.

DMKMNT, ALLOCSIO
Read the allocation record from DASDs which are available.

DMKMNT, CHKMSS
Issue a SUSPEND CCW to MSS devices which are available.

DMKMNT, CKCPOWND
If this is a CP owned DASD, then call DMKALOCP.

DMKALOCP
Entry point to process CP owned DASD.

DMKALO, ALOCBILD
Create the Maxi-ALOCBLOK for this device.

DMKALO, FLAGRDEV
Mark this device's RDEVBLOK as CP OWNED.

DMKALO, SAVEDRCT
Compute the displacement into the system file table for the SYSOWN volume. Save the directory and override pointers. If this is the IPL volume, then save the address of this table entry as the location of the primary directory pointer.

DMKPSTIN
Build ALOCBLOKs and RECBLOKs for each Paging Storage area specification.

DMKXSTOR
Called by DMKPST if SYSXSTOR macro was used (instead of SYSPAG macro) to build ALOCBLOKs and RECBLOKs for each Paging Storage area specification.

DMKVFIIN
Initialize Vector Facility.

DMKMNT, DEVONLIN
For each available device, mark the device and control unit as online.

DMKMNT, MOUNTDVI
Repeat DOHIO through DEVONLIN for each I/O device defined in DMKRIO.

DMKMNT, MOUNTCHI
After mounting all the devices in the system, call DMKCPZFD to initialize the cached control units.

DMKMNT
If the second processor is available in a multiprocessor configuration, then repeat DOHIO through MOUNTDVI for the second processor.

DMKMNT, SHIF3
Recompute the RDEVBLOK indexes if necessary.

DMKCPI
Call DMKSEGCP.

DMKSEGCP
Entry point to set up CP's segment, page, core and swap tables.

DMKSEG, VBUFFOK
Call DMKBLDRT to build CP's segment and page tables.

DMKSEG, GETSEG
Fill in the entries in CP's swap and core tables.

DMKSEG, CALCUL
Calculate the number of fixed head and moveable head paging pages.

DMKCPI, SSMTEST
Check to see if the VM assist is available.

DMKCPI, S370EPGM
Check to see if the 370E facility is available.

DMKCPI
Call DMKOPERC.

DMKOPERC
Entry point to locate the operator's console.

DMKOPE
Locate the operator's console.

DMKOPE, ONLINE
Move the system ID into the status area.

DMKOPE, WRITEINT
If CPID is "WARM," then this is an automatic WARM start following a system abend or SHUTDOWN REIPL. If CPID is "WARM" and the system is restarting after an ABEND, then display "VM/SP System restart due to system failure." If CPID is "WARM" and the system is restarting following a SHUTDOWN REIPL, then display "VM/SP System restart due to shutdown REIPL."

DMKOPE, INITWRIT
Display "VM/SP Release *x*, Service Level *xxxx*, created on *mm/dd/yy* at *hh:mm:ss*."

DMKCPI
Call DMKTODIN.

DMKTODIN
Entry point to initialize the time of day clock.

DMKTOD
Display "It is now *hh:mm:ss time-zone day mm/dd/yy*."

DMKTOD, GETDATE
If CPID is not "WARM," then allow the operator to change the date and time. If CPID is "WARM," then this is an automatic WARM start following a system abend or SHUTDOWN REIPL.

DMKTOD, NOTCHNG
Schedule a TRQBLOK for midnight tonight. This TRQBLOK will cause the midnight message to be displayed by DMKSCHMD.

DMKTOD
Schedule a TRQBLOK for 60 minutes from now. This TRQBLOK will cause storage subpools to be cleaned up by DMKTMRFR.

DMKCPI, APCHECK
For systems defined as AP during system generation, if the other processor is available, then call DMKAPICK and DMKCLKCK.

DMKAPICK
Entry point to test the other processor's clock, test for the virtual machine assist on the other processor, and complete initialization of the other processor.

DMKCLKCK
Entry point to synchronize the time-of-day clocks.

DMKCPI, DRCTSET
If the IPL volume contains a directory pointer, try to load it. Otherwise, notify the operator and try to load first the first valid directory found in SYSOWN order. Each time DMKUDRBV fails, try another backup directory from the SYSOWN list. If no more SYSOWN volumes are available, issue abend CPI002. If DMKUDRBV is successful, continue. If a valid override pointer is found on the save volume as the directory, call DMKUDROV to load it. Otherwise, notify the operator that the system defaults are used.

DMKCPI
Call DMKCPJNT.

DMKCPJNT
Entry point to continue CP initialization.

DMKCPJ, TLOOP
Verify that the interval timer is running.

DMKCPJ, OWNDECK
Verify that the warm start, checkpoint, and error recording volumes are mounted. If one of the volumes is missing, issue message DMKCPJ912W to the operator. Then load a disabled wait state PSW (code 009). If an AUTOIPL is requested, bypass getting the operator's input and initiate the WARM START.

DMKCPJ, WARMTEST
If CPID is not "WARM," then display "Start ((WARM|CKPT|FORCE|COLD) (DRAIN)) | (SHUTDOWN) :" and get the response. If CPID is "WARM," then this is an automatic WARM start following a system abend or SHUTDOWN REIPL.

DMKCPJ, PROEND
If response was "SHUTDOWN," then load a disabled wait state PSW (code 006).

DMKCPJ, AUTOWARM
If response was "WARM," "CKPT," "FORCE," or "COLD," then call DMKWRMST.

DMKWRMST
Entry point to handle WARM, CKPT, FORCE, or COLD starts. See the separate discussion of each type of start for more detail.

DMKCPJ
Call DMKIDUMP.

DMKIDUMP
Entry point to locate DASD dump space, update allocation counts, mark the 3705s as unloaded, and clear T-disk space.

DMKIDU, DMPALOC
Scan the ALOCBLOKs searching for enough contiguous CKD DASD space to hold a system dump.

DMKIDU, RECALOC
Create RECBLOKs for the dump cylinders on CKD DASD. Go to QUEUEIO.

DMKIDU, SPCFNDC
Fill in the dump SPOOL file SFBLOK for FBA DASD.

DMKIDU, DUMPOK
Update the allocation counts for FBA devices.

DMKIDU, SPCFOUND
Update the allocation counts for CKD devices.

DMKIDU
Call DMKPTRAN to write DMKSYM to the first record of the dump cylinder.

DMKIDU, ALOCLUP
Call DMKVDGAL to create ALOCBLOKs for each CP-owned volume.

DMKVDG
Create and chain ALOCBLOKs from the SYSPLIST and RDEVBLOK chain anchors.

DMKVDH
Build and initialize ALOCBLOKs and RECBLOKs, and check for spool files for DMKVDG.

DMKVDHOR
Called to reorder ALOCBLOKs following their creation.

DMKIDU, NR3705
Mark the 3705 RDEVBLOKs as not ready.

DMKIDU, RELTDK
Stack CPEXBLOKs to clear T-disk space.

DMKCPJ
Call DMKOPELO to logon the system operator.

DMKCPJ, MAPMSG

Display:

```
DMKCPJ957I   STORAGE SIZE = xxxxx K,
             NUCLEUS SIZE = xxx K,
             DYNAMIC PAGING SIZE = xxxxx K,
             TRACE TABLE SIZE = xxx K,
             FREE STORAGE SIZE = xxxx K,
             VIRTUAL=REAL SIZE = xxxxx K
```

DMKCPJ, COLDSPEC

If "DRAIN" was not specified, then call DMKCSORD to DRAIN (or STOP) 1) all system punches and printers for a COLD start, or 2) all punches and printers which are not drained for WARM, CKPT, or FORCE starts.

DMKCPJ, PPMAP

Call DMKHVDPP to initialize the program product bit map.

DMKCPJ

Call DMKSEGWR.

DMKSEGWR

Entry point to read in all pageable modules between DMKSAV and DMKCKP in order to force them to the backing device (that is, to create page image copies).

DMKCPJ

Call DMKIDUSF to get a valid SPOOL file ID for the dump SPOOL file.

DMKCPJ, STARTSYS

Call DMKIOEFL to format the error recording area if necessary and then DMKCPJ initializes the machine check new PSW.

DMKCPJ, LOAD37X

Call DMKNLDR to load 370X programs into the enabled 370X devices.

DMKCPJ, CPJENAB

Call DMKCPVAE to re-enable terminal and graphics devices and call DMKNETAE to re-enable lines and stations.

DMKCPJ

Call DMKOPEAC to auto start MONITOR and auto log the AUTOLOG1 virtual machine.

DMKCPJ

Initialize IUCV control blocks.

DMKCPJ, CPJMIH

Create and schedule missing interrupt handler TRQBLOKS.

DMKCPI, CPIEND
Display "DMKCPI966I Initialization complete."

DMKCPI
Call DMKOPEDC to disconnect the operator if the operator was not logged on when the system ABENDED.

DMKCPI, INITDONE
Go to DMKDSPCH to begin dispatching virtual machines.

Warm Start Processing

DMKWRMST
Entry point to handle WARM, CKPT, FORCE, or COLD starts. DMKWRMST is called by DMKCPJ during system initialization. For a WARM start, R2=02. Call DMKCKTMP.

DMKCKTMP
Get storage to create and initialize spool file bit map.

DMKWRM, EN3705
For 37xx devices create CKPBLOKs from the information saved in the terminal buffer.

DMKWRM, ENR3270
Enable binary synchronous lines by clearing NICBLOK offline flag (if appropriate).

DMKWRM, ACNTRT
Reconstruct the saved accounting ACNTBLOKs from the warm start data and chain them from the accounting card anchor (DMKRSPAC).

DMKWRM, WARMLOG
Retrieve the log message from the warm start area and save it in a buffer pointed to by DMKSYSLG.

DMKWRM, WARMSPL
Reconstruct the printer, punch, open reader, and delete SFBLOK chains from the data on the warm start area.

DMKWRM, WARMRECA
Update ALOCBLOKs and create corresponding RECBLOKs from the data on the warm start area.

DMKWRM, WARMHOLD
Reconstruct the SPOOL hold queue blocks (SHQBLOKs) from the data on the warm start area.

DMKWRM, RESTHASH
Reconstruct the Reader Hash Table.

DMKWRM, RESTEXT
Reconstruct the Reader Hash Table extension pages.

DMKWRMSY
Reconstruct SYSSPOOL's virtual storage.

DMKWRM
Call DMKWRNWM.

DMKWRNWM
Put open reader files in SYSSPOOL's virtual storage. Turn on the system-unique bit in the spool file bit map for each printer, punch, open reader, SHQBLOK, and spool RDEVBLOK.

DMKWRM, WARMCLR
Zero the first 8 bytes of the first record on the warm start area. Save the starting time for this system (STARTIME) in the first record and in DMKRSPCV. DMKCKP will compare these two values to ensure that the correct volume is mounted before checkpointing the system.

Checkpoint (CKPT) and FORCE Start Processing

DMKWRMST
Entry point to handle WARM, CKPT, FORCE, or COLD starts. DMKWRMST is called by DMKCPJ during system initialization. For a CKPT start, R2=20. For a FORCE start, R2=40. Call DMKCKTMP.

DMKWRM, ENDER1
Call DMKCKVWM.

DMKCKTMP
Get storage to create and initialize spool file bit map.

DMKCKVWM
Entry point to handle CKPT or FORCE start. Also, verify that a CKPT start is valid—if not, issue error message 917E and enter wait state.

DMKCKVWM, PROCCKPT
Read the first/next page from the checkpoint area.

DMKCKVWM, SCANPAGE
Get the first/next checkpoint slot on this page.

DMKCKVWM, CKVWM2B
For real device slots, restore the checkpointed RDEVBLOK data.

DMKCKVWM, CKVWM2G
For SPOOL hold queue slots, reconstruct checkpointed SPOOL hold queue blocks (SHQBLOKs) from the checkpointed data. Chain the SHQBLOKs off the SHQBLOK chain anchor (DMKRSPHQ).

DMKCKV, NEWTASK

For SPOOL file slots, reconstruct checkpointed SPOOL file blocks (SFBLOKs) from the checkpointed data. This is done by building task blocks (TSKBLOKs)—one for each SPOOL file to be reconstructed. The SPOOL file links (SPLINKs) are followed until each SPOOL file block has been retrieved and all the pages have been allocated.

DMKCKV, CKVWM5

For *non-dump* SPOOL file slots, allocate the DASD pages used by each reconstructed SPOOL file by beginning with the first page and following the SPOOL page linkage pointers (SPLINK blocks), allocating each page, until the last page is reached.

DMKCKV, CKVWM6E

For *dump* SPOOL file slots, allocate the DASD pages used by each reconstructed SPOOL file by beginning with the first page and sequentially allocating each page until the last page is reached. (Dump SPOOL file pages are allocated as contiguous space and are not SPLINKed together as are standard SPOOL file pages.)

DMKCKV, CKVM9

Mark the slot following the last checkpoint slot as the physical end of the checkpoint slots. Set the next SPOOL file ID to one more than the largest ID that was restored. Return to DMKWRM.

DMKWRM, WARMCLR

Zero the first 8 bytes of the first record on the warm start area. Save the starting time for this system (STARTIME) in the first record and in DMKRSPCV. DMKCKP will compare these two values to ensure that the correct volume is mounted before checkpointing the system.

DMKCKRSY

Rebuild SYSSPOOL's virtual storage. Chain the SFBLOKs off the printer or punch SPOOL file chains as appropriate. Chain the reader SFBLOKs in SYSSPOOL's virtual storage.

The following explains the difference between a CKPT start and a FORCE start:

DMKCKV, MSG915E

Display "DMKCKV915E PERMANENT I/O ERROR ON CHECKPOINT AREA" when an error occurs reading or writing a page from the checkpoint area. For a CKPT start, load a disabled wait state PSW (code 00E). For a FORCE start, if the error occurred while reading or writing the first page of the checkpoint area, then load a disabled wait state PSW (code 00E). For a FORCE start, if the error occurred while reading or writing any record other than the first page, then go to CKVWM1 to get the next checkpoint record. Display "DMKCKV944E ERROR ALLOCATING <CON|RDR|PRT|PUN> FILE user userid spoolid nnnn savedate savetime" when an error occurs allocating one of the DASD pages making up a SPOOL file. For a CKPT start, load a disabled wait state PSW (code 00E). For a FORCE start, delete the corresponding SPOOL file, and continue with the next checkpoint slot.

COLD Start Processing

DMKWRMST
Entry point to handle WARM, CKPT, FORCE, or COLD starts. DMKWRMST is called by DMKCPJ during system initialization. For a COLD start, R2=01. Call DMKCKTMP.

DMKCKTMP
Get storage to create and initialize spool file bit map.

DMKWRM
Call DMKWRNWM.

DMKWRNWM
Entry point to initialize the checkpoint area.

DMKWRM, WARMCLR
Zero the first 8 bytes of the first record on the warm start area. Save the starting time for this system (STARTIME) in the first record and in DMKRSPCV. DMKCKP will compare these two values to ensure that the correct volume is mounted before checkpointing the system.

System Shutdown

The system can be shutdown by using the SHUTDOWN command.

DMKCPSSH
Entry point to handle SHUTDOWN command.

DMKCPS, GOTMP
For AP/MP systems, halt work on the other processor.

DMKCPS, SIGPSTOP
If in single processor mode, then issue a SIGP STOP to the other processor.

DMKCPS, NOSPM
If monitor is active, then call DMKMNISH to stop it. Go to DMKDSPCH to wait for the necessary I/O to complete.

DMKCPS, CLOSECON
Call DMKVSPCO to close the caller's open console SPOOL file and the operator's open console SPOOL file.

DMKCPS, DASDCH
Locate and record statistical data for DASD, tapes and 3800 printers.

DMKCPS, NOSPM2
If SHUTDOWN without REIPL, then set CPID to "CPCP" to indicate a system shutdown. If SHUTDOWN with REIPL, then set CPID to "WARM" to indicate a system shutdown and automatic warm start. Go to DMKDMPRS to load DMKCKP.

DMKDMPRS
Entry point to re-IPL the system.

DMKDMP, RESTART
Get IPL device address.

DMKDMP, SIOIPL
Issue IPL CCW to load the IPL sequence. The 24 byte IPL sequence containing the IPL PSW and a read CCW is loaded from the IPL device. The read CCW reads DMKCKP from the IPL device into location X'1000'.

DMKDMP, B1
Load the IPL PSW read in by the IPL CCW. The IPL PSW has an instruction address pointing to DMKCKPT. DMKCKPT is the entry point to save the system warm start data. See Saving System Warm Start Data for more details.

Saving System Warm Start Data

DMKCKPT saves the system warm start data whenever the CPID is "CPCP" or "WARM." If the CPID is "CPCP," then DMKCKPT was invoked by the SHUTDOWN command (without the REIPL option) or following a dump to a tape or printer. (Or the system was STOPPED and then IPLed without clearing main storage.) In this case, a disabled wait state PSW (code 008) is loaded after the system warm start data is saved. If CPID is "WARM," then DMKCKPT was invoked following a dump to DASD or following the SHUTDOWN command (with the REIPL option). In this case the system will be reloaded and reinitialized after the system warm start data is saved.

DMKCKPT
Entry point to save the system warm start data and reload the system if indicated.

DMKCKP
Load the rest of the checkpoint program (DMKCKD, DMKCKF, DMKCKH, DMKCKM, DMKERP, DMKCKW, and DMKCKN).

DMKCKP, GETLIST
Copy the list of pointers from DMKRSP into DMKCKF. This list contains addresses needed by the checkpoint program. The beginning of the list is pointed to by the field ARSPPR in the PSA. This list must be copied because if the nucleus is rebuilt and then the system is shutdown, DMKCKP would be using addresses resolved by the loader for the *new* nucleus to checkpoint the *old* system. These addresses would not necessarily match.

DMKCKP
Call DMKCKDEV.

DMKCKDEV
Entry point to Halt I/O during system shutdown.

DMKCKD
Drain pending I/O interrupts.

DMKCKD, TRYSGICR
For MP systems, halt prefixing on the other processor, and drain pending I/O interrupts on the other processor.

DMKCKD, GETCON
Find a usable path to a console.

DMKCKD, CHINDEX
Get the RCHBLOK, RCUBLOK and RDEVBLOK for the first/next device defined in DMKRIO.

DMKCKD, NETWORK
For 37xx, if the system will be automatically restarting, then save the NCP name, real device address, number of NICBLOKs, and a bit map defining which terminals are enabled in the warm start terminal buffer.

DMKCKD, TEST3851
For 3851s, issue a suspend CCW.

DMKCKD, TESTTERM
For enabled terminals and graphics devices, save the device address in the warm start terminal buffer, and issue a HIO to the device.

DMKCKD, NONTERM
For non-terminal devices, issue a HIO to the device.

DMKCKD, NEXTDEV
Go to CHINDEX to handle the next system generated device.

DMKCKP, CALLERP
Call DMKERP to save MCH, CCH, OBR, and MDR errors on error recording cylinders.

DMKCKP, CALLCKF
Call DMKCKFIL.

DMKCKFIL
Entry point to checkpoint a warm system.

DMKCKF
Call DMKCKHST to validate the first record of the warm start area. Compare the clock value saved on the warm start area with the clock value saved in DMKRSPCV. DMKWRM saved the same value in both places when this system was IPLed. This ensures that the system is shutting down with the same volume it IPLed.

DMKCKF, OPLOGOFF
If CPID = "WARM," then the system will be auto restarting. Save the status of the operator (logged on/logged off) in the first record of the warm start area.

DMKCKF, TERMPUT
If CPID = "WARM," then the system will be auto restarting. Write the warm start terminal buffer containing the addresses of the enabled terminals out to the warm start area.

DMKCKF, DLM1
Write the first delimiter record.

DMKCKF, NXTLOOP
Create and save device and user accounting cards for active virtual machines.

DMKCKF, ACTCHAIN
Save the accounting records.

DMKCKF, DLM2
Write the second delimiter record.

DMKCKF, LOGNXT
Save the log message.

DMKCKF, DLM3
Write the third delimiter record. The third delimiter record contains the date, time, and day of the system log message.

DMKCKF, SAVESF
Save closed printer SPOOL file SFBLOKs.

DMKCKF, DLM4
Write the fourth delimiter record.

DMKCKF, RSPPU
Save closed punch SPOOL file SFBLOKs.

DMKCKF, DLM5
Write the fifth delimiter record.

DMKCKF, RSPRD
Save open reader SPOOL file SFBLOKs.

DMKCKF, CKPCPTRP
Save the last record of open CPTRAP SPOOL files.

DMKCKF, CKPMONIT
Save the last record of open monitor SPOOL files.

DMKCKF, DLM6
Write the sixth delimiter record.

DMKCKHPR
Save spool-to-tape SFBLOKs; monitor SFBLOK copy.

DMKCKF, SAVEDEL
Save SPOOL file SFBLOKs which are on the delete queue. These SPOOL files will be deleted after the system warm starts. They are saved in order to keep the saved allocation records valid.

DMKCKF, DLM7
Write the seventh delimiter record. The seventh delimiter record contains the SPOOL file ID count.

DMKCKF, OWNLP
Save the allocation records.

DMKCKF, DLM8
Write the eighth delimiter record.

DMKCKF, NEXTSHQ
Save the SHQBLOKs.

DMKCKF, DLM9
Write the ninth delimiter record.

DMKCKWHT
Save Reader Hash Table plus extension pages.

DMKCKWSP
Save SYSSPOOL's virtual storage.

DMKCKF
Call DMKCKHWM to write a valid first record out to the warm start area. The first record contains the STARTIME value from DMKRSPCV and the version "VM/SP."

DMKCKP
Call DMKCKMSV.

DMKCKMSV
Entry point to save the virtual machines indicated in the system name table.

DMKCKM, VMSAVEON
If there was an active V=R user and the dump was directed to DASD (CPID = "WARM"), then restore pages 1 through 4.

DMKCKM, ASGO
Call DMKCKNRD to fix up the RDEVBLOKs of the CP owned DASD so that DMKCKNIO can use them. Read in the system name table (DMKSNT).

DMKCKM, ASLOOP1
Call DMKCKNIO to save the pages specified for each virtual machine. DMKSNT specifies the name of the saved system, the volume serial number of the CP owned DASD that this system is to be saved on, the DASD cylinder and page to begin saving this system on, the userid of the virtual machine that is to be saved, and the pages of the virtual machine that are to be saved.

DMKCKM, ASTOD
Save the date, the time, the name under which this system is being saved, the userid of this virtual machine, the VMPSW, the general, floating point, and extended control registers, and the storage keys for the saved pages.

DMKCKP, AUTOWARM
Move "WARM" into CPID to indicate that the system is performing an automatic re-IPL.

DMKCKP, WARM
Load both pages of DMKSAV into the storage location assigned by the loader during system generation. Go to DMKSAVRS.

DMKSAVRS
Load the nucleus up to DMKSAV. For 3081 processors, issue a Test Block to each 4K block of main storage before loading that page from the nucleus area. Go to DMKCPINT to initialize the system. See CP Initialization for details.

Dumping the System and Automatic Re-IPL

DMKDMPDK
Entry point to write a system storage dump to the dump device. Entry occurs via ABENDxxx condition or by system restart. DMKDMP saves PSA values and determines if the dump is full of just CP portion (or SYSSPOOL's virtual storage).

DMKDMP, DMPMSG
Format and issue abend message to operator.

DMKVFRSV
Save all virtual machine's vector register save areas.

DMKDMP, DMPDASD
For dump to DASD, write out CP storage or all storage to selected DASD device.

DMKDMP, DSKEND
For dump to DASD, place the sending record number and the system file number in the dump file SFBLOK.

DMKDMP, RECSRCH
For dump to DASD, chain dump file RECBLOKs to RDEVBLOK, and link dump file SFBLOK onto the system reader chain.

DMKDMP, CKSEND
For dump to DASD, set CPID to "WARM" so that DMKCKP will automatically re-IPL the system. Go to RESTART.

DMKDMQ, DMPTAPE
For dump to tape, dump CP storage or all storage to the selected tape drive per specified tape parameters. Leave CPID equal to "CPCP." Go to RESTART.

DMKDMQ, DMPPRT
For dump to a printer, dump CP storage or all storage to the selected printer. Leave CPID equal to "CPCP."

DMKDMP, RESTART
Get IPL device address.

DMKDMP, SIOIPL
Issue IPL CCW to load the IPL sequence. The 24 byte IPL sequence containing the IPL PSW and a read CCW is loaded from the IPL device. The read CCW reads DMKCKP from the IPL device into location X'1000'.

DMKDMP, B1
Load the IPL PSW read in by the IPL CCW. The IPL PSW has an instruction address pointing to DMKCKPT. DMKCKPT is the entry point to save the system warm start data. See Saving System Warm Start Data for more details.

Dynamic Checkpoint of Spool Files and Spool Devices

DMKCKSPL
Entry from any routine that adds, deletes, or changes the status of closed spool files. Lock the routine, or wait until it becomes unlocked. Bring the map page and spool fileid bit map page into storage and set up the device code of the system residence volume.

DMKCKS, LOOPSHQ
If the change is applicable to a SHQBLOK (hold queue block), make appropriate change on the checkpoint cylinder.

DMKCKS, CKSPL1
If the change is applicable to a SFBLOK, either add, change, or delete it on the checkpoint cylinder.

DMKCKS, CKSPL5
If the change affects a spooling device RDEVBLOK (for example, a START or DRAIN command issued), mark the change on the checkpoint cylinder.

Dump a Virtual Machine with VMDUMP Command

DMKVMDEP
Entry occurs via VMDUMP command. The command options are verified.

DMKPGUVG
Used to get virtual pages for use as spool files.

DMKRPAPT
Used to write out spool records.

DMKPGTSG
Used to obtain temporary space CCPD.

DMKPTRAN
Used to fix and lock storage pages.

DMKPTRUL
Used to unlock main storage pages.

DMKPGUVR
Used to release main storage pages.

IPL the Virtual Machine

DMKCFGIP
For the IPL of a named saved system, the name is verified and resources are checked for availability. Virtual storage is set up with the saved system via SWAPTABLE, SEGTABLE, SHRTABLE updates. For the IPL of device address, the IPL simulator is loaded in the user's storage. Before the IPL simulator is loaded into the user's storage, the contents of that storage is preserved and the area is then used for the IPL simulator.

DMKVMIPL
Read in 24 bytes from the CTCA, reader, DASD or tape unit into the user's virtual location zero. The CCW pointer is now set to the IPLCCW at virtual location X'8' and the IPL CCW string is executed.

DMKVMI, IPLDONE
Control comes here upon completion of the IPL CCW string. For IPL STOP, the virtual machine is placed in console function mode to allow change to nucleus name and apparent storage size before continuation.

DMKVMI, LOADNOW
IPL address is inserted in X'02' if BC mode, or X'BA', if EC mode. The user's CAW and registers are restored and control is given to the user by invoking diagnose X'10' to restore the page which the IPL simulator overlaid and load the current PSW at virtual location zero.

Virtual Machine Termination

DMKUSOLG
Entry is the result of user invoking LOGOFF. Set flags in VMBLOK indicating logout operation.

DMKUSOFL
Entry is the result of class A user issuing a FORCE command. Set flags in the forced user's VMBLOK indicating logout operation.

DMKUSO, USO06
Retain line communication, if HOLD operand specified.

DMKUSO, USO08
Adjust return address to not run the user.

DMKUSOFF
Set VMBLOK flags. In the event of abnormal termination, save a virtual machine if it is enabled for VMSAVE.

DMKUSQFF
Continues logout processing. Called from DMKUSOFF.

DMKTRCND
Called to reset tracing.

DMKPENDA
Called to reset tracing.

DMKACOTM
Accounting called to compute the connect time for the LOGOFF message.

DMKQCNWT
Write the message to the user.

DMKSCHDL
Called to alter user dispatch status.

DMKCFPRR, DMKCSPO
Reset the virtual machine.

DMKMHCRE
Release HCBLOK for pending MSSF request.

DMKVMCAN
Release or return VMCBLOKs if VMCF is active.

DMKVATBC
Release shadow tables (if any).

DMKSCHRT
Dequeue clock comparator request (if any).

DMKBLDRL
Release segment tables, page and swap tables related to the user.

DMKUSO, USO94
Via DMKFRET return user VMBLOKs to free storage.

DMKUSO, USO93
For the system operator, clear and reinitialize the VMBLOK.

DMKFRET
Return all other virtual machine control blocks to free storage.

DMKACOFF
Punch an accounting card for the user.

DMKUSO, USO98
Free LOGOFF message area. Exit to do free storage maintenance. Exit to DMKCFM or DMKDSPCH.

DMKUSOFL
Entry is the result of the invoked FORCE command.

DMKSCNAU
Locate userid VMBLOK.

DMKUSOFL
Set VMKILL in VMBLOK, build CPEXBLOK and stack it for dispatcher.

DMKDSPCH
Upon CPEXBLOK execution, process as at LOGOFF entry DMKUSOFF.

DMKUSODS
Entry from an invoked CP DISCONN command. Set disconnected VMDISC in VMOSTAT.

DMKQCNWT
Send disconnect message to user.

DMKUSODS
Increment return address to DMKCFM by 4 to prevent a return read to the user's terminal. Clear VMTERM field to indicate the user terminal is disconnected.

DMKQCNWT
Send message to system operator informing him of user disconnect status. Exit to DMKCFM.

Console Function (CP Command) Processing

DMKCFMBK
Entry used when the ATTENTION key (or equivalent) is pressed once or twice (according to the VM or CP status) to allow the user to direct a line of input data for CP command processing. Set VMFCWAIT and VMCF bits in VMBLOK indicating wait state and console function mode.

DMKFREE
Builds an 18-doubleword CONBUF buffer for the read operation.

DMKQCORD
Read in the terminal input command line.

DMKSCNFD
Find the START and length of the command.

DMKCFCMD
Scans the CP command line, determines whether a requested CP console function is allowed for a user, and obtains the entry point of the command processor to handle the console function invoked on the command line.

DMKCFC, CMDCLC
Processes command with no subcommands.

DMKCFC, NEXTCMD
Processes command with no subcommands.

See "CP Diagnostic Aids" for a list of all CP commands and the associated processing modules.

DMKQCORD
Read in the terminal input command line.

DMKCFMAT
On NULL data and ATTN key indication, post attention interrupt pending in VDEVBLOK, VCUBLOK and VCHBLOK. Return to run the virtual machine.

DMKCFMRQ
On receipt of CP commands ATTN or REQUEST, process the same as previous entry, DMKCFMAT.

DMKCFM
On receipt of * (asterisk) return to DMKCFMBK to set up another read. If console spooling is enabled, all console input and output including comments are spooled for printer output.

DMKCFMBE
On receipt of BEGIN, simulate the start button on the virtual machine (If optional address is supplied with BEGIN command the supplied address is substituted for the location counter address).

DMKCVTHB
Convert this address to binary notation.

DMKCFMSL
On receipt of the SLEEP command or SLEEP with time value (simulation of virtual machine stop button depression) the VMBLOK's VMSLEEP bit is set. The terminal console keyboard is now inactive until the user hits an ATTENTION key or the SLEEP command times out.

Dispatching and Scheduling

Fast Reflection for the Dispatched Virtual Machine

DMKDSPA
Entry for fast reflection activity. If the user is no longer runnable, or if the system is extending, the fast reflect path is not continued and processing continues at the main dispatcher entry point.

DMKDSP, UPVIRT
If the user is running virtual timers, update and test the user's virtual timers.

DMKDSPA1
If the user is still dispatchable, build the new RUNPSW from either IOOPSW or PROPSW and redispatch the virtual machine.

PSW Validation

DMKDSPB
Entry to dispatcher when the user's PSW has been external to DMKDSP.

DMKDSP, CKPSW
Verify the PSW change.

DMKDSP, CKPEND
Unstack any pending interrupts for the user (if enabled).

MAIN Dispatch Entry

DMKDSPCH
Normal dispatch entry after each interrupt handler has finished processing, and after each CPEXBLOK, I/O request and external interrupt has been serviced.

DMKDSP, RUNTIME
If CPSTATUS indicates return from running a user (CPRUN on), first ensure that supervisor time is being charged to RUNUSER. Check the user for time-slice end or queue-slice end, store the time remaining in the time-slice, and update processor problem state time. Also update virtual timers if running.

DMKDSP, WAITIME
If CPSTATUS indicates return from wait (CPWAIT on), first ensure that supervisor time is being charged to the system. Determine the type of wait (I/O wait, page wait, or idle wait) and save the appropriate new wait time value.

DMKDSP, UNSTACK
For nonrunnable virtual machine, go to label CHKILL in DMKDSP.

DMKDSP, UNSTACK
For runnable user, check pending interruptions for the following:

- **DMKDSP, CKPEND**
  Per interruption (VMPERPND)
  If user PER is active, then reflect the PER event to the virtual machine. If CP PER is active, call DMKPERIL to handle the PER event.
  Pseudo page faults (VMPGPND)
  External interruptions (VMPXINT)

- **DMKDSP, UNSTIO**
  I/O interruptions (VMIOINT)

- **DMKDSP, STORECSW**
  I/O interruptions are reflected by swapping user PSWs and storing the unit address and status in low storage.

- **DMKDSP, CLEARVMX**
  Clear the pending bits in the VMBLOK.

DMKDSP, CKPSW
Validate the PSW.

- **DMKVATBC**
  For virtual machine leaving EC mode, clean up the shadow tables.

- **DMKVATMD**
  For virtual machine in BC mode and entering translate mode, initialize shadow tables.

DMKDSP, DSPERMSG
For PSW invalid, send error message to virtual machine, and place user in CP mode. If disconnected and invalid PSW, log off user.

DMKDSP, DISPATCH
Complete processing for current user. Call DMKSCHDL if necessary to alter user's dispatching priority.

Selecting the Next Unit of Work

DMKDSP, CKCPSTAK
Process a stacked request. First check the stack of IOBLOKs and TRQBLOKs. If system is not extending, unstack normally. Otherwise, only unstack paging or PCI IOBLOKs.

DMKDSP, WINDOW
Before examining the stack of CPEXBLOKs, open a window for interrupts if the system is not extending.

DMKDSP, CKCPREQ

Check the stack of CPEXBLOKs. If the system is extending, only unstack those blocks that will allow the extend to complete. If the system is not extending, unstack normally. If a CPEXBLOK for the other processor is encountered, give up the system lock and signal the other processor.

DMKDSP, CKUSERS

If no stacked requests can be unstacked, select a user for dispatching. If the system is locked for running users (such as during extend), load a wait state. Scan the run list for a dispatchable candidate. If none is found, load a wait state. If there is also a runnable user for the other processor, signal the other processor. If a runnable user is found, set up to dispatch this user.

DMKDSPWA

Loads base registers, branches to do wait time accounting, then branches to dispatch the VMBLOK.

Scheduling the Next Unit of Work

DMKSCHDL

Main entry to maintain queues of runnable and eligible users and to alter the user's dispatching status and, at queue-drop time, calculate his projected working set size and deadline priority. Also updates VMQBLOK statistical data.

DMKSCH, ADDQ

Decides to which processor the VMBLOK will have affinity.

DMKSCH, CKRSTAT

If the user is now not runnable but was runnable before, mark the user as not runnable. If the user is in the eligible list, drop him from the list. If the user is in a long wait state, set a 300-millisecond delay.

DMKSCH, CKRUN

If the user is now runnable after being in a long idle wait, add him to Q1 (recalculating his deadline priority as necessary).

DMKSCH, CKWAITING

Add users to the run list by searching the eligible list. Users are added on the basis of class (interactive or noninteractive), projected working set (compared with available main storage), availability of processor time, and priority of users presently on the run list.

Queue Drop Processing

DMKSCH, DROPTIM
Entry when a virtual machine is dropped from queue. Tells the scheduler from what queue a virtual machine is dropped by setting on the bit VMSWQ1DR in the field VMSWSTAT when a virtual machine is dropped from Q1.

DMKSCH, DROPQ
Does virtual machine resource evaluation, and drops a virtual machine from queue.

DMKTRQIL
This is the interrupt return address for the 300 millisecond queue drop delay in the scheduler. Sets the appropriate bits and calls DMKSCHDL to drop the virtual machine from queue.

Establishing Addressability

DMKTRQ, LOCKDSQ
Establishes addressability to the proper PXA before obtaining the TRL lock.

Page and Swap Set Migration

DMKPGM, DMKPGMEP
Invoked by the dispatcher if it has a CPEXBLOK stacked for page/swap set migration. It migrates pages to TYPE=PP,PG, and PM storage levels, and swap sets to non-Paging Storage TYPE=SW and PP storage levels.

DMKPGM, DMKPGMUS
Invoked from DMKSWMUS (after command parsing) if page/swap set migration is invoked via the MIGRATE command. If command is valid, a CPEXBLOK is stacked with an entry point of DMKPGMX to migrate all users, or an entry point of RESETUS to migrate one user.

DMKSWMIG
Migrate swap sets from Paging Storage TYPE=SW to next available TYPE=SW level.

DMKSWMUS
Parse MIGRATE command. If the command is valid, stack a CPEXBLOK for migration.

Swap Table Migration

DMKSTR, DMKSTRSM
Invoked if the dispatcher has a CPEXBLOK for swap table migration.

DMKSTR, DMKSTRAN
Invoked from DMKPTR if there is a segment exception.

System Performance Indicators

DMKSTP, DMKSTPX
At initialization time, control the generation of four system control constants. These control constants are then used in the calculation of the 28 system performance indicators, which in turn are used to produce six scheduling control fields. DMKSCH uses these control fields to calculate the deadline priority of the virtual machine at queue-drop time.

CP Timing Facilities

DMKSCHST
Set a clock comparator interrupt request.

DMKSCHRT
Reset a clock comparator interrupt request.

DMKTRQMD
Set up a request block for midnight date change.

DMKTRQ80
Process a real interrupt timer request.

DMKTRQCP
Process a real CPU timer interrupt.

DMKTRQTI
Update system performance indicators.

Spooling Virtual Device to Real Device

Processing Virtual Output Files

DMKVSPEX
Entry from DMKVSI to initiate SIO on a spooling device that is available (not busy and no interruptions pending).

DMKVSTOP
If output device needs to be opened, call DMKSPLOV to build control blocks SFBLOK and VSPLCTL.

DMKFREE
Get a work buffer for the CCWs and the data.

DMKVSRGC
Get first CCW.

DMKPGUVG
Obtain a virtual buffer; the address is stored in VSPVAGE.

DMKPGTSG
Obtain a DASD page; the address is stored in VSPDPAGE.

DMKVSP, PRINTER
Verify CCW opcode and set up initial CSW status.

DMKVSRMD
Get user's data in the work buffer.

DMKVSP, COMPRESS
Truncate all right-justified blanks.

DMKVSQ
Attempt to compress printer opcodes. If spooling space is available, move the CCW and data from the work buffer to the spool buffer; else call DMKPGTSG to get a new spool buffer and write out the full spool buffer.

DMKVSP, FLAGTEST
If the channel program ends (either all CCWs are processed or there is an error), go to LASTDDW; else, call DMKVSRGC to get the next CCW and go to DMKVSP, PRINTER.

DMKVSTCP
On console spooling, the following occurs:

1. Skip to channel 1 every 60 lines.
2. Write out the system console, spool file buffer every 16 lines.
3. Place the system console in a pseudo closed state for checkpoint recovery in the event of system failure.

DMKVSP, LASTCCW
When all CCWs are processed, post interruption pending to the VDEVBLOK, VDEVCSW and return control to the user.

DMKVSPPE
Close spooled printer/punch file via call to subroutine PRTEOF.

DMKVSP, PRTEOF
Page in the last page buffer if not resident. If the spool file is empty, purge it; otherwise, update pointers and write last page buffer to DASD. If PURGE was specified, call DMKSPKDL to purge the file; else, close the file via DMKSPLCV and call DMKFRET to fret the VSPLCTL.

Closing Virtual Output Files

DMKVSUCO
Entry via CP CLOSE command.

DMKVSPST
If the device is busy, defer close operation by building a CPEXBLOK. Stack the CPEXBLOK and exit to the dispatcher.

DMKVSPPE
On device not busy, write final buffer page to DASD storage.

DMKSPLCV
Queue closed virtual printer or punch spool file to the real spool output device, or transfer the file to another user's virtual reader. Also update the SFBLOK with number of copies printed/punched, distribution code, hold status, and file owner ID. If VSPXBLOK with TAG data exists for the spool device, copy the TAG data to the TAG record in the first spool file data buffer.

DMKSPL, TXTXFR
If a "spooled to" file, queue to the end of the reader file chain. Otherwise, chain the SFBLOK to the designated real spool printer or punch.

DMKCKSPL
Checkpoint the new spool file block.

DMKSPL, SETPEND
For a "spooled to" file find a virtual reader with the proper class and in the ready state with no active file, and no pending interrupts. Then build an IOBLOK with IOBIRA of DMKVIOIN.

DMKSTKIO
Stack the IOBLOK.

DMKSPL, SETPEND
Exit to DMKVSP.

DMKSPL, TSTHOLD
For not "spooled to" files and not in user or system hold, find printer or punch with the proper class. Then build an IOBLOK with IOBIRA of DMKRSPEX.

DMKSTKIO
Stack the IOBLOK.

DMKSPL, TSTHOLD
Exit to DMKVSP.

Processing Virtual Input Files

DMKVSWOR
Entry to open a spool input file. If VDEVSPL=0 the file needs to be opened. Build VSPLCTL block and a work buffer.

DMKVSP, SETFLAG
On file-found condition, place first DASD page address in VSPLCTL, VSPDPAGE. Obtain a virtual buffer and retain its address in the VSPLCTL block.

DMKVSP, READER
Check the CCWs for validity, move and expand the data back to its original size and the data is moved from the work buffer to user's virtual storage.

DMKVSP, RDRCOUNT
On EOF, set SFBEOF bit in SFBLOK and return to caller.

Closing Virtual Input Files

DMKVSUCR
For CLOSE operation requested via console command and the device is busy, initiate a delayed close by constructing and stacking the CPEXBLOK for the CLOSE.

DMKVSP, RDREOF
For normal end of file and VDEVSFLG indicates continuous read.

DMKVSP, OPENCONT
Locate the next file and continue reading.

DMKVSP, LASTFILE
For last file, post end status in RDEVBLOK.

DMKVSP, FILECLR
For HOLD status file (VDEVSFLG=VDEVHOLD), call DMKCKSPL.

DMKVSP, FILECLR
Call DMKVSDDL and DMKSPKDL.

DMKVSDDL
Unchain the file (except hold files) from the reader queue. Call DMKCKSPL.

DMKCKSPL
Checkpoint the file.

DMKSPKDL
Delete the file.

DMKVSP, DVICECLR
To clear the device, call DMKRPAGT.

DMKRPAGT
Releases the storage page.

DMKPGUVR
Releases the virtual buffer.

DMKFRET
Releases storage for the work buffer and VSPLCTL block.

Spooling to the Real Printer/Punch Output Device

DMKRSPEX
Entry from the dispatcher when an IOBLOK is unstacked with and interrupted for spooling unit record device. IOBRADD points to the RDEVBLOK RDEVTYPC input or output class.

DMKRSP, RSPLOUT
If RDEVSPOL indicates an available spool device (not active),

DMKFREE
Get storage for a work buffer and build a RSPLCTL block and link it to RDEVBLOK.

DMKRSP, PRNXTFIL
Search printer and punch SFBLOK chains for corresponding device, form, and class. On a found condition, unchain the block, put its address in RSPSFBLK. The FLASH name specified in the SPOOL command, if FLASH is specified, must match the flash overlay name for a 3800 printer.

DMKSEPSP
If called, provides separators for output pages or cards.

DMKTCSET
If the device is a 3800 printer, call this module to set it up.

DMKTCTET
If required, load character arrangement tables and graphic character modifications.

DMKRSP, PROCESS1
Bring first spool data DASD page to the work buffer and convert CCW addresses to real device addresses.

DMKIOSQR
Start the spool device.

DMKRSP, PRNXTPAG
Repeat the process until done.

DMKRSP, REPEAT
Reprocess and reaccess the buffer, if multiple copies are specified.

DMKCKSPL
Checkpoint records the change to COPY count.

DMKSPKDL
Delete the file on completion (unless HOLD specified). If the device is a 3800 printer, check for delayed purge.

DMKRSP, PRNXTFIL
Locate the next spool file to process.

DMKRSP, PRTIDLE
Processing for the device is complete as there are no more SFBLOK, for this device or the device was drained.

DMKFRET
Release work area and completed IOBLOK storage.

DMKDSPCH
Exit to the dispatcher.

Spooling to the Real Input Device

DMKRSPEX
Entry from the dispatcher when an IOBLOK is unstacked with an interrupt for a spooling unit record device. IOBRADD points to the RDEVBLOK RDEVTYPC input or output class.

DMKRSTIN
Handles the card reader.

DMKSPLOR
Assume there is no active file being processed on the real input file reader. The spooling operator has issued the START command to the device to "open" the reader.

DMKSPL, BUILDCTL
Build RSPLCTL and SFBLOK.

DMKPGUVG
Get virtual buffer and place its address in RSPVPAGE.

DMKPGUSG
Get DASD buffer and place its address in SFBSTART and RSPDPAGE, linked together by pointers.

DMKIOSQR
Start the reader.

DMKDSPCH
Await the interruption.

DMKRST, RDERGETID
Check that the first card in the buffer is the userid header. If so, proceed.

DMKRST, RDRCARDS
Preload the buffer with CCWs.

DMKIOSQR
Issue the SIO (SIO's of 42 cards per buffer load).

DMKRST, RDRSIO
Write the buffer to the DASD slot. Repeat until EOF detected.

DMKSPLCR
Close the file on EOF. Queue the file on reader spool chains.

DMKCKSPL
Add the spool reader file block to the checkpoint cylinder data.

DMKSPL, RDRPEND
If the file owner is logged on, and his virtual reader is available, an IOBLOK is constructed with device end pending.

DMKSTKIO
Stacks it.

DMKRST, RDREXIT4
Release storage for virtual buffer, RSPLCTL and the SFBLOK.

DMKDSPCH
Exit to the dispatcher.

Spool File Deletion

DMKSPKDL
With R7 not equal to zero, place the specified SFBLOK on the delete chain anchored to DMKRSPDL.

DMKSPKDL
If the delete routine is not running, build a CPEXBLOK to call DMKSPLDR.

DMKSTKCP
Stack it and exit to caller.

DMKSPKDR
Set the DELSW=X'80' (delete routine active).

DMKSPKDR
On unstacking the CPEXBLOK, if the SFBLOK is a system dump file, call DMKDRDDD.

 LY20-0897-7

DMKCKSPL
Delete the SFBLOK from checkpoint cylinder data.

DMKDRDDD
Deallocate DASD buffers.

DMKSPK, NEXTSFB
For complete allocation chains of RECBLOKS, call DMKPGTSR.

DMKPGUSR
Deallocate DASD buffer and return to storage held by the dummy RECBLOKs.

DMKSPK, DELSTART
For incomplete allocation RECBLOK chains, deallocate by calling DMKPGTSD.

DMKPGUSD
Deallocate a page at a time via SFBSTART and the IOBLOK until the last page is reached. Call DMKCKTSD and DMKSVGRI. After each SPLINK is deallocated, see if any more files have been added to the delete queue. If so, rechain the current file and deallocate a new file.

DMKFRET
Delete the SFBLOK, then go to DMKSPK, NEXTSFB.

DMKSPK, NEXTSFB
If the delete queue is not empty, process the next SFBLOK in identical manner. Continue until all SFBLOK deletions are completed; then call DMKFRET.

DMKFRET
Delete the IOBLOK.

DMKDSPCH
Exit to the dispatcher.

Recovery Management Support Operation

Establishing the Error Recording Base

DMKIOEFL
Entry from CP initialization module to set up pointers to error recording cylinders.

DMKIOGF1
The STIDP instruction stores processor version and model in CPUID of PSA.

DMKIOG, ISSUEINS
Check attached channels. If stand-alone channel on the 165 or 168, the address of the logout routines is stored in the DMKCCH module.

DMKIOG, CHANGEID
Set up pointers for machine check and channel check record area and extended logout areas.

DMKIOG, IOGMCHIN
Obtain storage for machine check record, extended logout area, and CPEXBLOK. The MCHAREA is also initialized.

DMKIOG, PASTDAVE
Determine the 90% full and 100% full capacity of designated error recording cylinders and store the amount in DMKIOEMX and DMKIOENI respectively.

DMKIOG, FINDREC
Check first record of the error recording cylinders for proper format. If invalid, reformat. If valid but clear, store pointer value in PSA as the first available slot for error record. If valid but used, search for first unused slot and store its value in PSA.

DMKIOGFR
If on a 3031, 3032, or 3033 processor, read frames from the SRF (service record file) device, and write them to the beginning of the error recording cylinders with unique record types.

DMKIOG, CYLFULL
When error recording area is full, inform the operator, and continue.

DMKIOEFL
Turn off the recording in progress switch and exit to caller.

Machine Check Interrupt Processing

DMKMCHIN
Entry via the machine check PSW upon detection of an unrecoverable and nonfatal processor or storage error. Disable soft machine recording; store logout area on the error recording cylinders. The system is enabled for hard machine checks by pointing the PSW to the termination routine.

DMKMCH, ENHARD
For the virtual machine, store status in the VMBLOK.

DMKMCH, MCHSYSIL
For system damage, timing facility, uncorrectable retry, multibit storage error, post system operator message, and flag the system as terminated. If the fault occurred in problem state, terminate the active virtual machine.

DMKMCH, SOFTSTG
For corrected ECC or processor retry, update soft error count and record the error and dispatch the virtual machine.

DMKMCH, MCHSKIP
For multibit storage error in problem mode, exercise storage location to clear up or flag as unavailable (permanent error).

DMKMCH, MCHCHANG
On an altered page condition, the virtual machine is reset, otherwise, the error is recorded and the virtual machine is redispatched.

DMKMCH SPFTEST
Storage key failure. Exercise the 2K page key. If CP area and solid error condition process as DMKMCH, MCHSYSIL, intermittent, restore the key and go to the dispatcher. If key failure and in virtual machine area if permanent error, mark page as unavailable, terminate the user. If intermittent condition refresh the key and dispatch the virtual machine.

DMKMCH, VIRTERM
On conditions that cause the termination or reset. The error is recorded, and both the user and the operator receive status messages. Per the termination flag, VMBLOK, the user is logged off and control returns to the dispatcher or is reset via DMKCFPRR.

DMKMCH, CHANTERM
For channel inoperative errors. DMKACR is called to attempt to recover the failing channel or channels.

DMKCFPRR
Virtual storage is released, the virtual machine is flagged dispatchable and placed in console function mode.

DMKMCH, TERM
On a hard machine check while handling a machine check, the machine check new PSW is loaded with a wait state PSW and the current PSW is enabled for hard machine checks.

DMKMCH, MCHTERM2
Locate the system or the user's VMBLOK.

DMKMCH, OPCOM
Call DMKMCTPT if system is running in attached processor or multiprocessor mode.

DMKMCH, MCHWAIT
Load disabled wait state for uniprocessor system.

DMKMCTPT
Complete processor termination for attached processor or multiprocessor system. If the error is on the attached processor and it is in problem state, signal for automatic processor recovery and stop the attached processor. If the processor is executing in multiprocessor mode and the error occurred while the processor was in problem state, signal for automatic processor recovery and stop the failing processor.

DMKMCT, SWITCH
Make sure processing is on an I/O processor and set up the appropriate wait state code.

DMKMCT, OPCOM
Issue a message to the operator and load a disabled wait state for the attached processor or multiprocessor system.

DMKMCTPR
Perform automatic processor recovery function. Allow system to convert to uniprocessor mode by calling DMKCPPUP.

DMKMCT, PREXIT
Terminate the virtual machine if it is in control. Reset the main processor timer. Clear all lock words and return to the dispatcher.

Channel Check Interrupt Processing

DMKCCHIS
Entry via DMKIOS via CSW channel error.

DMKFREE
Obtain storage and build a CCHREC block and if IOBLOK and RDEVBLOK exist, build an IOERBLOK.

DMKCCH, CCHIOERL
Store the CCHREC address, its length, and the CSW in the IOERBLOK.

DMKCCH, CCHDEPND
Call appropriate channel error analysis module. Analyze channel logout data for validity.

DMKCCH, SCNEND
Record the error on the error recording cylinder, if appropriate.

DMKCCH, CPTERM
Terminate CP if the PSA's terminate flag is set.

DMKCCH, CHANTERM
Attempt recovery from an I/O interface inoperative or a reset channel condition.

DMKCCH, CCHWAIT
Set up X'0F' wait state code and call DMKMCHST to terminate the system.

DMKMCHST
If the system is running in attached processor or multiprocessor mode, call DMKMCTST.

DMKMCH, CALLOPR
Issue an error message to the operator.

DMKMCH, MCHWAIT
Load a disabled wait state for a uniprocessor system.

DMKMCTST
Make sure system is running on an I/O processor; load disabled wait state.

DMKMCT, CALLOPR
Issue an error message to the operator.

DMKMCT, MFAWAIT
Load a disabled wait state for attached processor or multiprocessor system.

DMKCCH, SCNEND
Unless termination is established, return to DMKIOS for recovery.

Recording the Errors of the Virtual User via SVC 76

DMKVERD
Entry via DMSPSA as a result of SVC 76 detection. Check parameters passed in R0 and R1.

DMKFREE
Obtain storage for a record buffer for the user error record.

DMKVER, BUFFUL
Using valid record type (from the buffer) branch to an appropriate routine to format that particular record type.

DMKVER, VER30
Using RDEVBLOK, VDEVBLOK and VMBLOK, convert virtual data to real values and place in record.

DMKIOERV
Record the error.

DMKDSPCH
Exit to dispatcher.

User Directory Routines

DMKUDRFU
Entry after CP detected LOGON command. DMKSYSPL points to the directory. Determine length of userid, if valid call DMKLOCKQ.

DMKLOCKQ
Lock the directory in storage.

DMKUDR, NXTPAGE
Bring in each directory page and return each page (and clear the buffer) until a UDIRBLOK match occurs or directory's last page is detected.

DMKUDR, FINDUSER
On userid found, move UDIRBLOK to caller's area.

DMKLOCKQ
Unlock the directory in storage.

DMKUDR, EXITCC0
Return to caller.

DMKUDRFD
Entry from calling routine to find the addressed (cuu) device UDEVBLOK in users directory and move it to the caller. Via UMACBLOK locate the UDEVBLOKs.

DMKUDR, FINDDEV
Check to see if the user device address is the same as in the UDEVBLOK. Search the chain until match or end of chain occurs.

DMKUDR, DEVFOUND
For found condition, post condition code zero in user's VMPSW.

DMKUDRRD
Entry from calling routine to read the UDEVBLOK addressed into the caller's buffer. Using the DASD and the user displacement from the UMACBLOK, bring in the buffer page to storage. Determine if the virtual directory page address (UDBFVADD) exists in the user directory buffer blocks. If not call-

DMKPGUVG
and get a virtual page.

DMKRPAGT
For DASD address does not match the UMACBLOK, point to the DASD page and bring in the virtual buffer page. Move UDEVBLOK into callers area and set cc=0 in VMPSW. Return to caller.

DMKUDRRV
Entry to return a virtual page used as a buffer. Determine if UDBFBLOK contains a virtual buffer page pointer (UDBFVADD). If not, exit with cc=1 set in the VMPSW. If a buffer exists, check to see if it is resident; if it is, clear it to zeros.

DMKRPAGT
Return the real page to the system.

DMKRGUVR
Return the virtual page to the system.

DMKUDRRV
Set cc=0 and return to caller.

DMKUDRBV
Entry from DMKDIRCT or DMKCPINT to build page buffers for each UDIRBLOK.

DMKFREE
Get storage for the virtual buffer page list.

DMKUDR, GETVPAGE
Call DMKPGTVG and DMKRPAGT to get the virtual and real buffer. Save the virtual buffer address in the page list.

DMKUDR, FRETLIST
Encountered I/O error, free the virtual buffer page list, post fatal message, set cc=3 and return to caller.

DMKUDR, ENDLIST
Swap the new virtual buffer page list with the old list. Anchor the new list to DMKSYSPL.

DMKUDR, FRETLIST
If there was a previous buffer page list, free it. Save the start of the user directory pointer in DMKSYSUD, and return to caller with a cc=0 in the VMPSW.

Save the 3704/3705 Control Program Image Process

DMKSNCP
Entry from DMKHVC and DIAGNOSE code 50. Per the system VMBLOK, locate the DMKRNTBL. The CCPARM virtual address is contained in R1 of the DIAGNOSE instruction.

DMKSNC, NAMECHK
Match via search CCPARM; CCPNAME with DMKRNTBL entries.

DMKSNC, SIZECHK
Verify DASD space requirements for 3704/3705 control program and resource data. The volume required to save (NCPVOL) as indicated in the NCPTBL entry must be available and mounted on the system, on a CP owned and supported paging device.

DMKSNC, SVRESDAT
Save resource data on the NCPVOL device. CCPARM supplies the starting address and size parameters for this write operation.

DMKSNC, SVNCPIM
Save 3704/3705 control program image on NCPVOL device. CCPARM also provides the parameters for this similar operation.

DMKSNC, SAVEFINI
Store cc=0 on no errors and return to caller.

Spool File Checkpoint and Recovery

Dynamic Checkpoint of Spool Files and Spool Devices

DMKCKSPL
Entry from any routine that adds, deletes, changes, the status of closed spool files. Lock the routine, or wait until it becomes unlocked.

DMKCKS, LOOPSHQ
If the change is applicable to a SHQBLOK (hold queue block), make appropriate change on the checkpoint cylinder.

DMKCKS, CKSPL1
If the change is applicable to a SFBLOK, either add, change, or delete it on the checkpoint cylinder.

DMKCKS, CKSPL5
If the change affects a spooling device RDEVBLOK (for example, a START or DRAIN command issued), mark the change on the checkpoint cylinder.

Reconstruction of Checkpointed Closed Spool Files

DMKCKVWM
Entry via DMKWRM during initialization when CKPT or FORCE is specified.

DMKCKV, CKVWM2B
For slots having real device entries, set or reset the RDEVDISA and RDEVDRAN and move in the checkpointed device classes into RVDEVCLAS.

DMKCKV, CKVWM2G
For slots containing spool hold queue block, chain this to the SHQ chain.

DMKCKR, BUILDSFB
Get storage for SFBLOK space and set flags depending upon its last checkpoint activity.

DMKCKR, CKVWM3C
Chain the reader SFBLOK to the appropriate individual reader chain.

DMKCKR, PRTORPUN
Chain the print or punch SFBLOK to the appropriate printer or punch chain.

DMKCKV, TASKCODE
Allocate the DASD buffers of the spool file by reading each buffer to determine the next one and then allocate this page.

DMKCKV, CKVWM6E

For the dump spool file, the buffers are allocated sequentially from the beginning to the end.

DMKCKV, FINISHED

Return to DMKWRM.

Inter-Virtual Machine Communication

DMKVMCFC

Entry from DMKHVC and the DIAGNOSE instruction code X'68'. Builds a VMCBLOK and initializes it with data from the user's parameter list, VMCPARM. The virtual address of VMCPARM is contained in bits 8-11 (rx) of the DIAGNOSE instruction.

DMKVMC, VMCFTBL

Branch table to pass control to the appropriate subroutine based on the subfunction code in VMCPARM.

Subfunction Code	Subroutines
X'0000'	VMCAUTH
X'0001'	VMCUAUTH
X'0002'	VMCSEND
X'0003'	VMCSENDR
X'0004'	VMCSENDX
X'0005'	VMCRECV
X'0006'	VMCCNCL
X'0007'	VMCREPLY
X'0008'	VMCQIES
X'0009'	VMCRESUM
X'000A'	VMCIDENT

DMKVMC, VMCWAKUP

Notifies a virtual machine of a pending VMCF communication by posting a special external interrupt X'4001' unless:

- There is already a special external interrupt posted.
- The virtual machine is running disabled for VMCF interrupts (PSW bit 7 and CR0 bit 31).

DMKVMC, VMCXFER

Transfers data from one virtual storage to another virtual storage. Errors occurring during data transfer are reflected to originating virtual machine via the data transfer return code in the final response interrupt message header.

DMKVMCEX

Called from DMKDSP to reflect an external interrupt message header to a virtual machine. If the VMCF subfunction is a SENDX, the SOURCE data is moved into the external interrupt buffer immediately following the message header.

DMKVMCUA
Called by DMKCFP when a virtual machine is logged off or reset. Uses the VMCUAUTH subroutine (subfunction code X'0001') to dispose of existing VMCBLOKS before turning off virtual machine communication.

Inter-User Communications Vehicle

DMKIUAEP
Entry is a GOTO from DMKPRV when a B2F0 instruction is encountered in a virtual machine. An IUSAVE block is built to contain the information needed to process the request.

DMKIUACP
Entry is via a CALL from a CP module requesting an IUCV function on behalf of the system. An IUSAVE block is built to process the request.

DMKIUACU
Entry is via a CALL from a CP module requesting an IUCV function on behalf of a virtual machine. An IUSAVE block is built to process the request.

DMKIUAPD
The path description entry is located in the caller's path description segment.

DMKIUAPL
Validates a communication path for a system path. CP paths are enqueued or dequeued as requested.

DMKIUAQU
Entry to queue a MSGBLOK on an IUCV message queue. The block is queued in priority, then non-priority FIFO order. If the purge bit is on in the MSGBLOK, the MSGBLOK is not queued and DMKFRET is called to release the MSGBLOK space.

DMKIUA, CKIDLE
Resets the IUCV wait conditions to allow an IUCV external interrupt to be reflected.

DMKIUA, BTABLE
Branch table used to determine the module and subroutine to handle the IUCV request.

DMKIUBRK
Entry is via DMKDSP to determine if there are any IUCV interrupts to be reflected to the virtual machine. The interrupts are reflected in the following sequence: IUCV control interrupts, priority replies, non-priority replies, priority messages, and then non-priority messages.

 LY20-0897-7

DMKIUBTB
Table containing the CP system service entry points for connects, messages, severs, quiesces, and resumes.

DMKIUCEP
General entry for this IUCV module.

DMKIUC, ACCEPT
Handle IUCV request to complete a communications path.

DMKIUC, CONNECT
Handle IUCV request to establish a communications path.

DMKIUEEP
General entry point for this IUCV module.

DMKIUERC
Perform the copying of data from the source to the target virtual machine for APPC/VM RECEIVEs.

DMKIUE, RECEIVE
Handle IUCV request to receive the message and cause the actual data transfer. If the message did not require a response, the MSGBLOK is moved to the source's REPLY queue; otherwise, the MSGBLOK is moved to the target's RECEIVE queue.

DMKIUGEP
General entry point for this IUCV module.

DMKIUGGP
This routine is used to locate an available path description entry to be used in establishing a connection.

DMKIUG, PURGE
Handle the IUCV request of the source to cancel a message. The MSGBLOK is located and marked as purged.

DMKIUG, REJECT
Handle the IUCV request of the target to refuse a message. The MSGBLOK is moved to the source's REPLY queue and marked as rejected.

DMKIUJEP
General entry point for this IUCV module.

DMKIUJ, SEVER
Handle the IUCV request to sever an IUCV or APPC/VM path.

DMKIULEP
General entry point for the IUCV module.

DMKIULRP
Perform the copying of data from the source virtual machine to the target virtual machine for APPC/VM SENDS when there is a predefined receive or answer area defined by the target.

DMKIUL, REPLY
Handle the IUCV request to answer a message and cause the actual data transfer of the reply. The MSGBLOK is moved to the source's REPLY queue.

DMKIUL, SEVER
Handle IUCV request to halt communications over a particular path.

DMKIUL, TESTCMPL
Handle IUCV request to test completion of a previously sent message. If the message has completed, the MSGBLOK is moved to the source's REPLY queue.

DMKIUNEP
General entry point for this IUCV module.

DMKIUNIN
Stack an external interrupt of type X'4000' to indicate an IUCV event.

DMKIUN, DESCRIBE
Handle IUCV request to describe a message in the invoker's SEND queue.

DMKIUN, SEND
Handle IUCV request to send a message across a communication path. A MSGBLOK is built and queued on the target's SEND queue.

DMKIUN, SETCMASK
Handle the five types of IUCV control interrupts. These control interrupts are Connection Pending, Connection Complete, Path Severed, Path Quiesced, and Path Resumed. Only those bits set are enabled.

DMKIUN, SETMASK
Handle the IUCV request to change the external interrupt submask. Only those bits set to 1 are enabled, and only those type IUCV interrupts are reflected.

DMKIUN, TESTMSG
Handle IUCV request to test a message. If messages are queued, the condition code is set to 1. If replies are queued, the condition code is set to 2. If both are queued, the condition code is set to 3.

DMKIUPEP
General entry point for this IUCV module.

DMKIUP, DCLBFR
Handle IUCV request to declare a buffer to contain the external interrupt data.

DMKIUP, QUERY
Handle IUCV request to determine the length of an IUCV external interrupt buffer and the maximum number of connections allowed for this virtual machine.

DMKIUP, QUIESCE
Handle IUCV request to prevent incoming messages on the established communication path.

DMKIUP, RESUME
Handle IUCV request to allow incoming messages on a previously quiesced communication path.

DMKIUP, RTRVBFR
Handle IUCV request to retrieve a buffer and terminate IUCV communications.

DMKIUSEP
General entry for this IUCV module.

DMKIUSET
To set the state of an APPC/VM path when a function completes.

DMKIUS, RECEIVE
Handle APPC/VM request to receive a message or signal and to cause the actual data transfer.

DMKIUS, SEND
Handle APPC/VM request to send a message or signal across a communication path.

Console Communication Services

The Console Communications Services (CCS) function of CP links with the VTAM Communications Network Application (VCNA) to provide support for SNA terminals. The following processing paths are the major paths used by CCS.

Unsolicited Interrupt

DMKVCPIL
Entered via call from IUCV (DMKIUE). Register 1 contains IUCV external interrupt buffer. For this path DMKVCPIL is the controlling module that calls all other modules.

DMKIUACP
Does IUCV receive.

DMKVCVKS
Locates SNA control blocks.

DMKVCPIL
Decodes function code (WEBFUN) to determine operation (Attention interrupt with no data).

DMKCFMAT
Posts an attention to the virtual machine.

DMKVCPIL
Stacks a CPEXBLOK to dispatch the virtual machine.

DMKVCRNR
If a batched write is pending, send the write.

DMKVCPIL
Exits to caller.

If data was received with Attention interrupt, DMKVCPIL determines the environment.

DMKFREE
If VM environment, obtains a buffer.

DMKVCPIL
Moves the WEBLOK's WEBDATA field to the buffer.

DMKFREE
Obtains space for a dummy CONTASK for editing.

DMKSCNED
Edits the data.

DMKFRET
Releases storage for CONTASK.

DMKVCPIL
Determines if #CP is in the data.

If #CP not in data, DMKVCPIL chains the buffer address off the virtual machine's VCONRBUF.

DMKCFMAT
Posts attention to virtual machine.

DMKVCPIL
Stacks a CPEXBLOK to dispatch the virtual machine.

DMKVCVEB
Builds a WEBLOK.

DMKVCVLY
Issues an IUCV reply to indicate input data accepted.

DMKFRET
Releases the buffer storage.

DMKVCVNR
If a pending batched write exists, sends the write.

DMKVCPIL
Exits to caller.

If #CP was in data, DMKVCPIL calls:

DMKCFMEN
Passes input to CP.

DMKVCVEB
Builds a WEBLOK.

DMKVCVLY
Issues an IUCV reply to indicate input data accepted.

DMKFRET
Releases the buffer storage.

DMKVCVNR
If a pending batched write exists, sends the write.

DMKVCPIL
Exits to caller.

If not VM environment, DMKVCPIL calls:

DMKFREE
Obtains a buffer.

DMKVCPIL
Moves the information from WEBLOK's WEBDATA field to the buffer.

DMKFREE
Obtains space for a dummy CONTASK for editing.

DMKSCNED
Edits the data.

DMKFRET
Releases the CONTASK storage.

DMKCFMEN
Passes input to CP.

DMKVCVEB
Builds a WEBLOK.

DMKVCVLY
Issues an IUCV reply to indicate input data accepted.

DMKCFRET
Release the buffer storage.

DMKVCVNR
If a pending batched write exists, sends the write.

DMKVCPIL
Exits to caller.

Initiate a READ

DMKVCRRD
Called via SVC from DMKQCORD. For this path DMKVCRRD is the controlling module and calls all other modules.

DMKSCHRT
If a timer has been set for the user, resets it.

DMKVCVIX
If a batched write exists, sends the write.

DMKVCVCE
Writes a Trace Element.

DMKFREE
Gets storage for a WEIBLOK.

DMKVCVEB
Builds a WEBLOK.

DMVCVIX
Does a two-way send of the Read to the virtual service machine.

DMKVCVCE
Writes a Trace Element.

DMKVCRRD
Exits to Dispatcher.

Return from a READ

DMKVCRMT
Invoked from IUCV (DMKIUE). At entry all registers are set up by IUCV. For this path DMKVCRMT is the controlling module and calls all other modules.

DMKVCVCE
Writes a Trace Element.

DMKVCRMT
Decodes the WEBLOK for the function, mode, and logical aid (Enter key). Moves the data from WEBLOK's WEBDATA field to a buffer obtained via a call to DMKFREE. Edits the data for line end characters and tab characters. Moves the data to the user buffer pointed to by the CONTASK.

DMKSCNED
Edits as directed by the CONPARM.

DMKVCRMT
Stores a return address in the save area pointed to by CONRETN.

DMKVCVER
Sets a redisplay timer for the virtual service machine.

DMKVCVIN
Frees the WEIBLOK, IXBLOK, and CONTASK built when the Read was initiated.

DMKVCVND
If any Writes are chained to the SNARBLOK, sends then to the VSM.

DMKVCVND
If any Reads are chained to the SNARBLOK, sends them.

DMKVCRMT
Exits to Dispatcher via GOTO.

Initiate a WRITE

DMKVCRWT
Entered via branch from DMKQCN. For this path DMKVCRWT is the controlling module and calls all other modules. Determines type of Write by examining the CONTASK: If Full Screen Write, or Diagnose Write, and if a batched Write is pending, calls:

DMKVCVIX
Sends the batch write to the virtual service machine.

DMKVCVLD
Obtains a WEIBLOK.

DMKVCVEB
Obtains a WEBLOK.

DMKVCRWT
Moves the output line from the CONTASK to the WEBLOK. If Console Mode, sets on the batching bits in the SNARBLOK.

DMKVCVER
Sets a timer.

DMKVCRWT
If WEBLOK is to be sent immediately, that is if CONPARM= priority, or highlight, or alarm, or the pace counter has reached zero, calls:

DMKVCVIX
Sends the WEBLOK.

DMKVCVCE
Writes a Trace Element.

DMKVCRWT
If a response is expected to the Write, exits to Dispatcher via GOTO. If a response is not expected, sets the return code in savearea pointed to by CONRETN.

DMKQCOET
Release the CONTASK.

DMKVCRWT
Exits to Dispatcher via GOTO.

CP Directories

This part contains the following directories:

- CP Module Entry Point Directory
- CP Module-to-Label Cross-Reference
- CP Label-to-Module Cross-Reference.

CP Module Entry Point Directory

Module Name	Entry Points	Attributes, Function
DMKACO		Pageable.
	DMKACOCL	Closes the current accounting spool file.
	DMKACODS	Creates an account card buffer for a VMBLOK (for a disconnect user).
	DMKACODV	Builds an account card buffer for a VDEVBLOK.
	DMKACOFF	Creates an account card buffer for a VMBLOK (for a logged-off user).
	DMKACON	Provides additional accounting function at logon time (for installation use).
	DMKACOQU	Collects the accounting records on the system accounting chain (DMKRSPAC) and puts them into spool file format.
	DMKACOTM	Creates a connect and usage time message for a user.
	DMKACOSA	Creates an accounting buffer for a VCNA logical unit user.
DMKACR		Resident.
	DMKACRCO	Marks all paths through specified channels offline, and checks that there are still online paths to all system-owned devices.
	DMKACRCT	Terminates one or more failing channels.
	DMKACRCV	Recovers from loss of one or more channels or from the loss of a processor and its channels.
	DMKACRC3	Sends an offline message to the operator.
DMKACS		Resident.
	DMKACSCV	Recovers from the loss of one or more channels.
DMKALG		Pageable.
	DMKALGON	Handles the AUTOLOG command.
DMKALO		Pageable.
	DMKALOCP	Special processing required for mounting CP-owned volumes during system initialization.
DMKAPI		Pageable. This module is entered from DMKCPI only if in attached processor or multiprocessor mode. It is also entered from DMKCPU as part of the vary online processor function.
	DMKAPICK	Test for virtual machine assist and 370E on non-IPL'ed processor. Also test if its interval timer is running.
	DMKAPIPR	Initializes the PSAs for each processor.
	DMKAPIAP	Initializes the control registers for the second processor.
DMKAPS		Pageable.
	DMKAPSIL	Handles pending messages representing spool requests for the *SPL System Service.
	DMKAPSCN	Handles pending CONNECTs from a virtual machine to the *SPL System Service.
	DMKAPSSV	Handles pending SEVERs from a virtual machine to the *SPL System Service.
DMKAPT		Pageable.
	DMKAPTSP	Handles pending messages from a virtual machine requesting the *SPL System Service to read SPLINK(s) into the virtual machine's data area.
	DMKAPTSF	Handles pending messages from a virtual machine requesting the *SPL System Service to read the SFBLOK for the specified spool ID into the virtual machine's data area.
	DMKAPTRX	Handles pending messages from a virtual machine requesting the *SPL System Service to read the XABs (External Attribute Buffers) for the specified spool file ID into the virtual machine's data area.

Module Name	Entry Points	Attributes, Function
DMKAPU		Pageable.
	DMKAPUSE	Handles pending messages from a virtual machine requesting the *SPL System Service to select a spool file from the system queue for the virtual machine to process.
DMKAPV		Pageable.
	DMKAPVCL	Handles pending messages from a virtual machine requesting the *SPL System Service to either close and purge, close and requeue, or change the copy count of the specified spool ID.
DMKAPW		Pageable.
	DMKAPWPG	Removes an *SPL System Service logical spool control block from the logical print chain for a virtual machine.
DMKAPX		Pageable.
	DMKAPXMG	Sends messages from a virtual machine using the *SPL System Service.
DMKAPY		Pageable.
	DMKAPYSD	Routes commands and messages from the *SPL System Service to the virtual machine.
DMKAPZ		Pageable.
	DMKAPZNO	Notifies all idle virtual machines that are using the *SPL System Service when a spool file becomes available.
DMKATS		Pageable.
	DMKATSCF	Notifies the virtual machine that the command has replaced the shared system with a private copy of that shared system. The user continues to run without the shared copy of the named system. Called by the command processors via an SVC if the command execution is to change a shared page.
DMKBIO		Pageable.
	DMKBIOCN	Handles pending CONNECTs from a virtual machine to the DASD Block I/O System Service.
	DMKBIOIL	Handles pending messages representing I/O requests for DASD Block I/O.
	DMKBIOSV	Handles pending SEVERs from a virtual machine to the DASD Block I/O System Service.
	DMKBIORS	Handles the resetting of a connection to the DASD Block I/O System Service.
DMKBLD		Pageable.
	DMKBLDEC	Allocates storage for a virtual ECBLOK and the two TRQBLOKs required for a virtual machine with the ECMODE option, and initializes these blocks.
	DMKBLDRL	Releases real segment, page, and swap tables to free storage.
	DMKBLDRT	Creates and initializes segment, page, and swap tables as a function of virtual storage size, which is part of the process of building a user's virtual machine.
	DMKBLDVM	Creates and partially initializes a VMBLOK for a virtual machine, identified by its terminal real device block.
DMKBOX		Pageable. Provides the system or user logo (header) for printed output.
	DMKBOXMS	Minimum screen VM logo data.
	DMKBOXNS	Normal screen VM logo data.
	DMKBOXPR	Printer separator page logo data.

Module Name	Entry Points	Attributes, Function
DMKBSC		Pageable. Line error processing for remote 3270s on binary synchronous lines only.
	DMKBSCER	Examines the error condition resulting from a unit check or channel error that occurred while executing a CP-generated BSC line channel program. If the error is uncorrectable, DMKMSW is called to notify the operator. After return from DMKMSW, the original channel program is terminated and the fatal flag is set in the IOBLOK. If the error is correctable, the channel program is re-executed up to a maximum of seven retries.
DMKBTS		Pageable. (*Note: This module is not supplied with source code.*)
	DMKBTS	Bootstrap loader routines for the 37x5 processor.
DMKCAC		Pageable. Processes the CACHE command for 3880 control units.
	DMKCACOF	Processes the CACHE OFF command for 3880 Model 13 or 23.
	DMKCACON	Processes the CACHE ON command for 3880 Model 13 or 23.
	DMKCACQY	Processes the CACHE QUERY command for 3880 Model 11, 21, 13, or 23.
DMKCAO		Pageable.
	DMKCAOWN	Processes the CACHE OWN command for the 3880 Model 13 and Model 23 control units.
DMKCCD		Resident.
	DMKCCDAS	Validates CCW commands to nondedicated DASD.
	DMKCCDSK	Insert SEEK command.
DMKCCF		Resident.
	DMKCCFBA	Validates command codes for nondedicated FBA.
	DMKCCFBD	Validates command codes for dedicated FBA.
DMKCCH		Resident. Operates with the I/O interrupt handler to schedule a device-dependent error recovery procedure when a channel data check, control check, or interface control check is detected.
	DMKCCHER	Continue channel check processing after obtaining the global system lock.
	DMKCCHIS	Entry from DMKIOS when a channel check occurs when storing a CSW after a SIO.
	DMKCCHNT	Entry from DMKIOTIN when a channel check occurs on an I/O interrupt.
	DMKCCHRF	Reflects channel check information to the virtual machine.
	DMKCCHRT	Entry from DMKIOE to allow error messages to be printed.
DMKCCO		Resident.
	DMKCCODD	Validates dedicated DASD command codes.
	DMKCCOMP	Validates tape command codes.
	DMKCCOMS	Validates 3851 MSC command codes.
	DMKCCOTH	Validates other command codes.
DMKCCS		Resident.
	DMKCCSEN	Handles sense type ('X4') commands.
	DMKCCSLK	Obtains the system lock.
	DMKCCSRM	Gets larger rcwtask for CCW translation.

Module Name	Entry Points	Attributes, Function
DMKCCT		Resident.
	DMKCCTCN	Validates console command codes.
	DMKCCTDL	Validates dial command codes.
	DMKCCTLC	Validates SDLC command codes.
	DMKCCTRM	Validates terminal command codes.
DMKCCW		Resident.
	DMKCCWCN	Obtains control data for other CCW translation modules.
	DMKCCWCW	Writes control data for other CCW translation modules.
	DMKCCWRE	Reenters DMKCCWTR.
	DMKCCWRT	Entry for return from other CCW translation modules.
	DMKCCWSB	Obtains control data for a user.
	DMKCCWTR	Takes the list of virtual CCWs associated with a user's SIO and translates it into a real CCW list.
DMKCDB		Pageable. Processes DISPLAY, DCP commands.
	DMKCDBDC	Executes the DISPLAY command to display real storage locations.
	DMKCDBDI	Displays virtual storage locations, storage keys, general registers, floating-point registers, vector registers, PSW, CAW, and CSW at the terminal.
DMKCDM		Pageable Processes DUMP and DMCP commands.
	DMKCDMDM	Dumps the contents of the specified real storage locations on the virtual printer spool file.
	DMKCDMDU	Dumps the contents of the specified virtual storage locations, registers, PSW, and storage keys on the virtual printer spool file.
DMKCDS		Pageable. Processes STORE and STCP commands.
	DMKCDSCP	Stores data into real storage (STCP command).
	DMKCDSTO	Stores data into virtual storage (STORE command).
DMKCFC		Pageable. Gets the address of the routine that processes the CP console function that was requested.
	DMKCFCCO	Override command/function classes.
	DMKCFCMD	Processes a CP console function.
DMKCFD		Pageable. Processes LOCATE and ADSTOP commands.
	DMKCFDAD	Stops virtual machine at specified address (ADSTOP command).
	DMKCFDLO	Displays address of real device blocks, or VMBLOK and/or virtual device blocks (LOCATE command).
DMKCFF		Pageable.
	DMKCFFSB	Processes subroutines of DMKCFG.
DMKCFG		Pageable.
	DMKCFGCL	Handles Diagnose code X'64'.
	DMKCFGII	Entry to IPL from LOGON (DMKLOG).
	DMKCFGIP	Entry to IPL from a command line (DMKCFM).
	DMKCFGIR	Distinguishes a re-IPL for users of the Protected Application Facility.
DMKCFH		Pageable.
	DMKCFHAS	Saves a virtual machine's storage space including registers and PSW (for VMSAVE).
	DMKCFHSV	Saves a virtual machine's storage space including registers and PSW (SAVED System).

Module Name	Entry Points	Attributes, Function
DMKCFJ		Pageable.
	DMKCFJSL	Processes the SLEEP command.
	DMKCFJBE	Processes the BEGIN command.
	DMKCFJRQ	Presents an attention interruption to the virtual machine to simulate a real request key interruption.
DMKCFM		Resident. Processes DIAGNOSE code 8. It scans the command line and goes to the required module.
	DMKCFMAT	Posts an attention interrupt pending for the virtual machine.
	DMKCFMBK	Puts the terminal in console function (CP) mode (ATTN key pressed twice). Scans the command line and goes to the command- handling routine.
	DMKCFMEN	Entered when DIAGNOSE code 8 is executed. Scans the command line and goes to the command-handling routine.
	DMKCFMRU	Runs the virtual machine.
	DMKCFMSD	Execute a CP command that is in the text of the SEND command.
	DMKCFMWU	Entered when SLEEP time expires.
DMKCFO		Pageable.
	DMKCFOSA	Processes the SET SASSIST command.
	DMKCFOSC	Processes the SET CPASSIST command.
	DMKCFOSF	Processes the SET FAVORED command.
	DMKCFOSP	Processes the SET PRIORITY command.
	DMKCFOSQ	Processes the SET QDROP command.
	DMKCFOSR	Processes the SET RESERVE command.
	DMKCFOS3	Processes the SET S370E command.
DMKCFP		Pageable. Simulates the operator's console for the virtual machine.
	DMKCFPRR	Handles system resets for other CP routines. Resets the virtual machine.
DMKCFQ		Pageable.
	DMKCFQRD	Handles virtual device reset for other CP routines.
DMKCFR		Pageable.
	DMKCFREP	Handles virtual device reset for other CP routines.
DMKCFS		Pageable. Processes the CP SET command for general users.
	DMKCFSAC	Processes the SET ACNT command.
	DMKCFSAP	Processes the SET AUTOPOLL command.
	DMKCFSAS	Entry point for VMSAVE enable/disable.
	DMKCFSCC	Processes the SET CPCONIO command.
	DMKCFSCP	Processes the SET CUPID command.
	DMKCFSEC	Processes the SET ECMODE command.
	DMKCFSEM	Processes the SET EMSG command.
	DMKCFSIM	Processes the SET IMSG command.
	DMKCFSIS	Processes the SET ISAM command.
	DMKCFSLE	Processes the SET LINEDIT command.
	DMKCFSMG	Processes the SET MSG command.
	DMKCFSNT	Processes the SET NOTRANS command.
	DMKCFSPX	Processes the SET PAGEX command.
	DMKCFSRN	Processes the SET RUN command.
	DMKCFSSA	Processes the SET SVCACCL command.
	DMKCFSSM	Processes the SET SMSG command.
	DMKCFSVC	Processes the SET VMCONIO command.
	DMKCFSVS	Processes the SET VMSAVE command.
	DMKCFSWG	Processes the SET WNG command.
	DMKCFS37	Processes the SET 370E command.

Module Name	Entry Points	Attributes, Function
DMKCFT		Pageable.
		Processes user's terminal options.
	DMKCFTRM	Entry point for the TERMINAL command processor.
DMKCFU		Pageable.
	DMKCFUDU	Processes the SET DUMP command.
	DMKCFULO	Processes the SET LOGMSG command.
	DMKCFUMI	Processes the SET MITIME command.
	DMKCFUMO	Processes the SET MODE command.
	DMKCFUMW	Processes the SET MINWS command.
	DMKCFUPA	Processes the SET PAGING command.
	DMKCFURE	Processes the SET RECORD command.
DMKCFV		Pageable.
	DMKCFVSB	Processes the SET STBYPASS command.
	DMKCFVSM	Processes the SET STMULTI command.
	DMKCFVMI	Processes the SET MIH command.
DMKCFW		Pageable.
	DMKCFWEP	Processes SCREEN command parameters.
DMKCFY		Pageable.
	DMKCFYAG	Processes the SET AFFINITY command.
	DMKCFYAS	Processes the SET AFFINITY command.
	DMKCFYPF	Allocates or releases the retrieve buffer. Releases the PFKTABLE if not needed.
	DMKCFYSA	Processes the SET ASSIST command.
	DMKCFYSC	Processes the SET CONCEAL ON/OFF command, placing the user in the Protected Application Facility.
	DMKCFYSP	Processes the SET PFNN command.
	DMKCFYSM	Processes the SET TIMER command.
DMKCKD		Pageable.
	DMKCKDEV	Checkpoint program device halting routine.
	DMKCKDGP	Finds a usable path to a device.
	DMKCKDSW	Handles requests to switch from one processor to another.
DMKCKF		Pageable.
	DMKCKFIL	Checkpoints the onto the warm start cylinders during system shutdown. (Saves open CPTRAP file when the system crashes.)
DMKCKH		Pageable.
	DMKCKHST	Checks validity of warm start cylinders.
	DMKCKHPR	Handles read or write requests to and from the warm start area.
	DMKCKHAC	Saves printer or punch spool files which are active on the system printer or punch, or adds open spool file from the system reader to the delete queue.
	DMKCKHIO	Handles I/O requests.
DMKCKM		Pageable.
	DMKCKMSG	Displays messages for the checkpoint program.
	DMKCKMSV	Saves multiple virtual machines during system shutdown or system abend.
DMKCKN		Pageable.
	DMKCKNRD	Fixes up the RDEVBLOKs of all CP-owned volumes for use by DMKCKM and DMKCKNIO.
	DMKCKNIO	Routine to do I/O for the checkpoint program.

Module Name	Entry Points	Attributes, Function
DMKCKP		Pageable.
	DMKCKPT	IPL entry point for system initialization, system shutdown, and system abend.
	DMKCKPER	Displays fatal error message and loads a disabled wait state PSW (code 007).
	DMKCKPRG	Loads a disabled wait state PSW (code 007).
DMKCKR		Pageable.
	DMKCKRSY	Rebuilds SYSSPOOL's virtual storage from the CKPT cylinders.
DMKCKS		Pageable. Performs checkpoint processing.
	DMKCKSCV	Converts a user-unique ID to CCPD or PPPD.
	DMKCKSPL	Performs a checkpoint on any alterations in the spool file set up to allow the recovery routine to get them if warm start fails.
	DMKCKSIN	Initializes the check point cylinder after a successful warm start from the standard recovery procedure or after a cold start.
DMKCKT		Pageable.
	DMKCKTIN	Recovers spool file data if new RECBLOK needed.
	DMKCKTMP	Obtains pages for and initializes the Spool File Bit Map. Sets up constants in DMKRSP.
	DMKCKTSD	Gives system-unique spool ID back.
	DMKCKTSU	Finds a system-unique user ID.
	DMKCKTUU	Finds a user-unique spool ID and a system-unique spool ID.
DMKCKV		Pageable.
	DMKCKVWM	Allocates pages of spool files saved on CKPT cylinders, recovers RDEVBLOK information and SHQBLOKs, updates the system spool file bit map, and rebuilds the user reader chains.
DMKCKW		Pageable.
	DMKCKWHT	Saves the Reader Hash Table plus extension pages onto the warm start cylinders.
	DMKCKWSP	Saves SYSSPOOL's virtual storage onto the warm start cylinders.
DMKCLK		Pageable.
	DMKCLKCK	Determines if the clock should be synchronized. (Called from DMKCPI)
	DMKCLKCC	Handles CLKCHK signal request.
	DMKCLKMP	Synchronizes the clocks.
	DMKCLKAP	Handles SYNC signal request.
	DMKCLKSC	Handles the TOD-sync-check external interrupt.
DMKCMD		Pageable. Contains tables which describe the syntax, entry points, and functional groups for subcommands. These tables enable DMKCFCMD to do centralized class validation for CP commands.
DMKCNS		Resident. Real console terminal manager (along with DMKCNT).
	DMKCNSEN	Enables or disables a low-speed terminal line.
	DMKCNSIC	Entered from DMKQCN to initialize read and write CCWs for the CONTASK built by DMKQCN and DMKQCO.
	DMKCNSIN	Interruption return point and handler for terminal I/O.
DMKCNT		Resident. Real console terminal manager (along with DMKCNS).
	DMKCNTED	Edits the input line for the following characters: escape, line end, line delete, and character delete.

Module Name	Entry Point	Attributes, Function
DMKCPB		Pageable. Simulates the operator's console for the virtual machine.
	DMKCPBEX	Processes the EXTERNAL command to present an external interruption to the virtual machine.
	DMKCPBNR	Processes the NOTREADY command to cause the virtual device to appear not ready.
	DMKCPBRS	Processes the RESET command to reset all pending interrupts from the specified device.
	DMKCPBRW	Processes the REWIND command to issue a rewind to the real tape device.
	DMKCPBRY	Processes the READY command to simulate a device end interrupt to the specified device.
	DMKCPBSR	Processes the SYSTEM command to simulate system reset and PSW restart to allow clearing of storage.
DMKCPE		Resident. Contains data constants that define the end of the CP nucleus.
DMKCPI		Pageable. Prepares system for operation.
	DMKCPINT	Initializes CP during IPL. Mainline system initialization routine (first half).
DMKCPJ		Pageable.
	DMKCPJNT	Mainline system initialization routine (second half).
DMKCPM		Pageable.
	DMKCPMIO	Initializes all resources necessary for an AP/MP environment and reverts to uniprocessor mode.
DMKCPN		Pageable.
	DMKCPNAS	Assigns a 3480 tape device.
	DMKCPNVY	Processes the VARY ONLINE command.
DMKCPO		Pageable.
	DMKCPOFF	Process the VARY OFFLINE PROCESSOR command.
DMKCPP		Pageable.
	DMKCPPUP	Releases all resources necessary for an AP/MP environment and reverts to uniprocessor mode.
DMKCPS		Pageable. Processes the SHUTDOWN and HALT commands.
	DMKCPSSH	Processes the SHUTDOWN command.
	DMKCPSH	Processes the HALT command.
DMKCPT		Pageable. Processes the VARY command.
	DMKCPTNF	Processes the VARY command.
DMKCPU		Pageable.
	DMKCPUVY	Processes the VARY ONLINE PROCESSOR command.
DMKCPV		Pageable.
	DMKCPVAA	Punches user accounting records.
	DMKCPVAC	Processes the ACNT command to create accounting records for logged-on users. Also, resets accumulated accounting information.
	DMKCPVAE	Enables system low-speed lines for system restart.
	DMKCPVDS	Processes the DISABLE command to disable an active line after the current user is finished with it.
	DMKCPVEN	Processes the ENABLE command to enable the system's low-speed lines for system logon.

Module Name	Entry Points	Attributes, Function
DMKCPW		Pageable.
	DMKCPWFB	Initiates IOBLOKs and builds RDCBLOKs.
	DMKCPWUN	Unassigns a device and resigns it from the path group.
	DMKCPWVF	Varies a device offline.
DMKCPX		Pageable.
	DMKCPXCK	Checks the number of t-disks to be released or cleared.
	DMKCPXZT	Builds and stacks a CPEXBLOK.
DMKCPY		Pageable.
	DMKCPYLK	Processes the LOCK command to lock specified pages of a user's virtual storage space into real main storage.
	DMKCPYUL	Processes the UNLOCK command to unlock pages that were locked by operator command (LOCK).
DMKCPZ		Pageable.
	DMKCPZAS	Assigns a 3480 tape device to CP
	DMKCPZFD	Checks cached control units for status and cache validity.
	DMKCPZMG	Stacks message DMKCPJ328 on the DMKCPIMS queue
	DMKCPZPG	Varies on 3880 storage control devices.
	DMKCPZVR	Checks cached control units for V=R user owning the cache.
DMKCQC		Pageable.
	DMKCQCPT	Processes the QUERY CPTRAP command.
DMKCQG		Pageable.
		Processes the class G and class D QUERY commands.
	DMKCQGID	Processes the QUERY <VADDR>/<USERID> command.
	DMKCQGQA	Processes the QUERY VIRTUAL ALL command.
	DMKCQGQC	Processes the QUERY VIRTUAL CONSOLE command.
	DMKCQGQD	Processes the QUERY VIRTUAL DASD command.
	DMKCQGQG	Processes the QUERY VIRTUAL GRAF command.
	DMKCQGQH	Processes the QUERY VIRTUAL CHANNEL command.
	DMKCQGQL	Processes the QUERY VIRTUAL LINES command.
	DMKCQGQS	Processes the QUERY VIRTUAL STORAGE command.
	DMKCQGQT	Processes the QUERY VIRTUAL TAPES command.
	DMKCQGQU	Processes the QUERY VIRTUAL UR command.
DMKCQH		Pageable.
	DMKCQHFI	Entry point from DMKCPI/DMKLOH for QUERY FILES processing.
	DMKCQHFS	Processes the QUERY FILES command.
	DMKCQHIP	Processes the QUERY FILES command for the DMKCPI call.
	DMKCQHNG	Processes the QUERY PUN (general) command.
	DMKCQHNS	Processes the QUERY PUN (authorized) command.
	DMKCQHRG	Processes the QUERY RDR (general) command.
	DMKCQHRS	Processes the QUERY RDR (authorized) command.
	DMKCQHTG	Processes the QUERY PRT (general) command.
	DMKCQHTS	Processes the QUERY PRT (authorized) command.
DMKCQI		Pageable.
	DMKCQIFR	Format response to the QUERY RDR/PRT/PUN command.

Module Name	Entry Points	Attributes, Function
DMKCQP		Pageable.
		Processes the class B and class G QUERY command.
	DMKCQPQA	Processes the QUERY ALL command.
	DMKCQPQD	Processes the QUERY DASD command.
	DMKCQPQG	Processes the QUERY GRAF command.
	DMKCQPQL	Processes the QUERY LINES command.
	DMKCQPQP	Processes the QUERY PROCESSR command.
	DMKCQPQS	Processes the QUERY STOR command.
	DMKCQPQT	Processes the QUERY TAPES command.
	DMKCQPQU	Processes the QUERY UR command.
	DMKCQPSC	Entry called by DMKCQQ to handle the QUERY raddr command.
DMKCQQ		Pageable.
	DMKCQQID	Processes the QUERY <RADDR>/<USERID> command.
	DMKCQQQL	Processes the QUERY LINKS command.
	DMKCQQQS	Processes the QUERY SYSTEM command.
	DMKCQQQT	Processes the QUERY TDSK command.
	DMKCQQPT	Entry called by DMKCQP to get Path information.
DMKCQR		Pageable.
		Processes the QUERY command.
	DMKCQRAF	Processes the QUERY AFFINITY command.
	DMKCQRDP	Processes the QUERY DUMP command.
	DMKCQRHD	Processes the QUERY HOLD command.
	DMKCQRPG	Processes the QUERY PAGING command.
	DMKCQRPR	Processes the QUERY PRIORITY command.
	DMKCQRQD	Processes the QUERY QDROP command.
DMKCQS		Pageable.
	DMKCQSSC	Processes QUERY SCREEN command.
	DMKCQSMI	Processes QUERY MITIME command.
	DMKCQSNA	Processes QUERY NAME command.
	DMKCQSPS	Processes QUERY PSTOR command.
	DMKCQSQC	Processes COMMANDS command.
DMKCQT		Pageable.
	DMKCQTCL	Processes the QUERY CPLANG command.
	DMKCQTRE	Searches remote devices for QUERY GRAF.
	DMKCQTSN	Searches SNA devices for QUERY GRAF.
	DMKCQTST	Processes the QUERY STATUS command.
DMKCQU		Pageable.
	DMKCQUSE	Processes the QUERY SECUSER command.
	DMKCQUST	Processes the QUERY SET command.
	DMKCQUTE	Processes the QUERY TERMINAL command.
DMKCQY		Pageable.
	DMKCQYCA	Processes the QUERY CPASSIST command.
	DMKCQYCL	Processes the QUERY CPLEVEL command.
	DMKCQYCP	Processes the QUERY CPUID command.
	DMKCQYLM	Processes the QUERY LOGMSG command.
	DMKCQYPF	Processes the QUERY PF<NN> command.
	DMKCQYSA	Processes the QUERY SASSIST command.
	DMKCQYSP	Processes the QUERY SPMODE command.
	DMKCQYS3	Processes the QUERY S370E command.
	DMKCQYTI	Processes the QUERY TIME command.
	DMKCQYUI	Processes the QUERY USERID command.
	DMKCQYUS	Processes the QUERY USERS command.
	DMKCQYVS	Processes the QUERY VMSAVE command.
	DMKCQYRG	Displays the page range of a saved VMSAVE system.

Module Name	**Entry Points**	**Attributes, Function**
DMKCRM		Pageable. The Collection Resource Management (CRM) system service. It provides a communication path between the TSAF virtual machine and CP.
	DMKCRMCN	Handles pending connections.
	DMKCRMIL	Handles pending messages.
	DMKCRMQS	Handles quiesces.
	DMKCRMSV	Handles severs.
DMKCSB		Pageable.
	DMKCSBLD	Processes the LOADBUF command (real UCS or FCB buffer).
	DMKCSBLF	Reloads the last FCB buffer.
	DMKCSBSP	Spaces to the first line of the next form.
DMKCSC		Pageable.
	DMKCSCLV	Searches through module DMKFCB for the requested FCB image, which will be loaded into the virtual FCB buffer. This entry point will be called for virtual printer buffer loading.
	DMKCSCLD	Obtains the appropriate CCWs and points to the correct data module for loading the UCS/FCB buffer of the real printer. Searches for the requested UCS/FCB image in the data module. This entry point will be called for real printer buffer loading.
DMKCSF		Pageable. Processes real spooling commands for real unit record devices.
	DMKCSFBS	Processes the BACKSPACE command.
	DMKCSFFL	Processes the FLUSH command.
	DMKCSFRP	Processes the REPEAT command.
	DMKCSFSP	Processes the SPACE command.
DMKCSO		Pageable. Processes real spooling commands for real unit record devices.
	DMKCSODR	Processes the DRAIN command.
	DMKCSOSD	Restarts a device after it has been drained.
	DMKCSOST	Processes the START command by device type.
DMKCSP		Pageable.
	DMKCSPSP	Processes the SPOOL command.
DMKCSQ		Pageable.
	DMKCSQCL	Processes the CLOSE command.
	DMKCSQFR	Processes the FREE command.
	DMKCSQHL	Processes the HOLD command.
DMKCSR		Pageable.
	DMKCSRGT	Searches the device type from the SPOOL command.
	DMKCSRPO	Processes the SPOOL command from general users and spooling operators.
DMKCST		Pageable. Processes class G commands.
	DMKCSTAG	Entry point to process the TAG command.
DMKCSU		Pageable. Processes the class D and G spooling commands.
	DMKCSUCG	Processes the CHANGE (general) command.
	DMKCSUCS	Processes the CHANGE (authorized) command.

Module Name	Entry Points	Attributes, Function
DMKCSV		Pageable.
	DMKCSVOG	Processes the ORDER (general) command.
	DMKCSVOS	Processes the ORDER (authorized) command.
	DMKCSVPG	Processes the PURGE (general) command.
	DMKCSVPS	Processes the PURGE (authorized) command.
DMKCSW		Pageable.
	DMKCSWCH	Processes the CHANGE command for general users and spooling operators.
DMKCSX		Pageable.
	DMKCSXTG	Processes the TRANSFER command for general users.
	DMKCSXTS	Processes the TRANSFER command for general users and spooling operators.
DMKCSY		Pageable.
	DMKCSYTR	Along with DMKCSX, processes the TRANSFER command.
DMKCVT		Resident. Processes the conversion routines.
	DMKCVTAB	Forces a CVT001 abend.
	DMKCVTBD	Converts a word of binary data into a doubleword of decimal digits.
	DMKCVTBH	Converts a word of binary data into a doubleword of hexadecimal data.
	DMKCVTDB	Converts a decimal field into a fullword of binary data.
	DMKCVTDT	Converts data and time to EBCDIC and inserts it into a specified location.
	DMKCVTHB	Converts the designated hexadecimal field into a binary fullword.
DMKCVU		Pageable.
	DMKCVUFP	Converts a floating-point doubleword into 17 bytes of decimal data.
DMKDAD		Resident.
	DMKDADER	Processes 3375/3380 error recovery for CP and DIAGNOSE I/O.
DMKDAS		Resident. DASD error retry program.
	DMKDASER	Retries the failing DASD channel program.
DMKDAU		Resident. FBA error retry program.
	DMKDAUER	Retries the failing FBA channel program.
DMKDDC		Residency not applicable.
	DMKDDC	Performs data decompaction.
DMKDDR		Residency not applicable. This is the DASD dump restore program. It saves data from a direct access volume onto a tape or tapes. It returns data to DASD from tape that has been placed on the tape by this program. It copies data from one device to another of the same type. It prints a translation of each record specified on the SYSPRINT device. Prints a translation of each record specified on the console. Initial program loaded or run under CMS if on a CMS disk.
	DMKDDREP	DASD dump restore program entry point
	DMKDDRED	End-of-load module for CMS.

Module Name	Entry Points	Attributes, Function
DMKDEF		Pageable. Processes the DEFINE command to define a virtual device.
	DMKDEFDG	Processes the DEFINE (general) command.
	DMKDEFDS	Processes the DEFINE (authorized) command.
DMKDEG		Pageable.
	DMKDEGIN	Processes the DEFINE STORAGE and DEFINE CHANNEL commands.
DMKDEI		Pageable.
	DMKDEIMS	Define or redefine MSS disks.
DMKDEX		Resident.
	DMKDEXIN	Cache bypass routine for virtual I/O.
DMKDGD		Resident. Processes simple disk I/O.
	DMKDGDDK	Performs simple disk I/O of a standardized format with a minimum of CCW chain manipulation.
	DMKDGDUL	Unlocks user pages upon completion of I/O.
DMKDGF		Resident.
	DMKDGFIN	Handles simple disk interruptions.
DMKDIA		Pageable.
	DMKDIADR	Releases a terminal line that has been in use by the virtual machine via the DIAL command. The line is detached from the virtual machine and made available for normal log on to the system.
	DMKDIAL	Processes the DIAL command. Attaches a user's terminal as a dedicated device to an existing virtual 270X terminal line in the virtual machine addressed by the command line. This process is finished in DMKDIFDI.
DMKDIB		Pageable.
	DMKDIBSM	Simulates sense data and status for virtual I/O to a simulated I/O device (2702 line or CTCA) that has not yet been activated through either the console function DIAL for 2702 lines, or the console function COUPLE for virtual CTCAs.
	DMKDIBCP	COUPLE command processor. Establishes a virtual connection between two channel-to-channel adapters on a single virtual machine.
	DMKDIBDR	Releases a terminal line that has been in use by the virtual machine via the DIAL command. The line is detached from the virtual machine and made available for normal logon to the system.
DMKDID		Resident. Handles missing interruptions.
	DMKDIDDA	Examines RDEVBLOKs for DASD.
	DMKDIDGR	Examines RDEVBLOKs for graphic devices (except 1053 and 328X printers).
	DMKDIDLF	Examines RDEVBLOKs for missing interrupts during LOGOFF/FORCE.
	DMKDIDMS	Examines RDEVBLOKs for Mass Storage System devices.
	DMKDIDTA	Examines RDEVBLOKs for tape devices.
	DMKDIDUR	Examines RDEVBLOKs for unit record devices (except 3800 and 3289E printers).
DMKDIF		Pageable.
	DMKDIFDI	Finishes DIAL command processing begun in DMKDIAL.

Module Name	Entry Points	Attributes, Function
DMKDIR		Pageable or standalone. Initial program loaded or run under CMS if on a CMS disk.
	DMKDIRCT	Builds a user directory on a system owned volume using pre-allocated cylinders.
	DMKDIRED	End of load module for CMS.
DMKDMP		Resident. Writes a dump of main storage, control registers, floating-point registers, general registers, and clocks to a specified device. Calls DMKDMQ for additional functions.
	DMKDMPDK	Writes the dump on the specified device.
	DMKDMPRS	Initial program loads the system over again.
DMKDMQ		Resident.
	DMKDMQEN	Writes dump to tape or printer.
	DMKDMQLW	Loads a disabled wait state PSW.
	DMKDMQGP	Locates an enabled path to a device.
	DMKDMQMC	Handles machine checks during dump.
	DMKDMQPC	Handles program checks during dump.
	DMKDMQSW	Switches to other processor.
	DMKDMQTL	Types a line on the system console.
DMKDNC		Residency not applicable.
	DMKDNC	Performs data compaction.
DMKDRD		Pageable. Process spool files.
	DMKDRDDD	Delete system dump spool file.
	DMKDRDER	Manipulates input spool files via a DIAGNOSE code X'0014' issued by the virtual machine.
	DMKDRDMP	Reads a system dump spool file via a DIAGNOSE code X'0034' issued by the virtual machine.
	DMKDRDSY	Reads the system symbol table CSECT via a DIAGNOSE code X'0038' issued by the virtual machine.
DMKDSB		Resident. DASD error retry program.
	DMKDSBRD	Processes unsolicited device end interruptions.
	DMKDSBSD	Collects DASD sense data.
DMKDSP		Resident. Entered after each interruption handler is finished processing and after each stacked CPEXBLOK, I/O request, and external interruption has been serviced. It updates the CPU times charged to the user that has received service, updates all virtual timers, and reflects any pending interruptions for which the user is enabled. After the user's status has been updated, the highest-priority runnable user is dispatched.
	DMKDSPA	Immediate redispatch path for virtual machines. The only status update that occurs is for virtual timers.
	DMKDSPB	Process new virtual PSW and dispatch. Entered if the virtual PSW has been entered outside of DMKDSP.
	DMKDSPCH	Main entry point. Updates timers and dispatches user.
	DMKDSPIS	Saves pending interrupts for the V=R virtual machine.
	DMKDSPE	Processes interrupt from virtual interval timer.
	DMKDSPQS	Nonexecutable dispatched user's maximum time slice.
	DMKDSPRU	Entered in attached processor mode when the system lock is not held.
	DMKDSPWA	Entry point for leaving active wait with a runnable user (AP/MP only).

Module Name	Entry Points	Attributes, Function
DMKEIG		Pageable. Analyzes the 2880 channel logout and sets appropriate bits in the ECSW field according to the results of this analysis. It moves the channel logout to the channel check record.
DMKEMR		Pageable
	DMKEMR	Contains the framework for the common response messages which are generated at various places within CP. Module DMKERM references DMKEMR in order to write messages which require variable data to be inserted. This module contains no executable code and contains message text for responses 30,001 and up.
DMKENT		Resident. Meets the CP nucleus residency requirements for TRQBLOK and CPEXBLOK entries to pageable VM Monitor module DMKMIA.
	DMKENTEC	Used to invoke a MONITOR STOP command via a CPEXBLOK.
	DMKENTET	Used to invoke a MONITOR STOP command via a TRQBLOK request.
	DMKENTFI	Used to complete monitor shutdown processing, via CPEXBLOK.
	DMKENTGP	Used for CACHED controller monitoring via a CPEXBLOK.
	DMKENTGQ	Used to gather subsystem counts for CACHED controller monitoring via a CPEXBLOK.
	DMKENTKC	Used to invoke a MONITOR CLOSE command via a CPEXBLOK.
	DMKENTSC	Used to invoke a MONITOR START SPOOL command via a CPEXBLOK.
	DMKENTST	Used to invoke a MONITOR START SPOOL command via a TRQBLOK request.
	DMKENTTI	High frequency I/O status sampling routine, entered every two seconds via TRQBLOK request.
	DMKENT62	Collect control unit and device busy and queue counts.
	DMKENT63	Collect channel busy and queue counts.
DMKEPS		Pageable.
	DMKEPSWD	Prompts the user to enter a password, types masking characters if appropriate, reads the password from the terminal, and checks the password for a match.
DMKERM		Pageable.
	DMKERMSG	Accesses the requested message from the CP message repository and inserts the module ID, message number, and data. It also prints the message.
DMKERP		Pageable.
	DMKERPCP	Record CP OBR/MDR records queued by the ERP (error recording facility).
DMKEXT		Resident.
	DMKEXTIN	CP first level interrupt handler.
	DMKEXTSP	Handles all SIGNAL actions after CP initialization.
	DMKEXTSL	Second level interrupt handler for MP external interrupts.
DMKFCB		Pageable.
	DMKFCB	Contains the forms control load buffer images that the LOADBUF command uses to load the forms control buffer in the 3811 control unit for the 3203 or 3211 printer. The LOADVFCB command also uses DMKFCB to load the forms control buffer in the virtual 3203 or 3211 printer.

Module Name	Entry Points	Attributes, Function
DMKFMT		Standalone program. Accepts parameters from the console or IPL device (card reader) and per forms partial or complete formatting, allocating, and labeling of 3330, 3340, 3350, 3380 and 2305 DASD devices. The FORMAT program also write-checks the surfaces. Bad surfaces are flagged to prevent their use. No alternative tracks are assigned. OS labels are written to be compatible with OS, but labels indicate to OS that no space is left on the DASD device. All input parameters are verified for correctness.
DMKFPS		Resident.
	DMKFPS	Fast privileged simulation for selected privileged operations.
DMKFRE		Resident. Free storage manager.
	DMKFREDE	Entry to resume deferred free call when running AP/MP.
	DMKFREE	Gets space from free storage and processes the CP assist FREE instruction (E622).
	DMKFREEA	Supplies a cache-aligned block of storage, if the block requested will be referenced often and returned infrequently.
	DMKFREEP	Provides a fast path to obtain prime subpool blocks.
	DMKFREMX	Maxsize ... for CP assist use.
	DMKFRERC	Special entry point to acquire free storage. If the storage request cannot be satisfied, a condition code of one is returned to the caller. (This entry point may extend free storage to satisfy the request. See DMKFREXX below.)
	DMKFREXX	Special entry point to acquire free storage. If free storage would have to be extended to satisfy the request, a condition code of one is returned to the caller. Free storage will not be extended (and hence the caller will not lose control).
	DMKFRESC	Split counters for DMKMOO.
	DMKFRESW	Entry for Extend processing.
	DMKFRET	Processes the CP assist FRET instruction (E621).
DMKFRT		Resident. Free storage management module.
	DMKFRTRA	Returns any complete cache-aligned pages to the dynamic paging area.
	DMKFRTRS	Returns subpools to free storage chain.
	DMKFRTSN	Scans the free storage chain to return page frame sized blocks back to the dynamic paging area.
	DMKFRTT	Returns storage to a subpool or the large storage chain. Handles requests to return storage that cannot be handled by the CP Assist Fret instruction at DMKFRET.
	DMKFRTTE	Returns storage to the large storage chain. This entry is called by DMKUSP to return storage blocks known to have been obtained from the dynamic paging area.
	DMKFRTTR	Assigns storage to free storage management; does not release pages.
DMKGIO		Pageable. Initializes supervisor operations for tape, unit record, and nonstandard disk I/O operations.
	DMKGIOEX	Checks device validity and initializes I/O operations on tape, unit record, and nonstandard disk I/O programs per supervisor call. This module presents resultant condition code and CSW (if warranted) to the user.
DMKGRA		Pageable.
	DMKGRAOT	Performs APL/TEXT translations for outbound 3270 data.

Module Name	Entry Points	Attributes, Function
DMKGRC		Resident.
	DMKGRCUP	Generates the data stream necessary to update the 3270 screen, according to parameters input in register 2.
	DMKGRCQY	Determine if terminal or its controller has extended data stream capability.
	DMKGRCWC	Process a write CONTASK.
DMKGRD		Resident
	DMKGRDBR	Entry for branch table to DMKGRD routines from DMKGRE, DMKGRF, DMKGRG, and DMKGRI.
	DMKGRDCC	Builds CCWs for 3270 devices.
	DMKGRDIC	Starts a CONTASK.
	DMKGRDSR	Handles a read or control CONTASK.
DMKGRE		Resident.
	DMKGRECL	Clears the screen.
	DMKGREFD	Finishes processing after full-screen I/O is done.
	DMKGRESM	Handles screen management.
DMKGRF		Resident. Supports local 3270, 3278 Model 2A, and 3279 devices. DMKGRF processes interruptions and CCWs for the devices. The processing includes message handling and screen management.
	DMKGRFIN	Handles the interruption via an IOBLOK.
	DMKGRFEN	Enables or disables the device.
	DMKGRFIC	Starts a CONTASK from DMKQCN. If a buffer larger than 4K bytes is required, a CCW is built to indicate that indirect addressing will be used.
	DMKGRFTI	Processes clock comparator timer interrupts.
DMKGRG		Resident.
	DMKGRGBR	Entry for branch table to DMKGRG routines from DMKGRD, DMKGRE, DMKGRF, and DMKGRI.
	DMKGRGBK	Handles the BREAK (PA1) key (3270 only).
	DMKGRGCL	Handles the CANCEL (PA2) key (3270 and 3066).
	DMKGRGCR	Handles the CLEAR key (3270 only).
	DMKGRGER	Handles the ENTER key (3270 and 3066).
	DMKGRGPF	Handles the PF keys key (3270 only).
	DMKGRGTI	Handles the clock comparator interrupts.
	DMKGRGTR	Handles the TEST REQUEST key (3270 only).
DMKGRH		Resident.
	DMKGRHIN	Processes channel programs for 3066 display.
DMKGRI		Resident.
	DMKGRISB	Contains various subroutines used by DMKGRD, DMKGRE, DMKGRF, and DMKGRG.

Module Name	Entry Points	Attributes, Function
DMKGRT		Resident. Contains common data areas and subroutines for 3270 display support.
	DMKGRTAB	Computes the next tab position and creates the data stream to position the cursor and insert a logical tab character if necessary.
	DMKGRTAC	Count of entries in AID table.
	DMKGRTAI	Accesses total AID table.
	DMKGRTB	Orders for 327x Model 2 display terminal.
	DMKGRTBL	Table of buffer addresses for 3270s with 80 character lines.
	DMKGRTB5	Table of buffer addresses for 3270s with 132 character lines.
	DMKGRTFD	'VM/370 online' message.
	DMKGRTFM	Brings in the VM/370 logo and initializes buffer and the CCWs to write the logo in DMKGRF and DMKRGB.
	DMKGRTFO	Header for 'VM/370 online' message.
	DMKGRTPF	Accesses PF Key portion of the AID table.
	DMKGRTP6	Accesses table entry for PF6. This is used for the PA 3 key.
DMKGRU		Resident, non-executable.
	DMKGRUTB	Orders for 3278 Model 3 display terminal.
DMKGRV		Resident, non-executable.
	DMKGRVTB	Orders for 3278 Model 4 display terminal.
DMKGRW		Resident, non-executable.
	DMKGRWTB	Orders for 3278 Model 2A display terminal.
DMKGRX		Resident, non-executable.
	DMKGRXTB	Screen management data streams for 3278 Model 5 display station.
DMKHPS		Pageable.
	DMKHPSDG	Handles the graphic communication DIAGNOSE.
	DMKHPSQR	Handles an I/O operation directed to a logical device by CP.
	DMKHPSQV	Handles an I/O operation directed to a logical device by a virtual machine.
	DMKHPSHT	Executes virtual HALT I/O.
	DMKHPSEX	Reflects special external interrupt.
	DMKHPSDI	Terminates a logical device.
	DMKHPSRE	Resets all logical devices owned by a virtual machine.
DMKHPT		Pageable.
	DMKHPTEX	Reflects a special external interrupt.
	DMKHPTDI	Terminates a logical device.
	DMKHPTRE	Resets all logical devices owned by a virtual machine.
DMKHVC		Resident.
	DMKHVCAL	Performs services for the virtual machine as requested via the DIAGNOSE instruction. The specific service performed depends on the code in the DIAGNOSE instruction.
DMKHVD		Pageable.
	DMKHVDAL	Performs services for virtual machines as requested by the DIAGNOSE instruction.
DMKHVE		Pageable.
	DMKHVEAL	Performs DIAGNOSE X'2C' and X'30' for the virtual machine.
	DMKHVEYL	Data table, DASD cylinder/device.
	DMKHVEPC	Data table, DASD pages/cylinder.

Module Name	Entry Points	Attributes, Function
DMKHVF		Pageable.
	DMKHVFAL	Performs DIAGNOSE X'C8', X'CC', and X'D4' for the virtual machine.
	DMKHVFCR	Locates or creates a specified LANGBLOK.
DMKIDR		Pageable. The Identify System Service. It enables authorized virtual machines to identify themselves as owner of a specific resource or to revoke the ownership of a resource by other virtual machines.
	DMKIDRCN	Handles pending connections.
	DMKIDRFN	Handles requests to find the resource owner, received from APPCVM while processing connections to a resource name.
	DMKIDRFX	Cleans up pending messages sent to the TSAF virtual machine when the *CRM path has been severed.
	DMKIDRIN	Handles revoke messages received by the TSAF system service from the TSAF virtual machine.
	DMKIDRSV	Handles severs of the path to *IDENT (the Identify System Service).
DMKIDU		Pageable.
	DMKIDUMP	Initializes DASD dump cylinders.
	DMKIDUSF	Assigns a spool file ID to the dump SFBLOK.
DMKIMG		Pageable.
	DMKIMGBG	Provides a CMS interface for a VS-based IEBIMAGE program.
DMKIOC		Resident.
	DMKIOCVT	Converts VM/SP device type to OS/VS device type.
DMKIOE		Resident. This is the error recording module. It receives all requests for error recording and passes control to the proper pageable routine after checking if a recording is in progress. If a previous request for error recording is in progress, the current request is queued on the appropriate queue for recording at a later time. It makes a check to determine if the recording cylinder is full. DMKIOE also interfaces with the pageable module that initializes and erases the error recording cylinders.
	DMKIOECC	Handle channel check records passed from Channel Check Handler.
	DMKIOEFL	Format pages on the recording cylinders.
	DMKIOEFM	Clear and format the recording cylinders.
	DMKIOEMC	Handle machine check records passed from Machine Check Handler.
	DMKIOEMI	Format the Missing Interrupt Handler records.
	DMKIOERN	Process a 3704/3705 or remote 3270 request.
	DMKIOERR	Schedule recording for unit check, channel data check, and hardware environmental counts.
	DMKIOESD	Format hardware environmental counters.
	DMKIOESR	Schedule statistical data recording.
	DMKIOEST	Schedule the update of a statistical data request.
	DMKIOEVR	Process an SVC 76 or Missing Interrupt Handler request.

Module Name	Entry Points	Attributes, Function
DMKIOF		Resident. Records system and I/O errors on the system disk in predefined error recording cylinders.
	DMKIOFC1	Records channel check error from SIO in DMKIOS when cc=1.
	DMKIOFCN	Handles CONNECTs from a virtual machine to the Error Logging System Service.
	DMKIOFIN	Initializes pointers to available recording pages at IPL and after an erase has been completed.
	DMKIOFM1	Records machine checks.
	DMKIOFOB	Entry for a stacked outboard error recording request.
	DMKIOFST	Updates statistical data counters.
	DMKIOFSV	Handles SEVERS from a virtual machine to the Error Logging System Service.
	DMKIOFVR	Records errors when requested by SVC 76.
DMKIOG		Pageable. Called at initialization to locate the error recording device, locate the last outboard error record and system recordings made on the cylinders, and set the in-storage pointers to the correct values. Initialization for RMS functions is performed after first making a test to determine if CP is running under CP. RMS functions are not activated for a virtual CP environment. This module also erases the recording areas.
	DMKIOGF1	Contains all function of DMKIOG except erase.
	DMKIOGF2	Erases (1) error records or (2) error records and frame records from the error recording cylinders, depending on input parameters.
DMKIOH		Pageable Called at initialization if the system error recording area requires formatting, or called while processing a CPEREP CLEARF request. Reads and formats machine check and channel check frames obtained from an SRF device (7443). Recognizes the presence of multiple SRF devices in an attached processor environment. Attempts to read frames from each available SRF device that was generated and format the respective frames at the beginning at the error recording cylinders.
	DMKIOHFR	For 3031/3032/3033 processors, reads frames from SRF devices, formats them in 4096-byte blocks, and writes the records to the error recording cylinders. The appropriate CPU id data is stored in the header portion of each frame record formatted by DMKIOHFR.
DMKIOJ		Resident.
	DMKIOJBL	Builds and formats outboard and miscellaneous data records.
DMKIOQ		Resident. IOBLOK queue handler and device path finder.
	DMKIOQFC	Finds a channel path to the device.
	DMKIOQFP	Finds a path to the device.
	DMKIOQFX	Finds a fixed path to the device.
	DMKIOQQD	Queues an IOBLOK on a real device queue.
	DMKIOQUS	Queues an IOBLOK on a real control unit queue.
	DMKIOQSK	Queues an IOBLOK on a real channel queue.
	DMKIOQDE	Dequeues the next IOBLOK from a real device queue.
	DMKIOQDU	Dequeues the next IOBLOK from a real control unit queue.
	DMKIOQDH	Dequeues the next IOBLOK from a real channel queue.
	DMKIOQHU	Queues an IOBLOK first an a real channel queue.
	DMKIOQDQ	Subroutine to dequeue mini-IOBLOKs from RBLOKs and unchain mini-IOBLOKs from other IOBLOKs.
	DMKIOQQU	Subroutine to queue control unit busy IOBLOK.

Module Name	Entry Points	Attributes, Function
DMKIOS		Resident. Schedules requests for virtual machine and program I/O operations.
	DMKIOSEN	Handles sense operations after a unit check.
	DMKIOSER	Error recovery processing.
	DMKIOSHA	Halts an active device and drains all interruptions.
	DMKIOSQR	Schedules CP-generated I/O operation.
	DMKIOSQV	Schedules a virtual machine I/O operation.
	DMKIOSQE	Reschedules an I/O operation after an I/O error on the channel.
	DMKIOSRC	Error recording.
	DMKIOSRH	Restarts channel.
	DMKIOSRQ	Requeues request.
	DMKIOSRS	Restarts device.
	DMKIOSRU	Restart control unit - channel.
	DMKIOSRW	Processes the IOBLOK used for REWIND.
	DMKIOSC1	Handles a deferred condition code 1 interrupt.
	DMKIOSC3	Handles a deferred condition code 3 interrupt.
	DMKIOSST	Issues SIO.
DMKIOT		Resident.
	DMKIOTIN	Processes all I/O interrupts.
	DMKIOTRC	Processes all pseudo I/O interrupts generated by DMKACRCV.
DMKISM		Pageable.
	DMKISMTR	Finds and modifies an ISAM CCW string.
DMKIUA		Pageable.
	DMKIUAEP	Prepare to handle user IUCV request.
	DMKIUACP	Prepare to handle CP IUCV request on behalf of the system.
	DMKIUACU	Prepare to handle CP IUCV request on behalf of a virtual machine.
	DMKIUARF	Reflect IUCV message to a virtual machine.
	DMKIUAPD	Validate an IUCV communications path.
	DMKIUAQU	IUCV message queue routine.
DMKIUB		Pageable.
	DMKIUBRK	Reflects messages or replies to a virtual machine.
	DMKIUBTB	Entry to return CP service entry points.
DMKIUC		Pageable.
	DMKIUCEP	Process request for IUCV functions (Accept, Connect, Declare Buffer, Quiesce, Query, Resume, and Retrieve Buffer).
DMKIUE		Pageable.
	DMKIUEEP	Process request for IUCV functions (Send, Describe, Receive, Reply, Test Completion, Set Control Mask, Set Mask, and Test Message).
	DMKIUEIN	IUCV external interrupt queuing routine.
DMKIUG		Pageable.
	DMKIUGER	Processes request for IUCV functions (Reject and Purge).
	DMKIUGGP	IUCV routine to obtain a communications path.
DMKIUJ		Pageable.
	DMKIUJEP	Processes request for IUCV function (Sever).
DMKIUL		Pageable.
	DMKIULEP	Processes request for IUCV function (Reply and Test Completion).

Module Name	Entry Points	Attributes, Function
DMKIUN		Pageable.
	DMKIUNEP	Handles IUCV functions.
	DMKIUNIN	IUCV external interrupt queueing routine.
DMKIUP		Pageable.
	DMKIUPEP	Services request for IUCV functions.
DMKIUS		Pageable.
	DMKIUSEP	Handles IUCV functions.
	DMKIUSET	Sets states for APPC/VM functions.
DMKJRL		Pageable.
	DMKJRLQU	Processes the QUERY command.
	DMKJRLSE	Processes the SET JOURNAL command.
	DMKJRLLO	Processes LOGONs with invalid passwords.
	DMKJRLSL	Processes LINKs which are successful.
	DMKJRLIL	Processes LINKs with invalid passwords.
DMKLD00E		Loader - utility program.
	LDRGEN	Loads assembled program modules into storage at locations other than those assigned by the assembler. It completes linkage among the modules and transfers control to one of the loaded modules for execution.
DMKLNK		Pageable.
	DMKLNKIN	Links to a virtual DASD because of an issued LINK command.
	DMKLNKSB	LINK subroutines.
DMKLNM		Pageable.
	DMKLNMSG	Processes error and response messages for the LINK command.
DMKLOC		Resident.
	DMKLOCK	Allows a system resource to be marked in use or not available by a unique 8-character name.
	DMKLOCKD	Dequeues a locked name.
	DMKLOCKQ	Queues or locks a name.
	DMKLOCKT	Tests to determine if a name is locked.
DMKLOG		Pageable.
		Logs on a user or operator.
	DMKLOGA	Processes the AUTOLOG command.
	DMKLOGON	Logs on a user.
	DMKLOGOP	Logs on the operator.
DMKLOH		Pageable.
	DMKLOHRC	Updates VMBLOK to LOGON a user or to RECONNECT a user.
DMKLOJ		Pageable.
		Logs on a user.
	DMKLOJEP	Creates virtual devices for the virtual machine being logged on.
DMKLOK		Resident.
	DMKLOK	handles all locking requests when CP is in attached processor mode.
	DMKLOKDF	Processes an obtain, defer lock request.
	DMKLOKPS	Processes all spin lock requests.
	DMKLOKSO	Processes an obtain, defer request for VMBLOK lock.
	DMKLOKSP	Processes an obtain request for a spin lock that previously failed.
	DMKLOKVM	Processes an obtain, defer request for VMBLOK lock.
	DMKLOKVR	Processes a release request for VMBLOK lock.

Module Name	Entry Points	Attributes, Function
DMKLOM		Pageable.
	DMKLOMSG	Constructs and sends logon-related messages to a user or to the operator.
	DMKLOMSS	Handles the allocation of a MSS disk after the volume has been mounted.
DMKMCC		Pageable.
	DMKMCCCL	Handles first level MONITOR command processing.
DMKMCD		Pageable.
	DMKMCDIN	Processes MONITOR INTERVAL commands.
	DMKMCDLI	Processes MONITOR LIMIT commands.
	DMKMCDTI	Processes MONITOR TIME commands.
	DMKMCDST	Processes MONITOR STOP commands.
	DMKMCDSE	Processes MONITOR SEEKS commands.
DMKMCH		Resident.
	DMKMCHIN	Processes a machine check interruption.
DMKMCI		Pageable.
	DMKMCIMS	Enables or disables soft machine check recording.
DMKMCT		Resident. This module is called by the machine check handler in attached processor mode.
	DMKMCTFS	Handles unsuccessful SIGP recovery.
	DMKMCTMA	Handles malfunction alert.
	DMKMCTPR	Handles processor recovery.
	DMKMCTPT	Handles processor termination.
	DMKMCTST	Handles system termination.
DMKMES		Pageable, nonexecutable. This is the CP message repository. It contains most CP messages. Module DMKERM references this message repository to write out messages.
DMKMHC		Resident.
	DMKMHCIN	Processes MSSFCALL and Service Call external interrupts.
	DMKMHCRE	Releases a virtual machine HCBLOK at logoff.
	DMKMHCCP	Handles MSSF service requests and Service Call requests for CP.
	DMKMHCVM	Handles MSSF service requests for a virtual machine.
DMKMHV		Pageable.
	DMKMHVSM	Simulates DIAGNOSE X'0080' (MSSFCALL) or Service Call for a virtual machine.
DMKMIA		Pageable. Provides various facilities associated with automatic monitoring using spool files.
	DMKMIACC	Used for MONITOR CLOSE processing.
	DMKMIADL	Used for DMKMCC display function.
	DMKMIAEN	Used to invoke a MONITOR STOP command.
	DMKMIAIN	Used to invoke a MONITOR START command.
	DMKMIAKC	Used to invoke a MONITOR CLOSE command.
	DMKMIAMU	Generates informational messages for monitor user.
	DMKMIARO	Opens monitor spool file, gets SFB, etc.
	DMKMIAWO	Writes a monitor data buffer to a spool file buffer.

Module Name	Entry Points	Attributes, Function
DMKMID		Pageable.
	DMKMIDNT	Changes the date in the system low storage at midnight and resets the clock comparator for the next midnight occurrence. DMKMID also sends messages to all users about the date change.
DMKMNI		Pageable.
	DMKMNIDK	Constructs spool file header record.
	DMKMNIFI	Completes monitor shutdown.
	DMKMNISH	Initializes MONITOR shutdown.
	DMKMNIST	Processes MONITOR AUTO STOP/START command.
	DMKMNITH	Handles monitor tape header processing.
	DMKMNITR	Writes the MONITOR trailer record.
DMKMNJ		Pageable.
	DMKMNJDS	Displays automatic monitoring information defined by SYSMON macro in DMKSYS.
	DMKMNJGT	Initialize the monitor TRQBLOK for timer driven event sampling.
	DMKMNJSP	Handles monitor processing for SPOOL to USERID parameters of START command.
DMKMNL		Pageable.
	DMKMNLIN	Performs initialization functions required for 3880 storage control monitoring.
	DMKMNLCP	Reads and stores the subsystem status by issuing a Sense Subsystem Status SIO, and stacks a TRQBLOK to call DMKMNLTQ.
	DMKMNLTQ	Performs a Sense Subsystem Counts SIO.
	DMKMNLMR	Writes the latest subsystem counts to the monitor.
	DMKMNLFI	Terminates 3880 storage control monitoring.
DMKMNT		Pageable.
	DMKMNTIO	Marks the I/O devices online during system initialization.
DMKMON		Pageable.
		Processes commands and requests associated with the MONITOR, including MONITOR CALL interruptions within CP.
	DMKMONIO	Processes tape interruptions returned by DMKIOS.
	DMKMONMI	Processes a MONITOR CALL program interruption.
	DMKMONPR	Gets space for monitor record and manages buffers.
DMKMOO		Pageable.
		Paged in and locked when MONITOR START command is issued.
	DMKMOO00	Handles PERFORM (class 0) data collection routine.
	DMKMOO40	Handles USER (class four) data collection routine.
	DMKMOOTI	Handle timer request interruptions.
DMKMPO		Pageable.
	DMKMPOEX	Processes SPMODE multiprocessor external interrupts.
	DMKMPOPX	Simulates SPX instructions.
	DMKMPORS	Processes SPMODE restart interrupts.
	DMKMPOSP	Simulates SIGP instructions.

Module Name	Entry Points	Attributes, Function
DMKMSG		Pageable. Transmits messages to logged-on users for the MESSAGE, SMSG, or WARNING commands. Receives and retransmits lines for the ECHO command for the number of times specified.
	DMKMSGEC	ECHO command processor.
	DMKMSGMS	MESSAGE command processor.
	DMKMSGNH	MSGNOH command processor.
	DMKMSGSM	SMSG command processor.
	DMKMSGWN	WARNING command processor.
DMKMSW		Resident.
	DMKMSWR	Allows system communication with the operator for the enhancement of error recovery procedures.
DMKNEA		Pageable.
	DMKNEAAH	Process the network ATTACH function.
	DMKNEADF	Access the RDEVBLOK and NICBLOK.
	DMKNEADH	Process the network DETACH function.
	DMKNEAVT	Access the RDEVBLOK and NICBLOK to determine if the virtual address is already attached.
DMKNEM		Pageable.
	DMKNEMOP	Gets a 5-byte mnemonic opcode for a System/370 binary opcode.
DMKNES		Pageable. Processes NETWORK operands as follows: POLLDLAY SHUTDOWN VARY
	DMKNESEP	Processes the NETWORK VARY EP command to switch an NCP communication line to EP mode.
	DMKNESHD	Processes the NETWORK SHUTDOWN command.
	DMKNESPL	Processes the NETWORK POLLDLAY command.
	DMKNESWN	Processes the NETWORK VARY NCP command to switch an EP communication line to NCP mode.
DMKNET		Pageable. Decodes NETWORK command and enables bisync lines.
	DMKNETAE	Enable binary synchronous lines and remote stations.
	DMKNETDF	Obtains addresses of the RDEVBLOK and NICBLOK for remote devices being defined in DMKVDSDF.
	DMKNETWK	NETWORK command decoder.
DMKNLD		Pageable.
	DMKNLDR	Loads the 3705 network control program. These routines may be called by a console command from DMKNET or internally by DMKCPI (for LOAD) or DMKRNH (for DUMP).
DMKNLE		Pageable.
	DMKNLEMP	Dump the 3705 Network Control Program.
DMKNMT		Pageable.
	DMKNMTBL	Construct an Image Library for TEXT files on user disks.
DMKOPE		Pageable.
	DMKOPERC	Locates the operator's console during system initialization.
	DMKOPEEM	Returns from DMKCNS for entry DMKOPERC.
	DMKOPELO	Logs on the system operator during system initialization.
	DMKOPEDC	Disconnects the system operator during system initialization.
	DMKOPEAC	Sets up auto monitor and logs on the AUTOLOG1 user's virtual machine during system initialization.

Module Name	Entry Points	Attributes, Function
DMKOPR		Resident.
	DMKOPRWT	Provides the necessary support for the CP system console. Certain routines within the control program cannot call DMKQCN to issue writes to the system console. This module determines the system's primary console and builds a channel program to handle the requested call.
DMKOVR		Stand-alone. Uses preallocated cylinders to build a Command Class Override file on a specified DASD volume. DMKOVRDE This is the override entry point.
DMKPAG		Resident.
	DMKPAGIO	Constructs IOBLOKs and schedules the tasks that move virtual storage pages between auxiliary storage and main storage. It also calculates the total system paging load at user-specified intervals.
DMKPAH		Resident.
	DMKPAHIO	Handles paging I/O and swapping I/O interrupts.
DMKPEI		Pageable.
	DMKPEINT	Creates a PEXBLOK chain for the PER command.
DMKPEL		Pageable.
	DMKPELSD	Completes handling of the CP PER command.
	DMKPELCR	Computes PER control registers 9, 10, and 11.
	DMKPELCH	Changes/Merges two PEXBLOK chains.
DMKPEN		Pageable.
	DMKPENDA	Ends all CP PER tracing activity.
	DMKPEND	Handles the PER END subcommand.
	DMKPENSV	Handles the PER SAVE subcommand.
	DMKPENGT	Handles the PER GET subcommand.
DMKPEQ		Pageable.
	DMKPEQRY	Handles the QUERY PER command.
DMKPER		Pageable.
	DMKPERIL	Handles PER program interrupts.
DMKPET		Pageable.
	DMKPETAL	Produces an instruction display for the PER event.
	DMKPETAB	Produces a display of the TRACEBACK table.
DMKPGM		Pageable.
	DMKPGMEP	Invoked by the dispatcher if the dispatcher has a CPEXBLOK stacked for page/swap set migration. It migrates pages from storage levels corresponding to high-speed devices (PP, PG if necessary, SW), to lower levels. For swap set migration, DMKSWMIG is called.
DMKPGS		Pageable. Releases the pages of a user's virtual storage space, both from real storage and auxiliary DASD. Also locates and releases a named system which resides in the user's virtual storage.
	DMKPGSPO	Releases a user's entire virtual storage.
	DMKPGSPP	Releases only a specified amount of virtual storage.
	DMKPGSPR	Calls DMKPTSPW to ensure that the user is not in page wait. Then it releases and unlocks a specified amount virtual storage.
	DMKPGSPS	Releases a named system from a user's virtual storage.
	DMKPGSSS	Releases virtual storage but bypasses any virtual storage that contains a named system.

Module Name	Entry Points	Attributes, Function
DMKPGT		Resident. Allocates DASD pages (slots) that are used either for virtual storage paging or for spool file page buffers.
	DMKPGTCG	Allocates contiguous DASD pages to contain 370X dump spool file.
	DMKPGTDG	Allocates contiguous DASD pages to contain a system dump file that was previously spooled to tape.
	DMKPGTDT	Allocates contiguous DASD pages for system dump.
	DMKPGTGC	Creates prototype RECBLOK (for DMKVDG).
	DMKPGTPG	Allocates one DASD or Paging Storage page for system paging.
	DMKPGTPM	Allocates one DASD or one swap set page for page migration.
	DMKPGTSG	Allocates one DASD or Paging Storage page for system spooling.
	DMKPGTSW	Allocates contiguous DASD pages for swapping.
DMKPGU		Resident. Performs DASD storage management.
	DMKPGUAL	Locates the ALOCBLOK for a given DASD or Paging Storage slot address
	DMKPGUDU	De-allocates contiguous DASD space for system dump.
	DMKPGUPR	Releases DASD storage used for virtual storage paging.
	DMKPGUSD	Releases one page of DASD storage used for virtual spooling.
	DMKPGUSP	Releases one page of DASD storage used for virtual paging.
	DMKPGUVG	Allocates a page of virtual storage belonging to the CP paging VMBLOK.
	DMKPGUVR	Releases a virtual storage page.
	DMKPGUPP	Deallocates DASD slots that were formerly used for pages that are now in real storage and about to be swapped out.
	DMKPGUSR	Releases DASD pages belonging to a spool file that is no longer needed.
	DMKPGUSW	Deallocates DASD slots that were used for swapped pages.
DMKPIA		Pageable.
	DMKPIALD	Contains the *FOB* buffer images that the *LOADBUF* command uses to load the Font Offset Buffer of the 3289 Model 4 printer. This module does not contain executable code.
DMKPIB		Pageable.
	DMKPIBLD	Contains the buffer images that the LOADBUF command uses to load the UCSB of the IBM 3262 Printer, Models 1 and 11. This module does not contain executable code.
DMKPMA		Resident.
	DMKPMACD	Tests virtual device restrictions.
	DMKPMAEX	Terminates preferred machine assist for the virtual machine.
	DMKPMAI1	Initializes PSA and builds control register 2 channel masks.
	DMKPMAI2	Initializes fields for privileged operation simulation.
	DMKPMAIN	Processes PMA program interruptions.
	DMKPMAMC	Routes external damage machine checks to PMA virtual machine.
	DMKPMASS	Saves PMA environment and establishes virtual machine environment.
	DMKPMASW	Dispatches preferred virtual machine. Saves data for interruption handlers.
	DMKPMAWT	Processes when PMA guest loads a wait state PSW.
	DMKPMAXF	Processes extended storage fetch for PMA virtual machine.
	DMKPMAXS	Processes extended storage stores for PMA virtual machine.

Module Name	Entry Points	Attributes, Function
DMKPRG		Resident.
	DMKPRGIN	Processes a hardware program interruption.
	DMKPRGRF	Reflects an SVC interruption to the virtual machine.
	DMKPRGSM	Simulates a virtual program interruption.
DMKPRV		Resident.
	DMKPRVLG	Simulates a privileged operation.
DMKPRW		Resident.
	DMKPRWIP	Simulates IPTE instruction.
	DMKPRWTP	Simulates TPROT instruction.
	DMKPRWTB	Simulates TEST BLOCK instruction.
	DMKPRWSK	A flag that indicates that the extended-key instructions should be simulated.
	DMKPRWXK	Simulates ISKE, RRBE, and SSKE instructions. For guest operating systems with single key virtual storage, simulates ISK, SSK, and RRB instructions. (XKEYMODE must be set to B'1').
DMKPSA		Resident.
	DMKPSACG	Charges accumulated time to a virtual machine.
	DMKPSADU	PSW restart processing. Forces an SVC 0 type of dump.
	DMKPSAFC	Checks fetch protection per the CAW key.
	DMKPSAFP	Checks for fetch protection violation per PSW key.
	DMKPSAID	Gets virtual address for any instruction.
	DMKPSAPO	Returns the real storage location of virtual page 0 for a virtual machine if that page is in storage, or returns a nonzero return code.
	DMKPSARR	Gets the virtual address for an RR instruction.
	DMKPSARS	Gets the virtual address for RS, SI, or SS instruction.
	DMKPSARX	Gets the virtual address for an RX instruction.
	DMKPSASC	Checks storage protection per the CAW key.
	DMKPSASP	Checks for a storage protection violation per the PSW key.
DMKPST		Pageable.
	DMKPSTIN	Initialize Paging Storage ALOCBLOKs and RECBLOKs.
DMKPTR		Resident. Manages the inventory of real system pages, provides real storage space for CP functions and for pages of user and CP virtual storage.
	DMKPTRAN	Translates user virtual storage address to a real storage address.
	DMKPTRAQ	Performs working set calculations each time a page is acquired in real storage.
	DMKPTRCO	Obtains a page of real storage for cache-aligned pools.
	DMKPTREP	Obtains a page from >16Mb or <16Mb free list, (whichever is larger).
	DMKPTRFP	Resumes extend processing once a page frame has been obtained.
	DMKPTRFR	Gets a page of real storage.
	DMKPTRLK	Locks a page of real storage and processes the CP assist instruction, PTRLK (E602).
	DMKPTRPS	Determines swap table and page table addresses for a given virtual address.
	DMKPTRUL	Unlocks a page of real storage and processes the CP assist instruction, PTRUL (E603).

Module Name	Entry Points	Attributes, Function
DMKPTS		Resident.
	DMKPTSAD	Decodes a page frame entry into a 26-bit real address.
	DMKPTSAE	Encodes a 26-bit real address into a page frame table entry.
	DMKPTSPW	Called to defer execution of system reset functions when user's virtual machine is in page wait.
	DMKPTSRS	Resets pages belonging to a user.
DMKPTT		Resident.
	DMKPTTAL	Allocates virtual page slots that DMKPTTPM used for page moves between the less than 16 Mb dynamic paging area and the greater than 16 Mb dynamic paging area. This function is invoked when extended storage is initially brought online, or when the V=R area is unlocked and the greater than 16 Mb page frames are initialized for the first time.
	DMKPTTCL	Clears a 4K page of extended storage.
	DMKPTTFT	Releases a page of real storage.
	DMKPTTPM	Moves a 4K page of data from one page frame to another, regardless of whether the page is in the extended storage area.
DMKPXA		Resident, nonexecutable. Used as an extension to the PSA.
DMKPXB		Resident, nonexecutable. Used as an extension to the PSA.
DMKQCN		Resident.
	DMKQCNWT	Starts and queues a console write request. If a buffer larger than 4K bytes is required, an indirect address list is used to address the noncontiguous buffer.
DMKQCO		Resident.
	DMKQCOCL	Clears CONTASK stack and returns all blocks to free storage.
	DMKQCOET	Processes completed CONTASKs for virtual console spooling, return or no return options, and returns the CONTASK blocks to free storage.
	DMKQCONQ	Queues up CONTASK.
	DMKQCORD	Start and queues a console Read request.
	DMKQCOSY	Synchronizes virtual machine console activity with internal supervisor activity (used during virtual system reset and logoff).
DMKQCP		Pageable.
	DMKQCPTO	Disconnects the virtual machine. Sets the TOD clock comparator request to logoff the virtual machine after a 15-minute delay.
DMKQCQ		Resident.
	DMKQCQED	Edits data streams for console writes.
DMKQVM		Resident.
	DMKQVMEP	Process the CP QVM command.
	DMKQVMRT	Performs the switch to native mode for the V=R user.
	DMKQVMRS	Performs the switch back to VM/370 for non-370E operating system.
	DMKQVMRX	Performs the switch back to VM/370 for 370E operating system.
DMKREI		Pageable.
	DMKREIPL	Re-IPLs a user of the Protected Application Facility.

Module Name	Entry Points	Attributes, Function
DMKRET		Resident.
	DMKRETGT	Gets an input line from the retrieve buffer.
	DMKRETPT	Puts an input line into the retrieve buffer.
DMKRGA		Resident.
	DMKRGAIN	This is the second-level interruption handler for remote 3270 stations. This module supports the 3270 remote display and printer stations. It processes interruptions and CCWs for the remote stations, including message handling and screen management.
	DMKRGATM	Processes time interrupts for the following 3270 display conditions: - completion of poll delay period - completion of 60 second delay on priority messages - more timeout - 3 second 'Not Accept' message
	DMKRGASP	Complete processing full screen WSF and QUERY
	DMKRGA2	If a buffer larger than 4K bytes is required, an indirect address list is used to address the noncontiguous buffer.
DMKRGB		Resident. Supports the 3270 remote display and printer stations. It processes interruptions and CCWs for the remote stations including message handling and screen management.
	DMKRGBCL	Clears the display screen after full screen I/O.
	DMKRGBCO	Continues output operations on a currently selected remote station.
	DMKRGBIC	Initializes and schedules CONTASKs.
	DMKRGBEN	Enables and disables bisync lines and remote stations.
	DMKRGBSN	Screens NICBLOK list for output messages; does general poll if none found.
	DMKRGBMT	Formats the display screen.
	DMKRGBPL	Issues a specific or general poll.
	DMKRGBRE	Starts I/O on a teleprocessing line.
	DMKRGBSL	Selects a remote station.
	DMKRGBUP	Updates a remote 3270 screen.
DMKRGC		Resident. Processes 3270 input data and status messages after validation by DMKRGA.
DMKRGD		Resident.
	DMKRGDOB	Performs the blocking of output data to display terminals and updates the CONTASK. If a buffer larger than 4K bytes is required, an indirect address list is used to address the noncontiguous buffer.
	DMKRGDOI	Performs the blocking of output data to display terminals and updates the CONTASK. If a buffer larger than 4K bytes is required, an indirect address list is used to address the noncontiguous buffer.
DMKRGE		Resident.
	DMKRGESK	Handles skip processing for BSC and VM/SNA devices.
DMKRIO		Resident.
	DMKRIO	Exists as a CSECT and defines the machine's configuration. A basic DMKRIO is shipped with the system. DMKRIO can be changed at system generation or whenever new machines are added by using the appropriate macros.
DMKRND		Residency not applicable. Invoked via the NCPDUMP command in CMS.
	DMKRND	This is the interface between the dump spool file and the OS-SSP dump format program for printing and formatting dumps of the 3704 and 3705 communications controllers.

Module Name	Entry Points	Attributes, Function
DMKRNH		Resident.
	DMKRNHIC	Initializes and schedules the CONTASK fields that comprise the 3704 and 3705 Network Control Program transmission header.
	DMKRNHIN	This is the secondary interruption handler for the 3704 and 3705 communication controllers it is read when operating in NCP or PEP mode.
	DMKRNHND	Schedules control functions for the 3705 or 3704 Network Control Program.
DMKRPA		Resident. Virtual storage mapping.
	DMKRPAGT	Page-in from DASD to user's virtual storage.
	DMKRPAPT	Page-out to DASD from user's virtual storage.
DMKRPD		Pageable.
	DMKRPDEP	Process security DIAGNOSE X'A0' instruction.
DMKRPI		Pageable.
	DMKRPICN	Process IUCV connect request.
	DMKRPIRA	Process access verification.
DMKRPW		Pageable.
	DMKRPWEP	Process password verification.
DMKRSE		Pageable. Real UR device I/O error handler.
	DMKRSERR	Retries and attempts to recover from real unit record device I/O errors.
DMKRSF		Pageable. Real UR device I/O error handler.
	DMKRSFPR	Gets 3211 type printers error information.
	DMKRSFSD	Formats 3800 printer hardware environmental counters.
	DMKRSFPB	Purges 3800 page buffer on a 3800 Printing Subsystem when called due to an error.
DMKRSP		Resident. Manages all spooling operations on the real system unit record devices including printing and punching user-created spool files and reading and queueing reader files from the real card reader.
	DMKRSPER	Processes spooling errors (ERP).
	DMKRSPEX	Processes spooling operations. Entered via a GOTO when DMKDSPCH unstacks an IOBLOK with an interruption for the spooling unit record device.
DMKRSQ		Pageable. Handles the spool file buffers when data chaining between DASD buffer is required for example, for 3800 Load Graphic Modification or Load Copy Modification CCW.
	DMKRSQDC	Obtains buffers needed for data chaining and after the associated CCWs.
	DMKRSQFR	Free the buffers obtained by DMKRSQDC.
DMKRST		Pageable. Handles operations on the real system unit record card readers.
	DMKRSTIN	Processes an interrupt from a real card reader.
DMKSAD		Stand-alone.
	DMKSADM	Produces a stand-alone dump of real storage on a tape or printer.
	DMKSADWT	Utility that writes the stand-alone dump program on the IPL volume.

Module Name	Entry Points	Attributes, Function
DMKSAV		Pageable. DMKSAVNC is entered via an LDT card from DMKLDR. DMKSAVRS is entered via a BALR from DMKCKP. DMKSAV saves and restores a page image count of the CP nucleus on the system residence disk.
	DMKSAVNC	Writes a page image copy of the CP nucleus.
	DMKSAVRS	Restores a page image copy of the CP nucleus.
DMKSBL		Pageable.
	DMKSBLTR	Creates a line of small block letters for DMKSEP.
DMKSCH		Resident. Maintains the run list, the dispatch list (true run list), and the eligible list. Also calculates projected working set sizes and deadline priorities, keeps statistics on processor use, and monitors favored execution users.
	DMKSCHDL	Alters a user's dispatching status.
	DMKSCHST	Establishes a clock comparator interrupt request.
	DMKSCHRT	Resets a clock comparator interrupt request.
	SWAPIN	Interrupt return address for the prepaging function.
DMKSCN		Resident. Scans module.
	DMKSCNAU	Searches the chain of VMBLOKs for one whose userid matches the one pointed to by register one.
	DMKSCNFD	Finds the next field in an input message buffer.
	DMKSCNEP	Determines if there is an online path from either processor.
	DMKSCNLI	Searches the logged-on virtual machines for any links to a specified minidisk. A link is any virtual device whose RDEVBLOK pointer and relocation factor match those specified.
	DMKSCNP	Finds the RCHBLOK and RCUBLOK that represents the next logical path to the device.
	DMKSCNPH	Calculates a bit mask defining the indicated device path.
	DMKSCNRA	Computes a full real device address (in CCU form) from the RDEVADD, RCUADD, and RCHADD entries in the real device, control unit, and channel blocks.
	DMKSCNRD	Computes a real device address (in CCU form), from the RDEVADD, RCUADD, and RCHADD entries in the real device, control unit, and channel blocks.
	DMKSCNRN	Returns the name of the real device to the caller in register 1.
	DMKSCNRU	Returns the addresses of the real channel, control unit, and device blocks for a given real device to the caller.
	DMKSCNVD	Computes a full virtual device address (in CCU form), plus the addresses of the virtual channel and control unit blocks from a specific virtual device block.
	DMKSCNVN	Returns the name of the virtual device to the caller in R1.
	DMKSCNVS	Searches all the real device blocks for a device whose volume serial number matches the one pointed to by R1.
	DMKSCNVU	Returns the addresses of the virtual channel, control unit, and device blocks for a given real device to the caller.
	DMKSCNDC	Returns the addresses of the RDEVBLOK that is given the device code (the *D* of *CCPD* or *PPPD*).
DMKSCO		Pageable.
	DMKSCOLI	Searches the logged-on virtual machines for any links to a specified minidisk. A link is virtual device whose RDEVBLOK pointer and relocation factor match those specified.
	DMKSCONP	Finds the RCHBLOK and RCUBLOK that represent the next logical path to the device.

Module Name	Entry Points	Attributes, Function
DMKSEG		Pageable.
	DMKSEGPG	Initializes CP page, segment, core, and swap tables.
	DMKSEGWR	Creates page image copies of all the pageable modules between DMKSAV and DMKCKP.
DMKSEL		Resident Called when pages are needed to replenish the free list. Controls the order by which the system searches for free pages.
	DMKSELCT	Replenishes the free lists.
	DMKSELFE	Entered via a CPEXBLOK built by DMKPTRFR, on extend condition.
	DMKSELFD	Entered via a CPEXBLOK built by DMKPTRFR, after a page has been written.
	DMKSELSL	Entered via a CPEXBLOK built by DMKPTRFR, on deferred request.
DMKSEP		Pageable.
	DMKSEPSP	Prints and punches the respective output separators on real spooling devices.
	DMKSEPTL	Prints the trailer page.
DMKSEQ		Pageable.
	DMKSEQDA	Contains a data area.
	DMKSEQLA	Contains printer separator logo.
	DMKSEQSA	The area for the sequence number.
DMKSEV		Pageable but locked.
	DMKSEV70	Analyzes 2870 channel logout and sets appropriate bits in the ECSW field according to the results of analysis. It moves the channel logout to the check record.
DMKSFB		Pageable.
	DMKSFBNS	Process Diagnose Code X'D8'.
DMKSIX		Pageable but locked. Analyzes 2860 channel logout and sets appropriate bits in the ECSW field according to the results of analysis. It moves the channel logout to the check record.
DMKSNC		Pageable.
	DMKSNCP	Save a page-form version of a 3704/3705 network control program. The name of the network control program and the DASD location at which it is to be saved is defined in the CP module DMKSYS.
DMKSND		Pageable.
	DMKSNDNH	Processes the SEND command (which is used to send commands) and replies to disconnected virtual machines.
DMKSNT		Pageable.
	DMKSNTBL	This module is assembled by the installation system programmer. It describes the system to be saved via the SAVESYS command and to be initial program loaded by name. Shared segments may be specified. These segments consist of all reenterable code and no altering of this storage is allowed. There is no executable code in this module.
DMKSPC		Pageable. Used by PROFS to reduce the number of spool files used for electronic mail.
	DMKSPCEX	Determines if spool file concatenation is installed.
	DMKSPCHF	Concatenates reader files.
	DMKSPCQC	Processes the QUERY CHAIN command.
	DMKSPCSY	Processes the SET CHAIN SYSTEM command.
	DMKSPCUS	Processes the SET CHAIN USER command.

Module Name	Entry Points	Attributes, Function
DMKSPK		Pageable.
	DMKSPKDL	Deletes used files from the system and deallocates the DASD page space.
	DMKSPKDR	Deletes all SFBLOKs in the DMKRSPDL chain and exits to the dispatcher.
DMKSPL		Pageable.
		Spool file manager.
	DMKSPLCR	Closes and queues a real reader spool file for virtual input.
	DMKSPLCV	Closes and queues a virtual printer or punch spool file for processing.
	DMKSPLOR	Initializes control blocks and buffers for real input reader files.
	DMKSPLOV	Initializes control blocks and buffers for virtual printer and punch output spool files.
DMKSPM		Pageable.
	DMKSPMEP	Processes the CP SPMODE command. Turns the single processor mode environment on and off.
DMKSPR		Resident.
	DMKSPROT	Processes requests for new tape for SPTAPE DUMP.
DMKSPS		Pageable.
	DMKSPSIO	Performs the processing requested by the SPTAPE command and handles all returns from the I/O interrupt handler due to the command.
DMKSPT		Pageable
	DMKSPTEP	Validates the format of the SPTAPE command and initiates the processing to write, read, or scan a tape for specified printer and punch spool files.
DMKSRM		Pageable.
	DMKSRMEP	Sets and displays system performance indicators.
DMKSSP		This module is found in the starter system only. It builds RCHBLOKs, RCUBLOKs, and RDEVBLOKs necessary to configure a minimum CP system. From the starter system, a real CP system figured based on the REALIO deck of the installation.
	DMKSSP01	Entered as a result of an IPL operation. Constructs the I/O blocks and system modules for a minimum system configuration.

Module Name	Entry Points	Attributes, Function
DMKSSS		Resident. Services routines for all other modules that require access to the MSS.
	DMKSSSAS	Attaches a 3330V to the system with a VOLID.
	DMKSSSCA	The communicator device address.
	DMKSSSCV	The VMBLOK address of the communicator virtual machine.
	DMKSSSDE	Demounts an MSS volume from a 3330V.
	DMKSSSEN	Returns to the appropriate requesting routine after an MSS volume mount is complete.
	DMKSSSHR	The CCPD of the SDG table containing shared VUAs.
	DMKSSSL1	Processes a DEDICATE statement with the 3330V parameter.
	DMKSSSL2	Processes a DEDICATE statement with raddr and volid specified, when the raddr is a 3330V.
	DMKSSSL3	Processes a DEDICATE statement with a volid but no raddr.
	DMKSSSLN	Allocates a 3330V device and mounts the required 3330V system volume.
	DMKSSSMQ	Serves as the anchor for the MSSCOM control blocks that are queued for MSS mounts, demounts, and pack change interrupts. Does not contain executable code.
	DMKSSSNS	The CCPD of the SDG table containing nonshared VUAs.
	DMKSSSNV	The anchor of the SDG in which a VUA was last selected for a mount request.
	DMKSSSRL	Issues a relinquish request to destage any changed cylinders of the volume mounted on the specified VUAs.
	DMKSSSVA	Attaches a 3330V to the system or a virtual machine.
	DMKSSSVM	The userid of the communicator virtual machine.
DMKSST		Pageable. This is a service routine for modules that require access to MSS.
	DMKSSTBL	Builds SDG tables of VUAs in CP configuration.
	DMKSSTFV	Finds an available VUA on which to mount a virtual volume.
DMKSSU		Resident. Handles service requests for processing whenever an attention interrupt or a cylinder fault is detected on a 3330V and whenever a reset is required for a virtual device defined on a 3330V.
	DMKSSUCF	Resets a virtual device defined on a 3330V, including purging any I/O waiting for an MSS volume mount.
	DMKSSUI1	Reschedules an I/O operation that had previously caused a cylinder fault.
	DMKSSUI2	Queues an I/O request that has just caused a cylinder fault. Sets the missing attention handler timer interrupt value.
	DMKSSULO	Checks for unfinished MSS processing before completing logoff.
DMKSSV		Pageable. This is a service routine for the MSS communicator virtual machine.
	DMKSSVHV	Process DIAGNOSE Code X'78'.
	DMKSSVUS	Quiesces all MSS mount and demount activity.
DMKSTA		Pageable.
	DMKSTANT	Clears main storage, initializes the CORTABLE, and allocates main storage.

Module Name	Entry Points	Attributes, Function
DMKSTD		Resident, nonexecutable.
	DMKSTDAT	Starting address of STDATA table.
DMKSTK		Resident. Stacks I/O blocks.
	DMKSTKCP	Stacks a CPEXBLOK.
	DMKSTKDE	Stacks a deferred execution block.
	DMKSTKIO	Stacks an IOBLOK.
	DMKSTKLF	Stacks a CPEXBLOK LIFO (used by EXTEND and machine check).
	DMKSTKMP	Stacks CPEXBLOK for current processor only.
	DMKSTKOP	Stacks CPEXBLOK for the other processor only.
DMKSTP		Pageable.
	DMKSTPVP	Initializes the system performance indicators and scheduling control fields when a processor is varied online.
	DMKSTPX	Updates the system performance indicators and scheduling control fields.
DMKSTR		Pageable.
	DMKSTRAN	Invoked from DMKPTR if there is a segment exception.
	DMKSTRPM	Invoked from DMKSTRAN or DMKSTRSM to process a pseudo page.
	DMKSTRSM	Invoked if the dispatcher has a CPEXBLOK for swap table migration.
DMKSVC		Resident.
	DMKSVCIN	Handles any SVC interrupt.
DMKSVD		Resident.
	DMKSVDIN	Handles problem state SVC interrupts.
DMKSWA		Resident. Controls logical swap-in and physical swap activities.
	DMKSWAPO	Called by DMKSEL to process a physical swap out.
	DMKSWAPD	Called after completion of swap out. For successful swap outs, it resets in transit and change bits for SWPTABLE entries and puts page frames on the free list. For unsuccessful swap outs, it calls DMKPGTSW to allocate a new swap slot.
	DMKSWAPI	Processes pre-paging requests from DMKSCH.
	DMKSWAPR	Invalidates a swap set block (SSBLOK) entry.
DMKSWM		Pageable.
	DMKSWMIG	Migrates swap sets from Paging Storage (TYPE = SW) to next available TYPE = SW level or, as a second choice, to the first available non-Paging Storage TYPE = PP level.
	DMKSWMUS	Invoked if page/swap set migration is invoked by the MIGRATE command. If the command is valid, a CPEXBLOK is stacked with an entry point of DMKPGMX to migrate all users, or an entry point of RESETUS to migrate one user.
DMKSYM		Pageable.
	DMKSYM	Provides a symbol table of selected CSECTS and entry points.
DMKSYS		Resident.
	DMKSYS	Exists as a CSECT that defines the system residence volume, paging space, operator ID, dump ID, storage size, and time zone.
	DMKSYSRM	Real storage size of the processor.

Module Name	Entry Points	Attributes, Function
DMKTAP		Pageable.
	DMKTAP	Examines the error condition resulting from a unit check while executing a CP generated tape channel program. Positioning of the tape is required on read/write commands and the channel program is reexecuted. If the error condition is uncorrectable, a call is issued to the message writer (DMKMSW) to notify the operator. Upon regaining control from DMKMSW, the original channel program may be reexecuted or terminated.
	DMKTAPER	Retries the failing tape channel program, after a tape positioning command has been executed.
	DMKTAPRL	Performs tape release to determine two- or four-channel switch capability.
DMKTAQ		Pageable. Continues tape error recovery started by DMKTAP.
	DMKTAQRP	Repositions tape following a read-type error.
	DMKTAQSE	Continues checking for the cause of the original device error.
	DMKTAQRE	Entered following a tape reposition operation.
DMKTBL		Resident.
	DMKTBL	Contains the terminal translate tables.
DMKTBM		Pageable, nonexecutable. Contains terminal translate tables for APL/ASCII for non-TTY terminals.
DMKTBN		Pageable, nonexecutable. Contains terminal translate tables for APL/ASCII for TTY terminals.
	DMKTBNAO	EBCDIC APL to ASCII APL translation.
	DMKTBNAI	ASCII APL to EBCDIC APL translation.
	DMKTBNAE	ASCII to EBCDIC translation.
	DMKTBNEA	EBCDIC to ASCII translation.
DMKTCS		Pageable.
	DMKTCSET	Sets up the 3800 prior to printing the file.
	DMKTCSSP	Sets up the 3800 prior to printing the separator.
	DMKTCSCO	Sets up the forms overlay sequence control.
	DMKTCSTR	Prints trailer page on a 3800 printer.
	DMKTCSML	Loads members from image library.
DMKTCT		Pageable.
	DMKTCTET	Loads character arrangement table, WCGM's, LCS's, and graphic character modifications into a 3800 printer.
DMKTDK		Pageable. Allocates and deallocates cylinders (CKD) or blocks (FBA) of TDISK space from CP-owned volumes.
	DMKTDKGT	Allocates TDISK space.
	DMKTDKRL	Deallocates TDISK space.
DMKTEE		Residency not applicable.
	DMKTEESF	Formats the subtype specific information of a CP trace entry. (This module contains the first half of the process. The second half is contained in DMKTEF.)
DMKTEF		Residency not applicable.
	DMKTEFSF	Formats the subtype specific information of a CP trace entry. (This module contains the second half of the process. The first half is contained in DMKTEE.)

Module Name	Entry Points	Attributes, Function
DMKTEM		Residency not applicable.
	DMKTEMEP	Formats a CP trace entry. (This is the main module for trace entry formatting. The support modules are DMKTEE, DMKTEF, and DMKTES.)
DMKTES		Residency not applicable.
	DMKTESUB	This module performs miscellaneous subroutines for the main trace entry formatter, (DMKTEM).
DMKTHI		Pageable. Displays data about use of and contention for major system resources.
	DMKTHIFA	Processes the INDICATE FAVORED command.
	DMKTHIIO	Processes the INDICATE I/O command.
	DMKTHILO	Processes the INDICATE LOAD command.
	DMKTHIPA	Processes the INDICATE PAGING command.
	DMKTHIQQ	Processes the INDICATE QUEUES command.
	DMKTHIUG	Processes the INDICATE USER command for general users.
	DMKTHIUS	Processes the INDICATE USER command for primary system operators, system resource operators, and system analysts.
DMKTMR		Resident. Simulates the CPU timer and time-of-day clock comparator instructions for virtual machines operating in EC mode.
	DMKTMRCC	Entered after expanded virtual machine assist processing of a virtual SCKC instruction.
	DMKTMRCK	Simulates virtual clock comparator interruptions.
	DMKTMRPT	Calculates user's total virtual problem time.
	DMKTMRSP	Entered after expanded virtual machine assist processing of a virtual SPT instruction.
	DMKTMRTN	Simulates timer instruction.
	DMKTMRVT	Simulates virtual CPU timer interruptions.
DMKTOD		Pageable.
	DMKTODIN	Initializes the time of day clock.
DMKTPE		Pageable.
	DMKTPERP	Performs 3480 error recovery processing.
DMKTRA		Pageable. Processes the TRACE command line. Provides a virtual machine with facility to track SVC instructions, program interrupts, external interrupts, successful searches, or all instructions with output on the printer or terminal.
	DMKTRACE	TRACE command processor.
DMKTRC		Pageable. Processes the TRACE command functions.
	DMKTRCEX	Traces external interruptions.
	DMKTRCIO	Traces I/O interruptions.
	DMKTRCIT	Sets the needed SVC B2 for instruction tracing.
	DMKTRCND	Ends tracing.
	DMKTRCPB	Puts back user instructions altered by tracing.
	DMKTRCPG	Traces program interruptions.
	DMKTRCPV	Traces privileged instruction interruptions.
	DMKTRCSV	Processes an SVC, Branch, or full instruction TRACE.
	DMKTRCSW	Traces virtual and real CSWs.
DMKTRD		Pageable. Split from DMKTRC.
	DMKTRDSI	Traces I/O operations (SIO, TIO, HIO, TCH).
	DMKTRDWT	Serialization entry for I/O and CCW tracing.

Module Name	Entry Points	Attributes, Function
DMKTRK		Resident.
	DMKTRKIN	Handle interrupts caused by alternate tracks.
	DMKTRKFP	Examine Command Rejects for virtual SIO.
	DMKTRKVA	Verify alternate track address for DMKCCW.
DMKTRM		Pageable.
	DMKTRMID	Identifies a 2741 terminal as either a 2741P (PTTC/EBCD) or 2741C (correspondence) from the user command. It sets ADEVTYPE the RDEVBLOK to TYP2741P or TYP2741C and sets flag RDEVIDNT on if the terminal was successfully identified.
DMKTRP		Pageable.
	DMKTRPRE	Stops CPTRAP processing due to LOGOFF.
	DMKTRPST	Processes the CPTRAP command line.
DMKTRQ		Resident.
		Processes TRQBLOKs and maintains TRQBLOK queues.
	DMKTRQCP	Processes TRQBLOKs for the CPU timer.
	DMKTRQIL	Processes TRQBLOKs for queue drop delay.
	DMKTRQMD	Processes TRQBLOKs for midnight time and date changes.
	DMKTRQRT	Resets a TOD clock timer request.
	DMKTRQST	Establishes a TOD clock timer request.
	DMKTRQTI	Processes the end of the performance interval.
	DMKTRQ80	Processes TRQBLOKs for real timers.
DMKTRR		Residency not applicable.
	DMKTRRST	Reduces the reader file created by the CPTRAP facility.
DMKTRT		Pageable.
	DMKTRTCM	Common processing routine to enter data into a CPTRAP file.
DMKTRU		Pageable.
	DMKTRUAC	Activates the CPTRAP facility.
	DMKTRUAC	Stops an active CPTRAP facility.
DMKTRX		Pageable.
	DMKTRXCP	Interface to CPTRAP facility, records data generated by CP code.
	DMKTRXEX	Exit routine for the interface entry points.
	DMKTRXTT	Interface to CPTRAP facility, records internal CP trace table entries.
	DMKTRXVT	Interface to CPTRAP facility, records data generated by virtual machine code.
DMKTTX		Pageable.
		Determines I/O operations to be performed on a certain device, constructs CCWs and data streams. See also DMKTTY.
	DMKTTXIN	Handles input contasks (reads).
	DMKTTXSK	Entry for translation routines.
DMKTTY		Pageable.
		Determines I/O operations to be performed on a certain device, constructs CCWs and data streams. See also DMKTTX.
	DMKTTYOP	Handles output contasks (writes) and outbound reads.
DMKTTZ		Resident, nonexecutable.
	DMKTTZLF	Contains CCWs and data pointed to by certain CCWs for TTY terminals.

Module Name	Entry Points	Attributes, Function
DMKUCB		Pageable.
	DMKUCB	Contains the UCB buffer load images used by the LOAD command to load the universal character set buffer in the 3811 control unit. This module contains no executable code.
DMKUCC		Pageable.
	DMKUCCLD	Contains the UCB buffer load images used by the LOAD command to load the universal character set buffer in the 3203 printer control unit. This module does not contain executable code.
DMKUCS		Pageable. Contains the UCS buffer load images that the LOAD command uses to load the universal character set buffer in the 2821 control unit. This module does not contain executable code.
DMKUDR		Pageable.
	DMKUDRBV	Allows the DMKDIRCT or DMKCPINT programs to build a list of virtual page buffers, one for each UDIRBLOK page on disk.
	DMKUDRDS	Allows the DMKDIRCT program to swap the active user directory to the newly created user directory.
	DMKUDRFD	Puts specified UDEVBLOK into the caller's buffer.
	DMKUDRFU	Finds a given user ID in the user directory and moves the user's directory entry into the caller's buffer.
	DMKUDRMD	Reads the account block.
	DMKUDRRD	Reads the next user directory into the caller's buffer.
	DMKUDRRV	Releases a virtual page used by the directory program as a buffer.
	DMKUDRXI	Reads the UIPLBLOK into the caller's buffer.
DMKUDU		Pageable.
	DMKUDUMN	Updates in-place the CP directory on the object DASD page and updates in-place the virtual system page (if used) on the paging device. Entered from DMKHVD when a class B virtual machine issues a DIAGNOSE code '84' instruction.
DMKUNT		Resident. Untranslates CCWs and CSWs.
	DMKUNTFR	Releases pages and free storage used for the CCW chain. Also processes the CP assist instruction, UNTFR (E605).
	DMKUNTIS	Finds the RCWTASKS that have been patched to handle OS ISAM self-modifying sequences and put them back the way DMKCCW had them to allow DMKUNTRN and DMKUNTFR to operate correctly.
	DMKUNTRN	Translates a real CSW into a virtual CSW. Also processes the CP assist instruction, UNTRN (E610).
	DMKUNTRS	Relocates sense byte information. For a 3330, 3340, 3350, 3375, or 2305, computes virtual cylinder member in byte 5 and 6 of the sense byte from data by unrelocating the real cylinder number given by the hardware.
	DMKUNTFB	Relocates sense-byte information for FBA devices. The virtual physical address from sense bytes 3-6 is computed from the real physical address given by the hardware. For formats 0 4, sense bytes 18-21 and for format 1 message A, bytes 8-10 are also computed from the real physical address given by the hardware.

Module Name	Entry Points	Attributes, Function
DMKURS		Pageable.
	DMKURSTA	Displays status messages for real unit record devices.
DMKUSO		Pageable.
		Processes user termination.
	DMKUSODS	Processes the DISCONN (disconnect) command.
	DMKUSOFF	Logs off a user. (Calls DMKUSQ.)
	DMKUSOFL	Processes the FORCE command.
DMKUSP		Pageable.
	DMKUSPFM	Maintains free storage subpools and free storage chain.
DMKUSQ		Pageable.
	DMKUSQFF	Continues LOGOFF processing. (Called by DMKUSO.)
DMKVAT		Resident.
		Storage management for EC mode virtual machine.
	DMKVATAB	Allocates, initializes and maintains shadow, segment, and page tables for virtual machines that can relocate.
	DMKVATBC	Returns active shadow tables to free storage.
	DMKVATEX	Services page or segment exceptions for virtual EC machines.
	DMKVATMD	Allocates and initializes shadow tables.
	DMKVATPF	Handles pseudo page fault interruption from a VS1 virtual machine.
	DMKVATPX	Processes paging exceptions for a virtual machine that performs paging.
	DMKVATSI	Selectively invalidate shadow page table entries.
	DMKVATSX	Processes segment exception for a virtual machine that performs paging.
	DMKVATZP	Processes the CP assist instruction, ZAPPAGE (E60B).
	DMKVATZS	Processes the CP assist instruction, ZAPSEGS (E60A).
DMKVAU		Resident.
	DMKVAULA	Virtual - virtual to virtual address translation.
	DMKVAURN	Virtual (shadow) — virtual-to-real address translation.
DMKVCA		Pageable.
		Simulates I/O for a virtual channel-to-channel device (channel-to-channel adapter or 3088).
	DMKVCAST	Simulates the channel and device operations of the channel-to-channel device connected between two virtual machines under VM/SP HPO.
DMKVBM		Pageable.
	DMKVBMIN	Initializes the virtual buffer space. This involves creating an address apace, putting the new pseudo VMBLOK on the buffer manager chain, and building the segment, page, and allocation tables needed to manage the address space.
	DMKVBMSC	Scans the buffer manager chain for the requested pseudo VMBLOK.
	DMKVBMVG	Gets a page of virtual memory from the specified address space; marks the page and cylinder where this buffer was allocated.
	DMKVBMVM	Builds a pseudo VMBLOK for system paging.
	DMKVBMVR	Returns a page of virtual memory to the specified address space; deallocates the page and cylinder where the buffer resided.

Module Name	Entry Points	Attributes, Function
DMKVCB		Pageable. Simulates I/O for a virtual channel-to-channel device.
	DMKVCBRD	Selectively resets a virtual channel-to-channel device without decoupling the CTC device from the Y-side adapter.
	DMKVCBRS	Does a final reset for a virtual channel-to-channel device and and disconnects the device from its coupled twin on the Y-side virtual machine.
	DMKVCBSH	Simulates the execution of a HALT I/O or HALT DEVICE instruction for a virtual machine channel-to-channel device.
	DMKVCBTS	Simulates the TEST I/O instruction for a virtual channel-to-channel device that has no interruptions pending.
DMKVCH		Pageable.
	DMKVCHDC	Processes the ATTACH and DETACH real devices and channels) command.
DMKVCN		Resident.
	DMKVCNEX	Simulates all SIOs to a virtual console. If a buffer larger than 4K bytes is required, this module: • Gets the number of pages necessary for the noncontiguous buffer from the page manager, and locks the pages in storage. • Gets and builds an indirect address list from free storage, and uses the indirect address list to address the noncontiguous buffer. • Returns the pages of the buffer via the page manager, and returns the indirect address list space to free storage after the operation is complete.
DMKVCP		Resident.
	DMKVCPIL	Process the function specified in the control area of an IUCV SEND request.
	DMKVCPRE	Process the reply to the write request.
	DMKVCPSR	Process the asynchronous passback from the send.
DMKVCQ		Resident.
	DMKVCQRE	Processes the reply to write request.
	DMKVCQSR	Processes the asynchronous passback from a send.
	DMKVCQAT	TTY ATTENTION (BREAK key) processing.
DMKVCR		Resident.
	DMKVCRNR	Sends any pending batch writes to the VSM.
	DMKVCRMT	Processes the reply to the read. At completion of full-screen read, when an indirect address list is being used, data is moved from WEBLOK to storage using the indirect address list.
	DMKVCRRD	Request the VTAM Communications Network Application (VCNA) to perform a read operation.
	DMKVCRWT	Request the VCNA to perform a write operation.
DMKVCS		Resident.
	DMKVCSWT	Requests VCNA/VSCS to perform a write operation.

Module Name	Entry Points	Attributes, Function
DMKVCT	DMKVCTCH	Process color attribute changes for SNA logical units.
	DMKVCTCN	Initiates the communication path for the VTAM Service Machine and VTAM Logical Units and sets up the initial environment.
	DMKVCTCU	Process the accounting data.
	DMKVCTDA	Disable VSM access to CP.
	DMKVCTEN	Enables VSM access to CP.
	DMKVCTER	Process 'TERM' command change for LUs.
	DMKVCTLO	Release SNA CCS control blocks and process accounting data.
	DMKVCTQS	Suspend activity for the user's virtual machine.
	DMKVCTRM	Terminal characteristics.
	DMKVCTSV	Break the IUCV communication path.
	DMKVCTTM	Send a request to VCNA to redisplay the input line.
DMKVCU		Pageable.
	DMKVCUIL	Processes asynchronous requests for service from VM/VTAM terminals.
DMKVCV	DMKVCVCE	Write trace entry for the CCS transaction.
	DMKVCVEB	Build WEBLOK. When obtaining WEBLOK for a full-screen write, an indirect address list is used to move data to WEBLOK.
	DMKVCVER	Set redisplay timer.
	DMKVCVIN	Release 'read' control blocks.
	DMKVCVIX	Build IXBLOK and do IUCV SEND to VCNA.
	DMKVCVKS	Locates control blocks.
	DMKVCVLD	Build and chain a WEBLOK.
	DMKVCVLY	Issue an IUCV reply for a two-way send.
	DMKVCVND	Issue IUCV send for unsent writes/reads.
	DMKVCVUT	Release 'write' control blocks.
DMKVCW		Resident.
	DMKVCWCN	Initiates the IUCV communication path for the VSA and VTAM logical unit and sets up the initial environment.
	DMKVCWQS	Suspends activity for the virtual machine.
	DMKVCWRM	Resumes the IUCV communication path with the VSA.
	DMKVCWSV	Breaks the IUCV communication path with the VSA.
DMKVCX	DMKVCXD2	Process abend due to invalid control block chain.
	DMKVCXFU	Error handler for IUCV transactions.
	DMKVCXGF	Begin logoff of user.
	DMKVCXGO	Build WEBLOK and move in the system logo.
	DMKVCXIO	Process unrecoverable I/O error.
	DMKVCXOR	Send the color attribute for the LU to the VCNA.
	DMKVCXOX	Do one-way IUCV SEND of logo to VCNA.
	DMKVCXSA	Process VCNA logic error.
DMKVDA		Pageable.
	DMKVDAAA	Processes the ATTACH command.
	DMKVDAAC	Processes the ATTACH CHANNEL command.
DMKVDB		Pageable.
	DMKVDBMD	Handles multiple-address messages for DMKVDA.
DMKVDC		Pageable.
	DMKVDCPS	Acquires virtual blocks for devices that are likely to be attached by the ATTACH command.
	DMKVDCSC	Scans the ATTACH and DETACH command lines and checks syntax.
DMKVDD		Pageable.
	DMKVDDDE	Handles the DETACH command.

Module Name	Entry Points	Attributes, Function
DMKVDE		Pageable.
	DMKVDEDC	Verifies the existence of a device specified on an ATTACH command.
DMKVDF		Pageable.
	DMKVDFRE	Processes the error messages detected in the DETACH command.
DMKVDG		Pageable.
	DMKVDGAL	Creates and chains ALOCBLOKs from the SYSPLIST and RDEVBLOK chain anchors.
DMKVDH		Pageable.
	DMKVDHBB	Builds and initializes an ALOCBLOK for a specified type of system DASD area.
	DMKVDHFR	Builds and initializes a RECBLOK for SYSPP, SYSPG, and SYSSW type ALOCBLOKs.
	DMKVDHPG	Checks for spool files in the passed ALOCBLOK and, if they exist, adjusts the page RECBLOKs.
	DMKVDHOR	Orders ALOCBLOK chains according to SYSPAG macros or selects SYSTEM ordering, as appropriate.
DMKVDR		Pageable.
	DMKVDREL	Releases a virtual or real device from a virtual user.
DMKVDS		Pageable.
	DMKVDSAT	Attaches a virtual device to a user.
	DMKVDSDF	Defines a new virtual device for user.
	DMKVDSLK	Links a virtual DASD device to a user.
DMKVDT		Pageable.
	DMKVDTPG	Attaches a 3880 Storage Control Unit Model 11 or Model 21.
DMKVER		Pageable.
		Processes error records from virtual machine via SVC 76.
	DMKVERD	Processes SVC 76 from DOS or CMS/DOS.
	DMKVERO	Processes SVC 76 from OS, VS/1, VS/2, VM/370, or VM/SP.
DMKVFC		Pageable.
	DMKVFCQV	Process the QUERY VECTOR command.
	DMKVFCVV	Process the VARY ON/OFF VECTOR command.
DMKVFD		Pageable.
	DMKVFDDP	Display vector registers.
DMKVFE		Pageable.
	DMKVFEST	Store vector registers.
DMKVFI		Pageable.
		(*Note: This module is not supplied with source code.*)
	DMKVFIIN	Perform CP initialization of the Vector Facility.
DMKVFR		Resident.
	DMKVFRIN	Process Program Interrupt '19'.
	DMKVFRRE	Restore a virtual machine's Vector Facility status.
	DMKVFRSV	Save a virtual machine's Vector Facility status.
DMKVFS		Pageable.
	DMKVFSOS	Obtain a vector register save area and (if necessary) a VECBLOK.
	DMKVFSRS	Release a vector register save area.

Module Name	Entry Points	Attributes, Function
DMKVIO		Resident. Records and translates the interrupts and status associated with virtual I/O operations.
	DMKVIOC1	Reflects condition code 1 CSW status.
	DMKVIOCL	Clears the VDEVIO queue.
	DMKVIOIN	Translate a virtual I/O interruption.
	DMKVIOMK	Address of a table of interruption masks, indexable by device address.
DMKVMA		Resident.
	DMKVMAER	Issues message DMKVMA456W when a shared segment user attempts to store into the segment. The virtual machine is placed in console function mode.
	DMKVMASH	Checks all protected shared pages associated with shared named systems and determines if they have been changed. If they were changed, the page is returned to CP free storage and the condition code is made nonzero.
	DMKVMASW	Switches the user's segment table entries from one protected shared page table to the other.
DMKVMC		Pageable.
	DMKVMCFC	Main entry for all VMCF subfunctions. Called by DMKHVC when a DIAGNOSE X'0068' instruction is executed. Builds a VMCBLOK with information from user-supplied parameter list, validates the subfunction code, and passes control to appropriate VMCF subroutine.
	DMKVMCEX	Called by DMKDSP to reflect the VMCF external interrupt message header and optional SENDX data to a virtual machine. Copies the message header from the VMCBLOK to the user's external interrupt buffer. If interrupt is for a SENDX request, move SENDX data to the optional area in the external interrupt buffer.
	DMKVMCVA	Branched to from the DMKVMCFC entry point or called by DMKCFP during a system reset. Releases the master VMCBLOK and any final response VMCBLOKs (VMCRESP bit). Returns other VMCBLOKs to the original SOURCE users with the notification that this user is not available.
DMKVMD		Pageable.
	DMKVMDEP	Dumps guest virtual machine to spool blocks in binary form. The output is read by DIAGNOSE X'14'. VM/IPCS Extension or a user-written routine may be used to process the guest virtual machine dump.
DMKVME		Pageable.
	DMKVMEDP	Provides the dumping service for the VMDUMP command and DIAGNOSE Code X'94'.
DMKVMG		Pageable.
	DMKVMGCN	Handles pending CONNECTs from a virtual machine to the Signal system service.
	DMKVMGIL	Handles pending messages representing signals for the Signal system service.
	DMKVMGQS	Handles QUIESCEs from a virtual machine on a path to the Signal system service.
	DMKVMGRM	Handles RESUMES from a virtual machine on a path to the Signal system service.
	DMKVMGSV	Handles pending SEVERs from a virtual machine to the Signal system service.
DMKVMI		Pageable. Loaded into the user's virtual storage when invoked. Performs an IPL of a virtual machine.
	DMKVMIPL	Simulates a user's IPL sequence.

Module Name	Entry Points	Attributes, Function
DMKVRR		Resident.
	DMKVRRDD	For the V=R virtual machine, restores information to the RDEVBLOK associated with the dedicated device.
	DMKVRRIS	Saves data from interruptions that occur while a path to the dump device is being obtained.
	DMKVRRRC	Restores I/O interruption data to the appropriate IOBLOK for the V=R virtual machine. The IOBLOK is then placed on the IOB chain.
	DMKVRRRS	Restores the VMBLOK for the V=R virtual machine and ensures that all dedicated devices are attached. SPMODE is restored if it was active.
DMKVRS		
	DMKVRSSV	Saves the V=R virtual machine's VMBLOK, ECBLOK, CPU timer, and clock comparator. A list is built to include all dedicated devices.
DMKVSC		Resident.
	DMKVSCAN	Verifies CCW translation bypass for SIO or SIOF instruction.
	DMKVSCSC	Scans a V=R channel program for exceptional conditions, such as sense commands, no-ops, I/O to and from page 0, etc., without actually translating the program.
	DMKVSCVR	Scan for conditions indicating NOTRANS is a valid operation.
DMKVSD		Resident.
	DMKVSDAD	Adds a reader file to the user's reader chain.
	DMKVSDDL	Deletes a reader file from a user's reader chain.
	DMKVSDFH	Finds the Reader Hash Table entry for the given user.
DMKVSE		Resident.
	DMKVSEER	Recovers from SYSSPOOL paging error.
	DMKVSETR	Gets a control block from SYSSPOOL's virtual storage.
DMKVSF		Resident.
	DMKVSFID	Finds the specified spool file's SFBLOK.
	DMKVSFNS	Finds the next spool file in the system.
	DMKVSFNU	Finds the user's next SFBLOK.
DMKVSG		Resident.
	DMKVSGAI	Assigns a user-unique spoolid.
	DMKVSGRI	Returns a user-unique spoolid.
	DMKVSGUM	Updates a user's SPUMAP.
DMKVSI		Resident. Simulates the operation of privileged I/O instructions issued by virtual machines.
	DMKVSIEX	Simulates an SIO, TIO, HIO, TCH, or CLCH. (HIO and CLCH are processed by DMKVSJ.)
	DMKVSIFT	Reentry from DMKVSJ to release the IOBLOK.
DMKVSJ		Resident.
	DMKVSJEX	Simulates the operation of the privileged I/O instructions HIO and CLCH issued by virtual machines.

Module Name	Entry Points	Attributes, Function
DMKVSP		Resident.
		Simulates all user SIOs to a virtual unit record device (real reader, punch, printer, or pseudo timer) that is spooled rather than dedicated.
	DMKVSPDC	Clears a device.
	DMKVSPFC	Clears a file.
	DMKVSPEX	Simulates an SIO to a spooled unit record device.
	DMKVSPPE	Printer end-of-file processing.
	DMKVSPST	Stacks a CPEXBLOK.
	DMKVSPTO	Checks whether the virtual reader is empty.
	DMKVSPUS	Unstacks a CPEXBLOK.
	DMKVSPWA	Nonexecutable index work area for the 3211.
DMKVSQ		Resident.
	DMKVSQPD	Locate next available slot in a printer/punch DASD buffer and move CCW and data into that slot.
DMKVSR		Resident.
	DMKVSRGC	Finds and validates the next non-TIC CCW in the user's channel program.
	DMKVSRMD	Locates user's data area and moves data between the user's area and the in-storage work buffer.
DMKVST		Resident.
		Handles CP requests to print on the user's virtual printer.
	DMKVSTCP	Writes a print line to the console.
	DMKVSTOP	Opens output spool file.
	DMKVSTPT	Puts a CP-generated line on the user's spooled printer.
DMKVSU		Pageable.
	DMKVSUCO	Stops processing the file currently in the spooled printer or punch and clears all pending status from the spooled printer or punch.
	DMKVSUCR	Stops processing the file currently in the spooled card reader and clears all pending status from the spooled card reader.
DMKVSV		Resident.
	DMKVSVLD	Verifies the validity of 3800 load CCWs and sets up the in-storage work buffer.
DMKVSW		Resident.
	DMKVSWTO	TIO to spooled reader.
	DMKVSWOT	Reader file half open-finish open.
	DMKVSWOR	Opens reader file.
	DMKVSWOC	Searches the reader file chain.
	DMKVSWFC	Starts clearing old reader files.
	DMKVSWDC	Clears device blocks.
DMKVSX		Resident.
		Simulates all user SIOs to a virtual card reader or pseudo timer that is spooled rather than dedicated. Simulates sense CCWs for spooled unit record devices and sense ID CCWs for spooled printers.
	DMKVSXCL	Simulates a close of a spooled unit record reader.
	DMKVSXRD	Locates the next data record in a reader file.
	DMKVSXSE	Simulates a sense to a spooled unit record.
	DMKVSXSI	Simulates a sense ID to a spooled printer.
	DMKVSXSR	Simulates an SIO to a spooled unit record reader.
	DMKVSXTR	Simulates an SIO to timer.

Module Name	Entry Points	Attributes, Function
DMKWAI		Resident. Scans dispatch queues for dispatchable work that may have been readied by another processor.
	DMKWAIST	Scans the dispatch list of the other processor. When it finds a dispatchable unit of work, it moves it to the dispatch list of this processor and exits to the dispatcher.
	DMKWAITA	Active wait processing.
DMKWRM		Pageable.
	DMKWRMST	Warm start processing. Retrieves system log messages and accounting cards from the warm start cylinder of the IPL pack.
	DMKWRMSY	Rebuilds SYSSPOOL's virtual storage from the WARM cylinders.
DMKWRN		Pageable.
	DMKWRNWM	Initializes the checkpoint area cylinders and checkpoints any SHQBLOKs or open SFBLOKs.
	DMKWRNSB	Sets bit in system spool file bit map.
DMKXAB		Pageable.
	DMKXABDG	Processes the DIAGNOSE X'B4' and X'B8' instructions.
DMKXAD		Pageable.
	DMKXADVS	Copies the XAB (External Attribute Buffer) associated with a virtual printer to a spool file for that device.
DMKXST		Pageable.
	DMKXSTOR	Initialize Paging Storage ALOCBLOKs and RECBLOKs, if the SYSXSTOR macro was used in DMKSYS.
DMKZTD		Pageable.
	DMKZTDDF	Clears the first CKD cylinder or first FB-512 block of a TDSK minidisk if SYSCLR=NO was specified on the SYSRES macro.
	DMKZTDST	Cleans CKD and FBA-512 T-Disk Space.

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)								
DMKACO	ACCTBLOK	ACCTUSER	ACNTBACK	ACNTBLOK	ACNTCODE	ACNTCONT	ACNTDATA	ACNTDEVC	ACNTDEVM	ACNTDEVT
	ACNTIOCT	ACNTNCYL	ACNTNEXT	ACNTNUM	ACNTPGRD	ACNTRFLG	ACNTRMID	ACNTRMTE	ACNTSIZE	ACNTSNA
	ACNTSTOP	ACNTTIME	ACNTUSER	ACNTVTIM	ACNTVTT	ACNTVVT	ACOACCL	ACOACCM	ACOCHK	ACOEXIT
	ACORETN	ACTIAC	ACTIBF1R	ACTIBF1V	ACTIBF2R	ACTIBF2V	ACTIBLOK	ACTICL	ACTICLAS	ACTICNT
	ACTIDCUR	ACTIDISP	ACTIFLAG	ACTIID	ACTILIMT	ACTIPCH	ACTISFB	ACTISFCK	ADDSFB	ADSPCH
	AEXTSP	AFREE	AFREEP	AFRET	ALARM	AP	APSTAT1	APTRAN	APTRLK	APUOPER
	AQCNWT	ARSPAC	ARSPPU	ASYSLC	ASYSVM	BLANKLEN	BLANKS	BRING	CALLCVT	CC
	CHECKVF	CHGSFB	CLASDASD	CLASFBA	CLASGRAF	CLASTERM	CLASURI	CONTROL	CPEXADD	CPEXBLOK
	CPEXREGS	CPEXR0	CPEXR11	CPEXR12	CPEXSIZE	C1	DATE	DE	DEDTEST	DEFER
	DEVCARD	DFRET	DISCREC	DISPMSG	DMKCKSPL	DMKCKTUU	DMKCVTAB	DMKCVTBH	DMKCVTDT	DMKDSPCH
	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKLOCRD	DMKLOCRQ	DMKPGTSG	DMKPGUVG	DMKPTRAN	DMKPTRLK
	DMKPTRUL	DMKQCNWT	DMKQCORD	DMKRPAPT	DMKSCHDL	DMKSCNAU	DMKSCNRD	DMKSCNVD	DMKSTKCP	DMKSTKIO
	DMKSTKOP	DMKSYSAC	DMKSYSCK	DMKTMRPT	DMKVFRSV	DMKVIOIN	DMKVSDAD	DOLOCAL	DORDEV	DOTERMID
	DOVMTERM	FNAME	FTROPRDR	F0	F1	F255	F60	INHIBIT	IOBCSW	IOBIRA
	IOBLINK	IOBLOK	IOBSIZE	IOBUSER	IOBVADD	IPUADDRX	LOCGRAF	LOCK	MP	NICBLOK
	NICDTYPE	NICMDL	NICOPRDR	NICSIZE	NICTMAT	NICTYPE	NOAI	NOCARRY1	NORET	NOTMID
	NOTRESP	OPNSFB	PCHCHN	PREFIXB	PRIORITY	PSA	RDEVADD	RDEVBLOK	RDEVFTR	RDEVMDL
	RDEVNICL	RDEVPS	RDEVSNRB	RDEVSTA3	RDEVTMAT	RDEVTYPC	RDEVTYPE	RDRCHN	REMOTE	RETRY
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER10	SAVER2
	SAVER6	SAVEVAC	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK6	SAVEWRK7	SETTDK	SETUP
	SETUP1	SETUP2	SFBACNT	SFBCLAS	SFBCOPY	SFBDATE	SFBDIST	SFBFLAG3	SFBFNAME	SFBLAST
	SFBLOK	SFBOFORM	SFBORIG	SFBRECNO	SFBRECSZ	SFBSIZE	SFBSTART	SFBSYSID	SFBTIME	SFBTYPE
	SFBUFORM	SFBUSER	SIGEMS	SIGQUI	SIGRES	SIGXC	SILI	SKIPADD	SNACARD	SNARBLOK
	SNARLUN	SNAUSER	SPFILID	SPLINK	SPNXTPAG	SPPREPAG	SPRECNUM	SPTIME	STCODE	SYSLOCS
	SYSTEM	TEMPSAVE	TERMEXIT	TERMRMID	TIMEDISP	TODATE	TYPBSC	TYPRDR	TYP2540P	TYP3277
	USERCARD	VACOK	VACOVFR	VCONCTL	VCONNICB	VCONRDEV	VECOPVF	VECSTAT	VECUSER	VMBLOK
	ZEROES									
DMKACR	ACRLOCK	ADSPCH	AFREE	AFRET	ALARM	ALLCHANS	ALOKSP	AMCHAREA	AP	APSTAT1
	APSTAT4	APUOPER	AQCNWT	ARIOCH	ARIOCT	ARIOCU	ARIODV	ASYSVM	BCTWAIT	BLANKS
	CCS4DGRD	CCS4HARD	CCS4SOFT	CLASDASD	CLASFBA	CLASTERM	CPCREG0	CPSTAT5	C0	DMKACSCV
	DMKACSRF	DMKCVTBH	DMKDSPCH	DMKFREE	DMKFRET	DMKIOSQE	DMKLOKIO	DMKLOKSW	DMKMCHST	DMKQCNWT
	DMKSCNMU	DMKSCNRU	DMKSYSOC	DMKSYSOW	ERRMSG	EXTMASK	FFS	F0	F1	F2
	IOBCP	IOBFLAG	IOBFPNT	IOBLINK	IOBLOK	IOBRADD	IOBUSER	LOCK	LOCKSAV	LPUADDR
	MCHAREA	MCHFLAG1	MCH1GERR	MCNPSW	MFAMASK	MP	NICSIZE	NORET	OPERATOR	OWNDLIST
	OWNDRDEV	POFFLINE	PREFIXB	PSA	RCHADD	RCHBLOK	RCHCUTBL	RCHDISA	RCHSTAT	RCUBLOK
	RCUCHA	RCUCHD	RCUDISA	RCUDVTBL	RCUPRIME	RCUSTAT	RCUSUB	RCUTYPE	RDEVADD	RDEVALT
	RDEVAOF	RDEVBLOK	RDEVBOF	RDEVCUA	RDEVDISA	RDEVDISB	RDEVENAB	RDEVFLAG	RDEVFOFF	RDEVLCEP
	RDEVLNCP	RDEVMAX	RDEVNICL	RDEVPEND	RDEVPTHS	RDEVPTH1	RDEVPTH2	RDEVPTH3	RDEVPTH4	RDEVPTH5
	RDEVPTH6	RDEVPTH7	RDEVPTH8	RDEVRRES	RDEVSER	RDEVSTAT	RDEVSTA2	RDEVSTA4	RDEVTYPC	RDEVTYPE
	RDIDX	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVER1	SVMUNLOK	TIMEDISP
	TYP2305	TYP3705	VMBLOK	ZEROES						
DMKACS	ACRLOCK	AFREE	AFREEP	AFRET	ALLCHANS	ALOKSP	AP	APSTAT1	APUOPER	ARIOCH
	ARIOCT	ARIOCU	ARIODV	ASYSVM	BCTWAIT	BUSY	CAW	CCC	CCS4DGRD	CCS4HARD
	CCS4SOFT	CDC	CLASTAPE	CLASTERM	CODE	CONTINUE	CPEXADD	CPEXBLOK	CPEXREGS	CPEXR11
	CPEXSIZE	CPSHUT	CPSTAT4	CPSTAT5	CSW	CUE	DMKACRIO	DMKCCHCF	DMKCNSIN	DMKCVTAB
	DMKFREE	DMKFREEP	DMKFRET	DMKIOSMQ	DMKIOTRC	DMKLOKIO	DMKSCNMU	DMKSCNPH	DMKSCNRA	DMKSCNRD
	DMKSTKCP	DMKSTKIO	DMKSTKMP	FFS	F1	F2	F4	F8	F9	IOBBPNT
	IOBCAW	IOBCC3	IOBCP	IOBCSW	IOBFATAL	IOBFLAG	IOBFPNT	IOBHVC	IOBIOER	IOBLINK
	IOBLOK	IOBMINI	IOBMQDTO	IOBOERR	IOBPATHF	IOBQDTO	IOBRADD	IOBRETRY	IOBSIZE	IOBSPEC
	IOBSPEC2	IOBSPEC3	IOBSTAT	IOBTIO	IOBUNSL	IOBUSER	IOERBLOK	IOERCCH	IOERCCUA	IOERCHID

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	IOERCLOG	IOERCPID	IOERCSW	IOERECSW	IOEREXT	IOERLOGL	IOERSIZE	IPUADDRX	LOCK	LOCKSAV
	LPUADDR	LPUADDRX	MP	PREFIXA	PREFIXB	PROCIPL	PSA	PSAPGID	RCHADD	RCHBLOK
	RCHBUSY	RCHCUTBL	RCHDISA	RCHFIOB	RCHSTAT	RCUADD	RCUBLOK	RCUBUSY	RCUCHA	RCUDISA
	RCUDVTBL	RCUFIOB	RCUPRIME	RCUSCED	RCUSTAT	RCUSUB	RCUTYPE	RDEVADD	RDEVAIOB	RDEVALT
	RDEVASGN	RDEVBLOK	RDEVBZCH	RDEVCSW	RDEVCTL	RDEVCUA	RDEVCUB	RDEVDISA	RDEVFIOB	RDEVLIOB
	RDEVMID	RDEVPROC	RDEVPTHS	RDEVQCNT	RDEVSCHD	RDEVSTAT	RDEVSTA2	RDEVSTA4	RDEVSTA5	RDEVTFLG
	RDEVTYPC	RDEVTYPE	RDEVUSER	RDIDX	R0	R1	R10	R12	R13	R14
	R15	R2	R3	R4	R5	R6	R7	R8	R9	SIGPR
	SILI	SIOCCH	SM	TRACBEF	TRACCURR	TRACEND	TRACFLG2	TRACPROC	TRACSTRT	TRACSVCR
	TYP3480	UC								
DMKALG	ADSPCH	AFREE	AFREEP	AP	APSTAT1	APUOPER	BUFCNT	BUFFER	BUFNXT	BUFSIZE
	CPEXADD	CPEXBLOK	CPEXR0	CPEXR12	CPEXSIZE	DMKBLDVM	DMKERMSG	DMKFREE	DMKFREEP	DMKLOGB
	DMKLOKSW	DMKSCHDL	DMKSCNVD	DMKSTKCP	DMKSYSJR	F1	LOCK	MP	PSA	R0
	R1	R10	R11	R12	R13	R14	R15	R2	R3	R4
	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER11	SAVER2	SAVER9	SAVEWRK1	SAVEWRK8
	TIMEDISP	VCONCTL	VCONRBUF	VCONRCNT	VCONRDSZ	VMBLOK				
DMKALO	AFREE	ALOCBLOK	ALOCMAP	ALOCMAX	ALOCNPAG	ALOCPNT	ALOCRCUU	ALOCRDEV	AP	ARIODV
	ASYSVM	CLASDASD	CLASFBA	DMKCPIFT	DMKCPILF	DMKCPIMS	DMKCPINU	DMKCPIRP	DMKCPISH	DMKCPIVP
	DMKCVTBH	DMKFREE	DMKMNTFB	DMKSCNRD	DMKSYSOW	DMKSYSVL	DMKTODTB	FTRVIRT	FTR35MB	F1
	LOCK	MP	NORET	OWNDLIST	OWNDRDEV	PSA	RDCBLOK	RDCPAGAP	RDCPAGFA	RDCPAGMA
	RDEVALLN	RDEVBLOK	RDEVCODE	RDEVCUP	RDEVFLAG	RDEVFTR	RDEVMDL	RDEVMD02	RDEVNATH	RDEVOWN
	RDEVPRDV	RDEVRDC	RDEVSER	RDEVSTA5	RDEVSYS	RDEVTYPC	RDEVTYPE	R0	R1	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R8
	SAVEAREA	SAVEREGS	TYP2305	TYP2314	TYP3310	TYP3330	TYP3340	TYP3350	TYP3370	TYP3375
	TYP3380									
DMKAPI	ACTIVTRQ	AFREE	ALOKDS	ALOKSP	ALOKSY	ALOKTRL	ALOKVM	AP	APSTAT1	APSTAT4
	APUOPER	ARIOCT	ASYSVM	BALRSAVE	BALRSAV2	BALR1	CKCMASK	CPASTAVL	CPASTON	CPCREG0
	CPCREG6	CPDASAAV	CPDASAON	CPEXSIZE	CPINITD	CPMICAVL	CPMICON	CPSEGPRT	CPSHRLK	CPSTATUS
	CPSTAT2	CPSTAT3	CPSTAT4	CPSUPER	CPTMASK	CPUID	CPUMODEL	CPWAIT	CP370EAV	CP370EON
	CSADDR	CSSFEAT	C0	C1	C14	C2	C6	DMKCCWB1	DMKCCWB2	DMKCCWB3
	DMKCCWB4	DMKCCWB5	DMKCCWB6	DMKCCWB7	DMKCCWB8	DMKCCWGN	DMKCCWL1	DMKCCWL2	DMKCCWL3	DMKCCWL4
	DMKCCWL5	DMKCCW0	DMKCCW1	DMKCPEML	DMKDGDA0	DMKDGDA1	DMKDGDA2	DMKDGDA3	DMKDGDA4	DMKDGDA5
	DMKDGDA6	DMKDGDA8	DMKDGDA9	DMKDGDDK	DMKDGFA0	DMKDMPAA	DMKDMPC2	DMKDMPMA	DMKDMPPA	DMKDMPPX
	DMKDMPSA	DMKDSP0	DMKDSP1	DMKDSP2	DMKDSP3	DMKDSP4	DMKDSP5	DMKDSP6	DMKFREE	DMKFREEP
	DMKFRELP	DMKFRET	DMKIOGAP	DMKPRGIN	DMKPRVCA	DMKPRWSK	DMKPSADU	DMKPTRLX	DMKPTRUX	DMKPXA
	DMKPXB	DMKQCNWT	DMKQCOSY	DMKSCNMI	DMKSCNVU	DMKSTPX	DMKSVCIN	DMKSYSMP	DMKSYSNP	DMKUNTFR
	DMKUNTRN	DMKVATZP	DMKVATZS	DMKVAUZP	DMKVFRIP	DMKVFRNP	DMKVMASH	DMKWAITA	DPAPUOP	DPMICON
	DPOKTLB	DPSEGPRT	DPSTAT	DSPRQ	EMSMASK	EXNPSW	EXOPSW	EXTMASK	EXTMODE	FASTCPU
	FF	FFS	F1	F4096	IDLEWAIT	IOMASK	IONTWAIT	IPUADDR	IPUADDRX	LASTUSER
	LOCK	LPUADDR	LPUADDRX	L8	MCHEK	MFAMASK	MICBLOK	MICDASA	MICEVMA3	MICISKE
	MICSKYMD	MP	MPUOPER	MSSFMASK	PAGEWAIT	PGREAD	PMAAVAIL	PMASTAT	POFFLINE	PREFIXA
	PREFIXB	PRIMEHDR	PRIMEHI	PRIMELO	PRNPSW	PROBTIME	PROCIO	PROCIPL	PSA	PSADSPRQ
	PSAEXT	PSAEXTX	PSENDCLR	PXA	RQLOCK	RSRTNPSW	RUNCR0	RUNCR1	RUNPSW	RUNUSER
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R8	R9	SAVEAREA	SAVEREGS	SAVER11	SAVEWRK2	SAVEWRK3
	SHRLKCNT	SIGREST	SVCNPSW	TEMPR0	TEMPR2	TEMPR3	TEMPR4	TEMPR5	TEMPR6	TEMPR7
	TEMPSAVE	TIMEDISP	TIMER	TRACPROC	TRACSTRT	TRLLOCK	VECAVAIL	VECF	VECOPVF	VECSTAT
	VECUSER	VMBLOK	WAIT	XCMASK	XKEYASST	ZEROES				
DMKAPS	ADSPCH	AFREE	AFREEP	AFRET	ALPRTBLK	AP	APTRLK	ARSPPR	BLANK	CPEXADD
	CPEXBLOK	CPEXSIZE	DMKAPTEP	DMKAPUSE	DMKAPVCL	DMKAPXMG	DMKCVTHB	DMKDSPCH	DMKFREE	DMKFREEP
	DMKFRET	DMKIUACP	DMKPTRLK	DMKPTRUL	FFS	F3	F6	LOCK	LPRCPXAD	LPRFLG3

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	LPRLSPLC	LPRNAME	LPRNEXT	LPRPATH	LPRSEVER	LPRTBLOK	LPRTSIZE	LPRUSRID	LSPIOPND	LSPLCTL
	LSPLFLG1	LSPLNEXT	LSPLSIZE	LSPPRINT	LSPSFBLK	L8	MP	PSA	R0	R1
	R10	R11	R12	R13	R14	R15	R2	R3	R4	R5
	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SEVER	SFBBCONV	SFBFLAG	SFBFLAG4
	SFBINUSE	SFBLOK	SFBPNT							
DMKAPT	ADSPCH	AFREE	AFREEP	AFRET	AP	APTRAN	APTRLK	BRING	CPEXADD	CPEXBLOK
	CPEXREGS	CPEXSIZE	C1	DEFER	DMKAPSSV	DMKAPWPG	DMKDSPCH	DMKFREE	DMKFREEP	DMKFRET
	DMKIUACP	DMKPSASP	DMKPTRAN	DMKPTRLK	DMKPTRUL	DMKRPAGT	DMKSTKCP	DMKXABDG	FFS	F0
	F4096	IOERETN	IOKEEP	LOCK	LPRCPXAD	LPRCTIOP	LPRFLG3	LPRLSPLC	LPRREADB	LPRSEVER
	LPRTBLOK	LPRXAB	LSPIOPND	LSPLCTL	LSPLFLG1	LSPLNEXT	LSPPURGE	LSPSFBLK	LSPSPLNK	MP
	PSA	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER5
	SEVER	SFBFILID	SFBLOK	SFBPNT	SFBSIZE	SFBSTART	SFBUSER	SPLINK	SPNXTPAG	VMBLOK
	XPAGNUM									
DMKAPU	AFREE	ALPRTBLK	AP	ARSPPR	BLANK	BLANKS	DMKAPSSV	DMKFREE	DMKIUACP	DMKSYSFL
	FFS	FORMBLOK	FORMNTRY	FORMOPER	FORMSEND	FORMUSER	F4	LOCK	LPRAN38D	LPRBE38D
	LPRCHGE	LPRCLASS	LPRCNVD	LPRCNVRT	LPRDEFLT	LPRDEST1	LPRFLASH	LPRFLG1	LPRFLG1D	LPRFLG2
	LPRFLG3	LPRFORM	LPRIDLE	LPRLSELD	LPRLSPLC	LPRNCNVD	LPRNEXT	LPRNO38D	LPRN5AD	LPRSELDA
	LPRSESID	LPRTBLOK	LPR5AD	LSPCONVT	LSPID	LSPLCTL	LSPLFLG1	LSPLNEXT	LSPLPRTB	LSPLSIZE
	LSPPRINT	LSPSFBLK	LSPSPLNK	MP	PSA	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SFBBCONV	SFBCLAS	SFBCONV	SFBCOPY	SFBDEST	SFBFILID	SFBFLAG
	SFBFLAG3	SFBFLAG4	SFBFLASH	SFBINUSE	SFBLDBEG	SFBLDMID	SFBLOK	SFBOFORM	SFBPNT	SFBPURGD
	SFBSHOLD	SFBSTART	SFBTYPE	SFBUHOLD	TYP5ACCW	VMBLOK	ZEROES			
DMKAPV	AFRET	AP	ARSPPR	CHGSFB	DMKAPSSV	DMKAPZNO	DMKCKSPL	DMKCSOSD	DMKFRET	DMKIUACP
	DMKSPKDL	DMKSYSFL	FORMBLOK	FORMNTRY	FORMOPER	FORMSEND	FORMUSER	F255	LOCK	LPRLSPLC
	LPRRECBF	LPRTBLOK	LSPCONVT	LSPIOPND	LSPLCTL	LSPLFLG1	LSPLNEXT	LSPLSIZE	LSPPRINT	LSPSFBLK
	MP	PSA	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS
	SAVER5	SEVER	SFBBCONV	SFBCLAS	SFBCONV	SFBCOPY	SFBDEST	SFBFILID	SFBFLAG	SFBFLAG4
	SFBINUSE	SFBLOK	SFBOFORM	SFBPNT	SFBSHOLD	SFBTYPE	SFBUFORM	SFBUHOLD	VMBLOK	ZEROES
DMKAPW	AFREE	AFRET	ALPRTBLK	AP	DMKFREE	DMKFRET	DMKIUACP	LOCK	LPRLSPLC	LPRNEXT
	LPRPATH	LPRTBLOK	LSPID	LSPIOPND	LSPLCTL	LSPLFLG1	LSPLNEXT	LSPLSIZE	LSPPURGE	MP
	PSA	R0	R1	R11	R12	R13	R14	R15	R3	R4
	R5	R7	SAVEAREA	SAVEREGS	SFBFILID	SFBLOK				
DMKAPX	AFREE	AFRET	AP	BLANK	BLANKS	BUFCNT	BUFFER	BUFIN	BUFINLTH	BUFNXT
	BUFSIZE	DMKAPSSV	DMKFREE	DMKFRET	DMKIUACP	DMKMSGAL	DMKMSGMS	DMKMSGSM	F3	LOCK
	MP	PSA	R0	R1	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R7	R9	SAVEAREA	SAVEREGS	SAVEWRK1	SAVEWRK3	VMBLOK
	ZEROES									
DMKAPY	ADSPCH	AFREE	AFRET	ALPRTBLK	AP	APTRLK	BLANK	BUFCNT	BUFFER	BUFNXT
	DMKDSPCH	DMKFREE	DMKFRET	DMKIUACP	DMKPTRLK	DMKPTRUL	F2	IXBLOK	IXIRA	IXREGS
	IXSIZE	LOCK	LPRNAME	LPRNEXT	LPRPATH	LPRTBLOK	MP	PSA	R0	R1
	R10	R11	R12	R13	R14	R15	R2	R3	R4	R5
	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVEWRK1	VMBLOK		
DMKAPZ	AFREE	AFRET	ALPRTBLK	AP	DMKFREE	DMKFRET	DMKIUACP	LOCK	LPRFLG3	LPRIDLE
	LPRNEXT	LPRPATH	LPRTBLOK	MP	PSA	R0	R1	R11	R12	R13
	R14	R15	R8	SAVEAREA	SAVEREGS					

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
DMKATS	ACORETBL	ADSPCH	AFREE	AFREEP	AFRET	ALOCBLOK	ALOCFLG	ALOCPP	APSTAT1	APSTAT2
	APTRAN	APTRLK	APUOPER	ASYSVM	BRING	CLASFBA	CORBPNT	CORCFLCK	CORFLAG	CORFLUSH
	CORFPNT	CORFREE	CORIOLCK	CORPGPNT	CORSHARE	CORSWPNT	CORTABLE	CORVM	CPEXADD	CPEXBLOK
	CPEXREGS	CPEXSIZE	CPPTLBR	C1	DEFER	DMKDSPCH	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET
	DMKLOCKD	DMKLOCKQ	DMKPGUAL	DMKPGUPR	DMKPTRAN	DMKPTRAQ	DMKPTRLK	DMKPTRSC	DMKPTRUL	DMKPTSAD
	DMKPTSPW	DMKPTTFT	DMKSCNVS	DMKSNTBL	DMKSTKCP	DMKSYSAP	DMKVMASH	FFS	F0	F1
	F15	F16	F256	F4	F4096	F8	LASTUSER	LOCK	LPUADDR	MPFEAT
	PAGACT	PAGBMP	PAGCORE	PAGINVAL	PAGRBITS	PAGSHR	PAGSWP	PAGTABLE	PAGTOT	PAGTSWP
	PREFIXA	PROCIPL	PSA	RDEVBLOK	RDEVCODE	RDEVTYPC	RDEVTYPE	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SAVEAREA	SAVEREGS	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5
	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	SEGFLAG	SEGINV	SEGPAGE	SEGPROT	SEGTABLE	SHRBPNT
	SHRFLAG	SHRFPNT	SHRNAME	SHRNOPRT	SHRPAGE	SHRSEGCT	SHRSEGNM	SHRSGPRT	SHRTABLE	SHRTSIZE
	SHRUSECT	SWPALLOC	SWPAPP	SWPCHG1	SWPCODE	SWPCYL	SWPFLAG	SWPFLAG2	SWPKEY1	SWPPAG
	SWPPSTOR	SWPRECMP	SWPSHR	SWPTABLE	SWPVM	SWPVPAGE	SYSNAME	SYSTEM	TEMPR0	TEMPR1
	TEMPR10	TEMPR14	TEMPR15	TEMPR2	TTSEGCNT	TYP2314	TYP3330	TYP3350	TYP3375	TYP3380
	VMABLOK	VMAFPNT	VMANAME	VMASIZE	VMBLOK	XPAGNUM	ZEROES			
DMKBIO	ADSPCH	AFREE	AFRET	AP	APTRAN	APTRLK	BIOBIRS	BIOBKSZ	BIOBLKND	BIOBLKST
	BIOBLOK	BIODEVDA	BIODEVDS	BIOFBA	BIOFLAG2	BIONEXT	BIOOFFCP	BIOOFFST	BIOPARMA	BIOPARML
	BIOPARMU	BIOPATH	BIORC	BIOREAD	BIORESET	BIORPS	BIOSECDS	BIOSIZE	BIOSTAT1	BIOSVRD
	BIOVBLKS	BIOVDEV	BIOVDEVA	BIO3344K	BIO512BL	BIRBIOBL	BIRBLOK	BIRCCWS	BIRCMSRD	BIRCMSWR
	BIREXEND	BIREXIND	BIREXOFF	BIREXRD	BIREXTLI	BIREXWR	BIRIDAL	BIRIDAW1	BIRIDAW2	BIRIDAW3
	BIRLCBYT	BIRLCIND	BIRLCOFF	BIRLCRD	BIRLCWR	BIRLOCLI	BIRMSGID	BIRPAGES	BIRPAGE1	BIRPAGE2
	BIRPARML	BIRRWCNT	BIRRWIDA	BIRRWOP	BIRRWRD	BIRSECAR	BIRSECOP	BIRSECT	BIRSIZE	BIRSKCYL
	BIRSKDA	BIRSKDAT	BIRSKOP	BIRSKTRK	BIRSRCYL	BIRSRDA	BIRSRDAT	BIRSROP	BIRSRREC	BIRSRTRK
	BIRTICAD	BIRTRGCL	BIRWCNT	BIRWIDA	BIRWOP	BIRWRD	BLKSIZE	BRING	CLASDASD	CLASFBA
	C1	DEFER	DMKDSPCH	DMKFREE	DMKFRET	DMKIOSQR	DMKIUACP	DMKPSAFP	DMKPSASP	DMKPTRAN
	DMKPTRLE	DMKPTRLK	DMKPTRUL	DMKSCNVU	FTRRPS	FTR35MB	F0	F2	F4096	IL
	IOBCAW	IOBCSW	IOBCYL	IOBFATAL	IOBIOER	IOBIRA	IOBLOK	IOBMISC2	IOBSIZE	IOBSTAT
	IOBUSER	IOERSIZE	LOCK	MP	PSA	RDCBLKAP	RDCBLKMA	RDCBLOK	RDCPRIM	RDEVBLOK
	RDEVCRDC	RDEVFTR	RDEVMDL	RDEVRDC	RDEVSTA6	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK6	SAVEWRK9	SEVER	SYSVIRT
	TYP2314	TYP3310	TYP3330	TYP3340	TYP3350	TYP3370	TYP3375	TYP3380	VFAULT	VIRTUAL
	VMBLOK	VMSIZE	XPAGNUM	ZEROES						
DMKBLD	ACORETBL	AFREE	AFREEP	AFRET	AP	APSTAT1	APSTAT2	APTRAN	APUOPER	AQCNWT
	ASYSLC	ASYSVM	AVMREAL	CLASGRAF	CLASSPEC	CLASTERM	CORCFLCK	CORFLAG	CORFPNT	CORIOLCK
	CORLCNT	CORPGPNT	CORSWPNT	CORTABLE	CPEXSIZE	CPPTLBR	C1	DEFER	DELPAGES	DELSEGS
	DMKCVTAB	DMKCVTBH	DMKERMSG	DMKFREE	DMKFREEA	DMKFREEP	DMKFRET	DMKLOKDF	DMKPTRAN	DMKQCNWT
	DMKRIORN	DMKSCNRD	DMKSYSDS	DMKSYSLE	DMKSYSLL	DMKSYSSZ	DMKTMRCK	DMKTRQCP	DMKVATSI	DMKVRR
	FFS	F0	F1	F15	F16	F255	F4	F4095	F7	F8
	KEEPSEGS	LASTUSER	LLTTY	LL2741	LL3066	LOCK	LOCKSAV	LPUADDR	MICBLOK	MICRSEG
	MICSIZE	MP	NEWPAGES	NEWSEGS	NICBLOK	NICCIBM	NICGRAF	NICLLEN	NICNAME	NICTERM
	NICTYPE	NICUSER	NICWTH	NORET	OLDVMSEG	PAGBMP	PAGCORE	PAGPGSWP	PAGSWP	PAGTABLE
	PAGTONLY	PREFIXA	PREFIXB	PSA	PSALANG	RDEVASTB	RDEVAVM1	RDEVBLOK	RDEVFLAG	RDEVLLEN
	RDEVPCHG	RDEVPS	RDEVPSUP	RDEVSCRL	RDEVSNA	RDEVSNRB	RDEVSTA3	RDEVTFLG	RDEVTMCD	RDEVTYPC
	RDEVTYPE	RDEVUSC8	RDEVUSER	RDEVVM2	RDEVWTH	RDEV3101	R0	R1	R10	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	SAVEAREA	SAVEREGS	SAVER1	SAVER10	SAVER11	SAVER2	SAVER8	SAVEWRK1
	SAVEWRK2	SAVEWRK9	SEGENQ	SEGINV	SEGPAGE	SEGPLEN	SEGTABLE	SNARBLOK	SNARLUN	SPECIALV
	SSBENTRL	SSBHEADL	STARTIME	SWPFLAG	SWPPAG	SWPRECMP	SWPSEGNO	SWPTABLE	SWPVM	SYSLOCS
	TIMEDISP	TRQBIRA	TRQBLOK	TRQBSIZE	TRQBUSER	TTSEGCNT	TYPBSC	TYPTTY	TYP3066	TYP3705

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	VMBLOK	VMSIZE	VRALOC	WAIT	ZEROES					
DMKBOX	ATTRPRHI	ATTRSKIP	ATTR457	ATTR7	IC	SF				
DMKBSC	AFREE	AFRET	BSCBLOK	BSCFLAG	BSCFLAG1	BSCHALT	BSCPCCW1	BSCPCCW2	BSCRCVD	BSCREAD
	BSCRESP	BSCRSTRT	BSCRVI	BSCSHUT	CC	CCC	CDC	CHC	DMKFREE	DMKFRET
	DMKIOEST	DMKMSWR	DMKRGACC	ENQ	FTRDIAL	F1	F15	F7	F8	IFCC
	IOBCAW	IOBCSW	IOBERP	IOBFATAL	IOBFLAG	IOBIOER	IOBLOK	IOBMISC2	IOBRCAW	IOBRCNT
	IOBRSTRT	IOBSPEC	IOBSTAT	IOBUNSL	IOERACT	IOERBLOK	IOERCAN	IOERCCRA	IOERCCRL	IOERCSW
	IOERDATA	IOERDW	IOEREXT	IOERFLG2	IOERFLG3	IOERIND3	IOERINFO	IOERLOC	IOERMSW	IOERNUM
	IOERREAD	IOERSIZE	LOCK	NOTUSED	PRGC	PRTC	PSA	RDEVBLOK	RDEVBSC	RDEVFTR
	RDEVIOER	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SILI
	SKIP	TPOPFSLB	TPOPRDXP	UC	UE	WRITE	ZEROES			
DMKCAC	ADSPCH	AFREE	AFRET	AQCNWT	ARIOCU	ARIOUC	ASYSOP	BLANK	BLANKS	DMKCVTBD
	DMKCVTBH	DMKCVTHB	DMKDSPCH	DMKERMSG	DMKFREE	DMKFRET	DMKIOSQR	DMKQCNWT	DMKSCNFD	DMKSCNMU
	DMKSCNRA	DMKSCNRD	DMKSCNRU	F255	F3	F8	F9	IOBCAW	IOBCC3	IOBFATAL
	IOBIRA	IOBLINK	IOBLOK	IOBRADD	IOBSIZE	IOBSTAT	IOBUSER	LOCK	L2	NORET
	OFF	ON	OPERATOR	PSA	RANGE	RCUADD	RCUBLOK	RCUCACH	RCUDVTBL	RCUOWNER
	RCUSIZE	RCUTYPE	RDEVBLOK	RDEVBPAG	RDEVCUB	RDEVCUP	RDEVDISA	RDEVPPAG	RDEVSTAT	RDEVSTA5
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R7	R8	R9	SAVEAREA	SAVEREGS	SAVEWRK1	SAVEWRK2	SAVEWRK3
	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	SILI	VMBLOK	ZEROES	
DMKCAO	AFREE	AFRET	AP	APSTAT1	APUOPER	AQCNWT	ARIOCU	ARIOUC	BLANK	BLANKS
	DMKCVTBH	DMKCVTHB	DMKERMSG	DMKFREE	DMKFRET	DMKLOKSW	DMKQCNWT	DMKSCNAU	DMKSCNFD	DMKSCNMU
	DMKSCNRA	DMKSCNRU	DMKUDRFU	F3	LOCK	L7	L8	MP	NORET	OWNID
	PSA	RANGE	RCUBLOK	RCUCACH	RCUCHA	RCUDVTBL	RCUOWNER	RCUPRIME	RCUSIZE	RCUSUB
	RCUTYPE	RDEVBLOK	RDEVCUA	RDEVCUB	RDEVCUP	RDEVSTA5	R0	R1	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SAVER11	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6
	SAVEWRK8	SAVEWRK9	TIMEDISP	VMBLOK						
DMKCCD	AP	BADCCW	BADHEDNO	CC	CCSSENSE	CCWBD2	CCWBD4	CCWCLEAR	CCWCTLCM	CCWFORC
	CCWGEN	CCWINV	CCWMAN2	CCWNOOP	CCWNXT	CCWNX11	CCWNX13	CCWSRCH3	CCWTC	CCWUSIDA
	CD	CHEKISAM	CLASDASD	CLASFBA	C1	DENDCC	DESTRTCC	DEXTENT	DMKCCOCH	DMKCCSEN
	DMKCCSLK	DMKCCSRM	DMKCCWCN	DMKCCWRT	DMKSYSPC	DMKTRKVA	DRHA	DWHA	FIRSTRCW	FTR35MB
	F10	F15	F16	F2	F3	F4	F4095	F8	F9	HADISAM
	HADRCGEN	IDA	INVCCW	INVCCW1	IOBALTSK	IOBCYL	IOBLOK	IOBSTAT	LOCK	LOCR
	LOCRCCHH	LOCSCCHH	LOCSRR	L4	MEMO1	MEMO2	MEMO3	MP	NEEDSEEK	PRECCW
	PRECTL	PREFLAG	PRVFLAG	PSA	RCWADDR	RCWCCNT	RCWCCW	RCWCNT	RCWCOMND	RCWCTL
	RCWFLAG	RCWGEN	RCWHMR	RCWISAM	RCWPNT	RCWRCNT	RCWREL	RCWTASK	RCW2311	RDEVBLOK
	RDEVCKDX	RDEVECKD	RDEVFTR	RDEVLOW	RDEVMDL	RDEVMD02	RDEVSTA3	READ	R0	R1
	R10	R11	R12	R13	R14	R15	R2	R3	R4	R6
	R7	R8	SAVEAREA	SAVER10	SEEKOFF	SENSE	SHPSLIM	SILI	SKALTCYL	SKCECYL
	SKIP	SMCOM	SSM	SYSSERV	THISRCW	TYP2305	TYP2311	TYP2314	TYP3330	TYP3340
	TYP3350	TYP3375	TYP3380	VIRCOMND	VIRFLAG	VMBLOK	WCKDNXTK	WRTUKD	WRTUPD	XRIGHT16
	ZEROES									
DMKCCF	AP	CC	CCSBUFCK	CCSCKDON	CCSHNSEN	CCWCLEAR	CCWCTL	CCWFORC	CCWGEN	CCWINV
	CCWMAN2	CCWNOOP	CCWNX11	CCWTC	CCWUSIDA	CD	DMKCCSEN	DMKCCSRM	DMKCCWCN	DMKCCWRT
	DMKSYSPC	FFS	F1	F15	F16	F7	F8	HADRCGEN	IDA	INVCCW1
	IOBCYL	IOBLOK	LOCK	MEMO1	MEMO2	MP	NEEDSEEK	PRVFLAG	PSA	RCWADDR
	RCWCCW	RCWCOMND	RCWCTL	RCWFLAG	RCWGEN	RCWHMR	RCWREL	RCW2311	RDCBLKAP	RDCBLKFA

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)								
	RDCBLKMA	RDCBLOK	RDEVBLOK	RDEVRDC	R0	R1	R10	R11	R12	R13
	R14	R15	R3	R4	R6	R7	R8	SAVEAREA	SAVER10	SKIP
	SYSSERV	VIRCOMND	VMBLOK	XRIGHT16	ZEROES					
DMKCCH	ACRLOCK	ADSPCH	AEXTSP	AFREE	AFREEP	AFRET	ALARM	ALOKSP	ALOKSY	AMCHAREA
	AP	APSTAT1	APSTAT4	APTRAN	APUOPER	AQCNWT	ARIOCH	ARIOCT	ARIOCU	ARIODV
	ASYSVM	AVMREAL	BRING	CCC	CCCPUID	CCDEVTYP	CCHADDR	CCHANID	CCHCAV	CCHCHCUA
	CCHCLOGL	CCHCPEX	CCHCUA	CCHHIO	CCHINTB	CCHIOH	CCHLOG45	CCHLOG80	CCHLOG81	CCHRCV
	CCHREC	CCHSIOB	CCHSIZE	CCHSIZE1	CCHSNSB	CCHTIO	CCPROGID	CCRECTYP	CCSW1	CCSW4
	CCS1PASS	CCS4HARD	CCS4SOFT	CDC	CLASGRAF	CLEAR	COMPSYS	CONTROL	CPCCHLK	CPEXADD
	CPEXBLOK	CPEXREGS	CPEXR10	CPEXR11	CPEXR13	CPEXSIZE	CPID	CPSTAT5	CPUID	CSW
	C1	DEFER	DEVCCH	DMKACRCT	DMKCVTBH	DMKDSPCH	DMKDSPRU	DMKFREE	DMKFREEP	DMKFRET
	DMKIOECC	DMKLOKDF	DMKLOKIO	DMKMCHST	DMKOPRWT	DMKPSAPO	DMKPTRAN	DMKQCNWT	DMKSCNVU	DMKSTKCP
	DMKSYSRM	DUMPSAVE	ECSWBYT3	ECSWLOG	FAILADD	FAILCCW	FAILCSW	FAILECSW	FFS	FXDLOG
	F0	F1	F16	F2	F255	F4096	F7	HIOCCH	IFCC	IGPRGFLG
	IGTERMSQ	IGVALIDB	INTERCCH	INTTIO	IOBCAW	IOBCCH	IOBCP	IOBCSW	IOBFATAL	IOBFLAG
	IOBHIO	IOBHVC	IOBIOER	IOBLOK	IOBOERR	IOBRADD	IOBRCAW	IOBRSTRT	IOBSPEC	IOBSPEC3
	IOBSTAT	IOBTIO	IOBUNSL	IOBUSER	IOELPNTR	IOERBLOK	IOERB80	IOERCCH	IOERCCRA	IOERCCRL
	IOERCCUA	IOERCHID	IOERCLOG	IOERCPID	IOERCSW	IOERECSW	IOEREXT	IOERLG45	IOERLOGL	IOERSIZE
	IOERS80	IOER2860	IOER2870	IOOPSW	IPUADDR	IPUADDRX	LOCK	LPUADDR	MCHAREA	MCHMODEL
	MCNPSW	MODEL135	MODEL145	MODEL155	MODEL165	MOD3031	MOD3090	MOD4331	MOD4381	MP
	NORET	OPERATOR	PMAMODE	PMASTAT	PREFIXA	PREFIXB	PROCIO	PROCIPL	PSA	RCHADD
	RCHBLOK	RCHCUTBL	RCHDED	RCHSTAT	RCHSTIDC	RCUADD	RCUBLOK	RCUDVTBL	RDEVADD	RDEVBLOK
	RDEVBUZY	RDEVSTAT	RDEVSTA4	RDEVTYPC	RDIDX	RSRTNPSW	R0	R1	R10	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	SAVEAREA	SAVEREGS	SAVER10	SAVER4	SAVER6	SAVER7	SAVER8	SAVEWRK1
	SAVEWRK2	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	SIGREST	SIGSSS	SIGSTOP	SIOCCH
	TERMCHAN	TERMSYS	TIOCCH	VMBLOK	ZEROES					
DMKCCO	AP	CC	CCSHNSEN	CCSSENSE	CCWBD4	CCWCLEAR	CCWCTLCM	CCWFORC	CCWGEN	CCWIDAL1
	CCWIDASB	CCWINV	CCWNOOP	CCWNXT	CCWNX11	CCWNX13	CCWNX14	CCWNX9	CCWTC	CCWUSIDA
	CD	CHEKISAM	CLASDASD	CLASFBA	CLASSPEC	CLASTAPE	CLASURI	CMDRDC	DIAGCNT	DMKCCSEN
	DMKCCWCN	DMKCCWRT	DMKSYSPC	DWHA	FIRSTRCW	F15	F4	F4095	F9	HADISAM
	IDA	INVCCW	INVCCW1	IOBCYL	IOBFLAG	IOBLOK	IOBRELCU	LOCK	LOCR	LOCRCCHH
	MEMO1	MEMO2	MODNOP	MP	NOP	PRECCW	PRECTL	PREPRD	PSA	RCUBLOK
	RCUOWNER	RCUPRIME	RCUSUB	RCUTYPE	RCWADDR	RCWCCNT	RCWCCW	RCWCNT	RCWCOMND	RCWCTL
	RCWFLAG	RCWISAM	RCWREL	RCWTASK	RDATA	RDEVBLOK	RDEVCMDK	RDEVCRDC	RDEVCUA	RDEVFTR
	RDEVSTA6	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R6	R7	R8	SAVEAREA	SAVER10	SENSE	SENSEA4	SENSEE4
	SENSEPID	SENSE04	SENSE44	SETID	SHPSLIM	SKIP	SMCOM	SNSSUBCT	SNSSUBST	SSM
	SVSEN	SYSSERV	SYSVIRT	TEMPR6	THISRCW	TYPUNSUP	TYP1442R	TYP3330	TYP3340	TYP3350
	TYP3375	TYP3380	TYP3410	TYP3420	TYP3422	TYP3430	TYP3480	TYP3705	VIRCOMND	VIRFLAG
	VIRTUAL	VMBLOK	WCKDNXTK	WRTUKD	WRTUPD	XPAGNUM	XRIGHT16	ZEROES		
DMKCCS	AFREE	AFREEP	AFRET	ALOKSP	ALOKSY	AP	APSTAT1	APUOPER	CC	CCWCLEAR
	CCWCTL	CCWCTLCM	CCWFORC	CCWGEN	CCWINV	CCWMAN2	CCWNOOP	CCWNX1	CCWNX10	CCWNX18
	CCWUSIDA	CD	CLASDASD	CLASFBA	CLASSPEC	CLASTAPE	CLASURO	CMDRDC	CPEXADD	CPEXBLOK
	CPEXREGS	CPEXSIZE	C1	DEFRSENS	DIAGSNS	DMKCCOCH	DMKCCWCN	DMKCCWCW	DMKCCWRT	DMKDSBSD
	DMKDSPRU	DMKFREE	DMKFREEP	DMKFRET	DMKLOKDF	DMKLOKIO	DMKRSFSD	DMKSTKCP	DMKSTKDE	DMKSYSPC
	DMKUNTRS	FFS	FIRSTRCW	FTRRSRL	F4096	IDA	INVCCW	INVCCW1	IOBCSW	IOBLOK
	IOBRESRV	IOBSENSE	IOBSIZE	IOBSPEC2	IOBSPEC3	IOBSPEC5	IOBUNREL	IOERBLOK	IOERDATA	IOERLEN
	LOCK	LOCKSAV	LPUADDR	MEMO2	MEMO3	MP	PRECCW	PREVRCW	PRVFLAG	PSA
	RCWADDR	RCWCCNT	RCWCCW	RCWCOMND	RCWCTL	RCWFLAG	RCWHEAD	RCWHMR	RCWPNT	RCWRCNT
	RCWREL	RCWTASK	RCWVCAW	RCWVCNT	RCW2311	RDEVADD	RDEVALT	RDEVBLOK	RDEVCRDC	RDEVFTR
	RDEVRRES	RDEVSIZE	RDEVSTA2	RDEVSTA4	RDEVSTA6	RDEVTYPC	RDEVTYPE	RESINCHA	R0	R1

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	R10	R11	R12	R13	R14	R15	R2	R3	R4	R5
	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER10	SAVER12	SAVER8	SAVEWRK7
	SENSEE4	SENSEPID	SENSE04	SKIP	SKIPSENS	SNSSUBCT	SNSSUBST	SYSSERV	THISRCW	TYPUNSUP
	TYP2305	TYP3330	TYP3340	TYP3350	TYP3375	TYP3380	TYP3800	TYP38003	UNLOAD	VIRCOMND
	VIRFLAG	VMBLOK	XFERSENS	XRIGHT16	ZEROES					
DMKCCT	AP	AVMREAL	CC	CCSSENSE	CCWCLEAR	CCWGEN	CCWINV	CCWNOOP	CCWNX11	CCWTC
	CD	CIRCLEC	CLASSPEC	CLASTERM	CPSPMODE	CPSTAT2	DMKCCSEN	DMKCCSRM	DMKCCWRT	DMKSCNRU
	DMKSYSPC	F15	F16	F4	F8	HADAPOLL	IDA	LOCK	MEMO2	MEMO3
	MP	PSA	RCWADDR	RCWCCW	RCWCOMND	RCWCTL	RCWFLAG	RCWREL	RDEVBASE	RDEVBLOK
	RDEVSADN	RDEVTYPC	RDEVTYPE	RDEVUSER	R0	R1	R11	R12	R13	R14
	R15	R6	R7	R8	SAVEAREA	SENSE	SENSEIO	SILI	SKIP	SMCOM
	SYSPRIV	TEMPSAVE	TYPTTY	TYP2700	TYP2955	TYP3704	VIRCOMND	VIRFLAG	VIRTDISA	VIRTENAB
	VMBLOK	ZEROES								
DMKCCW	ACORETBL	ADSPCH	AFREE	AFREEP	AFRET	ALOKSY	APSTAT1	APTRAN	APTRLK	APUOPER
	BLANKS	BRING	CC	CCSSENSE	CCWCTL	CCWFIRST	CCWSUBR	CD	CDTIC	CHEKISAM
	CLASDASD	CLASFBA	CLASGRAF	CLASSPEC	CLASTAPE	CLASTERM	CLASURI	CLASURO	CMDRDC	CNTUNTIC
	CORFLAG	CORFLUSH	CORSHARE	CORSWPNT	CORTABLE	CPEXADD	CPEXBLOK	CPEXFPNT	CPEXMISC	CPEXR0
	CPEXR1	CPEXSIZE	CPSHRLK	CPSTAT2	C1	DEFER	DEFRSENS	DEVTABLE	DEXTENT	
	DIAGCNTL	DMKCCDAS	DMKCCDSK	DMKCCFBA	DMKCCFBD	DMKCCODD	DMKCCOMS	DMKCCOTH	DMKCCOTP	DMKCCSEN
	DMKCCSLK	DMKCCSRM	DMKCCTCN	DMKCCTDL	DMKCCTLC	DMKCCTRM	DMKDEXIN	DMKDIBSM	DMKDSPCH	DMKFREE
	DMKFREEP	DMKFRET	DMKISMTR	DMKPMAWD	DMKPSAPO	DMKPTRAN	DMKPTRFR	DMKPTRLK	DMKPTRUL	DMKSCNVD
	DMKSCNVU	DMKSTKCP	DMKSYSCS	DMKSYSRM	DMKSYSRV	DMKUNTFR	DMKVMASH	ENDWORK	FFS	FIRSTRCW
	FTREXTSN	FWDTIC	F0	F1	F15	F16	F2	F240	F3	F4
	F4095	F4096	F7	F8	GT16MEG	HADAPOLL	HADISAM	HADRCGEN	HADUTIC	IDA
	IOBCAW	IOBCLN	IOBCYL	IOBDIAG	IOBFLAG	IOBHVC	IOBLOK	IOBMISC	IOBMISC2	IOBSENSE
	IOBSPEC2	IOBSPEC3	IOBSPEC4	IOBUNREL	IOBWRAP	IOERBLOK	IOEREXT	IOERSIZE	LOCK	LOCR
	LOCWORK	LPUADDR	MEMO1	MEMO2	MEMO3	NEEDSEEK	PAGCORE	PAGSHR	PCIF	PRECCW
	PREPRD	PREVRCW	PROTCAW	PRVCOMND	PRVFLAG	PSA	RCWADDR	RCWCCNT	RCWCCW	RCWCNT
	RCWCOMND	RCWCTL	RCWFLAG	RCWGEN	RCWHEAD	RCWHMR	RCWINVL	RCWIO	RCWPNT	RCWRCNT
	RCWREL	RCWSHR	RCWTASK	RCWVCAW	RCWVCNT	RCW2311	RDCBLKAP	RDCBLOK	RDCLENG	RDCSTART
	RDEVBLOK	RDEVCKDX	RDEVCMDK	RDEVECKD	RDEVFTR	RDEVMDL	RDEVRDC	RDEVSTA3	RDEVSTA6	RDEVTYPC
	RESINCHA	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER1
	SAVER10	SAVER12	SAVER2	SAVER8	SAVER9	SAVEWRK1	SAVEWRK2	SAVEWRK4	SAVEWRK9	SAVREG14
	SENSE	SHRFLAG	SHRLKCNT	SHRNOPRT	SHRTABLE	SILI	SKIP	SKIPSENS	SMCOM	SPMPFX
	STRTNEW	SVSEN	SWPFLAG	SWPPAG	SWPVPAGE	SYSVIRT	TEMPR10	TEMPR14	TEMPR15	TEMPR2
	TEMPR3	THISRCW	TICBLK	TYPSDLC	TYPUNSUP	TYP3210	TYP3277	TYP3278	TYP3330	TYP3851
	UNTICLST	VIRCOMND	VIRFLAG	VIRTDISA	VIRTUAL	VMBLOK	VMSIZE	WRITEOP	XFERSENS	XKEYMODE
	XPAGNUM	XRIGHT16	XRIGHT24	X2048BND	ZEROES					
DMKCDB	ACORETBL	AFREE	AFRET	APSTAT1	APTRAN	APUOPER	AQCNWT	BRING	CORDISA	CORFLAG
	CORTABLE	CPSTAT4	C1	DEFER	DMKCVTBD	DMKCVTBH	DMKCVTDB	DMKCVTHB	DMKCVUFP	DMKDMPTR
	DMKERMSG	DMKFREE	DMKFRET	DMKPSAPO	DMKPTRAN	DMKPTSAD	DMKQCNWT	DMKSCNFD	DMKSYSAP	DMKSYSRM
	DMKVATAB	DMKVFDDP	FFS	F0	F1	F10	F15	F16	F2	F24
	F3	F4	F4095	F4096	F6	INVLD	LOCK	MPUOPER	NORET	PREFIXA
	PREFIXB	PROCIO	PROCIPL	PSA	RANGE	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	SAVEAREA	SAVEREGS	SAVER2	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK8
	SPMPFX	VMBLOK	VMSIZE	XKEYMODE	XPAGNUM	X2048BND	X40FFS	ZEROES		
DMKCDM	ACORETBL	AFREE	AFRET	APSTAT1	APTRAN	APUOPER	AQCNWT	BRING	BUFFER	BUFNXT
	CORDISA	CORFLAG	CORTABLE	CPSTAT4	C1	DEFER	DMKCVTBD	DMKCVTBH	DMKCVTDB	DMKCVTHB
	DMKCVUFP	DMKDMPTR	DMKERMSG	DMKFREE	DMKFRET	DMKPSAPO	DMKPTRAN	DMKPTSAD	DMKQCNWT	DMKSCNFD

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	DMKSYSAP	DMKSYSRM	DMKVATAB	DMKVMASH	DMKVSTPT	FFS	F0	F1	F10	F2
	F24	F3	F4	F4095	F4096	F6	INVLD	LASTUSER	LOCK	MPUOPER
	NORET	PREFIXA	PREFIXB	PROCIO	PROCIPL	PSA	RANGE	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SAVEAREA	SAVEREGS	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5
	SAVEWRK6	SAVEWRK8	SPMPFX	VMBLOK	VMSIZE	XKEYMODE	XPAGNUM	XRIGHT16	X2048BND	X40FFS
	ZEROES									
DMKCDS	ACORETBL	AFREEP	AFRET	ALOKRM	ALOKSP	AP	APSTAT1	APTRAN	APUOPER	AQCNWT
	AVMREAL	BLANKS	BRING	CORDISA	CORFLAG	CORPGPNT	CORSHARE	CORSWPNT	CORTABLE	CPEXADD
	CPEXBLOK	CPEXFPNT	CPEXMISC	CPEXR0	CPEXR14	CPEXR15	CPEXR5	CPEXR7	CPEXSIZE	CPSPMODE
	CPSTAT2	C1	DEFER	DMKATSCF	DMKCVTBH	DMKCVTDB	DMKCVTHB	DMKERMSG	DMKFREEP	DMKFRET
	DMKPAGIO	DMKPGTPG	DMKPSACC	DMKPSAPO	DMKPSASC	DMKPTRAN	DMKPTRWQ	DMKQCNWT	DMKSCNFD	DMKSYSAP
	DMKSYSRM	DMKTRCIT	DMKTRCPB	DMKVATAB	DMKVATBC	DMKVATMD	DMKVFEST	DMKVMASH	EXTMODE	F0
	F1	F15	F16	F2	F4	F4095	F4096	F5	F6	F7
	F8	LOCK	LOCKSAV	LPUADDR	MICBLOK	MICCREGO	MP	MPUOPER	NORET	PAGCORE
	PAGINVAL	PREFIXA	PREFIXB	PROCIO	PROCIPL	PSA	RUNUSER	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SAS	SAVEAREA	SAVEREGS	SAVER2	SAVEWRK1	SAVEWRK2	SAVEWRK3
	SAVEWRK4	SAVEWRK5	SAVEWRK8	SPMPFX	SWPCYL	SWPFLAG	SWPRECMP	SWPTRANS	TEMPR14	TEMPR15
	TRANMODE	TRQBLOK	TRQBVAL	VMBLOK	VMSIZE	WAIT	XPAGNUM	ZEROES		
DMKCFC	ACICMD	ACICODE	ACIFCN	ACIPARMS	ACIRGRP	ACIRUSR	ACISIZE	ACISTCP	AFREE	AFRET
	AP	APSTAT1	APTRAN	APTRLK	AQCNWT	ASYSVM	ATTN	BLANKS	BRING	BUFFER
	BUFNXT	CE	CLASTERM	CPIEXLOG	CPINITD	CPSTAT4	CTCADDR	CTCLASS	CTENTRY	CTFALIAS
	CTFCHANG	CTFLAG	CTFLAST	CTFSUBCM	CTNAME	CTSIZE	CTTRUNC	CTTYPE	C1	DEFER
	DMKALGON	DMKCDBDC	DMKCDBDI	DMKCDMDM	DMKCDMDU	DMKCDSCP	DMKCDSTO	DMKCFDAD	DMKCFDLO	DMKCFGIP
	DMKCFHSV	DMKCFJBE	DMKCFJRQ	DMKCFJSL	DMKCFTRM	DMKCFWEP	DMKCMD	DMKCMDAT	DMKCMDCA	DMKCMDDE
	DMKCMDDG	DMKCMDIG	DMKCMDIN	DMKCMDNO	DMKCMDNR	DMKCMDQA	DMKCMDQG	DMKCMDQP	DMKCMDQQ	DMKCMDQR
	DMKCMDQS	DMKCMDQV	DMKCMDSA	DMKCMDSC	DMKCMDSG	DMKCMDSO	DMKCMDSR	DMKCPBEX	DMKCPBNR	DMKCPBRS
	DMKCPBRW	DMKCPBRY	DMKCPBSR	DMKCPSH	DMKCPSSH	DMKCPTNF	DMKCPVAC	DMKCPVDS	DMKCPVEN	DMKCPYLK
	DMKCPYUL	DMKCQSQC	DMKCSBLD	DMKCSBVL	DMKCSFBS	DMKCSFFL	DMKCSFRP	DMKCSFSP	DMKCSODR	DMKCSOST
	DMKCSPSP	DMKCSQCL	DMKCSQFR	DMKCSQHL	DMKCSTAG	DMKCSUCG	DMKCSUCS	DMKCSVOG	DMKCSVOS	DMKCSVPG
	DMKCSVPS	DMKCSXTG	DMKCSXTS	DMKCVTDB	DMKCVTHB	DMKDEFDG	DMKDEFDS	DMKDIAL	DMKDIBCP	DMKERMCP
	DMKERMSG	DMKFREE	DMKFRET	DMKHVCDE	DMKHVCDG	DMKHVCDU	DMKLNKIN	DMKLOGON	DMKMCCCL	DMKMSGEC
	DMKMSGMA	DMKMSGMG	DMKMSGNH	DMKMSGSM	DMKMSGWN	DMKPEINT	DMKPTRAN	DMKPTRUL	DMKQCNWT	DMKQVMEP
	DMKRPIRA	DMKSCNFD	DMKSNDNH	DMKSPMEP	DMKSPTEP	DMKSWMUS	DMKTRACE	DMKTRPST	DMKVMDEP	DUMP
	FFS	F0	F256	F4	F8	F9	LOCK	MESSAGE	MP	NORET
	NOTIME	PFKCOUNT	PSA	RDEVBLOK	RDEVFLAG	RDEVIDNT	RDEVSNA	RDEVTYPC	RDEVTYPE	RESET
	REWIND	R0	R1	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER1	SAVER2
	SAVER3	SAVER4	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK8	SYSTEM
	TYP2741	VMBLOK	ZERO							
DMKCFD	AFREE	AFRET	APTRAN	AQCNWT	AVMREAL	BLANKS	BRING	CPSPMODE	CPSTAT2	C1
	DEFER	DMKATSCF	DMKCVTBH	DMKCVTHB	DMKERMSG	DMKFREE	DMKFRET	DMKPSASC	DMKPTRAN	DMKQCNWT
	DMKSCNAU	DMKSCNFD	DMKSCNRU	DMKSCNVU	F0	F1	F3	F4	F6	NORET
	PSA	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	SAVEAREA	SAVEREGS	SAVEWRK1	SAVEWRK2
	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK7	VMBLOK	VMSIZE	ZEROES			
DMKCFF	AFREE	AFRET	AP	APSTAT1	APTRAN	APUOPER	AQCNWT	ASYSVM	AVMREAL	BRING
	CPSEGPRT	CPSTAT4	C1	DEFER	DMKBLDRT	DMKCQYRG	DMKERMSG	DMKFREE	DMKFRET	DMKPGSPP
	DMKPGSPR	DMKPGSPS	DMKPTRAN	DMKQCNWT	DMKRPAGT	DMKSYSAP	DMKVMAS1	DMKVMAS2	EXTMODE	FFS
	F0	F1	F15	F16	F2	F255	F256	F3	F4	F4096

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	F7	F8	KEEPSEGS	LOCK	LPUADDR	MICBLOK	MICCREG0	MP	MPFEAT	NEWPAGES
	NORET	OLDVMSEG	PAGACT	PAGBMP	PAGCORE	PAGSHR	PAGSWP	PAGTABLE	PAGTOT	PAGTSWP
	PREFIXA	PROCIPL	PSA	PSWCC2	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVCREGS	SAVDATE	SAVEAREA	SAVEREGS	SAVER13	SAVER6	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK5
	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	SAVFPRES	SAVGREGS	SAVKEYS	SAVNAME	SAVPSW	SAVTABLE
	SAVTIME	SEGINV	SEGPAGE	SEGPROT	SHRBPNT	SHRFLAG	SHRFPNT	SHRNAME	SHRNOPRT	SHRPAGE
	SHRSEGCT	SHRSEGNM	SHRSGPRT	SHRTABLE	SHRTSIZE	SHRUSECT	SWPAPP	SWPCHG1	SWPCYL	SWPFLAG
	SWPFLAG2	SWPKEY1	SWPPAG	SWPRECMP	SWPSHR	SWPTABLE	SWPVM	SYSNAME	TRANMODE	TTSEGCNT
	VMABLOK	VMAFPNT	VMANAME	VMASHRBK	VMASIZE	VMBLOK	VMSIZE	XRIGHT16		
DMKCFG	ADSPCH	AFREEP	AFRET	AP	APSTAT1	APTRAN	APTRLK	APUOPER	ASCHN	ASYSVM
	AVMREAL	BRING	BUFFER	BUFNXT	CDCONF	CLASDASD	CLASFBA	CPEXADD	CPEXBLOK	CPEXMISC
	CPEXREGS	CPEXSIZE	CPSPMODE	CPSTAT2	C1	DEFER	DMKCFFSB	DMKCFPRR	DMKCVTBD	DMKCVTBH
	DMKCVTDB	DMKCVTHB	DMKDMPC2	DMKDSPCH	DMKERMSG	DMKFREEP	DMKFRET	DMKHVDIP	DMKPGSPO	DMKPGSPS
	DMKPGTPG	DMKPGUPR	DMKPMA	DMKPMACD	DMKPTRAN	DMKPTRLK	DMKPTRUL	DMKRPAGT	DMKRPAPT	DMKSCHRT
	DMKSCNFD	DMKSCNRU	DMKSCNVD	DMKSCNVS	DMKSCNVU	DMKSNTBL	DMKSSSMQ	DMKTRCND	DMKVATMD	DMKVMI
	EXTMODE	FFS	F0	F1	F16	F2	F3	F4	F4095	F4096
	F7	IOERETN	IOKEEP	LOCK	LPUADDR	L2	L2048	L4	MICBLOK	MICDASA
	MICEVMA3	MICFSSE	MICPMAMP	MICPMAV	MICPMMSK	MICPMPSA	MICVTMR	MP	MSSPRES	MVSA370E
	NORLSE	PAGCORE	PAGSHR	PAGSWP	PAGTABLE	PMAAVAIL	PMAMODE	PMAON	PMASTAT	PREFIXA
	PREFIXB	PROCIO	PROTBLOK	PROTBUFF	PROTERR	PSA	PSAMSS	PSWCC1	PSWCC2	RDEVBLOK
	RDEVCODE	RDEVFLAG	RDEVOWN	RDEVSER	RDEVTYPC	RDEVTYPE	R0	R1	R10	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	SAVEAREA	SAVEREGS	SAVERETN	SAVER5	SAVER6	SAVEWRK1	SAVEWRK2	SAVEWRK4
	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	SEGINV	SEGPAGE	SEGTABLE	SHRNAME	SHRTABLE
	SPMPFX	SWPCYL	SWPFLAG	SWPRECMP	SYSNAME	SYSTEM	TRANMODE	TRQBSIZE	TYP2314	TYP3330
	TYP3350	TYP3375	TYP3380	VMBLOK	VMSIZE	VPMAAVAI	XPAGNUM	XRIGHT16	X40FFS	ZEROES
DMKCFH	AFREE	AFRET	AP	APTRAN	APTRLK	AQCNWT	ASYSVM	BRING	CLASFBA	CPSTAT4
	C1	C14	C15	C2	DEFER	DMKCVTBH	DMKCVTDT	DMKERMSG	DMKFREE	DMKFRET
	DMKHVDIP	DMKPGUVG	DMKPGUVR	DMKPTRAN	DMKPTRUL	DMKQCNWT	DMKQCORD	DMKRPAGT	DMKRPAPT	DMKSCNFD
	DMKSCNVS	DMKSCNVU	DMKSNTBL	DMKVMAS1	EDIT	ERRMSG	EXTMASK	F0	F1	F2
	F256	F3	F4	F4096	F6	F8	IOERETN	IOKEEP	LOCK	MP
	NORET	NOTRESP	PSA	RDEVBLOK	RDEVCODE	RDEVCUP	RDEVFLAG	RDEVOWN	RDEVSTA5	RDEVTYPC
	RDEVTYPE	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVCREGS	SAVDATE	SAVEAREA
	SAVEREGS	SAVEWRK1	SAVEWRK2	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	SAVFPRES
	SAVGREGS	SAVKEYS	SAVNAME	SAVPSW	SAVTABLE	SAVTIME	SAVUSER	SEGINV	SHRBPNT	SHRFPNT
	SHRNAME	SHRTABLE	SWPFLAG	SWPKEY1	SYSNAME	SYSTEM	TYP2305	TYP2314	TYP3330	TYP3340
	TYP3350	TYP3375	TYP3380	UCASE	VMBLOK	VMSIZE	XKEYMODE	XPAGNUM	ZEROES	
DMKCFJ	AFREE	AP	DMKCFMAT	DMKCFMWU	DMKCVTAB	DMKCVTDB	DMKCVTHB	DMKERMSG	DMKFREE	DMKSCHRT
	DMKSCHST	DMKSCNFD	DMKTRCIT	DMKTRCPB	F1	F2	F3	F6	F60	LOCK
	MP	PSA	RUN	R0	R1	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	SAVEAREA	SAVEREGS	SAVERETN	SAVEWRK2	TRQBIRA
	TRQBLOK	TRQBSIZE	TRQBTOD	TRQBUSER	TRQBVAL	VMBLOK				
DMKCFM	ADSPCH	AFREE	AFREEP	AFRET	AP	APSTAT1	APTRAN	AQCNWT	ASYSVM	ATTN
	BALRSAVE	BLANKS	BOXBLOK	BRING	BUFCNT	BUFFER	BUFINLTH	BUFNXT	BUFSIZE	CLASGRAF
	CLASTERM	CONFSOP	CONFSS	CONPNT	CONTASK	CPEXADD	CPEXBLOK	CPEXREGS	CPEXR12	CPEXR8
	CPEXSIZE	CPINITD	CRTFSII	CRTFSSA	C1	DEFER	DMKBOXNS	DMKCFCMD	DMKDSPB	DMKDSPCH
	DMKERMCP	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKGRDIC	DMKGRTDT	DMKPTRAN	DMKQCNWT	DMKQCORD
	DMKSCHRT	DMKSCNFD	DMKSTKCP	DMKVIOMK	EDIT	ERRMSG	FFS	F0	F255	HILIGHT
	IOMASK	LOCK	MP	NICATRB	NICBLOK	NICDTYPE	NICD3277	NICD3279	NICSIZE	NOAUTO

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)								
	NORET	NOTIME	NOTRESP	PSA	RDEVAIRA	RDEVBLOK	RDEVCON	RDEVFLAG	RDEVIDNT	RDEVNICL
	RDEVSNA	RDEVSNRB	RDEVTYPC	RDEVTYPE	R0	R1	R11	R12	R13	R14
	R15	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA
	SAVEREGS	SAVER11	SAVER2	SAVEWRK2	SAVEWRK6	SNARBLOK	SNARFG1	SNARFG3	SNARILER	SNARPRMT
	SNARVMB	SYSTEM	TEMPR0	TEMPR1	TREXLOCK	TREXT	TREXTERM	TRQBCPQ	TRQBFLG2	TRQBLINA
	TRQBLOK	TRQBSIZE	TRQBTOD	TRQBVAL	TYPBSC	TYP2741	TYP3277	TYP3278	TYP3279	UCASE
	VCONANF	VCONCBRK	VCONCTL	VCONDIAG	VCONOPT	VCONSCRN	VCON3270	VMBLOK	WAIT	ZEROES
DMKCFO	ADSPCH	AFREEP	AP	APSTAT1	APUOPER	ASYSVM	BLANKS	BUFFER	BUFNXT	CPASTAVL
	CPASTON	CPCREG6	CPDASAAV	CPDASAON	CPEXADD	CPEXBLOK	CPEXR11	CPEXR2	CPEXR5	CPEXSIZE
	CPMICAVL	CPMICON	CPSTAT2	CPSTAT3	CP370EAV	CP370EON	C6	DMKCFMBK	DMKCFPRR	DMKCVTBH
	DMKCVTDB	DMKDSPCH	DMKERMSG	DMKFREEP	DMKLOKSW	DMKPTRRC	DMKSCNAU	DMKSCNFD	DMKSTKCP	DMKSTKOP
	DPMICON	DPSTAT	FF	F2	F256	F3	F4	F5	F8	IPUADDR
	IPUADDRX	LOCK	MICBLOK	MICDASA	MICEVMA3	MICFSSE	MP	PSA	R0	R1
	R10	R11	R12	R13	R14	R15	R2	R3	R4	R5
	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER11	SAVEWRK1	SAVEWRK2	SAVEWRK4
	SAVEWRK5	SAVEWRK7	SAVEWRK8	SVMNOUPD	TIMEDISP	VMBLOK	ZEROES			
DMKCFP	AEXTSP	AFREE	AFRET	AP	APSTAT1	APUOPER	AQCNWT	ARIOCT	ASYSVM	AVMREAL
	BIOBLOK	BIONEXT	BIORESET	BIOSTAT1	CPRUN	CPSPMODE	CPSTATUS	CPSTAT2	DELPAGES	DMKBLDRL
	DMKBLDRT	DMKCFQRD	DMKCVTBH	DMKFREE	DMKFRET	DMKHPTRE	DMKLOCKD	DMKLOCKQ	DMKMHCRE	DMKPGSPO
	DMKPGSPP	DMKPMAEX	DMKPTRUL	DMKPTSPW	DMKQCNWT	DMKQCOSY	DMKSCHRT	DMKSCNVD	DMKTRCPB	DMKVATBC
	DMKVDREL	DMKVMCUA	FFS	F1	F16	F2	F4096	IPUADDR	KEEPSEGS	LOCK
	LPUADDR	MICBLOK	MICCREG0	MICEVMA2	MICSPT	MP	NEWSEGS	OLDVMSEG	OPERATOR	PGBLOK
	PGBSIZE	PGPNT	PROBSTRT	PROBTIME	PROCIO	PSA	RCHADD	RCHBLOK	RCHDED	RCHSTAT
	RUNUSER	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVEWRK1
	SIGCR	SIGSENSE	SPMPFX	TRQBFPNT	TRQBLOK	TRQBQUE	TRQBVAL	VMBLOK	VMSIZE	WAIT
	XRIGHT24	ZEROES								
DMKCFQ	ACORETBL	ADSPCH	AFREE	AFREEP	AFRET	ALOKDS	ALOKSP	AP	APSTAT1	APTRLK
	APUOPER	ASYSVM	AVMREAL	BIOBIRS	BIOBLOK	BIONEXT	BIORESET	BIOSTAT1	BIOVDEVA	CC
	CLASDASD	CLASFBA	CLASGRAF	CLASSPEC	CLASTAPE	CLASTERM	CLASURI	CLASURO	CONACTV	CONFSOP
	CONFSS	CONPNT	CONRESET	CONRESP	CONRETN	CONSTAT	CONTASK	CONUSER	CPEXBLOK	CPEXR0
	CPEXR2	CPEXR8	CPEXR9	CPEXSIZE	CPSPMODE	CPSTAT2	CRTEXT	CRTEXTSZ	CUE	DMKBIORS
	DMKCFREP	DMKCVTAB	DMKCVTBH	DMKDIBDR	DMKDSPCH	DMKDSPRQ	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET
	DMKIOEMI	DMKIOSCB	DMKIOSQR	DMKLOKIO	DMKLOKSW	DMKPTRLK	DMKPTRUL	DMKPTTFT	DMKQCOCL	DMKSCHRT
	DMKSCHST	DMKSCNRD	DMKSSUCF	DMKSTKCP	DMKSYSTV	DMKVCBRD	DMKVIOMK	DMKVSUCO	DMKVSUCR	FFS
	F0	F1	F255	F4	F8	IDAENTRY	IDAEWRD1	IDAHEAD	IDAHPAGE	IDAHSIZD
	IDAHSTRT	IDAHWRK1	IOBBPNT	IOBCAW	IOBCC3	IOBCUBSY	IOBFATAL	IOBFPNT	IOBIOER	IOBIRA
	IOBLOK	IOBMISC	IOBMISC2	IOBSIZE	IOBSTAT	IOBUSER	IOBVADD	IOERBLOK	IOEREXT	IOERPNT
	IOERSIZE	LOCK	LOCKSAV	LPUADDR	MIHDE	MIHMISC	MIHMSG	MIHMSGCS	MIHMSGDV	MIHMSGDW
	MIHMSGID	MIHMSGLN	MP	MSSPRES	NICATRB	NICBLOK	NICQPNT	NICSIZE	PMAMODE	PMASTAT
	PSA	PSAMSS	RCUBLOK	RCUCUBSY	RCUPRIME	RCUSUB	RCUTYPE	RDEVAIOB	RDEVAIRA	RDEVBLOK
	RDEVBUZY	RDEVCLAS	RDEVCON	RDEVCSW	RDEVCUA	RDEVCUB	RDEVDED	RDEVFIOB	RDEVFTR	RDEVIOER
	RDEVNICL	RDEVNRDY	RDEVPIOB	RDEVPS	RDEVQBSY	RDEVQIOB	RDEVSCHD	RDEVSNA	RDEVSTAT	RDEVSTA3
	RDEVSTA4	RDEVTYPE	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS
	SAVERETN	SAVER11	SAVER12	SAVER4	SAVER8	SAVEWRK1	SAVEWRK2	SAVEWRK6	SAVEWRK7	SAVEWRK8
	SAVEWRK9	SILI	SYSVIRT	TIMEDISP	TRQBCHIO	TRQBCPQ	TRQBFLG2	TRQBIRA	TRQBLOK	TRQBSAVE
	TRQBSIZE	TRQBTOD	TRQBUSER	TRQBVAL	TWOENTS	TYPCTCA	TYPSS	TYP3210	TYP3215	TYP3330
	TYP3851	VCONANF	VCONANF2	VCONBFSZ	VCONBUF	VCONCTL	VCONFSS	VCONNCB	VCONNICB	VCONOPT
	VCONRBUF	VCONRDEV	VCONRDSZ	VCONREMD	VCONREMF	VCONWBSZ	VCONWBUF	VIRTUAL	VMBLOK	VMGENIO
	ZEROES									

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
DMKCFR	ADSPCH	AFREE	AFREEP	AFRET	ALOKSP	AP	APSTAT1	APTRLK	APUOPER	ASYSVM
	CLASDASD	CLASFBA	CLASGRAF	CLASSPEC	CLASTAPE	CLASTERM	CPEXADD	CPEXBLOK	CPEXFPNT	CPEXMISC
	CPEXR0	CPEXR11	CPEXSIZE	CRTEXT	CRTEXTSZ	DE	DMKCVTAB	DMKCVTBH	DMKDIBDR	DMKDSPCH
	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKHPSHT	DMKIOEMI	DMKIOSHA	DMKIOSQR	DMKIOSRW	DMKLOKIO
	DMKLOKSW	DMKPTRLK	DMKPTRUL	DMKSCHRT	DMKSCHST	DMKSCNRD	DMKSCNVD	DMKSTKCP	DMKSTKIO	DMKUNTFR
	DMKVIOIN	FFS	F1	F15	IOBCAW	IOBCSW	IOBFLAG	IOBHIO	IOBHVC	IOBIOER
	IOBIRA	IOBLINK	IOBLOK	IOBMID	IOBMISC	IOBMISC2	IOBMSG	IOBRES	IOBRREL	IOBSIZE
	IOBSPEC	IOBSPEC3	IOBSPEC4	IOBSPEC5	IOBTIO	IOBUNSL	IOBUSER	IOBVADD	IOERBLOK	IOEREXT
	IOERSIZE	LOCK	LOCKSAV	LPUADDR	MIHDE	MIHMSG	MIHMSGCS	MIHMSGDV	MIHMSGDW	MIHMSGID
	MIHMSGLN	MP	PSA	RCWCCNT	RCWCCW	RCWHEAD	RCWRCNT	RCWTASK	RDEVAIOB	RDEVBLOK
	RDEVCSW	RDEVIOER	RDEVMID	RDEVPS	RDEVRRES	RDEVSTA3	RDEVSTA4	RDEVSTMD	R0	R1
	R10	R11	R12	R13	R14	R15	R2	R3	R4	R5
	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER1	SAVER11	SAVER13	SAVER8
	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK6	SAVEWRK8	SAVEWRK9	TIMEDISP	TRQBIRA	TRQBLOK	TRQBSAVE
	TRQBSIZE	TRQBTOD	TRQBUSER	TRQBVAL	TRQREGO	TYPCTCA	TYPIBM1	TYPTTY	TYP8809	VMBLOK
	ZEROES									
DMKCFS	AFREE	AFRET	AP	APTRAN	APTRLK	AQCNWT	ARIODV	ASCHN	ASLOGON	ASOFF
	ASON	ASYSVM	AVMREAL	BLANKS	BRING	BUFFER	BUFNXT	CLASFBA	CPDASAAV	CPDASAON
	CPSTAT2	CPSTAT3	CPSTAT5	CP370EAV	CP370EON	C1	DEFER	DMKBLDEC	DMKCFPRR	DMKCVTBH
	DMKCVTHB	DMKERMSG	DMKFREE	DMKFRET	DMKPGUVG	DMKPGUVR	DMKPTRAN	DMKPTRUL	DMKQCNWT	DMKRPAGT
	DMKRPAPT	DMKSCNFD	DMKSCNRD	DMKSNTBL	DMKSYSOC	DMKSYSOW	F0	F2	F4	F4096
	F8	IOERETN	IOKEEP	LOCK	LPUADDR	LPUADDRX	MICBLOK	MICCREG	MICCREG0	MICDASA
	MICEVMA	MICEVMA2	MICEVMA3	MICEVMA4	MICFSSE	MICISKE	MICRRBE	MICRSEG	MICSIZE	MICSPT4
	MICSSKE	MICSTBVR	MICSVC4	MICVPSW	MICWORK	MP	NORET	OWNDLIST	OWNDRDEV	OWNDVSER
	PAGPGSWP	PREFIXA	PREFIXB	PSA	PSAEVMA	RDEVBLOK	RDEVCODE	RDEVTYPC	RDEVTYPE	RDIDX
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER2	SAVEWRK1
	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	SAVTABLE	SAVTIME
	SAVUSER	SYSNAME	SYSTEM	TEMPSAVE	TRQBSIZE	TYP2305	TYP2314	TYP3330	TYP3340	TYP3350
	TYP3375	VMBLOK	VMSIZE	XPAGNUM	XRIGHT16	ZEROES				
DMKCFT	AFREE	AFRET	AP	ASYSLC	CLASGRAF	CLASSPEC	CLASTERM	DMKCVTBH	DMKCVTDB	DMKERMSG
	DMKFREE	DMKFRET	DMKGRTPF	DMKQCOSY	DMKSCNFD	DMKSCNVD	DMKSYSCD	DMKSYSES	DMKSYSLD	DMKSYSLE
	DMKTBLGR	DMKUDRFU	DMKUDRMD	DMKUDRRV	DMKVCTER	F1	F2	F24	F255	F3
	F4	F5	F8	LOCK	L2	L3	L4	MP	NICAPL	NICATOF
	NICBLOK	NICDTYPE	NICD3277	NICD3278	NICFLAG	NICLLEN	NICPSUP	NICSIZE	NICTEXT	NICTMCD
	PSA	RDEVADD	RDEVAPLO	RDEVAPLP	RDEVASTB	RDEVATOF	RDEVAVM1	RDEVAVM2	RDEVBLOK	RDEVCORD
	RDEVFLAG	RDEVLLEN	RDEVNICL	RDEVPCHG	RDEVPSUP	RDEVSCRL	RDEVSNA	RDEVSNRB	RDEVTAPL	RDEVTEXT
	RDEVTFLG	RDEVTMCD	RDEVTYPC	RDEVTYPE	RDEVUSC8	RDEVVM2	RDEV3101	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R7	R8	SAVEAREA	SAVEREGS	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK8	SNARBLOK
	SNARVMB	SYSLOCS	TYPBSC	TYPTTY	TYP3277	TYP3278	VCONANF	VCONBRK	VCONCBRK	VCONCTL
	VCONOBRK	VCONOPT	VCONRBUF	VCONRDSZ	VCONSCRN	VCON3270	VMBLOK	X40FFS	ZEROES	
DMKCFU	ACORETBL	AFREE	AFRET	AMCHAREA	AP	APTRAN	AQCNWT	ARIODV	ASYSLC	ASYSVM
	BLANKS	BRING	BUFCNT	BUFFER	BUFNXT	CLASFBA	CLASTAPE	CLASURO	CORSWPNT	CORTABLE
	CPSTAT5	C1	DEFER	DMKCKTSD	DMKCKTUU	DMKCVTAB	DMKCVTBH	DMKCVTDB	DMKCVTDT	DMKCVTHB
	DMKDIDEP	DMKDIDTR	DMKDMPAL	DMKDMPAU	DMKDMPDV	DMKDMPRC	DMKDMPSF	DMKDMPSW	DMKERMSG	DMKFREE
	DMKFRET	DMKIOEIR	DMKLOCK	DMKLOCKD	DMKLOCRD	DMKLOCRQ	DMKMCIMS	DMKPGTDS	DMKPGTDT	DMKPGTVR
	DMKPGUDU	DMKPTRAN	DMKPTSMW	DMKQCNWT	DMKQCORD	DMKRPAPT	DMKSCHPG	DMKSCHRT	DMKSCHST	DMKSCNAU
	DMKSCNFD	DMKSCNPH	DMKSCNRA	DMKSCNRD	DMKSCNRU	DMKSYM	DMKSYSDT	DMKSYSDU	DMKSYSDW	DMKSYSLG
	DMKSYSLW	DMKSYSOW	DMKSYSTM	DMKSYSTV	DMKVSGRI	EDIT	F0	F1	F16	F2
	F3	F4	F5	F6	F60	F7	F8	IOERETN	IOKEEP	IRMAND
	IRMBIT1	IRMBIT2	IRMBLOK	IRMBYT1	IRMBYT2	IRMFLG	IRMLMT	IRMOR	IRMRLADD	IRMSIZE

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)								
	LOCK	LOGMBLOK	LOGMLEN	LOGMNXT	LOGMSEQ	LOGMSIZB	LOGMSIZD	LOGMTXT	MCHAREA	MCHMODEL
	MOD3033	MOD3081	MOD3090	MP	NOAUTO	NORET	OFF	OWNDLIST	OWNDRDEV	PSA
	RCHBLOK	RCUBLOK	RCUDISA	RCUSTAT	RDEVBLOK	RDEVDED	RDEVDISA	RDEVFLAG	RDEVIRM	RDEVPEND
	RDEVPTHS	RDEVSTAT	RDEVSYS	RDEVTYPC	RDEVTYPE	RDEVUSER	RDIDX	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER14	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4
	SAVEWRK5	SAVEWRK7	SAVEWRK8	SFBFILID	SFBLAST	SFBLOK	SFBORIG	SFBSTART	SFBSYSID	SFBUSER
	SWPCYL	SWPFLAG	SWPRECMP	SYSLOCS	SYSTEM	TRQBIRA	TRQBLOK	TRQBSIZE	TRQBTOD	TRQBUSER
	TRQBVAL	TYPPRT	UCASE	VCONCTL	VCONFLG2	VCONOMSG	VMBLOK	ZEROES		
DMKCFV	AFREE	AP	APSTAT1	APTRAN	APUOPER	AQCNWT	ASYSLC	AVMREAL	BLANKS	BRING
	BUFFER	BUFNXT	CPMICAVL	CPSPMODE	CPSTAT2	C1	DEFER	DFRET	DMKCVTBD	DMKCVTDB
	DMKERMSG	DMKFREE	DMKPTRAN	DMKQCNWT	DMKSCNFD	DMKSYSDS	DMKSYSMS	DMKVATAT	DMKVATBC	DMKVATMD
	EXTMODE	FF	F0	F1	F2	F255	F8	LOCK	MP	NORET
	PAGE2K	PREFIXB	PSA	R0	R1	R10	R11	R12	R13	R14
	R15	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA
	SAVEREGS	SAVEWRK1	SAVEWRK2	SAVEWRK4	SAVEWRK5	SAVEWRK7	SYSLOCS	TEMPRO	VMBLOK	ZEROES
DMKCFW	AP	AQCNWT	BUFFER	DMKERMSG	DMKQCNWT	DMKQCOSY	DMKSCNFD	DMKVCTCH	F3	F8
	LOCK	MP	NORET	PSA	RDEVBLOK	RDEVSNA	R0	R1	R10	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	SAVEAREA	SAVEREGS	SAVEWRK8	SCRCMDR	VMBLOK	ZEROES		
DMKCFY	AFREE	AFRET	AP	APSTAT1	APUOPER	BLANKS	BUFCNT	BUFFER	BUFINLTH	BUFNXT
	CLASGRAF	CLASTERM	CPASTAVL	CPMICAVL	CPMICON	CPSTAT2	CPSTAT4	C1	DMKCVTBH	DMKCVTDB
	DMKCVTHB	DMKERMSG	DMKFREE	DMKFRET	DMKSCHRT	DMKSCNAU	DMKSCNFD	DMKTRQ80	DMKUDRFU	DMKUDRMD
	DMKUDRRV	F0	F1	F2	F255	F3	F7	F8	IPUADDR	LOCK
	LPUADDR	LPUADDRX	MICBLOK	MICCREG	MICCREG0	MICEVMA	MICEVMA3	MICISKE	MICRRBE	MICRSEG
	MICSIZE	MICSKYMD	MICSSKE	MICVPSW	MICVTMR	MICWORK	MP	NICBLOK	NICDTYPE	NICD3277
	NICD3278	NICSIZE	PAGPGSWP	PFDATA	PFDCMD	PFDCPYSP	PFDVAL	PFKADDR	PFKCOUNT	PFKDWDS
	PFKFLAG	PFKIMM	PFKLNG	PFKRET	PFKSIZE	PFKTABLE	PFKTBLSZ	PFTAB	PREFIXA	PREFIXB
	PROTATRQ	PROTBLOK	PROTSIZE	PSA	PSAEVMA	RDEVBLOK	RDEVNICL	RDEVTYPC	RDEVTYPE	RETBUF
	RETDLEN	RETL132	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS
	SAVEWRK1	SAVEWRK2	SAVEWRK4	SAVEWRK5	SAVEWRK7	SAVEWRK8	TEMPSAVE	TIMER	TRQBIRA	TRQBLOK
	TRQBSIZE	TRQBUSER	TYPBSC	TYPTTY	TYP3277	TYP3278	VCONBRK	VCONCTL	VCONOBRK	VECUSER
	VMBLOK	XFF	XKEYMODE	ZEROES						
DMKCKD	AP	APSTAT1	ARIOCC	ARIOCH	ARIOCT	ARIOCU	ARIODV	AVMREAL	BALRSAVE	BALR1
	BCTWAIT	CAW	CC	CC0	CC1	CC2	CC3	CKPLOAD	CLASDASD	CLASFBA
	CLASGRAF	CLASSPEC	CLASTAPE	CLASTERM	CPCREG0	CPID	CPSTAT5	CSW	C0	C2
	DMKCKFCN	DMKCKFC2	DMKCKFTE	DMKCKFVR	DMKCKNTB	DMKCKP	DMKCKPMP	DMKCKPRM	DMKSAV	FFS
	FREESAVE	FXDLOG	F0	IOBLOK	IOBRADD	IPUADDR	IPUADDRX	LOCK	MP	MPUOPER
	NICBLOK	NICDISA	NICDISB	NICENAB	NICFLAG	NICLGRP	NICLINE	NICSIZE	NICSTAT	NICTERM
	NICTYPE	PREFIXA	PREFIXB	PRNPSW	PROCIPL	PSA	RCHADD	RCHBLOK	RCHCUTBL	RCUADD
	RCUBLOK	RCUCHA	RCUDVTBL	RCUPRIME	RCUSUB	RCUTYPE	RDEVADD	RDEVAIOB	RDEVAUTO	RDEVBLOK
	RDEVCUA	RDEVCUB	RDEVDED	RDEVDISB	RDEVENAB	RDEVFLAG	RDEVLCEP	RDEVLNCP	RDEVMAX	RDEVNCP
	RDEVNICL	RDEVPROC	RDEVSTAT	RDEVTYPC	RDEVTYPE	RDEVUSER	RDIDX	RSRTNPSW	R0	R1
	R10	R11	R12	R13	R14	R15	R2	R3	R4	R5
	R6	R7	R8	R9	SIGREST	SIGSENSE	SIGSSS	SILI	TYPBSC	TYP3210
	TYP3277	TYP3278	TYP3284	TYP3705	TYP3851	UC	VMBLOK	ZEROES		
DMKCKF	ACCNT	ACCNTERR	ACCTBLOK	ACCTUSER	ACNTBLOK	ACNTCODE	ACNTCONT	ACNTDATA	ACNTDEVC	ACNTDEVM
	ACNTDEVT	ACNTIOCT	ACNTNCYL	ACNTNEXT	ACNTNUM	ACNTPGRD	ACNTRFLG	ACNTRMID	ACNTRMTE	ACNTSTOP
	ACNTTIME	ACNTUSER	ACNTVTIM	ACNTVTT	ACNTVVT	ACTIBLOK	ACTIFLAG	ACTIPCH	ACTISFB	ACTISFCK

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	ADSPCH	AFREEP	ALOCBLOK	ALOCCYL2	ALOCDU	ALOCFLG	ALOCPNT	ALOCRECS	AP	APSTAT1
	APUOPER	ARIODV	ARIOPR	ARIOPU	ARIORD	ASFBACO	ASYSLC	ASYSOP	ASYSVM	BFFCCPD
	BFFENTRY	BFFFSIZE	BFFREAL	BLANK	BLANKLEN	BLANKS	BUFFCV	BUFFDLM	BUFFOPLG	CALLCVT
	CHECKVF	CKPLOAD	CKPTLIST	CL	CLASDASD	CLASFBA	CLASGRAF	CLASTERM	CLASURO	CPEXADD
	CPEXBLOK	CPEXREGS	CPEXSIZE	CPID	CPSTAT5	CPTRAPER	DATE	DEDTEST	DEVCARD	DISCREC
	DMKCKHAC	DMKCKHDL	DMKCKHIN	DMKCKHPR	DMKCKHST	DMKCKHSZ	DMKCKHWM	DMKCKNIO	DMKCKNRB	DMKCKNRD
	DMKCKNTB	DMKCKP	DMKCKPER	DMKCKPFL	DMKCKWHT	DMKCKWSP	DMKCVTBH	DMKDMPAS	DMKDMPC2	DMKDMPRC
	DMKDSPCH	DMKFREEP	DMKIOCVT	DMKIOECL	DMKIOECQ	DMKIOEES	DMKIOEID	DMKIOEIP	DMKIOEIQ	DMKIOEMQ
	DMKIOEMX	DMKIOFEP	DMKIOJBL	DMKOPRWT	DMKPTSAD	DMKRIOCN	DMKRSPAC	DMKRSPCV	DMKRSPDL	DMKRSPDP
	DMKRSPHQ	DMKRSPMN	DMKRSPPR	DMKRSPPU	DMKRSPRD	DMKRSPSF	DMKRSPSP	DMKRSPTR	DMKRSPWA	DMKSCNRD
	DMKSCNRU	DMKSTKOP	DMKSYSCK	DMKSYSDT	DMKSYSFL	DMKSYSHE	DMKSYSHL	DMKSYSHT	DMKSYSLG	DMKSYSOC
	DMKSYSOW	DMKSYSRM	DMKSYSSP	DMKSYSTP	DMKSYSVM	DMKSYSWI	DMKSYSWM	DMKTMRPT	DMKVFRSV	DMKVRR
	DOLOCAL	DORDEV	DOTERMID	DOVMTERM	FFS	FORMBLOK	FORMNTRY	FORMOPER	FORMSEND	FORMUSER
	FREESAVE	F0	F1	F255	F4096	LISTSIZE	LOCK	MP	NICBLOK	NICDTYPE
	NICMDL	NICSIZE	NICTMAT	NICTYPE	NOAI	NOCARRY1	NOTMID	OWNDLIST	OWNDRDEV	PREFIXA
	PREFIXB	PSA	RDEVADD	RDEVALLN	RDEVBLOK	RDEVCODE	RDEVMDL	RDEVNICL	RDEVPS	RDEVSNRB
	RDEVSTA3	RDEVTMAT	RDEVTYPC	RDEVTYPE	RDIDX	RECBLOK	RECBUFF	RECCYL	RECDUMP	RECFLG
	RECFSZ	RECFULL	RECMAX	RECPNT	RECUSED	REMOTE	RETRY	R0	R1	R10
	R11	R12	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	SAVEVAC	SETTDK	SETUP	SETUP1	SETUP2	SFBCLAS	SFBCONTO	SFBCOPY
	SFBDEST	SFBDIST	SFBFIRST	SFBFLAG2	SFBFLAG5	SFBLAST	SFBLOK	SFBOFORM	SFBORIG	SFBPNT
	SFBPURGE	SFBRECNO	SFBRECS	SFBSIZE	SFBSTART	SFBSYSID	SFBUFORM	SFBUSER	SHQBSIZE	SKIPADD
	SNACARD	SNARBLOK	SNARLUN	SNAUSER	SPLINK	SPNXTPAG	SPPREPAG	STARTIME	STCODE	SYSLOCS
	TEMPSAVE	TERMEXIT	TERMRMID	TODATE	TRAPDATA	TYPBSC	TYPPRT	TYPPUN	TYP3210	USERCARD
	VACOK	VACOVFR	VCONCTL	VCONNICB	VCONRDEV	VECOPVF	VECSTAT	VECUSER	VMBLOK	VSPLCTL
	VSPSFBLK	ZERO	ZEROES							
DMKCKH	ALARM	ALPRTBLK	AP	ARIODV	ATTN	AUTOIPL	AVMREAL	BALRSAVE	BUFFCV	BUFFDLM
	BUFFSTAT	BUFFVER	BUSY	CAW	CC	CC0	CC1	CC2	CC3	CE
	CKPLOAD	CL	CLASFBA	CLASURI	CLASURO	CPSTAT5	CSW	CUE	DE	DMKCKDSW
	DMKCKDVA	DMKCKFCV	DMKCKFDL	DMKCKFLG	DMKCKFRU	DMKCKFSF	DMKCKFWA	DMKCKFWM	DMKCKMSG	DMKCKNRB
	DMKCKP	DMKCKPCS	DMKCKPDV	DMKCKPER	DMKCKPFL	DMKCKPIP	DMKCKPRM	DMKCKPSN	DUMPSAVE	D1
	D4	ECKD	FFS	F4096	INTKFLIN	INTTIO	IONPSW	IOOPSW	LOCK	LPRLSPLC
	LPRNEXT	LPRTBLOK	LSPLCTL	LSPLFLG1	LSPLNEXT	LSPPRINT	LSPSFBLK	L3	L4	L5
	L8	MP	POWEROFF	PSA	RCHADD	RCHBLOK	RCUADD	RCUBLOK	RCUCHA	RCUPRIME
	RCUSUB	RCUTYPE	RDEVADD	RDEVBLOK	RDEVCKP	RDEVCLAS	RDEVCUA	RDEVDED	RDEVDELP	RDEVDISA
	RDEVDRAN	RDEVEXTN	RDEVFLAG	RDEVFSEP	RDEVIMAG	RDEVPRFG	RDEVSEP	RDEVSPL	RDEVSTAT	RDEVTYPC
	RDEVTYPE	RDEVUSER	RDEVXSEP	RDIDX	RECBUFF	RSPLCTL	RSPSFBLK	RSPXBLOK	RSPXDEST	RSPXFCB
	RSPXFLAG	RSPXFORM	RSPXINDX	R0	R1	R10	R11	R12	R14	R15
	R2	R3	R4	R5	R6	R8	R9	SFBDATE	SFBFLAG	SFBFLAG2
	SFBFNAME	SFBFTYPE	SFBLAST	SFBLOK	SFBOFORM	SFBPNT	SFBPURGE	SFBRECER	SFBRECS	SFBSIZE
	SFBSTART	SFBSYSID	SILI	SKIP	TEMPSAVE	TYPPUN	TYP2314	TYP3330	TYP3340	TYP3350
	TYP3375	TYP3380	TYP3800	TYP38003	TYP38008	UC	VMBLOK	ZERO		
DMKCKM	ALARM	AP	APSTAT1	ARIOCH	ARIOCT	ARIOCU	ARIODV	ASCHN	AVMREAL	BLANK
	CC0	CC1	CC2	CC3	CKPLOAD	CLASDASD	CLASFBA	CPID	CPSPMODE	CPSTAT2
	CPSTAT4	CPSTAT5	CPXSTOR	C1	C14	C15	C2	DATE	DMKCKDCP	DMKCKDSW
	DMKCKFAD	DMKCKFOC	DMKCKFOW	DMKCKFVM	DMKCKFVR	DMKCKFWT	DMKCKNBC	DMKCKNBH	DMKCKNEC	DMKCKNEH
	DMKCKNIO	DMKCKNPS	DMKCKNPW	DMKCKNPX	DMKCKNPY	DMKCKNPZ	DMKCKNRD	DMKCKP	DMKCKPMP	DMKCKPVR
	FREESAVE	F1	F4	F4096	F60	LOCK	MESSAGE	MP	NOADD	NOTIPL
	OWNDLIST	OWNDRDEV	OWNDVSER	PAGCORE	PAGINVAL	PROCIPL	PSA	RCHBLOK	RCHCUTBL	RCUBLOK
	RCUDVTBL	RCUPRIME	RCUSUB	RCUTYPE	RDCBLOK	RDCPAGAP	RDEVBLOK	RDEVCODE	RDEVCUP	RDEVECKD
	RDEVMDL	RDEVPRDV	RDEVQUED	RDEVRDC	RDEVSTA5	RDEVTYPC	RDIDX	RECBLOK	RECCYL	R0
	R1	R11	R12	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SAVCREGS	SAVDATE	SAVFPRES	SAVGREGS	SAVKEYS	SAVNAME	SAVPSW

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)								
	SAVTABLE	SAVTIME	SAVUSER	SEGINV	SEGMIG	SEGPAGE	SEGTABLE	SPM	SSBCYL	SSBENTRL
	SSBENTRY	SSBENUM	SSBHEADL	SSBLOK	SWPCYL	SWPFLAG	SWPKEY1	SWPRECMP	SYSNAME	TODATE
	VMBLOK	VMSIZE	XKEYMODE	ZEROES						
DMKCKN	ALARM	AP	ARIODV	CAW	CC	CC0	CC1	CC3	CE	CKPLOAD
	CLASFBA	CPSTAT5	CSW	DE	DMKCKDGP	DMKCKFOC	DMKCKFOW	DMKCKFTR	DMKCKMSG	DMKCKP
	DUMPSAVE	D4	D8	F255	IDA	INTTIO	IONPSW	LOCK	L3	L5
	L8	MP	OWNDLIST	OWNDRDEV	OWNDVSER	PSA	RCHADD	RCHBLOK	RCUADD	RCUBLOK
	RCUCHA	RCUPRIME	RCUSUB	RCUTYPE	RDEVADD	RDEVBLOK	RDEVCUA	RDEVCUP	RDEVDISA	RDEVECKD
	RDEVMDL	RDEVNAME	RDEVPRDV	RDEVQUED	RDEVSTAT	RDEVSTA5	RDEVTYPC	RDEVTYPE	RDIDX	R0
	R1	R11	R12	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SILI	TYP2305	TYP2314	TYP3330	TYP3340	TYP3350	TYP3375
	TYP3380	VMBLOK								
DMKCKP	ACCNT	ACCNTERR	ALARM	AP	ARSPPR	AVMREAL	CAW	CC	CC0	CC2
	CC3	CE	CKPLD	CKPLOAD	CLASDASD	CPID	CPSTAT5	CPTRAPER	CPUID	CSW
	C0	C2	C6	DE	DMKCKDEV	DMKCKDSW	DMKCKFDV	DMKCKFID	DMKCKFIL	DMKCKFLI
	DMKCKFLS	DMKCKFRC	DMKCKFTP	DMKCKFWA	DMKCKHDC	DMKCKHDL	DMKCKHIO	DMKCKMDV	DMKCKMSG	DMKCKMSV
	DMKDMPCP	DMKERPCP	DMKSAV	DMKSAVRS	ECKD	FFS	FREESAVE	F0	F4096	INTKFLIN
	INTPR	INTSVCL	INTTIO	IONPSW	IPLPSW	LOCK	MCNPSW	MP	OLDKEYOP	PMAMODE
	PMASTAT	POWEROFF	PRNPSW	PROPSW	PSA	PSAPGID	RDEVECKD	READ	R0	R1
	R10	R11	R12	R14	R15	R2	R3	R4	R5	R6
	SEGPAGE	SEGTABLE	SIGIPR	SIGSENSE	SILI	SKIP	TEMPR12	TEMPR2	TEMPR4	TEMPR6
	TEMPSAVE	TYP3375	TYP3380	VMBLOK						
DMKCKR	ACTSFB	AFREE	AP	ARSPPR	ARSPPU	BRING	CLASFBA	DMKERMSG	DMKFREE	DMKFRTRS
	DMKMCHST	DMKPGUVG	DMKPGUVR	DMKPTRUL	DMKQCOSY	DMKRPAGT	DMKRPAPT	DMKRSPCP	DMKRSPCR	DMKRSPCU
	DMKRSPDC	DMKRSPFC	DMKRSPND	DMKRSPPC	DMKRSPSC	DMKRSPST	DMKRSPSW	DMKRSPTP	DMKSCNRU	DMKSYSCO
	DMKSYSWI	DMKVSDAD	DMKVSETR	DMKVSGUM	FFS	FORMBLOK	FORMOPER	FORMUSER	F1	F4096
	LOCK	MP	OPNSFB	PRTCHN	PSA	RDEVBLOK	RDEVCLAS	RDEVDISA	RDEVDRAN	RDEVFLAG
	RDEVSPL	RDEVSTAT	RDRCHN	RHTBLOK	RHTVIRT	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SAVER2	SAVEWRK1	SAVEWRK3	SAVEWRK4	SAVEWRK6	SAVEWRK7	SAVEWRK8
	SFBCOPY	SFBDIST	SFBEOF	SFBFLAG	SFBFLAG2	SFBINUSE	SFBLOK	SFBMON	SFBOFORM	SFBOPEN
	SFBPNT	SFBRECER	SFBRECS	SFBRSTRT	SFBSIZE	SFBSIZEB	SFBSYSID	SFBTUSE	SFBUFORM	SFBUHOLD
	SPUBLOK	SPUIND	SYSTEM	VMBLOK	ZEROES					
DMKCKS	ACTSFB	ADDSFB	AP	BRING	CHGSFB	CLASFBA	CLASURO	DELSFB	DMKCKTSU	DMKERMSG
	DMKLOCKD	DMKLOCKQ	DMKMCHST	DMKPGUVG	DMKPGUVR	DMKQCOSY	DMKRPAGT	DMKRPAPT	DMKRSPDC	DMKRSPPC
	DMKRSPSC	DMKRSPST	DMKRSPSW	DMKRSPTP	DMKSCNRD	FFS	F4096	LOCK	MP	PCHCHN
	PRTCHN	PSA	RDEVBLOK	RDEVCKP	RDEVCLAS	RDEVEXTN	RDEVFSEP	RDEVIMAG	RDEVPRFG	RDEVSTAT
	RDEVTYPC	RDEVTYPE	RDEVXSEP	RDRCHN	RSPXBLOK	RSPXDEST	RSPXFCB	RSPXFLAG	RSPXFORM	RSPXINDX
	R0	R1	R11	R12	R13	R14	R15	R2	R3	R4
	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER0	SAVER1	SAVER2	SAVER8
	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	SFBCKPMP
	SFBFILID	SFBLOK	SFBORIG	SFBRECS	SFBSIZE	SFBSYSID	SHQBLOK	SHQBSIZE	SHQCKPT	SYSTEM
	TYP3800	TYP38003	TYP38008	VMBLOK	ZEROES					
DMKCKT	AFREE	ALOCBLOK	ALOCMAP	ALOCMAX	ALOCNPAG	ALOCRECS	AP	ASYSOP	CLASFBA	DMKCVTBD
	DMKERMSG	DMKFREE	DMKLOCRD	DMKLOCRQ	DMKQCOSY	DMKRSPEC	DMKRSPFC	DMKRSPFL	DMKRSPND	DMKRSPPC
	DMKRSPSC	DMKRSPSM	DMKRSPST	DMKRSPSW	DMKRSPTP	DMKRSP9P	DMKSCNRU	DMKSYSCA	DMKSYSCH	DMKSYSSL
	DMKSYSSP	DMKSYSWI	DMKVSGAI	DUMP	FFS	F2	F255	F256	F3	F4
	F6	LOCK	MP	NOTUSED	PSA	RDEVALLN	RDEVBLOK	RDEVCODE	RDEVSER	RDEVTYPC
	RDEVTYPE	RECBLOK	RECCYL	RECMAP	RECMAX	RECPNT	RECSIZE	RECUSED	RHTBLOK	RHTVIRT
	R0	R1	R11	R12	R13	R14	R15	R2	R3	R4

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	R5	R6	R7	R8	SAVEAREA	SAVEREGS	SAVER0	SAVER1	SAVEWRK1	SAVEWRK2
	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SFBFILID	SFBLOK	SFBSIZE	SFBSIZEB	SFBSYSID	SFBTYPE
	SFBUSER	TYP2314	TYP3330	TYP3350	TYP3375	TYP3380	VMBLOK	VMSIZE		
DMKCKV	ADSPCH	AFREE	AFREEP	AFRET	ALOCBLOK	ALOCNPAG	ALOCRECS	ALOKSP	AP	APSTAT1
	APTRAN	APTRLK	APUOPER	ARIODV	ASYSVM	BRING	CLASFBA	CLASURO	CPEXADD	CPEXBLOK
	CPEXREGS	CPEXSIZE	CPSTAT5	CPUID	C1	DEFER	DMKCKTIN	DMKCKTSD	DMKCVTBD	DMKDSPCH
	DMKDSPNP	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKLOKIO	DMKMCHST	DMKPGUBN	DMKPGUVG	DMKPGUVR
	DMKPTRAN	DMKPTRUL	DMKQCOSY	DMKRPAGT	DMKRPAPT	DMKRSPDC	DMKRSPHQ	DMKRSPND	DMKRSPPC	DMKRSPSC
	DMKRSPST	DMKRSPSW	DMKRSPTA	DMKRSPTC	DMKRSPTP	DMKSCNDC	DMKSCNRU	DMKSNTQN	DMKSTKCP	DMKSYSCO
	DMKSYSOW	DMKSYSWI	DMKWRNSB	FFS	F0	F1	F255	F256	F4096	IOKEEP
	LOCK	LPUADDR	MP	NPRCNT	NPRNAME	NPRPNT	NPRTBL	OPNSFB	OWNDLIST	OWNDRDEV
	OWNDVSER	PCHCHN	PRTCHN	PSA	RDEVALLN	RDEVBLOK	RDEVCLAS	RDEVDISA	RDEVDRAN	RDEVEXTN
	RDEVFLAG	RDEVFSEP	RDEVIMAG	RDEVPRFG	RDEVSTAT	RDEVTYPC	RDEVTYPE	RDEVXSEP	RDIDX	RECBLOK
	RECCYL	RECMAP	RECMAX	RECPNT	RECUSED	RHTBLOK	RHTVIRT	RSPXAUTO	RSPXBLOK	RSPXFCB
	RSPXFLAG	RSPXFOLD	RSPXFORM	RSPXINDX	RSPXSETU	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SAVER2	SAVEWRK1	SAVEWRK3	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8
	SFBDATE	SFBDUMP	SFBFILID	SFBFLAG	SFBFLAG2	SFBLAST	SFBLOK	SFBMISC1	SFBMON	SFBPNT
	SFBRECNO	SFBSIZEB	SFBSTART	SFBTIME	SFBUSER	SHQBSIZE	SPLINK	SPNXTPAG	SPPREPAG	SPRECNUM
	SPTIME	SYSTEM	TEMPR1	TSKBLOK	TSKCCPD	TSKFLAG1	TSKFLAG3	TSKFLAG4	TSKLAST	TSKMISC1
	TSKPBUFF	TSKRECS	TSKSFB	TSKSIZE	TSKSIZEB	TSKSTART	TSKSYSID	TSKTEMP	TSKTIME	TYPPRT
	TYP2314	TYP3330	TYP3350	TYP3375	TYP3380	TYP3800	TYP38003	TYP38008	VMBLOK	XPAGNUM
	ZEROES									
DMKCKW	AP	CKPLOAD	CONTINUE	C1	DMKCKHPR	DMKCKNIO	DMKCKNRB	DMKCKNRD	DMKCKP	DMKCKPER
	DMKRSPHE	DMKRSPHL	DMKRSPHT	DMKRSPSV	DMKRSPWI	F4096	LOCK	LOKSAVE	MP	PAGCORE
	PSA	RHTBLOK	RHTECNT	RHTVIRT	RHXADDR	RHXTABLE	R0	R1	R12	R14
	R15	R2	R3	R4	R5	R6	R7	R8	R9	SAVCNT
	SEGINV	SIGSAVE	SWPCODE	SWPCYL	SWPFLAG	SWPPSTOR	TEMPSAVE	TRANMODE	VMBLOK	
DMKCLK	AEXTSP	ALARM	APSTAT1	APSTAT4	APUOPER	ASYSOP	CPCREG0	CPID	CPSTATUS	CPSTAT3
	CPSUPER	CPTIDLE	CPWAIT	C0	DMKPSADU	DMKQCNWT	DMKQCOSY	EMSPCLKC	EMSPEND	EMSPSYNC
	EMSRCLKC	EMSREC	EMSRSYNC	EXTMODE	FF	IDLEWAIT	IPUADDRX	LOCK	LPUADDR	LPUADDRX
	L8	MCHEK	NOTIME	OPERATOR	PROCIPL	PROCSCHK	PSA	R0	R1	R11
	R12	R13	R14	R15	R2	SAVEAREA	SAVEREGS	SIGCLK	SIGEMS	SIGQUI
	SIGRES	SIGSYNC	SIGXC	SYNCMASK	TIMEDISP	TODSYNC	VMBLOK	WAITSTRT	ZEROES	
DMKCMD	CTFALIAS	CTFLAST	DMKCACOF	DMKCACON	DMKCACQY	DMKCAOWN	DMKCFOSA	DMKCFOSC	DMKCFOSF	DMKCFOSP
	DMKCFOSQ	DMKCFOSR	DMKCFOS3	DMKCFSAC	DMKCFSAP	DMKCFSCC	DMKCFSCP	DMKCFSEC	DMKCFSEM	DMKCFSIM
	DMKCFSIS	DMKCFSLE	DMKCFSMG	DMKCFSNT	DMKCFSPX	DMKCFSRN	DMKCFSSA	DMKCFSSM	DMKCFSVC	DMKCFSVS
	DMKCFSWG	DMKCFS37	DMKCFUDU	DMKCFULO	DMKCFUMI	DMKCFUMO	DMKCFUMW	DMKCFUPA	DMKCFURE	DMKCFVMI
	DMKCFVSB	DMKCFVSM	DMKCFYAG	DMKCFYAS	DMKCFYSA	DMKCFYSC	DMKCFYSM	DMKCFYSP	DMKCQCPT	DMKCQGID
	DMKCQGQA	DMKCQGQC	DMKCQGQD	DMKCQGQG	DMKCQGQH	DMKCQGQL	DMKCQGQS	DMKCQGQT	DMKCQGQU	DMKCQHFG
	DMKCQHFS	DMKCQHNG	DMKCQHNS	DMKCQHRG	DMKCQHRS	DMKCQHTG	DMKCQHTS	DMKCQPQA	DMKCQPQD	DMKCQPQG
	DMKCQPQL	DMKCQPQP	DMKCQPQS	DMKCQPQT	DMKCQPQU	DMKCQQID	DMKCQQQL	DMKCQQQS	DMKCQQQT	DMKCQRAF
	DMKCQRDP	DMKCQRHD	DMKCQRPG	DMKCQRPR	DMKCQRQD	DMKCQSMI	DMKCQSMW	DMKCQSNA	DMKCQSPS	DMKCQSRE
	DMKCQSSC	DMKCQTCL	DMKCQTST	DMKCQUSE	DMKCQUST	DMKCQUTE	DMKCQYCA	DMKCQYCL	DMKCQYCP	DMKCQYLM
	DMKCQYPF	DMKCQYSA	DMKCQYSP	DMKCQYS3	DMKCQYTI	DMKCQYUI	DMKCQYUS	DMKCQYVS	DMKJRLQU	DMKJRLSE
	DMKNEAAH	DMKNEADH	DMKNESDS	DMKNESHD	DMKNESPL	DMKNETDB	DMKNETEN	DMKNETQU	DMKNETVA	DMKNLDR
	DMKNLEMP	DMKPEQRY	DMKSRMQS	DMKSRMSS	DMKTHIDP	DMKTHIFA	DMKTHIIO	DMKTHILO	DMKTHIPA	DMKTHIPO
	DMKTHIQQ	DMKTHIUG	DMKTHIUS	DMKVDAAA	DMKVDAAC	DMKVDDDC	DMKVDDDG	DMKVDDDR	DMKVDDDS	DMKVFCQV
	DUMP	LOAD	OFF	ON	RUN	TIMER				
DMKCNS	ADSPCH	AFREE	AFREEP	AFRET	ALARM	ALOKSP	APSTAT1	APTRAN	APUOPER	ASYSVM

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)								
	ATTN	BALR9	BLANKS	BRING	BUFSIZE	BUSY	CAW	CC	CCC	CD
	CDC	CE	CHC	CLASTERM	CMDREJ	CODE	CONACTV	CONADDR	CONCCW1	CONCCW2
	CONCCW3	CONCCW4	CONCNT	CONCNTL	CONCOMND	CONDATA	CONESCP	CONFLAG	CONFLAGS	CONMORE
	CONOUTPT	CONPARM	CONPNT	CONRESP	CONRETN	CONRTRY	CONSPLT	CONSTAT	CONSYNC	CONTASK
	CONTSIZE	CONTSKSZ	CONUSER	CONWRTRD	CPEXADD	CPEXBLOK	CPEXREGS	CPEXR0	CPEXR10	CPEXSIZE
	CPID	CSW	C1	DATACHK	DE	DEFER	DMKBLDVM	DMKCFMAT	DMKCFMBK	DMKCNTED
	DMKCVTBH	DMKDSPCH	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKIOERR	DMKIOEST	DMKIOSQR	DMKLOKIO
	DMKLOKSW	DMKMSWR	DMKOPEEM	DMKPTRAN	DMKQCOCL	DMKQCOET	DMKQCPTO	DMKSCNEP	DMKSCNRD	DMKSCNRU
	DMKTBLCI	DMKTBLCO	DMKTBLPI	DMKTBLPO	DMKTBMMI	DMKTBMMO	DMKTBMNI	DMKTBMNO	DMKTBNAE	DMKTBNAI
	DMKTBNAO	DMKTBNBE	DMKTBNEA	DMKTBNEB	DMKTRMID	DMKTTXIN	DMKTTYOP	EDIT	F0	F1
	F10	F15	F16	F2	F4	F8	IFCC	IL	INHIBIT	INTREQ
	IOBCAW	IOBCC1	IOBCC3	IOBCSW	IOBERP	IOBFATAL	IOBFLAG	IOBIOER	IOBIRA	IOBLINK
	IOBLOK	IOBMISC	IOBMISC2	IOBRADD	IOBRES	IOBSIZE	IOBSPEC	IOBSTAT	IOBUNSL	IOBUSER
	IOERBLOK	IOERCCRA	IOERCCW	IOERDATA	IOEREXT	IOERFLG3	IOERNUM	IOERREAD	IOERSIZE	LOCK
	LOCKSAV	LOGDROP	LOGHOLD	LPUADDR	LPUADDRX	NOAUTO	PREFIXA	PRGC	PRIORITY	PRTC
	PSA	PSASYSID	RCHBLOK	RCHBUSY	RCHSTAT	RCUBLOK	RCUBUSY	RCUSCED	RCUSTAT	RDEVACTV
	RDEVAIOB	RDEVAPLO	RDEVASTB	RDEVATNC	RDEVATOF	RDEVATSW	RDEVAVM2	RDEVBLOK	RDEVBUZY	RDEVBZCH
	RDEVCON	RDEVCORR	RDEVCTL	RDEVDISA	RDEVDISB	RDEVDROP	RDEVENAB	RDEVEPMD	RDEVFCNS	RDEVFLAG
	RDEVHIO	RDEVIDNT	RDEVLOG	RDEVNDLF	RDEVNOHD	RDEVNRDY	RDEVPCNT	RDEVPREP	RDEVPRFG	RDEVPROC
	RDEVPSUP	RDEVPTTC	RDEVRCNT	RDEVREST	RDEVSADN	RDEVSCHD	RDEVSTAT	RDEVSTA2	RDEVSTA3	RDEVSTA4
	RDEVSYNC	RDEVTFLG	RDEVTMCD	RDEVTTYB	RDEVTYPC	RDEVTYPE	RDEVUSC8	RDEVUSER	RDEVVM2	RESET
	RETRYSW	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVER0	SAVER1
	SAVER2	SILI	SKIP	SM	SVMUNLOK	SYSTEM	TEMPSAVE	TIMEDISP	TRACBEF	TRACCURR
	TRACEND	TRACFLG2	TRACPROC	TRACSTRT	TRACSVCR	TYPTTY	TYPUNDEF	TYP1050	TYP2741	TYP3210
	UC	UCASE	UE	VMBLOK	ZEROES					
DMKCNT	AP	BALRSAVE	BALR3	BALR6	CONADDR	CONCNT	CONPARM	CONTASK	DMKTBLUP	EDIT
	F256	IOBLOK	LOCK	MP	PSA	RDEVBLOK	R0	R1	R10	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	TEMPR0	TEMPSAVE	UCASE	VMBLOK				
DMKCPB	ADSPCH	AFREE	AFREEP	AFRET	AP	APTRAN	AQCNWT	ASYSVM	ATTN	BLANKS
	BRING	CLASSPEC	CLASTAPE	CLASTERM	CLASURI	CLASURO	CPEXBLOK	CPEXMISC	CPEXR0	C1
	DE	DEFER	DEVICE	DMKCFPRR	DMKCFQRD	DMKCVTBH	DMKCVTHB	DMKDSPCH	DMKERMSG	DMKFREE
	DMKFREEP	DMKFRET	DMKIOSQR	DMKIOSRW	DMKPGSPO	DMKPSAPO	DMKPTRAN	DMKQCNWT	DMKSCNFD	DMKSCNVU
	DMKSSGMQ	DMKSTKCP	DMKTRCEX	DMKVATBC	DMKVATMD	DMKVIOMK	EXTMODE	FFS	F0	F1
	F16	F3	F4	F6	IOBCAW	IOBIRA	IOBLOK	IOBMISC	IOBSIZE	IOBUSER
	LOCK	MICBLOK	MICEVMA2	MICSPT	MP	NORET	PSA	RCWCCNT	RCWCCW	RCWHEAD
	RCWTASK	RDEVBLOK	RDEVBUZY	RDEVSTA4	REWIND	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	SAVEAREA	SAVEREGS	SAVERETN	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SILI	SYSTEM	TRANMODE
	TYPCTCA	TYP3210	TYP4245	TYP4248	UE	VMBLOK	X40FFS	ZEROES		
DMKCPE	AP	MP								
DMKCPI	ADSPCH	AEXTSP	AFREE	AFREEP	AFRET	ALARM	AP	APAGCP	APSTAT1	APTRAN
	APTRLK	APUOPER	ARIODV	ARSPPR	ASVCLIST	ASYSOP	ASYSVM	BALRSAVE	BALRSAV2	BCTWAIT
	BLKMPX	BRING	CAW	CCC	CKCMASK	CKPTLIST	CONTINUE	CPASTAVL	CPASTON	CPCREG0
	CPCREG6	CPDASAAV	CPDASAON	CPEXBLOK	CPEXFPNT	CPEXSIZE	CPID	CPINIT	CPMICAVL	CPMICON
	CPSEGPRT	CPSPMODE	CPSTATUS	CPSTAT2	CPSTAT3	CPSTAT4	CPSTAT5	CPSUPER	CPUID	CPULOG
	CPUVERSN	CPWAIT	CP370EAV	CP370EON	CSADDR	CSSFEAT	CSW	C0	C1	C14
	C6	DAMAGRPT	DEFER	DMKAPICK	DMKAPIPR	DMKCCWB1	DMKCCWB2	DMKCCWB3	DMKCCWB4	DMKCCWB5
	DMKCCWB6	DMKCCWB7	DMKCCWB8	DMKCCWGN	DMKCCWL1	DMKCCWL2	DMKCCWL3	DMKCCWL4	DMKCCWL5	DMKCCW0
	DMKCCW1	DMKCLKCK	DMKCPEML	DMKCPEND	DMKCPJNT	DMKCPJST	DMKCVTBH	DMKDGDA0	DMKDGDA1	DMKDGDA2

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)								
	DMKDGDA3	DMKDGDA4	DMKDGDA5	DMKDGDA6	DMKDGDA8	DMKDGDA9	DMKDGDDK	DMKDGFA0	DMKDMPAA	DMKDMPAS
	DMKDMPC2	DMKDMPMA	DMKDMPMP	DMKDMPRC	DMKDMPRY	DMKDMPSA	DMKDMPSD	DMKDSPCH	DMKDSPRQ	DMKDSP0
	DMKDSP1	DMKDSP2	DMKDSP3	DMKDSP4	DMKDSP5	DMKDSP6	DMKERMCP	DMKEXTIN	DMKFREAP	DMKFREE
	DMKFREEP	DMKFRELG	DMKFREMX	DMKFRET	DMKFRE14	DMKFRE15	DMKIOCVT	DMKIOECL	DMKIOECQ	DMKIOEES
	DMKIOEID	DMKIOEIP	DMKIOEIQ	DMKIOEMQ	DMKIOEMX	DMKIOFEP	DMKIOJBL	DMKIOTIN	DMKMCHST	DMKMNTIO
	DMKOPEDC	DMKOPERC	DMKOPRWT	DMKPAGRD	DMKPMAI1	DMKPMAI2	DMKPRGIN	DMKPRWSK	DMKPSADU	DMKPTRAN
	DMKPTRFA	DMKPTRLK	DMKPTRLX	DMKPTRPQ	DMKPTRUL	DMKPTRUX	DMKPTSAD	DMKPTSMW	DMKRIOCN	DMKRIOCU
	DMKRIODV	DMKRIOPR	DMKRIOPU	DMKRIORD	DMKRIORN	DMKRIOSF	DMKRSPPR	DMKRSPSF	DMKSCNRD	DMKSCNRU
	DMKSCNVU	DMKSEGPG	DMKSTANT	DMKSTKOP	DMKSVCIN	DMKSVCNO	DMKSVCNS	DMKSYSAP	DMKSYSFL	DMKSYSHE
	DMKSYSHL	DMKSYSHT	DMKSYSID	DMKSYSMP	DMKSYSNP	DMKSYSOC	DMKSYSOW	DMKSYSRM	DMKSYSSP	DMKSYSSZ
	DMKSYSTP	DMKSYSUD	DMKSYSVL	DMKSYSVM	DMKSYSWI	DMKSYSWM	DMKTMRPT	DMKTODIN	DMKTODTB	DMKUDRBV
	DMKUDROV	DMKUNTFR	DMKUNTF1	DMKUNTRN	DMKVATZP	DMKVATZS	DMKVAUZP	DMKVMASH	DMKVRR	DMKVRRDD
	DMKVRRRC	DPMICON	DPOKTLB	DPSEGPRT	DPSTAT	EXNPSW	EXTMASK	EXTMODE	FASTCPU	FF
	FFS	F0	F1	F2	F20	F3	F4	F4096	F7	HARDSTOP
	IDLEWAIT	IFCC	INTMASK	INTPR	INTREQ	IOERETN	IONTWAIT	IPLCCW1	IPLPSW	IPUADDR
	IPUADDRX	KEYMASK	LANGADDR	LANGBLOK	LANGFLAG	LANGHIGH	LANGLOW	LANGNEXT	LANGNTRY	LANGNTSZ
	LANGPAGE	LANGSIZE	LANGVMB	LANGVMBK	LASTUSER	LOCK	LPUADDR	LPUADDRX	L2	L4
	L6	MCHEK	MFAMASK	MICBLOK	MICDASA	MICDIAG	MICEVMA3	MICISKE	MICLPSW	MICSCSP
	MICSIO	MICSKYMD	MICSTPT	MICSTSM	MICTCH	MP	MPFEAT	MPGEND	MSSFMASK	MSSPRES
	NORET	NOTIME	NOTIPL	OWNDLIST	OWNDRDEV	PAGEWAIT	PAGE4K	PGREAD	PREFIXA	PREFIXB
	PRNPSW	PROBTIME	PROCIO	PROCIPL	PSA	PSACPXBP	PSADSPRQ	PSAEVMA	PSALANG	PSAMSS
	PSAPGID	PSASYSID	PSBCLR2	PSECLR2	PSENDCLR	RDEVADD	RDEVBLOK	RDEVFIOB	RDEVIOBL	RDEVLIOB
	RDEVSIZE	RDEVUSER	RDIDX	RESET	RUNCR0	RUNCR1	RUNUSER	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R8	R9	SAVESIZE	SFBSIZE	SIGIPR	SIGSENSE	SIGSSS	SIGWAKE	SIGXC	SILI
	SPM	SVCNPSW	SYNCLOG	SYSIPLDV	SYSTEM	TEMPR0	TEMPR1	TEMPR2	TEMPR3	TEMPR4
	TEMPR5	TEMPR6	TIMEDISP	TIMER	UC	VMBLOK	VRSVUID	WAIT	XKEYASST	XKEYMODE
	ZEROES									
DMKCPJ	AFREE	AFREEP	AFRET	ALARM	AP	APSTAT1	APUOPER	AQCNWT	ARIODV	ASYNELOG
	ASYNFLOG	ASYSOP	ASYSVM	AVMREAL	BUFSIZE	CLASTERM	CONFGRPT	CPASTAVL	CPEXADD	CPEXBLOK
	CPEXR11	CPEXR12	CPEXSIZE	CPID	CPINITD	CPSTAT2	CPSTAT5	CPUID	CPUMODEL	C14
	DAMAGRPT	DMKCKRSY	DMKCPEML	DMKCPEND	DMKCPICA	DMKCPIF1	DMKCPIMS	DMKCPIOL	DMKCPIWC	DMKCPVAE
	DMKCQHFI	DMKCSOSD	DMKCVTBD	DMKDIDEP	DMKDIDTR	DMKDMPCA	DMKDSPB	DMKERMCP	DMKERMSG	DMKFREE
	DMKFREEP	DMKFREHI	DMKFRELO	DMKFRET	DMKHVDPP	DMKIDUMP	DMKIDUSF	DMKIOEFL	DMKIOGAP	DMKIUACP
	DMKLOKSW	DMKMCHIN	DMKMCHST	DMKNETAE	DMKNLDR	DMKOPEAC	DMKOPELO	DMKQCNWT	DMKQCORD	DMKQCOSY
	DMKRIORN	DMKRSPWA	DMKRSPWC	DMKSCHST	DMKSCNEP	DMKSCNMU	DMKSCNRA	DMKSCNVS	DMKSEGWR	DMKSTKCP
	DMKSTKMP	DMKSTKOP	DMKSTPX	DMKSYSCA	DMKSYSCV	DMKSYSEA	DMKSYSEV	DMKSYSOC	DMKSYSOW	DMKSYSRM
	DMKSYSRV	DMKSYSTE	DMKSYSTS	DMKSYSTV	DMKSYSWA	DMKSYSWV	DMKVRRRS	DMKWRMST	DMKWRMSY	EDIT
	EXTMODE	FFS	F1	F4	F4095	F8	HARDSTOP	IOELPNTR	IOLOG	LOCK
	LPUADDR	LPUADDRX	MCNPSW	MP	NICBLOK	NICDISA	NICNAME	NICSIZE	NICSTAT	NOAUTO
	NORET	NOTIME	OPERATOR	OWNDLIST	OWNDRDEV	OWNDVSER	POWEROFF	PREFIXA	PREFIXB	PROCIO
	PROCIPL	PSA	RDEVAUTO	RDEVBLOK	RDEVDED	RDEVDISA	RDEVFLAG	RDEVMAX	RDEVNICL	RDEVNRDY
	RDEVSTAT	RDEVTYPC	RDEVTYPE	RDIDX	RECOVRPT	R0	R1	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	SSCBADDR	SSCBLOK	SSCIXADD	SSCIXLST	SSCRTADD	SSCRTLST	SSCSIZE	SSCTSFPT
	SYNCLOG	SYSTEM	TEMPR14	TEMPSAVE	TIMEDISP	TIMER	TRQBIRA	TRQBLOK	TRQBSIZE	TRQBTOD
	TRQBUSER	TRQBVAL	TYP2305	TYP2314	TYP3310	TYP3330	TYP3340	TYP3350	TYP3370	TYP3375
	TYP3380	UCASE	VMBLOK	WARNGRPT	XPAGNUM					
DMKCPM	AFREEP	ALOKSP	AP	APSTAT1	APTRAN	APTRLK	APUOPER	ARIOCH	ARIOCT	ARIOCU
	ASYSVM	BALR1	BCTWAIT	BRING	BUSY	CAW	CC	CCC	CLASDASD	CLASSPEC
	CLASTAPE	CLASTERM	CODE	CPEXADD	CPEXBLOK	CPEXFPNT	CPEXREGS	CPEXR11	CPEXR12	CPEXSIZE
	CPSHRLK	CPSTAT2	CPUVERSN	CSW	CUE	C1	DE	DEFER	DMKDSBRD	DMKFREEP
	DMKLOKIO	DMKPTRAN	DMKSCNMU	DMKSCNRA	DMKSTKCP	DMKSTKIO	DMKTAPRL	FFS	FTRFH	F0

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	IFCC	INTREQ	IOBCSW	IOBIRA	IOBLINK	IOBLOK	IOBRADD	IOBSIZE	IOBSPEC	IOBUNSL
	IOBUSER	LOCK	LPUADDR	MP	PREFIXA	PREFIXB	PSA	PSAPGID	RCHBLOK	RCHCUTBL
	RCHPEND	RCHPROC	RCHSTAT	RCUBLOK	RCUCHA	RCUCHD	RCUDISA	RCUDVTBL	RCUPRIME	RCUSTAT
	RCUSUB	RCUTYPE	RDEVADD	RDEVALT	RDEVASGN	RDEVBLOK	RDEVCUA	RDEVDED	RDEVDISA	RDEVFLAG
	RDEVFOFF	RDEVFTR2	RDEVLNKS	RDEVMOUT	RDEVNRDY	RDEVOWN	RDEVPPAG	RDEVPTHS	RDEVPTH1	RDEVPTH2
	RDEVPTH3	RDEVPTH4	RDEVPTH5	RDEVPTH6	RDEVPTH7	RDEVPTH8	RDEVRRES	RDEVSTAT	RDEVSTA2	RDEVSTA4
	RDEVSTA5	RDEVSYS	RDEVTYPC	RDEVTYPE	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK8	SHRLKCNT	SILI
	SM	SYSTEM	TEMPSAVE	TRACBEF	TRACCURR	TRACEND	TRACFLG2	TRACPROC	TRACSTRT	TRACSVCR
	TYPBSC	TYP2305	TYP3340	TYP3380	TYP3480	TYP3851	UC	VMBLOK		
DMKCPN	ADSPCH	AFREE	AFREEP	AFRET	ALOKSP	AP	APSTAT1	APUOPER	AQCNWT	BLANK
	CHANID	CLASDASD	CLASFBA	CLASTAPE	CPEXBLOK	CPEXR0	CPEXR12	CPEXSIZE	CPUVERSN	DMKCPWFB
	DMKCVTBD	DMKCVTBH	DMKDSPCH	DMKFREE	DMKFREEP	DMKFRET	DMKIOSQR	DMKLOKIO	DMKLOKSW	DMKQCNWT
	DMKSCNPH	DMKSCNRA	DMKSCNRN	DMKSCNVU	DMKSCONP	FFS	F1	F2	F4	F8
	IOBASN	IOBASNCT	IOBBADCH	IOBCAW	IOBCC3	IOBDYNP	IOBFATAL	IOBFUNCT	IOBIOER	IOBIRA
	IOBLINK	IOBLOK	IOBMISC	IOBPATHF	IOBPGID	IOBPROC	IOBRADD	IOBSETP	IOBSIZE	IOBSPEC5
	IOBSTAT	IOBUSER	IOBXTRA	IOERBLOK	IOEREXT	IOERSIZE	LOCK	LPUADDR	MP	NORET
	PREFIXA	PSA	PSAPGID	RANGE	RCHADD	RCHBLOK	RCHDED	RCHPROC	RCHSTAT	RCHSTIDC
	RCUBLOK	RCUCHB	RCUDISA	RCUSTAT	RDEVALT	RDEVASGN	RDEVATT	RDEVBLOK	RDEVDED	RDEVDISA
	RDEVFOFF	RDEVNATH	RDEVNRDY	RDEVPTHS	RDEVRRES	RDEVSTAT	RDEVSTA2	RDEVSTA4	RDEVSTA5	RDEVTYPC
	RDEVTYPE	RDEVUSER	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS
	SAVER11	SAVER13	SAVER3	SAVER4	SAVER5	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5
	SAVEWRK7	SAVEWRK8	SAVEWRK9	SILI	TIMEDISP	TYP3480	VMBLOK	ZEROES		
DMKCPO	ACRLOCK	ADSPCH	AEXTSP	AFREE	AFREEP	AFRET	ALOKSP	AMCHAREA	AP	APSTAT1
	APSTAT4	APTRLK	APUOPER	AQCNWT	ARIOCH	ARIOCT	ARIOCU	ASYSVM	AVMREAL	BRING
	CLASDASD	CLASFBA	CLASGRAF	CLASTAPE	CLASTERM	CLASURI	CLASURO	CPEXADD	CPEXBLOK	CPEXREGS
	CPEXR1	CPEXR11	CPEXR12	CPEXSIZE	CPSHRLK	CPSTAT2	CSSFEAT	C1	DEFER	DMKACRCO
	DMKACSCV	DMKCFPRR	DMKCPPUP	DMKCVTAB	DMKCVTBD	DMKCVTBH	DMKDSPCH	DMKERMSG	DMKEXTST	DMKFREE
	DMKFREEP	DMKFRET	DMKLOKIO	DMKLOKSW	DMKMCTAF	DMKPTRAN	DMKPTRUL	DMKQCNWT	DMKRIOPR	DMKRSFSD
	DMKSCHST	DMKSCNEP	DMKSCNMU	DMKSCNRA	DMKSCNRD	DMKSCNRN	DMKSPKDL	DMKSTKCP	DMKSTKMP	DMKSYSOC
	DMKSYSOW	DMKVFRSV	FF	FFS	F0	F1	F2	IOBSIZE	IPUADDR	IPUADDRX
	LOCK	LPUADDR	LPUADDRX	MCHAREA	MCHMODEL	MOD3081	MOD3090	MP	MPUOPER	NOADD
	NORET	OFFLPROC	OWNDLIST	OWNDRDEV	PMAMODE	PMASTAT	POFFLINE	PREFIXB	PRIORITY	PSA
	RCHBLOK	RCHCUTBL	RCHPEND	RCHSTAT	RCUBLOK	RCUCHA	RCUCHD	RCUDVTBL	RCUPRIME	RCUSUB
	RCUTYPE	RDEVADD	RDEVAOF	RDEVBLOK	RDEVBOF	RDEVBUZY	RDEVCUA	RDEVDED	RDEVDELP	RDEVDISA
	RDEVDRAN	RDEVENAB	RDEVEPLN	RDEVFLAG	RDEVLNKS	RDEVPEND	RDEVPTHS	RDEVPTH1	RDEVPTH2	RDEVPTH3
	RDEVPTH4	RDEVPTH5	RDEVPTH6	RDEVPTH7	RDEVPTH8	RDEVRCVY	RDEVRSVD	RDEVSCHD	RDEVSPL	RDEVSTAT
	RDEVSTA4	RDEVSYS	RDEVTYPC	RDEVTYPE	RDEVUSER	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SAVER1	SAVER11	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK7
	SAVEWRK9	SHRLKCNT	SIGEMS	SIGQUI	SIGSTOP	SYSTEM	TIMEDISP	TRQBIRA	TRQBLOK	TRQBSIZE
	TRQBTOD	TRQBUSER	TRQBVAL	TYPBSC	TYP3705	TYP3800	TYP38003	TYP38008	VECAVAIL	VECSAOK
	VECSTAT	VECUSER	VMBLOK	ZEROES						
DMKCPP	ACORETBL	ACTIVTRQ	ADSPCH	AEXTSP	AFREEP	AFRET	ALLCHANS	ALOKVM	AMCHAREA	AP
	APSTAT1	APSTAT2	APSTAT4	APTRAN	APTRLK	APUOPER	AQCNWT	ARIOCH	ARIOCT	ARIOCU
	ARIODV	ASCHN	ASFBACO	ASYSVM	ATMRSN	AVMREAL	BRING	CLASFBA	CORFLAG	CORIOLCK
	CORPGPNT	CORSHARE	CORSWPNT	CORTABLE	CORVM	CPAPRINP	CPCREG0	CPEXADD	CPEXBLOK	CPEXBPNT
	CPEXFPNT	CPEXPROC	CPEXREGS	CPEXR11	CPEXSIZE	CPID	CPPTLBR	CPSTAT2	CPSTAT5	CPUMCELL
	CP370EON	CSADDR	C0	C1	C2	DEFER	DMKCFMBK	DMKCVTBH	DMKDMPAA	DMKDMPC2
	DMKDMPMA	DMKDMPRY	DMKDMPSA	DMKDSPCH	DMKDSPNP	DMKERMSG	DMKEXTST	DMKFREAP	DMKFREEP	DMKFRELP

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	DMKFRET	DMKLOKDS	DMKLOKFR	DMKLOKIO	DMKLOKRL	DMKLOKRM	DMKLOKSY	DMKLOKTR	DMKMCTAF	DMKMCTFL
	DMKMCTPF	DMKMCTVM	DMKPGUSP	DMKPTRAN	DMKPTRLK	DMKPTRSC	DMKPTRUL	DMKPTSAD	DMKPTSPW	DMKPTTFT
	DMKQCNWT	DMKSCHCA	DMKSCHCU	DMKSCHLI	DMKSCHTQ	DMKSCNVS	DMKSNTBL	DMKSTKCP	DMKSYSNP	DMKVMASW
	DMKVMAS1	DPAPUOP	DPSTAT	DSPRQ	FF	F0	F1	F15	F16	F2
	F256	F4	F4096	F7	F8	IOBBPNT	IOBFPNT	IOBLOK	IPUADDR	IPUADDRX
	IUCVCNT	LASTUSER	LOCK	LPUADDR	LPUADDRX	MCHAREA	MCHCPEX	MCHFIX	MCHLEN1	MCHREC
	MICBLOK	MICPMPSA	MP	MPUOPER	NORET	OPERATOR	PAGBMP	PAGCORE	PAGECUR	PAGELOAD
	PAGERATE	PAGINVAL	PAGTABLE	PAGTSWP	PGSRATIO	PMAAVAIL	PMAMODE	PMASTAT	POFFLINE	PREFIXA
	PREFIXB	PRIMEHDR	PRIMEHI	PRIMELO	PROCIO	PROCIPL	PSA	PSACPXBP	PSADSPRQ	PSAEXT
	PSAEXTX	PSALANG	PSAPGID	PXA	RCHBLOK	RCHCUTBL	RCHPROC	RCUBLOK	RCUDVTBL	RDEVBLOK
	RDEVCODE	RDEVPROC	RDEVTYPC	RDEVTYPE	RDIDX	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SAVER1	SAVER11	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK6	SAVEWRK7
	SAVEWRK8	SAVEWRK9	SENSE	SHRFLAG	SHRFPNT	SHRNAME	SHRNOPRT	SHRPAGE	SHRSEGCT	SHRSEGNM
	SHRSGPRT	SHRTABLE	SIGIPR	SIGRES	SIGSENSE	SIGSTART	SIGXC	SNASTATS	SPMPFX	STACKVM
	SWPCHG1	SWPCYL	SWPFLAG	SWPRECMP	SWPTABLE	SWPVM	SYSNAME	SYSTEM	TIMEDISP	TRACSTRT
	TRAPDATA	TRLANCHR	TRLCT	TRLLABEL	TRQBFPNT	TRQBLOK	TRQBVAL	TTSEGCNT	TYP2314	TYP3330
	TYP3350	TYP3375	TYP3380	UNSHRVM	VECOPVF	VECSTAT	VGPTR	VMBLOK	VSMPTR	XCPEND
	XPAGNUM	ZEROES								
DMKCPS	ADSPCH	AEXTSP	AFREE	AFREEP	AFRET	ALOKSP	AP	APSTAT1	APTRAN	APUOPER
	AQCNWT	ARIOCH	ARIOCT	ARIOCU	ASYSOP	ASYSVM	BRING	CFSTOP	CLASDASD	CLASFBA
	CLASTAPE	CLASURO	CPABEND	CPCREG8	CPEXADD	CPEXBLOK	CPEXR0	CPEXR12	CPEXSIZE	CPID
	CPINITD	CPSHRLK	CPSHUT	CPSPMODE	CPSTAT2	CPSTAT4	CPSTAT5	CPUMODEL	CPUVERSN	C1
	C6	C8	DEFER	DEVICE	DFRET	DMKCFMBK	DMKCFQRD	DMKCPWUN	DMKCVTBH	DMKCVTHB
	DMKDMPRS	DMKDMPRY	DMKDMPSD	DMKDSBSD	DMKDSPCH	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKIOESR
	DMKIOSHA	DMKIOSQR	DMKLOKIO	DMKLOKSW	DMKMNISH	DMKPRGMC	DMKPTRAN	DMKQCNWT	DMKRSFSD	DMKSCNEP
	DMKSCNFD	DMKSCNMU	DMKSCNRA	DMKSCNRU	DMKSCNVD	DMKSCNVN	DMKSCNVU	DMKSNTQN	DMKSPKDL	DMKSTKCP
	DMKSYSTP	DMKVSUCO	F0	F4	F4096	INTREQ	IOBCAW	IOBCP	IOBFATAL	IOBFLAG
	IOBHVC	IOBIMSTK	IOBIOER	IOBIRA	IOBLINK	IOBLOK	IOBMISC	IOBMISC2	IOBRADD	IOBRUNLD
	IOBSIZE	IOBSNSE	IOBSPEC	IOBSTAT	IOBUSER	IOBVADD	IOBXTRA	IOERBLOK	IOERETN	IOEREXT
	IOERSIZE	IPUADDR	IPUADDRX	LOCK	LPUADDR	LPUADDRX	L4	MONAIOB	MONARDB	MONCOM
	MONFLAG1	MONFLAG3	MONIOBF	MONUSER	MP	MPUOPER	NORET	NPRCNT	NPRNAME	NPRPNT
	NPRTBL	PMAAVAIL	PMASTAT	POWEROFF	PREFIXA	PREFIXB	PRIORITY	PROCIPL	PSA	RCHBLOK
	RCHCUTBL	RCUBLOK	RCUCHA	RCUDVTBL	RCUPRIME	RCUSUB	RCUTYPE	RDEVAIOB	RDEVBLOK	RDEVCTRS
	RDEVCUA	RDEVDED	RDEVDELP	RDEVDISA	RDEVFIOB	RDEVIMAG	RDEVIOER	RDEVNRDY	RDEVQIOB	RDEVREW
	RDEVSTAT	RDEVSTA5	RDEVTYPC	RDEVTYPE	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	SAVER11	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8
	SAVEWRK9	SHRLKCNT	SIGEMS	SIGQUI	SIGSENSE	SIGSHD	SIGSTOP	SILI	SPMPFX	SPOOLED
	SPXCR6	SYSTEM	TIMEDISP	TYP3480	TYP3800	TYP38003	TYP38008	TYP8809	VMBLOK	XPAGNUM
	X40FFS	ZEROES								
DMKCPT	AFREE	AFRET	ALOKSP	AP	APSTAT1	APUOPER	AQCNWT	BLANK	BUFCNT	BUFFER
	BUFNXT	CLASDASD	CONTINUE	CPEXBLOK	CPEXSIZE	CPINITD	DMKAPYSD	DMKCPNVY	DMKCPUVY	DMKCPWVF
	DMKCPZPG	DMKCVTBD	DMKCVTBH	DMKCVTHB	DMKERMSG	DMKFREE	DMKFRET	DMKLOKIO	DMKQCNWT	DMKSCNFD
	DMKSCNRN	DMKSCNRU	DMKVFCVV	FFS	F1	F2	F3	F4	F5	F7
	F8	F9	IOBBADCH	IOBLOK	IOBMISC	IOBSIZE	IOBXTRA	LOCK	LPUADDR	MP
	NOADD	NORET	ON	PSA	RANGE	RCHBLOK	RCUBLOK	RCUCHA	RCUPRIME	RCUSUB
	RCUTYPE	RDEVADD	RDEVALT	RDEVBLOK	RDEVCUA	RDEVCUP	RDEVDISA	RDEVPBYP	RDEVPEND	RDEVPPAG
	RDEVPRDV	RDEVPTHS	RDEVSTAT	RDEVSTA2	RDEVSTA3	RDEVSTA5	RDEVTYPC	RDEVTYPE	RDEVUSER	R0
	R1	R10	R11	R12	R13	R14	R15	R2	R3	R4
	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER2	SAVER9	SAVEWRK1
	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	TYP2305	VMBLOK
	X40FFS									

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
DMKCPU	ACORETBL	AEXTSP	AFREE	AFREEP	AFRET	AMCHAREA	AP	APSTAT1	APSTAT4	APTRAN
	APTRLK	APUOPER	AQCNWT	ASYSVM	AVMREAL	BLANKS	BRING	CORCP	CORFLAG	CORFPNT
	CORTABLE	CPAPRINP	CPCREGO	CPEXBLOK	CPEXFPNT	CPEXSIZE	CPSHRLK	CPSPMODE	CPSTAT2	CPSTAT4
	CPUID	CPUMCELL	CPUMODEL	CP370EON	CSSFEAT	C0	C1	DEFER	DMKAPICK	DMKAPIPR
	DMKCFMBK	DMKCLKCK	DMKCPMIO	DMKCPOFF	DMKCVTBH	DMKDMPCP	DMKDSPNP	DMKERMSG	DMKFREAP	DMKFREE
	DMKFREEP	DMKFRET	DMKIOGAP	DMKLOKDS	DMKLOKFR	DMKLOKIO	DMKLOKRL	DMKLOKRM	DMKLOKSW	DMKLOKSY
	DMKLOKTR	DMKMCTPF	DMKMHCCP	DMKPTRAN	DMKPTRFR	DMKPTRUL	DMKPTTFT	DMKQCNWT	DMKSTPC2	DMKSTPX
	DMKSYSAP	DMKVMAS1	EXTMASK	FF	F0	F1	F16	F255	F4	F4096
	F7	HCBLOK	HCMSFCW	HCMSFDB	IOERETN	IPUADDR	IPUADDRX	LOCK	LPUADDR	LPUADDRX
	L2048	MCHAREA	MCHCPEX	MCHFIX	MCHLEN1	MCHREC	MFAMASK	MICBLOK	MICPMPSA	MP
	MPFEAT	MPUOPER	MSFBLOK	MSFLNG	MSFRSP	NORET	ON	PAGBMP	PAGTABLE	PAGTOT
	PGWAITIM	PMAAVAIL	PMAMODE	PMASTAT	POFFLINE	PREFIXA	PREFIXB	PROCIO	PROCIPL	PSA
	PSACPXBP	PWTPAGES	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS
	SAVER1	SAVER11	SAVEWRK1	SAVEWRK2	SAVEWRK4	SAVEWRK8	SAVEWRK9	SHRFLAG	SHRFPNT	SHRLKCNT
	SHRNOPRT	SHRPAGE	SHRSEGCT	SHRSEGNM	SHRSGPRT	SHRTABLE	SIGIPR	SIGSENSE	SIGWAKE	SIGXC
	SPMPFX	SYSTEM	TIMEDISP	VMBLOK	XKEYMODE	ZEROES				
DMKCPV	AFREEP	AP	APSTAT1	APUOPER	AQCNWT	ARIOCH	ARIOCT	ARIOCU	ASYSVM	BALRSAVE
	CLASDASD	CLASFBA	CLASGRAF	CLASTERM	CPEXADD	CPEXBLOK	CPEXREGS	CPEXR12	CPEXSIZE	DMKACOCL
	DMKACODV	DMKACOFF	DMKACOTM	DMKCNSEN	DMKCVTBH	DMKCVTHB	DMKERMSG	DMKFREEP	DMKGRFEN	DMKLOKSW
	DMKQCNWT	DMKRNH	DMKSCNAU	DMKSCNFD	DMKSCNMU	DMKSCNRA	DMKSCNRD	DMKSCNRN	DMKSCNRU	DMKSTKCP
	DMKVCTDA	DMKVCTEN	FFS	F3	F4	F5	F8	LOCK	MP	NORET
	NOTERM	PSA	RCHBLOK	RCHCUTBL	RCUBLOK	RCUDVTBL	RDEVBASE	RDEVBLOK	RDEVDED	RDEVDISA
	RDEVDISB	RDEVENAB	RDEVFLAG	RDEVLOG	RDEVNAME	RDEVPEND	RDEVSTAT	RDEVTFLG	RDEVTYPC	RDEVTYPE
	RDEVUSER	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER12
	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK8	TIMEDISP	TYPSDLC	TYPTTY	TYP3066	TYP3277
	TYP3278	TYP3284	VMBLOK	ZEROES						
DMKCPW	ADSPCH	AFREE	AFREEP	AFRET	ALOKSP	AP	APSTAT1	APTRAN	APTRLK	APUOPER
	AQCNWT	ARDCBLOK	ASYSVM	BLANKS	BRING	CCC	CLASDASD	CLASFBA	CLASGRAF	CLASTAPE
	CLASTERM	CLASURI	CLASURO	CLEAR	CPEXADD	CPEXBLOK	CPEXREGS	CPEXR0	CPEXR11	CPEXR12
	CPEXSIZE	CPSHUT	CPSTAT4	CPUVERSN	C1	DE	DEFER	DMKCVTBH	DMKDSBRD	DMKDSPCH
	DMKFREE	DMKFREEP	DMKFRET	DMKIOESR	DMKIOSQR	DMKLOKIO	DMKPTRAN	DMKQCNWT	DMKRSFSD	DMKSCNPH
	DMKSCNRA	DMKSCNRU	DMKSCONP	DMKSPKDL	DMKSSSDE	DMKSTKCP	DMKSTKIO	DMKTAPRL	F0	IFCC
	IOBASNCT	IOBCAW	IOBCC3	IOBCSW	IOBDYNP	IOBFATAL	IOBFUNCT	IOBIOER	IOBIRA	IOBLINK
	IOBLOK	IOBMISC	IOBMISC2	IOBPATHF	IOBPGID	IOBPROC	IOBRADD	IOBRESGN	IOBSIZE	IOBSPEC
	IOBSPEC5	IOBSTAT	IOBTIO	IOBUASN	IOBUNSL	IOBUSER	IOBXTRA	IOERBLOK	IOERDATA	IOEREXT
	IOERSIZE	LOCK	LPUADDR	MP	MSSPRES	NICSIZE	NORET	PREFIXA	PSA	PSAMSS
	PSAPGID	RCHBLOK	RCHDISA	RCHPROC	RCHSTAT	RCUBLOK	RCUDISA	RCUSTAT	RDCBLKAP	RDCBLKCG
	RDCBLKFA	RDCBLKMA	RDCBLKMX	RDCBLKPG	RDCBLOK	RDCCKD	RDCFBA	RDCFLAG	RDCFPNT	RDCLENG
	RDCLENGC	RDCLENGF	RDCPAGAP	RDCPAGCG	RDCPAGFA	RDCPAGMA	RDCRECSZ	RDCSIZE	RDCSTART	RDCTYPE
	RDEVADD	RDEVALT	RDEVASGN	RDEVBLOK	RDEVBPAG	RDEVBUZY	RDEVCRDC	RDEVCTRS	RDEVCUP	RDEVDED
	RDEVDELP	RDEVDISA	RDEVDRAN	RDEVENAB	RDEVEPLN	RDEVEXTN	RDEVFLAG	RDEVFOFF	RDEVFTR	RDEVLCEP
	RDEVLNCP	RDEVLNKS	RDEVMAX	RDEVMDL	RDEVMD00	RDEVMOUT	RDEVM000	RDEVM004	RDEVNATH	RDEVNICL
	RDEVNRDY	RDEVOWN	RDEVPPAG	RDEVPRDV	RDEVPTHS	RDEVRCVY	RDEVRDC	RDEVRRES	RDEVRSVD	RDEVSCHD
	RDEVSEL	RDEVSER	RDEVSPL	RDEVSTAT	RDEVSTA2	RDEVSTA4	RDEVSTA5	RDEVSTA6	RDEVSYS	RDEVTYPC
	RDEVTYPE	RDEVUNF	RSPXBLOK	RSPXFLAG	RSPXFMNT	RSPXFPND	RSPXPMNT	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER13	SAVER9	SAVEWRK1	SAVEWRK2	SAVEWRK3
	SAVEWRK4	SAVEWRK5	SAVEWRK8	SAVEWRK9	SILI	SYSTEM	SYSVIRT	TYPBSC	TYPFBA	TYP2305
	TYP3310	TYP3370	TYP3380	TYP3480	TYP3705	TYP3800	TYP38003	TYP38008	UT3310	UT3370
	UT3370M4	VIRTUAL	VMBLOK	ZEROES						

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
DMKCPX	ADSPCH	AFREEP	ALOCBLOK	AP	APTRLK	ASYSVM	CPEXADD	CPEXBLOK	CPEXR0	CPEXR11
	CPEXR2	CPEXR5	CPEXSIZE	DMKDSPCH	DMKFREEP	DMKPTRLK	DMKPTRUL	DMKSTKCP	DMKZTDST	LOCK
	L15	L16	MP	PSA	RDEVBLOK	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R6	R7	R8	SAVEAREA
	SAVEREGS	SAVEWRK1	SAVEWRK2	SAVEWRK3	VMBLOK					
DMKCPY	ACORETBL	AP	APAGCP	APSTAT1	APTRAN	APUOPER	AQCNWT	ASYSVM	AVMREAL	BRING
	CORCFLCK	CORDISA	CORFLAG	CORFPNT	CORSHARE	CORTABLE	CORVM	CPSPMODE	CPSTAT2	CPSTAT4
	CPUID	CPXSTOR	C1	DEFER	DMKCVTBH	DMKCVTHB	DMKDSPNP	DMKDSPN2	DMKERMSG	DMKLOKSW
	DMKPGSPR	DMKPTRAN	DMKPTRAQ	DMKPTRUL	DMKPTTAL	DMKPTTFT	DMKQCNWT	DMKSCNAU	DMKSCNFD	DMKSELCP
	DMKSELVR	DMKSYSAP	DMKSYSRC	DMKSYSRM	DMKSYSSP	FFS	F0	F1	F2	F3
	F4	F4095	F4096	F9	INTPR	IOKEEP	LOCK	MP	MPFEAT	NORET
	OPERATOR	PREFIXA	PREFIXB	PSA	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	SAVER11	SAVEWRK1	SAVEWRK2	SAVEWRK4	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9
	SPECIALV	SVMUNLOK	SYSTEM	TIMEDISP	VMBLOK	VMSIZE	VRSVUID	XKEYMODE	ZEROES	
DMKCPZ	ADSPCH	AFREE	AFRET	ALOCBLOK	ALOCMAP	ALOKSP	APSTAT1	APUOPER	AQCNWT	ARIOCU
	ARIODV	ARIOUC	ASYSVM	BUSY	CAW	CC	CCC	CE	CODE	CPEXBLOK
	CPEXFPNT	CPEXMISC	CPEXREGS	CPEXSIZE	CPSTAT5	CSW	DE	DMKALOCA	DMKCPIMS	DMKCVTBH
	DMKDSPCH	DMKERMSG	DMKFREE	DMKFRET	DMKIOSQR	DMKLOKIO	DMKMNLIN	DMKQCNWT	DMKSCNEP	DMKSCNRA
	DMKSCNRD	DMKSYSOC	DMKSYSOW	FFS	FLAG	F15	F255	IFCC	IOBCAW	IOBCC3
	IOBCP	IOBCSW	IOBFLAG	IOBIOER	IOBIRA	IOBLINK	IOBLOK	IOBMISC	IOBPROC	IOBRADD
	IOBSIZE	IOBSTAT	IOBUSER	IOERBLOK	IOERSIZE	LOCK	LPUADDR	MPUOPER	NORET	PREFIXA
	PROCIPL	PSA	RCHBLOK	RCHPROC	RCUBLOK	RCUCACH	RCUDISA	RCUDVTBL	RCUSIZE	RCUSTAT
	RCUSUB	RCUTYPE	RDEVALLN	RDEVALT	RDEVBLOK	RDEVBPAG	RDEVCODE	RDEVCUA	RDEVCUB	RDEVCUP
	RDEVCU11	RDEVCU2	RDEVDED	RDEVDISA	RDEVFLAG	RDEVFTR3	RDEVOWN	RDEVPEND	RDEVPPAG	RDEVPRDV
	RDEVPTHS	RDEVRRES	RDEVSER	RDEVSTAT	RDEVSTA2	RDEVSTA4	RDEVSTA5	RDEVSYS	RDEVUSER	RDIDX
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER1	SAVER8
	SAVEWRK1	SAVEWRK3	SAVEWRK8	SAVEWRK9	SILI	TEMPSAVE	TRACCURR	TRACEND	TRACPROC	TRACSTRT
	TRACSVCR	UC	VMBLOK							
DMKCQC	ADSPCH	AFREE	AFREEP	AFRET	AQCNWT	BLANK	COLON	CPEXADD	CPEXBLOK	CPEXR0
	CPEXSIZE	DMKCVTBH	DMKCVTHB	DMKDSPCH	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKQCNWT	DMKSCNFD
	DMKSTKCP	F0	F3	HALF1ENT	HALF1EN1	HALF1RLN	HALF1SZE	HALF1VAL	LOCK	NORET
	OFF	ON	O2ENTRY	O2MAXLEN	O2SIZE	O3SIZE	PREFIXA	PSA	R0	R1
	R10	R11	R12	R13	R14	R15	R2	R3	R4	R5
	R6	R7	R8	SAVEAREA	SAVEREGS	SAVEWRK1	SAVEWRK2	SAVEWRK4	SAVEWRK6	SELDATA
	SELDISP	SELECT	SELENTRY	SELFORW	SELLTH	TRAPDATA	TRAPTT	TYPNUMAX	VMBLOK	ZEROES
DMKCQG	AFREE	AFRET	AQCNWT	BLANK	BLANKS	CLASDASD	CLASFBA	CLASGRAF	CLASSPEC	CLASTAPE
	CLASTERM	CLASURI	CLASURO	DMKCVTBD	DMKCVTBH	DMKCVTHB	DMKERMSG	DMKFREE	DMKFRET	DMKQCNWT
	DMKSCNAU	DMKSCNFD	DMKSCNRD	DMKSCNRN	DMKSCNVN	DMKSCNVU	ERRCODE	F1	F255	F3
	F7	L4	NICBLOK	NICDXSC	NICRFLG	NICRSPL	NICSIZE	NICTYPE	NORET	NOTERM
	PSA	RANGE	RDEVADD	RDEVATT	RDEVBLOK	RDEVBPAG	RDEVNICL	RDEVPPAG	RDEVPRDV	RDEVPS
	RDEVSER	RDEVSNRB	RDEVSTA3	RDEVSTA5	RDEVTYPC	RDEVTYPE	R0	R1	R10	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	SAVEAREA	SAVEREGS	SAVER0	SAVER11	SAVER2	SAVEWRK1	SAVEWRK2	SAVEWRK3
	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SNARBLOK	SNARLUN	SNARVMB	TYPBSC	TYPCTCA
	TYPPUN	TYPTIMER	TYP2305	TYP2311	TYP2314	TYP3210	TYP3310	TYP3330	TYP3340	TYP3350
	TYP3370	TYP3375	TYP3380	TYP3800	TYP38003	VCONCTL	VCONNICB	VCONRDEV	VCONREMD	VCONREMF
	VMBLOK	ZEROES								

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)								
DMKCQH	AFREE	AFRET	APTRLK	AQCNWT	ARSPPR	ARSPPU	BLANK	BLANKS	DMKCQIFR	DMKCVTBD
	DMKCVTDB	DMKERMSG	DMKFREE	DMKFRET	DMKLOCRD	DMKLOCRQ	DMKPTRLK	DMKPTRUL	DMKQCNWT	DMKRSPCP
	DMKRSPCR	DMKRSPCU	DMKSCNFD	DMKVSFID	DMKVSFNS	DMKVSFNU	F1	F2	F3	F4
	F8	NORET	PSA	R0	R1	R10	R11	R12	R13	R14
	R15	R2	R3	R4	R5	R6	R8	R9	SAVEAREA	SAVEREGS
	SAVER2	SAVEWRK1	SAVEWRK2	SAVEWRK4	SAVEWRK6	SAVEWRK8	SFBBCONV	SFBCLAS	SFBDEST	SFBFLAG
	SFBFLAG4	SFBINUSE	SFBLOK	SFBNOHLD	SFBOFORM	SFBPNT	SFBPURGD	SFBSHOLD	SFBSYSID	SFBUFORM
	SFBUHOLD	SFBUSER	VMBLOK	ZEROES						
DMKCQI	AFREE	AFRET	APTRLK	AQCNWT	BLANK	BLANKS	BRING	DMKCVTBD	DMKERMSG	DMKFREE
	DMKFRET	DMKLOCRD	DMKLOCRQ	DMKPGUVG	DMKPGUVR	DMKPTRLK	DMKPTRUL	DMKQCNWT	DMKRPAGT	DMKVSFNS
	DMKVSFNU	ERRMSG	F1	NORET	NOTIME	PSA	RDEVBLOK	RDEVSNA	R0	R1
	R10	R11	R12	R13	R14	R15	R2	R3	R4	R5
	R6	R8	R9	SAVEAREA	SAVEREGS	SAVEWRK1	SAVEWRK2	SAVEWRK4	SAVEWRK6	SAVEWRK8
	SFBACNT	SFBBCONV	SFBCLAS	SFBCONV	SFBCOPY	SFBDATE	SFBDEST	SFBDIST	SFBDUMP	SFBFILID
	SFBFLAG	SFBFLAG2	SFBFLAG3	SFBFLAG4	SFBFLASH	SFBFNAME	SFBFTYPE	SFBINUSE	SFBLDBEG	SFBLDMID
	SFBLOK	SFBMON	SFBNOHLD	SFBOFORM	SFBORIG	SFBPNT	SFBPURGD	SFBRECNO	SFBSHOLD	SFBSTART
	SFBSYSID	SFBTIME	SFBTUSE	SFBTYPE	SFBUFORM	SFBUHOLD	SFBUSER	SPCHAR	SPCHAR1	SPCHAR2
	SPCHAR3	SPCMCHR	SPCMOD	SPCOPYFG	SPFCB	SPFLAG1	SPFLSHC	SPLINK	SPSPLNKC	SYSTEM
	TYPPRT	TYPRDR	TYP5ACCW	VMBLOK	ZEROES					
DMKCQP	AFREE	AFRET	ALARM	APSTAT1	APUOPER	AQCNWT	ARIOCH	ARIOCU	ASYSVM	AVMREAL
	BLANKS	CLASDASD	CLASFBA	CLASGRAF	CLASSPEC	CLASTAPE	CLASTERM	CLASURI	CLASURO	DMKAPYSD
	DMKCQQPT	DMKCQTRE	DMKCQTSN	DMKCVTBD	DMKCVTBH	DMKERMSG	DMKFREE	DMKFRET	DMKQCNWT	DMKRIOCT
	DMKRIORN	DMKSCNEP	DMKSCNFD	DMKSCNMU	DMKSCNRA	DMKSCNRN	DMKSCNRU	DMKSCNVU	DMKSYSRC	DMKSYSRM
	DMKSYSRV	DMKURSTA	ERRMSG	FFS	FLAGS	F1	F2	F3	F6	F8
	IPUADDR	IPUADDRX	LPUADDR	L8	MPGEND	MPUOPER	NORET	PROCIO	PROCIPL	PSA
	RANGE	RCHBLOK	RCHCUTBL	RCHPROC	RCUBLOK	RCUCHA	RCUDVTBL	RCUPRIME	RCUSUB	RCUTYPE
	RC40	RDEVADD	RDEVATT	RDEVAUTO	RDEVBLOK	RDEVBPAG	RDEVCUA	RDEVCUP	RDEVDED	RDEVDISA
	RDEVDISB	RDEVDRAN	RDEVENAB	RDEVFLAG	RDEVFTR	RDEVLCEP	RDEVLNCP	RDEVLNKS	RDEVMOUT	RDEVNCP
	RDEVNRDY	RDEVOWN	RDEVPPAG	RDEVPRDV	RDEVPS	RDEVSER	RDEVSLOW	RDEVSTAT	RDEVSTA3	RDEVSTA5
	RDEVSYS	RDEVTYPC	RDEVTYPE	RDEVUSER	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	SAVER2	SAVER9	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK7	SAVEWRK8
	SAVEWRK9	SYSVIRT	TYPBSC	TYP2305	TYP3330	TYP3705	URSDEV	URSFILE	URSPATH	URSSTACK
	VIRTUAL	VMBLOK								
DMKCQQ	AFREE	AFRET	APSTAT1	AQCNWT	BLANK	BLANKS	CLASDASD	CLASFBA	CLASSPEC	CLASTERM
	DMKAPYSD	DMKCQPSC	DMKCVTBD	DMKCVTBH	DMKCVTHB	DMKERMSG	DMKFREE	DMKFRET	DMKQCNWT	DMKSCNAU
	DMKSCNEP	DMKSCNFD	DMKSCNPH	DMKSCNRA	DMKSCNRD	DMKSCNRU	DMKSCNVD	DMKSCNVU	FLAGS	F1
	F2	F3	F4	F8	IPUADDR	IPUADDRX	LPUADDR	MPGEND	MPUOPER	NORET
	PSA	RANGE	RCHBLOK	RCHPROC	RCUBLOK	RCUCHA	RCUCHCNT	RCUDISA	RCUPRIME	RCUSTAT
	RCUSUB	RCUTYPE	RC40	RDEVADD	RDEVAOF	RDEVBLOK	RDEVBOF	RDEVCUA	RDEVCUB	RDEVCUP
	RDEVFLAG	RDEVOWN	RDEVPPAG	RDEVPRDV	RDEVPS	RDEVPTHS	RDEVSNRB	RDEVSTA3	RDEVSTA5	RDEVSYS
	RDEVTYPC	RDEVTYPE	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS
	SAVER0	SAVER1	SAVER11	SAVER2	SAVER9	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5
	SAVEWRK6	SAVEWRK7	SAVEWRK8	SNARBLOK	SNARLUN	TYPBSC	VMBLOK	ZEROES		
DMKCQR	AFREE	AFRET	AQCNWT	BLANK	BLANKS	CLASDASD	CLASFBA	CLASTAPE	DMKCVTBD	DMKCVTBH
	DMKDMPAL	DMKDMPDV	DMKDMPSW	DMKERMSG	DMKFREE	DMKFRET	DMKQCNWT	DMKRIOPR	DMKRSPHQ	DMKRSPPR
	DMKRSPPU	DMKSCHPG	DMKSCNAU	DMKSCNFD	DMKSCNMU	DMKSCNRD	DMKSCNRU	F1	F2	F8
	NORET	PRIORITY	PSA	RDEVBLOK	RDEVTYPC	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SAVER0	SAVEWRK2	SFBFLAG	SFBLOK	SFBSHOLD	SHQBLOK	SHQSHOLD

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	SHQUSER	TYPPRT	TYPPUN	VMBLOK						
DMKCQS	AFREE	AFRET	APTRAN	APTRLK	AQCNWT	ASYSVM	BRING	CLASSPEC	CLASTERM	CTCLASS
	CTENTRY	CTFALIAS	CTFLAG	CTFLAST	CTNAME	CTSIZE	C1	DEFER	DFRET	DMKCFCTB
	DMKCVTBD	DMKCVTBH	DMKDIDEP	DMKERMSG	DMKFREE	DMKFRET	DMKHVCDE	DMKHVCDG	DMKHVCDU	DMKPGTCP
	DMKPGTPB	DMKPGTPI	DMKPGTSB	DMKPGTSI	DMKPTRAN	DMKPTRUL	DMKPTSMW	DMKQCNWT	DMKSCNAU	DMKSCNFD
	DMKSCNRD	DMKSYSTV	F0	F1	F4	F5	F60	LOCK	NORET	PREFIXA
	PSA	RDEVADD	RDEVBLOK	RDEVPS	RDEVSNRB	RDEVSTA3	RDEVTYPC	RDEVTYPE	RDEVUSER	R0
	R1	R10	R11	R12	R13	R14	R15	R2	R3	R4
	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVEWRK1	SAVEWRK3	SNARBLOK
	SNARDIAL	SNARFG2	SNARLUN	SNARNXT	SYSTEM	TYPBSC	VMBLOK	VSMPTR	WRITE	
DMKCQT	ADSPCH	AFREE	AFRET	AQCNWT	ASYSVM	BLANKS	BUSY	CCC	CDC	CLASDASD
	CLASFBA	CLASGRAF	CLEAR	CPEXBLOK	CPEXR0	CPEXR2	CPEXSIZE	DFRET	DMKCVTAB	DMKCVTBH
	DMKCVTHB	DMKDSPCH	DMKERMSG	DMKFREE	DMKFRET	DMKIOSQR	DMKQCNWT	DMKRIORN	DMKSCHST	DMKSCNFD
	DMKSCNRD	DMKSCNRU	EQCHK	FLAGS	F1	F2	F8	IFCC	INTREQ	IOBCAW
	IOBCC3	IOBCSW	IOBFATAL	IOBIOER	IOBIRA	IOBLOK	IOBMISC2	IOBQSTAT	IOBRUN1	IOBSIZE
	IOBSPEC5	IOBSTAT	IOBUSER	IOERBLOK	IOERCSW	IOERDATA	IOEREXT	IOERLEN	IOERPNT	IOERSIZE
	LANGBLOK	LANGLANG	NICBLOK	NICDISA	NICDXSC	NICENAB	NICFLAG	NICNAME	NICRDED	NICRFLG
	NICSIZE	NICSTAT	NICUSER	NORET	PREFIXA	PSA	RANGE	RDEVBLOK	RDEVDED	RDEVDISA
	RDEVMAX	RDEVNICL	RDEVNRDY	RDEVPEND	RDEVSNRB	RDEVSTAT	RDEVTYPC	RDEVUSER	R0	R1
	R10	R11	R12	R13	R14	R15	R2	R3	R4	R5
	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER13	SAVEWRK1	SAVEWRK2	SAVEWRK3
	SAVEWRK6	SAVEWRK7	SILI	SNARBLOK	SNARDIAL	SNARFG2	SNARLUN	SNARNXT	TRQBIRA	TRQBLOK
	TRQBSIZE	TRQBTOD	TRQBUSER	TRQBVAL	TRQREGSD	TRQREG2	UC	VMBLOK	VSMPTR	
DMKCQU	ACNTDATA	AFREE	AFRET	APSTAT1	APUOPER	AQCNWT	BLANK	CL	CLASSPEC	CLASTERM
	CPMICON	CPSTAT2	DFRET	DMKCVTBD	DMKFREE	DMKFRET	DMKGRTPF	DMKQCNWT	DMKSCNAU	F0
	F255	LPUADDR	NICAPL	NICATOF	NICBLOK	NICFLAG	NICLLEN	NICSIZE	NICTEXT	NICTMCD
	NORET	PREFIXB	PSA	RDEVAPLP	RDEVASTB	RDEVATOF	RDEVAVM2	RDEVBLOK	RDEVLLEN	RDEVNICL
	RDEVSCRL	RDEVSNA	RDEVSNRB	RDEVTAPL	RDEVTEXT	RDEVTFLG	RDEVTMCD	RDEVTYPC	RDEVTYPE	RDEV3101
	R0	R1	R11	R12	R13	R14	R15	R2	R3	R4
	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVEWRK8	SNARBLOK	SNARFG2
	SNARTTY	TYPBSC	TYPTTY	VCONBRK	VCONCBRK	VCONCTL	VCONOBRK	VCONOPT	VCONSCRN	VCON3270
	VMBLOK									
DMKCQY	AFREE	AFRET	APSTAT1	APTRAN	APTRLK	APUOPER	AQCNWT	ASCHN	ASYSVM	BLANKS
	BRING	CLASSPEC	CLASTERM	CPASTON	CPDASAON	CPMICON	CPSPMODE	CPSTAT2	CPSTAT3	CPUID
	CP370EON	C1	DEFER	DFRET	DMKACOTM	DMKCPEID	DMKCPICD	DMKCVTBD	DMKCVTBH	DMKCVTDB
	DMKCVTDT	DMKERMSG	DMKFREE	DMKFRET	DMKPGUVG	DMKPGUVR	DMKPTRAN	DMKPTRUL	DMKQCNWT	DMKRPAGT
	DMKSCNAU	DMKSCNFD	DMKSCNMU	DMKSCNRD	DMKSNTBL	DMKSYSDW	DMKSYSLG	DMKSYSLU	DMKSYSND	DMKSYSNM
	DMKSYSOC	DMKSYSOW	DMKSYSTI	F0	F1	F2	F4	F4096	F8	HEADER
	IOKEEP	IPUADDR	IPUADDRX	LOCK	LOGMBLOK	LOGMLEN	LOGMNXT	LOGMTXT	NORET	ON
	OWNDLIST	OWNDRDEV	OWNDVSER	PFDATA	PFDCMD	PFDVAL	PFKADDR	PFKCOUNT	PFKFLAG	PFKIMM
	PFKLNG	PFKRET	PFKTABLE	PFTAB	PFTABS	PREFIXA	PREFIXB	PROCIPL	PSA	PSASYSID
	RDEVADD	RDEVBLOK	RDEVCODE	RDEVPS	RDEVSNRB	RDEVSTA3	RDEVTYPC	RDEVTYPE	R0	R1
	R10	R11	R12	R13	R14	R15	R2	R3	R4	R5
	R6	R7	R8	R9	SAVDATE	SAVEAREA	SAVEREGS	SAVER0	SAVEWRK1	SAVEWRK2
	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	SAVNAME	SAVTABLE	SAVTIME
	SNARBLOK	SYSNAME	SYSTEM	TYPBSC	VMBLOK	WRITE	XPAGNUM	ZEROES		
DMKCRM	ADSPCH	AFREE	AFRET	AP	APTRLK	ASYSVM	DMKDSPCH	DMKFREE	DMKFRET	DMKIDRFX
	DMKIDRIN	DMKIUACP	DMKPTRLK	DMKPTRUL	FFS	IXBLOK	IXIRA	IXR0	IXR1	IXR11
	IXSIZE	LOCK	MP	PMSGLIM	PSA	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	R9	SAVEAREA	SAVEREGS	SEVER	SRTBLOK	SRTFLAG	SRTGIND	SRTIPND	SRTNEXT	SRTRVPND
	SSCBADDR	SSCBLOK	SSCIXADD	SSCIXLST	SSCRTADD	SSCRTCNT	SSCTSFPT	SSCTSFVM	VMBLOK	
DMKCSB	ADSPCH	AFREE	AFRET	AP	AQCNWT	ARIOPR	ARIOPU	ARIORD	BLANKS	BUFFER
	CC	CHGRDV	CLASURI	CLASURO	CLEAR	DMKCKSPL	DMKCSCLD	DMKCSCLV	DMKCVTDB	DMKCVTHB
	DMKDSPCH	DMKERMSG	DMKFREE	DMKFRET	DMKIOSQR	DMKPTRUL	DMKQCNWT	DMKSCNFD	DMKSCNRD	DMKSCNRU
	DMKSCNVU	FTRUCS	F1	F2	F3	F4	F8	IMGBUFLN	IMGCCWDS	IMGCCWLN
	IMGFCBF1	IMGFCBPI	IMGFCBTI	IMGF3211	IMGHDR	IMGHDRSD	IMGNAME	IOBCAW	IOBCSW	IOBFATAL
	IOBFLAG	IOBIRA	IOBLINK	IOBLOK	IOBMISC	IOBMISC2	IOBRADD	IOBRES	IOBRSTRT	IOBSIZE
	IOBSTAT	IOBUSER	LOADFLG1	LOADHCPY	LOADHEAD	LOADIMAG	LOADIML1	LOADIML2	LOADNAME	LOADPAG1
	LOADPAG2	LOADPARM	LOADSIZE	LOADSPAC	LOADSZDW	LOADUCS	LOAD2CCW	LOCK	MP	NORET
	PSA	RDEVBLOK	RDEVDED	RDEVDISA	RDEVDRAN	RDEVEXTN	RDEVFLAG	RDEVFTR	RDEVSPL	RDEVSTAT
	RDEVTYPC	RDEVTYPE	RDEVUSER	RSPXBLOK	RSPXFCB	RSPXFLAG	RSPXFOLD	RSPXINDX	R0	R1
	R10	R11	R12	R13	R14	R15	R2	R3	R4	R5
	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER1	SAVEWRK1	SAVEWRK2	SAVEWRK4
	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SILI	SKIP	TYPPRT	TYP3203	TYP3211	TYP3262
	TYP3289E	TYP4245	TYP4248	UCSBBLOK	UCSBFOLD	UCSBNAME	UCSBSIZE	UCSBVER	UCSCCWS	UCSCCW0B
	UCSCCW06	UCSCCW07	UCSFCBAD	UCSFCBLD	UCSFCBL1	UCSFFLD	UCSLOAD	UCSNAME	UCSRDCCW	UCSREGS
	UE	VMBLOK								
DMKCSC	AFREE	AFRET	AP	APTRAN	APTRLK	ASYSVM	BRING	CC	CD	C1
	DEFER	DMKFCBLD	DMKFREE	DMKFRET	DMKPIALD	DMKPIBLD	DMKPTRAN	DMKPTRUL	DMKUCBLD	DMKUCCLD
	DMKUCSLD	F0	F1	F4095	F4096	IMGBUFLN	IMGHDR	IMGHDRSD	IMGHDRSZ	IMGNAME
	IMGNEXT	IOBLOK	IOERETN	IOKEEP	LOADFLG1	LOADHCPY	LOADHEAD	LOADIMAG	LOADIML1	LOADIML2
	LOADNAME	LOADPAG1	LOADPAG2	LOADPARM	LOADSPAC	LOADUCS	LOADVPAG	LOAD2CCW	LOCK	MP
	PSA	RDEVBLOK	RDEVTYPE	R0	R1	R10	R11	R12	R13	R14
	R15	R2	R4	R5	R6	R7	R8	SAVEAREA	SAVEREGS	SAVER1
	SILI	SKIP	SYSTEM	TYP3203	TYP3211	TYP3262	TYP3289E	TYP4245	TYP4248	UCSCCWS
	UCSCCW07	UCSRDCC1	VMBLOK	XPAGNUM	ZEROES					
DMKCSF	AFREE	AP	APSTAT1	AQCNWT	ARIOPR	ARIOPU	ASYSVM	CLASURO	CLEAR	DE
	DMKAPYSD	DMKCVTDB	DMKCVTHB	DMKERMSG	DMKFREE	DMKQCNWT	DMKRSPEX	DMKSCNFD	DMKSCNRU	DMKSTKIO
	F1	F2	F255	F3	F4	F8	IOBCP	IOBCSW	IOBFLAG	IOBIRA
	IOBLINK	IOBLOK	IOBMISC	IOBRADD	IOBSIZE	IOBUSER	LOCK	MP	MSGADDR	NORET
	PROCIO	PSA	RDEVBACK	RDEVBLOK	RDEVDED	RDEVDISA	RDEVEXTN	RDEVFLAG	RDEVRSTR	RDEVSPAC
	RDEVSPL	RDEVSTAT	RDEVTERM	RDEVTYPC	RDEVTYPE	RDEVUSER	RSPFLAG2	RSPLCTL	RSPMISC	RSPSFBLK
	RSPSFLOK	RSPXBLOK	RSPXFLAG	RSPXSFIL	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R7	R8	R9	SAVEAREA	SAVEREGS
	SAVEWRK1	SAVEWRK2	SAVEWRK4	SAVEWRK6	SAVEWRK7	SAVEWRK8	SFBCOPY	SFBFLAG	SFBFLAG3	SFBLDBEG
	SFBLDMID	SFBLOK	SFBRECER	SFBRECOK	SFBSHOLD	TYPPRT	TYPPUN	VMBLOK		
DMKCSO	AFREE	AFREEP	AFRET	ALOKSP	AP	APSTAT1	APTRAN	APUOPER	AQCNWT	ARIODV
	ARIOPR	ARIOPU	ARIORD	ASYSVM	BALRSAVE	BLANK	BLANKS	BRING	BUFFER	BUFNXT
	CHGRDV	CLASURI	CLASURO	CLEAR	CPSTAT5	C1	DE	DEFER	DMKAPYSD	DMKCKSPL
	DMKCVTDB	DMKCVTHB	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKLOKIO	DMKPTRAN	DMKQCNWT	DMKRSPEX
	DMKSCNFD	DMKSCNRU	DMKSNTQN	DMKSPKDL	DMKSTKIO	DMKURSTA	F0	F10	F2	F3
	F4	F4096	F6	F8	IOBCP	IOBCSW	IOBFATAL	IOBFLAG	IOBIRA	IOBLINK
	IOBLOK	IOBMISC	IOBRADD	IOBSIZE	IOBSTAT	IOBUSER	IOERETN	IOKEEP	LOCK	LPUADDR
	MP	MSGADDR	NORET	NPRCNT	NPRNAME	NPRPNT	NPRTBL	PSA	RDEVAIOB	RDEVBLOK
	RDEVBUZY	RDEVBZCH	RDEVCFCB	RDEVCLAS	RDEVDED	RDEVDFCB	RDEVDISA	RDEVDRAN	RDEVEXTN	RDEVFLAG
	RDEVFSEP	RDEVIMAG	RDEVIOER	RDEVLDBG	RDEVLDMD	RDEVLOAD	RDEVNRDY	RDEVOVLY	RDEVPRFG	RDEVPURG
	RDEVSCHD	RDEVSEP	RDEVSPAC	RDEVSPL	RDEVSTAT	RDEVSTA2	RDEVSTA4	RDEVTERM	RDEVTYPC	RDEVTYPE
	RDEVUSER	RDEVXSEP	RDIDX	RSPBF110	RSPBF210	RSPFLAG1	RSPFLAG2	RSPLCTL	RSPSFBLK	RSPSFLOK
	RSPXAUTO	RSPXBLOK	RSPXDEST	RSPXFLAG	RSPXFMNT	RSPXFORM	RSPXFPND	RSPXPMNT	RSPXRECT	RSPXSETU
	RSPXSFIL	R0	R1	R10	R11	R12	R13	R14	R15	R2

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)								
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVEWRK1
	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SFBFLAG2	SFBLOK	SFBREQUE
	SYSTEM	TYPPRT	TYPPUN	TYP3211	TYP3800	TYP38003	TYP38008	URSDEV	URSOPER	URSSTART
	URSUSER	VMBLOK	XPAGNUM	ZEROES						
DMKCSP	ACICLR1	ACICODE	ACIFCN	ACIMODE	ACIPARMS	ACIRGRP	ACIRUSR	ACISIZE	ACISPOOL	ACITGRP
	ACITUSR	ACIUDIR	AFREE	AFRET	AP	BLANK	BLANKS	BUFFER	CLASTERM	CLASURI
	CLASURO	CLEAR	DMKCSRGT	DMKCSRPO	DMKCVTDB	DMKERMSG	DMKFREE	DMKFRET	DMKRPIRA	DMKSCNAU
	DMKSCNFD	DMKUDRFU	F1	F2	F255	F3	F4	F8	LOCK	MP
	PSA	R0	R1	R11	R12	R13	R14	R15	R2	R3
	R4	R8	R9	SAVEAREA	SAVEREGS	SAVEWRK1	SAVEWRK2	SAVEWRK5	SAVEWRK7	SAVEWRK8
	SAVEWRK9	TYPRDR	TYP3210	VMBLOK						
DMKCSQ	AFREE	AFRET	AP	ARSPPR	ARSPPU	ARSPRD	BLANK	BLANKS	BUFFER	CHGSFB
	CHGSHQ	CLASTERM	CLASURI	CLASURO	CLEAR	DELSFB	DMKAPZNO	DMKCKSPL	DMKCKTSD	DMKCKTSU
	DMKCSOSD	DMKCVTHB	DMKERMSG	DMKFREE	DMKFRET	DMKRSPHQ	DMKRSPSC	DMKSCNFD	DMKSCNVU	DMKUDRFU
	DMKVSUCO	DMKVSUCR	FFS	F1	F2	F3	F4096	F7	F8	LOCK
	MP	PSA	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS
	SAVEWRK1	SAVEWRK2	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	SFBCLAS	SFBDEST
	SFBDIST	SFBFILID	SFBFLAG	SFBFLAG2	SFBFNAME	SFBHOLD	SFBINUSE	SFBLOK	SFBNOHLD	SFBSHOLD
	SFBSIZE	SFBUFORM	SFBUSER	SHQBLOK	SHQBSIZE	SHQCKPMP	SHQCKPT	SHQFLAGS	SHQPNT	SHQSHOLD
	SHQUSER	TYPPRT	TYPPUN	TYPRDR	TYP3210	VMBLOK	VSPLCTL	VSPSFBLK	ZEROES	
DMKCSR	AFREEP	AFRET	AP	APTRLK	BLANK	CLASTERM	CLASURI	CLASURO	DE	DMKCVTHB
	DMKERMSG	DMKFREEP	DMKFRET	DMKLOCRD	DMKLOCRQ	DMKPTRLK	DMKPTRUL	DMKSCNFD	DMKSCNVU	DMKSTKIO
	DMKSYSCO	DMKSYSPR	DMKSYSPU	DMKVIOIN	DMKVSFNU	DMKVSUCO	FFS	FORMBLOK	FORMUSER	F1
	F2	F3	IOBCSW	IOBIRA	IOBLINK	IOBLOK	IOBSIZE	IOBUSER	IOBVADD	LOCK
	MP	PSA	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS
	SAVER11	SAVER2	SAVER6	SAVER9	SAVEWRK2	SFBCLAS	SFBFILID	SFBFLAG	SFBINUSE	SFBLOK
	SFBSYSID	SFBUHOLD	TYPPRT	TYPPUN	TYPRDR	TYP3210	VMBLOK	VSPLCTL	VSPSFBLK	ZEROES
DMKCST	ACICODE	ACIFCN	ACINODE	ACIPARMS	ACIRGRP	ACIRUSR	ACISIZE	ACITAG	AFREE	AFRET
	AP	APTRLK	ARSPPR	ARSPRD	BLANKS	BRING	BUFCNT	BUFFER	BUFNXT	CLASTERM
	CLASURI	CLASURO	DMKCVTBH	DMKCVTDB	DMKCVTHB	DMKERMSG	DMKFREE	DMKFRET	DMKLOCRD	DMKLOCRQ
	DMKPGUVG	DMKPGUVR	DMKPTRLK	DMKPTRUL	DMKRPAGT	DMKRPAPT	DMKRPIRA	DMKSCNFD	DMKSCNVD	DMKSCNVN
	DMKSCNVU	DMKVSFID	FFS	F1	F2	F3	F4	F8	LOCK	MP
	PSA	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVEWRK1
	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	SFBFILID	SFBFLAG
	SFBFLAG4	SFBINUSE	SFBLOK	SFBPNT	SFBSTART	SFBTUSE	SFBUSER	SKIP	SYSTEM	TYPPRT
	TYPPUN	TYPRDR	TYP3210	VMBLOK	ZEROES					
DMKCSU	AFREE	AFRET	AP	ARSPPR	ARSPPU	ARSPRD	BLANK	BLANKS	BUFFER	BUFNXT
	CLEAR	DMKCSWCH	DMKCVTBD	DMKCVTDB	DMKERMSG	DMKFREE	DMKFRET	DMKPGUVR	DMKQCNPL	DMKSCNFD
	DMKUDRFU	FFS	F1	F2	F24	F255	F3	F4	F5	F6
	F7	F8	LOCK	MP	PLFLAG1	PLFRET	PLIMSG	PLIST	PLNORET	PLR2
	PLSIZE	PSA	R0	R1	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER11
	SAVEWRK1	SAVEWRK2	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK8	SFBLOK	TEMPR2	TYPPRT	TYPPUN
	VMBLOK									
DMKCSV	AFREE	AFRET	AP	ARSPPR	ARSPPU	ARSPRD	BLANK	BLANKS	BRING	BUFFER
	BUFNXT	CHGSFB	CLEAR	COUNT	DMKAPWPG	DMKCKSPL	DMKCVTBD	DMKCVTDB	DMKERMSG	DMKFREE

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)								
	DMKFRET	DMKLOCRD	DMKLOCRQ	DMKPGTSG	DMKPGUVG	DMKPGUVR	DMKPTRUL	DMKQCNPL	DMKRPAGT	DMKRPAPT
	DMKSCNFD	DMKSPKDL	DMKUDRFU	DMKVSDAD	DMKVSDDL	DMKVSFNS	DMKVSFNU	FFS	F1	F2
	F3	F4	F6	F8	LOCK	MP	NORET	NOTRESP	OFF	OWNID
	PLFLAG1	PLFRET	PLIMSG	PLIST	PLR2	PLSIZE	PSA	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SAVEAREA	SAVEREGS	SAVEWRK1	SAVEWRK2	SAVEWRK4	SAVEWRK5	SAVEWRK6
	SAVEWRK8	SAVEWRK9	SFBBCONV	SFBCLAS	SFBCONV	SFBDEST	SFBFILID	SFBFLAG	SFBFLAG4	SFBINUSE
	SFBINVS	SFBLAST	SFBLOK	SFBOFORM	SFBPNT	SFBPURGD	SFBSIZE	SFBSTART	SFBSYSID	SFBUFORM
	SFBUSER	SFBXAB	SFBXABL	SPLINK	SPNXTPAG	SPSPLNKC	SYSTEM	TEMPR2	TYPPRT	TYPPUN
	VMBLOK	ZEROES								
DMKCSW	AFREE	AFREEP	AFRET	AP	APTRLK	BLANK	BRING	CHGSFB	CLASURI	COUNT
	DE	DMKAPWPG	DMKAPZNO	DMKCKSPL	DMKCKTUU	DMKCSOSD	DMKFREE	DMKFREEP	DMKFRET	DMKLOCRD
	DMKLOCRQ	DMKPGTSG	DMKPGUVG	DMKPGUVR	DMKPTRLK	DMKPTRUL	DMKRPAGT	DMKRPAPT	DMKSCNAU	DMKSPKDL
	DMKSTKIO	DMKSYSFL	DMKVIOIN	DMKVSFNS	DMKVSFNU	FFS	FLAG	FORMBLOK	FORMNTRY	FORMOPER
	FORMSEND	FORMUSER	F0	F1	F4	F8	IOBCSW	IOBIRA	IOBLINK	IOBLOK
	IOBSIZE	IOBUSER	IOBVADD	LOCK	MP	PSA	R0	R1	R10	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	SAVEAREA	SAVEREGS	SAVER2	SAVER3	SAVER6	SAVER7	SAVEWRK1	SAVEWRK2
	SAVEWRK3	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SFBBCONV	SFBCLAS	SFBCONV	SFBCOPY	SFBDEST
	SFBDIST	SFBFILID	SFBFLAG	SFBFLAG4	SFBFLASH	SFBFNAME	SFBINUSE	SFBLAST	SFBLOK	SFBOFORM
	SFBPNT	SFBPURGD	SFBSHOLD	SFBSIZE	SFBSTART	SFBSYSID	SFBTUSE	SFBUFORM	SFBUHOLD	SFBUSER
	SFBXAB	SFBXABL	SPCHAR	SPCHAR1	SPCHAR2	SPCHAR3	SPCMCHR	SPCMOD	SPCOPYFG	SPFCB
	SPFLAG1	SPFLSHC	SPLINK	SPNXTPAG	SPSPLNKC	SYSTEM	TEMPR3	TEMPR4	TIMEDISP	TYPPRT
	TYPPUN	TYPRDR	VMBLOK	X2048BND	ZEROES					
DMKCSX	ACICLR1	ACICODE	ACIFCN	ACIMODE	ACIPARMS	ACIRGRP	ACIRUSR	ACISIZE	ACISPOOL	ACITGRP
	ACIUDIR	ACIUSRID	AFREE	AFREEP	AFRET	AP	APSTAT1	APUOPER	ARSPPR	ARSPPU
	ARSPRD	BLANK	BLANKS	BUFFER	BUFNXT	CLASURI	CLEAR	DE	DMKAPZNO	DMKCSOSD
	DMKCSYTR	DMKCVTBD	DMKCVTDB	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKLOCRD	DMKLOCRQ	DMKLOKSW
	DMKQCNPL	DMKRPIRA	DMKSCNAU	DMKSCNFD	DMKSTKIO	DMKUDRFU	DMKVIOIN	FFS	F1	F2
	F3	F4	F6	F8	IOBCSW	IOBIRA	IOBLINK	IOBLOK	IOBSIZE	IOBUSER
	IOBVADD	LOCK	MP	NORET	NOTRESP	OFF	OWNID	PLFLAG1	PLFRET	PLIMSG
	PLIST	PLR2	PLSIZE	PSA	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	SAVER11	SAVEWRK1	SAVEWRK2	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK8	SFBCLAS
	SFBFLAG	SFBLOK	SFBUHOLD	TEMPR0	TEMPR2	TIMEDISP	TYPPRT	TYPPUN	TYPRDR	VMBLOK
	ZEROES									
DMKCSY	AFREE	AFRET	APTRLK	ARSPPR	ARSPRD	BLANK	BRING	CHGSFB	COUNT	DMKAPWPG
	DMKCKSPL	DMKCKTSD	DMKCKTSU	DMKFREE	DMKFRET	DMKLOCRD	DMKLOCRQ	DMKPGTSG	DMKPGUVG	DMKPGUVR
	DMKPTRLK	DMKPTRUL	DMKRPAGT	DMKRPAPT	DMKRSPHQ	DMKVSDAD	DMKVSDDL	DMKVSETR	DMKVSFNS	DMKVSFNU
	DMKVSGAI	DMKVSGRI	F0	F1	LOCK	PCHCHN	PRTCHN	PSA	R0	R1
	R10	R11	R12	R13	R14	R15	R2	R3	R4	R5
	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER13	SAVER2	SFBBCONV	SFBCLAS
	SFBCONV	SFBDEST	SFBDUMP	SFBFILID	SFBFLAG	SFBFLAG2	SFBFLAG3	SFBFLAG4	SFBINUSE	SFBINVS
	SFBLAST	SFBLOK	SFBMON	SFBNORET	SFBOFORM	SFBORIG	SFBPNT	SFBPURGD	SFBSEEN	SFBSHOLD
	SFBSIZE	SFBSIZEB	SFBSTART	SFBSYSID	SFBTUSE	SFBTYPE	SFBUFORM	SFBUHOLD	SFBUSER	SFBXAB
	SFBXABL	SFBXFER	SHQBLOK	SHQPNT	SHQSHOLD	SHQUSER	SPLINK	SPNXTPAG	SPSPLNKC	SYSTEM
	TYPPRT	TYPPUN	TYPRDR	VMBLOK	ZEROES					
DMKCVT	BALRSAVE	BALR1	BALR2	CPID	DATE	F1	F10	F240	F60	LOCK
	PSA	R0	R1	R14	R15	R2	R3	TEMPSAVE	TODATE	
DMKCVU	F1	F240	F4	LOCK	PSA	R0	R1	R10	R11	R12

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	R13	R2	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	TEMPSAVE
DMKDAD	ADSPCH	AFREE	AFREEP	AFRET	ALARM	ALOCBLOK	ALOCFLG	ALOCPNT	ALOCPS	AQCNWT
	ASYSOP	ASYSVM	BLANKS	BUSOUT	CC	CCC	CD	CDC	CHC	CL
	CMDREJ	CPEXADD	CPEXBLOK	CPEXFPNT	CPEXR0	CPEXR12	CPEXR5	CPEXSIZE	C1	DATACHK
	DFRET	DMKCVTBH	DMKDSPCH	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKMSWR	DMKPTSAE	DMKPTTC1
	DMKPTTC2	DMKPTTC3	DMKQCNWT	DMKSTKCP	EQCHK	FFS	FLAG	FTRRSRL	F1	F10
	F16	F2	F3	F4095	F4096	F8	IDA	IFCC	IL	INTREQ
	IOBCAW	IOBCP	IOBCSW	IOBERP	IOBFATAL	IOBFLAG	IOBIOER	IOBLOK	IOBMISC	IOBPAG
	IOBRADD	IOBRCAW	IOBRCNT	IOBREL	IOBRSTRT	IOBSIZE	IOBSPEC	IOBSPEC2	IOBSTAT	IOBUNSL
	IOERACT	IOERADR	IOERBLOK	IOERCAN	IOERCCRA	IOERCCRL	IOERCEMD	IOERCPEX	IOERCSW	IOERDASD
	IOERDATA	IOERDEC	IOERDEF	IOERDW	IOERECF	IOEREXT	IOERFLG1	IOERFLG2	IOERFLG3	IOERHA
	IOERIGNR	IOERIND3	IOERIND4	IOERINFO	IOERLOC	IOERMSG	IOERMSW	IOERNUM	IOERPEND	IOERPNT
	IOERREAD	IOERSIZE	IOERSNSZ	IOERSTAT	IOERSTRT	IOERVOL1	IOERVSER	LOCK	L1	L2
	L4	L8	NORET	NOTUSED	OPERATOR	PAGCORE	PAGINVAL	PCI	PRGC	PRTC
	PSA	RCWCOMND	RCWHEAD	RCWPNT	RCWRCNT	RCWTASK	RDEVAIOB	RDEVALLN	RDEVBLOK	RDEVCUB
	RDEVDISA	RDEVFLAG	RDEVFTR	RDEVIOER	RDEVNRDY	RDEVOWN	RDEVRRES	RDEVSER	RDEVSTAT	RDEVSTA4
	RDEVTYPE	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER0
	SAVER11	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9
	SEGINV	SEGPAGE	SILI	SKIP	TEMPSAVE	TRANMODE	TYP3375	TYP3380	UC	VMBLOK
	XRIGHT16	X2048BND	ZEROES							
DMKDAS	ADSPCH	AFREE	AFREEP	AFRET	ALARM	ALOCBLOK	ALOCFLG	ALOCPNT	ALOCPS	AQCNWT
	ASYSOP	ASYSVM	BLANKS	CC	CCC	CD	CDC	CL	CPEXADD	CPEXBLOK
	CPEXFPNT	CPEXREGS	CPEXR0	CPEXR12	CPEXR5	CPEXSIZE	C1	DATACHK	DFRET	DMKCVTBH
	DMKDSPCH	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKIOEEX	DMKIOERR	DMKIOEST	DMKMSWR	DMKPTSAE
	DMKPTTC1	DMKPTTC2	DMKPTTC3	DMKQCNWT	DMKSTKCP	DMKTRKIN	FFS	FLAG	FTREXTSN	FTRRPS
	FTRRSRL	FTR35MB	FTR70MB	F1	F10	F15	F16	F2	F256	F4095
	F4096	F8	IDA	IFCC	IL	IOBALTSK	IOBCAW	IOBCP	IOBCSW	IOBERP
	IOBFATAL	IOBFLAG	IOBIOER	IOBLOK	IOBMISC	IOBPAG	IOBRADD	IOBRCAW	IOBRCNT	IOBREL
	IOBRSTRT	IOBSIZE	IOBSPEC	IOBSPEC2	IOBSTAT	IOBUNSL	IOERACT	IOERADR	IOERALTR	IOERBLOK
	IOERCAL	IOERCAN	IOERCCRA	IOERCCRL	IOERCEMD	IOERCPEX	IOERCSW	IOERDASD	IOERDATA	IOERDEC
	IOERDW	IOERECF	IOEREXT	IOERFLG1	IOERFLG2	IOERFLG3	IOERFLG4	IOERHA	IOERIGNR	IOERIND3
	IOERIND4	IOERINFO	IOERLOC	IOERMSG	IOERMSW	IOERNUM	IOERPEND	IOERPNT	IOERQUE	IOERRDRO
	IOERREAD	IOERSIZE	IOERSNSZ	IOERSTAT	IOERSTRT	IOERVOL1	IOERVSER	LOCK	L1	L2
	L24	L3	L4	NORET	NOTUSED	OPERATOR	PAGCORE	PAGINVAL	PCI	PRGC
	PRIORITY	PRTC	PSA	RCWCOMND	RDEVAIOB	RDEVALLN	RDEVBLOK	RDEVCUB	RDEVCUP	RDEVDED
	RDEVDISA	RDEVFLAG	RDEVFTR	RDEVIOER	RDEVMOUT	RDEVNRDY	RDEVOWN	RDEVRRES	RDEVSER	RDEVSTAT
	RDEVSTA4	RDEVSTA5	RDEVSYS	RDEVTYPE	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	SAVER0	SAVER11	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6
	SAVEWRK7	SAVEWRK8	SAVEWRK9	SEGINV	SEGPAGE	SILI	SKIP	TEMPSAVE	TRANMODE	TYP2305
	TYP2314	TYP3330	TYP3340	TYP3350	UC	VMBLOK	XRIGHT16	X2048BND	ZEROES	
DMKDAU	ADSPCH	AFREE	AFREEP	AFRET	ALARM	ALOCBLOK	ALOCFLG	ALOCPNT	ALOCPS	APSTAT1
	AQCNWT	ASYSOP	ASYSVM	BLANKS	CC	CCC	CD	CDC	CL	CPEXADD
	CPEXBLOK	CPEXFPNT	CPEXR0	CPEXR12	CPEXSIZE	C1	DFRET	DMKCVTBH	DMKDSPCH	DMKFREE
	DMKFREEP	DMKFRET	DMKMSWR	DMKPTSAE	DMKPTTC1	DMKPTTC2	DMKPTTC3	DMKQCNWT	DMKSTKCP	FFS
	FTRRSRL	F0	F1	F10	F16	F4095	F4096	F8	IDA	IFCC
	IOBCAW	IOBCP	IOBCSW	IOBERP	IOBFATAL	IOBFLAG	IOBIOER	IOBLOK	IOBRADD	IOBRCAW
	IOBRCNT	IOBRSTRT	IOBSIZE	IOBSPEC	IOBSTAT	IOBUNSL	IOERACT	IOERADR	IOERBLOK	IOERCAN
	IOERCCRA	IOERCCRL	IOERCEMD	IOERCPEX	IOERCSW	IOERDASD	IOERDATA	IOERDEC	IOERDW	IOERECF
	IOERETRY	IOEREXT	IOERFLG1	IOERFLG2	IOERFLG3	IOERIGNR	IOERIND3	IOERIND4	IOERINFO	IOERLOC
	IOERMSG	IOERMSW	IOERNUM	IOERPEND	IOERPNT	IOERREAD	IOERSIZE	IOERSNSZ	IOERSTAT	IOERSTRT

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	IOERVOL1	IOERVSER	LOCK	NORET	NOTUSED	OPERATOR	PAGCORE	PAGINVAL	PRGC	PROCIO
	PRTC	PSA	RDCBLKFA	RDCBLKMA	RDCBLOK	RDEVAIOB	RDEVALLN	RDEVBLOK	RDEVCUB	RDEVDED
	RDEVDISA	RDEVFLAG	RDEVFTR	RDEVIOER	RDEVMOUT	RDEVNRDY	RDEVOWN	RDEVRDC	RDEVRRES	RDEVSER
	RDEVSTAT	RDEVSTA4	RDEVSYS	RDEVTYPE	READ	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SAVER0	SAVER11	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK9
	SEGINV	SEGPAGE	SILI	SKIP	TRANMODE	TYP3370	UC	VMBLOK	WRITE	XRIGHT16
	ZEROES									
DMKDDC	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9				
DMKDDR	ATTN	BALRSAVE	BLANKS	BUSOUT	BUSY	CARDIN	CAW	CC	CCC	CD
	CE	CLASDASD	CLASFBA	CLASTAPE	CLASTERM	CMDREJ	CONTINUE	CPCLOSE	CSW	CUE
	C2	DATACHK	DE	DMKDDC	DMKDDT	DMKDNC	DMKDNT	DUMP	D1	D2
	D3	D5	EQCHK	ERRMSG	HEADER	IFCC	INPUT	INTREQ	LINECT	L1
	L10	L16	L2	L4	L5	L7	L8	L9	MODESET	READ
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SAVERET	SCAN	SENSE	SILI
	SKIP	SM	TIMER	TYPFBA	TYP2305	TYP2311	TYP2314	TYP2319	TYP2401	TYP2415
	TYP2420	TYP3330	TYP3340	TYP3350	TYP3375	TYP3380	TYP3410	TYP3411	TYP3420	TYP3422
	TYP3430	TYP3480	TYP8809	UC	UE	WAIT				
DMKDEF	AFREE	AFRET	BLANKS	CDDEF	CDVADD	CLASDASD	CLASFBA	CLASGRAF	CLASSPEC	CLASTERM
	CLASURI	CLASURO	CONSOLE	CONTINUE	DMKCFQRD	DMKCVTBD	DMKCVTBH	DMKCVTDB	DMKCVTHB	DMKDEGIN
	DMKDEIMS	DMKERMSG	DMKFREE	DMKFRET	DMKLOCKD	DMKLOCKQ	DMKNEARV	DMKPMACD	DMKQCNPL	DMKSCNFD
	DMKSCNVD	DMKSCNVN	DMKSCNVU	DMKVCBRS	DMKVDSDF	FFS	FTR3088	FTR4WCGM	F2	F255
	F3	F4	F5	F8	LINE	LOCK	L3	L7	NICBLOK	NICSIZE
	NICVDEVB	PLFLAG1	PLFRET	PLIMSG	PLIST	PLNORET	PLR2	PLSIZE	PSA	RC40
	RDEVATT	RDEVBLOK	RDEVCUP	RDEVNICL	RDEVSTA5	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SAVER2	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK6	SAVEWRK7
	SAVEWRK8	SAVEWRK9	SYSVIRT	TIMER	TYPCTCA	TYPFBA	TYPIBM1	TYPPRT	TYPTELE2	TYPTIMER
	TYP1052	TYP1403	TYP1443	TYP2305	TYP2314	TYP2319	TYP2501	TYP2540P	TYP2540R	TYP3088
	TYP3203	TYP3211	TYP3262	TYP3277	TYP3289E	TYP3310	TYP3330	TYP3340	TYP3350	TYP3370
	TYP3375	TYP3380	TYP3505	TYP3525	TYP3800	TYP38003	TYP4245	TYP4248	VCONCTL	VCONNICB
	VCONRDEV	VIRTUAL	VMBLOK	X40FFS	ZEROES					
DMKDEG	AFREE	AFRET	ASOFF	DELPAGES	DELSEGS	DMKBLDRL	DMKBLDRT	DMKCFPRR	DMKCFSAS	DMKCVTBD
	DMKCVTDB	DMKCVTHB	DMKERMSG	DMKFREE	DMKFRET	DMKPGSPO	DMKQCNPL	DMKSCNFD	DMKUDRFU	DMKUDRMD
	DMKUDRRV	FLAG	FTR4WCGM	F15	F16	F2	F3	F5	F8	LOCK
	NEWPAGES	NEWSEGS	PLFLAG1	PLIMSG	PLIST	PLNORET	PLR2	PLSIZE	PSA	R0
	R1	R10	R11	R12	R13	R14	R15	R2	R3	R4
	R5	R6	R7	SAVEAREA	SAVEREGS	SAVER0	SAVER1	SAVER2	SAVEWRK1	SAVEWRK2
	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	TYP38003	VMBLOK	VMSIZE
	VRALOC									
DMKDEI	AFREE	AFRET	ALOKSP	APSTAT1	APUOPER	AQCNWT	BLANK	BLANKS	CLASDASD	DMKCVTBH
	DMKCVTHB	DMKERMSG	DMKFREE	DMKFRET	DMKLOKIO	DMKQCNWT	DMKSCNFD	DMKSCNRU	DMKSSSDE	F8
	LOCK	LOCKSAV	LPUADDR	L3	L4	L7	L8	L9	MSSPRES	NORET
	PSA	PSAMSS	RDEVBLOK	RDEVDED	RDEVDISA	RDEVFLAG	RDEVFTR	RDEVLNKS	RDEVSER	RDEVSTAT
	RDEVTYPC	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R8	SAVEAREA	SAVEREGS	SAVER2	SAVEWRK1	SAVEWRK2
	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SYSVIRT	VIRTUAL	VMBLOK	

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
DMKDEX	AFREE	AFRET	AP	CC	DMKFREE	DMKFRET	IOBCAW	IOBLOK	IOBSENSE	IOBSPEC3
	LOCK	MP	PSA	RCWADDR	RCWCCNT	RCWCCW	RCWCOMND	RCWCTL	RCWFLAG	RCWGEN
	RCWPNT	RCWRCNT	RCWREL	RCWTASK	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK6	SAVEWRK8	SAVEWRK9	SILI	VMBLOK
DMKDGD	ACORETBL	ADSPCH	AFREE	AFREEP	AFRET	ALOKRM	ALOKSP	ALOKSY	AP	APSTAT1
	APTRAN	APTRLK	APUOPER	BALRSAVE	BALR0	BALR14	BLANK	BRING	CC	CD
	CLASDASD	CORFLAG	CORFLUSH	CORPGPNT	CORSHARE	CORSWPNT	CORTABLE	CPEXADD	CPEXBLOK	CPEXFPNT
	CPEXMISC	CPEXREGS	CPEXR0	CPEXSIZE	CPSHRLK	CPSTAT2	CPSTAT4	C1	DEFER	DMKDGFIN
	DMKDSPCH	DMKDSPRU	DMKFREE	DMKFREEP	DMKFREH1	DMKFRET	DMKIOSQV	DMKLOKDF	DMKPSACC	DMKPSASC
	DMKPTRAN	DMKPTRFR	DMKPTRLE	DMKPTRLK	DMKPTRUL	DMKPTTFT	DMKPTTPM	DMKSCNVU	DMKSSSMQ	DMKSTKCP
	DMKSTKDE	DMKSYSCS	DMKVMASH	FFS	F1	F15	F16	F3	F4	F4095
	F4096	F5	F6	F8	IDA	IOBCAW	IOBCYL	IOBFLAG	IOBHVC	IOBIRA
	IOBLOK	IOBMISC	IOBMISC2	IOBSIZE	IOERBLOK	IOEREXT	IOERSIZE	LOCK	LOCKSAV	LPUADDR
	L1	L16	L2	L20	L4	L5	L6	MP	MSSPRES	PCIF
	PSA	PSAMSS	RCWADDR	RCWCCW	RCWCNT	RCWCOMND	RCWCTL	RCWFLAG	RCWIO	RCWSHR
	RDEVBLOK	RDEVCMDK	RDEVCUP	RDEVECKD	RDEVFTR	RDEVLOW	RDEVMDL	RDEVSTA5	RDEVSTA6	READ
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER12	SAVER2
	SAVEWRK2	SAVEWRK4	SAVEWRK9	SHRLKCNT	SILI	SWPFLAG	SWPSHR	SYSVIRT	TYP2305	TYP2311
	TYP2314	TYP3330	TYP3340	TYP3350	TYP3375	TYP3380	VFAULT	VIRTUAL	VMBLOK	WRITE
	XKEYMODE	XPAGNUM	ZEROES							
DMKDGF	ACORETBL	ADSPCH	AFREE	AFRET	ALOKSY	APSTAT1	APTRAN	APUOPER	BRING	CC
	CPEXADD	CPEXBLOK	CPEXREGS	CSW	C1	DEFER	DMKDGDUL	DMKDSPCH	DMKDSPRU	DMKFREE
	DMKFRET	DMKIOSQV	DMKLOKDF	DMKPSACC	DMKSTKDE	DMKVMASH	FFS	F16	F8	IOBCAW
	IOBCC1	IOBCC3	IOBCSW	IOBFATAL	IOBFLAG	IOBHVC	IOBIOER	IOBLINK	IOBLOK	IOBMISC
	IOBSIZE	IOBSTAT	IOERBLOK	IOERDATA	IOEREXT	IOERPNT	IOERSIZE	LOCK	LOCKSAV	LPUADDR
	L16	L4	L6	PSA	RCWADDR	RCWCCW	RCWCOMND	RDEVBLOK	RDEVECKD	RDEVLOW
	RDEVMDL	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R8	R9	SILI	VMBLOK	ZEROES	
DMKDIA	ADSPCH	AFREE	AFREEP	AFRET	ALOKSP	APSTAT1	APUOPER	AQCNWT	BALRSAVE	BLANKS
	CLASGRAF	CLASTERM	DMKCVTBH	DMKCVTHB	DMKDIFDI	DMKDSPCH	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET
	DMKIOSHA	DMKLOKIO	DMKQCNWT	DMKQCOCL	DMKQCOSY	DMKSCHRT	DMKSCNAU	DMKSCNFD	DMKSCNRD	DMKSCNRN
	DMKSCNRU	DMKSCNVD	DMKSCNVU	FFS	F240	F255	F3	IOBCP	IOBFLAG	IOBIOER
	IOBIRA	IOBLINK	IOBLOK	IOBMISC	IOBRADD	IOBSIZE	IOBUSER	IOERBLOK	IOEREXT	IOERSIZE
	LOCK	LOCKSAV	LPUADDR	NICBLOK	NICCIBM	NICDTYPE	NICD3275	NICD3277	NICD3278	NICEPAD
	NICLINE	NICSIZE	NICSTAT	NICSWEP	NICTYPE	PSA	RDEVAIOB	RDEVBLOK	RDEVCORD	RDEVDIIP
	RDEVEPDV	RDEVHT	RDEVNICL	RDEVPEND	RDEVPS	RDEVSNA	RDEVSNRB	RDEVSTAT	RDEVSTA3	RDEVTYPC
	RDEVTYPE	RDEVUSER	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	SAVEAREA	SAVEREGS	SAVER0
	SAVER11	SAVER2	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8
	SAVEWRK9	SNARBLOK	SNARDIPG	SNARFG3	SNARLUN	SNARVDEV	SNARVMB	TIMEDISP	TRQBSIZE	TYPBSC
	TYPIBM1	TYPTELE2	TYP3210	TYP3277	TYP3278	VCONCTL	VCONEXTN	VCONNICB	VCONOPT	VCONRDEV
	VCONREMD	VCONREMF	VCONRMSZ	VCON3270	VMBLOK					
DMKDIB	ADSPCH	AFREE	AFREEP	AFRET	ALOKSP	APSTAT1	APTRLK	APUOPER	AQCNWT	ASYSVM
	BALRSAVE	BLANKS	CC	CD	CE	CLASGRAF	CLASSPEC	CLASTERM	CMDREJ	CPEXADD
	CPEXBLOK	CPEXSIZE	CSWLNCP	DE	DFRET	DMKACODV	DMKBLDVM	DMKCFQRD	DMKCVTBD	DMKCVTBH
	DMKCVTHB	DMKDSPCH	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKHPTDI	DMKLOKIO	DMKLOKSW	DMKPGTMV
	DMKPTRLK	DMKPTRUL	DMKQCNWT	DMKQCOSY	DMKRIORN	DMKRNHND	DMKSCHRT	DMKSCNAU	DMKSCNFD	DMKSCNRD
	DMKSCNRN	DMKSCNRU	DMKSCNVD	DMKSCNVU	DMKSTKCP	DMKSTKIO	DMKSYSND	DMKSYSRM	DMKVCBRS	DMKVCTLO
	FFS	F1	F240	F255	F3	F7	IDA	IL	INTREQ	IOBCAW

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	IOBCC1	IOBCC3	IOBCSW	IOBFLAG	IOBIOER	IOBLOK	IOBRCAW	IOBRSTRT	IOBSTAT	IOBUSER
	IOERBLOK	IOERCCW	IOERCSW	IOERDATA	IOERSIZE	LASTUSER	LOCK	LOCKSAV	LOGHOLD	LPUADDR
	NICBLOK	NICDISA	NICDMSG	NICDXSC	NICENAB	NICEPMD	NICFLAG	NICRDED	NICRFLG	NICSESN
	NICSIZE	NICSTAT	NICSWEP	NICTMAT	NICUSER	NICVDEVB	NORET	OPERATOR	PRGC	PRIORITY
	PRTC	PSA	RCUBLOK	RCUDVTBL	RCWADDR	RCWCCW	RCWCNT	RCWCOMND	RCWCTL	RCWFLAG
	RCWINVL	RDEVACTV	RDEVADD	RDEVATT	RDEVBASE	RDEVBLOK	RDEVCORD	RDEVCTL	RDEVCUA	RDEVCYL
	RDEVDED	RDEVDISA	RDEVDISB	RDEVEPDV	RDEVEPMD	RDEVFLAG	RDEVHIO	RDEVLCEP	RDEVLNCP	RDEVNICL
	RDEVNRDY	RDEVPREP	RDEVPS	RDEVRCVY	RDEVRUN	RDEVSNA	RDEVSNRB	RDEVSTAT	RDEVSTA3	RDEVTFLG
	RDEVTMAT	RDEVTYPC	RDEVTYPE	RUNUSER	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	SAVER0	SAVER11	SAVER2	SAVER8	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4
	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	SILI	SKIP	SNARBLOK	SNARDIS	SNARFG1
	SNARLUN	TIMEDISP	TRQBFPNT	TRQBLOK	TRQBSIZE	TYPBSC	TYPCTCA	TYP3277	TYP3278	UC
	VCONBFSZ	VCONBUF	VCONCTL	VCONNICB	VCONRDEV	VCONRMSZ	VMBLOK	ZEROES		
DMKDID	ADSPCH	AFREE	AFREEP	AFRET	ALOKSP	AMCHAREA	APSTAT1	APUOPER	ARIODC	ARIODV
	ASYSVM	CE	CLASDASD	CLASFBA	CLASGRAF	CLASSPEC	CLASTAPE	CLASURI	CLASURO	CPEXADD
	CPEXBLOK	CPEXREGS	CPEXR11	CPEXSIZE	CRTEXT	CRTEXTSZ	CUE	DE	DMKCCHCF	DMKCVTAB
	DMKCVTBH	DMKDSPCH	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKGRFIN	DMKIOEMI	DMKIOSHA	DMKIOTRC
	DMKLOKIO	DMKSCHST	DMKSCNEP	DMKSCNRA	DMKSCNRD	DMKSCNRU	DMKSTKIO	DMKSTKMP	DMKSTKOP	DMKSYSTV
	FTR3270E	F1	F10	F15	F2	F3	F4	F8	IFCC	INTREQ
	IOBCAW	IOBCLRIO	IOBCP	IOBCSW	IOBFATAL	IOBFCNS	IOBFLAG	IOBFPNT	IOBHIO	IOBHVC
	IOBIOER	IOBIRA	IOBLINK	IOBLOK	IOBMID	IOBMINI	IOBMISC2	IOBPROC	IOBRADD	IOBRCNT
	IOBRES	IOBSIZE	IOBSPEC	IOBSPEC4	IOBSPEC5	IOBSTAT	IOBUNSL	IOBUSER	IOERBLOK	IOERCCH
	IOERCCUA	IOERCHID	IOERCPID	IOERCSW	IOERDATA	IOERECSW	IOEREXT	IOERLEN	IOERLOGL	IOERPNT
	IOERSIZE	IOER2860	IOER2870	LOCK	LPUADDR	L2	MCHAREA	MCHMODEL	MIHCE	MIHDASD
	MIHGRAF	MIHMISC	MIHMSG	MIHMSGCS	MIHMSGDV	MIHMSGDW	MIHMSGID	MIHMSGLN	MIHTAPE	MIHUR
	MODEL135	PMAMODE	PMASTAT	PROCIPL	PSA	RCHADD	RCHBLOK	RCHSTIDC	RCUADD	RCUBLOK
	RCUCHA	RCUDISA	RCUFIOB	RCUPRIME	RCUQCNT	RCUSTAT	RCUSUB	RCUTYPE	RDEVADD	RDEVAIOB
	RDEVBLOK	RDEVBUZY	RDEVCSW	RDEVCUA	RDEVCUB	RDEVDISA	RDEVDROP	RDEVFIOB	RDEVFORC	RDEVFTR
	RDEVIOER	RDEVMID	RDEVNRDY	RDEVPEND	RDEVPIOB	RDEVPROC	RDEVPS	RDEVPTHS	RDEVQBSY	RDEVQIOB
	RDEVRSTA	RDEVSCHD	RDEVSIZE	RDEVSTAT	RDEVSTA2	RDEVSTA3	RDEVSTA4	RDEVSTA7	RDEVTYPC	RDEVTYPE
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SCAN	SYSVIRT
	TRQBIRA	TRQBLOK	TRQBSAVE	TRQBSIZE	TRQBTOD	TRQBUSER	TRQBVAL	TYP1053	TYP3284	TYP3289E
	TYP3330	TYP3800	TYP38003	TYP38008	TYP3851	UC	VIRTUAL	VMBLOK		
DMKDIF	AFRET	ALOKSP	APSTAT1	APUOPER	AQCNWT	ARIOCU	ARIODV	ASYSVM	BALR1	CCDESMD
	CLASGRAF	CLASTERM	CONCCW3	CONDATA	CONDCNT	CONSYSR	CPEXSIZE	CPSTAT5	CRESDQ	CRESIMD
	CRTEXT	CSETDSM	CSWLMEP	CTRMLTR	DE	DFRET	DMKCVTAB	DMKCVTBD	DMKDIBDR	DMKFRET
	DMKIOSQR	DMKLOKIO	DMKLOKSW	DMKPGTMV	DMKQCNWT	DMKQCOCL	DMKRNHND	DMKSCHRT	DMKSCNRD	DMKSCNRU
	DMKSCNVU	DMKSTKIO	DMKSYSCK	DMKSYSND	DMKVIOIN	F1	GRAFDEV	IOBCSW	IOBIRA	IOBLOK
	IOBRCAW	IOBSIZE	IOBUSER	IOBVADD	LASTUSER	LOCK	LOCKSAV	LPUADDR	NICATRB	NICBLOK
	NICDISA	NICDXSC	NICEPAD	NICEPMD	NICFLAG	NICLTRC	NICNAME	NICQPNT	NICRDED	NICRFLG
	NICSESN	NICSIZE	NICSTAT	NICTELE	NICTMAT	NICTRQ	NICTYPE	NICUSER	NICVDEVB	NORET
	OPERATOR	PSA	RCHBLOK	RCHCUTBL	RCUBLOK	RCUDVTBL	RDEVADD	RDEVAIOB	RDEVAIRA	RDEVATT
	RDEVBASE	RDEVBLOK	RDEVCON	RDEVCUA	RDEVCYL	RDEVDED	RDEVDIIP	RDEVEPDV	RDEVEPLN	RDEVEPMD
	RDEVFLAG	RDEVPTHS	RDEVSNRB	RDEVSTAT	RDEVSTA3	RDEVSTA4	RDEVTFLG	RDEVTMAT	RDEVTRQ	RDEVTYPC
	RDEVTYPE	RDEVUSER	RDIDX	RUNUSER	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	SAVERETN	SAVER11	SAVER13	SAVEWRK1	SAVEWRK2	SAVEWRK5	SAVEWRK6	SAVEWRK7
	SAVEWRK8	SAVEWRK9	SNARBLOK	SNARDIAL	SNARDIPG	SNARFG2	SNARFG3	TIMEDISP	TRQBFPNT	TRQBLOK
	TRQBSIZE	TYPIBM1	TYPTELE2	TYP3277	VCONBFSZ	VCONBUF	VCONCTL	VCONRMSZ	VMBLOK	ZEROES
DMKDIR	ATTN	BLANK	BUSY	CAW	CC	CD	CE	CL	CLASDASD	CLASFBA

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	CLASGRAF	CLASSPEC	CLASTERM	CLASURI	CLASURO	CSW	CUE	DE	DMKCFWEP	ERRMSG
	FLAG	FSCBANIT	FSCBBUFF	FSCBD	FSCBFN	FSCBNOIT	FSCBSIZE	FSTD	FSTLRECL	FSTRECCT
	FSTRECFM	FTR2311B	FTR2311T	FTR3088	FTR3270	FTR4WCGM	H7	L1	L16	L4
	L4096	L7	NOTUSED	READ	ROUTE	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVERET	SCAN	SENSE	SILI	SKIP	TIMER	TYPCTCA	TYPIBM1	TYPTELE2
	TYPTIMER	TYP1052	TYP1403	TYP1443	TYP2305	TYP2311	TYP2314	TYP2501	TYP2540P	TYP2540R
	TYP3138	TYP3148	TYP3158	TYP3203	TYP3210	TYP3211	TYP3215	TYP3262	TYP3277	TYP3289E
	TYP3330	TYP3340	TYP3350	TYP3375	TYP3380	TYP3505	TYP3525	TYP3800	TYP38003	TYP4245
	TYP4248	UC	UE	UIUCBLOK	UIUCCHN	UIUCDASD	UIUCDISP	UIUCGLBL	UIUCLAST	UIUCMLIM
	UIUCPRTY	UIUCRES	UIUCREVK	UIUCSIZE	UIUCSTAT	UIUCUSER	VIRTUAL	WRITE		
DMKDMP	ALOCBLOK	ALOCCYL2	ALOCPNT	ALOCRECS	ATTN	AVMREAL	BALRSAVE	BALR2	BUSY	CAW
	CC	CE	CHGSFB	CLASDASD	CLASFBA	CLASTAPE	CLASURO	CORCP	CORDISA	CORFLAG
	CORFPNT	CORTABLE	CORVM	CPABEND	CPCREGO	CPID	CPSPMODE	CPSTAT2	CPSTAT4	CPSTAT5
	CPUID	CPULOG	CPXSTOR	CSW	CUE	C0	C14	C15	C2	C6
	DAMAGRPT	DE	DMKCPEID	DMKDMQEN	DMKDMQGP	DMKDMQLW	DMKDMQMC	DMKDMQPC	DMKDMQSW	DMKDMQTL
	DMKPRGMC	DMKRIODV	DMKRIOPR	DMKRSPRD	DMKSCNRU	DMKSYSCH	DMKSYSCK	DMKSYSCS	DMKSYSRC	DMKSYSRM
	DMKSYSRV	DMKSYSSP	DMKSYSTP	DMKSYSVM	DMKVFRSV	DMKVRR	DMKVRRIS	DMKVRROP	DMKVRSSV	DMPABEND
	DMPCPUID	DMPFLAG	DMPFPRS	DMPGPRS	DMPINREC	DMPIPCS	DMPKEY	DMPKYREC	DMPLCORE	DMPPGMAP
	DMPPGMP2	DMPPRFRG	DMPPROCA	DMPSYSRM	DMPSYSRV	DMPTODCK	DUMPSAVE	D4	EXTMODE	FLAG
	FPRLOG	FXDLOG	F1	GRLOG	HALFPAGE	HARDSTOP	IDA	INTPRL	INTREQ	INTTIO
	IOMASK	IONPSW	IPLCCW1	IPLPSW	IPUADDR	IPUADDRX	LAP370E	LOCK	L1	L16
	L2	L3	L4	L5	L8	MACRO	MCHEK	PMAAVAIL	PMAMODE	PMASTAT
	PRNPSW	PROPSW	PSA	RDCBLKPG	RDCBLOK	RDEVALLN	RDEVBLOK	RDEVDISA	RDEVECKD	RDEVMDL
	RDEVMD02	RDEVRDC	RDEVSTAT	RDEVTYPC	RDEVTYPE	RDEVUSER	RDIDX	RDRCHN	RECBLOK	RECCYL
	RECMAP	RECMAX	RECPNT	RECUSED	RSRTNPSW	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SENSE	SFBDATE	SFBDIST	SFBDUMP	SFBLAST	SFBLOK	SFBPNT	SFBSIZE	SFBSTART
	SFBTIME	SFBTYPE	SIGCR	SIGIPR	SIGREST	SIGSENSE	SIGSSS	SIGSTOP	SILI	SKIP
	SM	SPMPFX	SPXCR6	TODATE	TYPPRT	TYP1403	TYP2314	TYP3340	TYP3350	TYP3375
	TYP3380	UC	VECSAOK	VECSTAT	VECUSER	VMBLOK	WAIT	Y0	Y2	Y4
	Y6	ZEROES								
DMKDMQ	ALARM	ALL	APSTAT1	ATTPSA	AVMREAL	BLANK	BREAKSUP	BUSY	CAW	CC
	CCC	CDC	CE	CLASTAPE	CLKCOMP	CORCP	CORDISA	CORFLAG	CORFPNT	CORTABLE
	CORVM	CPID	CPUID	CPUTIMER	CPVR	CSW	CTLREGS	CUE	C0	C2
	DE	DMKCPEND	DMKDMPAA	DMKDMPC2	DMKDMPDS	DMKDMPDV	DMKDMPMA	DMKDMPPA	DMKDMPPX	DMKDMPRY
	DMKDMPSW	DMKDMPTR	DMKOPRWT	DMKPTTPM	DMKRIOCN	DMKSCNRD	DMKSCNRU	DMKSYSCS	DMKSYSSP	DMKSYSVM
	DMKVRR	DMKVRRIS	DMKVRROP	DMKVRSSV	DMPAP	DMPIPCS	DMPMP	DMPROC	DMPSMSG	DMPSMSGL
	DMPSTAT	DMPUNI	DOUBLESP	DUMP	EXTMODE	FFS	FPREGS	GREGS	IFCC	INTTIO
	IOBSIZE	IOMASK	IONPSW	LINE	LINESUP	LOCK	LOWCORE	MAINPSA	MCHEK	MONAIOB
	MONARDB	MONCAD	MONCOM	MONFLAG2	MONFLAG3	PREFRG	PROCIPL	PSA	RCHADD	RCHBLOK
	RCUADD	RCUBLOK	RCUCHA	RCUPRIME	RCUSUB	RCUTYPE	RDEVADD	RDEVBLOK	RDEVCUA	RDEVCUB
	RDEVTYPC	RDEVTYPE	RSRTNPSW	R0	R1	R10	R11	R12	R13	R14
	R15	R2	R3	R4	R5	R6	R7	R8	R9	SENSE
	SIGREST	SIGSENSE	SILI	SKIPSUP	SM	SPOOLED	STORSIZE	TODCLOCK	TRUN	TYP3420
	TYP3422	TYP3430	TYP3480	TYP8809	UC	UE	VMBLOK	WAIT		
DMKDNC	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9				
DMKDRD	ACORETBL	ADSPCH	AFREE	AFREEP	AFRET	APTRAN	APTRLK	ARSPRD	ASYSVM	BRING
	CLASFBA	CLASURI	CONNECT	CORSWPNT	CORTABLE	C1	DEFER	DMKCKTSD	DMKCKTUU	DMKCVTAB
	DMKCVTBH	DMKDMPAL	DMKDMPAU	DMKDMPDV	DMKDMPRC	DMKDMPSF	DMKDSPCH	DMKERMSG	DMKFREE	DMKFREEP

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	DMKFRET	DMKHVCPC	DMKLOCRD	DMKLOCRQ	DMKPGTDS	DMKPGTDT	DMKPGUDU	DMKPGUSD	DMKPGUVG	DMKPGUVR
	DMKPSASP	DMKPTRAN	DMKPTRLK	DMKPTRUL	DMKRPAGT	DMKRPAPT	DMKSCHST	DMKSCNMU	DMKSCNRD	DMKSCNVU
	DMKSYM	DMKSYSDU	DMKSYSOW	DMKVSDAD	DMKVSDDL	DMKVSFID	DMKVSFNU	DMKVSGRI	DMKVSUCR	FFS
	F0	F1	F255	F256	F4096	F5	F8	IOERETN	IOKEEP	LOCK
	L1	L3	OWNDLIST	OWNDRDEV	PSA	RDEVBLOK	RDEVTYPC	RDEVTYPE	R0	R1
	R10	R11	R12	R13	R14	R15	R2	R3	R4	R5
	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER0	SAVER2	SAVER6	SAVEWRK1
	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	SFBCLAS	SFBCOPY
	SFBDUMP	SFBEOF	SFBFILID	SFBFLAG	SFBFLAG2	SFBFLAG3	SFBFLAG4	SFBINUSE	SFBINVS	SFBLAST
	SFBLOK	SFBMISC1	SFBMON	SFBOPEN	SFBORIG	SFBPNT	SFBSEEN	SFBSIZE	SFBSTART	SFBSYSID
	SFBTUSE	SFBTYPE	SFBUHOLD	SFBUSER	SKIP	SPCHAR	SPFILID	SPLINK	SPNXTPAG	SPPREPAG
	SPRECNUM	SPSIZE	SWPCYL	SWPFLAG	SWPKEY1	SWPKEY2	SWPRECMP	SYSTEM	TRQBIRA	TRQBLOK
	TRQBUSER	TRQBVAL	TRQREGS	TRQREGSD	TRQREGSZ	TYPPRT	TYPPUN	TYPRDR	TYP2305	TYP2319
	TYP3330	TYP3340	TYP3350	TYP3375	TYP3380	VMBLOK	VSPCAW	VSPCCW	VSPDPAGE	VSPLCTL
	VSPSFBLK	VSPSIZE	XPAGNUM	X2048BND						
DMKDRE	ACORETBL	ADSPCH	AFREE	AFRET	AP	APTRAN	APTRLK	ARSPRD	ASYSVM	BRING
	CLASFBA	CORSWPNT	CORTABLE	C1	DEFER	DMKCKTSD	DMKCKTUU	DMKCVTAB	DMKCVTBH	DMKDMPAL
	DMKDMPAU	DMKDMPDV	DMKDMPRC	DMKDMPSF	DMKDSPCH	DMKERMSG	DMKFREE	DMKFRET	DMKHVCPC	DMKLOCRD
	DMKLOCRQ	DMKPGTDS	DMKPGTDT	DMKPGUDU	DMKPGUSD	DMKPGUVG	DMKPGUVR	DMKPSASP	DMKPTRAN	DMKPTRLK
	DMKPTRUL	DMKRPAGT	DMKRPAPT	DMKSCHST	DMKSCNMU	DMKSCNRD	DMKSYM	DMKSYSDU	DMKSYSOW	DMKVSFID
	DMKVSFNU	DMKVSGRI	F0	F1	F256	F4096	F5	IOERETN	IOKEEP	LOCK
	L1	L3	MP	OWNDLIST	OWNDRDEV	PSA	RDEVBLOK	RDEVTYPC	RDEVTYPE	R0
	R1	R10	R11	R12	R13	R14	R15	R2	R3	R4
	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER0	SAVER2	SAVER6
	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK6	SAVEWRK7	SFBFILID	SFBFLAG	SFBFLAG3	SFBFLAG4
	SFBINUSE	SFBINVS	SFBLAST	SFBLOK	SFBORIG	SFBPNT	SFBSEEN	SFBSIZE	SFBSTART	SFBSYSID
	SFBTUSE	SFBUSER	SKIP	SPCHAR	SPFILID	SPLINK	SPSIZE	SWPCYL	SWPFLAG	SWPRECMP
	SYSTEM	TRQBIRA	TRQBLOK	TRQBUSER	TRQBVAL	TRQREGS	TRQREGSD	TRQREGSZ	TYP2305	TYP2319
	TYP3330	TYP3340	TYP3350	TYP3375	TYP3380	VMBLOK	XPAGNUM	X2048BND		
DMKDSB	ADSPCH	AFREE	AFREEP	AFRET	ALARM	ALOKSP	AP	APSTAT1	APUOPER	AQCNWT
	ASYSVM	BLANKS	CC	CCC	CDC	CL	CLASDASD	CLASFBA	CLASTAPE	CPEXADD
	CPEXBLOK	CPEXREGS	CPEXSIZE	DE	DFRET	DMKCVTBH	DMKDSPCH	DMKFREE	DMKFREEP	DMKFRET
	DMKIOESD	DMKIOSQR	DMKLOKIO	DMKQCNWT	DMKSCNRU	DMKSSSEN	DMKSSSMQ	DMKSTKCP	FTREXTSN	FTRRPS
	FTRRSRL	FTR35MB	FTR70MB	IFCC	IOBCAW	IOBCC3	IOBCSW	IOBFATAL	IOBIOER	IOBIRA
	IOBLINK	IOBLOK	IOBRADD	IOBREL	IOBSIZE	IOBSPEC	IOBSPEC2	IOBSTAT	IOBTIO	IOBUSER
	IOERBLOK	IOERCCRA	IOERCCRL	IOERCPID	IOERCSW	IOERDATA	IOERDW	IOEREXT	IOERLEN	IOERLOC
	IOERSIZE	IOERSNSZ	IOERVSER	LOCK	LOCKSAV	LPUADDR	LPUADDRX	MP	NORET	NOTUSED
	OPERATOR	PROCIPL	PSA	RCHADD	RCHBLOK	RCUADD	RCUBLOK	RCUCHA	RCUPRIME	RCUSUB
	RCUTYPE	RDCBLOK	RDCFEAT	RDCRRLSE	RDEVADD	RDEVBLOK	RDEVCFLT	RDEVCUA	RDEVCUB	RDEVCUP
	RDEVDED	RDEVDISA	RDEVFLAG	RDEVFTR	RDEVMOUT	RDEVOWN	RDEVPPAG	RDEVPRDV	RDEVRDC	RDEVRRES
	RDEVSEL	RDEVSER	RDEVSTAT	RDEVSTA4	RDEVSTA5	RDEVSTA6	RDEVSYS	RDEVTYPC	RDEVTYPE	RDEVVMNT
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	SAVEAREA	SAVEREGS	SILI	SYSVIRT	TYP2305
	TYP2314	TYP3330	TYP3340	TYP8809	UC	VIRTUAL	VMBLOK	ZEROES		
DMKDSP	ACTIVTRQ	ADSPCH	AEXTSP	AFREE	AFREEP	AFRET	ALOKDS	ALOKSP	ALOKSY	ALOKTRL
	ALOKVM	APSTAT1	APSTAT2	APSTAT3	APSTAT4	APTRAN	APUOPER	ARIODC	ARIODV	ASYSVM
	ATTN	AVMREAL	BALR2	BALR9	BRING	BUSY	CCC	CCTFLAG1	CCTFLCNT	CCTRPYN
	CCTRPYP	CC3	CDC	CLASGRAF	CLASTERM	CODE	CONACTV	CONPNT	CONSTAT	CONTASK
	CPAPRPND	CPCCHLK	CPCREG0	CPCREG6	CPCREG8	CPDASAAV	CPDASAON	CPEX	CPEXADD	CPEXBLOK
	CPEXBPNT	CPEXDEFR	CPEXFPNT	CPEXPRIO	CPEXPROC	CPEXREGS	CPEXR10	CPEXR11	CPEXR12	CPEXR13
	CPEXR8	CPEXSIZE	CPEXTYPE	CPFRELK	CPMCHSE	CPMICON	CPPTLBR	CPRSTPND	CPRUN	CPSEGPRT
	CPSHRLK	CPSPMODE	CPSTATUS	CPSTAT2	CPSTAT3	CPSTAT4	CPSTAT5	CPSUPER	CPSYSLK	CPTIDLE

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	CPTIONT	CPTPAGE	CPUID	CPWAIT	CPXTRANS	CP370EON	CSW	CUE	C0	C1
	C11	C2	C6	C7	C8	C9	DEFER	DMKCCHER	DMKCCHRF	DMKCFMBK
	DMKCVTBH	DMKDMPC2	DMKERMSG	DMKFREDB	DMKFREE	DMKFREEP	DMKFRESW	DMKFRET	DMKFRETL	DMKFRET1
	DMKHPTEX	DMKIOSER	DMKIOSRC	DMKIOSW1	DMKIUBRK	DMKLOKCT	DMKLOKDF	DMKLOKRL	DMKLOKSY	DMKLOKTR
	DMKMCHSE	DMKMCTPR	DMKPERIL	DMKPMAIO	DMKPMARW	DMKPMASW	DMKPMAWT	DMKPRGSM	DMKPSAPO	DMKPTRFP
	DMKPTRSC	DMKQVMRT	DMKREIPL	DMKSCHDL	DMKSCHQ1	DMKSCHQ2	DMKSCHRL	DMKSCHTQ	DMKSCNVU	DMKSELFD
	DMKSELFE	DMKSTKCP	DMKSTKDE	DMKSWAPD	DMKTMRPT	DMKTRCEX	DMKTRCIO	DMKTRCIT	DMKTRCPG	DMKVATAB
	DMKVATAT	DMKVATBC	DMKVATEX	DMKVATMD	DMKVCRNR	DMKVIOCL	DMKVIOMK	DMKVMASH	DMKVMCEX	DMKWAIST
	DMKWAITA	DPOKTLB	DPSTAT	DSPA	DSPB	DSPCH	DSPRQ	DSPRU	EMSINQSC	EMSMASK
	EMSPEND	EMSPEXT	EMSPQUI	EMSREC	EMSREXT	EMSRQUI	EXNPSW	EXOPSW	EXTMODE	FFS
	FRLKPROC	F1	F255	F3	F4096	IDLEWAIT	IFCC	INMSFBLK	INTEX	INTEXF
	INTPRL	INTTIO	IOBBPNT	IOBCSW	IOBFLAG	IOBFPNT	IOBIRA	IOBLOK	IOBMISC	IOBPAG
	IOBRADD	IOBSIZE	IOBUSER	IOERBLOK	IOERDATA	IOMASK	IONPSW	IONTWAIT	IOOPSW	IPLCCW1
	IPUADDR	IPUADDRX	IUCVBLOK	IUCVCODE	IUCVMASK	LAP370E	LASTUSER	LOCK	LOCKSAV	LOKREQ
	LPUADDR	LPUADDRX	MFAMASK	MICBLOK	MICEVMA2	MICEVMA3	MICIPTP2	MICISKE	MICLRA2	MICPEND
	MICPTLB2	MICRRBE	MICSSKE	MICSTBVR	MICVIP	MICVPFR2	MSSFINTR	MVSA370E	OFF	PAGEWAIT
	PCI	PERADD	PERBLOK	PERCODE	PERCR9	PERMODE	PGADDR	PGBLOK	PGBSIZE	PGPNT
	PGWAITIM	PMAAVAIL	PMASTAT	PREFIXA	PREFIXB	PRNPSW	PROBMODE	PROBSTRT	PROBTIME	PROB370E
	PROCIO	PROCIPL	PROPSW	PROTBLOK	PROTDPSW	PROTERR	PROTEXTL	PROTFLAG	PROTPSW	PROTRCNT
	PROTREI	PROTREIL	PSA	PSADSPRQ	PSAEXT	PSAEXTX	PWTPAGES	PXA	QUANTUM	QUANTUMR
	RDEVADD	RDEVAIOB	RDEVBLOK	RDEVCON	RDEVDED	RDEVFIOB	RDEVSCHD	RDEVSIZE	RDEVSNRB	RDEVSTAT
	RDEVSTA4	RDEVTYPC	RQLKBASE	RQLOCK	RUNCRINV	RUNCR0	RUNCR1	RUNPSW	RUNUSER	RUN370E
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SAS	SIGDISP	SIGMASK	SIGXC
	SM	SNARBACH	SNARBLOK	SNARFG1	STACKVM	SYNCMASK	TEMPR0	TEMPR10	TEMPR6	TEMPSAVE
	TIMEDISP	TIMER	TRACCURR	TRACEND	TRACFLG2	TRACFLG3	TRACPROC	TRACSTRT	TRACSVCR	TRACOA
	TRACOC	TRAC10	TRAFPFLG	TRANMODE	TRAPCR8	TRAPTT	TRCFPRUN	TRCRUN	TRCUNBLK	TRCUNSTK
	TREXFLAG	TREXIN1	TREXIN2	TREXNDSP	TREXT	TRLLABEL	TRLLOCK	TRQBFPNT	TRQBLOK	TRQBUSER
	TRQBVAL	UC	VECF	VECSAOK	VECSTAT	VECUSER	VMBLOK	VMSIZE	WAIT	WAITEND
	WAITSTRT	XCDISP	XCMASK	XCPEND	XKEYASST	XKEYMODE	XRIGHT16	XRIGHT24	XTNDCR2	XTNDLOCK
	X40FFS	Y0	Y2	Y4	Y6	ZEROES				
DMKEIG	CCC	CCHCMDV	CCHDAV	CCHDI	CCHLOG80	CCHRCV	CCHREC	CCHUSV	COMPFES	COMPSEL
	COMPSYS	CSW	FFS	IFCC	IGBLAME	IGTERMSQ	IGVALIDB	INTERCCH	IOELPNTR	IOERBLOK
	PSA	RTCODE0	RTCODE1	RTCODE2	RTCODE3	RTCODE4	RTCODE5	R0	R1	R12
	R13	R14	R15	R2	R3	R4	R9	SAVEAREA	SAVEWRK1	SAVEWRK9
	TERMSYS	TIOCCH								
DMKENT	ADSPCH	AP	APSTAT1	APSTAT4	APUOPER	ARIOCH	ARIOCT	AUTGO	DMKCVTAB	DMKDSPCH
	DMKMIAEN	DMKMIAIN	DMKMIAKC	DMKMNIFI	DMKMNLCP	DMKMNLTQ	DMKPRGMC	DMKSCHST	DMKSYSAT	F1
	LOCK	LPUADDR	LPUADDRX	MNCHDAT1	MNCHDAT2	MNCHDT11	MNCHDT22	MNCHLIST	MNCHSAMP	MNCHSAM1
	MNCHSIZ	MNCHSIZE	MNCHSZ	MNCUBSY	MNDEVLEN	MNDEVLST	MNDVBSY	MNDVBSY2	MNDVCNT	MNDVLEN
	MNDVMORE	MNRDEVB	MN602ADD	MN602CUB	MN602CUQ	MN602DEV	MN602DLN	MN602DVQ	MN602DV2	MN602HDR
	MN602HLN	MN602MLN	MN602SAM	MN603CBI	MN603CB2	MN603CB3	MN603CB4	MN603CH	MN603CQI	MN603CQ2
	MN603CQ3	MN603CQ4	MONCHPTR	MONCOM	MONDVLST	MONUTRB	MP	MPUOPER	POFFLINE	PREFIXA
	PREFIXB	PROCIO	PROCIPL	PSA	RCHADD	RCHBLOK	RCHQCNT	RCUADDR	RCUBLOK	RCUBUSY
	RCUCHA	RCUPRIME	RCUQCNT	RCUSTAT	RCUSUB	RCUTYPE	RDEVADD	RDEVBLOK	RDEVBUZY	RDEVCUA
	RDEVPROC	RDEVQCNT	RDEVSTA4	R0	R1	R10	R11	R12	R13	R14
	R15	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA
	SAVEREGS	TRQBLOK	TRQBTOD	TRQBVAL	VMBLOK					
DMKEPS	AFREE	AFRET	BLANKS	BUFFER	BUFINLTH	BUFNXT	BUFSIZE	CLASGRAF	CLASTERM	DMKERMCP
	DMKFREE	DMKFRET	DMKQCORD	DMKSCNFD	EDIT	FFS	F2	F4	F8	INHIBIT
	LOCK	NOTRESP	PSA	RDEVBLOK	RDEVTYPC	R0	R1	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R8	R9	SAVEAREA	SAVEREGS

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)								
	SAVER0	SAVER11	SAVEWRK1	SAVEWRK6	TYPBSC	UCASE	VMBLOK			
DMKERM	AFREE	AFRET	ALARM	APTRAN	ASYSOP	ASYSVM	BLANKS	BRING	BUFCNT	BUFFER
	BUFINLTH	BUFSIZE	CLASTERM	COUNT	C1	DEFER	DFRET	DMKCVTBD	DMKCVTDB	DMKFREE
	DMKFRET	DMKPTRAN	DMKQCNPL	DMKSYSRM	ERRMSG	F0	F1	F2	F255	F4
	IOERETN	IOKEEP	LANGADDR	LANGBLOK	LANGDBCS	LANGFLAG	LANGHIGH	LANGLOW	LANGNTRY	LANGNTSZ
	LANGPAGE	LANGSIZE	LANGVMB	LANGVMBK	LOCK	NICBLOK	NICSIZE	NICWTH	NORET	OPERATOR
	PLFLAG1	PLFRET	PLIST	PLLED	PLR2	PLSIZE	PSA	RDEVBLOK	RDEVNICL	RDEVTYPC
	RDEVWTH	REPCNT	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS
	SAVER0	SAVER1	SAVER2	SAVER3	SAVEWRK1	SAVEWRK2	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7
	SAVEWRK8	SAVEWRK9	SPECIALV	SYSTEM	TYPBSC	VMBLOK	XRIGHT16			
DMKERP	AP	CKPLOAD	CLASFBA	CPEXBLOK	CPEXFPNT	CPEXR6	DMKCKFBL	DMKCKFCL	DMKCKFCQ	DMKCKFEP
	DMKCKFES	DMKCKFIP	DMKCKFIQ	DMKCKFMQ	DMKCKFMX	DMKCKFVT	DMKCKNIO	DMKCKNPW	DMKCKNRD	DMKCKP
	ERRBLOK	ERROBR	ERRSRDEV	FFS	FLAG	FREESAVE	LOCK	MP	PSA	RECFLAG1
	RECNXT	RECPAG	RECPAGFL	R0	R1	R10	R12	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9			
DMKEXT	ACTIVTRQ	ADSPCH	AFREEP	ALOKSP	ALOKSY	AP	APSTAT1	APSTAT3	APSTAT4	APSTAT5
	APUOPER	ASYSOP	ASYSVM	AVMREAL	BLANKS	BUSY	CLASGRAF	CLASTERM	CLEAR	CODE
	CPAPRINP	CPAPRPND	CPCREG0	CPCREG8	CPEXADD	CPEXBLOK	CPEXREGS	CPEXR11	CPEXSIZE	CPINITD
	CPLOKFL	CPMFAWIA	CPRUN	CPSPMODE	CPSTATUS	CPSTAT2	CPSUPER	CPSYSLK	CPUID	CPWAIT
	CRESIMD	CSW	CUE	C0	C1	C2	C6	C8	DMKCLKAP	DMKCLKCC
	DMKCLKSC	DMKCVTAB	DMKDMPC2	DMKDSPB	DMKDSPCH	DMKDSPE	DMKDSPRU	DMKDSPWI	DMKFREEP	DMKHPTDI
	DMKLOKDF	DMKLOKIO	DMKLOKSY	DMKLOKTR	DMKMCTFS	DMKMCTMA	DMKMCTPR	DMKMHCIN	DMKMPOEX	DMKPMAEW
	DMKPMASS	DMKPMASW	DMKPSAER	DMKPSANX	DMKQCOCL	DMKRNHND	DMKSCHTQ	DMKSCNEP	DMKSCNRD	DMKSTKIO
	DMKSTKMP	DMKTMRVT	DMKVFRSV	DPAPUOP	DPSTAT	EMSPCLKC	EMSPEND	EMSPEXT	EMSPQUI	EMSPSHD
	EMSPSYNC	EMSPXEX	EMSREC	EMSRSHD	EMSRXEX	EXNPSW	EXOPSW	EXTMASK	F0	F1
	F2	F255	F256	F3	F4	F5	F6	F8	F9	INMSFBLK
	INTEX	INTEXF	IOMASK	IPUADDRX	LOCK	LOCKSAV	LPUADDR	MFAMASK	MP	MPUOPER
	MSSFINTR	NICBLOK	NICNAME	NICSIZE	NICUSER	NOCODE	NORET	PMAGUEST	PMAMODE	PMASTAT
	PREFIXA	PREFIXB	PROCIO	PROCIPL	PSA	QUANTUM	QUANTUMR	RCHBLOK	RCHPROC	RDEVBASE
	RDEVBLOK	RDEVFLAG	RDEVHIO	RDEVNICL	RDEVPROC	RDEVPS	RDEVSTA3	RDEVTYPC	RDEVUSER	RESET
	RUNCR0	RUNPSW	RUNUSER	R0	R1	R10	R11	R12	R13	R14
	R15	R2	R3	R4	R5	R6	R7	R8	R9	SAS
	SENSE	SIGDISP	SIGEMS	SIGEXT	SIGQUI	SIGRES	SIGSAVE	SIGSENSE	SIGWAKE	SIGXC
	SM	SYNCMASK	TEMPR11	TEMPR13	TEMPR2	TEMPSAVE	TIMEDISP	TIMER	TRACCURR	TRACEND
	TRACFLG1	TRACFLG2	TRACPROC	TRACSTRT	TRACSVCR	TRAC01	TRAC13	TRCEXT	TRCSIGP	TRQBBPNT
	TRQBFPNT	TRQBLOK	TRQBQUE	TRQBUSER	TRQBVAL	VACOVFR	VCONCTL	VCONOPT	VCONSCRN	VCON3270
	VECF	VMBLOK	WAIT	WAITEND	XCAPR	XCDISP	XCPEND	XCRES	XCWAK	XTNDCR2
	Y0	Y2	Y4	Y6	ZEROES					
DMKFCB	LINE	VERLEN								
DMKFMT	ATTN	BLANK	BUSY	CAW	CC	CD	CE	COUNT	CSW	CUE
	C2	DE	D1	D2	D28	D3	D4	D5	D8	EQCHK
	FLAG	F0	INTREQ	IONPSW	IOOPSW	L1	L1024	L16	L20	L3
	L4	L400	L4096	L5	L6	L7	L8	L80	MAXLEN	NOPRINT
	PROPSW	PSA	RANGE	READ	REGSAV	RESET	R0	R1	R10	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	SCAN	SENSE	SILI	SKIP	SM	UC	UE	ZERO
DMKFPS	AFRET	APSTAT2	AVMREAL	CPCREG0	CPPTLBR	CPSPMODE	CPSTAT2	CPSTAT4	CP370EON	C0
	C1	C6	DMKFRET	DMKPSAFP	DMKPSASP	DMKSCHQ1	DMKSCHQ2	DMKSCNVU	DMKTMRPT	EXTMODE

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	F1	F15	F240	F4	F4095	F4096	F6	F60	INTPRL	IPUADDR
	LAP370E	LOCK	L2	MATCH	MICBLOK	MICEVMA2	MICPEND	MICSPT	MICVIP	NOPTLB
	OLDKEYOP	PAGE4K	PREFIXB	PROBMODE	PROPSW	PSA	PURGESTO	QUANTUM	QUANTUMR	RUNCR0
	RUNCR1	RUNPSW	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SAS	SPMPFX
	STOACTV	STOBLOK	STOFLAG	STOLAST	STONEXT	STOSEGVR	STOSHCR1	STOUSPT	STOUSPTL	STOVCR1
	STOVPPG	STOVPSG	STO6CPG	STO6CSG	TEMPR12	TEMPR14	TEMPSAVE	TIMEDISP	TIMER	TRANMODE
	TRQBLOK	TRQBQUE	VECF	VECUSER	VMBLOK	VMSIZE	XKEYMODE	XPAGNUM	XRIGHT16	ZEROES
DMKFRE	AEXTSP	AFREE	AFREEP	AFRET	ALOKFR	ALOKRM	ALOKSP	ALOKSY	APSTAT1	APUOPER
	ASYSVM	BALRSAVE	BPFRESV	BYTBL	CODE	CONTROL	COUNT	CPEXADD	CPEXBLOK	CPEXBPNT
	CPEXFPNT	CPEXPROC	CPEXR0	CPEXR11	CPEXSIZE	CPFRELK	CPSPMODE	CPSTAT2	C0	C2
	C6	DMKCPE	DMKDMPC2	DMKDSPNP	DMKDSPRU	DMKFRTSN	DMKFRTT	DMKLOKDF	DMKLOKIO	DMKLOKSW
	DMKLOKSY	DMKPTRCO	DMKPTRFR	DMKPXA	DMKPXB	DMKSTKLF	DMKSYSCS	DSPRQ	ECPSOP	ECPSUBTB
	ENDSIZES	EXCLOCK	EXSUBTOP	FFS	FRADDR	FRECALLR	FREDPACA	FREE	FREEH	FREEP
	FREER0	FREER1	FREER12	FREER14	FREER15	FREESAVE	FREEWORK	FREEXT	FRELSNSS	FRELSTTS
	FRENAME	FRLKPROC	FROSIZE	FRPALLOC	FRPMCALR	FRPMEXT	FRPMNAME	FRPMSIZE	FRPMTAG	FRPSIZE
	FRSIZE	FRTAG	F0	F4096	HBOUND	HMAXSIZE	HWORD	INCRBY1	IPUADDRX	LASTUSER
	LOCK	LOCKSAV	LOKREQ	LPUADDR	LPUADDRX	LSPMSK	LSPSIZE	LSPSIZE$	MAXCSIZE	MAXHSIZ
	MAXSIZE	MAXSPSIZ	MINLEAVE	NUMCAPLS	NUMSTEAL	PMAAVAIL	PMASTAT	PNTR	PREFIXA	PREFIXB
	PRIMEHDR	PRIMEHI	PRIMELO	PROCIO	PSA	PSAEXT	PSAEXTX	PXA	RQLOCK	R0
	R1	R10	R11	R12	R13	R14	R15	R2	R3	R4
	R5	R6	R7	R8	R9	SEQMSK	SIGDISP	SIGEMS	SIGWAKE	SIGXC
	SIGXEX	SIZE	SPBOUND	SPNTR	SPTIME	SUBSIZES	SUBSTLA	SUBSTLS	SUBTABMX	SUBTOP
	SVMUNLOK	S12	S15	S18	S21	S24	S27	S3	S30	S33
	S6	S9	TABWDTH$	TEMPR6	TEMPSAVE	TIMEDISP	TIMEPOP	TRACCURR	TRACEND	TRACFLG1
	TRACPROC	TRACSTRT	TRACSVCR	TRAC67	TRCFREE	TRCFREEP	UPPINCR	USPMSK	USPSIZE	USPSIZE$
	VMBLOK	XCMASK	XTNDCR2	XTNDLOCK						
DMKFRT	ACORETBL	ALOKFR	ALOKSP	ALOKSY	APSTAT1	APUOPER	BALRSAVE	CODE	CORFPNT	CORPGPNT
	CORTABLE	COUNT	DMKCPE	DMKDSPNP	DMKFREE	DMKFREHI	DMKFREHP	DMKFRELN	DMKFRELO	DMKFRELP
	DMKFRELS	DMKFREMT	DMKFRENP	DMKFRESP	DMKFRESV	DMKPTTFT	DMKPXA	DMKPXB	DMKQCNFT	DMKQCOFT
	DMKSYSCS	DMKSYSRM	DMKVCNFT	D4	EXCLOCK	EXSUBTOP	FRADDR	FRBYTES	FREER0	FREER13
	FREER14	FREER15	FREESAVE	FREEWORK	FREEXT	FROSIZE	FRPALLOC	FRPCACHL	FRPMEXT	FRPMSIZE
	FRPMTAG	FRPRETRN	FRPSIZE	FRSIZE	FRTAG	F1	F4096	LOCK	LOCKSAV	LPUADDR
	LSPSIZE	LSPSIZE$	MAXHSIZ	MPFEAT	PAGBMP	PREFIXA	PREFIXB	PRIMEHDR	PRIMEHI	PRIMELO
	PSA	PSAEXT	PXA	R0	R1	R10	R11	R12	R13	R14
	R15	R2	R3	R4	R5	R6	R7	R8	R9	SPBOUND
	STAKSIZE	SUBHEAD	SUBTABLE	SUBTABMX	SUBTOP	TABWDTH$	TRACCURR	TRACEND	TRACFLG1	TRACPROC
	TRACSTRT	TRACSVCR	TRAC67	TRCFRET	TRCFRETP	UPPINCR	USPSIZE	USPSIZE$	VMBLOK	XPAGNUM
	XTNDLOCK									
DMKGIO	ADSPCH	AFREEP	AFRET	APTRAN	BRING	CCC	CLASDASD	CLASFBA	CLASTAPE	CLASURI
	CLASURO	CPEXADD	CPEXBLOK	CPEXFPNT	CPEXMISC	CPEXR0	CPEXR1	CPEXSIZE	CSW	C1
	DEFER	DMKCCWTR	DMKDSPCH	DMKFREEP	DMKFRET	DMKIOSQV	DMKPSAPO	DMKPTRAN	DMKSCNVD	DMKSCNVU
	DMKUNTFR	DMKUNTRN	F0	IFCC	IL	IOBCAW	IOBCC3	IOBCLN	IOBCSW	IOBFATAL
	IOBFLAG	IOBHVC	IOBIOER	IOBIRA	IOBLINK	IOBLOK	IOBMISC	IOBMISC2	IOBSIZE	IOBSPEC2
	IOBSTAT	IOERBLOK	IOERCSW	IOERDATA	IOEREXT	IOERPNT	IOERSIZE	LOCK	PSA	RDEVBLOK
	RDEVCUP	RDEVSTA5	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS
	SAVER2	TYP3340	UE	VMBLOK						
DMKGRA	AFREE	AFRET	APTRAN	ASYSVM	BRING	CONADDR	CONCNT	CONDATA	CONTASK	CONTSIZE
	CONTSKSZ	C1	DEFER	DMKFREE	DMKFRERC	DMKFRET	DMKPTRAN	DMKTBMAO	DMKTBMTO	DMKTBMXO
	DMKTBMZO	F0	F255	LOCK	PSA	R0	R1	R10	R11	R12

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SAVER6	SYSTEM	VMBLOK	XRIGHT16			
DMKGRC	AFREE	AFRET	APTRAN	APTRLK	ASYSVM	BALRSAVE	BALR10	BRING	CLASTERM	CONCNT
	CONDATA	CONPARM	CONPARM2	CONTASK	CONTINUE	CONTSIZE	CONTSKSZ	CONUSER	CONWORK1	CONWORK2
	C1	DEFER	DMKFREE	DMKFRET	DMKGRTB	DMKGRT12	DMKLOCKD	DMKLOCKQ	DMKPGUVG	DMKPGUVR
	DMKPTRAN	DMKPTRUL	FFS	F0	F1	F10	F2	F4096	F8	IOBCSW
	IOBLOK	IOERETN	LOCK	NICADFF	NICADVF	NICAWSF	NICBLOK	NICEWO	NICHT	NICPT
	NICQREP	NICRFLG	NICWTH	PSA	RDEVADFF	RDEVADVF	RDEVBLOK	RDEVECOL	RDEVEHLT	RDEVEWO
	RDEVFTR2	RDEVHT	RDEVPSS	RDEVPT	RDEVQREP	RDEVSNA	RDEVTYPC	RDEVWTH	RDEV12B	RDEV14AD
	RDEV14B	REW	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS
	SAVEWRK1	SAVEWRK2	SAVEWRK3	SYSTEM	TEMPR0	TEMPR1	TEMPR14	TEMPR15	TEMPR2	TEMPR3
	TEMPR4	TEMPR9	TYPBSC	VMBLOK	VMGENIO	XRIGHT16				
DMKGRD	ADSPCH	AFRET	ALARM	CC	CD	CONCCW1	CONCCW2	CONCCW3	CONCCW4	CONCNT
	CONDATA	CONDCNCL	CONDLN	CONDRFMT	CONDWC	CONEWRT	CONFLAGS	CONFSOP	CONFSS	CONLNCNT
	CONLNRES	CONOUTPT	CONPARM	CONPARM2	CONPNT	CONSTAT	CONSYNC	CONTASK	CONTGMXB	CONTSIZE
	CONWCC	CONWORK2	CPUVERSN	CRTALRM	CRTDIAG	CRTFMT	CRTFSII	CRTFSSA	CRTSIO	CRTUSEWA
	CRTWNG	DMKCFMAT	DMKDSPCH	DMKFRET	DMKGRCUP	DMKGRCWC	DMKGRECL	DMKGRESM	DMKGRFBR	DMKGRFCF
	DMKGRFIN	DMKGRGBR	DMKGRHIN	DMKGRISB	DMKGRTB	DMKGRT12	DMKHPSQR	DMKIOSQR	D66LNCNT	FFS
	FTRNODRP	FTR3270E	F2	F9	GRCALRM	GRCCLRIN	GRCCPRD	GRCECOL	GRCEHLT	GRCINH
	GRCMXDLN	GRCSFIN	GRCSFSTA	GRCVMRD	GREBI	GREBT	GRECR	GREC1	GREC2	GREFI
	GREFL	GREFSMOR	GREFTFMT	GREGC	GREGCMOR	GREGGCLR	GREGGRUN	GREGL	GREGN	GREGTHLD
	GREIC	GREIS	GRERT	GRESR	GREST	GRESV	GREWS	GRTBLOK	INHIBIT	IOBCAW
	IOBLINK	IOBLOK	IOBMISC2	IOBSIZE	IOBSPEC	IOBUNSL	IOBUSER	LOCK	LOGDROP	LOGHOLD
	NOTRESP	OPERATOR	PRIORITY	PSA	RB	RDEVACTV	RDEVADFF	RDEVADVF	RDEVAINH	RDEVAIOB
	RDEVBLOK	RDEVCON	RDEVCORD	RDEVCTL	RDEVECOL	RDEVEHLT	RDEVEWO	RDEVFLAG	RDEVFTR	RDEVGRF1
	RDEVHOLD	RDEVHSTR	RDEVHT	RDEVIOAT	RDEVIOCP	RDEVIODE	RDEVNOW	RDEVPS	RDEVREAD	RDEVRUN
	RDEVSTA3	RDEVTFLG	RDEVTRQ	RDEVTYPE	RDEVUSER	RDEVWIOB	RDEVWTH	RDEV14B	RDEV741D	RM
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SBA	SCRCMDR	SECUSER	SELECT
	SELRB	SELRM	SELWRT	SF	SILI	TRQBCHIO	TRQBCLIN	TRQBCPQ	TRQBCRT	TRQBFLAG
	TRQBFLG2	TRQBLINE	TRQBLOK	TRQINTEG	TYP3066	TYP3277	TYP3278	VCONANF	VCONANF2	VCONCBRK
	VCONCTL	VCONFLG2	VCONOPT	VCONREXW	VCONSCRN	VCON3270	VMBLOK	VMGENIO	WCCALRM	WCC3
	WR	XRIGHT16								
DMKGRE	ACORETBL	ADSPCH	CC	CONADDR	CONCCW1	CONCCW2	CONCCW4	CONCNT	CONCOMND	CONDATA
	CONDWC	CONEWA	CONEWRT	CONFLAG	CONFLG2	CONFSS	CONNCB	CONOUTPT	CONPARM	CONPNT
	CONPPA1	CONSTAT	CONTASK	CONTGMXB	CONTSIZE	CONWRT	CORCP	CORFLAG	CORFPNT	CORTABLE
	CRTALRM	CRTDIAG	CRTFSII	CRTFSSA	CRTPPA1	CRTUSEWA	CRTWNG	DMKDSPCH	DMKERMSG	DMKGRDBR
	DMKGRFBR	DMKGRFCF	DMKGRGBR	DMKGRISB	DMKPTRFR	EXCOLOR	EXHILIGH	EXPSS	F1	F4096
	F5	GREBI	GREBT	GRECR	GREC1	GREDATNR	GREDBCCW	GREDCIOS	GREDCK1D	GREDCVMP
	GREDNCTL	GREDNSEL	GREDRIOB	GREDRSCP	GREDWBOK	GREFC	GREFI	GREFL	GREFR	GREFS
	GREFT	GREFUTCP	GREGB77C	GREGC	GREGL	GREGN	GREIC	GREIS	GRERT	GRESR
	GREST	GRESV	GREWS	IC	IDA	IDAHEAD	IDAHSTRT	IOBCSW	IOBLOK	LOCK
	PSA	RB	RDEVADVF	RDEVBLOK	RDEVCON	RDEVCORD	RDEVCTL	RDEVECOL	RDEVEHLT	RDEVHSTR
	RDEVIOAT	RDEVIOCP	RDEVIODE	RDEVNOW	RDEVPS	RDEVPSS	RDEVSTA3	RDEVTFLG	RDEVWSF	RDEV741D
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SBA	SELRBP	SELRMP	SILI
	TRQBCPQ	TRQBCRT	TRQBFLAG	TRQBFLG2	TRQBLINE	TRQBLOK	VCONANF	VCONANF2	VCONBRK	VCONCBRK
	VCONCPRD	VCONCTL	VCONOBRK	VCONOPT	VCONRD	VCONSCOL	VCONSCRN	VCONSHI	VCONSPSS	VCONSRM
	VCONWA	VCON3270	VMBLOK	WCC0	WSF	XRIGHT16				
DMKGRF	ADSPCH	AFREE	AFRET	APTRAN	ASYSOP	ASYSVM	ATTN	BRING	BUFAPL	BUFFER

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	BUFINLTH	BUFNORM	BUFSIZE	CC	CCC	CD	CDC	CE	CHC	CONCCW1
	CONCCW2	CONCCW3	CONCCW4	CONDATA	CONFLAG2	CONFSOP	CONFSS	CONPARM	CONPNT	CONRMOD
	CONTASK	CONTGMXB	CONTSIZE	CPID	CPUVERSN	CRTAIO	CRTAPL	CRTCARD	CRTFMT	CRTFSII
	CRTFSSA	CRTPPA1	CRTUSEWA	C1	DE	DEFER	DMKBLDVM	DMKCFMAT	DMKCFMBK	DMKCVTAB
	DMKDSPCH	DMKFREE	DMKFRET	DMKGRCQY	DMKGRDBR	DMKGRDCC	DMKGRECL	DMKGREFD	DMKGRGBK	DMKGRGBR
	DMKGRGCL	DMKGRGCR	DMKGRGER	DMKGRGPF	DMKGRGTI	DMKGRGTR	DMKGRHIN	DMKGRISB	DMKGRTAC	DMKGRTAI
	DMKGRTFD	DMKGRTFM	DMKGRTP6	DMKHPTDI	DMKIOERR	DMKIOEST	DMKMSWR	DMKOPEEM	DMKPTRAN	DMKQCOCL
	DMKQCOET	DMKQCPTO	DMKRIOCN	DMKSCHST	DMKSCNRU	DMKSYSNM	DMKTBMAI	DMKTBMTI	DMKTBMXI	DMKTBMZI
	FFS	FTRNODRP	F0	F1	F4	F4096	F8	GRCMXDLN	GRCRUN	GREBI
	GREBT	GRECR	GREC1	GREC2	GREDATNR	GREDBCCW	GREDCIOS	GREDCK1D	GREDCVMP	GREDFIOB
	GREDIIOB	GREDNCTL	GREDNSEL	GREDRIOB	GREDRSCP	GREDWBOK	GREFI	GREFL	GREFR	GREFS
	GREFT	GREGC	GREGGCLT	GREGL	GREGN	GREGSBUF	GREIC	GREIS	GRERT	GRESR
	GREST	GRESV	IDA	IFCC	INTREQ	IOBCAW	IOBCOPY	IOBCSW	IOBFATAL	IOBFCNS
	IOBIOER	IOBLINK	IOBLOK	IOBMISC	IOBMISC2	IOBRADD	IOBRCNT	IOBSENS	IOBSIZE	IOBSPEC
	IOBSPEC5	IOBSTAT	IOBUNSL	IOBUSER	IOERBLOK	IOERCSW	IOERDATA	IOERFLG3	IOERNUM	IOERREAD
	IOERSIZE	LOCK	LOGDROP	LOGHOLD	PREFIXA	PRGC	PRTC	PSA	RDEVACTV	RDEVADVF
	RDEVAIOB	RDEVAIRA	RDEVAPLP	RDEVATSW	RDEVBLOK	RDEVBUZY	RDEVCON	RDEVCORD	RDEVCPNA	RDEVCTL
	RDEVDISB	RDEVDROP	RDEVENAB	RDEVERR	RDEVFCNS	RDEVFLAG	RDEVFORC	RDEVFTR	RDEVGRF1	RDEVHIO
	RDEVHOLD	RDEVHSTR	RDEVHT	RDEVIOAT	RDEVIOCP	RDEVIODE	RDEVIOER	RDEVLGDT	RDEVLOG	RDEVLTRM
	RDEVMDL	RDEVMORE	RDEVNOW	RDEVPRFG	RDEVPS	RDEVQRY	RDEVREAD	RDEVRUN	RDEVSCHD	RDEVSTA2
	RDEVSTA3	RDEVSTA4	RDEVSTA5	RDEVSTA6	RDEVSTA7	RDEVTEXT	RDEVTFLG	RDEVTMCD	RDEVTRQ	RDEVTYPE
	RDEVUSER	RDEVWIOB	RDEVWSF	RDEVWTH	RDEV741D	RDPTQRYG	R0	R1	R10	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	SENSEID	SILI	STYPRDPT	SYSTEM	TEMPR0	TRQBCHIO	TRQBCPQ	TRQBCRT
	TRQBFLAG	TRQBFLG2	TRQBFLG3	TRQBIRA	TRQBLINA	TRQBLINE	TRQBLOGP	TRQBLOK	TRQBUSER	TRQBVAL
	TRQINTEG	TYP3066	TYP3277	TYP3278	TYP3284	UC	UE	VCONBRK	VCONCBRK	VCONCPRD
	VCONCTL	VCONOBRK	VCONOPT	VCONSCRN	VCON3270	VMBLOK	WCC3	WSF	XTNDLOCK	
DMKGRG	ADSPCH	AFREE	AFREEP	AFRET	AQCNWT	ARIOCH	ARIODV	BLANKS	BUFCNT	BUFFER
	BUFINLTH	BUFSIZE	CC	CLASGRAF	CONADDR	CONCCW1	CONCCW2	CONCNT	CONDATA	CONFSOP
	CONFSS	CONPARM	CONPARM2	CONRETN	CONTASK	CONTGMXB	CONTINUE	CONTSIZE	CPEXADD	CPEXBLOK
	CPEXR0	CPEXSIZE	CPID	CPINIT	CPSTAT5	CRTAIO	CRTALRM	CRTCARD	CRTDIAG	CRTFMT
	CRTFSII	CRTFSSA	CRTUSEWA	CRTWNG	DMKCFMAT	DMKCFMBK	DMKCFMEN	DMKCNTED	DMKCVTBD	DMKCVTHB
	DMKDSPCH	DMKFREE	DMKFREEP	DMKFRERC	DMKFRET	DMKGRCUP	DMKGRDBR	DMKGRDCC	DMKGRDSR	DMKGRECL
	DMKGRFBR	DMKGRFCF	DMKGRHIN	DMKGRISB	DMKGRTAB	DMKGRTPF	DMKGRT12	DMKQCNWT	DMKRETGT	DMKRETPT
	DMKSCNRU	DMKSTKCP	DMKTBLUP	EDIT	EWA	F0	F1	F2	F3	F4095
	GRCALRM	GRCCLRIN	GRCCLROT	GRCECOL	GRCEHLT	GRCHOLD	GRCMORE	GRCMXLEN	GRCNAC	GRCRUN
	GRCSFIN	GRCSFSTA	GREBI	GREBT	GRECR	GREC1	GREDATNR	GREDBCCW	GREDCIOS	GREDCK1D
	GREDCVMP	GREDNCTL	GREDNSEL	GREDRD77	GREDRIOB	GREDRSCP	GREDWBOK	GREFC	GREFCCLR	GREFI
	GREFLTST	GREFR	GREFSINT	GREFSMOR	GREFSM77	GREFSREJ	GREFSR77	GREFSWNG	GREFSW77	GREFT
	GREGC	GREIC	GREIS	GRERT	GRESR	GRESV	GREWS	GRTABLEN	HILIGHT	IC
	INHIBIT	INTREQ	IOBCAW	IOBCOPY	IOBCSW	IOBLINK	IOBLOK	IOBMISC	IOBMISC2	IOBSENS
	IOBSPEC	LOCK	LOGDROP	LOGHOLD	MNCLRESP	MNCOERD	NOMC	NORET	NOTIME	NOTRESP
	PFDATA	PFDCMD	PFDCPAD4	PFDCPYSP	PFKADDR	PFKFLAG	PFKIMM	PFKLNG	PFKRET	PFKTABLE
	PSA	RCUBLOK	RCUDVTBL	RCUPRIME	RCUSUB	RCUTYPE	RDEVACTV	RDEVADVF	RDEVAINH	RDEVAIRA
	RDEVAPLP	RDEVBLOK	RDEVCON	RDEVCORD	RDEVCPNA	RDEVCTL	RDEVCUA	RDEVDED	RDEVDIIP	RDEVDISA
	RDEVECOL	RDEVEHLT	RDEVENAB	RDEVFLAG	RDEVGRF1	RDEVHOLD	RDEVHSTR	RDEVHT	RDEVIOAT	RDEVIOCP
	RDEVIODE	RDEVMDL	RDEVMORE	RDEVNAME	RDEVNOW	RDEVPS	RDEVREAD	RDEVRUN	RDEVSTAT	RDEVSTA3
	RDEVTFLG	RDEVTMCD	RDEVTRQ	RDEVTYPC	RDEVTYPE	RDEVWIOB	RDEVWSF	RDEVWTH	RDIDX	RDPTQRYG
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVER0	SBA	SILI
	STYPRDPT	TEMPR0	TEMPR3	TEMPSAVE	TREXFLAG	TREXMOR	TREXT	TRQBCHIO	TRQBCLIN	TRQBCRT
	TRQBDEV	TRQBFLAG	TRQBFLG2	TRQBLINE	TRQBLOK	TYP3277	TYP3278	TYP3284	UCASE	VCONANF
	VCONBRK	VCONCTL	VCONOBRK	VCONOPT	VCONRBUF	VCONRCNT	VCONRDSZ	VCON3270	VMBLOK	VMGENIO
	XRIGHT16									

MODULE EXTERNAL REFERENCES (LABELS AND MODULES)

MODULE										
DMKGRH	ALARM	BALRSAVE	BUFINLTH	CC	CONCCW1	CONCCW2	CONCCW4	CONCNT	CONDATA	CONDLN
	CONDWC	CONLNCNT	CONPARM	CONTASK	CRTALRM	CRTSIO	DMKGRTFO	IOBCAW	IOBLOK	IOBMISC
	LOCK	PSA	RDEVBLOK	RDEVCTL	RDEVHOLD	RDEVMORE	RDEVRUN	RDEVTFLG	R0	R1
	R10	R12	R14	R15	R2	R3	R4	R5	R6	R8
	SILI	TRQBFLAG	TRQBLINE	TRQBLOK						
DMKGRI	AFREE	AFREEP	AFRET	ALOKSP	APSTAT1	APUOPER	BLANK	BUFSIZE	CC0	CC1
	CC2	CC3	CONADDR	CONCCW1	CONCCW2	CONCNT	CONFLG2	CONLED	CONPARM	CONPNT
	CONRESP	CONRETN	CONSTAT	CONTASK	CONTGMXB	CONTINUE	CONTSIZE	CONTSKSZ	CONUSER	CRTAPL
	CRTEXT	CRTEXTSZ	CRTSIO	DMKFREE	DMKFREEP	DMKFRET	DMKGRAOT	DMKGRDBR	DMKGRFIN	DMKLOKIO
	DMKLOKSW	DMKQCOCL	DMKQCOET	DMKSCHRT	DMKSCNRD	DMKTBLGR	DMKTBLGT	DMKTBLRG	FAKEATT	F0
	F256	F4	F4096	F8	GRACDAPL	GRACDTXT	GRADEV77	GRADEV78	GRAFESC	GREDIIOB
	GREDRIOB	IC	IOBCAW	IOBERP	IOBFLAG	IOBIOER	IOBIRA	IOBLINK	IOBLOK	IOBMISC
	IOBSIZE	IOBUSER	IOERBLOK	IOEREXT	IOERSIZE	LOCK	LOCKSAV	LPUADDR	PRIORITY	PSA
	RDEVAIRA	RDEVAPLP	RDEVBLOK	RDEVCON	RDEVGRF1	RDEVHSTR	RDEVIOAT	RDEVIOCP	RDEVIODE	RDEVNOW
	RDEVTEXT	RDEVTFLG	RDEVTMCD	RDEVTRQ	RDEVTYPE	RDEVUSER	RDEVWIOB	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R5	R6	R7
	R8	R9	SAVEAREA	SAVER2	TAB	TEMPR0	TIMEDISP	TRQBCPQ	TRQBCRT	TRQBDEV
	TRQBFLAG	TRQBLINA	TRQBLOK	TRQBSIZE	TYP3277	TYP3278	VMBLOK	XRIGHT16		
DMKGRT	ADSPCH	AFREE	AFRET	APTRAN	ASYSVM	BALRSAVE	BALR0	BALR6	BOXBLOK	BOXINLNS
	BOXINWTH	BOXLINES	BOXLOGO	BOXWIDTH	BRING	BUFCNT	BUFFER	BUFINLTH	BUFNXT	BUFSIZE
	CLASTERM	CONADDR	CONCNT	CONDATA	CONPARM	CONTASK	CONTSIZE	C1	DEFER	DMKBOXMS
	DMKBOXNS	DMKCNTED	DMKDSPCH	DMKFREE	DMKFRET	DMKGRFMT	DMKPTRAN	DMKRGBFM	DMKSYSJR	EDIT
	FFS	F0	F1	F2	F3	NICADFF	NICBLOK	NICHT	NICWTH	PFDATA
	PFDVAL	PFKADDR	PFKLNG	PFKTABLE	PFTAB	PFTABS	PSA	PWDCHAIN	PWDIBLOK	PWDITRQ
	PWDLGCNT	RDEVADFF	RDEVBLOK	RDEVHT	RDEVTYPC	RDEVWTH	RDEV14B	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER0	SAVER8	SAVER9	SYSTEM	TEMPR0
	TEMPR1	TEMPR6	TRQBLINA	TRQBLOK	TRQBSIZE	TYPBSC	UCASE	VMBLOK		
DMKHPS	ADSPCH	AFREE	AFRET	AP	APSTAT1	APTRAN	APTRLK	APUOPER	ASYSVM	BRING
	CC	CCC	CD	CE	CLASGRAF	CMDREJ	CPEXADD	CPEXBLOK	CPEXFPNT	CPEXR0
	CPEXSIZE	C1	DE	DEFER	DMKDSPCH	DMKFREE	DMKFRET	DMKHPTRE	DMKHPUSR	DMKLOKSW
	DMKPTRAN	DMKSTKCP	DMKTRDSI	DMKVBMIN	FFS	F0	F1	F15	F4	F4096
	IDA	IL	INTREQ	IOBCAW	IOBCP	IOBCSW	IOBFLAG	IOBFPNT	IOBIOER	IOBLINK
	IOBLOK	IOBRADD	IOBRCNT	IOBSIZE	IOBSPEC	IOBSTAT	IOBVADD	IOERBLOK	IOERCSW	IOERDATA
	IOERETN	IOERLEN	IOERSIZE	LOCK	MP	PRGC	PSA	RDEVADD	RDEVADFF	RDEVAIOB
	RDEVBLOK	RDEVBUZY	RDEVCTRS	RDEVCUA	RDEVDIIP	RDEVENAB	RDEVEWO	RDEVFIOB	RDEVFLAG	RDEVHT
	RDEVLIOB	RDEVLLEN	RDEVMDL	RDEVNAME	RDEVPS	RDEVPSUP	RDEVSIZE	RDEVSTA3	RDEVSTA4	RDEVTYPC
	RDEVTYPE	RDEVUSER	RDEVWTH	RDEV12B	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	SAVER11	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK8	SAVEWRK9	SILI	SKIP
	SPECIALV	TIMEDISP	TYP3277	TYP3278	TYP3286	UC	UE	VMBLOK	XPAGNUM	
DMKHPT	ACORETBL	ADSPCH	AFREE	AFREEP	AFRET	AP	APTRAN	APTRLK	ASYSVM	BRING
	BUFSIZE	CCC	CE	CORCP	CORFLAG	CORTABLE	CPEXADD	CPEXBLOK	CPEXR0	CPEXR11
	CPEXSIZE	CRTEXT	C1	DE	DEFER	DMKDSPCH	DMKFREE	DMKFREEP	DMKFRET	DMKGRCSV
	DMKGRFIN	DMKPTRAN	DMKPTRLK	DMKPTRUL	DMKSCHDL	DMKSCHRT	DMKSTKCP	DMKSTKIO	DMKVIOIN	F0
	F4	F4096	F7	INTREQ	IOBCAW	IOBCP	IOBCSW	IOBFLAG	IOBFPNT	IOBIOER
	IOBIRA	IOBLINK	IOBLOK	IOBRADD	IOBRCNT	IOBSIZE	IOBSPEC	IOBUNSL	IOBUSER	IOBVADD
	IOERBLOK	IOERCSW	IOERDATA	IOEREXT	IOERLEN	IOERSIZE	LOCK	MP	PSA	RDEVADD
	RDEVAIOB	RDEVAIRA	RDEVATT	RDEVBLOK	RDEVBUZY	RDEVDED	RDEVFIOB	RDEVQREP	RDEVSIZE	RDEVSTAT
	RDEVSTA4	RDEVTFLG	RDEVTRQ	RDEVTYPE	RDEVUSER	R0	R1	R10	R11	R12

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SAVER11	SAVER5	SAVER8	SAVEWRK2	SAVEWRK3	SAVEWRK4	TRQBLINA
	TRQBLOK	TRQBSIZE	TYP3286	UC	VMBLOK					
DMKHPU	ACORETBL	ADSPCH	AFREE	AFREEP	AFRET	APTRAN	APTRLK	ASYSVM	ATTN	BRING
	CD	CORCP	CORFLAG	CORTABLE	CPEXADD	CPEXBLOK	CPEXR0	CPEXR11	CPEXSIZE	C1
	DE	DEFER	DMKDSPCH	DMKFREE	DMKFREEP	DMKFRET	DMKGRFIN	DMKPSAFC	DMKPSASC	DMKPTRAN
	DMKPTRLK	DMKPTRUL	DMKSTKCP	DMKSTKIO	DMKVIOIN	F0	F1	F4096	F7	IDA
	IL	IOBCP	IOBCSW	IOBFLAG	IOBFPNT	IOBIRA	IOBLINK	IOBLOK	IOBRADD	IOBSIZE
	IOBSPEC	IOBUNSL	IOBUSER	IOBVADD	IOERBLOK	LOCK	PSA	RDEVADD	RDEVAIOB	RDEVATT
	RDEVBLOK	RDEVBUZY	RDEVDED	RDEVFIOB	RDEVSCHD	RDEVSTAT	RDEVSTA4	RDEVUSER	R0	R1
	R10	R11	R12	R13	R14	R15	R2	R3	R4	R5
	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER1	SAVER10	SAVER13	SAVEWRK1
	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	SILI	SKIP
	VMBLOK	VMLDCTRS	VMPSW	VMPXINT	VMSEG	VMVSPACE				
DMKHVC	ADSPCH	AFREE	AFREEP	AFRET	AMCHAREA	APTRAN	APTRLK	ASYSVM	BLANKS	BRING
	BUFINLTH	BUFSIZE	CCC	CDC	CE	CHC	CLASTAPE	CPUID	C1	DE
	DEFER	DMKCCWTR	DMKCFGCL	DMKCFMBK	DMKCFMEN	DMKCVTDT	DMKDGDDK	DMKDSPB	DMKDSPCH	DMKFREE
	DMKFREEP	DMKFRET	DMKGIOEX	DMKHVDAL	DMKHVEAL	DMKHVFAL	DMKMHVSM	DMKPGSSS	DMKPRGSM	DMKPSAFP
	DMKPSASP	DMKPTRAN	DMKPTRUL	DMKRPDEP	DMKSCNVU	DMKSSVHV	DMKTMRPT	DMKUNTFR	DMKVMCFC	DMKVSIEX
	ECBLOK	EXTVPORL	FFS	F0	F1	F16	F2	F20	F256	F3
	F4	F4095	F4096	F5	F6	F60	F8	IFCC	IL	IOBCAW
	IOBCSW	IOBLOK	IOBMISC	IOBMISC2	IOBRADD	IOBSIZE	IOERETN	IOKEEP	LOCK	MCHAREA
	MCHMODEL	MOD3090	PARMSIZE	PCI	PRGC	PRTC	PSA	RCWCCW	RCWCTL	RCWGEN
	RCWHEAD	RCWPNT	RCWRCNT	RCWTASK	RCWVCAW	RCWVCNT	RDEVBLOK	RDEVSER	RDEVTYPC	RDEVTYPE
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	TEMPR6	TEMPR8	TYP3480	UC
	UE	VMBLOK	VMCF	VMCFWAIT	VMCLANY	VMCLASSA	VMCLASSB	VMCLASSC	VMCLASSD	VMCLASSE
	VMCLASSF	VMCLNULL	VMCMDLEV	VMCOMND	VMCONBUF	VMCONLEN	VMCPSV76	VMCPVIRT	VMCXSTAT	VMDVSTRT
	VMECEXT	VMESTAT	VMEXTCM	VMEXWAIT	VMFSTAT	VMGPRS	VMINST	VMIOWAIT	VMJSTAT	VMMCODE
	VMMLEVEL	VMMLVL2	VMMTEXT	VMNOCPRD	VMNPWOCL	VMOSTAT	VMPSTAT	VMPSW	VMRSTAT	VMSEG
	VMSLEEP	VMSPMFLG	VMSTOR	VMTDIAG8	VMTRCTL	VMTRPRV	VMTTIME	VMVIRCF	VMVPOREL	VMV370R
	XPAGNUM	ZEROES								
DMKHVD	ACCTACNO	ACCTBLOK	ACCTDIST	ACCTLENG	ACCTUSER	ACNTBLOK	ACNTCODE	ACNTDATA	ACNTNUM	ACNTSIZE
	ACNTUSER	ACORETBL	ADSPCH	AFREE	AFRET	APSTAT1	APTRAN	APTRLK	ASYSVM	AVMREAL
	BRING	CLASFBA	CLASGRAF	CLASTERM	CORDISA	CORFLAG	CORTABLE	CPSPMODE	CPSTAT2	CPUMCELL
	CPUVERSN	C1	DEFER	DMKACOQU	DMKCPEID	DMKCPEPP	DMKCPVAA	DMKCVTDB	DMKDRDER	DMKDRDMP
	DMKDRDSY	DMKDSPCH	DMKFREE	DMKFRET	DMKHPSDG	DMKIOEFM	DMKPGUPR	DMKPSAFP	DMKPSAPO	DMKPSASP
	DMKPTRAN	DMKRPAGT	DMKRPAPT	DMKSCNVS	DMKSNCP	DMKSNTQN	DMKSYSRM	DMKSYSTZ	DMKUDRDS	DMKUDRFU
	DMKUDRMD	DMKUDRRV	DMKUDUMN	F0	F1	F16	F2	F20	F24	F255
	F256	F3	F4	F4095	F4096	F8	IOERETN	IOKEEP	IPUADDR	LOCK
	MPUOPER	NICBLOK	NICGRAF	NICSIZE	NICTYPE	NPRCNT	NPRNAME	NPRPAGCT	NPRPNT	NPRSTART
	NPRTBL	NPRVOL	PAGCORE	PMAAVAIL	PMASTAT	PPMAP	PPMBSEPP	PPMHP0	PPMHP01	PPMHP02
	PPMHP025	PPMHP03	PPMHP032	PPMHP034	PPMHP036	PPMHP04	PPMHP042	PPMHP05	PPMHP2	PPMM21
	PPMPP	PPMSCH	PPMSEPP	PPMVDLE	PPMVMSP1	PPMVMSP2	PPMVMSP3	PPMVMSP4	PPMVMSP5	PREFIXA
	PREFIXB	PROCIO	PROCIPL	PSA	RDEVBLOK	RDEVCODE	RDEVFLAG	RDEVNICL	RDEVOWN	RDEVTYPC
	RDEVTYPE	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER0
	SAVEWRK2	SAVEWRK3	SAVEWRK4	SEGINV	SEGPAGE	SPMPFX	SWPCHG1	SWPCHG2	SWPCYL	SWPFLAG
	SWPRECMP	SWPREF1	SWPREF2	SYSTEM	TYPBSC	TYP2314	TYP3277	TYP3278	TYP3330	TYP3350
	TYP3375	TYP3380	VMACCOUN	VMACOUNT	VMAIP	VMAIP2	VMASCCPD	VMBLOK	VMESTAT	VMEXTCM
	VMEXWAIT	VMGPRS	VMIPLDEV	VMNOECPS	VMPA2APL	VMPSTAT	VMPSW	VMPSWDCT	VMQSTAT	VMRSTAT
	VMSEG	VMSIZE	VMSVSTAT	VMTERM	VMTRMID	VMUSER	VMV370R	XPAGNUM	XRIGHT24	X2048BND

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	ZEROES									
DMKHVE	ACORETBL	ADSPCH	AFREE	AFREEP	AFRET	APSTAT1	APTRAN	APTRLK	APUOPER	ASYSVM
	AVMREAL	BRING	BUFCNT	BUFFER	BUFIN	BUFINLTH	BUFNXT	BUFSIZE	CAW	CLASDASD
	CLASFBA	CLASGRAF	CLASSPEC	CLASTERM	CLASURI	CLASURO	CONTINUE	CORCFLCK	CORFLAG	CORFPNT
	CORTABLE	CORVM	CPEXADD	CPEXBLOK	CPEXMISC	CPEXREGS	CPEXR1	CPEXSIZE	C1	DEFER
	DMKDSPCH	DMKDSPNP	DMKFREE	DMKFREEP	DMKFRET	DMKIOEFR	DMKIOEHS	DMKLOCKD	DMKLOCKQ	DMKPSAFP
	DMKPSASP	DMKPTRAN	DMKPTRAQ	DMKPTRLK	DMKPTRUL	DMKRPAGT	DMKSCNRU	DMKSCNVD	DMKSCNVU	DMKSSSMQ
	DMKSYSCT	DMKSYSEA	DMKSYSER	DMKSYSTP	DMKVMDIA	DMKVSIEX	DMKXABDG	FFS	FTR3270E	FTR35MB
	F0	F1	F2	F255	F3	F4	F4096	F6	F60	F9
	IOERETN	IOKEEP	LOCK	L1	L2	L4	MSSPRES	NICADVF	NICBLOK	NICDTYPE
	NICDXSC	NICD3275	NICD3276	NICECOL	NICEHLT	NICLLEN	NICPSS	NICPT	NICQREP	NICRDED
	NICRFLG	NICSIZE	NICTMCD	NICWTH	NIC14AD	PROTBLOK	PROTBUFF	PROTERR	PROTPAGE	PROTPALT
	PROTTRAN	PSA	PSAMSS	RDCBLOK	RDCPAGAP	RDEVADVF	RDEVBLOK	RDEVCODE	RDEVECOL	RDEVEHLT
	RDEVFTR	RDEVLLEN	RDEVMDL	RDEVMD02	RDEVNICL	RDEVPSS	RDEVPT	RDEVQREP	RDEVRDC	RDEVTMCD
	RDEVTYPC	RDEVTYPE	RDEVWTH	RDEV14AD	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	SAVER0	SAVER5	SAVER6	SAVEWRK2	SAVEWRK3	SAVEWRK5	SYSTEM	TYPBSC
	TYP2305	TYP2319	TYP3210	TYP3277	TYP3278	TYP3284	TYP3330	TYP3340	TYP3350	TYP3375
	TYP3380	VCONCTL	VCONNICB	VCONRDEV	VMBLOK	XPAGNUM	ZEROES			
DMKHVF	ACIALTU	ACIAUSR	ACICODE	ACIFCN	ACINOAC	ACIPARMS	ACIRUSR	ACISIZE	ACIWUSR	ADSPCH
	AFREE	AFREEP	AFRET	ALTBLOK	ALTSIZE	ALTUSER	APTRAN	APTRLK	ASYSVM	BLANKS
	BRING	CLASFBA	CLASURI	CONTINUE	CPEXADD	CPEXBLOK	CPEXSIZE	C1	DEFER	DMKCVTBD
	DMKDSPCH	DMKFREE	DMKFREEP	DMKFRET	DMKPGUVG	DMKPGUVR	DMKPSAFP	DMKPSASP	DMKPTRAN	DMKPTRUL
	DMKRPAGT	DMKRPAPT	DMKRPIRA	DMKSCNAU	DMKSCNVS	DMKSCNVU	DMKSFBNS	DMKSNTLA	DMKSTKCP	DMKVBMIN
	DMKVBMVG	DMKVBMVR	DMKVSWDC	DMKVSWOR	F0	F1	F256	F4	F4095	F4096
	F7	IOERETN	IOKEEP	LANGADDR	LANGBLOK	LANGLANG	LANGLOCK	LANGNEXT	LANGNTRY	LANGNTSZ
	LANGPAGE	LANGSIZE	LANGVMBK	LOCK	MATCH	NLSNAME	NLSNEXT	NLSPGCT	NLSSTRT	NLSTBL
	NLSVOL	PREFIXA	PSA	PSALANG	PSWCC2	PSWCC3	RDEVBLOK	RDEVCODE	RDEVFLAG	RDEVOWN
	RDEVTYPC	RDEVTYPE	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS
	SAVER0	SAVER1	SAVER11	SAVER2	SAVER6	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5
	SAVEWRK7	SAVEWRK8	SAVEWRK9	SFBCLAS	SFBCOPY	SFBDEST	SFBDIST	SFBFILID	SFBFLAG	SFBFLAG3
	SFBFLAG4	SFBFLASH	SFBFNAME	SFBLDBEG	SFBLDMID	SFBLOK	SFBRECNO	SFBSHOLD	SFBSTART	SFBTUSE
	SFBTYPE	SFBUFORM	SFBUHOLD	SFBUSER	SPCHAR	SPCHAR1	SPCHAR2	SPCHAR3	SPCMCHR	SPCMOD
	SPFCB	SPFLSHC	SPLINK	SPSPLNKC	SWPCYL	SWPFLAG	SWPRECMP	SYSTEM	TYPPRT	TYPRDR
	VMBLOK	VSPLCTL	VSPSFBLK	XPAGNUM	X2048BND	ZEROES				
DMKIDR	ADSPCH	AFREE	AFRET	APTRLK	ASYSVM	BLANKS	CC0	CC1	CC2	CC3
	DMKDSPCH	DMKFREE	DMKFRET	DMKIUACP	DMKPTRLK	DMKPTRUL	DMKUDRIA	FFS	IXBLOK	IXEXBLOK
	IXIRA	IXNEXT	IXREGS	IXR11	IXR12	IXSIZE	LOCK	PMSGLIM	PSA	R0
	R1	R10	R11	R12	R13	R14	R15	R2	R3	R4
	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER1	SAVER12	SAVEWRK1
	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	SEVER	SRTBLOK
	SRTFLAG	SRTGIND	SRTIPND	SRTNEXT	SRTPATH	SRTPREV	SRTRESID	SRTRRPND	SRTRVPND	SRTSIZE
	SRTSPND	SRTVMADD	SSCBADDR	SSCBLOK	SSCIXADD	SSCIXLST	SSCRTADD	SSCRTCNT	SSCRTLST	SSCTSFPT
	SSCTSFVM	UIUCBLOK	UIUCGLBL	UIUCREVK	UIUCSTAT	VMBLOK	ZEROES			
DMKIDU	ACORETBL	AFREE	ALOCBLOK	ALOCMAP	ALOCMAX	ALOCNPAG	ALOCPNT	ALOCRDEV	ALOCRECS	APTRAN
	ASYSVM	BALR0	BALR1	BRING	CLASFBA	CLASTERM	CORSWPNT	CORTABLE	C1	DEFER
	DMKALODU	DMKCKSCV	DMKCKTUU	DMKDMPAL	DMKDMPAU	DMKDMPBG	DMKDMPDP	DMKDMPDV	DMKDMPRC	DMKDMPSF
	DMKDMPSI	DMKDRDDD	DMKERMCP	DMKFREE	DMKPGTDS	DMKPGTVR	DMKPGUDU	DMKPSTIN	DMKPTRAN	DMKRIORN
	DMKRPAPT	DMKRSPSC	DMKSCNEP	DMKSCNMU	DMKSCNRA	DMKSYM	DMKSYSDU	DMKSYSOC	DMKSYSOW	DMKSYSRC
	DMKSYSRM	DMKVDGAL	DMKVDHOR	F0	F1	F255	F4096	IOERETN	LOCK	NORET

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	OWNDLIST	OWNDRDEV	PSA	RDEVALLN	RDEVBLOK	RDEVCODE	RDEVNRDY	RDEVSTAT	RDEVTYPC	RECBLOK
	RECCYL	RECMAP	RECMAX	RECPNT	RECSIZE	RECUSED	R0	R1	R10	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	SAVEAREA	SAVEREGS	SFBCDMP	SFBCKPMP	SFBFLAG5	SFBLOK	SFBORIG	SFBSIZE
	SFBSTART	SFBSYSID	SFBUSER	SYSTEM	TEMPR1	TEMPSAVE	VMBLOK			
DMKIMG	CLEAR	CURRSAVE	EGPR0	EGPR15	EGPR8	LOCCNT	MAINHIGH	MODNAME	NUCON	RENAME
	R0	R1	R12	R13	R14	R15	R2	R3	R4	R5
	R6	R7	R8	R9	SAVEAR	SSAVE	STRTADDR	TEXT		
DMKIOC	BALRSAVE	CLASDASD	CLASFBA	CLASTERM	OBRDEVSH	OBRDEVTN	OBRRECN	OBRSHOBR	OBRSWSN	PSA
	RCUBLOK	RCUTYPE	RCU2701	RCU2702	RCU2703	RDCBLOK	RDCDVCLS	RDCOBR	RDEVBLOK	RDEVCUA
	RDEVMDL	RDEVM004	RDEVNAME	RDEVRDC	RDEVSADN	RDEVTMCD	RDEVTYPC	RDEVTYPE	R0	R10
	R12	R14	R15	R4	R5	R6	R8	R9	TYP2305	TYP2311
	TYP3310	TYP3330								
DMKIOE	ADSPCH	AFREE	AFREEP	AFRET	AP	ASYSVM	CCC	CDC	CLASDASD	CLASFBA
	CLASTAPE	CLASURO	CONCCW3	CONDATA	CONDCNT	CPEXADD	CPEXBLOK	CPEXFPNT	CPEXREGS	CPEXR11
	CPEXSIZE	CPUID	DMKCCHRT	DMKCVTAB	DMKCVTBD	DMKDSPCH	DMKFREE	DMKFREEP	DMKFRET	DMKIOCVT
	DMKIOFC1	DMKIOFIN	DMKIOFM1	DMKIOFOB	DMKIOFST	DMKIOFVR	DMKIOGF1	DMKIOGF2	DMKSCNRD	DMKSTKCP
	DMKSYSTZ	D2	EQCHK	ERRBLOK	ERRCCNT	ERRCCW	ERRCORR	ERRCPID	ERRFLAG	ERRHEADR
	ERRIOB	ERRIOER	ERRKEEP	ERRKEY	ERRMIOB	ERRMIOER	ERRMSIZE	ERROBR	ERRPARM	ERRSDR
	ERRSIZE	ERRSRDEV	ERRSW2	ERRVOLID	FTREXTSN	F1	F255	F256	F4	F4095
	F60	F7	F8	IFCC	IOBCP	IOBFATAL	IOBFLAG	IOBHVC	IOBIOER	IOBLOK
	IOBQSTAT	IOBRADD	IOBSIZE	IOBSPEC5	IOBSTAT	IOERBLOK	IOERCCRA	IOERCCRL	IOERCEMD	IOERCPID
	IOERCSW	IOERDATA	IOEREXT	IOERFLG2	IOERFLG3	IOERFLG4	IOERFLM	IOERLEN	IOERPNT	IOERQUE
	IOERSIZE	IOERTEMP	IOERVSER	IRMAND	IRMBIT1	IRMBIT2	IRMBLOK	IRMBYT1	IRMBYT2	IRMFLG
	IRMLMT	IRMLMTCT	IRMMAXCT	IRMOR	IRMRLADD	IRMSIZE	LOCK	MIHCPID	MIHCUA1	MIHCUA2
	MIHDEVT	MIHINT	MIHKEYN	MIHREC	MIHSIZE	MIHSWS3	MIHUSER	MIHVOL	MP	OBRDEVSH
	OBRRECN	OBRSHOBR	OBRSWSN	PSA	RDEVBLOK	RDEVCTRS	RDEVFTR	RDEVIRM	RDEVSER	RDEVSTAT
	RDEVTYPC	RDEVTYPE	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS
	SAVER1	SAVEWRK2	SAVEWRK6	SAVEWRK7	SDRBLOK	SDRBSIZE	SDRCPID	SDRCTRS	SDRCUA	SDRFLAGS
	SDRLNGTH	SDRRECD	SDRSHRT	TYP2305	TYP3330	TYP3340	TYP3350	TYP3375	TYP3380	TYP3410
	TYP3420	TYP3422	TYP3430	TYP3480	TYP3800	TYP38003	TYP38008	TYP4245	TYP4248	TYP8809
	UC	VMBLOK	XOBRFLAG	XOBRT1	XOBRT3	ZERO	ZEROES			
DMKIOF	ADSPCH	AFREE	AFREEP	AFRET	APTRAN	APTRLK	BRING	CDC	CLASDASD	CLASFBA
	CLASGRAF	CLASTAPE	CLASTERM	CLASURI	CLASURO	CPEXBLOK	CPEXFPNT	CPEXREGS	CPEXR6	CPEXSIZE
	CPUVERSN	DEFER	DMKDSPCH	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKIOCVT	DMKIOECL	DMKIOECQ
	DMKIOEEP	DMKIOEES	DMKIOEIP	DMKIOEIQ	DMKIOEMQ	DMKIOEMX	DMKIOENI	DMKIOENQ	DMKIOERP	DMKIOERQ
	DMKIOESQ	DMKIOEVQ	DMKIOJBL	DMKIUACP	DMKPGUVG	DMKPGUVR	DMKPTRAN	DMKPTRLK	DMKPTRUL	DMKRPAGT
	DMKRPAPT	DMKSTKCP	ERRBLOK	ERRFLAG	ERRIOB	ERRIOER	ERRKEEP	ERROBR	ERRPARM	ERRPATH
	ERRSDR	FFS	F15	F255	F4	F4095	F7	IOERBLOK	IOERCPID	IOERCSW
	IOERDATA	IOERFLG3	IOERREAD	IXBLOK	IXIRA	IXREGS	IXSIZE	LOCK	PREFIXA	PSA
	RDEVBLOK	RDEVCTRS	RDEVTYPC	RDEVTYPE	RECCCPD	RECFLAG1	RECNXT	RECPAG	RECPAGFL	RECPAGIU
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER6	SAVEWRK2
	SDRBLOK	SDRCPID	SDRCTRS	SDRCTR8	SDRCTR9	SDRCUA	SDRFLCT	SDRLNGTH	SDRMAX	SDROVFWK
	SDRPRMCT	SDRRDEV	SDRRECD	SDRSHRT	SDRSIZE	SDRSIZE1	SEVER	SYSTEM	TYPBSC	TYPSDLC
	TYPTTY	TYP1050	TYP1403	TYP1443	TYP2501	TYP2520R	TYP2540R	TYP2700	TYP2741	TYP3066
	TYP3210	TYP3211	TYP3410	TYP3420	TYP3422	TYP3430	TYP4248	VMBLOK	ZERO	
DMKIOG	AFREE	ALLCHANS	AMCHAREA	AP	APSTAT1	APSTAT4	APTRAN	APTRLK	APUOPER	ARIOCH
	ARIOCT	ASYSVM	BRING	CHANID	CLASFBA	CLEAR	CPEXSIZE	CPSTAT5	CPSTAT6	CPTCH

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	CPUID	CPUMCELL	CPUMODEL	CPUVERSN	C1	DEFER	DMKCCHCF	DMKCCHMX	DMKCCHSZ	DMKCCH60
	DMKEIG80	DMKERMSG	DMKFREE	DMKIOECE	DMKIOECL	DMKIOECT	DMKIOEEP	DMKIOEES	DMKIOEFR	DMKIOEHS
	DMKIOEIF	DMKIOEMX	DMKIOENI	DMKIOETY	DMKIOHFR	DMKPGUVG	DMKPGUVR	DMKPTRAN	DMKRPAGT	DMKRPAPT
	DMKSCNRU	DMKSEV70	DMKSIX60	DMKSYSCT	DMKSYSEA	DMKSYSER	ECSWLOG	FFS	FXDLOG	F0
	F1	F255	F7	G16CHAN	IOELPNTR	IPUADDR	LOCK	LPUADDR	LPUADDRX	MCHAREA
	MCHCPEX	MCHDAMLN	MCHFIX	MCHLEN1	MCHMODEL	MCHPROCA	MCHREC	MCNPSW	MODEL135	MODEL138
	MODEL145	MODEL148	MODEL155	MODEL158	MODEL165	MODEL168	MOD3031	MOD3032	MOD3033	MOD3081
	MOD3083	MOD3084	MOD3090	MOD4321	MOD4331	MOD4341	MOD4381	MOD9081	MOD9083	MOD9190
	MP	MPUOPER	NOMODEL	POFFLINE	PREFIXB	PROCIPL	PSA	RCHBLOK	RCHDISA	RCHSTAT
	RCHSTIDC	RCHTYPE	RCH370	RDEVBLOK	RDEVCODE	RDEVTYPC	RDEVTYPE	RECCCPD	RECFLAG1	RECFLAG2
	RECMODE	RECNXT	RECPAG	RECPAGDN	RECPAGFA	RECPAGFL	RECPAGFM	RECPAGFR	RECPAGIU	R0
	R1	R10	R11	R12	R13	R14	R15	R2	R3	R4
	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVEWRK2	SAVEWRK7	SYSTEM
	TYP2305	TYP3330	TYP3340	TYP3350	TYP3375	TYP3380	VMBLOK	WAIT	ZERO	
DMKIOH	ADSPCH	AFREE	AFRET	AMCHAREA	AP	APTRLK	BRING	CE	CLASFBA	CLASSPEC
	CPUID	DE	DMKCVTAB	DMKCVTBH	DMKDSPCH	DMKERMSG	DMKFREE	DMKFRET	DMKIOECE	DMKIOECL
	DMKIOEES	DMKIOEFR	DMKIOEHS	DMKIOEIF	DMKIOEMX	DMKIOENI	DMKIOETY	DMKIOSQR	DMKPGUVG	DMKPGUVR
	DMKPTRLK	DMKPTRUL	DMKRIODV	DMKRIOSF	DMKRPAGT	DMKRPAPT	DMKSCNMU	DMKSCNRU	DMKSYSER	DMKSYSTZ
	F1	F10	F6	IOBCAW	IOBCC2	IOBCSW	IOBFATAL	IOBFLAG	IOBIRA	IOBLINK
	IOBLOK	IOBRADD	IOBSIZE	IOBSTAT	IOBUSER	LOCK	MCHAREA	MCHMODEL	MOD3033	MP
	PSA	RECFLAG1	RECNXT	RECPAG	RECPAGFA	RECPAGFL	R0	R1	R10	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	SAVEAREA	SAVEREGS	SYSTEM	TYPSRF	UE	VMBLOK		
DMKIOJ	AP	BALRSAVE	CLASDASD	CLASFBA	CLASGRAF	CLASSPEC	CLASTAPE	CLASTERM	CLASURO	CPUID
	ERRBLOK	ERRCCNT	ERRCCW	ERRCONT	ERRCORR	ERRCPID	ERRIOB	ERRIOER	ERRKEY	ERRMIOB
	ERRMIOER	ERRSDR	ERRVOLID	FTREXTSN	F15	F24	F4	F7	F8	IOBFATAL
	IOERADR	IOERBLOK	IOERCSW	IOERDATA	IOEREXT	IOERLEN	IOERSIZE	IOERVSER	LOCK	LPUADDR
	MP	OBRBLKLN	OBRCCHS	OBRCORL	OBRCPIDN	OBRCSWN	OBRCUAIN	OBRCUAPR	OBRDDCNT	OBREOD
	OBRFCCWN	OBRHAN	OBRIORTY	OBRKEYN	OBRLSKN	OBRPBN	OBRPGMN	OBRRECN	OBRSDRCT	OBRSDRSH
	OBRSHOBR	OBRSNSCT	OBRSSDR1	OBRSWSN	OBRTEMP	OBRVOLN	OBR33SNS	PREFIXB	PSA	RDCBLOK
	RDCMDR	RDEVBLOK	RDEVCTRS	RDEVFTR	RDEVMDL	RDEVM004	RDEVRDC	RDEVTYPC	RDEVTYPE	RECPAG
	R0	R1	R10	R12	R14	R15	R2	R3	R4	R5
	R6	R7	R8	R9	SDRBLOK	SDRCTRS	SDRFLAGS	SDRLNGTH	SDRRECD	SDRSHRT
	TNSCPIDN	TNSDEVAD	TNSKEYN	TNSREC	TNSSNS1	TNSSWS3	TNSVOLID	TNS3480S	TNS3800	TNS8809S
	TYP2305	TYP3203	TYP3211	TYP3262	TYP3289E	TYP3310	TYP3330	TYP3340	TYP3350	TYP3370
	TYP3375	TYP3380	TYP3410	TYP3420	TYP3422	TYP3430	TYP3480	TYP3505	TYP3800	TYP38003
	TYP38008	TYP4245	TYP8809	VMBLOK	XOBRFCB	XOBRFLAG	XOBRT1	XOBRT3	XOBR010	XOBR150
	XOBR512	ZERO	ZEROES							
DMKIOQ	AFREEP	AFRET	ALOKSP	AP	APSTAT1	APUOPER	ARIOCH	ARIOCT	ARIOCU	ASYSVM
	AVMREAL	BUSY	CALLER	CLASDASD	CLASFBA	CPEXADD	CPEXBLOK	CPEXMISC	CPEXREGS	CPEXR11
	CPEXR15	CPEXR6	CPEXR7	CPEXSIZE	CPSPMODE	CPSTAT2	DE	DMKFREEP	DMKFRET	DMKIOSCB
	DMKIOSMQ	DMKIOSNM	DMKIOSW1	DMKSCHRT	DMKSCNRD	DMKSCNRU	DMKSTKIO	DMKSTKLF	DMKVIOIN	FFS
	F0	F1	IOBBPNT	IOBCC1	IOBCC3	IOBCP	IOBCSW	IOBCTRQ	IOBCUBSY	IOBCYL
	IOBERP	IOBFATAL	IOBFH	IOBFLAG	IOBFLT	IOBFPNT	IOBHVC	IOBIRA	IOBLINK	IOBLOK
	IOBMINI	IOBMQDTO	IOBOFF	IOBPAG	IOBPATHF	IOBPROC	IOBQDTO	IOBRADD	IOBRELCU	IOBRSTRT
	IOBSIOF	IOBSIZE	IOBSNSIO	IOBSPEC	IOBSPEC2	IOBSPEC3	IOBSPEC4	IOBSPM	IOBSTAT	IOBUNSL
	IOBUSER	IOBVADD	IOBVCUE	LOCK	LOCKSAV	LPUADDR	LPUADDRX	MP	MPGEND	MPUOPER
	PREFIXA	PREFIXB	PROCIPL	PSA	RCHADD	RCHBLOK	RCHBUSY	RCHCUTBL	RCHFIOB	RCHMPX
	RCHPEND	RCHPROC	RCHQCNT	RCHRSTQ	RCHSTAT	RCHTYPE	RCUADD	RCUBLOK	RCUBUSY	RCUCHA
	RCUCHCNT	RCUCUBSY	RCUDISA	RCUFIOB	RCUPRIME	RCUQCNT	RCURSTQ	RCUSCED	RCUSENIO	RCUSENSE
	RCUSHRD	RCUSTAT	RCUSUB	RCUTYPE	RDEVADD	RDEVAOF	RDEVBLOK	RDEVBOF	RDEVBUZY	RDEVCUA
	RDEVCUB	RDEVCYL	RDEVDED	RDEVFIOB	RDEVFLAG	RDEVIOBL	RDEVLIOB	RDEVPIOB	RDEVPTHS	RDEVQBSY

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)								
	RDEVQCNT	RDEVQIOB	RDEVSCHD	RDEVSKUP	RDEVSTAT	RDEVSTA4	RDEVTYPC	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R7	R8	SAVEAREA	SAVER11	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK6	SAVEWRK7
	SAVEWRK8	SAVEWRK9	TIMEDISP	TRQBSIZE	VMBLOK	XTNDLOCK				
DMKIOS	ADSPCH	AEXTSP	AFREE	AFREEP	AFRET	ALOKSP	ALOKSY	AP	APSTAT1	APUOPER
	ARIOCH	ARIOCT	ARIOCU	ASYSVM	ATTN	BALR0	BLANKS	BUSY	CALLER	CAW
	CC	CCC	CDC	CE	CHC	CLASDASD	CLASFBA	CLASGRAF	CLASSPEC	CLASTAPE
	CLASTERM	CLASURI	CLASURO	CLEAR	CODE	CPCREG8	CPEXADD	CPEXBLOK	CPEXR13	CPEXR5
	CPEXSIZE	CPRUN	CPSHUT	CPSPMODE	CPSTAT	CPSTATUS	CPSTAT2	CPSTAT4	CPSTAT6	CPTCH
	CPWAIT	CSW	CUE	DE	DFRCC1	DMKACRC3	DMKBSCER	DMKCCHIS	DMKDADER	DMKDASER
	DMKDAUER	DMKDSPA	DMKDSPCH	DMKENTSK	DMKFREE	DMKFREEP	DMKFRET	DMKIOERR	DMKIOQDE	DMKIOQDH
	DMKIOQDQ	DMKIOQDU	DMKIOQFC	DMKIOQFP	DMKIOQFX	DMKIOQQD	DMKIOQQU	DMKIOQSK	DMKIOQUS	DMKLOKDF
	DMKLOKIO	DMKLOKSY	DMKRSPER	DMKSCHDL	DMKSCNPH	DMKSCNRD	DMKSCNRU	DMKSSUI1	DMKSTKCP	DMKSTKIO
	DMKTAPER	DMKTPERP	DMKTRDSI	DMKVIOIN	FFS	FTRRPS	F10	F16	F2	F256
	F8	IFCC	IL	INTREQ	IOBBPNT	IOBCAW	IOBCC1	IOBCC2	IOBCC3	IOBCLRIO
	IOBCP	IOBCSW	IOBCUBSY	IOBCYL	IOBERCNT	IOBERP	IOBFATAL	IOBFH	IOBFLAG	IOBFPNT
	IOBHIO	IOBHVC	IOBIMSTK	IOBIOER	IOBIOLOK	IOBIRA	IOBLDRUN	IOBLDSZ	IOBLINK	IOBLOK
	IOBMID	IOBMINI	IOBOFF	IOBPAG	IOBPATH	IOBPATHF	IOBPROC	IOBPST	IOBPVM	IOBQDTO
	IOBQSTAT	IOBRADD	IOBRCAW	IOBRELCU	IOBRES	IOBRETRY	IOBRREL	IOBRSTRT	IOBRTCT	IOBRTCT2
	IOBSIMS	IOBSIOEX	IOBSIOF	IOBSIZE	IOBSNSIO	IOBSPEC	IOBSPEC2	IOBSPEC3	IOBSPEC4	IOBSPEC5
	IOBSPLT	IOBSPM	IOBSTAT	IOBTIO	IOBUC	IOBUNSL	IOBUSER	IOBVADD	IOBVCUE	IOBVHIO
	IOERBLOK	IOERCCRA	IOERCCW	IOERCPID	IOERCSW	IOERDATA	IOEREXT	IOERFLG3	IOERLEN	IOERNOLG
	IOERSIZE	IOERSNSZ	IPUADDR	IPUADDRX	LASTUSER	LOCK	LOCKSAV	LPUADDR	MNCLSEEK	MNCOCYL
	MP	MPUOPER	PCI	PREFIXA	PREFIXB	PRGC	PRTC	PSA	PSAIOSW	RCHADD
	RCHBLOK	RCHBMX	RCHBUSY	RCHCUTBL	RCHFIOB	RCHMPX	RCHPEND	RCHPROC	RCHQCNT	RCHRSTQ
	RCHSCED	RCHSEL	RCHSTAT	RCHTYPE	RCH370	RCUBLOK	RCUBUSY	RCUCHA	RCUCUBSY	RCUFIOB
	RCUPRIME	RCUQCNT	RCURSTQ	RCUSCED	RCUSENIO	RCUSENSE	RCUSHRD	RCUSTAT	RCUSUB	RCUTYPE
	RCU3880	RCWCCNT	RCWCCW	RCWTASK	RDEVAIOB	RDEVBLOK	RDEVBUZY	RDEVBZCH	RDEVCUA	RDEVCUB
	RDEVCYL	RDEVDED	RDEVDISA	RDEVFIOB	RDEVFLAG	RDEVFTR	RDEVIOBL	RDEVIOCT	RDEVIOER	RDEVMID
	RDEVNRDY	RDEVOFF	RDEVPIOB	RDEVPROC	RDEVQBSY	RDEVQCNT	RDEVQIOB	RDEVRDY	RDEVRSTA	RDEVSCHD
	RDEVSENS	RDEVSER	RDEVSTAT	RDEVSTA2	RDEVSTA3	RDEVSTA4	RDEVSTA5	RDEVSTA7	RDEVSTMD	RDEVTYPC
	RDEVTYPE	RDEVUNSN	RDEV333V	RUNUSER	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	SAVERETN	SAVER11	SAVEWRK1	SAVEWRK2	SAVEWRK4	SIGWAKE	SIGXC	SILI
	SKIP	SM	SYSVIRT	TEMPSAVE	TIMEDISP	TRACBEF	TRACCURR	TRACEND	TRACFLG2	TRACPROC
	TRACSTRT	TRACSVCR	TRCSIOF	TRCTCH	TYPBSC	TYPCTCA	TYPTTY	TYP2305	TYP2401	TYP3203
	TYP3211	TYP3262	TYP3289E	TYP3330	TYP3375	TYP3380	TYP3480	TYP4245	UC	VIRTUAL
	VMBLOK	XCDISP	XCPEND	XTNDLOCK	ZERO	ZEROES				
DMKIOT	ADSPCH	AFREEP	AFRET	ALOKSP	ALOKSY	AP	APSTAT1	APTRAN	APTRLK	APUOPER
	ASYSVM	ATTN	AVMREAL	BALR0	BALR2	BLANKS	BRING	BUSY	CAW	CCC
	CDC	CE	CHC	CLASDASD	CLASFBA	CLASGRAF	CLASSPEC	CLASTAPE	CLASTERM	CLASURI
	CLASURO	CMDREJ	CODE	CPCREG0	CPCREG8	CPEXADD	CPEXBLOK	CPEXFPNT	CPEXR13	CPEXSIZE
	CPINITD	CPQVMCU	CPRUN	CPSPMODE	CPSTATUS	CPSTAT2	CPSTAT4	CPSUPER	CPWAIT	CSW
	CUE	C0	C1	C2	C6	C8	DE	DEFER	DFRCC	DFRCC1
	DMKCCHNT	DMKCNSIN	DMKCPX	DMKCPXZT	DMKDIAIR	DMKDMPC2	DMKDSBRD	DMKDSPA	DMKDSPCH	DMKDSPQI
	DMKDSPVM	DMKFREEP	DMKFRET	DMKGRFIN	DMKIOSCB	DMKIOSC1	DMKIOSC3	DMKIOSEN	DMKIOSER	DMKIOSRC
	DMKIOSRH	DMKIOSRQ	DMKIOSRS	DMKIOSRU	DMKIOSST	DMKLOKIO	DMKLOKSY	DMKPMAEW	DMKPMAIO	DMKPTRAN
	DMKQVMCU	DMKQVMTS	DMKRGAIN	DMKRIOCN	DMKRNHIN	DMKRSPEX	DMKSCNPH	DMKSCNRU	DMKSSUI1	DMKSSUI2
	DMKSTKCP	DMKSTKIO	DMKSTKMP	DMKVIOIN	EMSMASK	FFS	F0	F1	F8	IFCC
	IL	INTREQ	INTTIO	IOBBPNT	IOBCAW	IOBCC1	IOBCC3	IOBCLRIO	IOBCP	IOBCSW
	IOBCUBSY	IOBERP	IOBFATAL	IOBFLAG	IOBFPNT	IOBHIO	IOBHVC	IOBIOER	IOBIRA	IOBLINK
	IOBLOK	IOBMISC	IOBPAG	IOBPMINT	IOBPVM	IOBQDTO	IOBQSTAT	IOBRADD	IOBRCAW	IOBRCNT
	IOBRES	IOBRETRY	IOBRSTRT	IOBSIOEX	IOBSIOF	IOBSIZE	IOBSNSIO	IOBSPEC	IOBSPEC2	IOBSPEC3

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	IOBSPEC4	IOBSPEC5	IOBSPLT	IOBSTAT	IOBTIO	IOBUC	IOBUNSL	IOBUSER	IOBVADD	IOBVCUE
	IOERBLOK	IOERCCRA	IOERCSW	IOERDATA	IOEREXT	IOERLEN	IOERSIZE	IOOPSW	LASTUSER	LOCK
	LPUADDR	MFAMASK	MICBLOK	MICPMMSK	MP	PCI	PMAMODE	PMASTAT	PREFIXA	PRGC
	PROCIO	PRTC	PSA	PSAIOSW	QUANTUMR	RCHBLOK	RCHBUSY	RCHSTAT	RCUBLOK	RCUBUSY
	RCUCUBSY	RCUDISA	RCUFIOB	RCUPRIME	RCUQCNT	RCUSCED	RCUSENIO	RCUSTAT	RCUSUB	RCUTYPE
	RDEVAIOB	RDEVATT	RDEVBLOK	RDEVBUZY	RDEVBZCH	RDEVCC3	RDEVCPEX	RDEVCPIO	RDEVCSW	RDEVDED
	RDEVDIIP	RDEVDISA	RDEVFIOB	RDEVFLAG	RDEVFTR	RDEVIOER	RDEVMID	RDEVNRDY	RDEVOFF	RDEVPIOB
	RDEVPROC	RDEVQBSY	RDEVQCNT	RDEVQIOB	RDEVRDY	RDEVRSTA	RDEVSCHD	RDEVSENS	RDEVSPT	RDEVSTAT
	RDEVSTA2	RDEVSTA3	RDEVSTA4	RDEVSTA5	RDEVSTA7	RDEVTYPC	RDEVTYPE	RDEVUNSN	RDEVUSER	RDEV333V
	RUNCRO	RUNUSER	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SAS	SAVEAREA
	SAVEWRK1	SAVEWRK2	SAVEWRK3	SILI	SKIP	SM	SPTBLOK	SPTFLAG3	SPTINTAD	SPTOFLOW
	SYNCMASK	SYSTEM	SYSVIRT	TEMPR9	TIMEDISP	TIMER	TRACBEF	TRACCURR	TRACEND	TRACFLG1
	TRACFLG2	TRACPROC	TRACSTRT	TRACSVCR	TRAC05	TRCSMINT	TYPBSC	TYP2401	TYP3277	TYP3278
	TYP3284	TYP3330	TYP3800	TYP38003	TYP38008	UC	UE	VIRTUAL	VMBLOK	WAITEND
	XCMASK	Y0	Y2	Y4	Y6	ZEROES				
DMKISM	AFREE	CD	C1	DMKFREE	DMKPTRAN	DMKPTRUL	DMKUNTIS	F16	F2	F4
	F8	IDA	IOBCAW	IOBIRA	IOBLOK	IOBMISC	PSA	RCWCCNT	RCWCCW	RCWIO
	RCWPNT	RCWRCNT	RCWTASK	RCWVCAW	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	VMBLOK	XPAGNUM						
DMKIUA	ADSPCH	AFREE	AFREEP	AFRET	APSTAT1	APTRAN	APUOPER	ASYSIUCV	ASYSVM	BRING
	CCTFLAG1	CCTFLAG2	CCTFLCNT	CCTMSGCT	CCTMXPID	CCTPDSEG	CCTPDSLO	CCTRCVHD	CCTRCVTL	CCTRPYHD
	CCTRPYN	CCTRPYP	CCTRPYPR	CCTRPYTL	CCTSNDHD	CCTSNDN	CCTSNDP	CCTSNDPR	CCTSNDTL	CC2
	CODE	CPEXADD	CPEXBLOK	CPEXFPNT	CPEXMISC	CPEXR0	CPEXR10	CPEXR3	CPEXR4	CPEXR5
	CPEXSIZE	C1	DEFER	DMKDSPCH	DMKFREE	DMKFREEP	DMKFRET	DMKIUCEP	DMKIUEEP	DMKIUGEP
	DMKIUJEP	DMKIULEP	DMKIUNEP	DMKIUNIN	DMKIUPEP	DMKIUSEP	DMKLOKSW	DMKPRGSM	DMKPSASP	DMKPTRAN
	DMKSTKCP	F0	F1	F4	F4095	F5	F6	F60	IETYPRNP	IETYPRP
	IUCBFAD1	IUCLINK1	IUCVBLOK	IUCVCPEX	IUCVMB	IUSAVE	IUSCCODE	IUSCPENT	IUSFCODE	IUSFLAGS
	IUSFLAG2	IUSINSTR	IUSIUCV	IUSIUCV2	IUSLEN1	IUSLEN2	IUSPAGE1	IUSPAGE2	IUSPARMS	IUSRCODE
	IUSSIZE	IUSVMBK	IXBLOK	IXEXBLOK	IXIRA	IXREGS	IXR11	LOCK	MSGANSLN	MSGAUDIT
	MSGBLOK	MSGERROR	MSGFLAGS	MSGFLAG3	MSGFPNT	MSGNOFL	MSGPRTY	MSGPURGE	MSGSCPID	MSGSIZE
	MSGSNDOP	MSGTAG	MSGTGPID	MSGUSED	MSGWHTRC	PDAPPC	PDAPPCFL	PDAPPEND	PDAPREQS	PDCPXQ
	PDENT	PDFLAGS	PDINUSE	PDMSGCT	PDSEVERD	PDSINV	PDTGIUCV	PDVALID	PREFIXA	PROBMODE
	PROPSW	PSA	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS
	SAVERETN	SAVER0	SAVER1	SAVESIZE	SAVEWRK4	SAVEWRK5	TIMEDISP	TRACCURR	TRACEND	TRACFLG3
	TRACIUCV	TRACPROC	TRACSTRT	TRACSVCR	TRCIUCV	VMBLOK	WAIT	XPAGNUM	ZEROES	
DMKIUB	AFREE	AFRET	APTRAN	BRING	CCTFLAG1	CCTFLAG2	CCTFLAG3	CCTICTRL	CCTPNDCT	CCTPNDRN
	CCTPNDRP	CCTPNDSN	CCTPNDSP	CCTRPYHD	CCTRPYN	CCTRPYP	CCTRPYPR	CCTRPYTL	CCTSNDHD	CCTSNDN
	CCTSNDP	CCTSNDPR	CCTSNDTL	C1	DEFER	DMKAPSCN	DMKAPSIL	DMKAPSSV	DMKBIOCN	DMKBIOIL
	DMKBIOSV	DMKCRMCN	DMKCRMIL	DMKCRMQS	DMKCRMSV	DMKFREE	DMKFRET	DMKIDRCN	DMKIDRSV	DMKIOFCN
	DMKIOFSV	DMKIUAPD	DMKIUAQU	DMKIUNIN	DMKIUSET	DMKMSGCN	DMKMSGC2	DMKMSGSV	DMKMSGS2	DMKPTRAN
	DMKRPICN	DMKRPIIL	DMKRPIQS	DMKRPIRM	DMKRPISV	DMKVCPIL	DMKVCWCN	DMKVCWQS	DMKVCWRM	DMKVCWSV
	DMKVMGCN	DMKVMGIL	DMKVMGSV	F0	IUCBFAD1	IUCCBFA1	IUCPNDHD	IUCPNDTL	IUCVBLOK	LOCK
	MSGANSLN	MSGAPPC	MSGAUDIT	MSGBLOK	MSGCTLS	MSGCTLT	MSGDESC	MSGERROR	MSGFLAGS	MSGFLAG2
	MSGFLAG3	MSGFPNT	MSGID	MSGMASK1	MSGNORPY	MSGPRM	MSGPRMD	MSGPRTY	MSGPURGE	MSGSCCLS
	MSGSCPID	MSGSNDLN	MSGSNDOP	MSGTAG	MSGTGCLS	MSGTGPID	MSGWHTRC	PDAPMSGP	PDAPPCFL	PDCNTRL
	PDENT	PDFLAGS	PDSTATE	PDSTRECV	PDSTSEND	PSA	RECQMSG	RPYQMSG	R0	R1
	R10	R11	R12	R13	R14	R15	R2	R3	R4	R5
	R6	R8	R9	SAVEAREA	SAVEREGS	SAVEWRK1	SAVEWRK2	SAVEWRK5	VMBLOK	

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
DMKIUC	ADSPCH	AFREE	AFREEP	AFRET	ALTBLOK	ALTUSER	APTRAN	ASYSIUCV	BRING	CCMASK
	CCTFLAG2	CCTFLCNT	CCTPNDCT	CC0	CC2	CC3	CONDATAL	CONEXT	CONEXTSZ	CONFMH5
	CONFSYNC	CONFTPN	CONIDATA	CONILUNM	CONIMODE	CONLENTH	CONMSGBK	CONNECT	CONP1LEN	CONP2LEN
	CONVFIXL	CONVLUNM	CONVMARE	CONVMODE	CONVTOTL	CPEXADD	CPEXBLOK	CPEXMISC	CPEXR0	CPEXR10
	CPEXR14	CPEXSIZE	C1	DEFER	DMKDSPCH	DMKFREE	DMKFREEP	DMKFRET	DMKIDRFN	DMKIUAPD
	DMKIUBTB	DMKIUGGP	DMKIUNIN	DMKPSAFP	DMKPTRAN	DMKSCNAU	DMKSTKCP	DMKUDRIA	DMKUDRRV	F0
	F10	IUCBFAD1	IUCCBFA1	IUCMXCN	IUCPNDHD	IUCPNDTL	IUCTOTCN	IUCVBLOK	IUCVCPEX	IUCVMB
	IUSAVE	IUSCCODE	IUSCPENT	IUSFLAG2	IUSIUCV	IUSPAGE1	IUSPATH	IXBLOK	IXEXBLOK	IXIRA
	IXNEXT	LOCK	MSGAPPC	MSGBLOK	MSGFLAGS	MSGFLAG2	MSGNORPY	MSGSCPID	MSGSNDAD	MSGSNDLN
	MSGSNDOP	MSGTGPID	PDALLOCD	PDAPPC	PDAPPCFL	PDAPSYCF	PDCNTRL	PDENT	PDFCNCD	PDFLAGS
	PDMSGLIM	PDPEND1	PDPEND2	PDPRMD	PDPRTY	PDSEND	PDSEVERD	PDSTALLC	PDSTATE	PDSTCONN
	PDSTRECV	PDSTSEND	PDTGIUCV	PDTGPID	PDVALID	PENDCONN	PREFIXA	PSA	RCBADDIR	RCBADFCN
	RCBADLIM	RCCOMSRV	RCEXTLEN	RCINVLUN	RCINVMOD	RCINVSRV	RCNLOG	RCNOPATH	RCNTRGIU	RCPTHSVD
	RC2MANY	RC2MANYT	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS
	SAVER0	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK7	SAVEWRK8	SAVEWRK9	UIUCBLOK	UIUCMLIM
	UIUCPRTY	UIUCSTAT	VMBLOK	XPAGNUM	ZEROES					
DMKIUE	AFRET	APSTAT1	APTRAN	APTRLK	APUOPER	ASYSIUCV	BALRSAVE	BALR15	BRING	CCMASK
	CCTSNDHD	CCTSNDPR	CCTSNDTL	CONEXT	CONLENTH	C1	DEFER	DMKATSCF	DMKFRET	DMKIUAPD
	DMKIUAQU	DMKIUSPD	DMKLOKSW	DMKPSACC	DMKPSAFC	DMKPSASC	DMKPSASP	DMKPTRAN	DMKPTRUL	F0
	F1	F4095	IUCVBLOK	IUCVMB	IUSAVE	IUSCCODE	IUSCPENT	IUSFLAG2	IUSIUCV	IUSMSGBK
	IUSPATH	LOCK	MSGABLEN	MSGABTOT	MSGAIINV	MSGAITRN	MSGANSLN	MSGAPPC	MSGARCAX	MSGARCPX
	MSGARLST	MSGASNAX	MSGASNPX	MSGAUDT1	MSGAUDT2	MSGAUDT3	MSGBLIST	MSGBLOK	MSGCTLT	MSGDESC
	MSGFLAGS	MSGFLAG2	MSGFLAG3	MSGFPNT	MSGID	MSGKEY	MSGMASK1	MSGNORPY	MSGPARTL	MSGPRM
	MSGPRMD	MSGPURGE	MSGSCPID	MSGSNDAD	MSGSNDLN	MSGSNDOP	MSGTGCLS	MSGTGPID	MSGWHTRC	PDALLOCD
	PDAPPC	PDAPPCFL	PDENT	PDFLAGS	PDFLAGS2	PDLRCINV	PDLRECL	PDSEVERD	PDSTATE	PDSTCONN
	PDTGIUCV	PSA	RCADDRCK	RCAPPC	RCBUFBND	RCMSGLEN	RCNEGLEN	RCNODATA	RCNOPATH	RCPROTCK
	RCPURGED	RCRECVSH	RCSNDLST	RCTINVLN	RCTOTLEN	RCTTRUNC	RECQMSG	RPYQMSG	R0	R1
	R10	R11	R12	R13	R14	R15	R2	R3	R4	R5
	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER0	SAVEWRK1	SAVEWRK2	SAVEWRK3
	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	SENDQMSG	TIMEDISP	VMBLOK	VMSIZE
	XRIGHT16	ZEROES								
DMKIUG	AFREE	AFRET	CCMASK	CCTMXPDE	CCTMXPDS	CCTPDSEG	CCTPDSLO	CCTRCVHD	CCTRCVTL	CCTRPYHD
	CCTRPYPR	CCTRPYTL	CCTSNDHD	CCTSNDPR	CCTSNDTL	DMKFREE	DMKFRET	DMKIUAPD	DMKIUAQU	F0
	F6	IUCMXCN	IUCVBLOK	IUSAVE	IUSCCODE	IUSCPENT	IUSFLAG2	IUSIUCV	IUSMSGBK	IUSPATH
	LOCK	MSGAPPC	MSGARJCT	MSGAUDIT	MSGAUDT1	MSGBLOK	MSGCTLS	MSGCTLT	MSGDESC	MSGFLAGS
	MSGFLAG2	MSGFLAG3	MSGFPNT	MSGID	MSGMASK1	MSGNORPY	MSGPRMD	MSGPURGE	MSGSCCLS	MSGSCPID
	MSGTAG	MSGTGCLS	MSGTGPID	PDAPPC	PDAPPCFL	PDAVAIL	PDENT	PDENTMAX	PDENTMIN	PDFLAGS
	PDSEVERD	PDSINV	PDSIZE	PDTGIUCV	PDTGPID	PDVALID	PSA	RCAPPC	RCNODATA	RCNOPATH
	RPYQMSG	R0	R1	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER0	SAVER1
	SAVER2	SENDQMSG	VMBLOK							
DMKIUJ	ADSPCH	AFREE	AFREEP	AFRET	APTRLK	ASYSIUCV	ASYSVM	CCMASK	CCTFLAG2	CCTFLCNT
	CCTMSGCT	CCTMXPDE	CCTMXPDS	CCTMXPID	CCTPDSEG	CCTPNDCT	CCTRCVHD	CCTRPYHD	CCTSNDHD	CC0
	CC1	CC2	CC3	CONEXT	CONLENTH	CPEXADD	CPEXBLOK	CPEXFPNT	CPEXMISC	CPEXR0
	CPEXR10	CPEXR14	CPEXR3	CPEXSIZE	DMKCVTAB	DMKDSPCH	DMKFREE	DMKFREEP	DMKFRET	DMKIUACU
	DMKIUAPD	DMKIUBTB	DMKIUNIN	DMKPTRLK	DMKPTRUL	DMKSCHST	DMKSTKCP	F1	IUCPNDHD	IUCPNDTL
	IUCTOTCN	IUCVBLOK	IUCVCPEX	IUCVMB	IUSAVE	IUSCCODE	IUSCPENT	IUSFLAG2	IUSIUCV	IUSPATH
	IXBLOK	IXEXBLOK	IXIRA	IXNEXT	IXSIZE	LOCK	MSGAPPC	MSGASVRD	MSGAUDT2	MSGBLOK
	MSGERROR	MSGFLAGS	MSGFLAG2	MSGFPNT	MSGID	MSGPURGE	MSGSCCLS	MSGSCPID	MSGSNDLN	MSGSNDOP
	MSGTGCLS	MSGTGPID	MSGWHTRC	PDALLOCD	PDAPLOCK	PDAPPC	PDAPPCFL	PDAPPEND	PDAVAIL	PDCNTRL
	PDCPXQ	PDENT	PDFCNCD	PDFLAGS	PDFLAGS2	PDLRCINV	PDLRECL	PDMSGCT	PDPEND1	PDPEND2

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	PDSEND	PDSEVERD	PDSIZE	PDSTALLC	PDSTATE	PDSTCONF	PDSTCONN	PDSTRECV	PDSTSEND	PDSTSEVR
	PDTGIUCV	PDTGPID	PDZERO	PENDCONN	PREFIXA	PSA	RCCOMSRV	RCINVCNF	RCINVCON	RCINVREC
	RCINVSCD	RCINVSEV	RCMSGCT	RCNONAPP	RCNOPATH	RCSNDOP	RCTRUNC	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER0	SAVER1	SAVEWRK1	SAVEWRK2	SAVEWRK3
	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	SEVER	TRQBIRA	TRQBLOK	TRQBSIZE	TRQBUSER	TRQBVAL
	TRQREGS	TRQREGSZ	VMBLOK	ZEROES						
DMKIUL	APSTAT1	APTRAN	APTRLK	APUOPER	ASYSIUCV	BRING	CCMASK	CCTRCVHD	CCTRCVTL	CCTRPYHD
	CCTRPYPR	CCTRPYTL	CC1	CC2	CC3	C1	DEFER	DMKATSCF	DMKIUAPD	DMKIUAQU
	DMKIUELR	DMKIUSET	DMKIUSPD	DMKLOKSW	DMKPSACC	DMKPSAFP	DMKPSASC	DMKPSASP	DMKPTRAN	DMKPTRUL
	F0	F4095	IUCVBLOK	IUCVMB	IUSAVE	IUSCCODE	IUSCPENT	IUSFLAG2	IUSIUCV	IUSMSGBK
	IUSPATH	LOCK	MSGAALEN	MSGAANAX	MSGAANPX	MSGAATOT	MSGALIST	MSGANSAD	MSGANSLN	MSGAPPC
	MSGAPRMD	MSGARLST	MSGARPAX	MSGARPLE	MSGARPPX	MSGATINV	MSGATTRN	MSGAUDIT	MSGAUDT1	MSGAUDT2
	MSGAUDT3	MSGBLOK	MSGCTLS	MSGERROR	MSGFLAGS	MSGFLAG2	MSGFLAG3	MSGFPNT	MSGID	MSGKEY
	MSGMASK1	MSGPRM	MSGPRMD	MSGPRTY	MSGPURGE	MSGSCCLS	MSGSCPID	MSGSNDOP	MSGTAG	MSGTGCLS
	MSGTGPID	MSGWHTRC	PDAPPC	PDAPPCFL	PDCNTRL	PDENT	PDFLAGS	PDFLAGS2	PDLRCINV	PDLRECL
	PDPRMD	PDSEVERD	PDTGIUCV	PSA	RCADDRCK	RCANSBND	RCAPPC	RCIINVLN	RCMSGLEN	RCNEGLEN
	RCNODATA	RCNOPATH	RCNOPRMD	RCPRMLST	RCPROTCK	RCPURGED	RCRECVSH	RCSNDLST	RCTOTLEN	RCTRUNC
	RECQMSG	RPYQMSG	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS
	SAVER0	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9
	TIMEDISP	VMBLOK	VMSIZE	XRIGHT16	ZEROES					
DMKIUN	ADSPCH	AFREE	AFREEP	AFRET	ASYSIUCV	CCMASK	CCTCLCC	CCTCLPC	CCTCLPQ	CCTCLPR
	CCTCLPS	CCTFLAG1	CCTFLAG2	CCTFLAG3	CCTFLCNT	CCTICTRL	CCTMSGCT	CCTRPYHD	CCTRPYN	CCTRPYP
	CCTSNDHD	CCTSNDN	CCTSNDP	CCTSNDPR	CCTSNDTL	CPEXADD	CPEXBLOK	CPEXR0	CPEXR11	CPEXR12
	CPEXSIZE	DMKDSPCH	DMKFREE	DMKFREEP	DMKFRET	DMKIUAPD	DMKIUAQU	DMKIUBTB	DMKSTKCP	F0
	F4	IETYPMNP	IETYPMP	IUCMSG	IUCVBLOK	IUCVCNT	IUCVMB	IUSAVE	IUSCC	IUSCCODE
	IUSCPENT	IUSFLAG2	IUSIUCV	IUSIUCV2	IUSMASK	IUSMSGBK	IUSPAGE1	IUSPATH	LOCK	MICBLOK
	MICPEND	MICVPSW	MSGALIST	MSGANSAD	MSGANSLN	MSGAPPC	MSGBLIST	MSGBLOK	MSGCTLS	MSGCTLT
	MSGDESC	MSGFLAGS	MSGFLAG2	MSGFLAG3	MSGFPNT	MSGID	MSGKEY	MSGMASK1	MSGNOFL	MSGNORPY
	MSGPRM	MSGPRMD	MSGPRTY	MSGPURGE	MSGSCCLS	MSGSCPID	MSGSIZE	MSGSNDAD	MSGSNDLN	MSGSNDOP
	MSGTAG	MSGTGCLS	MSGTGPID	MSGUSED	PDAPMSGP	PDAPPC	PDAPPCFL	PDCNTRL	PDENT	PDFCNCD
	PDFLAGS	PDMSGCT	PDMSGLIM	PDPRMD	PDPRTY	PDSEND	PDSTATE	PDSTRECV	PDTGIUCV	PDTGPID
	PREFIXA	PSA	RCANSBND	RCAPPC	RCBUFBND	RCMSGCT	RCMSGLEN	RCNOPATH	RCNOPRMD	RCNOPRTY
	RCNOSEND	RCPRMLST	RECQMSG	R0	R1	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	SAVEAREA	SAVEREGS	SAVEWRK1
	SAVEWRK2	SAVEWRK6	SAVEWRK8	SAVEWRK9	SENDQMSG	VMBLOK	WAIT			
DMKIUP	ADSPCH	AFREE	AFREEP	AFRET	ASYSIUCV	ASYSVM	CCMASK	CCTCLCC	CCTCLPC	CCTCLPQ
	CCTCLPR	CCTCLPS	CCTFLAG1	CCTFLAG2	CCTFLAG3	CCTICTRL	CCTMXPID	CCTPDSEG	CCTPNDCT	CCTRPYN
	CCTRPYP	CCTSNDN	CCTSNDP	CC2	CC3	CPEXADD	CPEXBLOK	CPEXMISC	CPEXR0	CPEXR10
	CPEXSIZE	DMKDSPCH	DMKFREE	DMKFREEP	DMKFRET	DMKIUACU	DMKIUAPD	DMKIUBTB	DMKIUNIN	DMKSTKCP
	DMKUDRFU	DMKUDRMD	DMKUDRRV	IUCBFAD1	IUCBFAD2	IUCBFLN1	IUCBFLN2	IUCCBFA1	IUCDWRD	IUCMSG
	IUCMXCN	IUCPNDHD	IUCPNDTL	IUCSIZE	IUCVBLOK	IUCVCPEX	IUCVMB	IUSAVE	IUSBUFF	IUSCCODE
	IUSCPENT	IUSFLAG2	IUSIPSIZ	IUSIUCV	IUSIUCV2	IUSMXCN	IUSPATH	LOCK	MSGBLOK	MSGFLAG3
	MSGSIZE	MSGUSED	PDAPPC	PDAPPCFL	PDAVAIL	PDCNTRL	PDENT	PDFCNCD	PDFLAGS	PDSEND
	PDSEVERD	PDTGIUCV	PDTGPID	PREFIXA	PREFIXB	PSA	RCAPPC	RCHASBFR	RCNOPATH	R0
	R1	R10	R11	R12	R13	R14	R15	R2	R3	R4
	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER0	SAVER1	SAVEWRK1
	SAVEWRK2	SAVEWRK4	SAVEWRK8	SAVEWRK9	SEVER	VMBLOK	XPAGNUM	ZEROES		
DMKIUS	ADSPCH	AFREE	AFREEP	AFRET	BALRSAVE	CCMASK	CCTFLAG2	CCTFLCNT	CCTMSGCT	CCTPDSEG
	CCTPNDSN	CCTRCVHD	CCTRCVTL	CCTSNDHD	CCTSNDPR	CCTSNDTL	CC0	CC1	CC2	CC3

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)								
	CONEXT	CONLENTH	CONMSGBK	CPEXADD	CPEXBLOK	CPEXMISC	CPEXR0	CPEXR10	CPEXSIZE	DMKCVTAB
	DMKDSPCH	DMKFREE	DMKFREEP	DMKFRET	DMKIUAPD	DMKIUAQU	DMKIUERC	DMKIULRP	DMKIUNIN	DMKSCHST
	FFS	F0	F1	IUCMSG	IUCVBLOK	IUCVCPEX	IUCVMB	IUSAVE	IUSCPENT	IUSFLAG2
	IUSMSGBK	IUSPATH	LOCK	MSGANSAD	MSGANSLN	MSGAPPC	MSGARPLE	MSGAUDIT	MSGAUDT1	MSGBLOK
	MSGCTLS	MSGCTLT	MSGDESC	MSGERROR	MSGFLAGS	MSGFLAG2	MSGFLAG3	MSGFPNT	MSGKEY	MSGNOFL
	MSGNORPY	MSGPARTL	MSGPRMD	MSGPRTY	MSGPURGE	MSGSCPID	MSGSIZE	MSGSNDAD	MSGSNDLN	MSGSNDOP
	MSGTGPID	MSGUSED	MSGWHTRC	PDALLOCD	PDAPLOCK	PDAPMSGP	PDAPPC	PDAPPCFL	PDAPPEND	PDAPRECQ
	PDAPREQS	PDAPSNDQ	PDAPSYCF	PDCNTRL	PDENT	PDFLAGS	PDFLAGS2	PDLRCINV	PDLRECL	PDMSGCT
	PDPEND1	PDPEND2	PDSEVERD	PDSTALLC	PDSTATE	PDSTCONF	PDSTCONN	PDSTRECV	PDSTSEND	PDSTSEVR
	PDTGIUCV	PDTGPID	PSA	RCANSBND	RCBUFBND	RCCOMSRV	RCINVCNF	RCINVCON	RCINVREC	RCINVSEV
	RCINVSND	RCMSGCT	RCMSGLEN	RCNONAPP	RCNOPATH	RCSNDOP	RCSYCLVL	RCTRUNC	RECQMSG	RPYQMSG
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER0	SAVEWRK1
	SAVEWRK2	SAVEWRK9	SENDQMSG	TEMPR2	TEMPSAVE	TRQBIRA	TRQBLOK	TRQBSIZE	TRQBUSER	TRQBVAL
	TRQREGS	TRQREGSZ	VMBLOK	VMCOMSRV	VMESTAT	VMEXTCM	VMIDLE	VMISTAT	VMIUCV	VMIUCVWT
	VMIUSTAT	VMNOFLU	VMPSW	VMRSTAT	WAIT	XRIGHT16	ZEROES			
DMKJRL	ACNTBLOK	ACNTSIZE	AFREE	AFRET	ALARM	APSTAT1	APUOPER	AQCNWT	ASYSOP	ASYSVM
	BLANK	BLANKS	CL	CLASSPEC	CLASTERM	DFRET	DMKACOQU	DMKCVTAB	DMKCVTBD	DMKCVTBH
	DMKCVTDT	DMKERMSG	DMKFREE	DMKFRET	DMKGRTTI	DMKLOKSW	DMKQCNWT	DMKSCHST	DMKSCNAU	DMKSCNFD
	DMKSCNRD	DMKSYSJR	FFS	F0	F4	F8	LOCK	NORET	PSA	PWDALOG
	PWDCHAIN	PWDDATE	PWDFLAGS	PWDIBLOK	PWDINVCT	PWDITRQ	PWDLGCNT	PWDLOG	PWDLUNAM	PWDSIZE
	PWDTERMA	PWDUSRID	PWDVSMNM	RDEVADD	RDEVBLOK	RDEVPS	RDEVSNA	RDEVSNRB	RDEVSTA3	RDEVTYPC
	RDEVTYPE	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER0
	SAVER2	SAVEWRK1	SNARBLOK	SNARLUN	SNARVMB	TIMEDISP	TRQBIRA	TRQBLOK	TRQBSIZE	TRQBUSER
	TRQBVAL	TYPBSC	VMBLOK	ZEROES						
DMKLD00	AFREE	AFRET	APOINT	BLANKS	CAW	CC	CLEAR	CONTINUE	CSW	CTL
	C2	DE	DEVICE	DMKCPE	DMKPSA	DMKWRM	EIGHT	FSIZE	IOMASK	IPLPSW
	LASTMSG	PSW	READ	R0	R1	R10	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SILI	UC
	UE	WRITE								
DMKLNK	ACIADDR	ACICLR1	ACICODE	ACIFCN	ACILINK	ACIMODE	ACIPARMS	ACIRGRP	ACIRUSR	ACISIZE
	ACITGRP	ACIUDIR	ACIUSRID	AFREE	AFRET	APTRAN	APTRLK	ARIODC	ARIODV	ASYSVM
	BLANKS	BRING	CDDED	CDVADD	CLASDASD	CLASFBA	CPEXBLOK	CPEXR1	C1	DEFER
	DMKCFQRD	DMKCVTBH	DMKCVTHB	DMKEPSWD	DMKFREE	DMKFRET	DMKJRLSL	DMKLNMSG	DMKLOCK	DMKLOCKD
	DMKLOCKQ	DMKPMACD	DMKPMADT	DMKPTRAN	DMKPTRUL	DMKRPIRA	DMKSCNAU	DMKSCNFD	DMKSCNRD	DMKSCNVS
	DMKSCNVU	DMKSCOLI	DMKSSSLN	DMKSSSMQ	DMKSTKCP	DMKSYSJR	DMKUDRFD	DMKUDRFU	DMKUDRRV	DMKVDSLK
	FTR2311B	FTR2311T	FTR35MB	FTR70MB	F0	F1	F2	F4	F4095	F7
	F8	JPSCBLOK	JPSFLAGS	JPSLNKDS	LINKJRL	LINKJRLI	LOCK	L1	MASKLINK	MSSPRES
	PSA	PSAMSS	RC32	RDCBLKMX	RDCBLOK	RDEVBLOK	RDEVCUP	RDEVDED	RDEVDISA	RDEVFLAG
	RDEVFTR	RDEVMDL	RDEVRDC	RDEVSEL	RDEVSER	RDEVSIZE	RDEVSTAT	RDEVSTA5	RDEVTYPC	RDEVTYPE
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVERETN	SAVER0
	SAVER1	SAVER10	SAVER11	SAVER12	SAVER13	SAVER2	SAVER3	SAVER6	SAVER7	SAVER8
	SAVESIZE	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9
	SYSTEM	SYSVIRT	TYPFBA	TYP2311	TYP2314	TYP3330	VIRTUAL	VMBLOK	WRITE	ZEROES
DMKLNM	AFREE	AFRET	AP	AQCNWT	BLANKS	DMKCVTBD	DMKCVTBH	DMKERMSG	DMKFREE	DMKFRET
	DMKJRLIL	DMKLOCKD	DMKQCNWT	DMKSCNRD	DMKSCNVN	DMKSTKCP	DMKSYSJR	FFS	F1	F2
	F3	F4	F5	F6	JPSCBLOK	JPSFLAGS	JPSLNKDS	LINKJRLI	LOCK	MP
	NORET	PSA	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	SAVER0	SAVER1	SAVER13	SAVER2	SAVESIZE	SAVEWRK1	SAVEWRK2	SAVEWRK4	SAVEWRK5	SAVEWRK6
	SAVEWRK7	SAVEWRK8	SAVEWRK9	VMBLOK						
DMKLOC	ADSPCH	AFREE	AFREEP	AFRET	AP	ASYSLC	CPEXADD	CPEXBLOK	CPEXFPNT	CPEXPROC
	CPEXREGS	CPEXSIZE	DMKDSPCH	DMKFREE	DMKFREEP	DMKFRET	DMKSTKMP	DMKSTKOP	DMKSYSLB	DMKSYSLR
	LOCK	LPUADDR	MP	PSA	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	SAVERETN	SAVEWRK1	SCAN	SYSLOCS				
DMKLOG	ACICODE	ACIFCN	ACILOG	ACINPMT	ACIPARMS	ACIPASS	ACIRGRP	ACIRUSR	ACISIZE	AFREE
	AFRET	AP	APSTAT1	APUOPER	BLANK	BLANKS	BUFSIZE	CLASGRAF	CLASSPEC	CLASTERM
	CPIEXLOG	CPSTAT4	DMKCFGII	DMKCVTBH	DMKEPSWD	DMKERMCP	DMKERMSG	DMKFREE	DMKFRET	DMKJRLLO
	DMKLOHRC	DMKLOJEP	DMKLOKSW	DMKRPWEP	DMKSCNFD	DMKSYSJR	DMKSYSSP	DMKSYSSZ	DMKUDRFU	DMKUDRMD
	DMKUDRRV	F1	F2	F255	F4	F5	F6	F7	F8	INHIBIT
	JPSCBLOK	JPSFLAGS	JPSLOGDS	LOCK	LOGONJRL	MASKLOG	MP	NICBLOK	NICFLAG	NICPSUP
	NICSIZE	NOTIME	PSA	RDEVATSW	RDEVBLOK	RDEVFLAG	RDEVNICL	RDEVPSUP	RDEVSNA	RDEVSNRB
	RDEVSTA3	RDEVTYPC	RDEVTYPE	R0	R1	R10	R11	R12	R13	R14
	R15	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA
	SAVEREGS	SAVERETN	SAVER10	SAVER11	SAVER2	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK6
	SAVEWRK8	SNARBLOK	SNARDIS	SNARFG1	SSBENTRL	SSBHEADL	TIMEDISP	TYPBSC	VMBLOK	
DMKLOH	AFREE	AFRET	ALOKSP	AP	APSTAT1	APUOPER	ASLOGON	ASYSOP	ASYSVM	AVMREAL
	BLANKS	BUFCNT	BUFFER	BUFIN	BUFINLTH	BUFNXT	BUFSIZE	CLASSPEC	CLASTERM	CPDASAAV
	CPDASAON	CPEXSIZE	CPIEXLOG	CPSTAT2	CPSTAT3	CPSTAT4	CPUID	CP370EAV	CP370EON	DMKBLDEC
	DMKBLDRT	DMKCFSAS	DMKCFYPF	DMKCVTBH	DMKERMSG	DMKFREE	DMKFRERC	DMKFRET	DMKHVFCR	DMKLOCKD
	DMKLOCKQ	DMKLOCRD	DMKLOCRQ	DMKLOKIO	DMKLOKSW	DMKPGTMV	DMKPTRUL	DMKQCOSY	DMKSCHRT	DMKSCNAU
	DMKSCNRD	DMKSCNRN	DMKSYSPC	DMKTRQ80	DMKUDRFU	DMKUDRMD	DMKUDRRV	DMKUDRXI	DMKVSDFH	D8
	FFS	F0	F255	F3	F4	F4096	F6	IOBLOK	IOBUSER	IXBLOK
	IXR11	LANGBLOK	LANGLANG	LANGNEXT	LOCK	LOCKSAV	LPUADDR	LPUADDRX	L8	MP
	NEWPAGES	NEWSEGS	NICATRB	NICBLOK	NICSIZE	NICSTAT	NICTRQ	NICUSER	PREFIXA	PREFIXB
	PROCIPL	PROTBLOK	PROTSIZE	PSA	PSALANG	RDEVAIOB	RDEVAIRA	RDEVBLOK	RDEVNICL	RDEVPS
	RDEVSNA	RDEVSNRB	RDEVSTA3	RDEVTFLG	RDEVTRQ	RDEVTYPC	RDEVTYPE	RDEVUSER	RUNUSER	R0
	R1	R10	R11	R12	R13	R14	R15	R2	R3	R4
	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER0	SAVER11	SAVER2
	SAVER3	SAVER5	SAVER6	SAVER7	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6
	SAVEWRK7	SAVEWRK8	SNARBLOK	SNARLUN	SNAROUT	SNARVMB	STACKVM	SYSDFLT	SYSOPER	TIMEDISP
	TRQBIRA	TRQBLOK	TRQBSIZE	TRQBUSER	TYPBSC	VMBLOK	VMSIZE	VRALOC	VRSVUID	WAIT
	XRIGHT16	ZEROES								
DMKLOJ	ADSPCH	AFREE	AFREEP	AFRET	ALOKSP	AP	APSTAT1	APUOPER	ARIODC	ARIODV
	ASYSLC	ASYSVM	AVMREAL	BLANK	BLANKS	CLASDASD	CLASTERM	CPASTAVL	CPASTON	CPEXADD
	CPEXBLOK	CPEXR0	CPEXR11	CPEXR12	CPEXR2	CPEXR8	CPEXSIZE	CPMICAVL	CPMICON	CPSPMODE
	CPSTAT2	CPSTAT4	DMKACON	DMKCVTBH	DMKDMPRY	DMKDSPCH	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET
	DMKLNKSB	DMKLOKIO	DMKLOKSW	DMKLOMSG	DMKNEAVR	DMKSCHRT	DMKSCNRU	DMKSCNVD	DMKSCNVN	DMKSCNVS
	DMKSCNVU	DMKSSSL1	DMKSSSL2	DMKSSSL3	DMKSSSMQ	DMKSTKCP	DMKSYSMU	DMKSYSNM	DMKUDRRD	DMKUDRRV
	DMKVDSAT	DMKVDSDF	DMKVDTPG	DMKVRROP	FFS	F0	F1	F3	F4	F7
	F8	LOCK	LOCKSAV	LPUADDR	MICBLOK	MICCREG	MICCREG0	MICDASA	MICEVMA	MICEVMA2
	MICEVMA3	MICFSSE	MICPTLB	MICRSEG	MICSIO	MICSIZE	MICSKYMD	MICSTBVR	MICSTSM2	MICVPSW
	MICVTMR	MICWORK	MP	MSSPRES	PAGPGSWP	PREFIXA	PREFIXB	PSA	PSAEVMA	PSAMSS
	RDEVADD	RDEVATSW	RDEVBLOK	RDEVBPAG	RDEVCUP	RDEVDED	RDEVDISA	RDEVFLAG	RDEVFTR	RDEVOWN
	RDEVPPAG	RDEVPRDV	RDEVSER	RDEVSIZE	RDEVSTAT	RDEVSTA3	RDEVSTA5	RDEVSYS	RDEVTYPC	RDEVVMNT
	RDEV333V	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER10	SAVER11
	SAVER2	SAVER5	SAVEWRK1	SAVEWRK5	SAVEWRK8	SAVEWRK9	SIGSENSE	SIGSTART	SYSLOCS	SYSVIRT
	TEMPSAVE	TIMEDISP	TRQBSIZE	TYP1052	TYP2305	TYP3330	VIRTUAL	VMBLOK	VMGENIO	XKEYMODE

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)								
	ZEROES								
DMKLOK	ADSPCH	AFREEP	ALOKSY	AP	APSTAT1	APSTAT3	APSTAT4	APSTAT5	APUOPER	ASYSVM
	CODE	CPCREG0	CPEXADD	CPEXBLOK	CPEXPROC	CPEXREGS	CPEXR1	CPEXR11	CPEXR14	CPEXSIZE
	CPLOKFL	CPMFAWIA	CPSYSLK	CPUID	C0	DMKCVTAB	DMKDSPCH	DMKDSPRU	DMKFREEP	DMKPXA
	DMKPXBND	DMKSTKSW	DMKVMASW	EMSMASK	F1	LASTUSER	LOCK	LOKSAVE	LOKSAV2	LPUADDR
	MFAMASK	MP	OFF	PREFIXA	PREFIXB	PSA	R0	R1	R10	R11
	R12	R14	R15	R2	R3	R4	R5	R6	R9	SVMNOUPD
	SVMSTAY	SVMUNLOK	TIMEDISP	TRACCURR	TRACEND	TRACFLG2	TRACPROC	TRACSTRT	TRACSVCR	TRAC12
	TRCLOK	VMBLOK	XCDISP	XCPEND	XTNDLOCK					
DMKLOM	ADSPCH	AFRET	AP	APSTAT1	APUOPER	AQCNWT	ASYSLC	BLANK	BLANKS	CLASSPEC
	CLASTERM	DMKCQHFI	DMKCVTAB	DMKCVTBD	DMKCVTBH	DMKCVTDT	DMKDSPCH	DMKERMSG	DMKFRET	DMKLOKSW
	DMKQCNWT	DMKSCNRD	DMKSCNRN	DMKSCNRU	DMKSCNVU	DMKSTKCP	DMKSYSCK	DMKSYSDT	DMKSYSDW	DMKSYSLG
	DMKSYSLW	DMKSYSMU	DMKSYSNM	DMKSYSTI	DMKSYSTM	DMKUDRFU	DMKVDSAT	F1	F3	F4
	LOCK	LOGMBLOK	LOGMLEN	LOGMNXT	LOGMTXT	MP	NORET	OPERATOR	PSA	RDEVBLOK
	RDEVFLAG	RDEVSNA	RDEVSNRB	RDEVTYPC	RDEVTYPE	RDEV333V	R0	R1	R10	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	SAVEAREA	SAVEREGS	SAVER2	SAVEWRK1	SAVEWRK5	SAVEWRK7	SAVEWRK8	SAVEWRK9
	SCRCMDR	SNARBLOK	SNARLUN	SYSLOCS	TIMEDISP	TYPBSC	VMBLOK			
DMKMCC	ACORETBL	AFREE	AFREEP	AFRET	ALOKSP	AP	APSTAT1	APTRAN	APTRLK	APUOPER
	AQCNWT	ASYSVM	AUTGO	AVMREAL	BLANKS	BRING	CC	CLASTAPE	CORCP	CORFLAG
	CORTABLE	CPCREG8	CPEXADD	CPEXBLOK	CPEXR0	CPEXR11	CPEXR12	CPEXSIZE	C1	C8
	DASDCL	DEFER	DMKCVTDB	DMKCVTHB	DMKDSPNP	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKLOKIO
	DMKMCDIN	DMKMCDLI	DMKMCDSE	DMKMCDST	DMKMCDTI	DMKMIACC	DMKMIADL	DMKMIAMU	DMKMIARO	DMKMNIDK
	DMKMNIST	DMKMNITH	DMKMNJGT	DMKMNJSP	DMKMNL	DMKMNLFI	DMKMNLIN	DMKMONMI	DMKPGUVG	DMKPRGC8
	DMKPRGMC	DMKPRGMI	DMKPTRAN	DMKQCNWT	DMKRSPMN	DMKSCHRT	DMKSCNFD	DMKSCNRU	DMKSTKCP	DMKSYSAT
	DMKSYSBF	DMKSYSEN	DMKSYSRC	DMKSYSRM	DMKSYSRV	FFS	F0	F1	F2	F3
	F4	F5	F8	HEADER	IOBCAW	IOBLOK	IOBMISC	IOBSIZE	IOBUSER	IOERETN
	LOCK	LPUADDR	MNBHDLEN	MNDEVLST	MNDVMORE	MNDVSIZE	MONAIOB	MONARDB	MONATRB	MONBUFAC
	MONBUFAV	MONBUF1	MONCHPTR	MONCOM	MONCRSLT	MONCURBF	MONDVLST	MONDVNUM	MONFLAG1	MONFLAG3
	MONIOBF	MONIOSLT	MONNEXT	MONSIZE	MONUSER	MONUTRB	MON1BUF	MP	MPFEAT	MPGEND
	NORET	PAGECUR	PAGEND	PERFCL	PREFIXA	PREFIXB	PSA	RDEVBLOK	RDEVDED	RDEVDISA
	RDEVFLAG	RDEVPEND	RDEVSTAT	RDEVSYS	RDEVTYPC	RDEVTYPE	RDEVUSER	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER12	SAVEWRK1	SAVEWRK3	SAVEWRK5	SCHEDCL
	SILI	SPOOLED	SPROFCL	SYSTEM	TRACCURR	TRACEFLG	TRACSTRT	TRQBSIZE	TYP3422	TYP3430
	TYP3480	TYP8809	USERCL	VMBLOK	ZERO	ZEROES				
DMKMCD	ADSPCH	AFREE	AFREEP	AFRET	AP	APSTAT1	APUOPER	AQCNWT	AUTGO	AUTOSPL
	BLANK	BLANKS	CFSTOP	CONTINUE	CPCREG8	CPEXADD	CPEXBLOK	CPEXREGS	CPEXSIZE	C8
	DEFINTVL	DMKCVTBH	DMKCVTDB	DMKCVTHB	DMKDSPCH	DMKENTSK	DMKENTUT	DMKERMSG	DMKFREE	DMKFREEP
	DMKFRET	DMKMIAMU	DMKMNISH	DMKMNIST	DMKPRGC8	DMKPRGMC	DMKPRGTI	DMKQCNWT	DMKSCHRT	DMKSCNFD
	DMKSCNRU	DMKSTKOP	DMKSYSAT	DMKSYSMX	DMKSYSTE	DMKSYSTS	ERROR	F0	F1	F2
	F3	F4	F5	F60	F7	F8	F9	LOCK	MONCHPTR	MONCOM
	MONFLAG1	MONFLAG3	MONIOBF	MONSLMT	MONUSER	MONUTRB	MP	NORET	PREFIXB	PSA
	RANGE	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER12
	SAVEWRK1	SAVEWRK3	SAVEWRK4	SAVEWRK6	SPOOLED	TODATE	TRACEFLG	TRQBSIZE	VMBLOK	X40FFS
	ZERO	ZEROES								
DMKMCH	ACORETBL	ACRLOCK	ADSPCH	AEXTSP	AFREE	AFREEP	AFRET	ALARM	ALOKSY	ALOKVM
	AMCHAREA	AP	APSTAT1	APSTAT2	APSTAT4	APUOPER	ASYSVM	AVMREAL	BLANK	CCS4DGRD
	CCS4HARD	CLEAR	CODE	CONTINUE	CORCFLCK	CORDISA	CORFLAG	CORIOLCK	CORPGPNT	CORSWPNT

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	CORTABLE	CORVM	CPCREG0	CPCREG8	CPEXADD	CPEXBLOK	CPEXREGS	CPEXSIZE	CPID	CPMCHLK
	CPMCHSE	CPPTLBR	CPSPMODE	CPSTATUS	CPSTAT2	CPSTAT4	CPUID	CPUMODEL	CPWAIT	CRBIT
	C0	C1	C13	C14	C7	DMKACRCT	DMKCFMBK	DMKCFPRR	DMKCVTAB	DMKCVTBH
	DMKDSPCH	DMKDSPRU	DMKFREE	DMKFREEP	DMKFRELO	DMKFRET	DMKIOEMC	DMKLOKDF	DMKMCTPT	DMKMCTST
	DMKOPRWT	DMKPGSPO	DMKPMAMC	DMKPMASW	DMKPTRFR	DMKPTTFT	DMKQCNWT	DMKSCHST	DMKSTKOP	DMKVATSI
	DMKVFRIM	DMKVFRNM	EXDCCF	EXDCNO	EXDINSTO	EXDINTTO	EXDRESVD	FFS	F1	F255
	F3	F4096	F6	F7	INTMC	INTRC	IPUADDR	IPUADDRX	LOCK	LOCKSAV
	LPUADDR	L3	L8	MCCPUID	MCFXDLOG	MCHAREA	MCHCPEX	MCHEK	MCHFIX	MCHFLAG0
	MCHFLAG1	MCHFLAG3	MCHFLAG4	MCHFLAG5	MCHFLAG6	MCHFLAG7	MCHFSAR	MCHLEN	MCHLEN1	MCHMODEL
	MCHPDAR1	MCHPDAR6	MCHPDAR7	MCHP1IDE	MCHP1IKE	MCHP1SDE	MCHP1SKE	MCHP1STD	MCHP6CBA	MCHREC
	MCHRESEV	MCHOHDWR	MCHOQUIT	MCHOSFTR	MCHOTERM	MCHOUSAD	MCH1BUFF	MCH1COST	MCH1GERR	MCH1IOTO
	MCH1MAIN	MCH1PROC	MCH1TODC	MCH3BCST	MCH3DATA	MCH3DGRD	MCH3HARD	MCH3INTE	MCH3PASS	MCH3PROT
	MCH3SOFT	MCH3SOLD	MCH4BURE	MCH4REPA	MCH5IFSA	MCH7CHTM	MCH7EXIT	MCH7GSTR	MCH7IOEM	MCH7OPSW
	MCH7PURG	MCH7RSRE	MCH7SMCR	MCH7SUP	MCH7SYST	MCH7TRQ	MCH7VEQR	MCH7VRTM	MCNPSW	MCOLDPW
	MCOPSW	MCPROGID	MCREC	MCRECORD	MCRECTYP	MCSWONE	MCSWTWO	MCS1PASS	MCS2HARD	MICBLOK
	MICVTMR	MODEL135	MODEL145	MODEL155	MODEL165	MOD3031	MOD3033	MOD3081	MOD3090	MOD4331
	MOD4381	MP	NORET	OFF	OPERATOR	PAGCORE	PAGINVAL	PAGREF	PMAMODE	PMAON
	PMASTAT	PREFIXA	PREFIXB	PROBMODE	PROCIO	PROCIPL	PSA	QUANTUMR	RECMODE	RECOVRPT
	RUNUSER	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SIGSTART	SIGSTOP	SWPCHG1
	SWPCHG2	SWPFLAG	SWPKEY1	SWPKEY2	TIMEDISP	TIMER	TRACCURR	TRACEND	TRACFLG1	TRACPROC
	TRACSTRT	TRACSVCR	TRAC04	TRANMODE	TRCMCH	TRQBIRA	TRQBLOK	TRQBSIZE	TRQBTOD	TRQBUSER
	TRQBVAL	VECAVAIL	VECF	VECOPVF	VECSTAT	VECUSER	VMBLOK	WAITEND	XKEYMODE	XPAGNUM
	Y0	Y2	Y4	Y6	ZERO	ZEROES				
DMKMCI	ADSPCH	AFREEP	AMCHAREA	AP	APSTAT1	APSTAT4	APUOPER	BLANKS	CPEXADD	CPEXBLOK
	CPEXREGS	CPEXSIZE	CPUVERSN	C14	DMKCVTDB	DMKDSPCH	DMKERMSG	DMKFREEP	DMKMCHLM	DMKQCNWT
	DMKSCNFD	DMKSTKOP	F2	F8	IPUADDR	IPUADDRX	LOCK	MCHAREA	MCHMODEL	MESSAGE
	MP	NORET	OPERATOR	PSA	RECMODE	RECOVRPT	R0	R1	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R9	SAVEAREA
	SAVEREGS	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK6	SAVEWRK7	VMBLOK			
DMKMCT	ACRLOCK	ADSPCH	AEXTSP	AFREEP	ALARM	ALOKDS	ALOKSP	ALOKSY	APSTAT1	APSTAT4
	APUOPER	AQCNWT	ASYSVM	AVMREAL	BLANK	CLEAR	CONTROL	CPAPRINP	CPAPRPND	CPEXADD
	CPEXBLOK	CPEXFPNT	CPEXPROC	CPEXREGS	CPEXSIZE	CPID	CPSTATUS	CPSTAT5	CPSUPER	CPTERMLK
	CPWAIT	CSADDR	CSSFEAT	C2	DMKACRCO	DMKACSCV	DMKCFMBK	DMKCFPRR	DMKCPPUP	DMKCVTBH
	DMKDMPAA	DMKDMPMA	DMKDMPSA	DMKDSPCH	DMKDSPRQ	DMKFREEP	DMKLOKDF	DMKLOKDS	DMKLOKFR	DMKLOKIO
	DMKLOKPS	DMKLOKRL	DMKLOKRM	DMKLOKSY	DMKLOKTR	DMKOPRWT	DMKPGSPO	DMKPMASW	DMKQCNWT	DMKRIOCN
	DMKRIODV	DMKSCNEP	DMKSTKMP	DUMPSAVE	EMSPEND	EMSPQUI	EXNPSW	FFS	F255	F3
	IPUADDR	IPUADDRX	LOCK	LPUADDR	LPUADDRX	MCHAREA	MCHFLAG1	MCHFLAG3	MCHFLAG7	MCH1TODC
	MCH3BCST	MCH7OPSW	MFASAVE	MPUOPER	NORET	NOTERM	OFF	OPERATOR	PMAMODE	PMAON
	PMASTAT	POFFLINE	PREFIXA	PREFIXB	PRIORITY	PROCIO	PSA	RDIDX	RESET	RSRTNPSW
	RUNUSER	R0	R1	R10	R11	R12	R14	R15	R2	R3
	R4	R6	R7	R8	R9	SIGAPR	SIGREST	SIGSSS	SIGSTOP	SIGXC
	SVCNPSW	TIMEDISP	VMBLOK	XCPEND	ZEROES					
DMKMHC	ADSPCH	AEXTSP	AFREE	AFREEP	AFRET	ALOKSY	AMCHAREA	APSTAT1	APUOPER	ASYSVM
	AVMREAL	CODE	CPEXADD	CPEXBLOK	CPEXREGS	CPEXR11	CPEXSIZE	CPSPMODE	CPSTAT2	CPUID
	C2	DMKDMPC2	DMKDSPCH	DMKDSPRU	DMKFREE	DMKFREEP	DMKFRET	DMKLOKDF	DMKPTRUL	DMKSCHST
	DMKSTKCP	DMKSTKMP	DMKSTKOP	EXTMODE	F1	F8	HCBLOK	HCBPNT	HCFLAG	HCFPNT
	HCIRTN	HCMSFACT	HCMSFCW	HCMSFDB	HCMSFDBV	HCMSFSYS	HCREG13	HCRSCP	HCSIZE	HCVMBLK
	HCVMREQ	HCVRX	INMSFBLK	LOCK	LPUADDR	MCHAREA	MCHMODEL	MOD3081	MSFBLOK	MSFDATA
	MSFRSP	MSFR01	PMAMODE	PMASTAT	PREFIXA	PROCIO	PSA	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R8	R9	SAVEAREA	SAVEREGS	SAVER1	SAVER2	SCCBIBLK	SCCBIFE	SCCBISIZ	SCCBKMAX

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	SCCBLOK	SCCBMXID	SCCBRESP	SCCBRMAX	SCCB0010	SIGSENSE	SIGSTART	SIGSTOP	TEMPR1	TEMPR2
	TRACCURR	TRACEND	TRACFLG3	TRACPROC	TRACSTRT	TRACSVCR	TRAC17	TRCMSSF	TRQBIRA	TRQBLOK
	TRQBSIZE	TRQBTOD	TRQBUSER	TRQBVAL	VMBLOK	VMSIZE	ZEROES			
DMKMHV	ACORETBL	AFREE	AMCHAREA	APTRAN	APTRLK	AVMREAL	BRING	CMDBEG	CORFLAG	CORSHARE
	CORTABLE	C1	DEFER	DMKFREE	DMKMHCLK	DMKMHCVM	DMKPTRAN	DMKPTRLK	DMKSYSPC	F0
	HCBLOK	HCMSFCW	HCMSFDB	HCMSFDBV	HCSIZE	LOCK	MCHAREA	MCHMODEL	MOD3081	MOD3090
	MOD4381	MSFBLOK	MSFCFLGS	MSFDATA	MSFDB1	MSFDB2	MSFDB3	MSFINFO	MSFLNG	MSFRSP
	MSFR01	MSFR41	MSF00	PSA	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	SAVEWRK8	SCCBCTL	SCCBLEN	SCCBLOK	SCCBMDRT	SCCBMODP	SCCBRESP	SCCBRMAX
	SCCB0010	SCCB0020	SCCB0100	SCCB40F0	SYSCPRD	SYSCPWT	VECAVAIL	VECSTAT	VMBLOK	XPAGNUM
DMKMIA	ADDSFB	AFREE	AFREEP	AFRET	APSTAT1	APTRAN	APTRLK	APUOPER	AQCNWT	ASYSOP
	ASYSVM	AUTOSPL	BLANKS	BRING	BUFCNT	BUFFER	BUFNXT	BUFSIZE	CFSTOP	CLCMD
	CLSUS	CPCREG8	CPEXADD	CPEXBLOK	CPEXR0	CPEXR10	CPEXR11	CPEXR12	CPEXSIZE	C1
	C8	DEFER	DMKCKSPL	DMKCKTSD	DMKCKTUU	DMKCVTBD	DMKCVTDT	DMKDSPNP	DMKENTFI	DMKERMSG
	DMKFREE	DMKFREEP	DMKFRET	DMKLOCRD	DMKLOCRQ	DMKLOKSW	DMKMCCCL	DMKMNIDK	DMKMNISH	DMKMNJDS
	DMKPGTSG	DMKPGUVG	DMKPGUVR	DMKPRGC8	DMKPRGMC	DMKPTRAN	DMKPTRLK	DMKPTRUL	DMKQCNWT	DMKRPAGT
	DMKRPAPT	DMKRSPMN	DMKSCNAU	DMKSCNFD	DMKSPLSP	DMKSTKCP	DMKSYSAT	DMKSYSBF	DMKSYSCL	DMKSYSEN
	DMKSYSUR	DMKVSDAD	DMKVSGRI	EXHAUST	FFS	F0	F1	F2	F3	F4
	F5	IOERETN	IPLPSW	LOCK	MNBHDLEN	MONAIOB	MONBUFAC	MONBUFIO	MONBUF1	MONCOM
	MONCURBF	MONDAS	MONDASA	MONDASB	MONEX	MONFLAG1	MONFLAG2	MONFLAG3	MONIOBF	MONNEXT
	MONSFB	MONSPLCT	MONUSER	MON1BUF	NORET	OPERATOR	OPNSFB	PAGEND	PERFCL	PREFIXA
	PREFIXB	PSA	RDRCHN	R0	R1	R10	R11	R12	R13	R14
	R15	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA
	SAVEREGS	SAVER1	SAVER2	SAVEWRK1	SAVEWRK3	SFBCLAS	SFBCOPY	SFBDATE	SFBDIST	SFBFILID
	SFBFLAG	SFBFLAG2	SFBFLAG4	SFBFNAME	SFBFTYPE	SFBINUSE	SFBLAST	SFBLOK	SFBMON	SFBOFORM
	SFBORIG	SFBPNT	SFBRECNO	SFBSIZE	SFBSTART	SFBSYSID	SFBTIME	SFBTUSE	SFBTYPE	SFBUFORM
	SFBUSER	SPFILID	SPLINK	SPNXTPAG	SPOOLED	SPPREPAG	SPRECNUM	SPTIME	SUSPEND	SYSTEM
	TIMEDISP	TRACEFLG	TRAP	TYP2540P	UNFIN	VMBLOK	ZEROES			
DMKMID	ADSPCH	AFREEP	ALARM	APSTAT1	APUOPER	AQCNWT	ASYSVM	AUTGO	CONTINUE	CPEXADD
	CPEXBLOK	CPEXR0	CPEXR11	CPEXR12	CPEXSIZE	DATE	DMKCVTDT	DMKDMPDT	DMKDMPTD	DMKENTKC
	DMKERMSG	DMKFREEP	DMKLOKSW	DMKMNIST	DMKPMAIO	DMKPRGMC	DMKQCNWT	DMKSCHST	DMKSTKCP	DMKSYSAT
	DMKSYSDW	DMKSYSTE	DMKSYSTI	DMKSYSTS	FFS	F60	LOCK	NORET	NOTRESP	PREFIXA
	PREFIXB	PSA	RESET	R0	R1	R10	R11	R12	R13	R14
	R15	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA
	SAVEREGS	SAVER11	SAVEWRK2	SAVEWRK5	SAVEWRK7	TEMPSAVE	TIMEDISP	TODATE	TRQBLOK	TRQBVAL
	VMBLOK	ZEROES								
DMKMNI	ACORETBL	ADSPCH	AFREE	AFREEP	AFRET	APSTAT1	APTRLK	APUOPER	AQCNWT	ARIOCH
	ARIOCU	ARIODV	ASYSVM	AUTGO	AVMREAL	CC	CFSTOP	CLASDASD	CLASFBA	CLASTAPE
	CLCMD	CLSUS	CORCP	CORFLAG	CORLCNT	CORTABLE	CPCREG8	CPEXADD	CPEXBLOK	CPEXR0
	CPEXR11	CPEXR12	CPEXSIZE	CPSTAT5	CPUID	CRTEXT	CRTEXTSZ	C1	C8	DASDCL
	DEFINTVL	DMKCPEID	DMKCPEND	DMKCPEPP	DMKCVTAB	DMKCVTDT	DMKDSPCH	DMKDSPNP	DMKENTBS	DMKENTEC
	DMKENTES	DMKENTET	DMKENTSC	DMKENTST	DMKENTTB	DMKENTTE	DMKENTTI	DMKENTUT	DMKERMSG	DMKFREE
	DMKFREEP	DMKFREHI	DMKFRELO	DMKFRET	DMKIOSQR	DMKMIACC	DMKMNL	DMKMNLFI	DMKMNLIN	DMKMONPR
	DMKMO000	DMKMO040	DMKPGTPI	DMKPGTSI	DMKPGUVR	DMKPRGC8	DMKPRGMC	DMKPRGMI	DMKPRGMO	DMKPRGTI
	DMKPTRLK	DMKPTRUL	DMKQCNWT	DMKRIOCT	DMKSCHRT	DMKSCHST	DMKSTKCP	DMKSYSAP	DMKSYSAT	DMKSYSNP
	DMKSYSRC	DMKSYSRM	DMKSYSRV	DMKSYSTE	DMKSYSTS	ERROR	FFS	F1	F4095	IOBCAW
	IOBFATAL	IOBFLAG	IOBIOER	IOBIRA	IOBLOK	IOBMISC	IOBMISC2	IOBSIZE	IOBSTAT	IOERSIZE
	LOCK	LPUADDR	LPUADDRX	MNCHSIZE	MNCHSZ	MNCHZMP	MNCLDAST	MNCLPERF	MNCLUSER	MNCODASH
	MNCOTH	MNCOTT	MNCOUSER	MNDEVLEN	MNDEVLMP	MNDEVLST	MNDVCNT	MNDVHDRL	MNDVLEN	MNDVMORE
	MNDVSIZE	MNRDEVB	MN097	MN097APL	MN097CPL	MN097CPP	MN097CPU	MN097CR8	MN097DAT	MN097DPA

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	MN097FSS	MN097LEN	MN097LEV	MN097LSN	MN097MOD	MN097NUC	MN097PCP	MN097PCS	MN097PSS	MN097TIM
	MN097TTS	MN097UID	MN097VR	MN098	MN098LEN	MN098UID	MN600ADD	MN600CNT	MN600DEV	MN600DLN
	MN600HDR	MN600HLN	MN600MAX	MN600MXS	MN600NUM	MN600SER	MN600TY	MONAIOB	MONARDB	MONATRB
	MONBUF1	MONBUF1V	MONCHPTR	MONCLASS	MONCODE	MONCOM	MONCURBF	MONDVLST	MONDVNUM	MONEX
	MONFLAG1	MONFLAG2	MONFLAG3	MONIOBF	MONNEXT	MONSIZE	MONSUSCT	MONUSER	MONUTRB	MPGEND
	MPUOPER	NORET	PAGECUR	PAGEND	PAGENXT	PERFCL	PREFIXA	PREFIXB	PROCIO	PROCIPL
	PSA	RCHADD	RCHBLOK	RCHCUTBL	RCUADD	RCUBLOK	RCUDVTBL	RCUPRIME	RCUSUB	RCUTYPE
	RDEVADD	RDEVBLOK	RDEVCUA	RDEVDISA	RDEVFLAG	RDEVIOCT	RDEVSER	RDEVSTAT	RDEVSYS	RDEVTYPC
	RDIDX	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SPOOLED
	SUSPEND	TEMPR1	TEMPR3	TODATE	TRQBIRA	TRQBLOK	TRQBSAVE	TRQBSIZE	TRQBTOD	TRQBUSER
	TRQBVAL	TRUN	USERCL	VMBLOK	XPAGNUM	ZEROES				
DMKMNJ	AFREE	APSTAT1	APTRAN	APTRLK	APUOPER	AQCNWT	ASYSVM	AUTGO	AUTOSPL	BLANKS
	BRING	C1	DEFER	DMKCVTAB	DMKCVTBD	DMKDSPNP	DMKERMSG	DMKFREE	DMKMOOTI	DMKPRGMC
	DMKPRGMO	DMKPRGTI	DMKPTRAN	DMKQCNWT	DMKSCHST	DMKSCNFD	DMKSYSAT	DMKSYSCL	DMKSYSMX	DMKSYSTE
	DMKSYSTS	DMKSYSUR	DMKUDRFU	FFS	F0	F1	F2	F5	F8	LOCK
	MONATRB	MONBUF1	MONCOM	MONSFB	MONSPLCT	NORET	PSA	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R7	R8	SAVEAREA	SAVEREGS	SAVER2	SFBFILID	SFBLOK	SYSTEM	TRQBIRA	TRQBLOK
	TRQBSIZE	TRQBTOD	TRQBUSER	TRQBVAL	VMBLOK					
DMKMNL	ADSPCH	AFREE	AFREEP	AFRET	ARIODC	ARIODV	ASYSVM	CACHBLOK	CACHCPXP	CACHCUU1
	CACHCUU2	CACHDPT	CACHERRC	CACHFCNT	CACHICCW	CACHIN1	CACHIN2	CACHIOBP	CACHISSS	CACHKX
	CACHNTDF	CACHNTIM	CACHNTMI	CACHPDP	CACHRCNT	CACHRDVP	CACHRPT	CACHRSD2	CACHSIZE	CACHSTA1
	CACHSTOD	CACHTRQP	CACHWR1	CACHWR2	CACHWSD1	CACHWSD2	CC	CNTSDATA	CNTSREC	CNTSSIZE
	CNTSTDAT	CNTSTOD	CNTS2DAT	CPEXADD	CPEXBLOK	CPEXMISC	CPEXR0	CPEXR10	CPEXR11	CPEXR13
	CPEXR8	CPEXSIZE	DASDCL	DMKCVTAB	DMKDSPCH	DMKENTGP	DMKENTGQ	DMKENTIM	DMKFREE	DMKFREEP
	DMKFRET	DMKIOSQR	DMKMONPR	DMKPRGC8	DMKPRGMC	DMKPRGTI	DMKSCHST	DMKSCNEP	DMKSCNPH	DMKSCNRA
	DMKSCNRD	DMKSTKCP	F0	F1	F15	F5	IOBCAW	IOBCP	IOBFATAL	IOBFLAG
	IOBIOER	IOBIRA	IOBLINK	IOBLOK	IOBMISC	IOBPATHF	IOBPROC	IOBRADD	IOBSIZE	IOBSTAT
	IOBUSER	IOERBLOK	IOEREXT	IOERSIZE	LOCK	MNCLDAST	MN3880DV	MN38801L	MN605	MN605BD
	MN605BW	MN605DBU	MN605DUO	MN605LN	MN605NPH	MN605NPO	MN605NPR	MN605PMC	MN605PMO	MN605RS1
	MN605RS2	MN605RS3	MN605SIO	MN605SIR	MN605SPH	MN605SPO	MN605SPR	MN605SRH	MONCLASS	MONCODE
	M3880CUU	M3880FLG	M3880M13	M3880NAC	M3880RCU	M3880RLN	M3880SD2	M3880SG	M3880SSC	M3880SSS
	M3880TOD	M3880TOT	PSA	PTRCACH	PTRDVCT	PTRENLN	PTRLIST	PTRNTRY	PTRRDEV	PTRSIZE
	RCHADD	RCHBLOK	RCHPROC	RCUADD	RCUBLOK	RCUCHA	RCUCHB	RCUCHC	RCUCHD	RCUPRIME
	RCUSUB	RCUTYPE	RDEVADD	RDEVAOF	RDEVBLOK	RDEVBOF	RDEVCMDK	RDEVCUA	RDEVCUB	RDEVCUP
	RDEVCU2	RDEVDED	RDEVDISA	RDEVFLAG	RDEVFTR3	RDEVMD13	RDEVOWN	RDEVPPAG	RDEVPTHS	RDEVSIZE
	RDEVSTAT	RDEVSTA5	RDEVSTA6	RDEVSTA7	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	SILI	TRQBIRA	TRQBLOK	TRQBTOD	TRQBUSER	TRQBVAL	TRQCACHL	TRQCACHP
	ZEROES									
DMKMNT	AFREE	ALOKSP	AP	APSTAT1	APUOPER	ARDCBLOK	ARIOCH	ARIOCT	ARIOCU	ARIODV
	ASYSOP	BALR1	BALR3	BALR4	BCTWAIT	BLANKS	BUSY	CAW	CC	CCC
	CLASDASD	CLASFBA	CLASGRAF	CLASSPEC	CLASTAPE	CLASTERM	CLASURO	CODE	CPSTAT5	CPUMODEL
	CPUSER	CPUVERSN	CSW	CUE	C2	DE	DMKALOCP	DMKCPIMS	DMKCPIRP	DMKCPISH
	DMKCPIVP	DMKCPIWT	DMKCPZFD	DMKCPZMG	DMKCPZVR	DMKCVTBH	DMKFREE	DMKIOSMQ	DMKIOTIN	DMKLOKIO
	DMKSCNMI	DMKSCNPH	DMKSCNRD	DMKSCNRU	DMKSCNVS	DMKSYSOC	DMKSYSOW	DMKSYSPR	DMKSYSPU	DMKTODTB
	DMKVRRDD	D4	FF	FFS	FORMBLOK	FORMOPER	FTRFH	FTRRPS	FTRRSRL	FTR35MB
	FTR70MB	F240	F256	F4096	IFCC	INTREQ	INTTIO	IOBMSIZE	IONPSW	IOOPSW
	IPUADDR	IPUADDRX	LOCK	LOCKSAV	LPUADDR	LPUADDRX	L3	L5	MP	MPUOPER
	NORET	OWNDLIST	OWNDVSER	PGIDADDR	PGIDCPU	PGIDMOD	PGIDTOD	PREFIXA	PREFIXB	PROCIPL
	PSA	PSAPGID	RCHADD	RCHBLOK	RCHCUTBL	RCHPROC	RCUADD	RCUBLOK	RCUCACH	RCUCHCNT

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	RCUDISA	RCUDVTBL	RCUPRIME	RCUSIZE	RCUSTAT	RCUSUB	RCUTYPE	RDCBLKAP	RDCBLKCG	RDCBLKFA
	RDCBLKMA	RDCBLKMX	RDCBLKPG	RDCBLOK	RDCCKD	RDCFBA	RDCFLAG	RDCFPNT	RDCLENG	RDCLENGC
	RDCLENGF	RDCPAGAP	RDCPAGCG	RDCPAGFA	RDCPAGMA	RDCPAGXT	RDCRECSZ	RDCSIZE	RDCSTART	RDCTYPE
	RDEVADD	RDEVALT	RDEVAOF	RDEVASGN	RDEVBLOK	RDEVBOF	RDEVCFLT	RDEVCKDX	RDEVCMDK	RDEVCRDC
	RDEVCUA	RDEVCUB	RDEVCUP	RDEVCU11	RDEVDISA	RDEVDUPL	RDEVEXTN	RDEVFLAG	RDEVFTR	RDEVFTR2
	RDEVFTR3	RDEVMDL	RDEVMD00	RDEVMD13	RDEVM000	RDEVM004	RDEVNAME	RDEVNATH	RDEVPPAG	RDEVPRDV
	RDEVPTHS	RDEVRDC	RDEVSER	RDEVSTAT	RDEVSTA2	RDEVSTA3	RDEVSTA5	RDEVSTA6	RDEVSTA7	RDEVTYPC
	RDEVTYPE	RDEVUSER	RDEVVMNT	RDIDX	RSPXBLOK	RSPXDST2	RSPXDST3	RSPXFORM	RSPXSIZE	R0
	R1	R10	R11	R12	R13	R14	R15	R2	R3	R4
	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SILI	SM	SYSIPLDV
	SYSVIRT	TEMPR14	TEMPR15	TEMPR9	TEMPSAVE	TRACCURR	TRACEND	TRACPROC	TRACSTRT	TRACSVCR
	TYPFBA	TYPPUN	TYP2305	TYP3310	TYP3340	TYP3370	TYP3375	TYP3380	TYP3480	TYP3800
	TYP38003	TYP38008	TYP3851	TYP8809	UC	UT3310	UT3370	UT3370M4	VIRTUAL	ZEROES
DMKMON	ADSPCH	AFREEP	AFRET	AP	APSTAT1	APUOPER	ASYSVM	BLANKS	CC	CFSTOP
	CLASFBA	CLCMD	CLSUS	CONADDR	CONCNT	CONFLG2	CONNCB	CONTASK	CPCREG8	CPEXADD
	CPEXBLOK	CPEXRO	CPEXSIZE	CUE	C8	DE	DMKCVTAB	DMKDSPCH	DMKDSPNP	DMKERMSG
	DMKFREEP	DMKFRET	DMKIOSQR	DMKMIAWO	DMKMNIFI	DMKMNITR	DMKPRGC8	DMKPRGMC	DMKSCHAL	DMKSCHMF
	DMKSCHN1	DMKSCHPU	DMKSCHQ1	DMKSCHW1	DMKSCHW2	DMKSTKCP	DMKSYSAT	DMKSYSMX	DMKSYSNM	DMKSYSOW
	ERROR	FFS	F0	F2	F3	F4	F4095	IDLEWAIT	IOBCAW	IOBCSW
	IOBCYL	IOBFATAL	IOBFH	IOBFLAG	IOBIOER	IOBLOK	IOBMISC	IOBMISC2	IOBPAG	IOBSIZE
	IOBSPEC2	IOBSTAT	IOERSIZE	IONTWAIT	IPLPSW	LOCK	MNCLPERF	MNCLSYS	MNCODA	MNCOSUS
	MNDEVLST	MNDVCNT	MNDVLEN	MNHCLASS	MNHCODE	MNHDR	MNHDRLEN	MNHRECSZ	MNHTOD	MNRDEVB
	MN000LEN	MN001LEN	MN099	MN099CNT	MN099LEN	MN099TOD	MN10X	MN10XADD	MN10XLEN	MN10XUID
	MN10YCNT	MN10YIO	MN10YLEN	MN20X	MN20XNPP	MN20XPRC	MN20XQNM	MN20XQ1E	MN20XQ1N	MN20XQ2E
	MN20XQ2N	MN20XSWS	MN20XUID	MN20XWSS	MN20YTTI	MN20YVTI	MN202APR	MN202IOC	MN202LEN	MN202LPR
	MN202PGR	MN202PRI	MN202PST	MN202QDP	MN202QDR	MN202QDT	MN202REF	MN202RES	MN202RV2	MN202SFG
	MN202VMR	MN202VOT	MN202VRL	MN202VVT	MN203LEN	MN204LEN	MN204PRI	MN30X	MN30XCYL	MN30XEPG
	MN30XFLG	MN30XLEN	MN30XSSZ	MN30XUID	MN30XUPG	MN30XVOL	MN30XVPG	MN300	MN300END	MN300LEN
	MN300RSP	MN300SST	MN300TS	MN300UID	MN305	MN305LEN	MN305PN	MN305PO	MN305RSP	MN305SN
	MN305SO	MN305UID	MN305WCT	MN500	MN500INS	MN500LEN	MN500OVH	MN500UID	MN500VAD	MN700
	MN700ADD	MN700CCY	MN700CHR	MN700CYL	MN700DIR	MN700LEN	MN700OPC	MN700PRO	MN700QCH	MN700QCU
	MN700QDV	MN700RS1	MN700RS2	MN700UID	MN802CLN	MN802CNT	MN802CTR	MN802DEV	MN802DLN	MN802NAU
	MN802NPP	MN802NUM	MN802PGR	MN802PGW	MN802PRB	MN802WID	MN802WIO	MN802WPG	MONAIOB	MONARDB
	MONBUFAC	MONBUFAV	MONBUFIO	MONBUF1	MONCLASS	MONCLKSA	MONCLOCK	MONCODE	MONCOM	MONCRSLT
	MONCURBF	MONDVLST	MONEX	MONFLAG1	MONFLAG2	MONFLAG3	MONIOBF	MONIOSLT	MONLSTBK	MONNEXT
	MONPDLY	MONREGS	MONSAVE1	MONSAVE2	MONSLMT	MONSPLCT	MONSUSCK	MONSUSCT	MONTPERR	MONUSER
	MON1BUF	MP	MPUOPER	OWNDLIST	OWNDVSER	PAGECYL	PAGEND	PAGEWAIT	PAGEXSKC	PGREAD
	PGWRITE	PREFIXA	PREFIXB	PROBTIME	PROCIO	PROCIPL	PROPSW	PSA	Q1DROP	RCHADD
	RCHBLOK	RCHPROC	RCHQCNT	RCUADD	RCUBLOK	RCUCHA	RCUPRIME	RCUQCNT	RCUSUB	RCUTYPE
	RDEVADD	RDEVBLOK	RDEVCUA	RDEVCYL	RDEVECKD	RDEVFLAG	RDEVIOCT	RDEVMDL	RDEVQCNT	RDEVSKUP
	RDEVSNA	RDEVTYPC	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS
	SAVER0	SAVER1	SAVER5	SAVEWRK1	SAVEWRK2	SPOOLED	SPROFCL	SSBCODE	SSBCYL	SSBEPGS
	SSBFLAG	SSBFPNT	SSBLOK	SSBNPGS	SSBOLD	SSBPGSP	SSBPSTOR	SSBUPGS	SSBVPAGE	STPDRP
	SUSPEND	TMRDRP	TRACPROC	TRAP	TRUN	TSEND	UC	UE	VMBLOK	ZEROES
DMKMOO	ADSPCH	AEXTSP	AFREE	ALOCALOC	ALOCBLOK	ALOCCHN	ALOCCSR	ALOCCYL1	ALOCFLG	ALOCMALC
	ALOCPAVL	ALOCPG	ALOCPNT	ALOCPP	ALOCPREC	ALOCPUSE	ALOCRDEV	ALOCSW	ALOCTDSK	ALOCUSED
	ALOKFR	ALOKSP	AP	APSTALOC	APSTAT1	APUOPER	ARIODV	ASYSVM	ATMRSN	BLANKS
	CPCREG8	CPSTAT	CPSTAT3	CPSTAT5	CPTIDLE	CPWAIT	DASDCL	DMKCVTAB	DMKDSPCH	DMKDSPCK
	DMKDSPIT	DMKDSPNP	DMKDSPN2	DMKDSPPT	DMKENTIM	DMKENT62	DMKENT63	DMKFRELN	DMKFRELS	DMKFRENP
	DMKFRESC	DMKFRETS	DMKHVCDI	DMKIOSNM	DMKIOTCT	DMKLOKCT	DMKLOKFR	DMKLOKIO	DMKLOKRL	DMKLOKRM
	DMKLOKSY	DMKLOKTR	DMKMIACC	DMKMNLMR	DMKMONPR	DMKPAGCC	DMKPAGPS	DMKPAGSS	DMKPGTDM	DMKPGTPB
	DMKPGTPT	DMKPGTPU	DMKPGTSB	DMKPGTSF	DMKPGTST	DMKPGTSU	DMKPRGCT	DMKPRGMC	DMKPRGMO	DMKPRGTI

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	DMKPRVCD	DMKPRVCE	DMKPRVCH	DMKPRVCP	DMKPRVCS	DMKPRVCT	DMKPRVDI	DMKPRVEK	DMKPRVEP	DMKPRVIK
	DMKPRVIP	DMKPRVLC	DMKPRVLP	DMKPRVLR	DMKPRVMN	DMKPRVMO	DMKPRVMS	DMKPRVNC	DMKPRVPB	DMKPRVPE
	DMKPRVPT	DMKPRVRR	DMKPRVTC	DMKPRVTE	DMKPRVTP	DMKPRVXI	DMKPRVXS	DMKPSANX	DMKPTRCS	DMKPTRFC
	DMKPTRFF	DMKPTRFN	DMKPTRFO	DMKPTRNS	DMKPTRN2	DMKPTRPR	DMKPTRRC	DMKPTRRF	DMKPTRRW	DMKPTRSC
	DMKPTRSS	DMKPTRSW	DMKPTTHL	DMKPXA	DMKPXACX	DMKPXADL	DMKPXAFC	DMKPXAIP	DMKPXAIS	DMKPXAMC
	DMKPXAOP	DMKPXAOS	DMKPXAPC	DMKPXARC	DMKPXATL	DMKPXAVS	DMKPXAWC	DMKPXB	DMKPXBCX	DMKPXBDL
	DMKPXBFC	DMKPXBIP	DMKPXBIS	DMKPXBMC	DMKPXBOP	DMKPXBOS	DMKPXBPC	DMKPXBRC	DMKPXBTL	DMKPXBVS
	DMKPXBWC	DMKSCHBS	DMKSCHCT	DMKSCHMS	DMKSCHN1	DMKSCHN2	DMKSCHPU	DMKSCHQT	DMKSCHQ1	DMKSCHQ2
	DMKSCHQ3	DMKSCHST	DMKSCHW1	DMKSCHW2	DMKSCNRD	DMKSELS1	DMKSWABS	DMKSWAIL	DMKSYSAT	DMKSYSMX
	DMKSYSND	DMKSYSNM	DMKSYSOC	DMKSYSOW	DMKSYSSP	DMKSYSSW	DMKSYSSZ	DMKTMRPT	DMKVFRIP	DMKVFRNC
	DMKVFRNO	DMKVFRNR	DMKVFRNS	DMKVFRSV	DMKVSICI	DMKVSICT	DMKVSICW	DMKVSISF	DMKVSISI	DMKVSITC
	DMKVSITI	DMKVSJHD	DMKVSJHI	FREE	FREENUM	F1	F2	F3	F4	F4096
	F5	F6	F7	F8	IDLEWAIT	IPLPSW	IPUADDRX	LOCK	LOCKSAV	LPUADDR
	LPUADDRX	MNCHLIST	MNCHSAM1	MNCHSAM2	MNCHSIZ	MNCHSIZE	MNCLDAST	MNCLPERF	MNCLUSER	MNCODAS
	MNCOSYS	MNCOUSER	MNDEVLST	MNDVCNT	MNDVLEN	MNDVMORE	MNRDEVB	MN000	MN000ATT	MN000CDC
	MN000CPA	MN000EXT	MN000INT	MN000ISD	MN000LEN	MN000PAP	MN000PAS	MN000PBP	MN000PBS	MN000PFP
	MN000PFS	MN000PPA	MN000PPC	MN000PSI	MN000PTP	MN000PTS	MN000Q1E	MN000Q2E	MN000RSV	MN000SRC
	MN000SSI	MN000VFO	MN000VFV	MN000VRC	MN000VSC	MN000WID	MN001	MN001LEN	MN001NXR	MN001WID
	MN002	MN002IBM	MN002IBS	MN002LEN	MN002MNS	MN002Q1A	MN002Q1B	MN002Q1C	MN002Q1D	MN002Q11
	MN002Q2A	MN002Q2B	MN002Q2C	MN002Q2D	MN002Q21	MN002SQT	MN002SQ3	MN003	MN003CDM	MN003CIE
	MN003CIP	MN003CMG	MN003CRE	MN003CSE	MN003CSR	MN003CTP	MN003LEN	MN003NSW	MN003PNP	MN003PNS
	MN003SGF	MN003SWP	MN004	MN004HL	MN004LEN	MN004NP	MN004N2	MN005	MN005ADR	MN005CCN
	MN005CIU	MN005DIR	MN005FLG	MN005LCN	MN005LEN	MN005MAL	MN005NOA	MN005PAL	MN005SER	MN005SZ
	MN005TYP	MN006	MN006END	MN006IL	MN006LEN	MN006S1	MN0061C	MN007	MN007AS1	MN007AS2
	MN007B1	MN007DPB	MN007FC1	MN007FC2	MN007LEN	MN007LSB	MN007N01	MN007PSB	MN007S1	MN4RSV1
	MN400	MN400ACT	MN400ARC	MN400INT	MN400IOC	MN400LEN	MN400LPR	MN400MHL	MN400MLH	MN400MWS
	MN400PDR	MN400PGR	MN400PSO	MN400PSP	MN400PUP	MN400PUS	MN400QDP	MN400QDR	MN400QDT	MN400RES
	MN400RST	MN400SST	MN400SWI	MN400TTI	MN400UID	MN400UPR	MN400VOT	MN400VRC	MN400VRL	MN400VSC
	MN400VTI	MN400VVT	MN400WCT	MN400WSS	MN410	MN410HWM	MN410LEN	MN410SST	MN600ADD	MN600CNT
	MN600DEV	MN600DLN	MN600HDR	MN600HLN	MN600NUM	MN600SER	MN600TY	MN602DLN	MN602HLN	MN602MLN
	MN603LNG	MN603LNM	MONCHPTR	MONCLASS	MONCODE	MONCOM	MONDVLST	MONFLAG1	MONSACT	MONSLMT
	MONSYSVM	MONUTRB	MP	MPUOPER	NUMPOOLS	PERFCL	PGREAD	PGWRITE	PREFIXB	PRIMEHDR
	PROCIO	PROCIPL	PSA	PSASVCCT	PXA	RCHADD	RCHBLOK	RCUADD	RCUBLOK	RCUCHA
	RCUPRIME	RCUSUB	RCUTYPE	RDEVADD	RDEVALLN	RDEVBLOK	RDEVCUA	RDEVIOCT	RDEVNAME	RDEVSER
	RDEVTYPC	RDIDX	RECBLOK	RECCYL	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	SAVEWRK6	SIGWAKE	SIGXC	SUBCNT	SUBHEAD	SUBSTLA	SUBTABLE	SYSPALOC
	SYSPFPNT	SYSPLIST	TABWDTH	TRQBLOK	TRQBTOD	TRQBVAL	USERCL	VACOVFR	VECINST	VECOPVF
	VECSTAT	VMBLOK	ZEROES							
DMKMPO	ADSPCH	AFREE	AFREEP	AP	APSTAT1	APSTAT2	APUOPER	AVMREAL	CPCREG0	CPEXADD
	CPEXBLOK	CPEXR11	CPEXR12	CPEXSIZE	CPPTLBR	CPRSTPND	CPSPMODE	CPSTATUS	CPSTAT2	CPSTAT3
	CPSUPER	C0	DMKCFMBK	DMKDSPB	DMKDSPCH	DMKERMSG	DMKFREE	DMKFREEP	DMKPERIL	DMKPRGSM
	DMKPSAPO	DMKPSAST	DMKSTKCP	DMKVATBC	DUMPSAVE	EXNPSW	EXOPSW	FFS	F1	F15
	F16	F2	F240	F4096	INTEX	INTEXF	IPLCCW1	IPUADDR	LOCK	LPUADDR
	MP	PERGPRS	PREFIXB	PROBMODE	PROCIPL	PROPSW	PSA	QUANTUMR	RSRTOPSW	RUNUSER
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SIGEMS	SIGREST	SIGSENSE	SIGSSS
	SIGSTART	SIGSTOP	SIGXC	SPMPFX	TIMEDISP	TIMER	TRACSTRT	TREXCR9	TREXPERA	TREXT
	VHD	VMBLOK	WAIT	WAITEND	XPAGNUM	Y0	Y2	Y4	Y6	
DMKMSG	ADSPCH	AFREE	AFRET	ALARM	AP	APSTAT1	APTRLK	APUOPER	AQCNWT	ASYSOP
	ASYSVM	BLANK	BLANKS	BUFFER	BUFNXT	CPEXADD	CPEXBLOK	CPEXR0	CPEXSIZE	DMKCVTBD
	DMKCVTDB	DMKCVTDT	DMKDSPCH	DMKERMSG	DMKFREE	DMKFRET	DMKIUACP	DMKLOCK	DMKLOCKD	DMKLOKSW
	DMKPTRLK	DMKPTRUL	DMKQCNWT	DMKQCORD	DMKSCNAU	DMKSCNFD	DMKSTKCP	DMKVMCFC	FFS	F0

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)								
	F1	F15	F2	F3	F4	F5	F6	F7	F8	F9
	HILIGHT	IXBLOK	IXIRA	IXREGS	IXSIZE	LOCK	MP	MSGMSG	MSGSMSG	MSGWNG
	NORET	NOTIME	NOTRESP	PRIORITY	PSA	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SAVER1	SAVER11	SAVER2	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4
	SAVEWRK5	SAVEWRK6	SAVEWRK8	SEVER	TIMEDISP	VCONCTL	VCONFLG2	VCONOMSG	VMBLOK	XRIGHT16
DMKMSW	AFREE	AFRET	ALARM	APSTAT1	APUOPER	AQCNWT	ASYSOP	CCC	CDC	CLASDASD
	CLASFBA	CLASTAPE	DMKCVTBD	DMKCVTBH	DMKFREE	DMKFRET	DMKLOKSW	DMKQCNWT	DMKQCORD	DMKSCNRN
	EDIT	FFS	F10	F4	F6	F8	IFCC	INTREQ	IOBLOK	IOBRADD
	IOERACT	IOERADR	IOERBLOK	IOERCNCL	IOERCSW	IOERDASD	IOERDATA	IOERDEC	IOERETRY	IOERFLG1
	IOERIGN	IOERIGNR	IOERIND3	IOERIND4	IOERINFO	IOERLEN	IOERNUM	IOERPEND	IOERSTRT	LOCK
	NORET	NOTIME	OPERATOR	PSA	RDEVBLOK	RDEVDED	RDEVNAME	RDEVSTAT	RDEVTYPC	RDEVTYPE
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER0	SAVER11
	SAVER7	TIMEDISP	TYP3340	TYP3350	TYP3480	UCASE	VMBLOK	ZEROES		
DMKNEA	AFREE	AFRET	APSTAT1	APUOPER	AQCNWT	BLANK	BLANKS	CLASGRAF	CLEAR	DMKCVTAB
	DMKCVTBH	DMKCVTHB	DMKERMSG	DMKFREE	DMKFRET	DMKLOKSW	DMKQCNWT	DMKRIORN	DMKSCNAU	DMKSCNFD
	DMKSCNMU	DMKSCNRD	DMKSCNVU	DMKSYSCK	DMKVDREL	DMKVDSDF	FFS	F1	F2	F255
	F3	F4	F6	F8	LOCK	NICBLOK	NICDISA	NICDTYPE	NICD3275	NICD3284
	NICENAB	NICEPAD	NICFLAG	NICPSUP	NICRDED	NICRFLG	NICROPER	NICRSPL	NICRUNN	NICSIZE
	NICSTAT	NICTERM	NICTMAT	NICTYPE	NICUSER	NICVDEVB	NORET	OPERATOR	PSA	RDEVBLOK
	RDEVDED	RDEVDISA	RDEVMAX	RDEVNICL	RDEVSTAT	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	SAVEAREA	SAVEREGS	SAVER2	SAVER9	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6
	SAVEWRK7	SAVEWRK8	SAVEWRK9	TIMEDISP	TYP3277	VCONCTL	VCONRDEV	VMBLOK	ZEROES	
DMKNEM	R0	R1	R11	R12	R13	R15	R2	R3	R4	R5
	SAVEAREA	SAVEREGS	SAVER0							
DMKNES	AFREE	AFRET	ALOKSP	APSTAT1	APUOPER	AQCNWT	ARIOCU	ARIODV	ASYSVM	BLANKS
	CACTLTR	CDISPLY	CLASSPEC	CLASTERM	CONCCW3	CONDATA	CONSYSR	CONTASK	CPSTAT5	CSWLMEP
	CSWLNCP	CTRMLTR	DMKCVTBH	DMKCVTDB	DMKCVTHB	DMKERMSG	DMKFREE	DMKFRET	DMKIOESR	DMKLOKIO
	DMKLOKSW	DMKQCNWT	DMKQCOCL	DMKQCPTO	DMKRGBEN	DMKRIORN	DMKRNHND	DMKRNHTR	DMKSCNFD	DMKSCNRD
	DMKSCNRU	FFS	F1	F2	F255	F3	F4	F6	F7	F8
	LOCK	LPUADDR	MODESWT	NICBLOK	NICCIBM	NICDISA	NICENAB	NICEPAD	NICEPMD	NICFLAG
	NICLBSC	NICLINE	NICLTRC	NICPSUP	NICQPNT	NICSESN	NICSIZE	NICSTAT	NICSWEP	NICTYPE
	NICUSER	NORET	PSA	RCHBLOK	RCHCUTBL	RCUBLOK	RCUDISA	RCUDVTBL	RCUSTAT	RDEVADD
	RDEVASTB	RDEVAVM2	RDEVBASE	RDEVBLOK	RDEVCON	RDEVCTRS	RDEVCUA	RDEVDED	RDEVDISA	RDEVDISB
	RDEVENAB	RDEVEPDV	RDEVEPLN	RDEVEPMD	RDEVFLAG	RDEVIRM	RDEVLNCP	RDEVMAX	RDEVMDL	RDEVNICL
	RDEVNRDY	RDEVPDLY	RDEVPEND	RDEVPTHS	RDEVPTTC	RDEVRCVY	RDEVRSVD	RDEVSADN	RDEVSDR	RDEVSLOW
	RDEVSTAT	RDEVSTA4	RDEVSTA6	RDEVTBTU	RDEVTCTL	RDEVTMCD	RDEVTYPC	RDEVTYPE	RDEVUSC8	RDEVUSER
	RDEVVM2	RDEVWAIT	RDIDX	R0	R1	R10	R11	R12	R13	R14
	R15	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA
	SAVEREGS	SAVER11	SAVER2	SAVER9	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK7
	SAVEWRK8	SAVEWRK9	TIMEDISP	TYPBSC	TYPTTY	TYPUNDEF	TYP2700	TYP3705	VMBLOK	
DMKNET	AFREE	AFRET	AP	APSTAT1	APUOPER	AQCNWT	ARIODV	ASYSVM	BLANKS	BSCBLOK
	BSCFLAG1	BSCHALT	BSCSHUT	CACTLIN	CDCTLIN	CLASSPEC	CLASTERM	CONCCW3	CONSYSR	CONTACT
	CPSTAT5	CRESIMD	DMKCFQRD	DMKCVTBH	DMKCVTHB	DMKERMSG	DMKFREE	DMKFRET	DMKIOESR	DMKLOKSW
	DMKNESEP	DMKNESWN	DMKQCNWT	DMKRGBEN	DMKRIORN	DMKRNHND	DMKSCNFD	DMKSCNRD	DMKSCNRU	DMKSCNVU
	F255	F3	F4	F6	F60	F7	F8	LOCK	MP	NICADVF
	NICBLOK	NICCIBM	NICDISA	NICDISB	NICDTYPE	NICDXSC	NICD3284	NICENAB	NICEPAD	NICEPMD
	NICFLAG	NICFMT	NICLBSC	NICLGRP	NICLINE	NICNAME	NICPSUP	NICQRY	NICRATTD	NICRATTN

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	NICRDED	NICRFLG	NICRSPL	NICRUNN	NICSESN	NICSIZE	NICSTAT	NICTELE	NICTERM	NICTYPE
	NICUSER	NICVDEVB	NORET	PSA	RDEVAUTO	RDEVBASE	RDEVBLOK	RDEVBSC	RDEVCTRS	RDEVDED
	RDEVDISA	RDEVDISB	RDEVENAB	RDEVFLAG	RDEVLNCP	RDEVMAX	RDEVNICL	RDEVNRDY	RDEVPEND	RDEVRSVD
	RDEVSTAT	RDEVTYPC	RDEVUSER	RDIDX	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	SAVER2	SAVER9	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK7
	SAVEWRK9	TEMPSAVE	TIMEDISP	TYP3705	VMBLOK	ZEROES				
DMKNLD	ABORT	ADSPCH	AFREE	AFREEP	AFRET	ALOKSP	AP	APSTAT1	APTRAN	APTRLK
	APUOPER	AQCNWT	ASYSVM	ATTN	BLANKS	BRING	CC	CCPARM	CCPENTRY	CCPMAXID
	CCPNAME	CCPPSIZE	CCPRESID	CCPRSTAT	CCPRSTEP	CCPRSTYP	CCPSIZE	CCPTNCP	CCPTPEP	CCPTYPE
	CDC	CE	CLASFBA	CLASSPEC	CUE	C1	DE	DEFER	DMKCVTBH	DMKCVTHB
	DMKDSPCH	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKIOSQR	DMKLOKIO	DMKLOKSW	DMKPGUVG	DMKPGUVR
	DMKPTRAN	DMKPTRUL	DMKQCNWT	DMKQCOCL	DMKQCORD	DMKQCPTO	DMKRNHIN	DMKRPAGT	DMKSCNFD	DMKSCNRD
	DMKSCNRU	DMKSCNVS	DMKSCNVU	DMKSNTRN	DMKSTKIO	DMKVDREL	DMKVDRES	EDIT	ERRMSG	FFS
	FTRTYP1	F0	F1	F256	F3	F4096	F6	F8	IL	INTREQ
	IOBBPNT	IOBCAW	IOBCC1	IOBCC3	IOBCP	IOBCSW	IOBFLAG	IOBFPNT	IOBIOER	IOBIRA
	IOBLINK	IOBLOK	IOBMISC	IOBMISC2	IOBRADD	IOBRCAW	IOBRCNT	IOBRSTRT	IOBSIZE	IOBSPEC
	IOBSTAT	IOBTIO	IOBUNSL	IOBUSER	IOERBLOK	IOERDATA	IOERETN	IOEREXT	IOERSIZE	IPLREQ
	LOCK	LOCKSAV	LPUADDR	MP	NCPNAME	NCPPAGCT	NCPPNT	NCPSTART	NCPTBL	NCPVOL
	NICBLOK	NICCIBM	NICEPAD	NICEPMD	NICFLAG	NICNAME	NICPSUP	NICSIZE	NICSTAT	NICSWEP
	NICTERM	NICTYPE	NICUSER	NOAUTO	NORET	NOTRESP	OPERATOR	PSA	RCUBLOK	RCUDISA
	RCUDVTBL	RCUSTAT	RDEVADD	RDEVAIOB	RDEVATT	RDEVBASE	RDEVBLOK	RDEVCODE	RDEVCUA	RDEVDED
	RDEVDISA	RDEVENAB	RDEVEPDV	RDEVEPLN	RDEVEPMD	RDEVFIOB	RDEVFLAG	RDEVFTR	RDEVIOER	RDEVIRM
	RDEVLCEP	RDEVLNCP	RDEVMAX	RDEVMDL	RDEVNCP	RDEVNICL	RDEVNRDY	RDEVOWN	RDEVPEND	RDEVPTHS
	RDEVPTTC	RDEVRCVY	RDEVRSVD	RDEVSTAT	RDEVSTA2	RDEVSTA4	RDEVTFLG	RDEVTMCD	RDEVTYPC	RDEVTYPE
	RDEVUSER	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER11
	SAVER2	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9
	SILI	SM	SYSTEM	TEMPSAVE	TIMEDISP	TYPBSC	TYPIBM1	TYPUNDEF	TYP2314	TYP3330
	TYP3350	TYP3375	TYP3380	TYP3705	UC	UCASE	VMBLOK	XPAGNUM	ZEROES	
DMKNLE	ABORT	ADDSFB	ADSPCH	AFREE	AFREEP	AFRET	APTRAN	APTRLK	AQCNWT	ASYSVM
	ATTN	BLANKS	BRING	CC	CDC	CLASFBA	CLASSPEC	CUE	C1	DE
	DEFER	DMKCKSPL	DMKCKTSD	DMKCKTUU	DMKCVTBH	DMKCVTDT	DMKCVTHB	DMKDSPCH	DMKERMSG	DMKFREE
	DMKFREEP	DMKFRET	DMKIOSQR	DMKLOCRD	DMKLOCRQ	DMKPGTCG	DMKPGUSD	DMKPGUVG	DMKPGUVR	DMKPTRAN
	DMKPTRUL	DMKQCNWT	DMKQCORD	DMKRNHIN	DMKRPAPT	DMKSCNFD	DMKSCNRD	DMKSCNRU	DMKSTKIO	DMKSYSDU
	DMKVSDAD	DMKVSGRI	EDIT	ERRMSG	FTRTYP1	F0	F1	F2	F256	F3
	F4	F4096	F5	F6	F8	IL	INTREQ	IOBCAW	IOBCC1	IOBCC3
	IOBCP	IOBCSW	IOBFLAG	IOBIOER	IOBIRA	IOBLINK	IOBLOK	IOBMISC	IOBMISC2	IOBRADD
	IOBRCAW	IOBRCNT	IOBRSTRT	IOBSIZE	IOBSPEC	IOBSTAT	IOBTIO	IOBUNSL	IOBUSER	IOERBLOK
	IOERDATA	IOERETN	IOEREXT	IOERSIZE	IPLREQ	LOCK	NOAUTO	NORET	NOTRESP	OPERATOR
	PSA	RDEVAUTO	RDEVBLOK	RDEVDED	RDEVDISA	RDEVFLAG	RDEVFTR	RDEVMDL	RDEVNRDY	RDEVPEND
	RDEVRCVY	RDEVRSVD	RDEVSTAT	RDEVTYPC	RDEVTYPE	RDEVUSER	RDRCHN	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER2	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4
	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	SFBCLAS	SFBCOPY	SFBDATE	SFBDIST	SFBDUMP
	SFBFILID	SFBFLAG	SFBFNAME	SFBFTYPE	SFBLAST	SFBLOK	SFBOFORM	SFBORIG	SFBRECNO	SFBRECSZ
	SFBSIZE	SFBSTART	SFBSYSID	SFBTIME	SFBTYPE	SFBUFORM	SFBUSER	SILI	SM	SYSTEM
	TYPPRT	TYP2314	TYP3330	TYP3350	TYP3375	TYP3380	TYP3705	UC	UCASE	VMBLOK
DMKNMT	FREELOWE	FSTFMODE	FSTFNAME	NUCON	R0	R1	R12	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9			
DMKOPE	AFREE	AFRET	ALARM	AP	APSTAT1	APTRAN	APTRLK	APUOPER	AQCNWT	ASYSOP

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	ASYSVM	AUTGO	BLANKS	BRING	BUFCNT	BUFFER	BUFIN	BUFNXT	BUFSIZE	BUSY
	CAW	CE	CLASFBA	CLASGRAF	CLASTERM	CONTINUE	CPABEND	CPID	CPIEXLOG	CPSTAT4
	CPSTAT5	CPUMODEL	CSW	CUE	C1	C2	DE	DEFER	DMKALGON	DMKCFMEN
	DMKCNSEN	DMKCPEID	DMKCPIBD	DMKCPICD	DMKCPIOL	DMKCPITR	DMKCPIWC	DMKCVTBD	DMKCVTBH	DMKERMCP
	DMKERMSG	DMKFREE	DMKFRET	DMKGRCCP	DMKGRCHL	DMKGRCMR	DMKGRCRU	DMKGRCVM	DMKIOTIN	DMKLOGOP
	DMKMCHST	DMKMNIST	DMKPGUVG	DMKPGUVR	DMKPTRAN	DMKQCNWT	DMKQCORD	DMKQCOSY	DMKRIOCN	DMKRPAGT
	DMKRPAPT	DMKSCNEP	DMKSCNMU	DMKSCNVS	DMKSTABD	DMKSYSAT	DMKSYSUR	DMKSYSWM	DMKSYSWV	DMKUDRFU
	DMKUDRMD	DMKUDRRV	EDIT	FF	FFS	F0	F1	F4	F8	INTTIO
	IOBLOK	IOBUSER	IONPSW	LOCK	LPUADDR	LPUADDRX	MP	NOAUTO	NORET	NOTIME
	OPERATOR	POWEROFF	PREFIXA	PSA	PSALANG	PSASYSID	RDEVAIOB	RDEVATOF	RDEVBLOK	RDEVCODE
	RDEVENAB	RDEVFLAG	RDEVIDNT	RDEVPTTC	RDEVRUN	RDEVTFLG	RDEVTMCD	RDEVTYPC	RDEVTYPE	RDEVUSER
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R8	R9	SAVEAREA	SAVEREGS	SILI	SYSTEM	TEMPSAVE	TIMEDISP
	TYPBSC	TYP2741	TYP3066	TYP3210	TYP3277	TYP3278	UC	UCASE	VMBLOK	ZEROES
DMKOPR	ALARM	AVMREAL	BCTWAIT	BUSY	CAW	CC	CD	CE	CLASGRAF	CLASTERM
	CONTINUE	CPID	CPSTAT5	CPUID	CPUVERSN	CSW	C2	DE	DMKDMPLK	DMKIOTIN
	DMKRIOCN	DMKRIODV	DMKSCNRU	DMKVRRIS	FFS	F0	F1	INTTIO	IONPSW	LOCK
	NOAUTO	PSA	RDEVBLOK	RDEVCORD	RDEVGRTY	RDEVTYPC	RDEVTYPE	RDEVUSER	RDIDX	R0
	R1	R10	R12	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SENSE	SILI	SM	TYP3066	TYP3210	TYP3277	TYP3278
	UC	XRIGHT16								
DMKOVR	BLANKS	CC	CSW	DEVICE	ERRMSG	INBUFF	INFILE	R1	R10	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	SENSE	SILI	WAIT					
DMKPAG	ACORETBL	ADSPCH	AFREE	ALOCBLOK	ALOCCYL1	ALOCCYL2	ALOCFLG	ALOCPG	ALOCPM	ALOCPNT
	ALOCPP	ALOCPRD	ALOCPS	ALOCPSIO	ALOCPWRT	ALOCSW	ALOCTMS	ALOKRM	ALOKSP	ALOKSY
	APSTAT1	APUOPER	ASYSVM	CC	CD	CLASDASD	CLASFBA	CPEXADD	CPEXBLOK	CPEXBPNT
	CPEXFPNT	CPEXR0	CPEXR11	CPEXR5	CPEXR7	CPEXR9	DMKDSPCH	DMKDSPRU	DMKFREE	DMKIOSQR
	DMKPAHIO	DMKPGTPB	DMKPGUAL	DMKPTRPS	DMKPTRRQ	DMKPTRWQ	DMKPTSAD	DMKSCNMU	DMKSTKCP	DMKSTKIO
	DMKSYSOC	DMKSYSOW	D3	D4	FTRFH	FTR70MB	F1	F10	F2	F3
	F4	F6	F8	IDA	IOBBPNT	IOBCAW	IOBCC3	IOBCP	IOBCSW	IOBCYL
	IOBFATAL	IOBFH	IOBFLAG	IOBFPNT	IOBIRA	IOBLINK	IOBLOK	IOBMINI	IOBMISC	IOBMISC2
	IOBOFF	IOBPAG	IOBRADD	IOBSIZE	IOBSPEC	IOBSPEC2	IOBSTAT	IOBUSER	LASTUSER	LOCK
	LPUADDR	L1	L16	L3	L4	L8	OWNDLIST	OWNDRDEV	PAGECCWS	PAGECYL
	PAGEFBAT	PAGEFSNS	PAGEFTIC	PAGEHEAD	PAGEIDA1	PAGEIDA2	PAGEIOB	PAGELOCA	PAGELOCB	PAGELOCD
	PAGELOCN	PAGELOCO	PAGELOCW	PAGENUME	PAGEPARM	PAGEPRI	PAGERCD	PAGERW	PAGESECT	PAGESEEK
	PAGESIZE	PAGESK	PAGESLOT	PAGESNS	PAGESRCD	PAGESRCH	PAGESS	PAGETYPE	PAGEXDED	PAGEXECC
	PAGEXEHH	PAGEXFER	PAGEXLR	PAGEXLRD	PAGEXOP	PAGEXRW	PAGEXSHC	PAGEXSHH	PAGEXSIZ	PAGEXSKC
	PAGEXSKH	PAGEXSLT	PAGEXSL1	PAGEXSNS	PAGEXSTR	PAGEXST2	PAGEXTIC	PAGSWARG	PAGSWCC	PAGSWCHN
	PAGSWCSZ	PAGSWDA1	PAGSWDA2	PAGSWDSZ	PAGSWEXT	PAGSWHH	PAGSWNOP	PAGSWR	PAGSWRW	PAGSWSCH
	PAGSWTIC	PAGXIDA1	PAGXIDA2	PCIF	PREFIXA	PREFIXB	PSA	PSAEXT	PSTPINPP	PSTPOUPP
	PXA	RDCBLKAP	RDCBLKPG	RDCBLOK	RDCPAGXT	RDEVADD	RDEVALLN	RDEVBLOK	RDEVBUZY	RDEVCODE
	RDEVCUP	RDEVECKD	RDEVFIOB	RDEVFLAG	RDEVFTR	RDEVFTR2	RDEVIOBL	RDEVLOW	RDEVMDL	RDEVMD02
	RDEVPRDV	RDEVQCNT	RDEVRDC	RDEVSCHD	RDEVSIZE	RDEVSTA4	RDEVSTA5	RDEVTYPC	RDEVTYPE	RDEVUSER
	RDEV333V	READ	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SILI	SKIP
	SSBCYL	SSBEFLG	SSBEINVL	SSBENTRL	SSBENTRY	SSBENUM	SSBEPGS	SSBHEADL	SSBLOK	SSBNPGS
	SSBUPGS	SSBVPAGE	SWPCHG1	SWPCYL	SWPFLAG	SWPPSTOR	SWPRECMP	SWPTRANS	TIMEDISP	TYP2305
	TYP2314	TYP3330	TYP3340	TYP3350	TYP3375	TYP3380	VMBLOK	XCDISP	XCPEND	XTNDLOCK
DMKPAH	ADSPCH	AFRET	ALOKRM	ALOKSP	APSTAT1	APUOPER	CC	CD	CE	CORTABLE
	CPEXADD	CPEXBLOK	CPEXBPNT	CPEXFPNT	CPEXMISC	CPEXR0	CPEXR5	CPEXR9	CUE	DE

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	DMKCVTBH	DMKDSPCH	DMKFRET	DMKIOSQR	DMKMCHST	DMKPAGEX	DMKPAGOU	DMKPAGSK	DMKPAGXS	DMKSCNMU
	DMKSCNRD	DMKSCNRU	DMKSTKCP	DMKSTKMP	DMKSTKOP	DMKSYSOW	F5	IL	IOBCAW	IOBCSW
	IOBFATAL	IOBLOK	IOBMISC	IOBMISC2	IOBRADD	IOBSIZE	IOBSTAT	LOCK	LPUADDR	OWNDLIST
	OWNDRDEV	PAGECCWS	PAGEFBAT	PAGEIOB	PAGELOCA	PAGELOCD	PAGELOCN	PAGERW	PAGESEEK	PAGESK
	PAGESNS	PAGESS	PAGETYPE	PAGEXFER	PAGEXLR	PAGEXRW	PAGEXSNS	PAGEXTIC	PAGSWCHN	PAGSWEXT
	PCI	PCIF	PSA	RDCBLKPG	RDCBLOK	RDEVBLOK	RDEVRDC	R0	R1	R10
	R11	R12	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	SILI	SKIP	SSBCODE	SSBLOK	SWPCODE	SWPFLAG	SWPRECMP	VMBLOK
DMKPEI	AFREE	AFRET	APTRAN	AQCNWT	BRING	COLON	COUNT	C1	DEFER	DMKCVTBD
	DMKCVTDB	DMKCVTHB	DMKERMSG	DMKFREE	DMKFRET	DMKLOCKD	DMKLOCKQ	DMKPELSD	DMKPEND	DMKPENGT
	DMKPENSV	DMKPETAB	DMKPTRAN	DMKQCNWT	DMKSCNFD	DMKVAURN	EXTMODE	FLAGS	F0	F1
	F15	F2	F240	F3	F5	F6	F7	F8	LOCK	NORET
	PERANYTH	PERBLIP	PERBLOK	PERBUF	PERCHAIN	PERCMDPT	PERCR9	PERCTACT	PERDATA	PERERROR
	PERFLAG	PERGPRP	PERHITS	PERINTO	PERLNEND	PEROPTN	PERPASCT	PERPASSP	PERPUSED	PERREGSV
	PERRNGTB	PERSCADD	PERSCAN	PERSIZE	PERSTPCT	PERSTPSP	PERTBAK	PERTBLEN	PERTMPCH	PERTOTAL
	PERWKFLG	PERWKFL2	PERWKLEN	PERWORK	PERWRKCT	PEXBLOK	PEXBR	PEXDATA	PEXDINV	PEXDLEN
	PEXELIM	PEXFLAG0	PEXFLAGT	PEXFROM2	PEXGPR	PEXGREG	PEXINST	PEXINTO	PEXINTO1	PEXLEN
	PEXMASK	PEXNEXT	PEXPRINT	PEXRUN	PEXSECND	PEXSIZE	PEXSTORE	PEXTERM	PEXTHIRD	PSA
	RANGE	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	SAVEAREA	SAVEREGS	SAVER2	SAVEWRK1
	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	TRANMODE	VMBLOK	XPAGNUM	XRIGHT16	XRIGHT24	ZEROES
DMKPEL	AFREE	AFRET	DMKERMSG	DMKFREE	DMKFRET	EXTMODE	F2	F3	F4	F5
	F6	LOCK	OVERLAP	PERBLOK	PERCHAIN	PERCMDPT	PERCNTCH	PERCOUNT	PERCR10	PERCR11
	PERCR9	PERMODE	PEROFF	PERON	PERPASCT	PERPASSP	PERRNGTB	PERSTPCT	PERSTPSP	PERTMPCH
	PERTOTAL	PERWKFLG	PERWRKCT	PEXBLOK	PEXBR	PEXCHCMP	PEXCMND	PEXDATA	PEXDLEN	PEXELIM
	PEXFLAG0	PEXFLAGT	PEXFROM	PEXGPR	PEXGREG	PEXINST	PEXINTO	PEXLEN	PEXMASK	PEXMECMP
	PEXNEXT	PEXPASS	PEXPRINT	PEXRANGE	PEXRUN	PEXSECND	PEXSTEP	PEXSTORE	PEXTERM	PEXTHIRD
	PSA	RANGE	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	SAVEAREA	SAVEREGS	SAVER2
	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK8	VMBLOK	XRIGHT24	ZEROES	
DMKPEN	AFREE	AFRET	AQCNWT	BLANKS	CL	COUNT	DMKCVTBD	DMKCVTDB	DMKERMSG	DMKFREE
	DMKFRET	DMKPELCH	DMKPELCR	DMKQCNWT	DMKSCNFD	EXTMODE	F1	F2	F3	F4
	F8	LOCK	NORET	PERANYTH	PERAPPND	PERBLOK	PERBUF	PERCHAIN	PERCHANG	PERCLEAR
	PERCNTCH	PERCOUNT	PERCTACT	PERCTEND	PERENDIT	PERFLAG	PERGPRP	PERHITS	PERMODE	PERREGSV
	PERSAVED	PERSCAN	PERSCDLN	PERSIZE	PERTBAK	PERTBLEN	PERTMPCH	PERTOTAL	PERWKFLG	PERWKFL2
	PERWRKCT	PESBLOK	PESCHAIN	PESCOUNT	PESELIM	PESFLAG	PESNAME	PESNEXT	PESSIZE	PEXBLOK
	PEXBR	PEXCMND	PEXELIM	PEXFLAG0	PEXFLAGT	PEXGPR	PEXINST	PEXLEN	PEXMASK	PEXNEXT
	PEXSTORE	PSA	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	SAVEAREA	SAVEREGS	SAVER2
	SAVEWRK1	SAVEWRK2	SAVEWRK3	VMBLOK	ZEROES					
DMKPEQ	AQCNWT	BLANK	BLANKS	CL	CLEAR	DATAOUT	DMKCVTBD	DMKCVTBH	DMKERMSG	DMKLOCKD
	DMKLOCKQ	DMKQCNWT	DMKSCNFD	FFS	F1	F15	F2	F3	F4	F5
	F6	F8	LOCK	PERBLOK	PERBUF	PERCHAIN	PERCTACT	PERFLAG	PERHITS	PERSAVED
	PESBLOK	PESCHAIN	PESNAME	PESNEXT	PEXBLOK	PEXBR	PEXCMND	PEXDATA	PEXDLEN	PEXFLAG0
	PEXFLAGT	PEXFROM	PEXGPR	PEXGREG	PEXINTO	PEXNEXT	PEXPASS	PEXRUN	PEXSECND	PEXSTEP
	PEXSTORE	PEXTHIRD	PSA	RANGE	RESET	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SAVER2	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6
	SAVEWRK8	SAVEWRK9	VMBLOK	XRIGHT24						
DMKPER	AFREE	APTRAN	AQCNWT	BRING	C1	DEFER	DMKCFMBK	DMKCVTBD	DMKERMSG	DMKFREE

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	DMKLOCKD	DMKLOCKQ	DMKPELCR	DMKPENDA	DMKPETAL	DMKPTRAN	DMKQCNWT	DMKVAURN	EXTMODE	F0
	F1	F15	F4095	LOCK	PERADDR	PERBLIP	PERBLOK	PERBUF	PERCDE	PERCHAIN
	PERCR9	PERCTACT	PERDATON	PEREX	PEREXADD	PEREXMOD	PERFLAG	PERGALT	PERGPRP	PERHITS
	PERINST	PEROPNOT	PEROPQU	PEROP1	PEROP2	PERSTLEN	PERTBAK	PERTBLEN	PEXBLOK	PEXBR
	PEXDATA	PEXDINV	PEXDLEN	PEXFLAG0	PEXFLAGT	PEXFROM	PEXGPR	PEXGREG	PEXGSUC	PEXINST
	PEXINTO	PEXINTO1	PEXMASK	PEXNEXT	PEXPASS	PEXPASSN	PEXSECND	PEXSTORE	PEXSUCC	PEXTERM
	PEXTHIRD	PSA	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS
	SAVEWRK1	SAVEWRK3	SAVEWRK4	SAVEWRK5	VMBLOK	XPAGNUM	XRIGHT16	XRIGHT24	ZEROES	
DMKPET	AFREE	AFRET	AP	AQCNWT	ATTNHIT	BUFCNT	BUFFER	BUFNXT	BUFSIZE	COLON
	DMKCFMBK	DMKCFMEN	DMKERMSG	DMKFREE	DMKFRET	DMKLOCKD	DMKNEMOP	DMKPENDA	DMKQCNWT	DMKSCNFD
	DMKVSTPT	EXTMODE	F1	F3	LOCK	MP	PERADDR	PERBLOK	PERBUF	PERCHAIN
	PERDATON	PEREX	PEREXADD	PEREXMOD	PERFLAG	PERINST	PEROPNOT	PEROP1	PEROP2	PERSAVED
	PERSEQP	PERSEQT	PERTBAK	PEXBLOK	PEXBR	PEXCMND	PEXDATA	PEXDLEN	PEXFLAG0	PEXFLAGT
	PEXGPR	PEXGSUC	PEXINTO	PEXMASK	PEXNEXT	PEXPRINT	PEXRUN	PEXSTEP	PEXSTEPN	PEXSTORE
	PEXSUCC	PEXTERM	PSA	R0	R1	R10	R11	R12	R13	R14
	R15	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA
	SAVEREGS	SAVER2	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8
	VMBLOK	ZEROES								
DMKPGM	ACORETBL	ADSPCH	AFREE	AFREEP	AFRET	ALOCBLOK	ALOCFLG	ALOCMALC	ALOCMGIN	ALOCMGOU
	ALOCPG	ALOCPP	ALOCPRD	ALOCPSIO	ALOCSW	ALOKRM	ALOKSP	AP	APSTAT1	APUOPER
	AQCNWT	ASYSVM	CORFLAG	CORFLUSH	CORTABLE	CPEXADD	CPEXBLOK	CPEXFPNT	CPEXMISC	CPEXR0
	CPEXR9	CPEXSIZE	DFRET	DMKCVTAB	DMKDSPCH	DMKFREE	DMKFREEP	DMKFRET	DMKLOKSW	DMKPAGIO
	DMKPGTDC	DMKPGTDP	DMKPGTMT	DMKPGTMV	DMKPGTMX	DMKPGTPB	DMKPGTPC	DMKPGTPM	DMKPGTPN	DMKPGTSF
	DMKPGTSU	DMKPGUAL	DMKPGUPR	DMKPTREP	DMKPTRPS	DMKPTRRQ	DMKPTRWQ	DMKPTTFT	DMKQCNWT	DMKSCHMS
	DMKSCNAU	DMKSTKCP	DMKSTRAN	DMKSWAIS	DMKSWAPR	DMKSWMIG	DMKSYSPG	DMKSYSPM	DMKSYSPP	DMKTRQST
	F0	F1	F16	F2	LOCK	LOCKSAV	LPUADDR	MP	NORET	PAGINVAL
	PAGRBITS	PAGTABLE	PAGTOT	PGREAD	PGWRITE	PSA	PSAEXT	PSTPINPP	PSTPINSW	PXA
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVERETN	SAVER0
	SAVER1	SAVER11	SAVER12	SAVER13	SAVER7	SAVER9	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4
	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	SEGFLAG	SEGMIG	SEGMIGPG	SEGMIGPP	SEGPAGE
	SEGTABLE	SSBALLOC	SSBBPNT	SSBCODE	SSBCYL	SSBEFLG	SSBEINVL	SSBENTRL	SSBENTRY	SSBENUM
	SSBEPGS	SSBFLAG	SSBFPNT	SSBHEADL	SSBLOK	SSBMGREF	SSBNDLCT	SSBNPGS	SSBPSTOR	SSBTRANS
	SSBUPGS	SSBVPAGE	SWPALLOC	SWPCHG1	SWPCODE	SWPCYL	SWPFLAG	SWPPSTOR	SWPRECMP	SWPTABLE
	SWPTRANS	SWPVPAGE	SYSPFLG2	SYSPFPNT	SYSPLIST	SYSPPST	TEMPSAVE	TIMEDISP	TRQBIRA	TRQBLOK
	TRQBTOD	TRQBUSER	TRQBVAL	TRQREGS	TRQREGSD	VMBLOK	XMIGACT	XPGFULL	XPPFULL	XSWFULL
	ZEROES									
DMKPGS	ACORETBL	ADSPCH	AFREE	AFREEP	AFRET	ALOCBLOK	ALOCFLG	ALOCPP	AP	APSTAT1
	APSTAT2	APTRAN	APUOPER	ASYSVM	AVMREAL	CORCFLCK	CORDISA	CORFLAG	CORFPNT	CORFREE
	CORIOLCK	CORPGPNT	CORSHARE	CORSWPNT	CORTABLE	CORVM	CPEXADD	CPEXBLOK	CPEXR0	CPEXSIZE
	CPPTLBR	C1	DEFER	DMKBLDRL	DMKBLDRT	DMKCVTAB	DMKDSPCH	DMKDSPNP	DMKFREE	DMKFREEP
	DMKFRET	DMKHVDIP	DMKIUACU	DMKLOCKD	DMKLOCKQ	DMKPGUAL	DMKPGUPR	DMKPGUSP	DMKPTRAN	DMKPTRAQ
	DMKPTRSC	DMKPTRUL	DMKPTSAD	DMKPTSPW	DMKPTSRS	DMKPTTFT	DMKSCHST	DMKSTKCP	DMKSWAPR	DMKVATSI
	DMKVFSRS	DMKVMASH	FFS	F0	F1	F15	F16	F4096	F8	INTPR
	IUCVBLOK	IUCVCPEX	KEEPSEGS	LASTUSER	LOCK	MICBLOK	MICVTMR	MP	OLDVMSEG	PAGACT
	PAGBMP	PAGCORE	PAGRBITS	PAGREF	PAGSHR	PAGTABLE	PAGTONLY	PAGTOT	PAGTSWP	PREFIXB
	PRNPSW	PROCIPL	PSA	R0	R1	R10	R11	R12	R13	R14
	R15	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA
	SAVEREGS	SAVER1	SAVER12	SAVER13	SAVER2	SAVER3	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4
	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK9	SEGFLAG	SEGINV	SEGPAGE	SEGPROT	SEGTABLE	SHRBPNT
	SHRFLAG	SHRFPNT	SHRNAME	SHRNOPRT	SHRSEGCT	SHRSEGNM	SHRSGPRT	SHRTABLE	SHRTSIZE	SHRUSECT

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	SPMPFX	SSBALLOC	SSBENTRL	SSBENTRY	SSBENUM	SSBFLAG	SSBHEADL	SSBLOK	SSBTRANS	SWPALLOC
	SWPCODE	SWPCYL	SWPFLAG	SWPKEY1	SWPPSTOR	SWPRECMP	SWPSEGNO	SWPSHR	SWPTABLE	SWPTRANS
	SWPVM	SWPVPAGE	TEMPR1	TIMER	TREXANSI	TREXIN1	TREXNSI	TREXT	TRQBIRA	TRQBLOK
	TRQBTOD	TRQBUSER	TRQBVAL	TRQREGSD	TRQREGO	VMABLOK	VMAFPNT	VMANAME	VMASIZE	VMBLOK
	VMSIZE	XRIGHT16	ZEROES							
DMKPGT	ADSPCH	AFREE	AFREEP	ALARM	ALOCALOC	ALOCBLOK	ALOCCHN	ALOCCSR	ALOCCYL1	ALOCCYL2
	ALOCDOWN	ALOCDU	ALOCFLG	ALOCMAP	ALOCMAX	ALOCNPAG	ALOCPAGE	ALOCPAVL	ALOCPG	ALOCPMAX
	ALOCPP	ALOCPREC	ALOCPS	ALOCPUSE	ALOCRDEV	ALOCRECS	ALOCRSET	ALOCSW	ALOCTMS	ALOCTMSI
	ALOCUSED	AP	APSTALOC	AQCNWT	ASYSVM	BALRSAVE	BALR0	BALR1	BALR15	BALR8
	CLASFBA	CPEXADD	CPEXBLOK	CPEXR11	CPEXSIZE	CPID	DMKDMPAL	DMKDMPRC	DMKDMPSW	DMKDSPCH
	DMKFREE	DMKFREEP	DMKQCNWT	DMKSTKCP	DMKSYSDP	DMKSYSPE	DMKSYSPG	DMKSYSPP	DMKSYSPS	DMKSYSSW
	DMKSYSSZ	DMKSYSVM	F1	LOCK	MP	NORET	OPERATOR	PSA	PSTRCPG$	RDEVBLOK
	RDEVCODE	RDEVTYPC	RDEVTYPE	RECBAK	RECBLOK	RECCYL	RECMAP	RECMAX	RECPNT	RECSIZE
	RECUSED	R0	R1	R10	R11	R12	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SWPPSTOR	SYSPALOC	SYSPBPNT	SYSPFLG
	SYSPFPNT	SYSPLIST	TYP3350	TYP3375	TYP3380	VMBLOK	XPGFULL	XPPFULL	XSWFULL	ZEROES
DMKPGU	AFRET	ALOCBLOK	ALOCCSR	ALOCCYL1	ALOCCYL2	ALOCDLOC	ALOCDOWN	ALOCFLG	ALOCMAP	ALOCPAGE
	ALOCPG	ALOCPMAX	ALOCPNT	ALOCPP	ALOCPREC	ALOCPS	ALOCPUSE	ALOCRDEV	ALOCRECS	ALOCRSET
	ALOCSREC	ALOCSW	ALOCUSED	APSTALOC	ARIODV	ASYSVM	BALRSAVE	BALR1	CLASFBA	CPSTAT5
	DMKCKP	DMKFRET	DMKPGTDC	DMKPGTMS	DMKPGTPC	DMKPGTPU	DMKPGTSU	DMKPGTTU	DMKPGT90	DMKPTRPS
	DMKSYSOW	LOCK	L8	OWNDLIST	OWNDRDEV	PSA	PSTRCPG$	RDCBLOK	RDCPAGAP	RDEVALLN
	RDEVBLOK	RDEVRDC	RDEVTYPC	RDIDX	RECBAK	RECBLOK	RECCYL	RECMAP	RECMAX	RECPNT
	RECSIZE	RECUSED	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SSBCYL	SSBEFLG
	SSBEINVL	SSBENTRL	SSBENTRY	SSBENUM	SSBHEADL	SSBLOK	SSBNPGS	SSBVPAGE	SWPCYL	SWPFLAG
	SWPPSTOR	SWPRECMP	TEMPSAVE	VMBLOK						
DMKPIA	CC	LOCK	SILI							
DMKPIB	CC	LOCK	PIB	SILI						
DMKPMA	ACORETBL	ACTIVTRQ	AFRET	ALLCHANS	APSTAT1	APTRAN	APTRLK	APUOPER	ARIOCT	ASYSVM
	AVMREAL	BRING	CDCONF	CDDED	CDDEF	CKCMASK	CLASDASD	CLASFBA	CODE	CORCP
	CORFLAG	CORFPNT	CORTABLE	CPRUN	CPSPMODE	CPSTATUS	CPSTAT2	CPSUPER	CPTMASK	CRLOG
	C0	C1	C15	C2	C6	DAMAGRPT	DEFER	DMKDMPCP	DMKDMPC2	DMKDMPG2
	DMKDSPB	DMKDSPNP	DMKFRET	DMKPGUVG	DMKPRGCT	DMKPRVLG	DMKPSAPO	DMKPTRAN	DMKPTRFR	DMKSCHRT
	DMKSCHST	DMKSCHTQ	DMKSCNRD	DMKSCNRU	DMKSCNVD	DMKSCNVU	DMKTMR	DMKTMRSP	DMKVRR	EMSMASK
	EXNPSW	EXOPSW	EXTMODE	FAILSTAD	FFS	FPRLOG	FTR35MB	FTR70MB	FXDLOG	F0
	F1	F2	F4095	F4096	F8	GRLOG	INTEX	INTEXF	INTMC	INTPR
	INTPRL	INTRC	IPUADDR	LOCK	LPUADDR	LPUADDRX	L1	L8	MCHEK	MCNPSW
	MCOPSW	MFAMASK	MICBLOK	MICEVMA3	MICEVMA4	MICIOSW	MICPMAV	MICSIZE	MSSFMASK	NOADD
	PAGCORE	PAGE4K	PAGINVAL	PAGPGSWP	PERADD	PERCODE	PERMODE	PMAAVAIL	PMAGUEST	PMAMODE
	PMAON	PMASTAT	PREFIXA	PREFIXB	PRNPSW	PROCIO	PROCIPL	PROPSW	PSA	QUANTUMR
	RDCBLKMX	RDCBLOK	RDEVBLOK	RDEVFTR	RDEVMDL	RDEVMD00	RDEVMD02	RDEVRDC	REGNCODE	RUNCR0
	RUNCR1	RUNPSW	RUNUSER	R0	R1	R10	R11	R12	R13	R14
	R15	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA
	SAVEREGS	SAVER0	SAVER1	SAVER2	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK8	SEGINV	SEGPAGE
	SIGEMS	SIGXC	SPMPFX	SPXCR6	SYNCMASK	SYSTEM	TEMPR11	TEMPR2	TEMPR4	TIMEDISP
	TIMER	TRACCURR	TRACEND	TRACFLG1	TRACPROC	TRACSTRT	TRACSVCR	TRAC03	TRANMODE	TRCPGM
	TRQBFPNT	TRQBLOK	TRQBTOD	TRQBVAL	TYP2305	TYP2314	TYP3330	TYP3340	TYP3350	TYP3375
	TYP3380	VMBLOK	VPMAAVAI	WAIT	XCMASK	XPAGNUM	X40FFS	Y0	Y2	Y4
	Y6	ZEROES								

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
DMKPRG	ADSPCH	ALOKSY	APSTAT1	APTRAN	APUOPER	AQCNWT	AVMREAL	BRING	CODE	CPABEND
	CPCREG0	CPCREG8	CPEXADD	CPEXBLOK	CPEXREGS	CPRUN	CPSTATUS	CPSTAT2	CPSTAT5	CPSUPER
	CP370EON	C0	C1	C6	C7	C8	DEFER	DMKCFMBK	DMKDMPDK	DMKDSPA
	DMKDSPB	DMKDSPCH	DMKDSPRU	DMKFPS	DMKLOKDF	DMKPERIL	DMKPMAIN	DMKPRVLG	DMKPSAP0	DMKPTRAN
	DMKQCNWT	DMKREIPL	DMKSCHQ1	DMKSCHQ2	DMKSTKDE	DMKTMRPT	DMKTRCPG	DMKVATAB	DMKVATPF	DMKVATPX
	DMKVATSX	DMKVAULA	DMKVFRIN	DUMPSAVE	ERRMSG	EXTMODE	FF	FFS	F0	F1
	INTPR	INTPRL	LAP370E	LOCK	LPUADDR	MONCLASS	MONCODE	MONREGS	MVSA370E	NORET
	PERADD	PERADDR	PERBLOK	PERCODE	PERDATON	PERFLAG	PMAGUEST	PMASTAT	PREFIXA	PREFIXB
	PRNPSW	PROBMODE	PROB370E	PROPSW	PROTBLOK	PROTERR	PROTFLAG	PROTPRGL	PROTPSW	PROTRCNT
	PROTREI	PROTREIL	PROTTRAN	PSA	QUANTUM	QUANTUMR	RAPF	RUNCRINV	RUNCR0	RUNCR1
	RUNPSW	RUNUSER	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SAS	SASPF
	SSEB	SSEBC	SSEF	STOACTV	STOBLOK	STOFLAG	STOSHCR1	SVCNPSW	SVCOPSW	TEMPR1
	TEMPR10	TEMPR12	TEMPR13	TEMPR14	TEMPR15	TEMPSAVE	TIMEDISP	TIMER	TRACCURR	TRACEND
	TRACFLG1	TRACPROC	TRACSTRT	TRACSVCR	TRAC03	TRANMODE	TRAPCR8	TRAPVT	TRCPGM	TREXADD
	VECAVAIL	VECF	VECOPVF	VECSAOK	VECSTAT	VECUSER	VFAULT	VMBLOK	Y0	Y2
	Y4	Y6	ZEROES							
DMKPRV	ADSPCH	ADTRANS	AFREEP	ALOKSY	AMCHAREA	APSTAT1	APSTAT2	APSTAT4	APTRAN	APUOPER
	AVMREAL	BRING	CHANID	CLASDASD	CLASFBA	CPCREG0	CPEXADD	CPEXBLOK	CPEXREGS	CPEXSIZE
	CPMCHSE	CPMICON	CPPTLBR	CPRUN	CPSPMODE	CPSTATUS	CPSTAT2	CPSTAT4	CPUID	CPUMCELL
	CPUMODEL	CPUSER	CPUVERSN	CSADDR	CSSFEAT	C0	C1	C14	C15	C6
	C7	DEFER	DMKDSPA	DMKDSPB	DMKDSPCH	DMKDSPRU	DMKFREEP	DMKHVCAL	DMKIUAEP	DMKLOKDF
	DMKLOKSY	DMKMHVSM	DMKMPOPX	DMKMPOSP	DMKPERIL	DMKPMAWD	DMKPMAXF	DMKPMAXS	DMKPRGSM	DMKPRWIP
	DMKPRWTB	DMKPRWTP	DMKPRWXK	DMKPSAFP	DMKPSAP0	DMKPSASP	DMKPTRAN	DMKPTTDM	DMKSCNVU	DMKSTKCP
	DMKSTKDE	DMKTMR	DMKTMRCC	DMKTMRSP	DMKTMRTN	DMKTRCPB	DMKTRCPV	DMKVATAB	DMKVATAT	DMKVATEX
	DMKVAULA	DMKVAURN	DMKVFRSV	DMKVSIEX	DMKVSIVS	EXTMODE	EXTRACTL	FFS	F0	F1
	F15	F16	F240	F4	F4095	F4096	F5	F6	F60	F7
	F8	INTPR	INTPRL	IPUADDR	LOCK	LOCKSAV	LPUADDR	L2	L4	MCHAREA
	MCHMODEL	MICBLOK	MICCREG0	MNCLINST	MNCOSIM	MOD3090	MPFEAT	OLDKEYOP	PAGE2K	PERADDR
	PERBLOK	PERCR9	PERDATON	PERFLAG	PERGPRS	PERSALT	PREFIXB	PRNPSW	PROBMODE	PROCIO
	PROCIPL	PROPSW	PSA	RCHBLOK	RCHDISA	RCHSTAT	RCHSTIDC	RCUBLOK	RCUCHA	RCUCHB
	RCUCHC	RCUCHD	RCUPRIME	RCUSUB	RCUTYPE	RDEVBLOK	RDEVCUA	RDEVPS	RDEVSTA3	RUNCR0
	RUNCR1	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SEG1M	SHRFLAG	SHRNOPRT
	SHRTABLE	SWPFLAG	SWPKEY1	SWPSHR	TEMPR0	TEMPR4	TEMPR5	TEMPSAVE	TRANMODE	TREXADD
	TREXFLAG	TREXIN1	TREXNDSP	TREXNSI	TREXT	VACOVFR	VECAVAIL	VECF	VECSTAT	VECUSER
	VMBLOK	VMSIZE	XKEYMODE	XPAGNUM	ZEROES					
DMKPRW	ACORETBL	ADSPCH	AP	APSTAT1	APSTAT2	APTRAN	APUOPER	AVMREAL	BALR0	BRING
	CORFLAG	CORSHARE	CORSWPNT	CORTABLE	CPPTLBR	CPSEGPRT	CPSIMLTB	CPSPMODE	CPSTAT2	CPSTAT4
	CP370EON	C1	DEFER	DMKDSPA	DMKDSPCH	DMKPMAWD	DMKPMAXS	DMKPRGSM	DMKPRVXI	DMKPRVXR
	DMKPRVXS	DMKPSAFP	DMKPSAP0	DMKPSASP	DMKPTRAN	DMKSYSRV	DMKVATAB	DMKVAULA	EXTMODE	F0
	F15	F240	F4	F4095	F4096	F60	F8	INTPR	LOCK	MNCLINST
	MNCOSIM	MP	PERADDR	PERBLOK	PERCR9	PERGPRS	PERSALT	PREFIXB	PRNPSW	PSA
	R0	R1	R10	R11	R12	R14	R15	R2	R3	R4
	R5	R6	R7	R8	R9	SEGPROT	SHRFLAG	SHRNOPRT	SHRTABLE	SWPCHG1
	SWPFLAG	SWPKEY1	SWPKEY2	SWPREF1	SWPSHR	TRANMODE	VMBLOK	VMSIZE	XKEYASST	XKEYMODE
	XPAGNUM	ZEROES								
DMKPSA	ACORETBL	AP	APSTAT1	APUOPER	BALR0	BALR1	BALR15	BALR2	BALR7	CORFLAG
	CORSHARE	CORSWPNT	CORTABLE	CPABEND	CPSEGPRT	CPSPMODE	CPSTAT2	CPSTAT4	CP370EON	C1
	DMKCPE	DMKDMPDK	DMKDSPCH	DMKEXTSP	DMKFREE	DMKFREEP	DMKFRET	DMKLOKFR	DMKLOKRM	DMKLOKSP
	DMKLOKSY	DMKLOKVM	DMKMPORS	DMKPRVMA	DMKPTRAN	DMKPTRLK	DMKPXA	DMKPXADL	DMKPXADQ	DMKPXATL
	DMKQCNWT	DMKRIOCC	DMKRIOCH	DMKRIOCT	DMKRIOCU	DMKRIODC	DMKRIODV	DMKRIOPR	DMKRIOPU	DMKRIORD

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	DMKRIOUC	DMKRSPAC	DMKRSPLP	DMKRSPPR	DMKRSPPU	DMKRSPRD	DMKRSPSP	DMKSYSAC	DMKSYSCS	DMKSYSLC
	DMKSYSOP	DMKSYSSP	DMKSYSVM	DMKTMRSN	DPAPUOP	DPMICON	DPOKTLB	DPSEGPRT	DUMPSAVE	EXTMODE
	F15	F2	F240	F4096	F60	F8	LOCK	MP	PAGCORE	PAGSHR
	PREFIXA	PREFIXB	PSA	PSAEXT	R0	R1	R11	R12	R14	R15
	R2	R7	SAS	SEGFLAG	SEGINV	SEGPROT	SEGTABLE	SHRFLAG	SHRNOPRT	SHRTABLE
	SPMPFX	SWPFLAG	SWPPAG	SWPVPAGE	TRANMODE	TRAPCP	VMBLOK	XKEYMODE	X2048BND	ZEROES
DMKPST	ACORETBL	AFREE	AFRET	ALOCBLOK	ALOCCHN	ALOCCSR	ALOCCYL1	ALOCCYL2	ALOCFLG	ALOCMAP
	ALOCMAX	ALOCNPAG	ALOCPAGE	ALOCPAVL	ALOCPMAX	ALOCPNT	ALOCPUSE	ALOCRCUU	ALOCRDEV	ALOCRWRT
	ALOCSW	ALOCTMSI	ALOCUSED	AP	APSTALOC	APSTAT1	APUOPER	ASYSVM	CORCP	CORFLAG
	CORTABLE	CORVM	CPUID	DMKCVTBD	DMKDSPNP	DMKERMSG	DMKFREE	DMKFRET	DMKMHCCP	DMKPGTCP
	DMKPGTDL	DMKPGTDM	DMKPGTPB	DMKPGTPI	DMKPGTPN	DMKPGTPT	DMKPGTPU	DMKPGTSB	DMKPGTSI	DMKPGTSL
	DMKPGTST	DMKPGTSU	DMKPTRFR	DMKPTTFT	DMKSYSPG	DMKSYSSW	DMKSYSSZ	DMKXSTOR	FFS	FLAGS
	F1	F4	F4096	F8	HCBLOK	HCMSFCW	HCMSFDB	HCRMAP	HCRSCP	HCSIZE
	LOCK	MP	PREFIXB	PSA	PSTRCBYT	PSTRCPAG	PSTRCPG$	RECBAK	RECBLOK	RECCYL
	RECMAP	RECMAX	RECPNT	RECSIZE	RECUSED	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SAVEWRK1	SAVEWRK2	SAVEWRK4	SAVEWRK9	SCCBIBLK	SCCBIMAP	SCCBINUM
	SCCBKMAX	SCCBLEN	SCCBLOK	SCCBRESP	SCCB0010	SYSPALOC	SYSPATYP	SYSPCNT	SYSPCYL1	SYSPCYL2
	SYSPFLG	SYSPFLG2	SYSPFPNT	SYSPLIST	SYSPPAG	SYSPPST	SYSPVLEN	SYSPVLST	SYSPVOL	SYSPXST
DMKPTR	ACORETBL	ADSPCH	AEXTSP	AFREE	AFREEP	AFRET	ALOCBLOK	ALOCPRD	ALOCPSIO	ALOKRM
	ALOKSP	ALOKSY	AP	APAGCP	APSTAT1	APTRAN	APUOPER	AQCNWT	ASYSVM	AVMREAL
	BALRSAVE	BALR15	BALR2	BRING	CONTINUE	CONTROL	CORBPNT	CORCFLCK	CORCP	CORFLAG
	CORFPNT	CORFREE	CORIOLCK	CORLCNT	CORPGPNT	CORSHARE	CORSWAP	CORSWPNT	CORTABLE	CORVM
	CPEXADD	CPEXBLOK	CPEXFPNT	CPEXMISC	CPEXREGS	CPEXR0	CPEXR11	CPEXR12	CPEXR2	CPEXR9
	CPEXSIZE	CPSTAT4	CPXSTOR	C1	DEFER	DMKCFMBK	DMKDSPCH	DMKDSPRU	DMKFREAP	DMKFREE
	DMKFREEP	DMKFRET	DMKFRTTE	DMKFRTTR	DMKLOKDF	DMKPAGIO	DMKPAGQ	DMKPGTPB	DMKPGTSB	DMKPGUAL
	DMKPGUSW	DMKPMAIO	DMKPTSAD	DMKPTSAE	DMKPTTCL	DMKPTTFT	DMKPTTPM	DMKQCNWT	DMKREIPL	DMKSCHDL
	DMKSCHN1	DMKSCHN2	DMKSELCT	DMKSELFE	DMKSTKCP	DMKSTKDE	DMKSTKMP	DMKSTKOP	DMKSTRAN	DMKSWAIS
	DMKSWAPR	DMKSYSCS	DMKSYSRM	DMKSYSSZ	FF	FFS	FREEWORK	F0	F1	F15
	F16	F2	F4095	F4096	IOERETN	IOKEEP	IPUADDRX	KEYCHG	LOCK	LOCKSAV
	LPUADDR	LPUADDRX	L2048	L4	MICBLOK	MICVTMR	MNCLSWAP	MNCODEIS	MNCODSIS	MP
	NORESTR	NORET	NOSIGP	PAGACT	PAGBMP	PAGCORE	PAGINVAL	PAGRBITS	PAGREF	PAGSHR
	PAGTABLE	PAGTSWP	PGREAD	PREFIXA	PROTBLOK	PROTERR	PROTFLAG	PROTPAGE	PROTRCNT	PROTREI
	PROTREIL	PSA	PSAEXT	PSTPINPP	PSTPINSW	PXA	R0	R1	R10	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	SAVEAREA	SAVEPROC	SAVEREGS	SAVERETN	SAVER0	SAVER1	SAVER12	SAVER13
	SAVER2	SAVER3	SAVER7	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK8
	SAVEWRK9	SEGINV	SEGPAGE	SHRFLAG	SHRNOPRT	SHRSEGCT	SHRSGPRT	SHRTABLE	SIGEMS	SIGEXT
	SIGQUI	SIGRES	SIGXC	SPECIALV	SSBALLOC	SSBBPNT	SSBCODE	SSBCYL	SSBEFLG	SSBEINVL
	SSBENTRL	SSBENTRY	SSBENUM	SSBEPGS	SSBEXTND	SSBFLAG	SSBFPNT	SSBHEADL	SSBLOK	SSBNDLCT
	SSBNPGS	SSBOLD	SSBPSTOR	SSBTRANS	SSBUPGS	SSBVMSCB	SSBVPAGE	SWPALLOC	SWPAPP	SWPCHG1
	SWPCODE	SWPCYL	SWPFLAG	SWPFLAG2	SWPKEY1	SWPPAG	SWPPSTOR	SWPRECMP	SWPSHR	SWPTABLE
	SWPTRANS	SWPVPAGE	SYSTEM	TEMPR1	TEMPR2	TEMPR3	TIMER	VFAULT	VMBLOK	XKEYMODE
	XPAGNUM	XTNDLOCK	ZEROES							
DMKPTS	ACORETBL	ADSPCH	AFREEP	ALOKRM	ALOKSP	AP	APSTAT1	APSTAT2	APUOPER	ASYSVM
	AVMREAL	CODE	CORBPNT	CORCP	CORFLAG	CORFLUSH	CORFPNT	CORPGPNT	CORSHARE	CORSWAP
	CORSWPNT	CORTABLE	CORVM	CPEXADD	CPEXBLOK	CPEXFPNT	CPEXMISC	CPEXR0	CPEXR11	CPEXSIZE
	CPPTLBR	CPSTAT4	CPXSTOR	DMKCVTAB	DMKDSPCH	DMKFREEP	DMKPAGQ	DMKPTRFQ	DMKPTRQ2	DMKPTRSC
	DMKPTRU1	DMKSTKCP	DMKVATSI	FF	FREESAVE	F0	F1	F16	F6	KEYCHG
	KEYREF	LOCK	LOCKSAV	LPUADDR	MNCLSWAP	MNCODLSO	MP	PAGACT	PAGCORE	PAGINVAL
	PAGREF	PAGSHR	PAGTABLE	PREFIXA	PREFIXB	PSA	R0	R1	R10	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7

```
MODULE     EXTERNAL REFERENCES (LABELS AND MODULES)

           R8        R9        SAVEAREA  SAVEREGS  SEGINV    SEGPAGE   SHRFLAG   SHRNOPRT  SHRSEGCT  SHRSGPRT
           SHRTABLE  SSBFLAG   SSBFPNT   SSBLOK    SSBOLD    SWPFLAG   SWPREF1   SWPREF2   SWPVPAGE  TEMPR0
           TEMPR4    TRACCURR  TRACEND   TRACFLG1  TRACPROC  TRACSTRT  TRACSVCR  TRAC08    TRCPTSRS  VMBLOK
           XKEYMODE  XPAGNUM

DMKPTT     ACORETBL  AEXTSP    AFREE     ALOKRM    ALOKSP    ALOKSY    AP        APSTAT1   APUOPER   ASYSVM
           BALRSAVE  BALR0     CONTROL   CORBPNT   CORCP     CORFLAG   CORFLUSH  CORFPNT   CORFREE   CORPGPNT
           CORSHARE  CORSWPNT  CORTABLE  CORVM     CPCREG0   CPEXADD   CPEXBLOK  CPEXFPNT  CPEXMISC  CPEXREGS
           CPEXR7    CPEXSIZE  C0        C1        DMKDSPRU  DMKFREE   DMKLOKDF  DMKPGUVG  DMKPTRFN  DMKPTRFQ
           DMKPTRF1  DMKPTRF2  DMKPTRN2  DMKPTRQC  DMKPTRQ2  DMKPTSAE  DMKSTKCP  DMKSTKDE  DMKSYSCS  DMKVATSI
           FF        F1        F16       F4096     F8        IPUADDRX  LOCK      LOCKSAV   LOKSYS    LPUADDR
           MP        NOINVPTE  NORESTR   NOSIGP    OPPRSTRT  PAGCORE   PAGE2K    PAGINVAL  PAGSHR    PAGTABLE
           PREFIXA   PSA       RUN370E   R0        R1        R10       R11       R12       R13       R14
           R15       R2        R3        R4        R5        R6        R7        R8        R9        SAVEAREA
           SAVEREGS  SAVER1    SAVER2    SAVER3    SAVEWRK1  SAVEWRK2  SEGINV    SEGPAGE   SEG1M     SHRFLAG
           SHRNOPRT  SHRSGPRT  SHRTABLE  SIGEMS    SIGQUI    SIGRES    SIGXC     SWPFLAG   SWPPAG    SWPSEGNO
           SWPTABLE  SWPVPAGE  TEMPR8    TRANMODE  VMBLOK    XTNDLOCK

DMKPXA     CPXTRANS  DSPA      DSPB      DSPCH     DSPRQ     DSPRU     DSPWA     FREE      NUMPOOLS  PSTPINPP
           PSTPINSW  PSTPOUPP  PSTPOUSW  PXAEND    RQLOCK    SPBOUND   SUBHEAD   SUBSTLA   SUBTABLE  TRLLABEL
           TRLLOCK   TRLSTL    VMBLOK

DMKPXB     CPXTRANS  DSPA      DSPB      DSPCH     DSPRQ     DSPRU     DSPWA     FREE      NUMPOOLS  PSTPINPP
           PSTPINSW  PSTPOUPP  PSTPOUSW  PXAEND    RQLOCK    SPBOUND   SUBHEAD   SUBSTLA   SUBTABLE  TRLLABEL
           TRLLOCK   TRLSTL    VMBLOK

DMKQCN     ADSPCH    AFREE     AFRET     ANYWRITE  AP        APSTAT1   APTRAN    APUOPER   AQCNWT    ASYSOP
           BLANK     BRING     CLASGRAF  CLASSPEC  CLASTERM  CONADDR   CONCNT    CONDATA   CONDIAG   CONDWC
           CONFLAGS  CONFLG2   CONFSS    CONLED    CONNCB    CONOUTPT  CONPARM   CONPARM2  CONSTAT   CONTASK
           CONTGMXB  CONTGMXD  CONTSIZE  CONTSKSZ  CONUSER   CONVDEVB  CONWORK1  CONWORK2  CPID      C1
           DEFER     DFRET     DMKCVTDT  DMKDSPCH  DMKFREE   DMKFRERC  DMKFRET   DMKLOKSW  DMKMSGIU  DMKPTRAN
           DMKQCOCS  DMKQCONQ  DMKQCQED  DMKSCNAU  DMKSCNVU  FFS       F0        F2        F255      F4095
           F9        GRAFDEV   GRTSALEN  IDAHEAD   IDAHSTRT  INHIBIT   LOCK      LOGDROP   LOGHOLD   MNCLRESP
           MNCOWRIT  MP        MSGCPIO   MSGEMSG   MSGIMSG   MSGSCIF   MSGVMIO   NICBLOK   NICCORD   NICHT
           NICSIZE   NICWTH    NOMC      NORET     NOTIME    NOTRESP   OPERATOR  PLALARM   PLDFRET   PLDIAG
           PLERRMSG  PLFLAG1   PLFRET    PLHILITE  PLIMSG    PLINHIBT  PLIST     PLLED     PLLOGDRP  PLLOGHLD
           PLNCB     PLNOAUTO  PLNORESP  PLNORET   PLNOTIME  PLOPERTR  PLPRIOR   PLR2      PLSECUSR  PLSIZE
           PLSVR2    PLSVR20   PLSVR21   PLSVR22   PLSVR23   PLVIRDVD  PLVMGNIO  PSA       RDEVBLOK  RDEVCORD
           RDEVECOL  RDEVEHLT  RDEVHT    RDEVNICL  RDEVSNA   RDEVTYPC  RDEVTYPE  RDEVWTH   RESET     RTYPE
           R0        R1        R10       R11       R12       R13       R14       R15       R2        R3
           R4        R5        R6        R7        R8        R9        SAVEAREA  SAVEREGS  SAVER0    SAVER1
           SAVER11   SAVER2    SAVER3    SAVER4    SAVER7    SAVER8    SAVEWRK1  SAVEWRK2  SAVEWRK3  SAVEWRK5
           SAVEWRK7  SCRCMDR   SECUSER   TIMEDISP  TYPBSC    TYP1050   TYP3277   TYP3278   VCONCTL   VCONFLG2
           VCONNICB  VCONOMSG  VCONRDEV  VMBLOK    VMGENIO   XPAGNUM   XRIGHT16

DMKQCO     ADSPCH    AFREE     AFREEP    AFRET     AP        APSTAT1   APTRAN    APUOPER   BALRSAVE  BALR11
           BLANK     BRING     BUFCNT    BUFFER    BUFSIZE   CLASGRAF  CLASSPEC  CLASTERM  CONADDR   CONADDR2
           CONCCW1   CONCCW4   CONCNT    CONCNTL   CONCNT2   CONDATA   CONFLAGS  CONFLG2   CONFSOP   CONFSS
           CONLNCNT  CONMORE   CONNCB    CONOUTPT  CONPARM   CONPNT    CONRD     CONRESP   CONRETN   CONSPLT
           CONSTAT   CONSYNC   CONTASK   CONTSIZE  CONTSKSZ  CONUSER   CONVDEVB  CONWORK1  CONWORK2  CONWRTRD
           CORFPNT   CORTABLE  CPEXADD   CPEXBLOK  CPEXREGS  CPEXR12   CPEXSIZE  C1        DEFER     DMKCNSIC
           DMKDSPCH  DMKERMSG  DMKFREE   DMKFREEP  DMKFRET   DMKGRDIC  DMKLOKSW  DMKPTRAN  DMKPTTFT  DMKQCPTO
           DMKRGBIC  DMKRNHIC  DMKSCNAU  DMKSCNVU  DMKSTKCP  DMKTBLSF  DMKVCRRD  DMKVCSWT  DMKVSTVP  FFS
           F0        F1        F2        F255      F256      F7        F8        IDA       LOCK      MNCLRESP
           MNCOBRD   MNCOERD   MP        NICBLOK   NICQPNT   NICSIZE   NOTIME    NOTRESP   PLEDIT    PLFLAG1
```

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	PLFRET	PLINHIBT	PLIST	PLLOGDRP	PLLOGHLD	PLNCB	PLNORESP	PLOPERTR	PLPRIOR	PLR2
	PLSIZE	PLSVR2	PLSVR21	PLSVR22	PLSVR23	PLUCASE	PLVIRDVD	PLVMGNIO	PLWRTRD	PSA
	RDEVACTV	RDEVBLOK	RDEVCON	RDEVDROP	RDEVFLAG	RDEVNICL	RDEVSNA	RDEVSNRB	RDEVSTA2	RDEVTYPC
	RDEVTYPE	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVERETN
	SAVER0	SAVER1	SAVER11	SAVER2	SAVER4	SAVER8	SAVESIZE	SAVEWRK1	SAVEWRK2	SAVEWRK3
	SAVEWRK4	SAVEWRK5	SECUSER	SNARBLOK	SNARFG2	SNARINN	SNAROUT	SNARTTY	TIMEDISP	TYPBSC
	VCONCCW2	VCONCNT2	VCONCTL	VCONNICB	VCONRDEV	VMBLOK	XPAGNUM	XRIGHT16		
DMKQCP	ADSPCH	AFREE	AFREEP	AFRET	AP	APTRLK	AQCNWT	BLANKS	CLASSPEC	CLASTERM
	CONTASK	CPEXADD	CPEXBLOK	CPEXREGS	CPEXR11	CPEXR12	CPEXSIZE	DFRET	DMKACODS	DMKCVTAB
	DMKCVTBD	DMKCVTBH	DMKDSPCH	DMKFREE	DMKFREEP	DMKFRET	DMKPTRLK	DMKPTRUL	DMKQCNWT	DMKSCHDL
	DMKSCHRT	DMKSCHST	DMKSCNRD	DMKSCNRN	DMKSTKCP	DMKSYSNM	DMKVCTLO	LOCK	MP	NORET
	OPERATOR	PSA	RDEVBLOK	RDEVPS	RDEVSNA	RDEVSNRB	RDEVSTA3	RDEVTYPC	RDEVTYPE	RETBUF
	RETDLEN	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R8	SAVEAREA	SAVEREGS	SAVEWRK1	SNARBLOK	SNARLUN
	TRQBIRA	TRQBLOK	TRQBSIZE	TRQBUSER	TRQBVAL	TYPBSC	VMBLOK			
DMKQCQ	AFREE	AFRET	ANYWRITE	BALRSAVE	BALR4	BALR5	CLASGRAF	CLASTERM	CONADDR	CONCNT
	CONDATA	CONDCNCL	CONDIAG	CONDLN	CONDRFMT	CONDWC	CONFLAGS	CONFLG2	CONFSS	CONLED
	CONLNCNT	CONLNRES	CONNEWL	CONOUTPT	CONPARM	CONPARM2	CONPNT	CONRESP	CONRETN	CONSFCNT
	CONSPLT	CONSTAT	CONTASK	CONTGMXB	CONTGMXD	CONTSIZE	CONTSKSZ	CONUSER	CONVDEVB	CONWORK
	CONWORK1	CONWORK2	DMKFREE	DMKFRET	DMKTBLSF	DMKVCSWT	D66LNCNT	D66LNLEN	F0	F1
	F10	F2	F256	F3	F9	GRTHILEN	INHIBIT	LOCK	NORET	NOTIME
	NOTRESP	PLALARM	PLDFRET	PLDIAG	PLFLAG1	PLHILITE	PLIST	PLLED	PLLOGDRP	PLLOGHLD
	PLNOAUTO	PLNORESP	PLNORET	PLOPERTR	PLPRIOR	PLSVR2	PLSVR20	PLSVR22	PLVIRDVD	PLVMGNIO
	PSA	RDEVAPLP	RDEVBLOK	RDEVSNA	RDEVSNRB	RDEVTAPL	RDEVTFLG	RDEVTMCD	RDEVTYPC	RDEVTYPE
	RTYPE	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVER2	SAVER3
	SAVER4	SAVEWRK1	SAVEWRK2	SAVEWRK4	SCRCMDR	SECUSER	SF	SNARBLOK	SNARFG2	SNARTTY
	TEMPR0	TEMPR1	TEMPSAVE	TYPBSC	TYPTTY	TYP3066	TYP3277	TYP3278	VMBLOK	XRIGHT16
DMKQVM	ACORETBL	ACTIVTRQ	AFREEP	AFRET	APSTAT1	APUOPER	AQCNWT	ARIOCC	ARIOCH	ARIOCU
	ARIODC	ARIODV	ARIOUC	ASYSOP	ASYSVM	AVMREAL	BALRSAVE	BLANKS	BUFSIZE	CLASGRAF
	CLASTERM	CLASURI	CLASURO	CONCCW1	CONCCW2	CONPNT	CONTASK	CONTSKSZ	CORPGPNT	CORSWPNT
	CORTABLE	CPCREG0	CPCREG8	CPEXADD	CPEXBLOK	CPEXREGS	CPEXR12	CPEXSIZE	CPQVMCU	CPSPMODE
	CPSTATUS	CPSTAT2	CPSTAT4	CPSUPER	CRTEXT	CRTFMT	CUE	C0	C1	C14
	C15	C3	C5	C6	C8	DMKCNSEN	DMKDSPB	DMKDSPQI	DMKDSPVM	DMKERMSG
	DMKFREEP	DMKFRERC	DMKFRET	DMKGRFEN	DMKPMARW	DMKPMATM	DMKPSAP0	DMKQCNWT	DMKRIOCH	DMKSCHQ1
	DMKSCHQ2	DMKSCHRT	DMKSCHST	DMKSCNFD	DMKSCNRD	DMKSCNRU	DMKSCNVD	DMKSTKCP	DMKVATBC	FXDLOG
	F1	F15	F4096	F6	F8	IOBBPNT	IOBCAW	IOBCSW	IOBFPNT	IOBLINK
	IOBLOK	IOBRADD	IOBSIZE	IOBUSER	IOERBLOK	IOEREXT	IOERSIZE	LOCK	MICBLOK	MICEVMA2
	MICSPT	MPFEAT	MVSA370E	NORET	PMAAVAIL	PMAMODE	PMAON	PMASTAT	PSA	QUANTUM
	QUANTUMR	RCHADD	RCHBLOK	RCHBMX	RCHBUSY	RCHSCED	RCHSIZE	RCHSTAT	RCHTYPE	RCUADD
	RCUBLOK	RCUBUSY	RCUCHA	RCUCUBSY	RCUPRIME	RCUSCED	RCUSIZE	RCUSTAT	RCUSUB	RCUTYPE
	RDEVACTV	RDEVAIOB	RDEVAIRA	RDEVALT	RDEVBLOK	RDEVBUZY	RDEVBZCH	RDEVCON	RDEVCTL	RDEVCUA
	RDEVDED	RDEVDISA	RDEVDRAN	RDEVFLAG	RDEVPREP	RDEVSCHD	RDEVSIZE	RDEVSTAT	RDEVSTA2	RDEVSTA4
	RDEVTFLG	RDEVTYPC	RDEVTYPE	RDEVUSER	RSRTNPSW	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SAVEWRK2	SPMPFX	SPXCR6	SWPFLAG	SWPKEY1	SWPKEY2	TIMER
	TRQBFLAG	TRQBFPNT	TRQBLINA	TRQBLOK	TRQBQUE	TRQBSIZE	TRQBTOD	TRQBVAL	TYPBSC	VHD
	VMBLOK	VMSIZE	WAIT	XKEYMODE	X40FFS	Y0	Y2	Y4	Y6	ZEROES
DMKREI	ADSPCH	AFREE	AFRET	APTRLK	BLANKS	BUFCNT	BUFFER	BUFIN	BUFINLTH	BUFNXT
	BUFSIZE	DMKCFGIR	DMKCVTAB	DMKDSPCH	DMKFREE	DMKFRET	DMKPTRLK	DMKPTRUL	DMKSCHST	DMKUDRFU

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	DMKUDRMD	DMKUDRRV	DMKUDRXI	LOCK	PROTATRQ	PROTBEG	PROTBLOK	PROTBUFF	PROTERR	PROTFLAG
	PROTRCNT	PROTREI	PSA	R0	R1	R10	R11	R12	R13	R14
	R15	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA
	SAVEREGS	SAVER2	TRQBIRA	TRQBLOK	TRQBSIZE	TRQBTOD	TRQBUSER	TRQBVAL	VCONCTL	VMBLOK
	ZEROES									
DMKRET	BALRSAVE	PSA	RETBUF	RETDATA	RETDLEN	RETLEN1	RETLEN2	RETNXTD	RETNXTS	R0
	R1	R11	R12	R14	R15	R2	R3	R4	R5	R6
	R7	R8	VMBLOK	ZEROES						
DMKRGA	ADSPCH	AFREE	AFREEP	AFRET	ALOKSP	AP	APSTAT1	APUOPER	ASYSVM	ATTN
	BSCAUSER	BSCBLOK	BSCCNT	BSCCOPY	BSCDCNT	BSCECCW1	BSCECCW2	BSCENQ	BSCETB	BSCFLAG
	BSCFLAG1	BSCFORCE	BSCHALT	BSCIGN	BSCINBID	BSCINDEX	BSCLOG	BSCOPIED	BSCPA1	BSCPCCW1
	BSCPCCW2	BSCPCCW4	BSCRCVD	BSCREAD	BSCREGEN	BSCRESP	BSCRPTR	BSCRSTRT	BSCRVI	BSCSCAN
	BSCSCCW2	BSCSCCW3	BSCSEND	BSCSENSE	BSCSIZE	BSCSIZE1	BSCSPTR	BSCTMRQ	BSCTSTRQ	BSCUCOPY
	BSCUP	BUFCNT	BUFFER	BUFSIZE	CC	CD	CE	CLASTERM	CLEAR	CONACTV
	CONADDR	CONCCW1	CONCCW2	CONCCW3	CONCMD1	CONCNT	CONCNTL	CONDATA	CONDCNT	CONFLG2
	CONFSS	CONLABEL	CONNCB	CONPARM	CONPNT	CONRESET	CONRESP	CONRETN	CONSTAT	CONSTX
	CONSYNXP	CONSYN1	CONTASK	CONTSIZE	CONTSKSZ	CONUSER	CONWRT	CPEXADD	CPEXBLOK	CPEXR0
	CPEXSIZE	CRTEXT	CRTEXTSZ	CRTFSII	CRTFSSA	CRTPPA1	DE	DMKBLDVM	DMKCFMBK	DMKCFMEN
	DMKCVTAB	DMKCVTBH	DMKDSPCH	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKGRCQR	DMKGRTB	DMKIOERN
	DMKLOKIO	DMKLOKSW	DMKQCOCL	DMKQCOET	DMKQCPTO	DMKRGBCL	DMKRGBCO	DMKRGBEN	DMKRGBMT	DMKRGBPL
	DMKRGBRE	DMKRGBSL	DMKRGBSN	DMKRGBUP	DMKRGC	DMKRGDOB	DMKRGEIO	DMKRGESK	DMKSCHRT	DMKSCHST
	DMKSCNRD	DMKSCNRU	DMKSCNVU	DMKSTKCP	DMKSTKIO	DMKVIOIN	FTRDIAL	F1	F2	F6
	F8	IDAHEAD	IDAHSTRT	IDAHWRK1	IDAHWRK2	INHIBIT	IOBCAW	IOBCC3	IOBCSW	IOBFATAL
	IOBIOER	IOBIRA	IOBLINK	IOBLOK	IOBMISC	IOBMISC2	IOBRADD	IOBSIZE	IOBSPEC	IOBSTAT
	IOBUNSL	IOBUSER	IOBVADD	IOERBLOK	IOEREXT	IOERSIZE	LOCK	LOGDROP	LOGHOLD	LPUADDR
	MP	NICADVF	NICALRM	NICAPL	NICATRB	NICBLOK	NICCORD	NICCPNA	NICDIAG	NICDISB
	NICDMSG	NICDXSC	NICENAB	NICFLAG	NICFMT	NICHOLD	NICLGDRP	NICLGRP	NICMORE	NICNAME
	NICNTRL	NICPOLL	NICQPNT	NICQRY	NICRATTD	NICRATTN	NICRDED	NICREAD	NICRFLG	NICRFLG1
	NICRSPL	NICRUNN	NICSELT	NICSIO	NICSIZE	NICSTAT	NICTERM	NICTEXT	NICTMCD	NICTRQ
	NICTYPE	NICUSER	NICUSEWA	NICVDEVB	NICWSF	NIC3275	ONEENT	PSA	RDEVADVF	RDEVBLOK
	RDEVBSC	RDEVCON	RDEVDISA	RDEVDISB	RDEVENAB	RDEVFLAG	RDEVFTR	RDEVMAX	RDEVNICL	RDEVNRDY
	RDEVPDLY	RDEVRSVD	RDEVSTAT	RDEVTYPC	RDEVTYPE	RDEVWAII	R0	R1	R10	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	SAVEAREA	SAVER2	SILI	SKIP	SVMUNLOK	TEMPR1	TEMPR15	TEMPR2
	TEMPR5	TEMPR6	TEMPSAVE	TIMEDISP	TRQBCHIO	TRQBCPQ	TRQBCRT	TRQBDEV	TRQBFLG2	TRQBIRA
	TRQBLINA	TRQBLINE	TRQBLOK	TRQBPA1R	TRQBPOLL	TRQBSIZE	TRQBUSER	TRQBVAL	TRQNAME	TYPBSC
	UC	UE	VCONBRK	VCONCTL	VCONOBRK	VCONOPT	VCONRINX	VCON3270	VMBLOK	VMGENIO
	XRIGHT16	XRIGHT24	ZEROES							
DMKRGB	ADSPCH	AFREE	AFREEP	AFRET	ALARM	AP	APSTAT1	APUOPER	ASYSVM	ATTN
	BLANK	BSCAUSER	BSCBLOK	BSCCNT	BSCFLAG	BSCFLAG1	BSCHALT	BSCINBID	BSCINDEX	BSCPCCW1
	BSCPCCW2	BSCPCCW4	BSCRCVD	BSCREAD	BSCRESP	BSCRROBN	BSCSCAN	BSCSCCW1	BSCSCCW2	BSCSCCW3
	BSCSEL	BSCSHUT	BSCSIZE	BSCSIZE1	BSCSIZE2	BSCSPTR	BSCUP	CC	CD	CONACTV
	CONADDR	CONCCW1	CONCCW2	CONCCW3	CONCMD	CONCMD1	CONCNT	CONCNTL	CONDATA	CONDCNCL
	CONDLE	CONDLE1	CONDLN	CONDRFMT	CONDWC	CONEWA	CONEWRT	CONFLAGS	CONFLG2	CONFSOP
	CONFSS	CONLABEL	CONLED	CONLNCNT	CONLNRES	CONNCB	CONOUTPT	CONPARM	CONPARM2	CONPNT
	CONPPA1	CONRESP	CONRETN	CONSBADR	CONSTAT	CONSTX	CONSTX1	CONSYNC	CONSYNXP	CONSYN1
	CONTASK	CONTSIZE	CONTSKSZ	CONUSER	CONWCC	CONWORK2	CONWSF	CPEXADD	CPEXBLOK	CPEXSIZE
	CRTEXT	CRTEXTSZ	CRTFSII	CRTFSSA	CRTPPA1	DE	DMKCFMAT	DMKCVTAB	DMKDSPCH	DMKFREE
	DMKFREEP	DMKFRET	DMKGRAOT	DMKGRCUP	DMKGRCWC	DMKGRTB	DMKGRTFD	DMKGRTFM	DMKGRT12	DMKIOSHA
	DMKIOSQR	DMKLOKSW	DMKQCOET	DMKRGAIN	DMKRGATM	DMKRGDOI	DMKSCHRT	DMKSCHST	DMKSCNRD	DMKSTKCP
	DMKSTKIO	DMKTBLGR	DMKTBLGT	DMKTBLRG	DMKVIOIN	FTRDIAL	F1	F2	F255	F256
	F3	F4	IDAHEAD	IDAHSTRT	INHIBIT	IOBCAW	IOBCP	IOBCSW	IOBFATAL	IOBFLAG

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	IOBIOER	IOBIRA	IOBLINK	IOBLOK	IOBMISC	IOBMISC2	IOBRSTRT	IOBSIZE	IOBSPEC	IOBSTAT
	IOBUNSL	IOBUSER	IOBVADD	IOERBLOK	IOEREXT	IOERSIZE	LOCK	LOGDROP	LOGHOLD	MP
	NICADFF	NICADVF	NICAINH	NICALRM	NICAPL	NICATRB	NICAWSF	NICBLOK	NICCDCNT	NICCORD
	NICDIAG	NICDISB	NICDTYPE	NICD3275	NICD3276	NICD3277	NICD3278	NICD3284	NICECOL	NICEHLT
	NICEWO	NICFLAG	NICFMT	NICHOLD	NICHT	NICMORE	NICNTRL	NICPOLL	NICQDONE	NICQPNT
	NICQRY	NICRATTD	NICRATTN	NICRDED	NICREAD	NICRFLG	NICRUNN	NICSELT	NICSIO	NICSIZE
	NICSTAT	NICTEXT	NICTMCD	NICTRQ	NICTYPE	NICUSER	NICUSEWA	NICVDEVB	NICWSF	NICWTH
	NIC14B	NIC3274	NIC3276	PRIORITY	PSA	RDEVAIOB	RDEVBLOK	RDEVBSC	RDEVDED	RDEVDISA
	RDEVDISB	RDEVENAB	RDEVFLAG	RDEVFTR	RDEVMAX	RDEVNICL	RDEVNRDY	RDEVPEND	RDEVRSVD	RDEVSTAT
	RDEVWAII	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER2
	SCRCMDR	SILI	TAB	TIMEDISP	TRQBCHIO	TRQBCPQ	TRQBCRT	TRQBDEV	TRQBFLG2	TRQBIRA
	TRQBLINE	TRQBLOK	TRQBPA1R	TRQBSIZE	TRQBUSER	TRQBVAL	TRQINTEG	VCONANF2	VCONCTL	VCONOPT
	VCON3270	VMBLOK	VMGENIO	XRIGHT16	ZEROES					
DMKRGC	AFREE	AFREEP	AFRET	AP	APTRAN	AQCNWT	ASYSVM	ATTN	BALRSAVE	BLANKS
	BRING	BSCBLOK	BSCCNT	BSCCOPY	BSCDCNT	BSCENQ	BSCETB	BSCFLAG	BSCFLAG1	BSCIGN
	BSCINBID	BSCINDEX	BSCOPIED	BSCPA1	BSCPCCW3	BSCPCCW4	BSCREAD	BSCRESP	BSCRPTR	BSCRSTRT
	BSCRVI	BSCSCCW1	BSCSCCW2	BSCSEND	BSCSENSE	BSCSIZE1	BSCSPTR	BSCTSTRQ	BSCUCOPY	BSCUP
	BUFCNT	BUFFER	BUFINLTH	BUFSIZE	CC	CD	CONACTV	CONADDR	CONCCW1	CONCCW2
	CONCCW3	CONCNT	CONCNTL	CONDATA	CONDLE1	CONFSOP	CONFSS	CONLABEL	CONPARM	CONPNT
	CONRESET	CONRESP	CONRETN	CONSTAT	CONSTX1	CONSYN	CONTASK	CONTINUE	CONTSIZE	CONTSKSZ
	CONUSER	CRTFSII	CRTFSSA	CRTPPA1	C1	DEFER	DMKCFMAT	DMKCFMBK	DMKCNTED	DMKCVTBD
	DMKCVTHB	DMKFREE	DMKFREEP	DMKFRET	DMKGRAOT	DMKGRCUP	DMKGRTAB	DMKGRTAC	DMKGRTAI	DMKGRTPF
	DMKGRTP6	DMKGRT12	DMKPTRAN	DMKQCNWT	DMKQCOET	DMKRETGT	DMKRETPT	DMKRGAIN	DMKRGASP	DMKRGBCL
	DMKRGBCO	DMKRGBMT	DMKRGBRE	DMKRGBUP	DMKSCHRT	DMKSTKIO	DMKTBLGR	DMKTBLGT	DMKTBLRG	DMKTBLUP
	DMKTBMAI	DMKTBMTI	DMKTBMXI	DMKTBMZI	DMKVIOIN	EDIT	FTRDIAL	F0	F1	F2
	F255	F3	F4095	F6	HILIGHT	INHIBIT	IOBCAW	IOBCSW	IOBIRA	IOBLINK
	IOBLOK	IOBMISC2	IOBREMOT	IOBSIZE	IOBSPEC	IOBSPEC5	IOBUNSL	IOBUSER	IOBVADD	LOCK
	MNCLRESP	MNCOERD	MP	NICADVF	NICAINH	NICALRM	NICAPL	NICATRB	NICAWSF	NICBLOK
	NICCARD	NICCORD	NICCPNA	NICDIAG	NICDISA	NICDISB	NICDMSG	NICDTYPE	NICDXSC	NICD3275
	NICD3277	NICD3284	NICECOL	NICEHLT	NICENAB	NICFLAG	NICFMT	NICHOLD	NICHT	NICLOGDT
	NICMORE	NICNAME	NICNTRL	NICQDONE	NICQPNT	NICQRY	NICRATTN	NICRDED	NICREAD	NICRFLG
	NICRFLG1	NICRSPL	NICRUNN	NICSIZE	NICSTAT	NICTEXT	NICTMCD	NICTRQ	NICTYPE	NICUSER
	NICUSEWA	NICVDEVB	NICWSF	NICWTH	NIC3275	NOMC	NORET	NOTIME	NOTRESP	PFDATA
	PFDCMD	PFDCPAD4	PFDCPYSP	PFKADDR	PFKFLAG	PFKIMM	PFKLNG	PFKRET	PFKTABLE	PREFIXA
	PSA	RDEVBLOK	RDEVFTR	RDEVMAX	RDEVNICL	RDEVSTAT	RDEVWAII	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SAVEAREA	SAVER0	SAVER2	SILI	SYSTEM	TAB	TEMPR0
	TEMPR3	TREXFLAG	TREXMOR	TREXT	TRQBCLIN	TRQBFLG2	TRQBLINA	TRQBLINE	TRQBLOK	TRQBPA1R
	TRQINTEG	UCASE	VCONBRK	VCONCTL	VCONOBRK	VCONOPT	VCONRBUF	VCONRCNT	VCONRDSZ	VCON3270
	VMBLOK	VMGENIO	XRIGHT16	XTNDLOCK						
DMKRGD	AP	BALRSAVE	BLANK	BLKLEN	BSCBLOK	CC	CD	CONCCW1	CONCCW2	CONCNT
	CONDATA	CONDLE1	CONFLAG	CONFLG2	CONFSS	CONLABEL	CONNCB	CONTASK	CONWORK	CONWSF
	DLE	ETB	ETX	F1	F255	IDA	IDAENTRY	IDAEWRD1	IDAHCURR	IDAHEAD
	IDAHSTRT	IOBLOK	LOCK	MP	NICAWSF	NICBLOK	NICRFLG	ONEENT	PSA	RTNBLOCK
	RTNNOCTL	R0	R1	R10	R11	R12	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SILI	STX	VMBLOK	WR
	XRIGHT16	XRIGHT24								
DMKRGE	AFREE	BALRSAVE	BALR6	BSCBLOK	BSCSENSE	CD	CE	CONADDR	CONFLG2	CONNCB
	CONTASK	DE	DMKFREE	DMKSTKIO	DMKVIOIN	F1	IDAHEAD	IDAHWRK1	IDAHWRK2	IOBCSW
	IOBIOER	IOBIRA	IOBLINK	IOBLOK	IOBREMOT	IOBSIZE	IOBSPEC	IOBSPEC5	IOBUNSL	IOBUSER
	IOBVADD	IOERBLOK	IOERCSW	IOERDATA	IOERLEN	IOERSIZE	LOCK	L1	L2	NICBLOK

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	NICVDEVB	PSA	RDEVATT	RDEVBLOK	RDEVSNA	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SKIP	UC	VCONCTL	VCONRBYT	VCONRFLD	VCONRIND	VMBLOK
	XRIGHT16									
DMKRND	DUMP	INPUT	ON	R0	R1	R10	R11	R12	R13	R14
	R15	R2	R3	R4	R6	R7	SAVEAR			
DMKRNH	ABORT	ADSPCH	AFREE	AFREEP	AFRET	ALARM	AP	APSTAT1	APUOPER	AQCNWT
	ASYSVM	ATTN	BALRSAVE	BLANKS	BUSOUT	BUSY	CACTDEV	CACTLIN	CACTLTR	CC
	CCDESMD	CDC	CHC	CKPBITS	CKPBKSZ	CKPBLOK	CKPNAME	CKPRMAX	CKPSIZE	CLASSPEC
	CMDREJ	CNTLBTU	CODE	CONACTV	CONADDR	CONCCW1	CONCCW2	CONCCW3	CONCNT	CONCNTL
	CONCOMND	CONDATA	CONDCNT	CONDEST	CONESCP	CONEXTR	CONFLAG	CONOUTPT	CONPARM	CONPNT
	CONRESP	CONRETN	CONRTAG	CONRTRY	CONSPLT	CONSRID	CONSTAT	CONSYNC	CONSYSR	CONTACT
	CONTASK	CONTCMD	CONTSIZE	CONTSKSZ	CONUSER	CPEXADD	CPEXBLOK	CPEXSIZE	CRESCND	CRESERL
	CRESIMD	CSETDSM	CTRMLTR	DE	DFRET	DISCEOC	DISCNCT	DMKBLDVM	DMKCFMAT	DMKCFMBK
	DMKCNTED	DMKCPVAE	DMKCVTBH	DMKCVTDT	DMKDSPCH	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKIOERN
	DMKIOSQR	DMKLOKSW	DMKNLDR	DMKNLEMP	DMKQCNWT	DMKQCOCL	DMKQCOET	DMKQCPTO	DMKRIORN	DMKSCNAU
	DMKSCNRU	DMKSTKCP	DMKVSTPT	EDIT	ERRMSG	F1	F16	F255	F256	F4
	F60	F8	IL	INHIBIT	INTREQ	IOBCAW	IOBCC1	IOBCC3	IOBCP	IOBCSW
	IOBFLAG	IOBIOER	IOBIRA	IOBLINK	IOBLOK	IOBMISC	IOBMISC2	IOBRADD	IOBRCAW	IOBRCNT
	IOBRSTRT	IOBSIZE	IOBSPEC	IOBSTAT	IOBUNSL	IOBUSER	IOERBLOK	IOERDATA	IOEREXT	IOERSIZE
	IPLREQ	LOCK	LOGDROP	LOGHOLD	MP	NICATOF	NICATTN	NICBLOK	NICCIBM	NICDED
	NICDISA	NICDISB	NICENAB	NICEPMD	NICERLK	NICFLAG	NICLINE	NICLTRC	NICMTA	NICNAME
	NICNTRL	NICPSUP	NICQPNT	NICRCNT	NICSESN	NICSIZE	NICSTAT	NICTELE	NICTERM	NICTYPE
	NICUSER	NOAUTO	NORET	OPERATOR	PCI	PREFIXA	PRGC	PRIORITY	PRTC	PSA
	RDBUFLN	RDBUFNO	RDEVAUTO	RDEVBLOK	RDEVBUZY	RDEVCKPT	RDEVCON	RDEVDED	RDEVDISA	RDEVFLAG
	RDEVLCEP	RDEVLNCP	RDEVMAX	RDEVNCP	RDEVNICL	RDEVNRDY	RDEVPEND	RDEVRCVY	RDEVRSVD	RDEVSCHD
	RDEVSLOW	RDEVSTAT	RDEVSTA4	RDEVTBTU	RDEVTYPC	RDEVWAIT	READNRM	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER0	SAVER1	SAVER2	SILI	SVMUNLOK
	SYSTEM	TEMPSAVE	TIMEDISP	TRACCURR	TRACEND	TRACFLG2	TRACPROC	TRACSTRT	TRACSVCR	TRAC11
	TRCNCP	TYP3705	UC	UCASE	UE	VMBLOK	WRITBRK	WRITEOT	WRITNRM	XRIGHT16
	ZEROES									
DMKRPA	ACORETBL	AFREEP	ALOKRM	ALOKSP	AP	APSTAT1	APSTAT2	APTRAN	APUOPER	AVMREAL
	BRING	CORCFLCK	CORFLAG	CORFPNT	CORIOLCK	CORLCNT	CORPGPNT	CORSWPNT	CORTABLE	CPEXADD
	CPEXBLOK	CPEXFPNT	CPEXMISC	CPEXR0	CPEXSIZE	CPPTLBR	DEFER	DMKFREEP	DMKPAGIO	DMKPGSPR
	DMKPGUPR	DMKPGUSP	DMKPTRAN	DMKPTRUL	DMKPTRWQ	DMKPTTFT	DMKSCHDL	DMKSWAPR	DMKVATSI	DMKVMI
	FFS	F1	F4	IOERETN	LOCK	LOCKSAV	LPUADDR	MICBLOK	MICVTMR	MP
	NORLSE	PAGCORE	PAGINVAL	PAGREF	PREFIXB	PSA	R0	R1	R11	R12
	R13	R14	R15	R2	R3	R5	R7	R9	SAVEAREA	SAVEREGS
	SAVER1	SAVER2	SAVER3	SAVEWRK1	SAVEWRK2	SAVEWRK3	SPECIALV	SWPCHG1	SWPCHG2	SWPCYL
	SWPFLAG	SWPRECMP	SWPSHR	SWPTRANS	SYSTEM	VMBLOK	XPAGNUM	ZEROES		
DMKRPD	AP	APTRAN	APTRLK	BLANKS	BRING	C1	DEFER	DMKPRGSM	DMKPTRAN	DMKPTRUL
	DMKUDRFU	F0	LOCK	MP	PSA	R0	R1	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	SAVEAREA	SAVEREGS	SAVEWRK2
	VMBLOK									
DMKRPI	ACICODE	ACIDEFR	ACIPARMS	AP	DMKIUACP	LOCK	MP	RCNOSEND	R0	R1
	R11	R12	R13	R15	R5	SAVEAREA	SAVEREGS	SEVER		
DMKRPW	ACICODE	ACIDEFR	ACIPARMS	AP	LOCK	MP	R0	R1	R11	R12
	R13	SAVEAREA	SAVEREGS							

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
DMKRSE	AFREE	AFREEP	AFRET	ALOKSP	AP	APSTAT1	APUOPER	ATTN	BUSOUT	CC
	CCC	CDC	CE	CHC	CLASURI	CLASURO	CMDREJ	CUE	DATACHK	DE
	DMKCVTBD	DMKCVTBH	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKIOEST	DMKLOKIO	DMKMSWR	DMKRSFPB
	DMKRSFPR	DMKRSFSD	DMKRSPPR	DMKRSP83	EQCHK	F1	F15	F20	F3	F4
	F5	F7	F8	IFCC	INTREQ	IOBCAW	IOBCC1	IOBCC3	IOBCSW	IOBERP
	IOBFATAL	IOBFLAG	IOBIOER	IOBLOK	IOBMISC2	IOBRADD	IOBRCAW	IOBRCNT	IOBRSTRT	IOBSIZE
	IOBSTAT	IOERACT	IOERBLOK	IOERCCRA	IOERCCRL	IOERCEMD	IOERCSW	IOERDATA	IOERDEPD	IOERDERD
	IOERECSW	IOERERP	IOERETRY	IOEREXT	IOERFLG1	IOERFLG2	IOERFLG3	IOERIGN	IOERIND3	IOERINFO
	IOERNUM	IOERPEND	IOERPNT	IOERREAD	IOERSIZE	IOERXERP	LOCK	LPUADDR	MP	PCI
	PRGC	PRTC	PSA	RDEVAIOB	RDEVBACK	RDEVBLOK	RDEVBUZY	RDEVDELP	RDEVFLAG	RDEVIOER
	RDEVNRDY	RDEVRSTR	RDEVSPL	RDEVSTAT	RDEVSTA4	RDEVTERM	RDEVTYPC	RDEVTYPE	R0	R1
	R10	R11	R12	R13	R14	R15	R2	R3	R4	R5
	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK5
	SFBFILID	SFBFLAG	SFBLOK	SFBPNT	SFBRECER	SFBSHOLD	SM	TYPPUN	TYP1403	TYP1443
	TYP2501	TYP2520P	TYP2540P	TYP2540R	TYP3211	TYP3505	TYP3800	TYP38003	TYP38008	TYP4245
	TYP4248	UC	VMBLOK	ZEROES						
DMKRSF	ADSPCH	AFREE	AFRET	AP	CC	CCC	CDC	DATACHK	DMKDSPCH	DMKFREE
	DMKFRET	DMKIOESD	DMKIOSQR	FFS	FTREXTSN	F8	IFCC	IOBCAW	IOBCC3	IOBERP
	IOBFATAL	IOBFLAG	IOBIOER	IOBIRA	IOBLOK	IOBMISC	IOBRCAW	IOBRSTRT	IOBSTAT	IOBUSER
	IOERBLOK	IOERCCRA	IOERCCRL	IOERCPID	IOERCSW	IOERDATA	IOEREXT	IOERFLG1	IOERLEN	IOERPNT
	IOERSIZE	IOERXERP	LOCK	MP	NOTUSED	PSA	RDEVBLOK	RDEVFTR	RDEVIOER	RDEVPROC
	RDEVTYPE	R0	R1	R10	R11	R12	R13	R14	R15	R3
	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER6	SAVER7
	SAVEWRK1	SAVEWRK2	SAVEWRK4	SAVEWRK6	SILI	SKIP	TYP3203	TYP3262	TYP3289E	TYP4245
	VMBLOK	XOBRCCW1	XOBRCCW2	XOBRCCW3	XOBRCCW4	XOBREXT	XOBRFCB	XOBRFLAG	XOBRMIS1	XOBRMIS2
	XOBRRT1	XOBRRT2	XOBRRT3	XOBRRT4	XOBRRT5	XOBRRT6	XOBRSIZE	XOBRSTAT	XOBRT1	XOBRT2
	XOBRT3	XOBR010	XOBR150	XOBR512	ZEROES					
DMKRSP	ADSPCH	AFREE	AFREEP	AFRET	ALOKSP	AP	APSTAT1	APTRAN	APTRLK	APUOPER
	ASYSVM	ATTN	BLANK	BRING	CC	CCC	CD	CDC	CE	CHGSFB
	CKPTLIST	CLASFBA	CLASURI	CLASURO	CPEXADD	CPEXBLOK	CPEXSIZE	CPUID	C1	DE
	DEFER	DMKAPZNO	DMKCKSPL	DMKCSBLF	DMKCSBSP	DMKCSOSD	DMKCVTBH	DMKCVTDT	DMKDMPAS	DMKDMPC2
	DMKDMPRC	DMKDSPCH	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKIOCVT	DMKIOECL	DMKIOECQ	DMKIOEES
	DMKIOEID	DMKIOEIP	DMKIOEIQ	DMKIOEMQ	DMKIOEMX	DMKIOFEP	DMKIOJBL	DMKIOSQR	DMKLOKIO	DMKOPRWT
	DMKPGUVG	DMKPGUVR	DMKPTRAN	DMKPTSAD	DMKRIOCN	DMKRPAGT	DMKRSERR	DMKRSQDC	DMKRSQFR	DMKRSTIN
	DMKSCNDC	DMKSCNRD	DMKSCNRU	DMKSEPSP	DMKSEPTL	DMKSNTQN	DMKSPKDL	DMKSTKCP	DMKSTKOP	DMKSYSFL
	DMKSYSHE	DMKSYSHL	DMKSYSHT	DMKSYSOC	DMKSYSOW	DMKSYSRM	DMKSYSSP	DMKSYSTP	DMKSYSVM	DMKSYSWI
	DMKSYSWM	DMKTCSCO	DMKTCSET	DMKTCSTR	DMKTMRPT	DMKURSTA	DMKVRR	FFS	F0	F1
	F2	F255	F3	F4	F4095	F4096	F8	IFCC	IOBCAW	IOBCC1
	IOBCP	IOBCSW	IOBERP	IOBFATAL	IOBFLAG	IOBIOER	IOBIRA	IOBLINK	IOBLOK	IOBMISC
	IOBMISC2	IOBRADD	IOBRCAW	IOBRCNT	IOBRSTRT	IOBSIZE	IOBSPEC	IOBSTAT	IOBUNSL	IOERBLOK
	IOERCCRA	IOERCCRL	IOERCSW	IOERDATA	IOERDEPD	IOERDERD	IOERERP	IOERETN	IOEREXT	IOERFLG1
	IOERSIZE	IOKEEP	LOCK	LPUADDR	MP	NPRCNT	NPRNAME	NPRPNT	NPRTBL	PSA
	RDCBLOK	RDCPAGAP	RDEVALGN	RDEVBACK	RDEVBLOK	RDEVBUZY	RDEVCFCB	RDEVCLAS	RDEVDED	RDEVDELP
	RDEVDFCB	RDEVDISA	RDEVDRAN	RDEVEXTN	RDEVFLAG	RDEVIMAG	RDEVIOER	RDEVISPL	RDEVLDBG	RDEVLDMD
	RDEVLOAD	RDEVMAXP	RDEVNRDY	RDEVOVLY	RDEVPRFG	RDEVPTHS	RDEVPURG	RDEVRDC	RDEVRSTR	RDEVSEP
	RDEVSPAC	RDEVSPL	RDEVSTAT	RDEVSTA2	RDEVSTA4	RDEVTERM	RDEVTYPC	RDEVTYPE	RECBLOK	RECCYL
	RECMAP	RECPNT	RECSIZE	RECUSED	RSPBF1DC	RSPBF1IO	RSPBF1VL	RSPBF2DC	RSPBF2IO	RSPBF2VL
	RSPCLPRT	RSPDEFER	RSPDPAGE	RSPDPAG2	RSPFLAG1	RSPFLAG2	RSPIMIDL	RSPLCTL	RSPMISC	RSPRPAGE
	RSPRPAG2	RSPRSTRT	RSPSEP	RSPSFBLK	RSPSFLOK	RSPSIZE	RSPVPAGE	RSPVPAG2	RSPXAUTO	RSPXBLOK
	RSPXDEST	RSPXDST1	RSPXFCB	RSPXFILE	RSPXFLAG	RSPXFMNT	RSPXFORM	RSPXFPND	RSPXNOPL	RSPXPMNT
	RSPXRECT	RSPXSEQ	RSPXSETU	RSPXSFIL	RSPXVTRC	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	R9	SAVEAREA	SAVEREGS	SFBCLAS	SFBCONV	SFBCOPY	SFBDEST	SFBFCB	SFBFCBNL	SFBFCBXL
	SFBFLAG	SFBFLAG2	SFBFLAG3	SFBFLAG4	SFBFLASH	SFBFNAME	SFBINUSE	SFBLAST	SFBLDBEG	SFBLDMID
	SFBLOK	SFBOFORM	SFBPNT	SFBRECER	SFBRECNO	SFBRECOK	SFBRECS	SFBREQUE	SFBRSTRT	SFBSHOLD
	SFBSIZE	SFBSTART	SFBTICER	SFBTUSE	SFBTYPE	SFBUHOLD	SFBUSER	SILI	SKIP	SPLINK
	SPNXTPAG	SPPREPAG	SPRECNUM	SPRMISC	SPSIZE	SYSTEM	TEMPR1	TYPPRT	TYPPUN	TYP3203
	TYP3211	TYP3262	TYP3800	TYP38003	TYP38008	TYP4245	TYP4248	TYP5ACCW	UC	UE
	URSBACK	URSDEV	URSFILE	URSFLUSH	URSHELD	URSOPER	URSREP	VMBLOK	XPAGNUM	ZEROES
DMKRSQ	AFREE	AP	BRING	CD	CLASFBA	DMKFREE	DMKPGUVG	DMKPGUVR	DMKRPAGT	DMKSCNDC
	F255	IOBLOK	LOCK	MP	PSA	RDCBLOK	RDCPAGAP	RDEVBLOK	RDEVRDC	RDEVTYPC
	RECBLOK	RECCYL	RECMAP	RECPNT	RECSIZE	RECUSED	RSPLCTL	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER2	SAVEWRK2	SAVEWRK3	SAVEWRK4	SFBFLAG
	SFBLOK	SFBPNT	SFBRECER	SFBRECOK	SFBRECS	SILI	SKIP	SPLINK	SPNXTPAG	SPPREPAG
	SPRMISC	SPSIZE	SYSTEM	TEMPR1	VMBLOK					
DMKRST	ADSPCH	AFREE	AFREEP	AFRET	APAGCP	APTRAN	APTRLK	ASYSVM	BLANK	BRING
	BUFCNT	BUFFER	BUFNXT	BUFSIZE	CC	CE	CPEXADD	CPEXBLOK	CPEXSIZE	C1
	DEFER	DMKCKTSD	DMKCVTBH	DMKDSPCH	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKIOSQR	DMKLOCRD
	DMKLOCRQ	DMKPGTSG	DMKPGUSD	DMKPGUVG	DMKPGUVR	DMKPTRAN	DMKPTRLK	DMKPTRUL	DMKRPAGT	DMKRPAPT
	DMKSCNFD	DMKSPKDL	DMKSPLCR	DMKSPLOR	DMKSTKCP	DMKUDRFU	DMKURSTA	DMKVSGAI	DMKVSGRI	F0
	F24	F6	F8	IL	IOBCAW	IOBCC1	IOBCP	IOBCSW	IOBFATAL	IOBFLAG
	IOBLINK	IOBLOK	IOBMISC	IOBRADD	IOBRCNT	IOBRSTRT	IOBSIZE	IOBSTAT	IOERETN	LOCK
	PSA	RDEVBLOK	RDEVDED	RDEVDRAN	RDEVFLAG	RDEVNRDY	RDEVSPL	RDEVSTAT	RDEVTYPE	RSPDPAGE
	RSPERR	RSPFLAG2	RSPLCTL	RSPRPAGE	RSPSFBLK	RSPSIZE	RSPSWAP	RSPVPAGE	RSPVPG2	R0
	R1	R10	R11	R12	R13	R14	R15	R2	R3	R4
	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVEWRK2	SFBCLAS	SFBFILID
	SFBFNAME	SFBFTYPE	SFBLAST	SFBLOK	SFBORIG	SFBPNT	SFBRECNO	SFBSIZE	SFBSTART	SFBSYSID
	SFBUSER	SILI	SKIP	SPLINK	SPNXTPAG	SPPREPAG	SPRECNUM	SPRMISC	SPSIZE	SYSTEM
	TYP2540R	UE	URSDEV	URSFILE	URSOPER	URSREAD	VMBLOK			
DMKSAD	ADMKCPE	APSTAT1	APUOPER	ASVCLIST	ASYSOP	BLANK	BUSY	CAW	CC	CCC
	CC0	CC1	CC2	CC3	CD	CDC	CE	CHC	CLASDASD	CLASTAPE
	CLASURO	CLOCKCMP	COLON	CONTINUE	CPABEND	CPID	CPSPMODE	CPSTAT2	CPUID	CPUTIMR
	CPUVERSN	CRLOG	CSW	CUE	CURNTPSW	C0	C1	C14	C2	DATE
	DE	DMPABEND	DMPCKCOM	DMPCPUID	DMPCPUTM	DMPCRS	DMPFPRS	DMPGPRS	DMPINREC	DMPIPCS
	DMPLCORE	DMPPGMAP	DMPPGMP2	DMPPRFRG	DMPSYSRM	DMPSYSRV	DMPTODCK	DMPVMTYP	EIGHT	EXNPSW
	EXOPSW	EXTMODE	FPRLOG	GRLOG	HEADER	IFCC	INTPR	INTTIO	IOMASK	IONPSW
	IOOPSW	IPLPSW	LINE	MCHEK	MCNPSW	MCOPSW	MICBLOK	MICPMPSA	MPUOPER	NOTUSED
	PMAAVAIL	PMAON	PMASTAT	PREFIXA	PREFIXB	PREFIXVL	PRGC	PRNPSW	PROPSW	PRTC
	PSA	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SENSE	SFBDATE	SFBDIST
	SFBLOK	SFBORIG	SFBRECNO	SFBTIME	SFBUSER	SILI	SM	SPMPFX	SPSIZE	SVCNPSW
	SVCOPSW	TODATE	TRACSTRT	TRACSVST	TRANMODE	TYPPRT	UC	UE	WAIT	
DMKSAV	ALARM	AP	AVMREAL	BUSY	CAW	CC	CC0	CE	CKPLOAD	CLASDASD
	CLASFBA	CPID	CSW	CUE	C2	DE	DMKCKD	DMKCKF	DMKCKH	DMKCKM
	DMKCKN	DMKCKNND	DMKCKP	DMKCKPLD	DMKCKPLE	DMKCKPLF	DMKCKPLH	DMKCKPLM	DMKCKPLN	DMKCKPLW
	DMKCKPNC	DMKCKPND	DMKCKPRS	DMKCKPR2	DMKCKPST	DMKCKPT	DMKCKW	DMKCPICD	DMKCPINT	DMKCVTBD
	DMKCVTBH	DMKERP	DMKOPRWT	DMKSYSAL	DMKSYSNL	DMKSYSNU	DMKSYSRS	DMKSYSTP	DMKSYSTZ	DMKSYSVL
	D2	D4	EXNPSW	F1	F2	F3	F4	F4096	INTREQ	IONPSW
	IOOPSW	LOCK	L3	L4	L5	L7	L8	MCNPSW	MP	NDISK
	PRNPSW	PSA	PSTARTSV	QDISK	REGSAV	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SILI	SKIP	SM	TEMPR2	TEMPR4	TEMPSAVE	TYP2305	TYP2314	TYP3330

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	TYP3340	TYP3350	TYP3375	TYP3380	UC	UE	VMBLOK			
DMKSBL	AP	CL	F1	LOCK	MP	PSA	R0	R1	R11	R12
	R13	R14	R15	R3	R4	R6	R7	R8	SAVEAREA	SAVEREGS
DMKSCH	ADSPCH	AEXTSP	AFREE	ALOKSP	ALOKSY	AP	APSTAT1	APUOPER	ARPCHK1	ARPCHK2
	ASYSVM	ATMRSN	AVMREAL	BALRSAVE	BALRSAV2	BALR11	CODE	CPEXADD	CPEXBLOK	CPEXMISC
	CPEXPROC	CPEXR0	CPEXSIZE	CPRUN	CPSTATUS	CPUID	CPWAIT	C1	DMKCVTAB	DMKDSPCH
	DMKDSPNP	DMKDSPN2	DMKFREE	DMKFREXX	DMKLOKRL	DMKPTRRC	DMKPTRSC	DMKPTSRS	DMKSTKCP	DMKSWAPI
	DMKSYSSP	DMKTRQRT	DMKTRQST	FFS	F1	F16	F2	F24	F256	F6
	F8	IPUADDRX	LOCK	LOCKSAV	LPUADDR	LPUADDRX	MICBLOK	MICEVMA2	MICSPT	MNCLSCH
	MNCOAEL	MNCOAQ	MNCODQ	MP	NOALCARY	NOSLBORO	PREFIXA	PREFIXB	PROBSTRT	PROBTIME
	PSA	PSAEXT	PSAEXTX	PXA	Q1DROP	RUNUSER	R0	R1	R10	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	SIGWAKE	SIGXC	SLNONPOS	STARTIME	TEMPR0	TEMPR10	TEMPR11	TEMPR12
	TEMPR13	TEMPR14	TEMPR15	TEMPR4	TEMPR7	TEMPR9	TEMPSAVE	TIMEDISP	TIMER	TRACCURR
	TRACEND	TRACFLG1	TRACPROC	TRACSTRT	TRACSVCR	TRAC08	TRAC09	TRCDROP	TRCSCH	TRLANCHR
	TRLCT	TRLLABEL	TRLLOCK	TRQBLOK	TRQBQUE	TRQBTOD	TRQBVAL	TSEND	VMBLOK	XCAPR
	XCDISP	XCPEND	XCRES	XCWAK	XRIGHT16	XTNDLOCK	X40FFS	ZEROES		
DMKSCN	AP	APSTAT1	ARIOCH	ARIOCT	ARIOCU	ARIODC	ARIODV	ASYSVM	BALRSAVE	BALR0
	BALR1	BALR2	BALR8	BLANKS	BUFFER	BUFNXT	CLASDASD	CLASFBA	CLASSPEC	CLASTERM
	CLASURI	CLASURO	CPSTAT5	C1	DMKSYSOC	DMKSYSOW	FFS	FTR3088	F0	F255
	F4095	F4096	F5	F7	LOCK	MP	MPGEND	MPUOPER	NICBLOK	NICRSPL
	NICSIZE	NICTYPE	OWNDLIST	OWNDRDEV	OWNDVSER	PROCIO	PROCIPL	PSA	RCHADD	RCHBLOK
	RCHCUTBL	RCUADD	RCUBLOK	RCUCHA	RCUCHCNT	RCUCHD	RCUDISA	RCUDVTBL	RCUPRIME	RCUSTAT
	RCUSUB	RCUTYPE	RDEVADD	RDEVAOF	RDEVBLOK	RDEVBOF	RDEVCUA	RDEVCUB	RDEVDED	RDEVDISA
	RDEVFLAG	RDEVFTR	RDEVMOUT	RDEVNICL	RDEVPS	RDEVPTHS	RDEVSER	RDEVSIZE	RDEVSTAT	RDEVSTA3
	RDEVTYPC	RDIDX	R0	R1	R11	R14	R15	R2	R3	R4
	R5	R6	R7	R8	R9	TYPCTCA	TYPIBM1	TYPPRT	TYPPUN	TYPRDR
	TYPSDLC	TYPTELE2	TYP2700	TYP3210	TYP3705	TYP3800	TYP38003	TYP38008	TYP3851	VCONCTL
	VCONNICB	VCONRDEV	VCONREMD	VCONREMF	VMBLOK					
DMKSCO	CLASDASD	CLASFBA	FTR2311B	FTR2311T	LOCK	PSA	RCHBLOK	RCUBLOK	RCUCHA	RCUCHC
	RCUPRIME	RCUSUB	RCUTYPE	RDEVBLOK	RDEVCUA	RDEVCUB	RDEVLNKS	R0	R1	R10
	R11	R12	R13	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SAVER0	SAVER11	SAVER3	SAVER6	SAVER8	TYP2311	VMBLOK
	ZEROES									
DMKSEG	ACORETBL	APTRAN	ASYSVM	BALR0	BRING	CC2	CLASFBA	CORCFLCK	CORFLAG	CORIOLCK
	CORLCNT	CORPGPNT	CORSWPNT	CORTABLE	CPSTAT4	CPXSTOR	C1	DEFER	DMKBLDRT	DMKCKP
	DMKMCHST	DMKPGTPG	DMKPGUBN	DMKPMAI2	DMKPTRAN	DMKPTTAL	DMKRPAPT	DMKSAV	DMKSCNRU	DMKSCNVS
	DMKSYM	DMKSYSNU	DMKSYSRM	DMKSYSRV	DMKSYSSP	DMKSYSVL	DMKVMI	F0	F1	F16
	F2	F4	F4096	IOERETN	LOCK	NEWPAGES	NEWSEGS	PAGCORE	PMAAVAIL	PMASTAT
	PSA	RDEVBLOK	RDEVCODE	RDEVTYPC	RDEVTYPE	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SEGPAGE	SWPCHG1	SWPCYL	SWPFLAG	SYSIPLDV	SYSTEM	TYP2314
	TYP3330	TYP3350	TYP3375	TYP3380	VMBLOK	VMSIZE				
DMKSEL	ACORETBL	ADSPCH	AEXTSP	AFREE	AFREEP	AFRET	ALOCBLOK	ALOCCSR	ALOCPSIO	ALOCPWRT
	ALOCRWRT	ALOKRM	ALOKSP	ALOKSY	ALOKVM	APSTAT1	APSTAT2	APUOPER	ASYSVM	CHG
	CONTROL	CORBPNT	CORCP	CORFLAG	CORFLUSH	CORFPNT	CORIOLCK	CORPGPNT	CORSHARE	CORSWAP
	CORSWPNT	CORTABLE	CORVM	CPEXADD	CPEXBLOK	CPEXFPNT	CPEXMISC	CPEXR0	CPEXR11	CPEXR13
	CPEXR2	CPEXSIZE	CPPTLBR	CPSTAT4	CPXSTOR	C1	DMKCVTAB	DMKDSPCH	DMKDSPNP	DMKDSPN2
	DMKDSPRU	DMKFREE	DMKFREEP	DMKFRET	DMKLOKDF	DMKLOKSW	DMKPAGIO	DMKPGTPB	DMKPGTPG	DMKPGUAL

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	DMKPGUPR	DMKPTRAQ	DMKPTRCS	DMKPTREE	DMKPTREF	DMKPTRES	DMKPTRFA	DMKPTRFF	DMKPTRFN	DMKPTRFQ
	DMKPTRF2	DMKPTRNS	DMKPTRN2	DMKPTRPO	DMKPTRPS	DMKPTRQC	DMKPTRQ2	DMKPTRRC	DMKPTRRF	DMKPTRSC
	DMKPTRSS	DMKPTRSW	DMKPTRS2	DMKPTRU1	DMKPTRWQ	DMKPTSAD	DMKPTSAE	DMKPTSMW	DMKPTTFT	DMKPTTPM
	DMKSTKCP	DMKSWAPO	DMKSYSCS	DMKSYSSZ	DMKVATSI	FF	FFS	F0	F1	F16
	F2	F4096	F8	IPUADDRX	KEYCHG	KEYREF	LASTUSER	LOCK	LOCKSAV	LPUADDR
	MICBLOK	MICVTMR	NOSIGP	OPPRSTRT	PAGCORE	PAGINVAL	PAGRBITS	PAGREF	PAGSHR	PAGTABLE
	PAGTSWP	PGWRITE	PREFIXA	PREFIXB	PROCIPL	PSA	PSAEXT	PSTPOUPP	PXA	RUNUSER
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER11	SAVER13
	SAVER2	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK8	SAVEWRK9	SEGFLAG	SEGINV	SEGPAGE	SHRFLAG
	SHRNOPRT	SHRSGPRT	SHRTABLE	SIGEMS	SIGQUI	SIGRES	SIGXC	SSBBPNT	SSBCPG	SSBEFLG
	SSBEINVL	SSBENTRL	SSBENTRY	SSBFLAG	SSBFLUSH	SSBFPNT	SSBHEADL	SSBLOK	SSBNPGS	SSBTRANS
	SSBVMSCB	SSBVPAGE	SWPALLOC	SWPAPP	SWPCHG1	SWPCHG2	SWPCODE	SWPCYL	SWPFLAG	SWPFLAG2
	SWPKEY1	SWPKEY2	SWPPAG	SWPPSTOR	SWPRECMP	SWPREF1	SWPREF2	SWPSEGNO	SWPSHR	SWPTABLE
	SWPTRANS	SWPVM	SWPVPAGE	TEMPR14	TIMEDISP	TIMER	UNCHG	UNREF	VMBLOK	XKEYMODE
	XTNDLOCK									
DMKSEP	ADSPCH	AFRET	APTRAN	APTRLK	ASYSVM	BLANK	BLANKS	BOXBLOK	BOXLOGO	BRING
	CC	CC3	CPUMODEL	CPUSER	C1	DEFER	DMKBOXPR	DMKCPEID	DMKCVTBD	DMKCVTBH
	DMKCVTDT	DMKDSPCH	DMKERMSG	DMKFRET	DMKIOSQR	DMKPGUVG	DMKPGUVR	DMKPTRAN	DMKPTRUL	DMKRPAGT
	DMKSBLTR	DMKSCNRD	DMKSEQDA	DMKSEQLA	DMKSEQSA	DMKSYSFL	DMKSYSSF	DMKTBLSF	DMKTCSSP	FFS
	FORMBLOK	FORMFLAG	FORMNARR	FORMNTRY	FORMOPER	FORMSEND	FORMUSER	F0	F4	F9
	IOBCAW	IOBCSW	IOBFATAL	IOBFLAG	IOBIRA	IOBLINK	IOBLOK	IOBMISC	IOBMISC2	IOBRSTRT
	IOBSIZE	IOBSTAT	IOERETN	IOKEEP	LOCK	PSA	RDEVBLOK	RDEVEXTN	RDEVFLAG	RDEVLOAD
	RDEVPRFG	RDEVSEP	RDEVSEPF	RDEVTYPE	RSPXBLOK	RSPXSEQ	R0	R1	R10	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	SAVEAREA	SAVEREGS	SAVER10	SAVER8	SAVEWRK1	SAVEWRK2	SAVEWRK4	SAVEWRK5
	SAVEWRK7	SAVEWRK8	SAVEWRK9	SFBCLAS	SFBDEST	SFBDIST	SFBFILID	SFBFLAG2	SFBLOK	SFBOFORM
	SFBORIG	SFBRECNO	SFBRSTRT	SFBSTART	SFBUSER	SFBPBLOK	SFBPCLASS	SFPEND	SFPLEN	SFPTITLE
	SILI	SKIP	SPSIZE	SYSTEM	TYPPUN	TYP3203	TYP3211	TYP3262	TYP3800	TYP38003
	TYP38008	TYP4245	TYP4248	UE	VMBLOK	ZEROES				
DMKSEV	CCC	CCHCHNL	CCHCMDV	CCHCNTB	CCHCPU	CCHDAV	CCHDI	CCHINTFC	CCHLOG70	CCHREC
	CCHSTG	CCHUSV	COMPFES	COMPSEL	COMPSYS	CSW	FFS	F7	F8	HIOCCH
	IFCC	IGBLAME	IGPRGFLG	IGTERMSQ	IGVALIDB	INTERCCH	IOERBLOK	PSA	RTCODE1	RTCODE2
	RTCODE3	RTCODE4	RTCODE5	RTCODE7	R0	R1	R12	R13	R14	R15
	R2	R3	R4	R9	SAVEAREA	SAVEWRK1	SAVEWRK9	TERMSYS	TIOCCH	XRIGHT16
DMKSFB	APTRAN	APTRLK	ARSPPR	ARSPPU	BLANKS	BRING	C1	DEFER	DMKLOCRD	DMKLOCRQ
	DMKPSAFP	DMKPSASP	DMKPTRAN	DMKPTRUL	DMKVSFID	DMKVSFNS	DMKVSFNU	F0	LOCK	PSA
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER0	SAVER2
	SAVEWRK1	SFBFILID	SFBFLAG	SFBINUSE	SFBLOK	SFBPNT	SFBSIZEB	SFBSYSID	SFBUSER	VMBLOK
	XPAGNUM	ZEROES								
DMKSIX	CCHCHNL	CCHCMDV	CCHCNTB	CCHCPU	CCHDAV	CCHDI	CCHINTFC	CCHLOG60	CCHREC	CCHSTG
	CCHUSV	COMPFES	COMPSEL	CSW	FFS	F1	F7	F8	HIOCCH	IFCC
	IGBLAME	IGPRGFLG	IGTERMSQ	IGVALIDB	IOERBLOK	PSA	RTCODE1	RTCODE2	RTCODE3	RTCODE4
	RTCODE5	R0	R1	R12	R13	R14	R15	R2	R3	R4
	R9	SAVEAREA	SAVEWRK1	SAVEWRK9	TERMSYS	TIOCCH	XRIGHT16			
DMKSNC	APTRAN	APTRLK	ASYSVM	BRING	CCPADDR	CCPARM	CCPNAME	CCPPSIZE	CCPSIZE	CLASFBA
	C1	DEFER	DMKERMSG	DMKPTRAN	DMKPTRUL	DMKRPAPT	DMKSCNVS	DMKSNTRN	F0	F1
	F256	F4096	IOERETN	LOCK	NCPNAME	NCPPAGCT	NCPPNT	NCPSTART	NCPTBL	NCPVOL
	PSA	RDEVBLOK	RDEVCODE	RDEVFLAG	RDEVOWN	RDEVTYPC	RDEVTYPE	R0	R1	R10

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)								
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER0	SAVER2	SAVER6	SAVEWRK1	SAVEWRK2
	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK8	SAVEWRK9	SYSTEM	TYP2314	TYP3330	TYP3350
	TYP3375	TYP3380	VMBLOK	XPAGNUM	ZEROES					
DMKSND	ADSPCH	AFREE	AFREEP	AFRET	AP	APSTAT1	APUOPER	BLANK	BUFCNT	BUFFER
	BUFIN	BUFNXT	BUFSIZE	CPCMD	CPEXADD	CPEXBLOK	CPEXREGS	CPEXR1	CPEXR11	CPEXR12
	CPEXR2	CPEXR8	CPEXSIZE	DMKAPYSD	DMKCFMAT	DMKCFMBK	DMKCFMSD	DMKERMSG	DMKFREE	DMKFREEP
	DMKFRET	DMKLOKSW	DMKSCHRT	DMKSCNAU	DMKSCNFD	DMKSTKCP	F2	LOCK	MP	PSA
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER11	SAVER9
	SAVEWRK7	SAVEWRK8	TIMEDISP	VCONCTL	VCONRBUF	VCONRCNT	VCONRDSZ	VMBLOK	VMGENIO	
DMKSPC	BUFCNT	BUFFER	BUFNXT	DMKCFYSP	DMKCQQID	R0	R1	R11	R12	R13
	R15	R9	SAVEAREA	SAVEREGS						
DMKSPK	ADSPCH	AFREE	AFREEP	AFRET	AP	APTRAN	APTRLK	ARSPRD	ASYSVM	BLANKS
	BRING	CC	CLASFBA	CPEXADD	CPEXBLOK	CPEXREGS	CPEXSIZE	C1	DEFER	DELSFB
	DMKCKSPL	DMKCKTSD	DMKDRDDD	DMKDSPCH	DMKFREE	DMKFREEP	DMKFRET	DMKIOSQR	DMKLOCRD	DMKLOCRQ
	DMKPGUSD	DMKPGUSR	DMKPGUVG	DMKPGUVR	DMKPTRAN	DMKPTRLK	DMKPTRUL	DMKRPAGT	DMKRSPDL	DMKRSPHQ
	DMKSCNDC	DMKSTKCP	DMKSYSFL	DMKVSGRI	FFS	FTR70MB	F0	F10	F2	F3
	F4	F8	IOBCAW	IOBCP	IOBCYL	IOBFATAL	IOBFLAG	IOBIRA	IOBLOK	IOBMISC2
	IOBSTAT	IOBUSER	IOERETN	LOCK	MP	OWNDLIST	PSA	RDCBLKAP	RDCBLKPG	RDCBLOK
	RDCPAGXT	RDEVBLOK	RDEVFTR	RDEVMDL	RDEVRDC	RDEVTYPC	RDEVTYPE	READ	RECBLOK	RECPNT
	RECSIZE	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER11
	SFBDIST	SFBDUMP	SFBFILID	SFBFIRST	SFBFLAG	SFBFLAG2	SFBFLAG4	SFBFLASH	SFBLAST	SFBLOK
	SFBMISC1	SFBMON	SFBPNT	SFBPURGE	SFBRECER	SFBRECNO	SFBRECS	SFBSIZE	SFBSTART	SFBSYSID
	SFBTYPE	SFBUSER	SFBXAB	SFBXABER	SFBXABL	SILI	SPLINK	SPRMISC	SYSTEM	TIMEDISP
	TYPRDR	TYP2305	TYP2314	TYP3340	TYP3350	TYP3375	TYP3380	VMBLOK	VMSEG	
DMKSPL	ACCTBLOK	ACCTDIST	ACCTUSER	ADDSFB	AFREE	AFREEP	AFRET	ALTBLOK	ALTUSER	AP
	APSTAT1	APTRAN	APTRLK	APUOPER	AQCNWT	ARIOPR	ARIOPU	ARSPPR	ARSPPU	ARSPRD
	ASYSVM	BLANK	BLANKS	BRING	CHGSFB	CLASURI	CLEAR	C1	DE	DEFER
	DMKAPZNO	DMKCKSPL	DMKCKTSU	DMKCKTUU	DMKCSOSD	DMKCVTBD	DMKCVTDT	DMKFREE	DMKFREEP	DMKFRET
	DMKLOCRD	DMKLOCRQ	DMKLOKSW	DMKPGTSG	DMKPGUVG	DMKPTRAN	DMKQCNPL	DMKQCNWT	DMKRPAGT	DMKRPAPT
	DMKRSPHQ	DMKSCNAU	DMKSPKDL	DMKSTKIO	DMKSYSFL	DMKUDRFU	DMKUDRMD	DMKUDRRV	DMKVIOIN	DMKVSDAD
	DMKVSGAI	DMKVSGRI	FFS	FORMBLOK	FORMNTRY	FORMOPER	FORMSEND	FORMUSER	F0	F1
	F3	F4096	F8	IOBCSW	IOBIRA	IOBLINK	IOBLOK	IOBSIZE	IOBSPEC	IOBUNSL
	IOBUSER	IOBVADD	LOCK	MP	NOCOPY	NORET	NOTRESP	PCHCHN	PLFLAG1	PLFRET
	PLIMSG	PLIST	PLR2	PLSIZE	PRTCHN	PSA	RDEVBLOK	RDEVSPL	RDRCHN	RSPDPAGE
	RSPLCTL	RSPRPAGE	RSPSFBLK	RSPSIZE	RSPVPAGE	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SAVER11	SAVER7	SAVER8	SAVER9	SAVEWRK1	SAVEWRK2	SAVEWRK8
	SFBCLAS	SFBCOPY	SFBDATE	SFBDEST	SFBDIST	SFBFILID	SFBFLAG	SFBFLAG2	SFBFLASH	SFBFNAME
	SFBHOLD	SFBLAST	SFBLOK	SFBNOHLD	SFBOFORM	SFBORIG	SFBPNT	SFBRECSZ	SFBREQUE	SFBRSTRT
	SFBSHOLD	SFBSIZE	SFBSTART	SFBSTCPY	SFBSYSID	SFBTIME	SFBTYPE	SFBUFORM	SFBUHOLD	SFBUSER
	SHQBLOK	SHQPNT	SHQSHOLD	SHQUSER	SKIP	SPBTSTAC	SPCHAR	SPCHAR1	SPCHAR2	SPCHAR3
	SPCMCHR	SPCMOD	SPCOPYFG	SPFCB	SPFLAG1	SPFLSHC	SPLINK	SPPGLEN	SPRECMAX	SPSPLNKC
	SYSTEM	TIMEDISP	TYPPRT	TYPPUN	TYPRDR	TYP1052	TYP3210	TYP3211	TYP3800	TYP38003
	TYP4248	VMBLOK	VSPLCTL	VSPSFBLK	VSPSIZE	VSPVPAGE	ZEROES			
DMKSPM	ACORETBL	ADSPCH	AFREE	AFREEP	AFRET	AP	APSTAT1	APSTAT2	APUOPER	AQCNWT
	ASYSVM	AVMREAL	BLANKS	CORCP	CORFLAG	CORTABLE	CORVM	CPEXADD	CPEXBLOK	CPEXREGS
	CPEXR12	CPEXR14	CPEXSIZE	CPPTLBR	CPSPMODE	CPSTAT2	CPSTAT4	CUE	DMKDMPCP	DMKERMSG

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	DMKFREE	DMKFREEP	DMKFRET	DMKIOSCB	DMKPTRFR	DMKPTTFT	DMKQCNWT	DMKSCNFD	DMKSTKCP	DMKVATBC
	DMKVATMD	DMKVIOIN	F1	F16	F3	F4	F4096	IOBCSW	IOBCUBSY	IOBCUE
	IOBIRA	IOBLINK	IOBLOK	IOBSIZE	IOBSPEC	IOBSPEC3	IOBUNSL	LOCK	LPUADDR	MICBLOK
	MICEVMA2	MICEVMA3	MICPMAMP	MICSTBVR	MICSTSM2	MP	MPFEAT	NORET	PMAAVAIL	PMAMODE
	PMASTAT	PREFIXA	PREFIXB	PSA	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	SAVER11	SAVEWRK2	SPMPFX	SWPFLAG	SWPKEY1	SWPKEY2	TIMEDISP	TREXBRAN
	TREXCTL2	TREXINST	TREXT	VHD	VMBLOK	XKEYASST	XKEYMODE	ZEROES		
DMKSPR	AFREE	AP	APSTAT1	APUOPER	AQCNWT	ASYSVM	BLANK	DFRET	DMKCVTBD	DMKFREE
	DMKLOKSW	DMKQCNWT	DMKSCNAU	LOCK	MP	NORET	OPERATOR	PSA	R0	R1
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R9	SAVEAREA	SAVEREGS	SPTBLOK	SPTISSUR	SPTOSFID	SPTOUSER	SPTRADDR	SVMNOUPD	SVMUNLOK
	TIMEDISP	VMBLOK								
DMKSPS	ADDSFB	ADSPCH	AFREE	AFRET	AP	APSTAT1	APUOPER	AQCNWT	ARIODV	ARSPPR
	ARSPPU	ARSPTA	ASYSOP	ASYSVM	BLANK	BLANKS	BRING	CC	CLASFBA	CPSTAT5
	DMKCKSPL	DMKCKTSD	DMKCKTUU	DMKCVTAB	DMKCVTBD	DMKDSPCH	DMKERMSG	DMKFREE	DMKFRET	DMKHVCPC
	DMKIOSQR	DMKLOCRD	DMKLOCRQ	DMKLOKSW	DMKPGTDG	DMKPGTSG	DMKPGUSD	DMKPGUVR	DMKPTRUL	DMKQCNWT
	DMKRPAGT	DMKRPAPT	DMKRSPFL	DMKSCNAU	DMKSPKDL	DMKSPROT	DMKSYSDU	DMKSYSOW	DMKVSDAD	DMKVSDDL
	DMKVSETR	DMKVSGRI	F0	F1	IL	IOBCSW	IOBFATAL	IOBFLAG	IOBIOER	IOBLOK
	IOBMISC	IOBSIZE	IOBSTAT	IOERSIZE	LOCK	MP	NORET	OWNDLIST	OWNDRDEV	PCHCHN
	PRTCHN	PSA	RDEVBLOK	RDEVFLAG	RDEVSPT	RDEVSYS	RDEVTYPC	RDEVTYPE	RDEVUSER	RDIDX
	RDRCHN	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SFBBCONV	SFBCLAS
	SFBCONV	SFBDEST	SFBDUMP	SFBFILID	SFBFLAG	SFBFLAG4	SFBFNAME	SFBINUSE	SFBLAST	SFBLOK
	SFBMISC1	SFBOFORM	SFBPNT	SFBRECNO	SFBRECS	SFBSHOLD	SFBSIZE	SFBSTART	SFBSYSID	SFBTUSE
	SFBUHOLD	SFBUSER	SFBXAB	SILI	SPCPTRAP	SPLINK	SPNXTPAG	SPPREPAG	SPRMISC	SPTBLOK
	SPTBUFR1	SPTBUFR2	SPTBUFV1	SPTBUFV2	SPTCAN	SPTCLAS	SPTCLASS	SPTCODE	SPTDEST	SPTDESTS
	SPTDONE	SPTENTPT	SPTFILES	SPTFLAG	SPTFLAG1	SPTFLAG2	SPTFLAG3	SPTFORM	SPTFRMST	SPTINTAD
	SPTINTR	SPTIOBAD	SPTISSUR	SPTLAST	SPTLDPRT	SPTLDRDR	SPTLINK	SPTLKLST	SPTLOAD	SPTMSGAD
	SPTNOH	SPTOFLOW	SPTOSFID	SPTOUSER	SPTPRT	SPTPUR	SPTRADDR	SPTRDEV	SPTRDR	SPTREAD
	SPTREW	SPTRUN	SPTSAD	SPTSFB	SPTSHOLD	SPTSIZE	SPTSPID1	SPTSPID2	SPTSTOP	SPTTM
	SPTUHOLD	SPTUNLD	SPTUSER	SPTXAB	SPTXABLK	SVMNOUPD	SVMUNLOK	SYSTEM	TIMEDISP	TYP2305
	TYP2314	TYP3330	TYP3340	TYP3350	TYP3380	UC	UE	VMBLOK	ZEROES	
DMKSPT	AFREE	AFRET	APTRAN	APTRLK	ARSPPR	ARSPPU	ASYSVM	BLANKS	BRING	CC
	CC3	CLASTAPE	C1	DE	DEFER	DMKCVTDB	DMKCVTHB	DMKERMSG	DMKFREE	DMKFRET
	DMKIOSQR	DMKLOCRD	DMKLOCRQ	DMKPGUVG	DMKPTRAN	DMKPTRUL	DMKSCNFD	DMKSCNRU	DMKSPSIO	DMKSTKIO
	DMKVSFNS	DMKVSFNU	DUMP	FLAGS	F0	F1	F2	F3	F4	F5
	F6	F7	F8	IOBCAW	IOBCSW	IOBFLAG	IOBIRA	IOBLOK	IOBMISC	IOBUSER
	IOKEEP	LOAD	LOCK	OFF	PSA	RDEVBLOK	RDEVDED	RDEVDISA	RDEVFLAG	RDEVPEND
	RDEVSPT	RDEVSTAT	RDEVSYS	RDEVTYPC	RDEVTYPE	RDEVUSER	REWIND	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R6	R7
	R8	SAVEAREA	SAVEREGS	SAVER2	SAVEWRK1	SAVEWRK2	SAVEWRK4	SAVEWRK7	SAVEWRK8	SAVEWRK9
	SFBCLAS	SFBDEST	SFBFLAG	SFBFLAG4	SFBINUSE	SFBLOK	SFBOFORM	SFBSHOLD	SFBSYSID	SFBTUSE
	SFBUHOLD	SILI	SPTBLOK	SPTBUFR1	SPTBUFR2	SPTBUFV1	SPTBUFV2	SPTCAN	SPTCLAS	SPTCLASS
	SPTDEST	SPTDESTS	SPTENTPT	SPTFILES	SPTFLAG	SPTFLAG2	SPTFLAG3	SPTFORM	SPTFRMST	SPTINTR
	SPTIOBAD	SPTISSUR	SPTLOAD	SPTMODE	SPTMSGAD	SPTNOH	SPTOFLOW	SPTPRT	SPTPUR	SPTRADDR
	SPTRDEV	SPTRDR	SPTREW	SPTRUN	SPTSAD	SPTSHOLD	SPTSIZE	SPTSPID1	SPTSPID2	SPTSTOP
	SPTSYS	SPTUHOLD	SPTUSER	SYSTEM	TYP3422	TYP3430	TYP3480	TYP8809	VMBLOK	ZEROES
DMKSRM	AFREE	AFRET	ALOCPG	ALOCPP	ALOCPS	ALOCSW	AP	APTRAN	AQCNWT	ASYSVM
	BLANKS	BRING	CC3	C1	DEFER	DMKCVTBD	DMKCVTDB	DMKDSPQS	DMKERMSG	DMKFREE
	DMKFRET	DMKPAGDP	DMKPGMST	DMKPGTDF	DMKPGTDK	DMKPGTDL	DMKPGTDM	DMKPGTDP	DMKPGTPL	DMKPGTPN

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	DMKPGTQT	DMKPGTSL	DMKPGTST	DMKPTRAN	DMKPTRNS	DMKQCNWT	DMKSCHIB	DMKSCHPB	DMKSCHQB	DMKSCHQC
	DMKSCHWX	DMKSCNFD	DMKSWABS	DMKSWABT	DMKSWAQ1	DMKSWAQ2	DMKSYSSW	FFS	F0	F1
	F2	F3	F4	F8	HEADER	IOKEEP	LMSG	LOCK	L4	MP
	NORET	OFF	PSA	R0	R1	R10	R11	R12	R13	R14
	R15	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA
	SAVEREGS	SAVEWRK1	SAVEWRK2	SAVEWRK4	SYSPALL	SYSPEXT	SYSPEXTN	SYSPFLG	SYSPFPNT	SYSPFULL
	SYSPLIST	SYSPLVNO	SYSPTOTL	SYSTEM	SYSXMGCT	SYSXOVCT	SYSXRWCT	SYSXSHRT	VMBLOK	ZEROES
DMKSSP	ARIOCH	ARIOCT	ARIOCU	ARIODV	ATTN	BLANK	BLKLEN	BUSY	CAW	CC
	CD	CE	CLASDASD	CLASFBA	CLASGRAF	CLASTERM	CLASURI	CLASURO	CPUID	CSW
	CUE	C2	DE	DMKCPINT	DMKCVTBH	DMKCVTHB	DMKRIOCH	DMKRIOCN	DMKRIOCU	DMKRIODV
	DMKRIOPR	DMKRIOPU	DMKRIORD	DMKSYSNU	DMKSYSTP	D4	FTREXTSN	FTRUCS	INPUT	IONPSW
	IOOPSW	L3	L5	MAXLEN	MCNPSW	PRNPSW	PSA	RCHBLOK	RCHCUTBL	RCHSIZE
	RCUADD	RCUBLOK	RCUCHA	RCUCHB	RCUCHC	RCUCHD	RCUDVTBL	RCUSIZE	RCUTYPE	RDEVADD
	RDEVBLOK	RDEVCKDX	RDEVCLAS	RDEVCUA	RDEVFTR	RDEVGRTY	RDEVHT	RDEVLLEN	RDEVMDL	RDEVNAME
	RDEVSIZE	RDEVSTA3	RDEVTYPC	RDEVTYPE	RDEVWTH	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SCAN	SENSE	SILI	SM	SYSTEM	TYPPRT	TYPPUN	TYP2314
	TYP3066	TYP3210	TYP3277	TYP3278	TYP3340	TYP3350	TYP3375	TYP3380	UC	UE
	XRIGHT16									
DMKSSS	ADSPCH	AFREE	AFREEP	AFRET	ALOKSP	AP	APSTAT1	APTRAN	APTRLK	APUOPER
	ARIODC	ARIODV	ASYSVM	ATTN	BLANKS	BRING	CLASDASD	CLEAR	CPEXADD	CPEXBLOK
	CPEXMISC	CPEXREGS	CPEXR0	CPEXR11	CPEXR12	CPEXR14	CPEXR3	CPEXSIZE	C1	DEFER
	DMKCVTBH	DMKDSPCH	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKLNKSS	DMKLOKIO	DMKLOKSW	DMKLOMSS
	DMKPGUVG	DMKPGUVR	DMKPTRAN	DMKRPAGT	DMKSCNAU	DMKSCNRU	DMKSCNVS	DMKSCNVU	DMKSSTFV	DMKSTKCP
	DMKVDAS1	DMKVDAS2	DMKVIOMK	FFS	F0	F8	IOERETN	LOCK	LOCKSAV	LPUADDR
	L2	L4	MP	PSA	RCHADD	RCHBLOK	RCUADD	RCUBLOK	RCUCHA	RCUPRIME
	RCUSUB	RCUTYPE	RDEVADD	RDEVATT	RDEVBLOK	RDEVCUA	RDEVDED	RDEVDISA	RDEVFLAG	RDEVFTR
	RDEVLNKS	RDEVOWN	RDEVSEL	RDEVSER	RDEVSIZE	RDEVSTAT	RDEVSYS	RDEVTYPC	RDEVTYPE	RDEVUSER
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVERETN	SAVER1
	SAVER11	SAVER2	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8
	SAVEWRK9	SYSTEM	SYSVIRT	TIMEDISP	TYP3330	VIRTUAL	VMBLOK	ZERO	ZEROES	
DMKSST	ALOKSP	AP	APSTAT1	APTRAN	APTRLK	APUOPER	ASYSVM	BLANKS	BRING	CLASDASD
	C1	DEFER	DMKERMSG	DMKLOKIO	DMKPGTPG	DMKPGUVG	DMKPGUVR	DMKPTRAN	DMKRPAGT	DMKRPAPT
	DMKSCNRU	DMKSSSHR	DMKSSSNS	DMKSSSNV	FFS	F0	F1	F16	F4	F4096
	F8	IOERETN	LOCK	LOCKSAV	LPUADDR	MP	MSSPRES	PSA	PSAMSS	RDEVBLOK
	RDEVDED	RDEVDISA	RDEVFLAG	RDEVFTR	RDEVNRDY	RDEVOWN	RDEVSEL	RDEVSER	RDEVSTAT	RDEVSYS
	RDEVTYPC	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER2
	SAVER3	SAVER5	SAVER6	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7
	SAVEWRK8	SAVEWRK9	SYSTEM	SYSVIRT	TYP3330	VMBLOK				
DMKSSU	ADSPCH	AFREE	AFREEP	AFRET	ALOKSP	AP	APSTAT1	APUOPER	ASYSVM	CPEXADD
	CPEXBLOK	CPEXREGS	CPEXSIZE	DE	DMKCVTBH	DMKDSBRD	DMKDSPCH	DMKERMSG	DMKFREE	DMKFREEP
	DMKFRET	DMKLOKIO	DMKSCHRT	DMKSCHST	DMKSCNVD	DMKSSSMQ	DMKSTKCP	DMKSTKIO	F1	INTTIO
	IOBBPNT	IOBCC3	IOBCSW	IOBCTRQ	IOBCYL	IOBFATAL	IOBFLAG	IOBFLT	IOBFPNT	IOBIRA
	IOBLINK	IOBLOK	IOBRADD	IOBRSTRT	IOBSIZE	IOBSPEC	IOBSTAT	IOBUNSL	IOBUSER	IOBVADD
	LOCK	LOCKSAV	LPUADDR	MP	PSA	RDEVBLOK	RDEVCFLT	RDEVFIOB	RDEVQCNT	RDEVSTA6
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R8	SAVEAREA	SAVEREGS	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5
	SAVEWRK6	SAVEWRK7	SAVEWRK8	TEMPR10	TEMPR12	TEMPR14	TEMPSAVE	TRQBIRA	TRQBLOK	TRQBSIZE
	TRQBTOD	TRQBUSER	TRQBVAL	VMBLOK						

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
DMKSSV	AFRET	ALOKSP	AP	APSTAT1	APTRAN	APUOPER	ARIODC	ARIODV	BLANKS	BRING
	CPEXBLOK	CPEXR0	CPEXR1	CPEXR11	CPEXR13	CPEXSIZE	C1	DEFER	DMKCVTBH	DMKERMSG
	DMKFRET	DMKLOKIO	DMKLOKSW	DMKPGUSP	DMKPTRAN	DMKSCNAU	DMKSCNRD	DMKSCNRU	DMKSSSCA	DMKSSSCV
	DMKSSSHR	DMKSSSMQ	DMKSSSNS	DMKSSSVM	DMKSSTBL	DMKSSTFV	DMKSTKCP	F0	F4	F8
	LOCK	LOCKSAV	LPUADDR	MP	MSSPRES	PREFIXB	PSA	PSAMSS	RDEVBLOK	RDEVFLAG
	RDEVSEL	RDEVSER	RDEVSIZE	RDEVVMNT	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	SAVEWRK2	SAVEWRK3	SAVEWRK9	SCAN	TIMEDISP	VMBLOK		
DMKSTA	ACORETBL	AP	ASYSVM	BALRSAVE	CORCP	CORDISA	CORFLAG	CORFPNT	CORFREE	CORIOLCK
	CORLCNT	CORTABLE	CPCREG0	CPSIMLTB	CPSTAT4	CPUID	CPXSTOR	C0	DMKCPEND	DMKCPIBD
	DMKCPIFL	DMKCPISW	DMKCPITR	DMKCPJST	DMKCPJTA	DMKDMPAA	DMKDMPMA	DMKDMPSA	DMKDSPNP	DMKDSPN2
	DMKFREAP	DMKFREHI	DMKFREHP	DMKFRELO	DMKFRELP	DMKFREPP	DMKFRTTR	DMKPAGHI	DMKPAGLO	DMKPRGIN
	DMKPSAST	DMKPTRFN	DMKPTRF1	DMKPTRF2	DMKPTRNS	DMKPTRN2	DMKPTRRM	DMKSAV	DMKSELCP	DMKSELSA
	DMKSELSB	DMKSELSG	DMKSVCHI	DMKSVCLO	DMKSYSAP	DMKSYSFP	DMKSYSNP	DMKSYSRC	DMKSYSRM	DMKSYSRV
	DMKSYSSZ	DMKSYSTR	DMKVRSSV	EXTMODE	FF	FFS	F1	F10	F255	F4
	F4095	F4096	F8	INTPR	LOCK	L2048	MCHEK	MP	OLDKEYOP	PAGESIZE
	PREFIXA	PRIMEHDR	PRIMEHI	PRIMELO	PRNPSW	PROPSW	PSA	RUNCR0	R0	R1
	R10	R11	R12	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SAVESIZE	TRACCURR	TRACEFLG	TRACEND	TRACSTRT	TRACSVST	VMBLOK
	XKEYMODE	XPAGNUM								
DMKSTD	R12									
DMKSTK	AEXTSP	ALOKSP	AP	APSTAT1	APUOPER	CPEXBLOK	CPEXBPNT	CPEXDEFR	CPEXMISC	CPEXPRIO
	CPEXPROC	CPEXR11	CPEXTYPE	CPRUN	CPSTATUS	CPUID	CPWAIT	DMKLOKSY	IOBFPNT	IOBLOK
	IOBUSER	IPUADDRX	LOCK	LOKREQ	LPUADDR	LPUADDRX	MP	PREFIXB	PSA	PSADSPRQ
	PSAEXT	PSAEXTX	PXA	RQLOCK	R0	R1	R10	R11	R12	R14
	R15	R2	R3	R4	R5	R6	R7	R8	SIGDISP	SIGXC
	SWTHSAVE	VMBLOK	XCDISP	XCPEND						
DMKSTP	AFREE	AFREEP	ALOCALOC	ALOCBLOK	ALOCCHN	ALOCMALC	ALOCMGOU	ALOCPAVL	ALOCPRD	ALOCPS
	ALOCPUSE	ALOCPWRT	AP	APSTAT1	APUOPER	ASYSVM	CC	CD	CPEXADD	CPEXBLOK
	CPEXR0	CPEXR11	CPEXR12	CPEXSIZE	DMKCVTAB	DMKDSPNP	DMKDSPN2	DMKDSPQS	DMKFREE	DMKFREEP
	DMKFREHI	DMKLOKSW	DMKPAGPS	DMKPGMEP	DMKPGTDF	DMKPGTMX	DMKPGTPN	DMKPGTQT	DMKPTREE	DMKPTRER
	DMKPTRES	DMKPTRPR	DMKPTRRC	DMKPTRRF	DMKPTRSC	DMKPTRSS	DMKPTTHL	DMKPTTLH	DMKSCHAL	DMKSCHAP
	DMKSCHBS	DMKSCHCA	DMKSCHCO	DMKSCHCU	DMKSCHDL	DMKSCHET	DMKSCHFS	DMKSCHIS	DMKSCHKA	DMKSCHLI
	DMKSCHNS	DMKSCHPB	DMKSCHPG	DMKSCHQB	DMKSCHQC	DMKSCHQT	DMKSCHQ1	DMKSCHQ2	DMKSCHRL	DMKSCHSC
	DMKSCHSM	DMKSCHST	DMKSCHS1	DMKSCHS2	DMKSCHUC	DMKSTDAT	DMKSTKCP	DMKSTRSM	DMKSWABS	DMKSWABT
	DMKSWAIS	DMKSWAOP	DMKSWAOS	DMKSWART	DMKSYSNM	DMKSYSRM	DMKSYSSW	DMKSYSSZ	DMKTRQIL	DMKTRQTI
	F0	F1	F2	F3	IDLEWAIT	IONTWAIT	LOCK	MP	NOALCARY	NOSLBORO
	PAGELOAD	PAGERATE	PAGEWAIT	PGREAD	PGSRATIO	PGWAITIM	PGWRITE	PREFIXA	PREFIXB	PROCIPL
	PSA	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVERETN
	SAVER0	SAVER1	SAVER12	SAVER13	SAVEWRK1	SAVEWRK2	SAVEWRK4	SAVEWRK6	SAVEWRK8	SAVEWRK9
	STARTIME	STPDRP	SYSPALL	SYSPALOC	SYSPEXT	SYSPEXTN	SYSPFLG	SYSPFPNT	SYSPFULL	SYSPLIST
	SYSPMGOU	SYSPOVFL	SYSPPCNT	SYSPPUSE	SYSPTOTL	SYSXCDEL	SYSXCKP	SYSXLONG	SYSXPDEL	SYSXRWCT
	SYSXSHRT	SYSXSIZE	TEMPR0	TEMPR1	TEMPR10	TEMPR2	TEMPR3	TEMPR4	TEMPR5	TEMPR6
	TEMPR7	TEMPSAVE	TIMEDISP	TRQBFPNT	TRQBIRA	TRQBLOK	TRQBSIZE	TRQBTOD	TRQBUSER	TRQBVAL
	TTSEGCNT	VMBLOK	XMIGACT	XMIGSWT	XPGFULL	XPPFULL	XRIGHT16	XSWFULL		
DMKSTR	ACORETBL	ADSPCH	AFREEP	ALOCBLOK	ALOCFLG	ALOCPG	ALOCPP	ALOKRM	ALOKSP	APSTAT1
	APUOPER	ASYSVM	CORBPNT	CORFLAG	CORFPNT	CORFREE	CORIOLCK	CORLCNT	CORPGPNT	CORSWPNT
	CORTABLE	CPEXADD	CPEXBLOK	CPEXFPNT	CPEXMISC	CPEXR0	CPEXR12	CPEXR13	CPEXR4	CPEXR9

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	CPEXSIZE	CPSTAT4	CPXSTOR	C1	DELPAGES	DMKBLDRL	DMKBLDRT	DMKDSPCH	DMKFREEP	DMKLOKSW
	DMKPAGIO	DMKPAGQ	DMKPGTMX	DMKPGUAL	DMKPTRAQ	DMKPTREP	DMKPTRER	DMKPTRFN	DMKPTRFQ	DMKPTRN2
	DMKPTRPR	DMKPTRRQ	DMKPTSAD	DMKPTSAE	DMKPTTCL	DMKPTTDM	DMKSCHDL	DMKSCHMS	DMKSTKCP	DMKSYSCS
	F0	F16	F4096	F8	LOCK	LOCKSAV	LPUADDR	MICBLOK	MICVTMR	NEWPAGES
	PAGCORE	PAGINVAL	PAGPGSWP	PAGRBITS	PAGREF	PAGSWP	PAGTABLE	PAGTBSIZ	PAGTONLY	PGREAD
	PSA	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER0
	SAVER1	SAVER11	SAVER2	SAVER3	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SEGENQ
	SEGFLAG	SEGINV	SEGMIG	SEGMIGPG	SEGMIGPP	SEGPAGE	SEGPLEN	SPECIALV	SSBALLOC	SSBENTRL
	SSBENTRY	SSBENUM	SSBFLAG	SSBHEADL	SSBLOK	SSBTRANS	SWPALLOC	SWPCYL	SWPFLAG	SWPPAG
	SWPRECMP	SWPSEGNO	SWPTABLE	SWPTRANS	SWPVM	SWPVPAGE	TIMEDISP	TIMER	VMBLOK	XKEYMODE
	XMIGSWT	XRIGHT24								
DMKSVC	ADSPCH	AFREEP	AFRET	ALOKSY	APAGCP	APSTAT1	APTRAN	APTRLK	APUOPER	ASYSVM
	BRING	CODE	CPABEND	CPASTAVL	CPASTON	CPEXADD	CPEXBLOK	CPEXFPNT	CPEXMISC	CPEXPROC
	CPEXREGS	CPEXR12	CPEXSIZE	CPSTATUS	CPSTAT2	CPSUPER	C1	DEFER	DMKDMPDK	DMKDSPCH
	DMKDSPRU	DMKFREEP	DMKFRELP	DMKFRESV	DMKFRET	DMKLOKIO	DMKLOKSY	DMKPTRAN	DMKPTRUL	DMKSTKLF
	DMKSTKSW	DMKSVDIN	DUMPSAVE	FREEXT	FRPALLOC	FRPMCALR	FRPMEXT	FRPMNAME	FRPMSIZE	FRPMTAG
	FRPRETRN	FRPSIZE	FRTAG	F0	F4095	INTSVC	INTSVCL	IOERETN	LOCK	LOCKSAV
	LPUADDR	LPUADDRX	PREFIXA	PREFIXB	PRIMEHDR	PRIMEHI	PRIMELO	PROBMODE	PROCIPL	PSA
	PSACPXBP	R0	R1	R11	R12	R13	R14	R15	R2	R3
	R4	R7	R8	SAVEAREA	SAVEPROC	SAVERETN	SAVERTN	SAVER0	SAVER12	SAVER13
	SAVESIZE	SAVETRAC	SAVEWRK2	SVCNPSW	SVCOPSW	SVCREGS	SYSTEM	TEMPR14	TRACCURR	TRACEND
	TRACFLG1	TRACPROC	TRACSTRT	TRACSVCR	TRAC02	TRCSVC	VMBLOK	XCDISP	XCPEND	ZEROES
DMKSVD	ADSPCH	AFREE	AFRET	ALOKSY	AP	APSTAT1	APTRAN	APUOPER	AQCNWT	BRING
	CODE	CPCREG0	CPCREG8	CPEXADD	CPEXBLOK	CPEXREGS	CPRUN	CPSTATUS	C0	C1
	C8	DEFER	DFRET	DMKCFMBK	DMKCVTBH	DMKDSPB	DMKDSPCH	DMKDSPRU	DMKFREE	DMKFRET
	DMKLOKDF	DMKLOKSY	DMKPERIL	DMKPRGRF	DMKPTRAN	DMKQCNWT	DMKSTKDE	DMKTMRPT	DMKTRCIT	DMKTRCPB
	DMKTRCSV	DMKVAULA	DMKVAURN	DMKVERD	DMKVERO	EXTMASK	EXTMODE	FFS	F0	F1
	F4	F60	INTSVC	INTSVCL	IOMASK	LAP370E	LOCK	LPUADDR	MCHEK	MP
	NORET	PERBLOK	PERDATON	PERFLAG	PERMODE	PREFIXA	PROBMODE	PSA	PSASVCCT	QUANTUM
	QUANTUMR	RUN	RUNCR0	RUNCR1	RUNPSW	RUNUSER	R0	R1	R10	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	SAS	SVCNPSW	SVCOPSW	SVCREGS	TEMPR7	TIMEDISP	TIMER	TRACCURR
	TRACEND	TRACFLG1	TRACPROC	TRACSTRT	TRACSVCR	TRAC02	TRANMODE	TRAPCR8	TRCSVC	TREXIN1
	TREXT	TRQBLOK	TRQBQUE	VMBLOK	WAIT	XRIGHT24	Y0	Y2	Y4	Y6
	ZEROES									
DMKSWA	ACORETBL	ADSPCH	AFREE	AFREEP	AFRET	ALOCBLOK	ALOCPSIO	ALOCPWRT	ALOKRM	ALOKSP
	AP	APSTAT1	APTRAN	APUOPER	ASYSVM	BRING	CODE	CORBPNT	CORFLAG	CORFLUSH
	CORFPNT	CORLCNT	CORSWAP	CORSWPNT	CORTABLE	CORVM	CPEXADD	CPEXBLOK	CPEXFPNT	CPEXMISC
	CPEXR0	CPEXSIZE	C1	DMKDSPCH	DMKFREE	DMKFREEP	DMKFRET	DMKPAGIO	DMKPGTSB	DMKPGTSW
	DMKPGUAL	DMKPGUPP	DMKPGUSW	DMKPTRAN	DMKPTRAQ	DMKPTROQ	DMKPTRPS	DMKPTRSW	DMKPTRS2	DMKPTRU1
	DMKPTSAD	DMKPTTFT	DMKSYSCS	DMKSYSSZ	F1	F8	LASTUSER	LOCK	LOCKSAV	LPUADDR
	MICBLOK	MICVTMR	MNCLSWAP	MNCODEOS	MNCODLSI	MNCODSOS	MP	PAGCORE	PAGINVAL	PAGREF
	PREFIXA	PSA	PSAEXT	PSTPOUSW	PXA	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SAVER11	SAVER2	SAVESIZE	SAVEWRK2	SAVEWRK3	SAVEWRK5	SAVEWRK6
	SAVEWRK8	SEGINV	SEGPAGE	SSBBPNT	SSBCODE	SSBCYL	SSBEFLG	SSBEINVL	SSBENTRL	SSBENTRY
	SSBENUM	SSBEPGS	SSBEXTND	SSBFLAG	SSBFLUSH	SSBFPNT	SSBHEADL	SSBLOK	SSBMGREF	SSBNDLCT
	SSBNPGS	SSBOLD	SSBPGSP	SSBPSTOR	SSBSWAP	SSBTRANS	SSBUPGS	SSBVMSCB	SSBVPAGE	SWPCHG1
	SWPCHG2	SWPCYL	SWPFLAG	SWPRECMP	SWPSEGNO	SWPTABLE	SWPTRANS	SWPVPAGE	TIMEDISP	TIMER
	TRACCURR	TRACEND	TRACFLG1	TRACPROC	TRACSTRT	TRACSVCR	TRAC08	TRCSWAPI	VFAULT	VMBLOK

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)								
DMKSWM	ACORETBL	AFREE	AFREEP	AFRET	ALOCBLOK	ALOCMALC	ALOCMGIN	ALOCMGOU	ALOCPRD	ALOCPSIO
	ALOCSW	ALOKRM	ALOKSP	APSTAT1	APUOPER	ASYSVM	CORBPNT	CORFPNT	CORPGPNT	CORSWPNT
	CORTABLE	CORVM	CPEXADD	CPEXBLOK	CPEXFPNT	CPEXMISC	CPEXREGS	CPEXSIZE	DMKERMSG	DMKFREE
	DMKFREEP	DMKFRET	DMKPAGIO	DMKPGMUS	DMKPGTPM	DMKPGTSF	DMKPGUAL	DMKPGUSW	DMKPTREP	DMKPTRFN
	DMKPTRN2	DMKPTROQ	DMKPTRPS	DMKPTRSW	DMKPTRS2	DMKPTSAD	DMKPTSAE	DMKPTTFT	DMKSCNAU	DMKSCNFD
	DMKSTRAN	DMKSWAIS	DMKSWAOS	DMKSYSCS	DMKSYSSW	F1	F6	F8	LOCK	LOCKSAV
	LPUADDR	PAGCORE	PAGINVAL	PAGREF	PSA	PSAEXT	PSTPINSW	PXA	REQUSER	R0
	R1	R10	R11	R12	R13	R14	R15	R2	R3	R4
	R5	R6	R7	R9	SAVEAREA	SAVEREGS	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4
	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SEGFLAG	SEGMIG	SEGPAGE	SSBALLOC	SSBCYL	SSBEFLG
	SSBEINVL	SSBENTRL	SSBENTRY	SSBEPGS	SSBEXTND	SSBFLAG	SSBHEADL	SSBLOK	SSBNDLCT	SSBNPGS
	SSBTRANS	SSBUPGS	SSBVPAGE	SWPCHG1	SWPCHG2	SWPFLAG	SWPRECMP	SWPTRANS	SYSPFLG	SYSPFPNT
	SYSPLIST	VMBLOK								
DMKSYM	DMKACR	DMKCCHIS	DMKCCHNT	DMKCCHRT	DMKCCH60	DMKCCWSB	DMKCCWTR	DMKCFMAT	DMKCFMBK	DMKCFMEN
	DMKCNS	DMKCPEID	DMKCPEML	DMKCPEND	DMKCPEPP	DMKCVT	DMKDADER	DMKDASER	DMKDGDDK	DMKDIDEP
	DMKDMPDK	DMKDSBRD	DMKDSBSD	DMKDSPA	DMKDSPB	DMKDSPCH	DMKDSPE	DMKDSPEC	DMKDSPIT	DMKDSPNP
	DMKDSPN2	DMKDSPQS	DMKDSPQS	DMKDSPRU	DMKDSPWA	DMKENT	DMKENTIM	DMKEXTIN	DMKEXTSP	DMKFPS
	DMKFREAP	DMKFREE	DMKFREEA	DMKFREEP	DMKFREHI	DMKFRELG	DMKFRELN	DMKFRELO	DMKFRELP	DMKFRELS
	DMKFREMX	DMKFRENP	DMKFREPP	DMKFRERC	DMKFRESC	DMKFRESV	DMKFRET	DMKFRETL	DMKFRTRS	DMKFRTTR
	DMKGRCQY	DMKGRCUP	DMKGRF	DMKGRHIN	DMKGRT	DMKHVCAL	DMKHVCDI	DMKIOEFM	DMKIOERR	DMKIOSHA
	DMKIOSMQ	DMKIOSNM	DMKIOSQR	DMKIOSQV	DMKIOTCT	DMKIOTIN	DMKLOKDF	DMKLOKDS	DMKLOKFR	
	DMKLOKPS	DMKLOKRL	DMKLOKSP	DMKLOKSW	DMKLOKSY	DMKLOKTR	DMKLOKVM	DMKMCHIN	DMKMCTFS	DMKMCTMA
	DMKMCTPR	DMKMCTPT	DMKMCTST	DMKMSWR	DMKOPRWT	DMKPAGIO	DMKPAGPS	DMKPAGSS	DMKPGTDF	DMKPGTDK
	DMKPGTDM	DMKPGTPC	DMKPGTPG	DMKPGTPL	DMKPGTPN	DMKPGTPT	DMKPGTSF	DMKPGTTM	DMKPGTTU	DMKPGUBN
	DMKPMA	DMKPMACD	DMKPMAEW	DMKPMAEX	DMKPMAIN	DMKPMAI1	DMKPMAI2	DMKPMAMC	DMKPMASW	DMKPMAWT
	DMKPMAXF	DMKPMAXS	DMKPRGCT	DMKPRGC8	DMKPRGIN	DMKPRGMC	DMKPRGRF	DMKPRGSM	DMKPRVCD	DMKPRVCH
	DMKPRVCP	DMKPRVCS	DMKPRVCT	DMKPRVDI	DMKPRVEK	DMKPRVEP	DMKPRVIK	DMKPRVIP	DMKPRVLC	DMKPRVLG
	DMKPRVLP	DMKPRVLR	DMKPRVMN	DMKPRVMO	DMKPRVMS	DMKPRVNC	DMKPRVPB	DMKPRVPE	DMKPRVRR	DMKPRVTC
	DMKPRVTE	DMKPRWIP	DMKPRWTP	DMKPSADU	DMKPSANX	DMKPTRAN	DMKPTRCO	DMKPTRCS	DMKPTRCT	DMKPTREE
	DMKPTREF	DMKPTRES	DMKPTRFA	DMKPTRFC	DMKPTRFF	DMKPTRFN	DMKPTRFP	DMKPTRFQ	DMKPTRFR	DMKPTRFO
	DMKPTRF1	DMKPTRF2	DMKPTRIQ	DMKPTRLK	DMKPTRLX	DMKPTRNF	DMKPTRNS	DMKPTRN2	DMKPTROQ	DMKPTRPL
	DMKPTRPM	DMKPTRPO	DMKPTRPQ	DMKPTRPR	DMKPTRPS	DMKPTRQ2	DMKPTRRC	DMKPTRRF	DMKPTRRM	DMKPTRRQ
	DMKPTRRW	DMKPTRSC	DMKPTRSS	DMKPTRSW	DMKPTRS2	DMKPTRUL	DMKPTRUX	DMKPTRU1	DMKPTRWQ	DMKPTTHL
	DMKPXA	DMKPXB	DMKQCNWT	DMKQCOCL	DMKQCOET	DMKQCORD	DMKQCOSY	DMKQVMEP	DMKRGAIN	DMKRGBEN
	DMKRGBIC	DMKRGC	DMKRGDOB	DMKRGDOI	DMKRIOCH	DMKRIOCN	DMKRIOCU	DMKRIODV	DMKRIOPR	DMKRIOPU
	DMKRIORD	DMKRIORN	DMKRNHIC	DMKRPAGT	DMKRPAPT	DMKRSPAC	DMKRSPCV	DMKRSPDL	DMKRSPER	DMKRSPEX
	DMKRSPHQ	DMKRSPPR	DMKRSPPU	DMKRSPRD	DMKRSP83	DMKSCHAL	DMKSCHAP	DMKSCHCT	DMKSCHDL	DMKSCHIB
	DMKSCHN1	DMKSCHN2	DMKSCHPB	DMKSCHPG	DMKSCHPU	DMKSCHQ1	DMKSCHQ2	DMKSCHRL	DMKSCHRT	DMKSCHST
	DMKSCHTQ	DMKSCHW1	DMKSCHW2	DMKSCNRU	DMKSELS1	DMKSTKCP	DMKSTKDE	DMKSTKIO	DMKSTKLF	DMKSTKMP
	DMKSTKOP	DMKSTKSW	DMKSVCIN	DMKSVCNS	DMKSWAIL	DMKSYSCS	DMKSYSLC	DMKSYSNM	DMKSYSOC	DMKSYSOP
	DMKSYSOW	DMKSYSRM	DMKSYSRS	DMKSYSRV	DMKSYSSW	DMKSYSVL	DMKSYSVM	DMKTBL	DMKTMRTN	DMKTRKFP
	DMKTRKIN	DMKTRKVA	DMKTRQCP	DMKTRQMD	DMKTRQTI	DMKTRQ80	DMKUNTFB	DMKUNTFR	DMKUNTIS	DMKUNTRN
	DMKUNTRS	DMKVATAB	DMKVATBC	DMKVATEX	DMKVATMD	DMKVATPX	DMKVATSX	DMKVAULA	DMKVAURN	DMKVCNEX
	DMKVCNFT	DMKVCPIL	DMKVCRMT	DMKVCTCH	DMKVCVER	DMKVCXIO	DMKVFRCH	DMKVIOIN	DMKVMASH	DMKVRR
	DMKVRRDD	DMKVRRIS	DMKVRRRC	DMKVRRRS	DMKVRSSV	DMKVSC	DMKVSICI	DMKVSICW	DMKVSIEX	DMKVSISF
	DMKVSISI	DMKVSITC	DMKVSITI	DMKVSJHD	DMKVSJHI	DMKVSPEX	DMKVSPWA	DMKVSQPD	DMKVSRGC	DMKVSRMD
	DMKVSTOP	DMKVSWDC	DMKVSWFC	DMKVSWTO	DMKWAIST	DMKWAITA	LOCK	VMBLOK		
DMKTAP	ADSPCH	AFREE	AFREEP	AFRET	APTRLK	CCC	CD	CDC	CHC	CONTINUE
	CPEXADD	CPEXBLOK	CPEXFPNT	CPEXREGS	CPEXR0	CPEXR12	CPEXSIZE	CUE	DE	DMKDSPCH
	DMKFREE	DMKFREEP	DMKFRET	DMKIOERR	DMKIOEST	DMKIOSQR	DMKMSWR	DMKPTRLK	DMKPTRUL	DMKSTKCP
	DMKTAQRE	DMKTAQRP	DMKTAQSE	FFS	FTRRSRL	F0	F1	F15	F16	F3
	F5	F6	F8	IFCC	IOBCAW	IOBCC3	IOBCP	IOBCSW	IOBERP	IOBFATAL
	IOBFLAG	IOBHVC	IOBIOER	IOBIRA	IOBLINK	IOBLOK	IOBMISC	IOBRADD	IOBRCAW	IOBRCNT

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	IOBRSTRT	IOBSIZE	IOBSPEC	IOBSTAT	IOBTIO	IOBUNSL	IOBUSER	IOERACT	IOERBLOK	IOERBSR
	IOERCAN	IOERCCRA	IOERCCRL	IOERCEMD	IOERCLN	IOERCPEX	IOERCSW	IOERDATA	IOERDW	IOERERG
	IOEREXT	IOERFLG1	IOERFLG2	IOERFLG3	IOERFLM	IOERFSR	IOERIGNR	IOERIND3	IOERIND4	IOERINFO
	IOERLOC	IOERMSG	IOERMSW	IOERNUM	IOERORA	IOERPEND	IOERPNT	IOERRBK	IOERREAD	IOERREW
	IOERSIZE	IOERSTAT	IOERSTRT	LOCK	NOTUSED	PRGC	PRTC	PSA	RDEVBLOK	RDEVCUB
	RDEVFTR	RDEVIOER	RDEVNRDY	RDEVSTAT	RDEVSTA3	RDEVSTMD	RDEVTYPE	REWIND	R0	R1
	R10	R11	R12	R13	R14	R15	R2	R3	R4	R5
	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER0	SAVEWRK1	SAVEWRK4	SILI
	TYP2401	TYP2415	TYP2420	TYP3422	TYP3430	TYP8809	UC	UE	XRIGHT16	ZEROES
DMKTAQ	AFREE	AFRET	AP	CC	CCC	CD	CDC	CHC	CPEXBLOK	CPEXFPNT
	DMKFREE	DMKFRET	DMKIOEST	DMKMSWR	DMKSTKCP	FFS	F1	F16	F2	F4
	F5	F8	IDA	IFCC	IL	IOBCAW	IOBCP	IOBCSW	IOBERP	IOBFATAL
	IOBFLAG	IOBIOER	IOBLOK	IOBRCAW	IOBRCNT	IOBRSTRT	IOBSTAT	IOERADR	IOERBLOK	IOERBSR
	IOERCAN	IOERCCRA	IOERCCRL	IOERCLN	IOERCPEX	IOERCSW	IOERDATA	IOERDW	IOERERG	IOEREXT
	IOERFLG1	IOERFLG2	IOERFLG3	IOERFSR	IOERIGNR	IOERIND3	IOERIND4	IOERINFO	IOERLOC	IOERMSW
	IOERNUM	IOERORA	IOERPEND	IOERRBK	IOERREAD	IOERREW	IOERSIZE	IOERSTRT	IOERSUPP	IOERVLD
	IOERWRK	LOCK	MP	NOTUSED	PRGC	PRTC	PSA	RDEVBLOK	RDEVTYPE	REWIND
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVEWRK1	SAVEWRK4
	SILI	SKIP	TRACK7	TYP2401	TYP2415	TYP2420	TYP3410	TYP3420	TYP3422	TYP3430
	TYP8809	ZEROES								
DMKTCS	ADSPCH	AFREE	AFRET	AP	APSTAT1	APTRAN	APTRLK	APUOPER	ASYSVM	BRING
	CC	CLASFBA	C1	DEFER	DMKCVTBD	DMKCVTBH	DMKDSPCH	DMKERMSG	DMKFREE	DMKFRET
	DMKIOSQR	DMKLOKSW	DMKPGUVG	DMKPGUVR	DMKPTRAN	DMKRPAGT	DMKSCNAU	DMKSCNRD	DMKSCNVS	DMKSEPTL
	DMKSNTQN	DMKTCTET	F0	F1	F2	F256	F3	F4095	F4096	HLDADDR1
	HLDAREA	HLDCCW1	HLDCCW2	HLDCCW3	HLDCCW4	HLDCCW5	HLDCCW6	HLDCCW7	HLDCHARS	HLDCHAR1
	HLDCHAR2	HLDCHAR3	HLDCPY	HLDFCB	HLDFLAG	HLDFLSHC	HLDMCHR	HLDMDFY	HLDSIZE	HLDSTCPY
	IDA	IOBCAW	IOBFATAL	IOBFLAG	IOBIRA	IOBLINK	IOBLOK	IOBMISC	IOBMISC2	IOBRSTRT
	IOBSIZE	IOBSTAT	IOERETN	IOKEEP	LOCK	MP	NPRNAME	NPRPNT	NPRSTART	NPRTBL
	NPRVOL	PSA	RDEVBLOK	RDEVCODE	RDEVCURP	RDEVEXTN	RDEVFLAG	RDEVFSEP	RDEVIMAG	RDEVOWN
	RDEVPURG	RDEVSTA2	RDEVTYPC	RDEVTYPE	RDEVXSEP	RSPDPAGE	RSPLCTL	RSPRPAGE	RSPRPAG2	RSPVPAGE
	RSPXBLOK	RSPXCHR	RSPXCHR1	RSPXCHR2	RSPXCHR3	RSPXCMOD	RSPXFCB	RSPXMCHR	RSPXVTRC	R0
	R1	R10	R11	R12	R13	R14	R15	R2	R3	R4
	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER11	SAVEWRK1	SAVEWRK2
	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	SFBCOPY	SFBFILID	SFBLOK
	SFBSTART	SFBSTCPY	SFBUSER	SILI	SKIP	SPCHAR	SPCHAR1	SPCHAR2	SPCHAR3	SPCMCHR
	SPCMOD	SPCOPYFG	SPFCB	SPFLAG1	SPFLSHC	SPLINK	SPNXTPAG	SPPREPAG	SYSTEM	TIMEDISP
	TYP2314	TYP3330	TYP3350	TYP3375	TYP3380	TYP3800	TYP38003	TYP38008	VMBLOK	XPAGNUM
	ZEROES									
DMKTCT	ADSPCH	AFRET	APSTAT1	APTRAN	APTRLK	APUOPER	ASYSVM	BLANKS	BRING	CC
	C1	DEFER	DMKCVTBD	DMKCVTBH	DMKDSPCH	DMKERMSG	DMKFRET	DMKIOSQR	DMKLOKSW	DMKPGUVG
	DMKPGUVR	DMKPTRAN	DMKRPAGT	DMKSCNAU	DMKSCNRD	DMKTCSML	FTR4WCGM	F0	F1	F16
	F2	F3	HLDADDR1	HLDAREA	HLDCCUU	HLDCCW1	HLDCCW2	HLDCCW3	HLDCCW4	HLDCCW5
	HLDCCW6	HLDCHARS	HLDCHAR1	HLDCHAR2	HLDCHAR3	HLDHLPU	HLDMNAM	HLDSFID	IDA	IOBCAW
	IOBFATAL	IOBFLAG	IOBIRA	IOBLINK	IOBLOK	IOBMISC	IOBMISC2	IOBRSTRT	IOBSIZE	IOBSTAT
	LOCK	PSA	RDEVBLOK	RDEVEXTN	RDEVFTR	RDEVPRFG	RDEVPURG	RDEVSEPF	RDEVSTA2	RDEVTYPE
	RSPLCTL	RSPRPAGE	RSPRPAG2	RSPSFBLK	RSPXBLOK	RSPXCHR	RSPXCHR1	RSPXCHR2	RSPXCHR3	RSPXCMOD
	RSPXFCB	RSPXOTRC	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS
	SAVER11	SAVER8	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SFBFILID	SFBLOK	SFBUSER
	SILI	SKIP	SYSTEM	TIMEDISP	TYP3800	TYP38003	TYP38008	VMBLOK	ZEROES	

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)								
DMKTDK	ALOCBLOK	ALOCCHN	ALOCCYL1	ALOCCYL2	ALOCMAP	ALOCMAX	ALOCRDEV	ALOCTDK	ALOCTDSK	ALOCUSED
	CLASFBA	CLEARBIT	DMKCPXCK	DMKPGTAE	DMKPGTAN	DMKPGTAT	DMKPGTAW	DMKPGTA0	DMKPGTA4	DMKPGTA5
	DMKPGT4A	DMKPGT5A	DMKPGT7A	DMKPGT8A	DMKSYSTD	F255	F256	LASTALOC	LOCK	L1
	PSA	RDCBLKAP	RDCBLOK	RDEVBLOK	RDEVLNKS	RDEVRDC	RDEVTYPC	RDEVTYPE	R0	R1
	R10	R11	R12	R13	R14	R15	R2	R3	R4	R5
	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER0	SAVER1	SAVER8	SAVEWRK1
	SAVEWRK2	SAVEWRK3	SAVEWRK4	TYP3310	TYP3330	TYP3340	TYP3350	TYP3375	TYP3380	
DMKTED	CODE									
DMKTEE	BLANKS	CODE	DEVICE	DMKTEDAT	DMKTEFSF	DMKTESUB	LOCK	PROBMODE	RUN	R1
	R10	R11	R12	R13	R14	R15	R4	R5	R6	R8
	R9	SUBADDR	SUBBDTE	SUBCSFLG	SUBCSWPT	SUBFBYTE	SUBLEVEL	SUBLIST	SUBREAL	SUBSTAT
	SUBVIRT	SUBVMBAD	SUBVMFLG	SYSTEM	TRCENTRY	TRCOUTLN	TRCOUTRN	TRCPARM	VMBLOK	
DMKTEF	CALLER	CC	DEVICE	DMKTEDAT	DMKTESUB	R1	R10	R11	R12	R13
	R14	R15	R4	R5	R8	R9	SUBADDR	SUBBDTE	SUBCSFLG	SUBCSWPT
	SUBFBYTE	SUBLEVEL	SUBLIST	SUBREAL	SUBSTAT	SUBVIRT	SUBVMBAD	SUBVMFLG	TRCENTRY	TRCOUTLN
	TRCOUTRN	TRCPARM	VMBLOK							
DMKTEM	DMKTEDAT	DMKTEESF	PROBMODE	R1	R10	R11	R12	R13	R14	R15
	R3	R4	R5	R8						
DMKTES	CSW	DMKTEDAT	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	VMBLOK	
DMKTHI	ADSPCH	AFREE	AFREEP	AFRET	APSTAT1	APUOPER	AQCNWT	ASYSVM	BLANKS	CONTINUE
	CPEXADD	CPEXBLOK	CPEXREGS	CPEXSIZE	DFRET	DMKCVTBD	DMKCVTBH	DMKDSPCH	DMKDSPNP	DMKDSPN2
	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKQCNWT	DMKSCHCA	DMKSCHCO	DMKSCHCU	DMKSCHEL	DMKSCHIS
	DMKSCHLI	DMKSCHNS	DMKSCHQT	DMKSCHQ2	DMKSCHRL	DMKSCHS1	DMKSCHS2	DMKSCNFD	DMKSCNRD	DMKSCNVU
	DMKSTKOP	DMKSWART	DMKSYSMP	DMKTMRPT	DMKVFRSV	F1	F3	F4	F60	F8
	IPUADDR	IPUADDRX	LOCK	MPUOPER	NORET	PAGELOAD	PAGERATE	PGSRATIO	PREFIXA	PREFIXB
	PROCIO	PROCIPL	PSA	RUNUSER	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	SAVER11	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK8
	VACOVFR	VCONCTL	VCONRDEV	VECOPVF	VECSTAT	VECUSER	VMBLOK			
DMKTMR	ADSPCH	AFREE	AFRET	ALOKSY	AP	APSTAT1	APTRAN	APUOPER	ATMRSN	BALR14
	BALR2	BRING	CHGREGS	CPCREG0	CPEXADD	CPEXBLOK	CPEXREGS	CPRUN	CPSTATUS	C0
	C1	DEFER	DMKCVTAB	DMKDSPA	DMKDSPCH	DMKDSPRU	DMKFREE	DMKFRET	DMKFRTLT	DMKFRTRS
	DMKLOKDF	DMKLOKSY	DMKPRGSM	DMKPSAFP	DMKPSASP	DMKPTRAN	DMKSCHQ1	DMKSCHQ2	DMKSCHRT	DMKSCHST
	DMKSTKDE	DMKSTKIO	DMKVATEX	DMKVAURN	EXTMASK	F0	F1	F4	F4095	F5
	F60	F7	LOCK	LPUADDR	MICBLOK	MICEVMA2	MICSPT	MP	PERADDR	PERBLOK
	PERCR9	PERSALT	PROBSTRT	PROBTIME	PSA	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	TRANMODE	TRQBFPNT	TRQBLOK	TRQBQUE	TRQBTOD	TRQBVAL	VMBLOK	ZEROES	
DMKTOD	AFREE	AP	APSTAT1	APUOPER	ASYSVM	BLANK	BRING	CC0	CC1	CC2
	CC3	CHNGMSG	CPID	CPSTAT5	C1	DATE	DEFER	DMKCQYID	DMKCQYIT	DMKCVTDT
	DMKDMPDT	DMKDMPTD	DMKERMCP	DMKFREE	DMKFRTLT	DMKMCHST	DMKPRGIN	DMKPTRAN	DMKQCORD	DMKQCOSY
	DMKSCHST	DMKSYSDW	DMKSYSTI	DMKSYSTZ	DMKTMRFR	DMKTRQMD	EDIT	EXTMODE	FFS	F0
	F1	F2	F3	F4	F5	F60	F7	F8	LOCK	LPUADDR
	LPUADDRX	L8	MCHEK	MP	NOAUTO	NORET	NOTIME	POWEROFF	PREFIXA	PREFIXB
	PRNPSW	PROCIPL	PSA	R0	R1	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	STARTIME	SYSTEM	TEMPSAVE	TODATE	TRQBIRA	TRQBLOK	TRQBSIZE	TRQBTOD	TRQBUSER	TRQBVAL
	UCASE	VMBLOK	ZEROES							
DMKTPE	ADSPCH	AFREE	AFREEP	AFRET	BLANKS	CC	CCC	CD	CDC	CHC
	CMDREJ	CONTINUE	CPEXADD	CPEXBLOK	CPEXFPNT	CPEXR0	CPEXR12	CPEXSIZE	DMKDSPCH	DMKFREE
	DMKFREEP	DMKFRET	DMKMSWR	DMKSTKCP	F1	F6	F8	IFCC	IL	IOBASNCT
	IOBCAW	IOBCP	IOBDYNP	IOBERP	IOBFATAL	IOBFLAG	IOBIOER	IOBLOK	IOBPATHF	IOBPROC
	IOBRCAW	IOBRCNT	IOBRSTRT	IOBSPEC5	IOBSTAT	IOERACT	IOERBLOK	IOERBSR	IOERCAN	IOERCCRA
	IOERCCRL	IOERCPEX	IOERCSW	IOERDATA	IOERDW	IOERERG	IOEREXT	IOERFIXP	IOERFLG1	IOERFLG2
	IOERFLG3	IOERFSR	IOERIND3	IOERINFO	IOERLOC	IOERMSW	IOERNOLG	IOERNUM	IOERPEND	IOERSIZE
	IOERSPID	IOERTEMP	IOERVSER	LOCK	PRGC	PRTC	PSA	PSAPGID	RDEVBLOK	RDEVIOER
	RDEVNRDY	RDEVPROC	RDEVSER	RDEVSTAT	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R7	R8	SAVEAREA	SAVEREGS
	SAVER0	SILI	UC	VMBLOK	ZEROES					
DMKTRA	AFREE	AFRET	AQCNWT	AVMREAL	COUNT	CPSPMODE	CPSTAT2	CSW	C1	DMKERMSG
	DMKFREE	DMKFRET	DMKLOCKD	DMKLOCKQ	DMKQCNWT	DMKSCNFD	DMKTRCIT	DMKTRCPB	FFS	FLAG2
	F1	F2	F3	F8	LOCK	MICBLOK	MICVTMR	NORET	PSA	RUN
	R0	R1	R11	R12	R13	R14	R15	R2	R3	R4
	R5	R9	SAVEAREA	SAVEREGS	SAVER2	SAVEWRK1	SAVEWRK2	SAVEWRK7	TIMER	TREXANSI
	TREXBRAN	TREXCCW	TREXCSW	TREXCTL	TREXINST	TREXIN1	TREXPRNT	TREXRUNF	TREXSIZE	TREXT
	TREXTERM	VMBLOK	WAIT	XRIGHT16						
DMKTRC	AFRET	APTRAN	APTRLK	AQCNWT	BLANKS	BRING	CPCREG0	C0	C1	DEFER
	DMKATSCF	DMKCFMBK	DMKCVTBH	DMKFRET	DMKLOCKD	DMKLOCKQ	DMKNEMOP	DMKPSARR	DMKPSARS	DMKPSARX
	DMKPSASC	DMKPSASP	DMKPTRAN	DMKPTRLK	DMKPTRUL	DMKQCNWT	DMKVAURN	DMKVSTPT	EXTMASK	EXTMODE
	FFS	F0	F15	F16	F2	F60	F8	INTSVCL	IOBCSW	IOBLOK
	IOBRADD	IOBVADD	IOMASK	LOCK	NORET	PERMODE	PSA	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER0	SAVER1	SAVER2	SAVER4	SAVER5
	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	SVCNPSW
	SVCOPSW	TRANMODE	TREXANSI	TREXBRAN	TREXBUFF	TREXCSW	TREXCTL1	TREXCTL2	TREXFLAG	TREXINST
	TREXIN1	TREXIN2	TREXLCNT	TREXNDSP	TREXNSI	TREXPRNT	TREXRUNF	TREXSIZE	TREXSVC1	TREXSVC2
	TREXT	TREXTERM	TREXVAT	VMBLOK	WAIT	ZEROES				
DMKTRD	ADSPCH	AFREEP	AFRET	APTRAN	AQCNWT	BLANKS	BRING	CAW	CLASDASD	CLASFBA
	CLASGRAF	CLASSPEC	CLASTERM	CLASURI	CLASURO	CPCREG0	CPEXADD	CPEXBLOK	CPEXSIZE	CSW
	C0	C1	DEFER	DMKCCWSB	DMKCFMBK	DMKCVTBH	DMKDSPCH	DMKFREEP	DMKFRET	DMKLOCKD
	DMKLOCKQ	DMKNEMOP	DMKPTRAN	DMKQCNWT	DMKSCNRD	DMKSCNRN	DMKSCNVN	DMKSCNVU	DMKSTKCP	DMKSYSRM
	DMKVAURN	DMKVSTPT	EXTMODE	FFS	F0	F1	F15	F16	F2	F240
	F255	F3	F4	F60	F8	IDA	IOBCAW	IOBCSW	IOBLOK	IOBRADD
	IOBSTAT	LOCK	NORET	PSA	RCWCCW	RCWGEN	RCWPNT	RCWRCNT	RCWTASK	RCWVCAW
	RCWVCNT	RDEVBLOK	RDEVSNA	RDEVSNRB	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	SAVER0	SAVER1	SAVER2	SAVER8	SAVEWRK1	SAVEWRK2	SAVEWRK4	SAVEWRK6
	SAVEWRK7	SAVEWRK8	SAVEWRK9	SNARBLOK	SNARLUN	TRANMODE	TREXANSI	TREXBRAN	TREXBUFF	TREXCCW
	TREXCCWI	TREXCTL1	TREXCTL2	TREXFLAG	TREXINST	TREXIN1	TREXIN2	TREXLCNT	TREXMOR	TREXNDSP
	TREXPRNT	TREXRUNF	TREXSIZE	TREXSVC1	TREXSVC2	TREXT	TREXTERM	TREXVAT	VCONCTL	VCONNICB
	VMBLOK	X2048BND	ZEROES							
DMKTRK	ADSPCH	AFREE	AFREEP	AFRET	APTRAN	BRING	CC	CD	C1	DEFER
	DMKCCWSB	DMKDSPCH	DMKFREE	DMKFREEP	DMKFRET	DMKIOSQR	DMKIOSQV	DMKPSAPO	DMKPTRAN	DMKUNTRN
	FFS	F0	F1	F3	F4096	F6	F8	IL	IOBALTSK	IOBCAW
	IOBCC3	IOBCP	IOBCSW	IOBFATAL	IOBFLAG	IOBHVC	IOBIOER	IOBIRA	IOBLOK	IOBMISC2
	IOBRCAW	IOBRCNT	IOBRSTRT	IOBSIOF	IOBSIZE	IOBSPEC	IOBSTAT	IOBUSER	IOERADR	IOERALTR

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	IOERBLOK	IOERCEMD	IOERCSW	IOERDATA	IOERDW	IOEREXT	IOERFLG2	IOERFLG3	IOERLOC	IOERPNT
	IOERRDRO	IOERSIZE	IOERWRK	LOCK	L1	L2	L3	L4	L8	PCI
	PSA	RCWADDR	RCWCCNT	RCWCCW	RCWCNT	RCWCOMND	RCWCTL	RCWFLAG	RCWGEN	RCWHEAD
	RCWINVL	RCWPNT	RCWRCNT	RCWREL	RCWTASK	RDEVBLOK	RDEVIOER	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER1	SAVER7	SAVER8	SAVEWRK1	SAVEWRK2
	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SILI	SKIP	UC	VMBLOK
	XRIGHT16	ZEROES								
DMKTRM	BLANK	FF	F7	LOCK	PSA	RDEVATOF	RDEVBLOK	RDEVCORR	RDEVFLAG	RDEVIDNT
	RDEVPTTC	RDEVTFLG	RDEVTMCD	R0	R1	R11	R12	R13	R2	R3
	R4	R5	R8	SAVEAREA	SAVEREGS					
DMKTRP	ADSPCH	AFREE	AFRET	APTRAN	APTRLK	AQCNWT	ASYSVM	BLANKS	BRING	BUFCNT
	BUFFER	BUFNXT	CL	CPEXADD	CPEXBLOK	CPEXR0	CPEXSIZE	C1	DEFER	DMKCVTDB
	DMKCVTHB	DMKDSPCH	DMKERMSG	DMKFREE	DMKFRET	DMKPTRAN	DMKPTRUL	DMKQCNWT	DMKSCNAU	DMKSCNFD
	DMKSTKCP	DMKTRT	DMKTRUAC	DMKUDRFU	DMKUDRMD	DMKUDRRV	FFS	F0	F1	F2
	F3	F4	F8	HALF1	HALF1ENT	HALF1EN1	HALF1EN2	HALF1RLN	HALF1SZE	HALF1VAL
	HALF2	IOKEEP	LOCK	NAME	NAMEENV	NAMEMIN	NAMENTRY	NAMEROUT	NAMESIZE	NAMNOPRO
	NAMPROC	NAM2WORD	NORET	ORIGSEL	O2ENTRY	O2LENGTH	O2NAME	O2SIZE	O2TABEND	O2TABLE
	O3DISP	O3ENTRY	O3HEXLTH	O3MCHFLD	O3NUMSZ	O3NUM00	O3SIZE	O3TABLE	O3TRPEND	O4TABLE
	PREFIXA	PREFIXB	PSA	RESET	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	SAVER2	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK7	SAVEWRK8
	SELDATA	SELDISP	SELECT	SELENTRY	SELFORW	SELLTH	SELSIZE	SFBLOK	SFBRECNO	SYSTEM
	TRAPCP	TRAPDATA	TRAPSTRT	TRAPTT	TYPNUMAX	TYPNUMIN	VGPTR	VMBLOK	ZEROES	
DMKTRQ	ACTIVTRQ	ADSPCH	AFREE	AFRET	ALOKSP	APSTAT1	APUOPER	ASYSVM	C1	DMKCVTAB
	DMKDSPCH	DMKFREE	DMKFRET	DMKLOKSW	DMKLOKTR	DMKMIDNT	DMKSCHAL	DMKSCHDL	DMKSCHQ1	DMKSCHQ2
	DMKSCHRL	DMKSCHTQ	DMKSTPX	F1	LOCK	LOCKSAV	LPUADDR	MICBLOK	MICEVMA2	MICSPT
	NOALCARY	PREFIXB	PSA	PSAEXT	PSAEXTX	PXA	RQLOCK	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SAVEAREA	TEMPSAVE	TIMEDISP	TIMER	TMRDRP	TRQBBPNT	TRQBFPNT
	TRQBLOK	TRQBQUE	TRQBTOD	TRQBUSER	TRQBVAL	TRQCHN	VMBLOK			
DMKTRR	BLANK	BLANKS	CC3	CONTINUE	DMKTEMEP	EDIT	ERRMSG	HALF1	HALF1ENT	HALF1EN1
	HALF1EN2	HALF1RLN	HALF1SZE	HALF1VAL	HALF2	INTBLK	INTCTRL	INTLTH	INTMTYPE	INTNULL
	INTTCP	INTTMACH	INTTVG	INTTVT	INTTYPE	NAME	NAMEENV	NAMEMIN	NAMENTRY	NAMEROUT
	NAMESIZE	NAM2NUM	NAM2WORD	ORIGSEL	O2ENTRY	O2LENGTH	O2NAME	O2SIZE	O2TABEND	O2TABLE
	O3DISP	O3ENTRY	O3HEXLTH	O3MCHFLD	O3NUMSZ	O3NUM00	O3SIZE	O3TABEND	O3TABLE	O4ENTRY
	O4FMTADR	O4FORMAT	O4MINLEN	O4NAME	O4SIZE	O4TABEND	O4TABLE	O4VALUE	RESET	R0
	R1	R10	R11	R12	R13	R14	R15	R2	R3	R4
	R5	R6	R7	R8	R9	SELDATA	SELDISP	SELECT	SELENTRY	SELFORW
	SELLTH	SELSIZE	SFBDUMP	SFBFLAG	SFBLOK	SFBMISC1	SFBSTART	SPFILID	SPLINK	SPNXTPAG
	SPPREPAG	SPRMISC	SPSIZE	TTCODE	TTENTRY	TTLENGTH	TYPNUMAX	TYPNUMIN	WAIT	
DMKTRT	ADDSFB	ADSPCH	AFREE	AFREEP	AFRET	ASYSVM	BFFCCPD	BFFCLOSE	BFFENTRY	BFFFSIZE
	BFFPEND	BFFREAL	BFFSTAT	BFFSTOP	BFFVIRT	BRING	CPEXADD	CPEXBLOK	CPEXR0	CPEXR1
	CPEXSIZE	DMKCKSPL	DMKCKTUU	DMKCVTBD	DMKCVTDT	DMKDSPCH	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET
	DMKLOCRD	DMKLOCRQ	DMKPGTSG	DMKQCNPL	DMKRPAGT	DMKRPAPT	DMKRSPTR	DMKSTKCP	DMKTRUST	DMKVSDAD
	FFS	F0	F1	F4095	F4096	HALF1	HALF1ENT	HALF1SZE	HALF1VAL	INTBLK
	INTCODE	INTCTRL	INTLTH	INTMTYPE	INTTYPE	LOCK	OPNSFB	PLFLAG1	PLFRET	PLIMSG
	PLIST	PLNORET	PLR2	PLSIZE	PREFIXA	PREFIXB	PSA	RDRCHN	R0	R1
	R10	R11	R12	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SELDATA	SELDISP	SELECT	SELENTRY	SELFORW	SELLTH	SFBCLAS

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)								
	SFBCOPY	SFBDATE	SFBDIST	SFBDUMP	SFBFILID	SFBFLAG	SFBFNAME	SFBFTYPE	SFBLAST	SFBLOK
	SFBMISC1	SFBOFORM	SFBORIG	SFBRECNO	SFBRECSZ	SFBSIZE	SFBSTART	SFBTIME	SFBTYPE	SFBUFORM
	SFBUSER	SPFILID	SPLINK	SPNXTPAG	SPPREPAG	SPRMISC	SPSIZE	SYSTEM	TRACCURR	TRACEND
	TRACSTRT	TRAPCP	TRAPCR8	TRAPDATA	TRAPOK	TRAPTT	TRAPVT	TTCODE	TTENTRY	TTLENGTH
	TYP1403	VMBLOK	ZEROES							
DMKTRU	ADDSFB	ADSPCH	AFREE	AFREEP	AFRET	APTRAN	APTRLK	AQCNWT	ASYSVM	BFFCCPD
	BFFENTRY	BFFFSIZE	BFFREAL	BFFVIRT	BRING	CPEXADD	CPEXBLOK	CPEXRO	CPEXSIZE	C1
	DEFER	DELSFB	DMKCKSPL	DMKCKTSD	DMKCKTUU	DMKCVTDT	DMKDSPCH	DMKERMSG	DMKFREE	DMKFREEP
	DMKFRET	DMKLOCRD	DMKLOCRQ	DMKPGTSG	DMKPGUSD	DMKPGUVG	DMKPGUVR	DMKPTRAN	DMKPTRLK	DMKPTRUL
	DMKQCNWT	DMKRSPTR	DMKSTKCP	DMKSYSRM	DMKTRTCM	DMKTRXCP	DMKTRXEX	DMKTRXLK	DMKTRXTT	DMKTRXVT
	DMKVSGRI	F0	F1	F2	F4096	IOKEEP	LOCK	NORET	OPNSFB	PREFIXA
	PREFIXB	PSA	RDRCHN	R0	R1	R10	R11	R12	R13	R14
	R15	R2	R3	R5	R6	R7	R9	SAVEAREA	SAVEREGS	SAVEWRK1
	SFBCLAS	SFBCOPY	SFBDATE	SFBDIST	SFBDUMP	SFBFILID	SFBFLAG	SFBFNAME	SFBFTYPE	SFBLOK
	SFBMISC1	SFBOFORM	SFBORIG	SFBRECSZ	SFBSIZE	SFBSTART	SFBSYSID	SFBTIME	SFBTYPE	SFBUFORM
	SFBUSER	SPFILID	SPLINK	SPRECNUM	SPSIZE	SPTIME	SYSTEM	TRACCURR	TRAPCP	TRAPCR8
	TRAPDATA	TRAPOK	TRAPSTRT	TRAPTT	TRAPVT	TTLENGTH	TYP1403	VGPTR	VMBLOK	ZEROES
DMKTRX	ADSPCH	AFREE	AFREEP	AFRET	ALOKSY	AP	APSTAT1	APTRAN	APTRLK	APUOPER
	BRING	CPCREG0	CPEXADD	CPEXBLOK	CPEXREGS	CPEXRO	CPEXSIZE	C0	C1	DEFER
	DMKCVTBH	DMKDSPA	DMKDSPB	DMKDSPCH	DMKDSPRU	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKLOKDF
	DMKLOKSY	DMKPTRAN	DMKPTRUL	DMKSTKCP	DMKSTKDE	DMKSYSRM	FFS	F0	F1	F2
	F4095	F4096	HALF1SZE	INTTCP	INTTVG	INTTVT	LOCK	LOCKSAV	LPUADDR	MONCODE
	MP	PREFIXA	PSA	RUNCR0	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	TRAPDATA	VMBLOK	VMSIZE	XPAGNUM	ZEROES				
DMKTTX	AFREE	AFRET	AP	APTRAN	ASYSVM	BRING	BUFINLTH	BUFSIZE	CC	CD
	CONACTV	CONADDR	CONCCW1	CONCCW2	CONCCW3	CONCNT	CONCNTL	CONCNT2	CONCOMND	CONDATA
	CONESCP	CONFLAG	CONFLAGS	CONFLAG2	CONMORE	CONOUTPT	CONPARM	CONPFDEL	CONPFWRT	CONPNT
	CONRETN	CONSTAT	CONSYNC	CONTASK	CONTSIZE	CONTSKSZ	CONUSER	CONWRTRD	C1	DEFER
	DMKCVTBD	DMKFREE	DMKFRET	DMKPTRAN	DMKRETGT	DMKRETPT	DMKTBLCI	DMKTBLCO	DMKTBLPI	DMKTBLPO
	DMKTBMMI	DMKTBMMO	DMKTBMNI	DMKTBMNO	DMKTBNAE	DMKTBNAI	DMKTBNAO	DMKTBNBE	DMKTBNEA	DMKTBNEB
	DMKTTZLF	DMKVSTVP	F0	F1	F15	F2	F3	INHIBIT	IOBLOK	LOCK
	MP	PFDATA	PFDTEXT	PFKADDR	PFKFLAG	PFKIMM	PFKLNG	PFKRET	PFKTABLE	PRIORITY
	PSA	RDEVAPLI	RDEVAPLO	RDEVASTB	RDEVAVM1	RDEVBLOK	RDEVCON	RDEVCORD	RDEVNDLF	RDEVNOCR
	RDEVNOLF	RDEVPCHG	RDEVPSUP	RDEVSCRL	RDEVTAPL	RDEVTFLG	RDEVTMCD	RDEVTTYB	RDEVVM2I	RDEV3101
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER13	SAVER2
	SAVER3	SAVEWRK1	SAVEWRK2	SAVEWRK8	SILI	SKIP	SYSTEM	TRTABLE	VMBLOK	
DMKTTY	AFREE	ALARM	AP	BUFINLTH	BUFSIZE	CC	CD	CONADDR	CONADDR2	CONCCW1
	CONCCW2	CONCCW3	CONCNT	CONCNTL	CONCNT2	CONDATA	CONESCP	CONFLAG	CONFLAGS	CONFLAG2
	CONMORE	CONOUTPT	CONPARM	CONPNT	CONSPLT	CONSTAT	CONSYNC	CONTASK	CONTSIZE	CONTSKSZ
	CONUSER	CONWRTRD	DMKFREE	DMKTBLCO	DMKTBLPO	DMKTBMMO	DMKTBMNO	DMKTBNAO	DMKTBNEA	DMKTBNEB
	DMKTTXTR	DMKTTZLF	DMKVSTVP	F8	INHIBIT	IOBLOK	LOCK	MP	NOAUTO	PSA
	RDEVACTV	RDEVAPLO	RDEVATOF	RDEVBLOK	RDEVCON	RDEVCORD	RDEVFLAG	RDEVNDLF	RDEVNOCR	RDEVNOLF
	RDEVPCHG	RDEVPSUP	RDEVSCRL	RDEVTFLG	RDEVTMCD	RDEVTTYB	RDEV3101	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SAVEAREA	SAVEREGS	SAVEWRK1	SAVEWRK9	SILI	SKIP	TEMPSAVE
	VMBLOK									
DMKTTZ	AP	CC	CD	CONDATA	CONTASK	LOCK	MP	SILI	SKIP	VMBLOK

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
DMKUCB	AP	CC	LOCK	MP	SILI					
DMKUCC	AP	CC	LOCK	MP	SILI					
DMKUCS	CC	LOCK	SILI							
DMKUDR	ADSPCH	AFREE	AFRET	ALARM	APTRAN	ASYSLC	ASYSVM	BLANKS	BRING	CC
	CLASDASD	CLASFBA	C1	DEFER	DMKCFCCO	DMKDSPCH	DMKERMCP	DMKERMSG	DMKFREE	DMKFRET
	DMKIOSQR	DMKLOCKD	DMKLOCKQ	DMKPGUVG	DMKPGUVR	DMKPTRAN	DMKRPAGT	DMKSCNMU	DMKSCNVU	DMKSYSOC
	DMKSYSOW	DMKSYSPL	DMKSYSUD	DPLEND	DPLENDFG	DPLHIGH	DPLIST	DPLLOW	DPLNGTH	DPLSIZE
	DPLSIZ2	DPLVADD	F0	F4096	F8	IOBCAW	IOBCP	IOBFATAL	IOBFLAG	IOBIRA
	IOBLOK	IOBMISC	IOBMISC2	IOBSIZE	IOBSTAT	IOBUSER	IOERETN	LOCK	NORET	OPERATOR
	OWNDLIST	OWNDRDEV	OWNDVSER	PSA	RDEVBLOK	RDEVTYPC	R0	R1	R10	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	SAVEAREA	SAVEREGS	SAVER0	SAVER2	SAVEWRK1	SAVEWRK2	SAVEWRK4	SAVEWRK6
	SAVEWRK7	SAVEWRK9	SILI	SYSLOCS	SYSTEM	UIUCBLOK	UIUCCHN	UIUCDASD	UIUCDISP	UIUCLAST
	UIUCRES	UIUCSIZE	UIUCSTAT	UIUCUSER	VMBLOK	ZEROES				
DMKUDU	ACORETBL	AFREE	AFRET	APSTAT1	APTRAN	APUOPER	ASYSLC	ASYSVM	BLANKS	BRING
	BUFCNT	BUFFER	BUFNXT	BUFSIZE	CORSWPNT	CORTABLE	C1	DEFER	DMKCPVAC	DMKFREE
	DMKFRET	DMKLOCKD	DMKLOCKQ	DMKLOKSW	DMKPGUVG	DMKPGUVR	DMKPTRAN	DMKRPAGT	DMKRPAPT	DMKSCNAU
	DMKSYSPL	DMKSYSUD	DPLEND	DPLENDFG	DPLIST	DPLSIZE	DPLVADD	FFS	F0	F2
	F240	F3	F7	IOERETN	IOKEEP	LOCK	PSA	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER3	SAVEWRK2	SWPCYL	SWPFLAG	SWPRECMP
	SYSLOCS	SYSTEM	TIMEDISP	UCNTRL	UCNTRLSZ	UCURPASS	UDASDDEV	UDASDDIR	UDASDIPL	UDASDMAC
	UDEVAD	UDEVCNT	UDEVCODE	UDEVF	UDIRAD	UDIRF	UDISPDEV	UDISPMAC	UFLAGS	UIPARMS
	UIPARMSZ	UIPL	UIPLOP	UIPLPRMS	ULOCDVAD	UMACAD	UMACF	UMDISKAD	UMDISKMD	UMDISKMP
	UMDISKRP	UMDISKWP	UNEWPASS	UNOUPF	UOBJVMBK	UOP	UOPTIONS	UPRIOR	UPRIVLGE	URECMP
	URETCODE	URPAGDEV	URPAGDIR	URPAGMAC	URPAGXIP	USCRCPO	USCRINA	USCRINR	USCRSTA	USCRVMO
	USTORAGE	USVDASD	UTESTMD	UUSERID	UVMBLOK	UVPAGBUF	UVPAGDIR	UWORK	UWORK2	UXIPLAD
	VMBLOK	ZEROES								
DMKUNT	ACORETBL	ADSPCH	AFREEP	AFRET	AVMREAL	BALRSAVE	BALR14	CCC	CD	CDC
	CHC	CLASFBA	CORPGPNT	CORTABLE	CPEXBLOK	CPEXFPNT	DATACHK	DE	DMKDSPCH	DMKFREEP
	DMKFRET	DMKFRETL	DMKFRET1	DMKPTRPL	DMKPTRUL	DMKPTTFT	DMKSCNVD	DMKSTKCP	DMKSTKIO	DMKSYSRC
	DMKSYSRM	DMKSYSRV	DMKTRKFP	DMKVIOIN	D4	FFS	FTR35MB	F0	F1	F15
	F16	F240	F255	F4	F6	F7	F9	IDA	IFCC	IOBALTSK
	IOBCAW	IOBCC3	IOBCLN	IOBCSW	IOBFLAG	IOBIRA	IOBLINK	IOBLOK	IOBMISC	IOBRES
	IOBSIZE	IOBSPEC2	IOBSTAT	IOBUNREL	IOBUSER	IOBVADD	IOERBLOK	IOERCYLR	IOERDATA	IOERFLG2
	IOERLEN	LOCK	PRGC	PRTC	PSA	RCWADDR	RCWCCNT	RCWCCW	RCWCNT	RCWCOMND
	RCWCTL	RCWFLAG	RCWGEN	RCWHMR	RCWIO	RCWPNT	RCWRCNT	RCWSHR	RCWTASK	RCWVCAW
	RCW2311	RDCALTCL	RDCALTRK	RDCBLKAP	RDCBLKCE	RDCBLKCG	RDCBLKFA	RDCBLKMA	RDCBLOK	RDCDIAG
	RDCDIAGN	RDCDVCYL	RDCDVTRK	RDCPRIM	RDEVBLOK	RDEVFTR	RDEVMDL	RDEVRDC	R0	R1
	R10	R11	R12	R13	R14	R15	R2	R3	R4	R5
	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER11	SAVER8	SAVEWRK1	SAVEWRK2
	SAVEWRK3	SAVEWRK5	SAVEWRK6	SAVEWRK8	SAVEWRK9	SKIP	TYP2305	TYP3330	TYP3340	TYP3350
	TYP3370	TYP3375	TYP3380	UC	VMBLOK	XPAGNUM	X2048BND	ZEROES		
DMKURS	AFREE	AFRET	ALARM	AP	AQCNWT	BLANK	BLANKS	CL	CLASURI	CLASURO
	DMKCVTBD	DMKCVTBH	DMKFREE	DMKFRET	DMKQCNWT	DMKRSPDL	DMKSCNPH	DMKSCNRD	ERRMSG	FFS
	F1	IPUADDR	IPUADDRX	LOCK	LPUADDR	MP	NORET	NOTIME	OPERATOR	PSA
	RCHBLOK	RCHPROC	RCUBLOK	RCUCHA	RCUCHB	RCUPRIME	RCUSUB	RCUTYPE	RDEVBLOK	RDEVCFCB
	RDEVCLAS	RDEVCUA	RDEVDFCB	RDEVDRAN	RDEVEXTN	RDEVFLAG	RDEVFSEP	RDEVIMAG	RDEVLDBG	RDEVLDMD
	RDEVNRDY	RDEVOVLY	RDEVPRFG	RDEVPTHS	RDEVPURG	RDEVSEP	RDEVSPL	RDEVSTAT	RDEVSTA2	RDEVTYPC

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	RDEVTYPE	RDEVXSEP	RSPLCTL	RSPSFBLK	RSPXAUTO	RSPXBLOK	RSPXDST1	RSPXDST2	RSPXDST3	RSPXDST4
	RSPXFLAG	RSPXFORM	RSPXFPND	RSPXPMNT	RSPXRECT	RSPXSEQ	RSPXSETU	RSPXSFIL	R0	R1
	R10	R11	R12	R13	R14	R15	R2	R4	R5	R6
	R7	R8	R9	SAVEAREA	SAVEREGS	SAVEWRK3	SAVEWRK5	SAVEWRK6	SAVEWRK8	SFBCLAS
	SFBCOPY	SFBDIST	SFBFILID	SFBLOK	SFBPNT	SFBRECNO	SFBTYPE	SFBUSER	TYPPRT	TYPPUN
	TYPRDR	TYP3211	TYP3800	TYP38003	TYP38008	URSDEV	URSFILE	URSOPER	URSPATH	URSSTACK
	URSSTART	URSUSER	VMBLOK	VMOSTAT	VMSTKLST	VMSTKO	VMSYSOP	ZEROES		
DMKUSP	AFREE	AFRET	ALOCBLOK	ALOCCHN	ALOCPAGE	ALOCPREC	ALOCRECS	ALOKRM	ALOKSP	AP
	APSTAT1	APUOPER	ARSPPR	ARSPPU	DMKFREE	DMKFREHI	DMKFRELO	DMKFRENP	DMKFRET	DMKFRTTE
	DMKLOCKD	DMKLOCKQ	DMKPAGEX	DMKPAGSK	DMKPAGXS	DMKPAGXZ	DMKSYSPM	DMKVSPWA	IOBLOK	IOBMISC
	LOCK	LOCKSAV	LPUADDR	MP	PAGESIZE	PAGEXSIZ	PSA	RECBAK	RECBLOK	RECPNT
	RECSIZE	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVEWRK1
	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK6	SAVEWRK7	SFBFLAG	SFBINUSE	SFBLOK	SFBSIZE	SYSPALOC
	SYSPFPNT	SYSPLIST	VMBLOK							
DMKUSQ	ACIDEL	ACIFCN	ACIPARMS	ACIRGRP	ACIRUSR	ACISIZE	ADSPCH	AFREE	AFREEP	AFRET
	APSTAT1	APUOPER	AQCNWT	ASYSLC	ASYSOP	ASYSVM	BLANK	BLANKS	CLASGRAF	CLASSPEC
	CLASTERM	CPEXADD	CPEXBLOK	CPEXFPNT	CPEXR0	CPEXR8	CPEXSIZE	DELPAGES	DELSEGS	DMKACODS
	DMKACOFF	DMKACOTM	DMKBLDRL	DMKCFPRR	DMKCVTBD	DMKCVTBH	DMKCVTDT	DMKDIDLF	DMKDSPCH	DMKERMCP
	DMKFREE	DMKFREEP	DMKFRET	DMKHPTDI	DMKLOCKD	DMKLOCKQ	DMKLOKSW	DMKPENDA	DMKPGSPO	DMKPGSPP
	DMKPTRRC	DMKPTSPW	DMKQCNWT	DMKQCOCL	DMKRPIRA	DMKSCHDL	DMKSCHRT	DMKSCNAU	DMKSCNRD	DMKSCNRN
	DMKSCNVD	DMKSSSVM	DMKSSULO	DMKSSVUS	DMKSTKCP	DMKSYSDW	DMKSYSNM	DMKSYSTI	DMKTRCND	DMKVATBC
	DMKVCTLO	DMKVDREL	DMKVMASH	FFS	F0	F1	F15	F4	LASTUSER	LOCK
	LOGDROP	LOGHOLD	MICSIZE	MSSPRES	NORET	NOTIME	OPERATOR	PFKADDR	PFKCOUNT	PFKDWDS
	PFKSIZE	PFKTABLE	PFKTBLSZ	PRIORITY	PSA	PSAMSS	RDEVAIRA	RDEVBLOK	RDEVFCNS	RDEVLOGC
	RDEVLTRM	RDEVPRFG	RDEVPS	RDEVSNA	RDEVSNRB	RDEVSTA3	RDEVSTA6	RDEVTYPC	RDEVTYPE	RDEVUSER
	RUNUSER	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER9
	SAVEWRK1	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SNARBLOK	SNARLUN	SYSLOCS	TIMEDISP
	TRQBCHIO	TRQBFLG2	TRQBFPNT	TRQBLOK	TRQBSIZE	TYPBSC	TYP3705	VMBLOK	VMSIZE	ZEROES
DMKVAT	ADSPCH	AFREE	AFREEP	AFRET	ALOKSY	AMCHAREA	APSTAT1	APSTAT2	APTRAN	APUOPER
	ASYSVM	BALRSAVE	BALR12	BALR13	BALR14	BALR4	BALR5	BRING	CONTINUE	CPCREG0
	CPEXADD	CPEXBLOK	CPEXREGS	CPEXR0	CPEXR2	CPEXSIZE	CPPTLBR	CPSEGPRT	CPSPMODE	CPSTAT2
	CPSTAT4	CPXSTOR	CP370EON	C1	DEFER	DMKATSCF	DMKDSPCH	DMKDSPRU	DMKERMSG	DMKFREE
	DMKFREEP	DMKFRET	DMKLOKDF	DMKPRGSM	DMKPTRAN	DMKPTSAE	DMKPTSPW	DMKSTKCP	DMKSTKDE	EXTMODE
	F0	F1	F16	F2	F240	F255	F4096	F8	INTPR	LAP370E
	LOCK	LOCKSAV	LOKSAVE	LPUADDR	MCHAREA	MCHMODEL	MICBLOK	MICDASA	MICEVMA2	MICEVMA3
	MICFSSE	MICIPTP2	MICLCTL2	MICLRA2	MICPTLB2	MICSTBVR	MICSTSM2	MICVPFR2	MOD3081	NOPTLB
	OFF	ON	PAGE2K	PAGSWP	PAGTABLE	PGADDR	PGBLOK	PGBSIZE	PGPNT	PREFIXB
	PSA	PURGESTO	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SASPF	SAVEAREA
	SAVEREGS	SAVERETN	SAVER12	SAVER13	SEGPROT	SPMPFX	SSEB	STOACTV	STOBLKLN	STOBLOK
	STOFLAG	STOFSTUS	STOLAST	STONEXT	STONXTUS	STOPAGVR	STOSEGCT	STOSEGVR	STOSHCR1	STOSHLEN
	STOSHSEG	STOSIZE	STOUSPT	STOUSPTL	STOVCR1	STOVLEN	STOVPPG	STOVPSG	STO6CPG	STO6CSG
	SWPFLAG	SWPSHR	SWPTABLE	TEMPR3	TRANMODE	TREXADD	VFAULT	VMBLOK	VMSIZE	XKEYMODE
	XPAGNUM	ZEROES								
DMKVAU	AFREE	ALOKSY	AMCHAREA	AP	APSTAT1	APSTAT2	APTRAN	APUOPER	BALR4	BRING
	CPEXADD	CPEXBLOK	CPEXREGS	CPPTLBR	CPSEGPRT	CPSTAT2	CPSTAT4	CPXSTOR	CP370EON	C1
	DEFER	DMKDSPRU	DMKFREE	DMKLOKDF	DMKPRGSM	DMKPTRAN	DMKPTSAE	DMKSTKDE	DMKVATBC	DMKVATMD
	F0	F1	F2	F255	LOCK	LOCKSAV	LPUADDR	MCHAREA	MCHMODEL	MOD3081
	MP	PREFIXB	PSA	R0	R1	R10	R11	R12	R13	R14

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	R15	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA
	SAVEREGS	SAVER0	SAVER1	SAVER2	SAVER3	SAVEWRK1	SAVEWRK2	SEGPROT	STOACTV	STOBLOK
	STOFLAG	STOFSTUS	STONEXT	STONXTUS	STOSEGCT	STOSEGVR	STOSHCR1	STOUSPT	STOVLEN	TREXADD
	VMBLOK	XKEYMODE	XPAGNUM							
DMKVBM	AFREE	AP	APSTAT1	APTRAN	APTRLK	APUOPER	ASYSVM	BRING	C1	DEFER
	DMKBLDRT	DMKFREE	DMKPTRAN	DMKPTRUL	DMKSYSSZ	FFS	F0	F256	F4096	IOKEEP
	LOCK	MP	NEWPAGES	NEWSEGS	PREFIXA	PSA	PSALANG	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	SAVEAREA
	SAVEREGS	SAVER0	SAVER1	SAVER10	SAVEWRK2	SSBENTRL	SSBHEADL	SYSTEM	VMBLOK	VMSIZE
	WAIT									
DMKVCA	ADSPCH	AFREE	AFREEP	AFRET	AP	ATTN	BALRSAVE	BALR14	BALR15	BALR2
	BALR3	BALR9	BUSY	CC	CD	CE	CPEX	CPEXADD	CPEXBLOK	CPEXFPNT
	CPEXR0	CPEXR12	CPEXR13	CPEXSIZE	CPWAIT	DE	DMKDIBSM	DMKDSPCH	DMKFREE	DMKFREEP
	DMKFRET	DMKSCHDL	DMKSCNVU	DMKSTKCP	DMKSTKIO	DMKSYSRM	DMKVIOIN	FFS	FREESAVE	F1
	F2	F240	F7	IDA	IL	INTREQ	IOBCAW	IOBCC1	IOBCC3	IOBCSW
	IOBFLAG	IOBIOER	IOBIRA	IOBLINK	IOBLOK	IOBRES	IOBRSTRT	IOBSIZE	IOBSPEC	IOBSTAT
	IOBUNSL	IOBUSER	IOBVADD	IOERBLOK	IOERCCW	IOERCSW	IOERDATA	IOERLEN	IOERSIZE	LOCK
	MP	PCI	PCIF	PRGC	PRTC	PSA	RCWADDR	RCWCCW	RCWCNT	RCWCOMND
	RCWCTL	RCWFLAG	RCWINVL	RUNUSER	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	SAVERETN	SAVER0	SAVER10	SAVER11	SAVER12	SAVEWRK1	SAVEWRK2	SAVEWRK6
	SILI	SKIP	TEMPR0	TEMPR2	TEMPR3	TEMPR4	TEMPR5	TYPCTCA	UC	UE
	VMBLOK	ZEROES								
DMKVCB	ADSPCH	AFREE	AFREEP	AFRET	AP	APSTAT1	APUOPER	AQCNWT	ATTN	BLANKS
	CE	CONTINUE	CPEXADD	CPEXBLOK	CPEXFPNT	CPEXR0	CPEXSIZE	DE	DMKCVTBH	DMKDSPCH
	DMKFREE	DMKFREEP	DMKFRET	DMKLOKSW	DMKQCNWT	DMKSCNVU	DMKSTKCP	DMKSTKIO	DMKVIOIN	FFS
	F240	IDA	IOBCSW	IOBIRA	IOBLINK	IOBLOK	IOBSIZE	IOBUSER	IOBVADD	IOERBLOK
	IOERCCW	IOERCSW	IOERDATA	IOERLEN	IOERSIZE	LOCK	MP	NORET	PCI	PCIF
	PSA	RCWADDR	RCWCCW	RCWCNT	RCWCOMND	RCWCTL	RCWFLAG	RCWINVL	R0	R1
	R10	R11	R12	R13	R14	R15	R2	R3	R4	R5
	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER1	SAVER11	SAVER5	SAVEWRK1
	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK6	SAVEWRK9	SILI	TIMEDISP	UC	VMBLOK	ZEROES
DMKVCH	AFREE	AFRET	ALOKSP	AP	APSTAT1	APSTAT4	APTRAN	APTRLK	APUOPER	AQCNWT
	ARIOCH	ARIOCT	ARIOCU	ARIODV	ASYSOP	ASYSVM	BRING	CLASDASD	CLASFBA	CLASGRAF
	CLASSPEC	CLASTAPE	CLASTERM	CLASURI	CLASURO	CPSTAT5	C1	DEFER	DMKCVTBH	DMKERMSG
	DMKFREE	DMKFRERC	DMKFRET	DMKLOCKD	DMKLOCKQ	DMKLOKIO	DMKLOKSW	DMKPTRAN	DMKPTRUL	DMKQCNWT
	DMKSCNAU	DMKSCNEP	DMKSCNMU	DMKSCNRA	DMKSCNRU	DMKSCNVU	DMKVDREL	DMKVDSAT	DMKVDTPG	FFS
	F0	F1	F15	F2	F3	F4	LOCK	LOCKSAV	LPUADDR	LPUADDRX
	MP	MPUOPER	NORET	NOTERM	OFFLPROC	OPERATOR	POFFLINE	PREFIXB	PROCIO	PSA
	RCHBLOK	RCHCUTBL	RCHDED	RCHDISA	RCHPROC	RCHSTAT	RCUBLOK	RCUCHCNT	RCUDISA	RCUDVTBL
	RCUPRIME	RCUSTAT	RCUSUB	RCUTYPE	RDEVADD	RDEVALT	RDEVATT	RDEVBLOK	RDEVBUZY	RDEVCUA
	RDEVCUB	RDEVCUP	RDEVDED	RDEVDISA	RDEVDRAN	RDEVENAB	RDEVFLAG	RDEVOWN	RDEVPPAG	RDEVRCVY
	RDEVRSVD	RDEVSCHD	RDEVSPL	RDEVSTAT	RDEVSTA2	RDEVSTA4	RDEVSTA5	RDEVSTA6	RDEVSYS	RDEVTALY
	RDEVTYPC	RDEVTYPE	RDEVUSER	RDIDX	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	SAVER10	SAVER11	SAVER2	SAVEWRK1	SAVEWRK2	SAVEWRK4	SAVEWRK5	SAVEWRK6
	SAVEWRK8	SAVEWRK9	SYSTEM	TIMEDISP	TYP2305	TYP3705	VMBLOK	X40FFS	ZEROES	
DMKVCN	ACORETBL	ADSPCH	AFREE	AFREEP	AFRET	ALARM	ALTWRT	ANYWRITE	AP	APTRAN
	APTRLK	AQCNWT	ATTN	BLANKS	BRING	BUFCNT	BUFIN	BUFINLTH	BUFNXT	BUFSIZE
	BUSY	CC	CD	CE	CLASGRAF	CLASTERM	CMDREJ	CONDLN	CONDRFMT	CONTGMXB

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	CONTGMXD	CORCP	CORFLAG	CORTABLE	CSW	C1	DE	DEFER	DMKCFMAT	DMKCFMBK
	DMKCFMEN	DMKDSPCH	DMKFREE	DMKFREEP	DMKFRERC	DMKFRET	DMKPSACC	DMKPSAPO	DMKPSASC	DMKPTRAN
	DMKPTRFR	DMKPTRLK	DMKPTRUL	DMKPTTFT	DMKQCNPL	DMKQCNWT	DMKQCOPL	DMKSCNVU	DMKSTKIO	DMKTBLSF
	DMKTBLUP	DMKVIOIN	DMKVIOMK	DMKVMASH	EDIT	ERASWRT	FFS	FSREAD	FSWRITE	FTRAWSF
	FTRDIAL	F0	F1	F2	F255	F256	F3	F4	F4095	F4096
	F7	F8	IDA	IDAENTRY	IDAEWRD1	IDAEWRD2	IDAHCNT	IDAHCURR	IDAHEAD	IDAHPAGE
	IDAHSIZD	IDAHSTRT	IDAHWRK1	IDAHWRK2	IL	INHIBIT	INTREQ	IOBCSW	IOBIOER	IOBIRA
	IOBLINK	IOBLOK	IOBREMOT	IOBSIZE	IOBSPEC5	IOBUSER	IOBVADD	IOERBLOK	IOERCSW	IOERDATA
	IOERLEN	IOERSIZE	LOCK	MP	NICAPL	NICAWSF	NICBLOK	NICDTYPE	NICD3277	NICD3278
	NICD3284	NICHT	NICLGRP	NICRATTN	NICRDED	NICRFLG	NICSIZE	NICTEXT	NICTMCD	NICTYPE
	NICVDEVB	NICWTH	NIC3274	NOAUTO	NORET	NOTIME	ONEENT	PCI	PCIF	PLFLAG1
	PLIST	PLLED	PLR2	PRGC	PRIORITY	PRTC	PSA	RDEVAPLP	RDEVATT	RDEVBLOK
	RDEVFTR	RDEVFTR2	RDEVHT	RDEVNICL	RDEVNRDY	RDEVSNA	RDEVSNRB	RDEVSTAT	RDEVTEXT	RDEVTMCD
	RDEVTYPC	RDEVTYPE	RDEVWTH	READMOD	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SILI	SKIP	SNARBLOK	SNARFG2	SNARTTY	SNARVMB	TWOENTS	TYPBSC	TYPTTY	TYP3066
	TYP3277	TYP3278	TYP3284	UC	UE	VCONADDR	VCONBFSZ	VCONBUF	VCONCAW	VCONCCW
	VCONCCW2	VCONCNT	VCONCNT2	VCONCOMD	VCONCTL	VCONDWC	VCONEWA	VCONEWRT	VCONEXTN	VCONFLAG
	VCONFLG2	VCONFSOP	VCONFSS	VCONIDAP	VCONLED	VCONNCB	VCONNICB	VCONNTRM	VCONOPT	VCONPLF
	VCONPLST	VCONPLSZ	VCONPPA1	VCONRBUF	VCONRBYT	VCONRCNT	VCONRD	VCONRDEV	VCONRDSZ	VCONREMD
	VCONREMF	VCONREXW	VCONRFLD	VCONRIND	VCONRINX	VCONRLN	VCONRMAX	VCONRMCT	VCONRMOD	VCONSCRN
	VCONSKIP	VCONWBFS	VCONWBSZ	VCONWBUF	VCONWCNT	VCONWRRD	VCONWRRM	VCONWRT	VCONWSF	VCON3270
	VMBLOK	VMGENIO	WRITE	WRTREAD	WRTSFLD	XRIGHT16	ZEROES			
DMKVCP	ADSPCH	AFREE	AFREEP	AFRET	AP	APSTAT1	APUOPER	ATTN	BLANKS	BUFSIZE
	CPEXADD	CPEXBLOK	CPEXREGS	CPEXR0	CPEXR10	CPEXR11	CPEXR12	CPEXR13	CPEXSIZE	DMKBLDVM
	DMKCFMAT	DMKCFMBK	DMKDSPCH	DMKFREE	DMKFREEP	DMKFRET	DMKGRCQY	DMKIUACP	DMKLOKSW	DMKSTKCP
	DMKSTKIO	DMKVCQAT	DMKVCRNR	DMKVCUIL	DMKVCVCE	DMKVCVEB	DMKVCVKS	DMKVCVLD	DMKVCVLY	DMKVCXFU
	DMKVCXGF	DMKVCXIO	DMKVCXSA	DMKVIOIN	FTRAWSF	F0	F1	IOBCSW	IOBIRA	IOBLINK
	IOBLOK	IOBREMOT	IOBSIZE	IOBSPEC	IOBSPEC5	IOBUNSL	IOBUSER	IOBVADD	IXBLOK	IXSIZE
	LOCK	MP	PSA	RDEVADVF	RDEVASTB	RDEVATT	RDEVAVM2	RDEVBLOK	RDEVECOL	RDEVEHLT
	RDEVFTR2	RDEVHT	RDEVLLEN	RDEVLOG	RDEVMDL	RDEVPSS	RDEVPT	RDEVREAD	RDEVTFLG	RDEVTMCD
	RDEVTYPC	RDEVTYPE	RDEVUSC8	RDEVUSER	RDEVVM2	RDEVWTH	R0	R1	R10	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	SAVEAREA	SAVEREGS	SAVERETN	SAVER1	SAVER11	SAVESIZE	SAVEWRK1	SAVEWRK2
	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	SNARBACH	SNARBLOK	SNARCPFD	SNARCPT
	SNARDIAL	SNARDIS	SNARFG1	SNARFG2	SNARINN	SNAROUT	SNARPCT	SNARPVL	SNARSPT	SNARTTY
	TIMEDISP	TYPTTY	TYP3277	TYP3278	VMBLOK	VMCF	VMCFWAIT	VMDISC	VMKILL	VMLOGOFF
	VMOSTAT	VMPROT	VMRSTAT							
DMKVCQ	ADSPCH	AFREE	AFRET	AP	CONADDR	CONCNT	CONCNTL	CONDATA	CONOUTPT	CONRESP
	CONRETN	CONSTAT	CONTASK	CONTSIZE	CONTSKSZ	CONUSER	DMKDSPCH	DMKFREE	DMKFRET	DMKVCSWT
	DMKVCVCE	DMKVCVIN	DMKVCVIX	DMKVCVND	DMKVCVUT	DMKVCXFU	DMKVCXIO	DMKVCXSA	F0	F4
	F8	IXEXBLOK	IXSIZE	LOCK	MP	PSA	RDEVAPLI	RDEVAPLO	RDEVBLOK	RDEVCON
	RDEVSNRB	RDEVTAPL	RDEVTFLG	RDEVTMCD	RDEVTYPE	RDEVVM2I	R0	R1	R10	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	SAVEAREA	SAVEREGS	SAVERETN	SAVER0	SAVER2	SAVER7	SAVESIZE	SAVEWRK1
	SAVEWRK2	SAVEWRK4	SAVEWRK5	SNARBLOK	SNARDIAL	SNARDIS	SNAREXWT	SNARFG1	SNARFG2	SNARFG3
	SNARFORC	SNARINN	SNAROUT	SNARPCT	SNARPVL	SNARTTY	TYP3277	TYP3278	VCONBRK	VCONCTL
	VCONOBRK	VMBLOK	ZEROES							
DMKVCR	ADSPCH	AFREE	AFRET	AP	BUFCNT	BUFFER	BUFINLTH	BUFSIZE	CLASGRAF	CONACTV
	CONADDR	CONADDR2	CONADDR3	CONCCW2	CONCCW3	CONCNT	CONCNTL	CONCNT2	CONCNT3	CONDATA
	CONDIAG	CONDWC	CONESCP	CONFLAGS	CONFLG2	CONFSOP	CONFSS	CONNCB	CONOUTPT	CONPARM
	CONPFDEL	CONPFWRT	CONPNT	CONRD	CONRESP	CONRETN	CONRMOD	CONSPLT	CONSTAT	CONTASK

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	CONTSIZE	CONTSKSZ	CONUSER	DMKCFMBK	DMKCNTED	DMKDSPCH	DMKFREE	DMKFRET	DMKRETPT	DMKRGESK
	DMKSCHRT	DMKTTXIN	DMKTTYOP	DMKVCQAT	DMKVCQSR	DMKVCVCE	DMKVCVEB	DMKVCVER	DMKVCVIN	DMKVCVIX
	DMKVCVLD	DMKVCVND	DMKVCXFU	DMKVCXGF	DMKVCXIO	DMKVCXOR	DMKVCXSA	EDIT	F0	IDAENTRY
	IDAEWRD1	IDAEWRD2	IDAHEAD	IDAHSTRT	IDAHWRK1	IDAHWRK2	INHIBIT	IXEXBLOK	LOCK	MP
	PSA	RB	RDEVAIRA	RDEVAPLP	RDEVBLOK	RDEVCON	RDEVREAD	RDEVSNRB	RDEVTEXT	RDEVTFLG
	RDEVTMCD	RDEVTRQ	RDEVTYPC	RDEVTYPE	RM	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SAVERETN	SAVER0	SAVER2	SAVESIZE	SAVEWRK1	SAVEWRK2	SAVEWRK4
	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	SNARBACH	SNARBLOK	SNARCONQ	SNARDIAL	SNARDIS
	SNARFG1	SNARFG2	SNARFG3	SNARFORC	SNARFSS	SNARIF	SNARILER	SNARINN	SNARMDE	SNAROUT
	SNARPASS	SNARPCT	SNARPFIM	SNARPVL	SNARTTY	SNARVDEV	SNARVMB	TEMPR14	TEMPR7	TRAVSARP
	TYP3277	TYP3278	UCASE	VCONBRK	VCONCTL	VCONOBRK	VCONRINX	VCONRMCT	VMBLOK	VMGENIO
	XRIGHT16	XRIGHT24	ZEROES							
DMKVCS	ADSPCH	AFRET	ALARM	AP	APSTAT1	APUOPER	CLASGRAF	CONACTV	CONADDR	CONCNT
	CONDATA	CONDCLR	CONDCNCL	CONDIAG	CONDRFMT	CONDWC	CONEWA	CONEWRT	CONFLAGS	CONFLG2
	CONFSOP	CONFSS	CONLED	CONMORE	CONNCB	CONOUTPT	CONPARM	CONPARM2	CONPNT	CONPPA1
	CONRESP	CONRETN	CONSPLT	CONSTAT	CONSYNC	CONTASK	CONTSKSZ	CONWRT	CONWSF	DMKDSPCH
	DMKFRET	DMKLOKSW	DMKQCOET	DMKSCHRT	DMKTTXIN	DMKTTYOP	DMKVCQAT	DMKVCQRE	DMKVCQSR	DMKVCVCE
	DMKVCVEB	DMKVCVER	DMKVCVIX	DMKVCVLD	DMKVCVUT	DMKVCXFU	DMKVCXSA	F0	F1	IXEXBLOK
	LOCK	LOGDROP	LOGHOLD	MP	NCAUTO	PRIORITY	PSA	RDEVAIRA	RDEVBLOK	RDEVCON
	RDEVLLEN	RDEVSNRB	RDEVTFLG	RDEVTRQ	RDEVTYPC	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SAVERETN	SAVER11	SAVER2	SAVESIZE	SAVEWRK1	SAVEWRK2	SAVEWRK4
	SAVEWRK5	SAVEWRK6	SAVEWRK8	SAVEWRK9	SNARBACH	SNARBLOK	SNARBTCT	SNARCONQ	SNARDIAL	SNARDIPG
	SNARDIS	SNAREXWT	SNARFG1	SNARFG2	SNARFG3	SNARFSS	SNARINN	SNARMDE	SNAROUT	SNARPASS
	SNARPCT	SNARPRMT	SNARPVL	SNARTTY	SNARVMB	TIMEDISP	TRACFLG3	TRAVCSET	TRAVSARP	VCONANF2
	VCONCTL	VCONOPT	VCON3270	VMBLOK	VMDVSTRT	VMEXWAIT	VMGENIO	VMLOGOFF	VMLOGON	VMRSTAT
	VMVTERM	ZEROES								
DMKVCT	ADSPCH	AFREE	AFRET	AP	APSTAT1	APUOPER	ASYSLC	BLANKS	CRTEXT	DMKACOSA
	DMKDSPCH	DMKFREE	DMKFRET	DMKGRCSV	DMKIUACP	DMKLOKSW	DMKSCHRT	DMKSCNFD	DMKSYSLU	DMKUDRFU
	DMKVCQSR	DMKVCVCE	DMKVCVEB	DMKVCVIN	DMKVCVIX	DMKVCVLD	DMKVCVUT	DMKVCXD2	DMKVCXFU	DMKVCXGO
	DMKVCXOR	DMKVCXOX	F0	F8	IXEXBLOK	IXSIZE	LOCK	MP	PREFIXA	PSA
	RDEVAIRA	RDEVATOF	RDEVBLOK	RDEVFLAG	RDEVPSUP	RDEVQREP	RDEVSIZE	RDEVSNA	RDEVSNRB	RDEVTFLG
	RDEVTRQ	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER0
	SAVER11	SAVESIZE	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8
	SAVEWRK9	SNAENBLE	SNARACO	SNARBACH	SNARBLOK	SNARCPT	SNARDIAL	SNARDIS	SNARFG1	SNARFG2
	SNARFG3	SNARINN	SNARNXT	SNAROUT	SNARSIZ	SNARVMB	SNASTATS	SYSLOCS	TIMEDISP	TRQBSIZE
	VMBLOK	VSMPTR	ZEROES							
DMKVCU	ADSPCH	AFREE	AFREEP	AFRET	AP	APSTAT1	APUOPER	BLANKS	BUFCNT	BUFFER
	BUFINLTH	BUFSIZE	CONADDR	CONCNT	CONPARM	CONRETN	CONTASK	CONTSIZE	CPEXADD	CPEXBLOK
	CPEXREGS	CPEXR11	CPEXR12	CPEXSIZE	DMKCFMAT	DMKCFMBK	DMKCFMEN	DMKCNTED	DMKCVTBD	DMKDSPCH
	DMKFREE	DMKFREEP	DMKFRET	DMKLOKSW	DMKRETGT	DMKRETPT	DMKSTKCP	DMKTBLUP	DMKVCRNR	DMKVCVEB
	DMKVCVER	DMKVCVLD	DMKVCVLY	DMKVCXSA	DMKVSTVP	EDIT	F0	F3	F4	IXBLOK
	IXSIZE	LOCK	MP	PFDATA	PFDCMD	PFDCPYAD	PFDVAL	PFKADDR	PFKFLAG	PFKIMM
	PFKLNG	PFKRET	PFKTABLE	PFTAB	PFTABS	PSA	RDEVAPLP	RDEVBLOK	RDEVPTTC	RDEVREAD
	RDEVSNRB	RDEVTEXT	RDEVTFLG	RDEVTMCD	RDEVTYPE	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SAVER13	SAVER2	SAVEWRK1	SAVEWRK2	SAVEWRK5	SAVEWRK6	SAVEWRK7
	SAVEWRK9	SNARBACH	SNARBLOK	SNARCPFD	SNARDIAL	SNARFG1	SNARFG2	SNARFORC	SNARFSS	SNARINN
	SNARMDE	SNAROUT	SNARPFIM	SNARPFKI	SNARVDEV	TAB	TIMEDISP	TYP3277	TYP3278	UCASE
	VCONBRK	VCONCTL	VCONOBRK	VCONOPT	VCONRBUF	VCONRCNT	VCONRDEV	VCONRDSZ	VCON3270	VMBLOK

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	VMCF	VMCFWAIT	VMDVSTRT	VMGRFTAB	VMLOGON	VMMCPENV	VMMHLITE	VMMLEVEL	VMMLINED	VMMLVL2
	VMOSTAT	VMPA2APL	VMPFUNC	VMPROT	VMPXINT	VMQSTAT	VMRSTAT	VMTERM	VMVTERM	
DMKVCV	AFREE	AFRET	AP	APSTAT1	APUOPER	BUFSIZE	CC	CD	CODE	CONADDR
	CONCCW1	CONCNT	CONFLAG	CONFLAGS	CONMORE	CONOUTPT	CONPNT	CONRESP	CONSPLT	CONSTAT
	CONSYNC	CONTASK	CONTSKSZ	CRTEXT	CRTEXTSZ	DMKCVTAB	DMKFREE	DMKFRET	DMKIUACP	DMKQCOET
	DMKSCHRT	DMKSCHST	DMKVCTTM	DMKVCXFU	FFS	F0	F1	F2	IDAENTRY	IDAEWRD1
	IDAEWRD2	IDAHEAD	IDAHPAGE	IDAHSTRT	IXBLOK	IXEXBLOK	IXIRA	IXREGS	IXR12	IXR13
	IXSIZE	LOCK	MP	ONEENT	PREFIXA	PSA	RCMSGCT	RDEVAIRA	RDEVAPLP	RDEVBLOK
	RDEVCON	RDEVLLEN	RDEVSNRB	RDEVTEXT	RDEVTFLG	RDEVTMCD	RDEVTRQ	RDEVUSER	R0	R1
	R10	R11	R12	R13	R14	R15	R2	R3	R4	R5
	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER0	SAVER1	SAVER6	SAVER7
	SAVER9	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK7	SAVEWRK8	SAVEWRK9	SNARBLOK
	SNARCNTX	SNARCONQ	SNARCPT	SNARDIAL	SNARDIS	SNARFG1	SNARFG2	SNARINN	SNARLUN	SNARNOPR
	SNARNXT	SNAROUT	SNARSPT	SNARTTY	SNARVDEV	SNARVMB	TEMPR3	TRACCURR	TRACEND	TRACEVCS
	TRACFLG3	TRACHAR	TRACPROC	TRACPSAF	TRACSTRT	TRACSVCR	TRAEDCHR	TRAFUNCT	TRAINSTR	TRAIPRCD
	TRAIXBLK	TRALGAID	TRAMODE	TRAMSGLM	TRAREPLY	TRASEND1	TRASEND2	TRATIMER	TRATNTYP	TRAUDATA
	TRAUDIT1	TRAUDIT2	TRAUSER1	TRAVCSET	TRAVCSPA	TRAVMADR	TRAVSAMC	TRAVSAPA	TRAVSAQS	TRAVSARM
	TRAVSARP	TRAVSASV	TRCVCS	TRQBIRA	TRQBLOK	TRQBSIZE	TRQBUSER	TRQBVAL	VCONBUF	VCONCTL
	VCONNCB	VCONOPT	VCONPLF	VCONREMF	VCONSKIP	VCON3270	VMBLOK	VSMPTR	ZEROES	
DMKVCW	ADSPCH	AFREE	AFRET	AP	APSTAT1	APUOPER	ASYSLC	ASYSVM	CONPNT	CONTASK
	CPEXADD	CPEXBLOK	CPEXREGS	CPEXSIZE	DMKDSPCH	DMKFREE	DMKFRET	DMKGRCSV	DMKIUACP	DMKLOKSW
	DMKQCOCL	DMKQCPTO	DMKRGEIO	DMKSTKCP	DMKSYSLU	DMKVCVCE	DMKVCVKS	DMKVCVND	DMKVCXD2	DMKVCXFU
	DMKVCXGO	DMKVCXOX	F0	LOCK	MP	PREFIXA	PSA	RDEVBLOK	RDEVCON	RDEVPTTC
	RDEVQREP	RDEVSIZE	RDEVSNA	RDEVSNRB	RDEVTMCD	RDEVUSER	R0	R1	R10	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	SAVEAREA	SAVEREGS	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6
	SAVEWRK7	SAVEWRK8	SEVER	SNAENBLE	SNARACO	SNARBLOK	SNARCPT	SNARDIAL	SNARDIS	SNARFG1
	SNARFG2	SNARFG3	SNARINN	SNARLUN	SNARNXT	SNAROUT	SNARRSE	SNARSIZ	SNARSPN	SNARVMB
	SNASTATS	SYSLOCS	TIMEDISP	TRACCEPT	TRACFLG3	TRALUCON	TRASEVER	TRAVCSET	TRAVSAQS	TRAVSARM
	TRAVSASV	TRAVSMCN	VMBLOK	VMCFREAD	VMCOMND	VMLOGOFF	VMLOGON	VMQSTAT	VMRSTAT	VMUSER
	VSMPTR	ZEROES								
DMKVCX	ADSPCH	AFREE	AFREEP	AFRET	APSTAT1	APTRAN	APUOPER	ASYSVM	BOXBLOK	BOXINLNS
	BOXINWTH	BOXLINES	BOXLOGO	BOXWIDTH	BRING	CODE	CONPNT	CONTASK	CPEXADD	CPEXBLOK
	CPEXREGS	CPEXSIZE	C1	DEFER	DMKACOSA	DMKBOXMS	DMKBOXNS	DMKDSPCH	DMKFREE	DMKFREEP
	DMKFRET	DMKGRCSV	DMKIUACP	DMKPTRAN	DMKQCOCL	DMKQCPTO	DMKRGEIO	DMKSCHRT	DMKSTKCP	DMKVCQSR
	DMKVCRNR	DMKVCVCE	DMKVCVEB	DMKVCVIX	DMKVCVLD	F0	IXBLOK	IXIRA	IXREGS	IXR12
	IXSIZE	LOCK	PREFIXA	PSA	RCMSGCT	RDEVADVF	RDEVAIRA	RDEVBLOK	RDEVCON	RDEVECOL
	RDEVEHLT	RDEVQREP	RDEVSIZE	RDEVSNA	RDEVSNRB	RDEVTFLG	RDEVTRQ	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER1	SAVER9	SAVEWRK1	SAVEWRK4	SAVEWRK5
	SAVEWRK8	SAVEWRK9	SEVER	SNARBLOK	SNARCNTX	SNARCPT	SNARDIAL	SNARDIS	SNAREXWT	SNARFG1
	SNARFG2	SNARFG3	SNARFORC	SNARINN	SNARNXT	SNAROUT	SNARSIZ	SYSTEM	TEMPR2	TRACCURR
	TRACEND	TRACFLG3	TRACPROC	TRACSTRT	TRACSVCR	TRAVCSET	TRCVCS	VMBLOK	VSMPTR	XRIGHT16
DMKVDA	ADSPCH	AFREE	AFRET	ALOKSP	APSTAT1	APUOPER	AQCNWT	ARIODV	ASYSVM	BLANK
	BLANKS	CDDED	CDVADD	CLASDASD	CLASFBA	CLASGRAF	CLASSPEC	CLASTERM	CLASURI	CLASURO
	CPEXBLOK	CPEXR0	CPEXR1	CPEXSIZE	CPSTAT5	CRTEXT	CRTEXTSZ	DMKCVTAB	DMKCVTBH	DMKDSPCH
	DMKFREE	DMKFRET	DMKLOCKD	DMKLOCKQ	DMKLOKIO	DMKLOKSW	DMKMNLIN	DMKPMACD	DMKQCNWT	DMKSCHST
	DMKSCNAU	DMKSCNRN	DMKSCNRU	DMKSCNVS	DMKSCNVU	DMKSCONP	DMKSSSAS	DMKSSSMQ	DMKSSSVA	DMKSTKCP
	DMKSYSOC	DMKSYSOW	DMKVCHDC	DMKVDBMD	DMKVDCPS	DMKVDCSC	DMKVDEDC	DMKVDERR	DMKVDGAL	DMKVDSAT
	DMKVDTPG	F1	F5	F7	F8	IOBLOK	IOBMISC	IOBSIZE	LOCK	LOCKSAV
	LPUADDR	MSSPRES	NORET	OPERATOR	OWNDLIST	OWNDRDEV	OWNDVSER	PSA	PSAMSS	RANGE

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	RCHBLOK	RCHDED	RCHSTAT	RCUBLOK	RCUCHA	RCUPRIME	RCUSUB	RCUTYPE	RC100	RC104
	RDEVADD	RDEVBLOK	RDEVBUZY	RDEVCMDK	RDEVCUA	RDEVCUP	RDEVDED	RDEVDISA	RDEVDISB	RDEVDRAN
	RDEVENAB	RDEVFLAG	RDEVFTR	RDEVLOG	RDEVMOUT	RDEVNATH	RDEVOWN	RDEVPEND	RDEVPPAG	RDEVPRDV
	RDEVPREP	RDEVPS	RDEVSER	RDEVSPL	RDEVSTAT	RDEVSTA3	RDEVSTA4	RDEVSTA5	RDEVSTA6	RDEVSYS
	RDEVTFLG	RDEVTYPC	RDEVTYPE	RDEVUSER	RDEVVMNT	RDEV333V	RDIDX	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER0	SAVER11	SAVER2	SAVER4	SAVESIZE
	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	SYSVIRT	TIMEDISP
	TRQBIRA	TRQBLOK	TRQBSAVE	TRQBSIZE	TRQBTOD	TRQBUSER	TRQBVAL	TYPCTCA	TYPSDLC	TYPTTY
	TYP2305	TYP3066	TYP3277	TYP3278	TYP3284	TYP3330	VIRTUAL	VMBLOK		
DMKVDB	APSTAT1	APUOPER	AQCNWT	BLANK	DMKCVTBH	DMKLOKSW	DMKQCNWT	FFS	LOCK	NORET
	OPERATOR	PSA	RDEVBLOK	RDEVCUP	RDEVSTA5	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SAVER11	SAVER13	SAVEWRK1	SAVEWRK3	SAVEWRK5	SAVEWRK9	TIMEDISP
	VMBLOK									
DMKVDC	ADSPCH	AFREE	AFREEP	AFRET	APSTAT1	ASYSVM	BLANK	BLANKS	BUFFER	BUFNXT
	CLASDASD	CLASFBA	CLASGRAF	CLASSPEC	CLASTAPE	CLASTERM	CLASURI	CLASURO	CONTINUE	CPEXBLOK
	CPEXR0	CPEXSIZE	DMKCVTHB	DMKDSPCH	DMKFREE	DMKFREEP	DMKFRERC	DMKFRET	DMKIOSQR	DMKSCNAU
	DMKSCNFD	DMKSCNRU	DMKSCNVU	DMKVIOMK	FFS	FTRVIRT	F0	F1	F2	F3
	F4	F6	F7	F8	F9	IOBCC3	IOBCP	IOBCSW	IOBFLAG	IOBIOER
	IOBIRA	IOBLOK	IOBMISC2	IOBSIZE	IOBSPEC	IOBSTAT	IOBTIO	IOBUSER	IOERBLOK	IOERDATA
	IOEREXT	IOERSIZE	IPUADDR	IPUADDRX	LOCK	MPUOPER	PREFIXB	PROCIO	PSA	RANGE
	RC32	RC40	RDEVADD	RDEVBLOK	RDEVCUP	RDEVDED	RDEVDISA	RDEVDRAN	RDEVENAB	RDEVFLAG
	RDEVFTR	RDEVOWN	RDEVPPAG	RDEVRCVY	RDEVRSVD	RDEVSPL	RDEVSTAT	RDEVSTA5	RDEVSYS	RDEVTYPC
	RDEVTYPE	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER0
	SAVER10	SAVER13	SAVER2	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8
	SAVEWRK9	SCAN	TYP2305	TYP3705	UC	VMBLOK	ZEROES			
DMKVDD	AFREE	AFREEP	AFRET	ALOKSP	APSTAT1	APUOPER	AQCNWT	ASYSVM	BLANK	BLANKS
	BRING	CLASDASD	CLASFBA	CLASTAPE	CLASURI	CLASURO	DE	DFRET	DMKCVTBH	DMKDSBRD
	DMKFREE	DMKFREEP	DMKFRET	DMKLOCKD	DMKLOCKQ	DMKLOKIO	DMKLOKSW	DMKPGUSD	DMKPGUVG	DMKPGUVR
	DMKPTRUL	DMKQCNPL	DMKQCNWT	DMKRPAGT	DMKSCNAU	DMKSCNRD	DMKSCNRN	DMKSCNRU	DMKSCNVN	DMKSCNVU
	DMKSTKIO	DMKVCHDC	DMKVDCSC	DMKVDFER	DMKVDREL	FFS	F0	F20	F8	IOBCSW
	IOBIRA	IOBLINK	IOBLOK	IOBRADD	IOBSIZE	IOBSPEC	IOBUNSL	IOBUSER	LOCK	LOCKSAV
	LPUADDR	L4	NORET	OPERATOR	PLDFRET	PLFLAG1	PLFRET	PLIMSG	PLIST	PLNORET
	PLR2	PLSIZE	PSA	RANGE	RDEVADD	RDEVATT	RDEVBLOK	RDEVCUP	RDEVDED	RDEVDISA
	RDEVFLAG	RDEVMOUT	RDEVOWN	RDEVPPAG	RDEVPRDV	RDEVSER	RDEVSTAT	RDEVSTA5	RDEVSYS	RDEVTYPC
	RDEVUSER	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER11
	SAVER2	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	SPLINK
	SPNXTPAG	SYSTEM	TIMEDISP	TYPPRT	TYP2305	TYP8809	VMBLOK			
DMKVDE	ADSPCH	AFREE	AFREEP	AFRET	ALOKSP	APSTAT1	APUOPER	ASYSVM	BLANKS	CC
	CL	CLASDASD	CLASFBA	CLASTAPE	CLASTERM	CPEXBLOK	CPEXR0	CPEXR13	CPEXR5	CPEXSIZE
	DMKCVTBD	DMKCVTBH	DMKDSPCH	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKIOSQR	DMKLOCKD	DMKLOKIO
	DMKLOKSW	DMKSCNRD	DMKSCNRN	DMKSCNRU	FFS	FTRFH	FTRRPS	FTRVIRT	FTR35MB	FTR70MB
	F1	F2	F3	F4	F5	F6	F8	IOBCAW	IOBCC1	IOBCC3
	IOBCP	IOBCSW	IOBFATAL	IOBFLAG	IOBIOER	IOBIRA	IOBLOK	IOBMISC	IOBMISC2	IOBSIZE
	IOBSPEC	IOBSTAT	IOBTIO	IOBUSER	IOERBLOK	IOERDATA	IOEREXT	IOERSIZE	IOERSNSZ	LOCK
	LOCKSAV	LPUADDR	PSA	RANGE	RCUBLOK	RCUCACH	RCUPRIME	RCUSUB	RCUTYPE	RDEVBLOK
	RDEVCKDX	RDEVCMDK	RDEVCRDC	RDEVCUA	RDEVCUB	RDEVCUP	RDEVDISA	RDEVFTR	RDEVFTR2	RDEVLNKS
	RDEVMDL	RDEVMD02	RDEVMD13	RDEVNRDY	RDEVPPAG	RDEVPRDV	RDEVSER	RDEVSTAT	RDEVSTA3	RDEVSTA5

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)								
	RDEVSTA6	RDEVSTA7	RDEVTYPC	RDEVTYPE	RDEVUSER	RESET	R0	R1	R10	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	SAVEAREA	SAVEREGS	SAVER1	SAVER10	SAVER11	SAVER13	SAVER2	SAVER4
	SAVER6	SAVER7	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK9	SILI	SKIP
	TIMEDISP	TYPBSC	TYP2305	TYP3340	TYP3380	UC	VMBLOK	X40FFS	ZEROES	
DMKVDF	APSTAT1	APUOPER	BLANK	BLANKS	CLASDASD	CLASFBA	CONTINUE	DMKCVTBD	DMKCVTBH	DMKERMSG
	DMKLOCKD	DMKLOKSW	DMKSCNRN	DMKSCNRU	FFS	F2	F5	F6	F8	LOCK
	PSA	RANGE	RDEVADD	RDEVBLOK	RDEVCUP	RDEVLNKS	RDEVSTA5	RDEVTYPC	RDEVTYPE	RDEVUSER
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER1	SAVER13
	TIMEDISP	TYP2305	VMBLOK							
DMKVDG	AFREE	AFRET	ALOCBLOK	ALOCCHN	ALOCCSR	ALOCCYL1	ALOCCYL2	ALOCDOWN	ALOCDU	ALOCFLG
	ALOCMAP	ALOCMAX	ALOCNPAG	ALOCPAGE	ALOCPAVL	ALOCPG	ALOCPM	ALOCPMAX	ALOCPNT	ALOCPP
	ALOCPREC	ALOCPS	ALOCPUSE	ALOCRDEV	ALOCRECS	ALOCRSET	ALOCRWRT	ALOCSW	ALOCTDSK	ALOCTMSI
	ALOCUSED	ARIODV	CLASFBA	CPSTAT5	DMKDMPAL	DMKDMPDV	DMKDMPRC	DMKERMSG	DMKFREE	DMKFRET
	DMKPAGRD	DMKPGTAN	DMKPGTAW	DMKPGTA0	DMKPGTA4	DMKPGTA5	DMKPGTDK	DMKPGTDL	DMKPGTDM	DMKPGTPL
	DMKPGTPN	DMKPGTTM	DMKPGTTU	DMKPGT4A	DMKPGT5A	DMKPGT7A	DMKPGT8A	DMKPGT90	DMKSCNRU	DMKSYSDP
	DMKSYSOW	DMKSYSPE	DMKSYSPG	DMKSYSPM	DMKSYSSW	DMKSYSSZ	DMKVDHBB	DMKVDHFR	DMKVDHPG	FTR70MB
	F1	F10	F4096	F9	IOBLOK	IOBMISC	IOBSIZE	LOCK	OWNDLIST	OWNDRDEV
	PSA	RDCBLKFA	RDCBLKMA	RDCBLOK	RDCPAGAP	RDCRECSZ	RDEVADD	RDEVALLN	RDEVBLOK	RDEVCODE
	RDEVCUP	RDEVECKD	RDEVFIOB	RDEVFLAG	RDEVFTR	RDEVIOBL	RDEVLIOB	RDEVLOW	RDEVMDL	RDEVMD02
	RDEVOWN	RDEVPRDV	RDEVRDC	RDEVSER	RDEVSIZE	RDEVSTA5	RDEVTYPC	RDEVTYPE	RDEVUSER	RDIDX
	RECBAK	RECBLOK	RECCYL	RECPNT	RECUSED	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SAVER10	SAVER2	SAVER4	SAVER8	SAVEWRK1	SAVEWRK2	SAVEWRK3
	SAVEWRK4	SAVEWRK5	SYSPALOC	SYSPBPNT	SYSPCNT	SYSPCYL1	SYSPCYL2	SYSPFLG	SYSPFPNT	SYSPLIST
	SYSPVLEN	SYSPVOL	TEMPSAVE	TYP2305	TYP2314	TYP3330	TYP3340	TYP3350	TYP3370	TYP3375
	TYP3380	VMBLOK								
DMKVDH	AFREE	AFRET	ALOCASGN	ALOCBLOK	ALOCCHN	ALOCCYL1	ALOCCYL2	ALOCFLG	ALOCMAP	ALOCMAX
	ALOCNPAG	ALOCPAGE	ALOCPAVL	ALOCPMAX	ALOCPNT	ALOCPUSE	ALOCRCUU	ALOCRDEV	ALOCSW	ALOCTDK
	ALOCTDSK	ALOCUSED	APSTALOC	CLEARBIT	DMKCPXCK	DMKDMPAL	DMKFREE	DMKFRET	DMKPGTCP	DMKPGTGC
	DMKSYSSW	DMKSYSSZ	DMKSYSTD	FFS	F3	F4	LOCK	PSA	PSTRCPG$	RDEVBLOK
	RDEVFLAG	RDEVOWN	RDEVSER	RDEVSYS	RDEVTYPE	RECBAK	RECBLOK	RECCYL	RECMAP	RECMAX
	RECPNT	RECSIZE	RECUSED	R0	R1	R10	R11	R12	R13	R14
	R15	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA
	SAVEREGS	SAVEWRK1	SAVEWRK2	SYSPALOC	SYSPATYP	SYSPCNT	SYSPCYL1	SYSPCYL2	SYSPDASD	SYSPFLG2
	SYSPFPNT	SYSPLIST	SYSPUSER	SYSPVLEN	SYSPVLST	SYSPVOL	TYP3350	TYP3375	TYP3380	
DMKVDR	ACORETBL	ADSPCH	AFREE	AFREEP	AFRET	ALOKSP	APSTAT1	APTRAN	APTRLK	APUOPER
	AQCNWT	ASYSVM	BLANKS	BRING	CC	CLASDASD	CLASFBA	CLASGRAF	CLASSPEC	CLASTAPE
	CLASTERM	CLASURI	CLASURO	CPEXADD	CPEXBLOK	CPEXR0	CPEXR1	CPEXR8	CPEXSIZE	CRTEXT
	CRTEXTSZ	C1	DEFER	DMKACODV	DMKCFQRD	DMKCFREP	DMKCVTAB	DMKCVTBH	DMKDSPCH	DMKFREE
	DMKFREEP	DMKFRET	DMKIOSQR	DMKIOSRW	DMKLOKIO	DMKPTRAN	DMKPTRUL	DMKPTSPW	DMKPTTFT	DMKQCNWT
	DMKSCHST	DMKSCNRD	DMKSCNRN	DMKSCNVU	DMKSSSRL	DMKSSSVM	DMKSSULO	DMKSTKCP	DMKTDKRL	DMKVCBRS
	DMKVSUCO	DMKVSUCR	FFS	F0	F1	F255	IDAENTRY	IDAEWRD1	IDAHEAD	IDAHPAGE
	IDAHSIZD	IDAHSTRT	IDAHWRK1	IOBCAW	IOBFLAG	IOBIRA	IOBLDCCW	IOBLDRUN	IOBLDSZ	IOBLDTXT
	IOBLOK	IOBRELCU	IOBSIZE	IOBUSER	LOCK	LOCKSAV	LPUADDR	MSSPRES	NICADVF	NICBLOK
	NICDISB	NICENAB	NICFLAG	NICFMT	NICQRY	NICRATTD	NICRATTN	NICRDED	NICRFLG	NICROPER
	NICSIZE	NICTMAT	NICUSER	NICVDEVB	NORET	OPERATOR	PSA	PSAMSS	RCWCCNT	RCWCCW
	RCWHEAD	RCWTASK	RDEVADD	RDEVATT	RDEVBLOK	RDEVCURP	RDEVDED	RDEVDELP	RDEVFLAG	RDEVLNKS
	RDEVMOUT	RDEVNICL	RDEVOVLY	RDEVOWN	RDEVSTAT	RDEVSYS	RDEVTMAT	RDEVTYPC	RDEVTYPE	RDEVUSER
	RDEVVMNT	RDEV333V	R0	R1	R10	R11	R12	R13	R14	R15

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	R2	R3	R4	R5	R6	R7	R8	SAVEAREA	SAVEREGS	SAVER1
	SAVER9	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK6	SAVEWRK9	SILI	SYSTEM	TRQBIRA
	TRQBLOK	TRQBSAVE	TRQBSIZE	TRQBTOD	TRQBUSER	TRQBVAL	TWOENTS	TYPCTCA	TYP1052	TYP2305
	TYP3211	TYP3480	TYP3800	TYP38003	TYP38008	VCONBFSZ	VCONBUF	VCONCTL	VCONEXTN	VCONNCB
	VCONNICB	VCONPLF	VCONRBUF	VCONRDEV	VCONRDSZ	VCONREMD	VCONREMF	VCONRMSZ	VCONSIZE	VCONWBSZ
	VCONWBUF	VMBLOK	ZEROES							
DMKVDS	ADSPCH	AFREE	AFREEP	AFRET	ALOKSP	APSTAT1	APUOPER	BALR1	BALR2	CLASDASD
	CLASFBA	CLASGRAF	CLASSPEC	CLASTAPE	CLASTERM	CLASURI	CLASURO	CLEARBIT	CPEXADD	CPEXBLOK
	CPEXRO	CPEXR12	CPEXR8	CPEXSIZE	CRTEXT	CRTEXTSZ	DMKCVTAB	DMKCVTBH	DMKDSPCH	DMKERMSG
	DMKFREE	DMKFREEP	DMKFRERC	DMKFRET	DMKLOKIO	DMKNEADF	DMKSCHST	DMKSCNRD	DMKSCNRU	DMKSCNVU
	DMKSCOLI	DMKSPKDL	DMKSYSCK	DMKSYSCO	DMKSYSPR	DMKSYSPU	DMKSYSTD	DMKTDKGT	DMKTDKRL	DMKZTDDF
	FFS	FORMBLOK	FORMUSER	FTRRSRL	FTR3088	FTR3270	FTR4WCGM	F0	F16	F240
	F255	F8	LOCK	LOCKSAV	LPUADDR	NICBLOK	NICDTYPE	NICD3277	NICD3278	NICSIZE
	PSA	RCUBLOK	RCUSHRD	RCUTYPE	RC32	RC40	RDEVATT	RDEVBLOK	RDEVCTL	RDEVDED
	RDEVDELP	RDEVDISA	RDEVDRAN	RDEVENAB	RDEVEPLN	RDEVFLAG	RDEVFTR	RDEVLNCP	RDEVLNKS	RDEVMAX
	RDEVMOUT	RDEVNICL	RDEVNRDY	RDEVOWN	RDEVPEND	RDEVPS	RDEVRCVY	RDEVRSVD	RDEVSNA	RDEVSNRB
	RDEVSPL	RDEVSTAT	RDEVSTA3	RDEVSYS	RDEVTFLG	RDEVTMAT	RDEVTYPC	RDEVTYPE	RDEVUSER	RESET
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER1	SAVER10
	SAVER13	SAVER2	SAVER8	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7
	SAVEWRK8	SAVEWRK9	SNARBLOK	SNARVMB	TRQBCRT	TRQBIRA	TRQBLOK	TRQBSIZE	TRQBUSER	TRQBVAL
	TYPBSC	TYPCTCA	TYPFBA	TYPPUN	TYP1052	TYP2305	TYP2311	TYP3210	TYP3277	TYP3278
	TYP3704	TYP3705	TYP3800	TYP38003	TYP38008	VCONBRK	VCONCTL	VCONNICB	VCONOBRK	VCONOPT
	VCONRDEV	VCONREMD	VCONREMF	VCONSIZE	VCON3270	VMBLOK	X2048BND	ZEROES		
DMKVDT	ADSPCH	AFREE	AFRET	ALOKSP	APSTAT1	APUOPER	AQCNWT	ARIODV	BLANK	BLANKS
	CC	CLASDASD	CLASFBA	CPSTAT5	DMKCVTAB	DMKCVTBH	DMKDSPCH	DMKFREE	DMKFRERC	DMKFRET
	DMKIOSQR	DMKLOKIO	DMKLOKSW	DMKQCNWT	DMKSCNEP	DMKSCNRA	DMKSCNRD	DMKSCNRU	DMKSCNVU	DMKSYSCK
	FFS	F15	F16	F240	F7	IOBCAW	IOBCP	IOBFLAG	IOBIRA	IOBLINK
	IOBLOK	IOBPATHF	IOBRADD	IOBSIZE	IOBSTAT	IOBUSER	IPUADDR	LOCK	LOCKSAV	LPUADDR
	NOADD	NORET	OPERATOR	PSA	RCHBLOK	RCUBLOK	RCUDVTBL	RCUPRIME	RCUSIZE	RCUSUB
	RCUTYPE	RC32	RDEVADD	RDEVATT	RDEVBLOK	RDEVCU2	RDEVDED	RDEVDISA	RDEVFLAG	RDEVFTR3
	RDEVMOUT	RDEVOWN	RDEVPEND	RDEVPPAG	RDEVPRDV	RDEVSER	RDEVSTAT	RDEVSTA5	RDEVSYS	RDEVTMAT
	RDEVTYPC	RDEVUSER	RDIDX	R0	R1	R10	R11	R12	R13	R14
	R15	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA
	SAVEREGS	SAVER1	SAVER11	SAVER2	SAVER3	SAVER5	SAVER8	SAVEWRK1	SAVEWRK2	SAVEWRK3
	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	SILI	TIMEDISP	VMBLOK	
DMKVER	ADSPCH	AFREE	AFRET	ALARM	AP	APSTAT1	APTRAN	AQCNWT	BRING	CLASDASD
	CLASFBA	CLASSPEC	CONTINUE	CPSTAT5	CPUID	C1	DDRCUA1	DDRCUA2	DDRKEYN	DDRREC
	DDRSIZE	DEFER	DMKCVTBH	DMKCVTHB	DMKFREE	DMKFRET	DMKIOEVR	DMKPSAFP	DMKPTRAN	DMKQCNWT
	DMKSCNRD	DMKSCNVU	DMKVAURN	EXTMODE	FTR2311B	FTR2311T	FTR35MB	F0	F1	F15
	F24	F4	F4095	F7	F8	G16CHAN	LOCK	MDRCUA1	MDRKEYN	MDRREC
	MDRSENS	MDRSIZE	MDRSIZE1	MDRSWS3	MDRVOL	MIHCUA1	MIHKEYN	MIHREC	MIHSIZE	MIHVOL
	MP	NORET	OBRCCHS	OBRCPIDN	OBRCUA	OBRCUAIN	OBRCUAPR	OBRDEMNT	OBRDEVTN	OBRFBSNS
	OBRHAN	OBRHSIZE	OBRKEYN	OBRLSIZE	OBRLSKN	OBRPGMN	OBRRECN	OBRSENSN	OBRSSIZE	OBRSWSN
	OBRVOLN	OBR2SIZE	OBR3SIZE	OBR33SNS	OPERATOR	PREFIXB	PROCIPL	PSA	RDEVBLOK	RDEVDED
	RDEVFTR	RDEVMDL	RDEVSER	RDEVSTAT	RDEVTYPC	RDEVTYPE	R0	R1	R10	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	SAVEAREA	SAVEREGS	SAVERETN	SAVER12	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4
	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	TRANMODE	TYPCTCA	TYP2305	TYP2314	TYP3310
	TYP3330	TYP3340	TYP3350	TYP3375	TYP3380	VMBLOK				
DMKVFC	ACORETBL	ADSPCH	AFREE	AFREEP	AFRET	AP	APSTAT1	APUOPER	AQCNWT	ASYSVM

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	CORCP	CORFLAG	CORTABLE	CORVM	CPCREG0	CPEXADD	CPEXBLOK	CPEXREGS	CPEXSIZE	CPSTAT6
	C0	DMKCVTBD	DMKCVTBH	DMKDSPCH	DMKDSPNP	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKMHCCP
	DMKPTRFR	DMKPTTFT	DMKQCNWT	DMKSCNFD	DMKSTKOP	DMKVFRCH	DMKVFRIA	DMKVFRIP	DMKVFRNA	DMKVFRNP
	DMKVFRSV	FFS	F1	F24	F4096	F5	HCBLOK	HCCVU	HCDVU	HCMSFCW
	HCMSFDB	HCSIZE	IPUADDR	IPUADDRX	IREALVF	LOCK	MP	NORET	PREFIXB	PROCIPL
	PSA	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER1
	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SCCBLEN	SCCBLOK	SCCBRESP
	SCCB0020	SCCB0120	VECAVAIL	VECF	VECINST	VECOPVF	VECSAOK	VECSTAT	VECUSER	VECVARY
	VMBLOK	ZEROES								
DMKVFD	ACORETBL	ADSPCH	AFREE	AFREEP	AFRET	ALOKRM	ALOKSP	AP	APSTAT1	APUOPER
	AQCNWT	ASYSVM	CORCP	CORFLAG	CORFPNT	CORTABLE	CPEXADD	CPEXBLOK	CPEXR0	CPEXSIZE
	DMKCVTBH	DMKCVTDB	DMKCVTHB	DMKCVUFP	DMKDSPCH	DMKDSPNP	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET
	DMKPTRFR	DMKPTSAD	DMKPTTFT	DMKPTTPM	DMKQCNWT	DMKSCNFD	DMKSTKOP	DMKVFRSS	DMKVFRSV	DMKVFSOS
	F24	F3	F4096	LOCK	LOCKSAV	LPUADDR	MP	NOINVPTE	NORET	PREFIXB
	PSA	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	SAVEAREA	SAVEREGS	SAVEWRK1	SAVEWRK3
	SAVEWRK4	SAVEWRK5	SAVEWRK7	VECAVAIL	VECOPVF	VECSAOK	VECSTAT	VECUSER	VMBLOK	X40FFS
	ZEROES									
DMKVFE	ACORETBL	ADSPCH	AFREE	AFREEP	AFRET	ALOKRM	ALOKSP	AP	APSTAT1	APUOPER
	AQCNWT	ASYSVM	BUFIN	BUFNXT	CORCP	CORFLAG	CORFPNT	CORTABLE	CPEXADD	CPEXBLOK
	CPEXR0	CPEXSIZE	DMKCVTBH	DMKCVTDB	DMKCVTHB	DMKDSPCH	DMKDSPNP	DMKERMSG	DMKFREE	DMKFREEP
	DMKFRET	DMKPTRFR	DMKPTSAD	DMKPTTFT	DMKPTTPM	DMKQCNWT	DMKSCNFD	DMKSTKOP	DMKVFRSS	DMKVFRSV
	DMKVFSOS	F16	F24	F3	F4096	F8	LOCK	LOCKSAV	LPUADDR	MP
	NOINVPTE	NORET	PREFIXB	PSA	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	SAVER0	SAVER1	SAVER10	SAVEWRK1	SAVEWRK2	SAVEWRK4	SAVEWRK5	SAVEWRK6
	SAVEWRK8	VECAVAIL	VECOPVF	VECSAOK	VECSTAT	VECUSER	VMBLOK	X40FFS	ZEROES	
DMKVFR	ADSPCH	ALOKSY	AP	APSTAT1	APSTAT4	APUOPER	ASYSVM	BALRSAVE	BALR15	CPCREG0
	CPEXADD	CPEXBLOK	CPEXREGS	CPMCHSE	C0	C1	DMKDSPA	DMKDSPCH	DMKDSPRU	DMKLOKDF
	DMKPRGSM	DMKSCHDL	DMKSTKDE	DMKVFSOS	F1	F4095	LOCK	LPUADDR	LPUADDRX	MP
	PAGCORE	PREFIXB	PROCIPL	PSA	RUNCR0	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R9
	SAVEAREA	SAVEREGS	SAVEWRK1	SAVEWRK2	SAVEWRK8	SEGINV	TEMPR0	TEMPR4	TEMPR5	TEMPR6
	TEMPR8	TEMPR9	TRANMODE	VACOVFR	VECAVAIL	VECF	VECOPVF	VECSAOK	VECSTAT	VECUSER
	VMBLOK	ZEROES								
DMKVFS	ACORETBL	ADSPCH	AFREE	AFREEP	ALOKRM	ALOKSP	APSTAT1	APUOPER	ASYSVM	CLEAR
	CORFLAG	CORIOLCK	CORLCNT	CORPGPNT	CORTABLE	CORVM	CPEXADD	CPEXBLOK	CPEXREGS	CPEXSIZE
	DMKDSPCH	DMKFREE	DMKFREEP	DMKPTREP	DMKPTSAD	DMKPTSAE	DMKPTTCL	DMKPTTFT	DMKSTKOP	DMKVFRCH
	DMKVFRSA	DMKVFRSS	DMKVFRSV	DMKVFRVL	F4096	LOCK	LOCKSAV	LPUADDR	PREFIXB	PSA
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVEWRK1	VACOVFR	VECUSER
	VMBLOK									
DMKVIO	ADSPCH	AFREEP	AFRET	APTRAN	ATTN	AVMREAL	BALRSAVE	BRING	BUSY	CCC
	CDC	CE	CLASDASD	CLASFBA	CLASGRAF	CLASSPEC	CLASTERM	CLASURI	CLASURO	CPEXADD
	CPEXBLOK	CPEXFPNT	CPEXR1	CPEXR11	CPEXR12	CPEXR13	CPEXR4	CPEXSIZE	CPSPMODE	CPSTAT2
	CSW	CUE	C1	DE	DEFER	DFRCC1	DFRCC3	DMKCCHRF	DMKDSPCH	DMKFREEP
	DMKFRET	DMKHPTDI	DMKHVCAL	DMKHVCVI	DMKIOSCB	DMKPSAPO	DMKPTRAN	DMKPTRUL	DMKSCNVD	DMKSCNVU
	DMKSTKCP	DMKSTKIO	DMKTRCSW	DMKTRDSI	DMKTRDWT	DMKTRKFP	DMKUNTFR	DMKUNTRN	DMKVDRES	FFS
	F0	F16	F3	F4095	F4096	F8	IFCC	IL	INTREQ	IOBALTSK

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	IOBCAW	IOBCC2	IOBCC3	IOBCSW	IOBCUBSY	IOBCUE	IOBDIAG	IOBFATAL	IOBFLAG	IOBFPNT
	IOBHIO	IOBIOER	IOBLINK	IOBLOK	IOBMISC	IOBMISC2	IOBPST	IOBRELCU	IOBREMOT	IOBRES
	IOBRESRV	IOBSENSE	IOBSIOF	IOBSIZE	IOBSPEC	IOBSPEC2	IOBSPEC3	IOBSPEC4	IOBSPEC5	IOBSTAT
	IOBTIO	IOBUNREL	IOBUNSL	IOBUSER	IOBVADD	IOBVDVIO	IOBVHIO	IOBWRAP	IOERBLOK	IOERCSW
	IOERDATA	IOERECSW	IOEREXT	IOERSIZE	LOCK	PCI	PCIF	PREFIXA	PSA	RCWCCNT
	RCWCCW	RCWHEAD	RCWPNT	RCWRCNT	RCWTASK	RCWVCAW	RDEVBLOK	RDEVPS	RDEVSTA3	R0
	R1	R10	R11	R12	R14	R15	R2	R3	R4	R5
	R6	R7	R8	R9	SM	SPMPFX	TEMPSAVE	TREXCSW	TREXCTL2	TREXT
	TYPCTCA	TYP3340	TYP3704	TYP3705	UC	UE	VMBLOK	XPAGNUM	XTNDLOCK	ZEROES
DMKVMA	ACORETBL	ADSPCH	AFREE	AFREEP	AFRET	ALOKSY	APSTAT1	APSTAT2	APUOPER	ASYSVM
	BALRSAVE	BALR0	BALR2	CORCFLCK	CORFLAG	CORFLUSH	CORFREE	CORIOLCK	CORPGPNT	CORSWPNT
	CORTABLE	CORVM	CPEXADD	CPEXBLOK	CPEXREGS	CPEXSIZE	CPPTLBR	CPSTAT4	CPXSTOR	DMKCFMBK
	DMKCFMRU	DMKCVTBH	DMKDSPCH	DMKDSPNP	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKLOKDF	DMKLOKSY
	DMKPTRAQ	DMKPTRSC	DMKPTRUL	DMKPTSAD	DMKPTTFT	DMKREIPL	DMKSTKCP	DMKSTKMP	FFS	F1
	F16	F256	F4095	F8	LOCK	LOCKSAV	LPUADDR	PAGACT	PAGCORE	PAGINVAL
	PAGTABLE	PROCIPL	PROTBEG	PROTBLOK	PROTERR	PROTFLAG	PROTPALT	PROTPGAD	PROTRCNT	PROTREI
	PROTREIL	PROTSYSN	PSA	R0	R1	R10	R11	R12	R13	R14
	R15	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA
	SEGINV	SEGPAGE	SEGPLEN	SEGTABLE	SHRNAME	SHRSEGCT	SHRSEGNM	SHRTABLE	SWPFLAG	SWPTABLE
	SWPTRANS	SWPVPAGE	SWTHSAVE	TEMPR0	TEMPR2	TEMPR5	TEMPR6	VMABLOK	VMAFPNT	VMASHRBK
	VMBLOK									
DMKVMC	ACORETBL	ADSPCH	AFREE	AFREEP	AFRET	APSTAT1	APTRAN	APTRLK	APUOPER	ASYSVM
	BRING	CORCP	CORFLAG	CORTABLE	CPEXADD	CPEXBLOK	CPEXR0	CPEXR11	CPEXSIZE	C1
	DEFER	DMKCVTAB	DMKDSPCH	DMKFREE	DMKFREEP	DMKFRET	DMKLOKSW	DMKPSAFC	DMKPSAFP	DMKPSASC
	DMKPSASP	DMKPTRAN	DMKPTRLK	DMKPTRUL	DMKPTSPW	DMKSCHDL	DMKSCHST	DMKSCNAU	DMKSTKCP	DMKSTKMP
	DMKSTKOP	F0	F1	F7	LOCK	LPUADDR	PSA	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER11	SAVER2	SAVER5	SAVER6	SAVEWRK1
	SAVEWRK2	SAVEWRK5	SAVEWRK6	SAVEWRK7	SYSTEMID	TIMEDISP	TRQBIRA	TRQBLOK	TRQBTOD	TRQBUSER
	TRQBVAL	TRQREGSD	TRQREGO	VMBLOK	XPAGNUM	ZEROES				
DMKVMD	AFREE	AFRET	APTRAN	APTRLK	BLANK	BLANKS	BRING	BUFCNT	BUFFER	BUFNXT
	COLON	C1	DEFER	DMKCVTBH	DMKCVTHB	DMKERMSG	DMKFREE	DMKFRET	DMKPSAFP	DMKPTRAN
	DMKPTRUL	DMKSCNFD	DMKSYSDU	DMKUDRFU	DMKVMEDP	DMPIPCS	DUMP	ERRCODE	FFS	F0
	F1	F10	F2	F3	F4	F4096	F5	F6	F8	HEADER
	IOERETN	IOKEEP	LOCK	PSA	RESET	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7
	SAVEWRK8	SAVEWRK9	SFBDATE	SFBFLAG4	SFBFNAME	SFBFTYPE	SFBLOK	SFBNORET	SFBOFORM	SFBSIZE
	SFBTIME	SFBUFORM	SFBUSER	SYSSIZE	VMBLOK	VMSIZE	XPAGNUM	X40FFS		
DMKVME	ADDSFB	AFREE	AFRET	APTRAN	APTRLK	AQCNWT	ASYSVM	BRING	CPSTAT4	CPUID
	C1	DATE	DEFER	DMKCKSPL	DMKCKTSD	DMKCKTUU	DMKCPEID	DMKCVTDT	DMKERMSG	DMKFREE
	DMKFRET	DMKLOCRD	DMKLOCRQ	DMKPGTSG	DMKPGUVG	DMKPGUVR	DMKPSAFP	DMKPSAPO	DMKPTRAN	DMKPTRLK
	DMKPTRUL	DMKQCNWT	DMKRPAGT	DMKRPAPT	DMKRSPDP	DMKSPKDL	DMKVSDAD	DMKVSGRI	DMPCKCOM	DMPCPUID
	DMPCRS	DMPDMPID	DMPFPRS	DMPGPRS	DMPINREC	DMPIPCS	DMPPGMAP	DMPPSW	DMPSYSRM	DMPSYSRV
	DMPTODCK	DMPVMTYP	F0	F1	F4095	F4096	F5	F6	IOERETN	IOKEEP
	LOCK	MODESET	NORET	PREFIXB	PSA	RDRCHN	R0	R1	R10	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	SAVEAREA	SAVEREGS	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6
	SAVEWRK7	SAVEWRK8	SAVEWRK9	SFBCLAS	SFBCOPY	SFBDATE	SFBDIST	SFBDUMP	SFBFILID	SFBFLAG
	SFBFTYPE	SFBLAST	SFBLOK	SFBMISC1	SFBORIG	SFBPNT	SFBRECNO	SFBRECSZ	SFBSIZE	SFBSTART
	SFBSYSID	SFBTIME	SFBTYPE	SFBUSER	SPLINK	SPMPFX	SPNXTPAG	SPPREPAG	SPRECNUM	SPSIZE

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	SWPFLAG	SWPKEY1	SWPKEY2	SYSTEM	TODATE	TYP1403	VMBLOK	VMSIZE	XKEYMODE	XPAGNUM
	ZEROES									
DMKVMG	AFREE	AFRET	DMKFREE	DMKFRET	DMKIUACP	F1	LOCK	PREFIXA	PSA	R0
	R1	R11	R12	R13	R14	R15	R2	R3	R4	R5
	R6	R7	R9	SAVEAREA	SAVEREGS	SEVER	UFLAGS	VGPTR	VMBLOK	ZEROES
DMKVMI	ATTN	BUSY	CAW	CC	CD	CE	CLASDASD	CLASFBA	CLASSPEC	CLASTAPE
	CLASURI	CSW	DE	EXTMODE	FLAG2	IL	INTTIO	IPLADDR	IPLCCW1	IPLPSW
	LOCK	PSA	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SENSE	SILI
	SKIP	SM	TAB	TYPCTCA	TYPRDR	TYPUNSUP	TYP2401	TYP2415	TYP2420	TYP2501
	TYP2540R	UC	UE	VMBLOK						
DMKVRR	ACORETBL	AFREE	AFREEP	AFRET	APSTAT1	APUOPER	ARIOCT	ASYSVM	AVMREAL	BALRSAVE
	CAW	CLASDASD	CLASFBA	CORCP	CORDISA	CORFLAG	CORTABLE	CORVM	CPSPMODE	CPSTAT2
	CSW	DMKBLDVM	DMKCPIBD	DMKCVTAB	DMKDMPCP	DMKDMPC2	DMKDMPRY	DMKDSPQI	DMKDSPVM	DMKERMSG
	DMKFREE	DMKFREEA	DMKFREEP	DMKFRET	DMKLOGOP	DMKPMAIO	DMKPMATM	DMKPTRFR	DMKQCNWT	DMKQVMTS
	DMKSAV	DMKSCHDL	DMKSCHRT	DMKSCHST	DMKSCNVU	DMKSTABD	DMKSTKIO	DMKSYSND	DMKVATOF	DMKVDSAT
	DMKVDTPG	FF	FFS	F1	F4	F4096	INTTIO	IOBBPNT	IOBCSW	IOBFPNT
	IOBLINK	IOBLOK	IOBPMINT	IOBRADD	IOBSIZE	IOBSPEC4	IOERBLOK	IOERDATA	IOERSIZE	IOMASK
	LOCK	LPUADDR	MICBLOK	MICDASA	MICEVMA2	MICEVMA3	MICFSSE	MICPMAMP	MICPMAV	MICPMMSK
	MICPMPSA	MICSTBVR	MVSA370E	NORET	OPERATOR	PMAAVAIL	PMAMODE	PMAON	PMASTAT	PREFIXA
	PREFIXB	PRNPSW	PROCIO	PROCIPL	PSA	RCUBLOK	RCUCACH	RCUDISA	RCUOWNER	RCUPRIME
	RCUSTAT	RCUSUB	RCUTYPE	RDEVALT	RDEVBLOK	RDEVCUA	RDEVCUB	RDEVCUP	RDEVDED	RDEVDISA
	RDEVENAB	RDEVFLAG	RDEVFTR	RDEVPTHS	RDEVPTH1	RDEVPTH2	RDEVPTH3	RDEVPTH4	RDEVPTH5	RDEVPTH6
	RDEVPTH7	RDEVPTH8	RDEVSER	RDEVSTAT	RDEVSTA2	RDEVSTA5	RDEVSTA6	RDEVTYPC	RDEVUSER	RDEV333V
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	SAVEAREA	SAVEREGS	SENSE	SIGSENSE	SIGSTART
	SILI	SPMPFX	TEMPSAVE	TRQBLOK	TRQBQUE	TRQBTOD	TRQBVAL	TYP3330	UC	VMBLOK
	X40FFS									
DMKVRS	ACORETBL	AP	ARIOCU	ARIODC	ARIODV	ARIOUC	AVMREAL	BALRSAVE	CAW	CE
	CORDISA	CORFLAG	CORTABLE	CPCREG0	CSW	C0	C1	C2	DE	DMKDMPAA
	DMKDMPC2	DMKDMPMP	DMKDMPSA	DMKDSPIS	DMKDSPQI	DMKPMARW	DMKPMATM	DMKSAV	DMKSCNRU	DMKSCNVD
	DMKVRROP	FFS	F1	F4096	INTTIO	IOBCSW	IOBFPNT	IOBLOK	IOBPMINT	IOBRADD
	IOBSPEC4	IOMASK	IONPSW	IOOPSW	LOCK	MP	PRNPSW	PSA	RCUBLOK	RCUCACH
	RCUOWNER	RCUSIZE	RCUTYPE	RDEVATT	RDEVBLOK	RDEVDED	RDEVFLAG	RDEVFTR	RDEVPTHS	RDEVSER
	RDEVSIZE	RDEVSTAT	RDEVSTA2	RDEVSTA6	RDEVUSER	RSRTNPSW	R0	R1	R10	R11
	R12	R14	R15	R2	R3	R4	R5	R7	R8	SENSE
	SIGREST	SILI	TREXFLAG	TREXIN1	TREXIN2	TREXSVC1	TREXSVC2	TREXT	TREXVAT	TRQBLOK
	TRQBQUE	UC	UE	VMBLOK	XRIGHT16	ZEROES				
DMKVSC	AFRET	AP	BALRSAVE	BALR8	CC	CD	CLASDASD	CLASFBA	CLASTAPE	CLASTERM
	CPEXBLOK	CPEXFPNT	DMKFRET	DMKSTKCP	FTR35MB	FTR70MB	F1	F10	F15	F4
	F4096	F8	IDA	IOBFLAG	IOBFPNT	IOBLOK	IOBRELCU	IOBSIZE	IOERBLOK	IOEREXT
	IOERSIZE	LOCK	L2	MP	PSA	RCUBLOK	RCUOWNER	RCUPRIME	RCUSUB	RCUTYPE
	RDCBLKMX	RDCBLOK	RDEVALT	RDEVBLOK	RDEVCMDK	RDEVCUA	RDEVFTR	RDEVMDL	RDEVMD02	RDEVRDC
	RDEVSTA2	RDEVSTA6	RDEVTYPC	R0	R1	R10	R11	R12	R13	R14
	R15	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA
	SKIP	SYSVIRT	TEMPR15	TEMPSAVE	TYP2314	TYP3330	TYP3340	TYP3350	TYP3375	TYP3380
	VIRTUAL	VMBLOK	XRIGHT16							
DMKVSD	ACORETBL	ADDSFB	AFREE	AFRET	AP	APSTAT1	APTRLK	APUOPER	ARSPRD	ASYSVM
	BLANKS	CHGSFB	CORCP	CORFLAG	CORTABLE	CORVM	DMKCKSPL	DMKDSPNP	DMKFREE	DMKFRET

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	DMKLOCRD	DMKLOCRQ	DMKPTRFR	DMKPTRLK	DMKPTRUL	DMKRSPCR	DMKSCNAU	DMKSYSHE	DMKSYSHT	DMKSYSWI
	DMKVSETR	FFS	F1	LOCK	MP	NOCOPY	PSA	RDRCHN	RHTBLOK	RHTECNT
	RHTINDEX	RHTLEN	RHTNUMEN	RHTPAGNO	RHTRSRV	RHTSFB1	RHTSPU1	R0	R1	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7
	R8	SAVEAREA	SAVEREGS	SAVER0	SAVER1	SAVER2	SAVER7	SAVEWRK2	SAVEWRK4	SAVEWRK6
	SAVEWRK7	SAVEWRK8	SFBFILID	SFBFLAG4	SFBINVS	SFBLOK	SFBPNT	SFBSIZE	SFBSYSID	SFBUSER
	SPUBLOK	SPUFIRST	SPULAST	SPURCNT	SPUUSER	TEMPSAVE	VMBLOK	ZEROES		
DMKVSE	ADSPCH	AFREEP	AP	APTRAN	APTRLK	BRING	CPEXADD	CPEXBLOK	CPEXREGS	CPEXSIZE
	C1	DEFER	DMKCKSCV	DMKCKTSD	DMKCVTBH	DMKDSPCH	DMKFREEP	DMKLOCRD	DMKLOCRQ	DMKMCHST
	DMKPTRAN	DMKPTRLK	DMKPTRPS	DMKPTRUL	DMKRPASV	DMKRSPPR	DMKRSPPU	DMKRSPSC	DMKSCNMU	DMKSCNRD
	DMKSTKCP	DMKSYSHE	DMKSYSHT	DMKSYSOW	DMKSYSSP	DMKSYSWI	DMKVSGUM	F0	IOERETN	LOCK
	MP	OWNDLIST	OWNDRDEV	PSA	RDRCHN	RHTBLOK	RHTFLAG	RHTINDEX	RHTLEN	RHTNUMEN
	RHTPAGNO	RHTRSRV	RHTSFB1	RHTSPRF	RHTSPU1	RHTSPU3	R0	R1	R10	R11
	R12	R13	R14	R15	R2	R3	R4	R5	R6	R7
	R8	R9	SAVEAREA	SAVEREGS	SAVER1	SAVER2	SAVEWRK1	SAVEWRK2	SAVEWRK7	SAVEWRK8
	SFBFLAG4	SFBINVS	SFBLOK	SFBPNT	SFBSIZEB	SFBSYSID	SFBUSER	SPECIALV	SPUBLOK	SPUFIRST
	SPULAST	SPUNEXT	SPUSIZEB	SWPCHG1	SWPCODE	SWPFLAG	SWPPSTOR	SWPRECMP	VMBLOK	
DMKVSF	AP	DMKRSPSC	DMKRSPSM	DMKSYSWI	DMKVSDFH	DMKVSETR	F4095	F4096	LOCK	MP
	PSA	RHTBLOK	RHTVIRT	R0	R1	R10	R11	R12	R13	R14
	R15	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA
	SAVEREGS	SAVER0	SAVER2	SAVEWRK2	SFBFILID	SFBLOK	SFBORIG	SFBPNT	SFBSIZEB	SPUBLOK
	SPUFIRST	SPURCNT								
DMKVSG	ADSPCH	AFREE	AFREEP	AFRET	AP	APSTAT1	APTRAN	APTRLK	APUOPER	BRING
	CPEXADD	CPEXBLOK	CPEXREGS	CPEXSIZE	C1	DEFER	DMKCKTSD	DMKCKTSU	DMKCVTBH	DMKDSPCH
	DMKERMSG	DMKFREE	DMKFREEP	DMKFRET	DMKMCHST	DMKPTRAN	DMKPTRLK	DMKPTRPS	DMKPTRUL	DMKRSPFC
	DMKRSPPR	DMKRSPPU	DMKSCNAU	DMKSCNMU	DMKSCNRD	DMKSTKCP	DMKSYSOW	DMKSYSSP	DMKSYSWI	DMKVSDFH
	DMKVSEER	DMKVSELK	DMKVSETR	FFS	F0	IOERETN	LOCK	MP	OWNDLIST	OWNDRDEV
	PSA	RHTBLOK	RHTFLAG	RHTRSRV	RHTSPRF	RHTSPU1	R0	R1	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SAVER0	SAVER7	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK5	SAVEWRK6
	SAVEWRK7	SAVEWRK8	SAVEWRK9	SFBFILID	SFBLOK	SFBPNT	SFBTYPE	SFBUSER	SPECIALV	SPUBLOK
	SPUFIRST	SPUIND	SPULASGN	SPULAST	SPUMAP	SPUMAPSZ	SPUNEXT	SPURCNT	SPUSIZEB	SPUSRTID
	SPUSYSID	SPUUSER	SWPCODE	SWPFLAG	SWPPSTOR	TEMPR4	VMBLOK			
DMKVSI	ACRLOCK	ADSPCH	AFREE	AFREEP	AFRET	AP	APTRAN	ATTN	AVMREAL	BLKMPX
	BRING	BUSY	CAW	CC	CC3	CD	CE	CLASDASD	CLASFBA	CLASGRAF
	CLASSPEC	CLASTAPE	CLASTERM	CLASURI	CLASURO	CONTINUE	CPSPMODE	CPSTAT2	CUE	C1
	DE	DEFER	DMKACRCT	DMKCCWTR	DMKDSPCH	DMKFREE	DMKFREEP	DMKFRET	DMKHPSQV	DMKIOSQV
	DMKPMACR	DMKPSAPO	DMKPTRAN	DMKSCHDL	DMKSCNVU	DMKSTKCP	DMKSTKIO	DMKTRDSI	DMKUNTFR	DMKVCAST
	DMKVCBTS	DMKVCNEX	DMKVIOC1	DMKVIOIN	DMKVIOMK	DMKVSCVR	DMKVSJEX	DMKVSPEX	DMKVSWTO	FFS
	F0	F1	F2	F240	F4095	F8	INTREQ	IOBCAW	IOBCC3	IOBCLN
	IOBCSW	IOBFLAG	IOBIRA	IOBLINK	IOBLOK	IOBMISC	IOBPVM	IOBRCAW	IOBRELCU	IOBSENSE
	IOBSIOF	IOBSIZE	IOBSPEC	IOBSPEC2	IOBSPEC3	IOBSPEC4	IOBSPM	IOBSTAT	IOBTIO	IOBUSER
	IOBVADD	IOBVCUE	IOERBLOK	IOERCSW	IOERDATA	IOERLEN	IOERSIZE	LOCK	L2	MP
	PCI	PSA	RDEVBLOK	RDEVPS	RDEVSTA3	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SM	TEMPR5	TREXCCWI	TREXCTL2	TREXT	TYPCTCA	TYPRDR	TYP3210	TYP3480
	TYP3705	UC	VMBLOK	ZEROES						
DMKVSJ	ADSPCH	AFREEP	ALOKSP	ALOKSY	AP	APSTAT1	APTRAN	APUOPER	AVMREAL	BRING
	BUSY	CAW	CCC	CC3	CDC	CE	CLASDASD	CLASSPEC	CLASTERM	CLASURI
	CLASURO	CODE	CPEXADD	CPEXBLOK	CPEXMISC	CPEXREGS	CPEXSIZE	CPSPMODE	CPSTAT2	CSW

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	CUE	C1	DE	DEFER	DMKCCHRF	DMKDSPA	DMKDSPCH	DMKDSPRU	DMKFREEP	DMKHPSHT
	DMKIOSCB	DMKIOSQV	DMKLOKDF	DMKLOKIO	DMKLOKSY	DMKPSAPO	DMKPTRAN	DMKSCNRU	DMKSSSMQ	DMKSTKDE
	DMKSTKIO	DMKTRDSI	DMKVCBSH	DMKVIOCL	DMKVIOIN	DMKVIOMK	DMKVIOXK	DMKVSICH	DMKVSIEX	DMKVSIFT
	F0	IFCC	IOBCSW	IOBCUBSY	IOBFPNT	IOBHIO	IOBIRA	IOBLOK	IOBPATHF	IOBPROC
	IOBRADD	IOBRCAW	IOBSIZE	IOBSPEC	IOBSPEC3	IOBSPEC4	IOBSTAT	IOBUSER	IOBVADD	IOBVCUE
	IOBVHIO	LOCK	LOCKSAV	LPUADDR	MP	MSSPRES	PREFIXA	PREFIXB	PSA	PSADSPRQ
	PSAMSS	RCUBLOK	RCUCUBSY	RDEVAIOB	RDEVBLOK	RDEVPROC	RDEVPS	RDEVSTA3	R0	R1
	R10	R11	R12	R13	R14	R15	R2	R3	R4	R5
	R6	R7	R8	R9	SM	TEMPSAVE	TRACBEF	TRACCURR	TRACEND	TRACFLG2
	TRACPROC	TRACSTRT	TRACSVCR	TRACOD	TRCCLCH	TRCCSW	TRCHALT	TREXCCWI	TREXCTL2	TREXT
	TYPCTCA	UE	VMBLOK	XCDISP	XCPEND	ZEROES				
DMKVSP	ADSPCH	AFREE	AFREEP	AFRET	AP	APTRAN	ASYSVM	BRING	CC	CD
	CE	CLASURI	CLASURO	CMDREJ	CPEXADD	CPEXBLOK	CPEXFPNT	CPEXR11	CPEXR8	CPEXSIZE
	CSW	C1	DATACHK	DE	DEFER	DMKCVTBH	DMKDSPCH	DMKERMSG	DMKFREE	DMKFREEP
	DMKFRET	DMKPGTSG	DMKPGUSD	DMKPGUVR	DMKPSAPO	DMKPTRAN	DMKPTRUL	DMKRPAGT	DMKRPAPT	DMKSCNVU
	DMKSPKDL	DMKSPLCV	DMKSTKCP	DMKVIOMK	DMKVSQPD	DMKVSRGC	DMKVSRMD	DMKVSTOP	DMKVSVLD	DMKVSWDC
	DMKVSWFC	DMKVSWOR	DMKVSXCL	DMKVSXSE	DMKVSXSI	DMKVSXSR	DMKVSXTR	DMKXADVS	FFS	F0
	F1	F15	F2	F256	F3	F4	F4096	F5	F6	F7
	F8	IDA	IL	INTREQ	IOERETN	LOCK	MP	PCI	PCIF	PRGC
	PRTC	PSA	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS
	SAVER7	SAVER8	SAVER9	SAVEWRK2	SAVEWRK6	SFBCLAS	SFBFCB	SFBFILID	SFBFLAG	SFBFLAG2
	SFBFLAG3	SFBFLNMT	SFBHOLD	SFBLAST	SFBLDBEG	SFBLDMID	SFBLOK	SFBPNT	SFBPURGE	SFBRECER
	SFBRECNO	SFBRECSZ	SFBSIZE	SFBSTART	SFBSYSID	SFBTIME	SFBTYPE	SILI	SKIP	SPFILID
	SPLINK	SPNXTPAG	SPPREPAG	SPRMISC	SPTIME	SYSTEM	TEMPR3	TYPPRT	TYPPUN	TYPTIMER
	TYP3203	TYP3210	TYP3211	TYP3262	TYP3289E	TYP3800	TYP38003	TYP4245	TYP4248	TYP5ACCW
	UC	UE	VBFBLOK	VBFBUF	VBFCCW1	VBFCOUNT	VBFRADD	VMBLOK	VSPBIGBF	VSPBUFBK
	VSPBUFSZ	VSPCAW	VSPCCW	VSPDCFOP	VSPDPAGE	VSPERR	VSPFLAG1	VSPIDACT	VSPIDASW	VSPLCTL
	VSPMISC	VSPSFBLK	VSPSIZE	VSPVPAGE	VSPVPG2	VSP5ACCW	X40FFS	ZEROES		
DMKVSQ	ADSPCH	AFREE	AFREEP	AFRET	AP	APTRAN	APTRLK	ASYSVM	BLANKS	BRING
	CC	CD	CPEXADD	CPEXBLOK	CPEXR11	CPEXR12	CPEXSIZE	C1	DEFER	DMKFREE
	DMKFREEP	DMKFRET	DMKPGTSG	DMKPGUSD	DMKPGUVG	DMKPGUVR	DMKPTRAN	DMKPTRUL	DMKRPAGT	DMKRPAPT
	DMKSPKDL	DMKSPLCV	DMKSTKCP	DMKSYSFL	DMKSYSSF	FFS	FORMBLOK	FORMFLAG	FORMNARR	FORMNTRY
	FORMSEND	FORMUSER	F0	F1	F24	F255	F256	F3	F4	F4095
	F4096	IOERETN	LOCK	MP	PSA	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SAVERTN	SAVER2	SAVER7	SAVER8	SAVER9	SAVEWRK9	SFBCLAS
	SFBFCB	SFBFCBNL	SFBFCBXL	SFBFLAG2	SFBFLAG3	SFBFLAG4	SFBFLNMT	SFBLAST	SFBLOK	SFBMISC1
	SFBPNT	SFBPURGE	SFBRECNO	SFBSTART	SFBSYSID	SFBTIME	SFBTYPE	SFBVLEN	SFPBLOK	SFPBOT
	SFPCLASS	SFPEND	SFPLEN	SFPOPTS	SFPTITLE	SFPTOP	SILI	SKIP	SPFILID	SPLINK
	SPNXTPAG	SPPREPAG	SPRECNUM	SPRMISC	SPSIZE	SPTIME	SYSTEM	TEMPR1	TEMPR15	TYPPRT
	TYP3210	TYP3211	TYP3800	TYP38003	TYP4248	VBFBLOK	VBFCCW1	VBFDATLF	VBFDCACT	VBFDCUSD
	VBFFLAG1	VBFLGLFT	VBFRADD1	VBFRADD2	VBFTIC	VBFVSQR0	VBFVSQR3	VMBLOK	VSPBIGBF	VSPBUFBK
	VSPBUFSZ	VSPCCW	VSPDPAGE	VSPERR	VSPFLAG1	VSPLCTL	VSPMISC	VSPMISC2	VSPNEXT	VSPSFBLK
	VSPSIZE	VSPVPAGE	VSPVPG2	VSP5ACCW						
DMKVSR	AP	APTRAN	APTRLK	ASYSVM	AVMREAL	CLASURI	DMKPGUVG	DMKPSACC	DMKPSASC	DMKPTRAN
	DMKVMASH	F0	F255	F256	F4	F4095	F4096	MP	PSA	SAVEAREA
	TEMPR0	TEMPR1	TEMPR12	TEMPR4	TYP3203	TYP3211	TYP3800	TYP38003	TYP4248	VBFBLOK
	VBFBUF	VBFBUF1	VBFBUF2	VBFCOUNT	VBFDATA	VBFRADD	VBFRADD1	VBFRADD2	VBFRADD8	VBFVSQR0
	VBFVSQR1	VBFVSQR2	VBFWORK	VMBLOK	VMSIZE	VSPBIGBF	VSPBUFBK	VSPCAW	VSPCCW	VSPFLAG1
	VSPIDACT	VSPIDAL	VSPIDASW	VSPIDAW2	VSPLCTL	VSP5ACCW	XPAGNUM	ZEROES		

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
DMKVST	ADDSFB	ADSPCH	AFREE	AFREEP	AFRET	AP	CC	CLASURI	CLASURO	CPEXADD
	CPEXBLOK	CPEXFPNT	CPEXR1	CPEXR11	CPEXR8	CPEXSIZE	DMKCKSPL	DMKCVTBH	DMKDSPCH	DMKERMSG
	DMKFREE	DMKFREEP	DMKFRET	DMKPGUVR	DMKPTRUL	DMKRPAGT	DMKSCNVD	DMKSPKDL	DMKSPLOV	DMKSTKCP
	DMKVSPWA	DMKVSQPD	FFS	F1	F2	F3	F4	F4096	F5	F6
	F60	F7	LOCK	MP	OPNSFB	PRTCHN	PSA	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER0	SAVER1	SAVER2	SAVER7	SAVER8
	SAVER9	SAVEWRK2	SFBFILID	SFBLOK	SFBRECNO	SFBRECSZ	SFBSTART	SILI	SKIP	SPLINK
	SYSTEM	TYPPRT	TYP1052	TYP3210	TYP3211	TYP3800	TYP38003	TYP4248	VMBLOK	VSPBUFBK
	VSPBUFSZ	VSPCCW	VSPLCTL	VSPMISC	VSPSFBLK	VSPSIZE	VSPVPAGE	X40FFS	ZEROES	
DMKVSU	AP	DMKVSPPE	DMKVSPST	DMKVSPUS	DMKVSWDC	DMKVSWFC	LOCK	MP	PSA	R0
	R11	R12	R13	R15	R6	R7	R8	R9	SAVEAREA	SAVEREGS
	SAVER1	SAVER2	SAVEWRK1	SFBFLAG	SFBLOK	SFBRECER	TYP3210	VMBLOK	VSPLCTL	VSPSFBLK
DMKVSV	AP	DMKVSP	FFS	F1	F2	F256	F3	F4	F4095	F4096
	LOCK	MP	PSA	R0	R1	R10	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R8	R9	TYP38003	VBFBLOK	VBFCCW1
	VBFDATA	VBFRADD1	VBFRADD2	VBFTIC	VSPBUFBK	VSPCCW	VSPLCTL	XPAGNUM	ZEROES	
DMKVSW	AFREE	AFRET	AP	DMKFREE	DMKFRET	DMKLOCRD	DMKLOCRQ	DMKPGUVG	DMKPGUVR	DMKPTRUL
	DMKRPAGT	DMKSPKDL	DMKVSDAD	DMKVSDDL	DMKVSFNU	DMKVSXRD	F3	LOCK	MP	PSA
	R0	R1	R11	R12	R13	R14	R15	R2	R5	R6
	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER6	SAVER7	SAVER8	SAVER9	SFBCLAS
	SFBDUMP	SFBEOF	SFBFILID	SFBFLAG	SFBFLAG2	SFBHOLD	SFBINUSE	SFBLOK	SFBMON	SFBNOHLD
	SFBOPEN	SFBRECER	SFBRECNO	SFBSIZE	SFBSTART	SFBSYSID	SFBUHOLD	SFBUSER	SPLINK	SYSTEM
	VMBLOK	VSPBUFBK	VSPBUFSZ	VSPCAW	VSPLCTL	VSPMISC	VSPSFBLK	VSPSIZE	VSPVPAGE	
DMKVSX	AFREE	APTRAN	ASYSVM	BLANKS	BRING	CC	CD	CE	CLASFBA	CLASURI
	CMDREJ	C1	DATACHK	DE	DEFER	DMKCVTDT	DMKFREE	DMKPSACC	DMKPSASC	DMKPTRAN
	DMKRPAGT	DMKSCNDC	DMKTMRPT	DMKVMASH	DMKVSRGC	DMKVSRMD	DMKVSWDC	DMKVSWFC	DMKVSWOC	DMKVSWOT
	F0	F1	F3	F4095	IDA	IL	INTREQ	IOERETN	LOCK	PCI
	PCIF	PRGC	PSA	RDCBLOK	RDCPAGAP	RDEVBLOK	RDEVRDC	RDEVTYPC	RECBLOK	RECCYL
	RECMAP	RECPNT	RECSIZE	RECUSED	R0	R1	R11	R12	R13	R14
	R15	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA
	SAVEREGS	SAVER6	SAVER7	SAVER8	SFBEOF	SFBFLAG	SFBFLAG3	SFBLAST	SFBLDBEG	SFBLDMID
	SFBLOK	SFBOPEN	SFBRECER	SFBRECS	SFBRECSZ	SFBTYPE	SILI	SKIP	SPLINK	SPNXTPAG
	SPRECNUM	SPSIZE	SYSTEM	TEMPR0	TEMPR1	TYP3203	TYP3211	TYP3262	TYP3289E	TYP3505
	TYP3800	TYP38003	TYP4245	TYP4248	TYP5ACCW	UC	UE	VMBLOK	VSPBUFBK	VSPBUFSZ
	VSPCAW	VSPCCW	VSPDPAGE	VSPIDACT	VSPIDASW	VSPLCTL	VSPNEXT	VSPRECNO	VSPSFBLK	VSPVPAGE
DMKWAI	ADSPCH	ALOKSY	APSTAT1	APUOPER	CPEXBLOK	CPEXFPNT	CPEXR11	CPSHRLK	CPSTAT2	CPUID
	DMKDSP	DMKDSPCH	DMKDSPWA	DMKVMASW	DSPRQ	DSPWA	IOBFPNT	IOBLOK	IOBUSER	LOCK
	LPUADDR	PREFIXA	PSA	PSAEXT	PXA	RQLKBASE	RUNPSW	R0	R1	R10
	R11	R12	R13	R14	R15	R2	R3	R4	R5	R6
	R7	R8	R9	TIMEDISP	TRLANCHR	TRLCT	TRLINQ	TRLLABEL	TRLLOCK	TRLSTL
	VMAFF	VMBLOK	VMBTRL	VMCPWAIT	VMDEFSTK	VMDSTAT	VMEPRIOR	VMFTRL	VMIHIST	VMLKHIST
	VMLOCK	VMOSTAT	VMPATH	VMPEND	VMRLPROC	VMRSTAT	VMRUN	VMSHR	VMSHRPRC	VMSTKCPU
	VMXUNSTK	WAITEND	XTNDLOCK							
DMKWRM	ACNTBACK	ACNTBLOK	ACNTDATA	ACNTNEXT	ACNTNUM	ACNTSIZE	ACNTUSER	ACORETBL	ADDSFB	AFREE
	AFRET	ALARM	ALOCBLOK	ALOCMAP	ALOCNPAG	ALOCRECS	APSTAT1	APTRAN	APTRLK	APUOPER
	AQCNWT	ARIODV	ARSPPR	ARSPPU	ASYSVM	BRING	BUFFVER	CKPBLOK	CKPNAME	CKPRMAX
	CKPSIZE	CLASFBA	CLASSPEC	CLASTERM	CORCP	CORFLAG	CORTABLE	CORVM	CPSTAT5	C1
	DEFER	DMKCKTMP	DMKCKVWM	DMKCPIOL	DMKCVTBD	DMKDSPNP	DMKERMCP	DMKERMSG	DMKFREE	DMKFRET

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	DMKMCHST	DMKPGUVG	DMKPGUVR	DMKPTRAN	DMKPTRFR	DMKPTRPS	DMKQCNWT	DMKQCOSY	DMKRPAGT	DMKRPAPT
	DMKRPASV	DMKRSPAC	DMKRSPCP	DMKRSPCR	DMKRSPCU	DMKRSPCV	DMKRSPDC	DMKRSPDL	DMKRSPFC	DMKRSPHQ
	DMKRSPPC	DMKRSPPR	DMKRSPPU	DMKRSPRD	DMKRSPSC	DMKRSPST	DMKRSPTP	DMKRSPWC	DMKSCNRU	DMKSNTQN
	DMKSYSDT	DMKSYSHE	DMKSYSHL	DMKSYSHT	DMKSYSLG	DMKSYSOW	DMKSYSSP	DMKSYSWA	DMKSYSWI	DMKSYSWM
	DMKWRNSB	DMKWRNWM	DUMP	FF	FFS	F0	F1	F2	F256	F4096
	IOKEEP	LOCK	NICBLOK	NICDISA	NICENAB	NICFLAG	NICLGRP	NICSIZE	NICSTAT	NICTERM
	NICTYPE	NPRCNT	NPRNAME	NPRPNT	NPRTBL	OPERATOR	OWNDLIST	OWNDRDEV	PCHCHN	PRTCHN
	PSA	RDEVALLN	RDEVAUTO	RDEVBLOK	RDEVCKP	RDEVCKPT	RDEVCLAS	RDEVCODE	RDEVDISA	RDEVDRAN
	RDEVENAB	RDEVEXTN	RDEVFLAG	RDEVFSEP	RDEVIMAG	RDEVMAX	RDEVNCP	RDEVNICL	RDEVPRFG	RDEVPTHS
	RDEVSEP	RDEVSER	RDEVSPL	RDEVSTAT	RDEVTYPC	RDEVTYPE	RDEVXSEP	RDIDX	RDRCHN	RECBLOK
	RECBUFF	RECCYL	RECFSZ	RECPNT	RECSIZE	RECUSED	RHTBLOK	RHTECNT	RHTVRFY	RHXADDR
	RHXTABLE	RSPXAUTO	RSPXBLOK	RSPXDEST	RSPXFCB	RSPXFLAG	RSPXFOLD	RSPXFORM	RSPXINDX	RSPXSETU
	R0	R1	R10	R11	R12	R13	R14	R15	R2	R3
	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVER2	SAVEWRK1
	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK6	SAVEWRK8	SFBBCONV	SFBCONV	SFBDATE	SFBEOF	SFBFLAG
	SFBFLAG2	SFBFLAG4	SFBFNAME	SFBFTYPE	SFBINUSE	SFBLOK	SFBOFORM	SFBOPEN	SFBPNT	SFBPURGD
	SFBPURGE	SFBRECER	SFBRECS	SFBRSTRT	SFBSIZE	SFBSIZEB	SFBSP3	SFBSYSID	SFBTUSE	SFBXAB
	SFBXABER	SHQBLOK	SHQBSIZE	SPECIALV	SPUBLOK	SPUIND	SPUSYSID	STARTIME	SWPCHG1	SWPFLAG
	SWPRECMP	SWPTABLE	SYSTEM	TYPBSC	TYP3705	TYP3800	TYP38003	TYP38008	VERNULL	VERSP3
	VMBLOK	VMSIZE	XPAGNUM	ZEROES						
DMKWRN	AFRET	APTRAN	APTRLK	ASYSVM	BRING	CHGSHQ	CLASFBA	CPSTAT5	C1	DEFER
	DMKCKSCV	DMKCKSPL	DMKERMSG	DMKFRET	DMKPGUVG	DMKPGUVR	DMKPTRAN	DMKPTRUL	DMKRIODV	DMKRIOPR
	DMKRIOPU	DMKRPAGT	DMKRPAPT	DMKRSPCP	DMKRSPCU	DMKRSPDL	DMKRSPHQ	DMKRSPND	DMKRSPPC	DMKRSPPR
	DMKRSPPU	DMKRSPRD	DMKRSPSC	DMKRSPSM	DMKRSPST	DMKRSPSW	DMKRSPTP	DMKSYSWI	DMKVSDAD	DMKVSDDL
	DMKVSGAI	DMKVSGRI	F0	F1	LOCK	PSA	RDEVBLOK	RDEVSPL	RDIDX	RHTBLOK
	RHTVIRT	R0	R1	R10	R11	R12	R13	R14	R15	R2
	R3	R4	R5	R6	R7	R8	SAVEAREA	SAVEREGS	SFBCDMP	SFBCONTO
	SFBFILID	SFBFLAG4	SFBFLAG5	SFBINVS	SFBLOK	SFBORIG	SFBSIZE	SFBSIZEB	SFBSYSID	SFBUSER
	SHQBLOK	SHQCKPT	SYSTEM	VMBLOK	ZEROES					
DMKXAB	AFREE	AFRET	APTRAN	APTRLK	ARSPPR	ASYSVM	BRING	CHGSFB	CLASURO	C1
	DEFER	DMKCKSPL	DMKFREE	DMKFRET	DMKLOCRD	DMKLOCRQ	DMKPGTSG	DMKPGUSD	DMKPGUVG	DMKPGUVR
	DMKPSAFP	DMKPSASP	DMKPTRAN	DMKPTRLK	DMKPTRUL	DMKRPAGT	DMKRPAPT	DMKSCNVU	DMKVSFID	FFS
	F0	F1	F10	F16	F4096	IOERETN	LOCK	PSA	R0	R1
	R10	R11	R12	R13	R14	R15	R2	R3	R4	R5
	R6	R7	R8	R9	SAVEAREA	SAVEREGS	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4
	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK8	SAVEWRK9	SFBCONV	SFBFILID	SFBFLAG	SFBFLAG4	SFBINUSE
	SFBLOK	SFBPNT	SFBPURGD	SFBTUSE	SFBTYPE	SFBUHOLD	SFBUSER	SFBXAB	SFBXABL	SFLAG
	SPLINK	SPNXTPAG	SPPREPAG	SYSTEM	TYPPRT	VMBLOK	VMSIZE	XPAGNUM	X2048BND	
DMKXAD	AP	BRING	CLASURO	DEFER	DMKPGTSG	DMKPGUVG	DMKPGUVR	DMKRPAGT	DMKRPAPT	FFS
	F0	LOCK	MP	PSA	R0	R1	R10	R11	R12	R13
	R14	R15	R2	R3	R4	R5	R6	R7	R8	R9
	SAVEAREA	SAVEREGS	SAVEWRK1	SAVEWRK2	SAVEWRK3	SFBFLAG4	SFBLOK	SFBXAB	SFBXABER	SFBXABL
	SPLINK	SPNXTPAG	SPPREPAG	SYSTEM	TYPPRT	VMBLOK	ZEROES			
DMKXST	AFREE	ALOCBLOK	ALOCCHN	ALOCCSR	ALOCCYL1	ALOCCYL2	ALOCFLG	ALOCMAP	ALOCMAX	ALOCNPAG
	ALOCPAGE	ALOCPAVL	ALOCPMAX	ALOCPNT	ALOCPUSE	ALOCRDEV	ALOCRWRT	ALOCSW	ALOCTMSI	ALOCUSED
	AP	APSTALOC	DMKCVTBD	DMKERMSG	DMKFREE	DMKMHCCP	DMKPGTDL	DMKPGTDM	DMKPGTPB	DMKPGTPI
	DMKPGTPN	DMKPGTPT	DMKPGTPU	DMKPGTSB	DMKPGTSI	DMKPGTSL	DMKPGTST	DMKPGTSU	DMKSYSPP	DMKSYSSW
	DMKSYSSZ	F1	F4096	F8	HCBLOK	HCMSFCW	HCMSFDB	HCRMAP	HCSIZE	LOCK
	MP	PSA	PSTRCBYT	PSTRCPAG	RECBAK	RECBLOK	RECCYL	RECMAP	RECMAX	RECPNT
	RECSIZE	RECUSED	R0	R1	R10	R11	R12	R13	R14	R15
	R2	R3	R4	R5	R6	R7	R8	R9	SAVEAREA	SAVEREGS

MODULE	EXTERNAL REFERENCES (LABELS AND MODULES)									
	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5	SAVEWRK6	SAVEWRK7	SAVEWRK9	SCCBIMAP	SCCBINUM
	SCCBLEN	SCCBLOK	SCCBRESP	SCCB0010	SYSPALOC	SYSPCYL1	SYSPCYL2	SYSPFLG	SYSPFLG2	SYSPLIST
	SYSPXST									
DMKZTD	ADSPCH	AFREE	AFREEP	AFRET	ALOCBLOK	ALOCTDK	ALOCUSED	AP	CC	CLASFBA
	CPEXADD	CPEXBLOK	CPEXFPNT	CPEXR0	CPEXR1	CPEXR11	CPEXR13	CPEXR4	CPEXR6	CPEXR8
	CPEXSIZE	DMKCPXZT	DMKDSPCH	DMKFREE	DMKFREEP	DMKFRET	DMKIOSQR	D8	FFS	FTR70MB
	F0	F8	IOBCAW	IOBCC3	IOBCP	IOBCSW	IOBCYL	IOBFATAL	IOBFLAG	IOBIRA
	IOBLINK	IOBLOK	IOBMISC	IOBSIZE	IOBSTAT	IOBUSER	LOCK	L16	L8	MP
	PSA	RDCBLKAP	RDCBLOK	RDEVBLOK	RDEVCC3	RDEVCPEX	RDEVCPIO	RDEVECKD	RDEVFTR	RDEVMDL
	RDEVRDC	RDEVSTAT	RDEVSTA3	RDEVTYPC	RDEVTYPE	R0	R1	R10	R11	R12
	R13	R14	R15	R2	R3	R4	R5	R6	R7	R8
	R9	SAVEAREA	SAVEREGS	SAVER8	SAVER9	SAVEWRK1	SAVEWRK2	SAVEWRK3	SAVEWRK4	SAVEWRK5
	SAVEWRK6	SAVEWRK9	SILI	TYP2305	TYP2314	TYP3330	TYP3340	TYP3350	TYP3375	TYP3380
	ZEROES									

LABEL	COUNT	REFERENCES									
ABORT	000003	DMKNLD	DMKNLE	DMKRNH							
ACCNT	000005	DMKCKF	DMKCKP								
ACCNTERR	000005	DMKCKF	DMKCKP								
ACCTACNO	000003	DMKHVD									
ACCTBLOK	000005	DMKACO	DMKCKF	DMKHVD	DMKSPL						
ACCTDIST	000002	DMKHVD	DMKSPL								
ACCTLENG	000003	DMKHVD									
ACCTUSER	000004	DMKACO	DMKCKF	DMKHVD	DMKSPL						
ACIADDR	000001	DMKLNK									
ACIALTU	000001	DMKHVF									
ACIAUSR	000001	DMKHVF									
ACICLR1	000003	DMKCSP	DMKCSX	DMKLNK							
ACICMD	000001	DMKCFC									
ACICODE	000009	DMKCFC	DMKCSP	DMKCST	DMKCSX	DMKHVF	DMKLNK	DMKLOG	DMKRPI	DMKRPW	
ACIDEFR	000002	DMKRPI	DMKRPW								
ACIDEL	000001	DMKUSQ									
ACIFCN	000010	DMKCFC	DMKCSP	DMKCST	DMKCSX	DMKHVF	DMKLNK	DMKLOG	DMKUSQ		
ACILINK	000001	DMKLNK									
ACILOG	000002	DMKLOG									
ACIMODE	000007	DMKCSP	DMKCSX	DMKLNK							
ACINOAC	000001	DMKHVF									
ACINODE	000002	DMKCST									
ACINPMT	000001	DMKLOG									
ACIPARMS	000027	DMKCFC	DMKCSP	DMKCST	DMKCSX	DMKHVF	DMKLNK	DMKLOG	DMKRPI	DMKRPW	DMKUSQ
ACIPASS	000001	DMKLOG									
ACIRGRP	000009	DMKCFC	DMKCSP	DMKCST	DMKCSX	DMKLNK	DMKLOG	DMKUSQ			
ACIRUSR	000008	DMKCFC	DMKCSP	DMKCST	DMKCSX	DMKHVF	DMKLNK	DMKLOG	DMKUSQ		
ACISIZE	000025	DMKCFC	DMKCSP	DMKCST	DMKCSX	DMKHVF	DMKLNK	DMKLOG	DMKUSQ		
ACISPOOL	000002	DMKCSP	DMKCSX								
ACISTCP	000001	DMKCFC									
ACITAG	000001	DMKCST									
ACITGRP	000004	DMKCSP	DMKCSX	DMKLNK							
ACITUSR	000002	DMKCSP									
ACIUDIR	000003	DMKCSP	DMKCSX	DMKLNK							
ACIUSRID	000006	DMKCSX	DMKLNK								
ACIWUSR	000001	DMKHVF									
ACNTBACK	000004	DMKACO	DMKWRM								
ACNTBLOK	000018	DMKACO	DMKCKF	DMKHVD	DMKJRL	DMKWRM					
ACNTCODE	000018	DMKACO	DMKCKF	DMKHVD							
ACNTCONT	000002	DMKACO	DMKCKF								
ACNTDATA	000028	DMKACO	DMKCKF	DMKCQU	DMKHVD	DMKWRM					
ACNTDEVC	000004	DMKACO	DMKCKF								
ACNTDEVM	000004	DMKACO	DMKCKF								
ACNTDEVT	000002	DMKACO	DMKCKF								
ACNTIOCT	000002	DMKACO	DMKCKF								
ACNTNCYL	000004	DMKACO	DMKCKF								
ACNTNEXT	000012	DMKACO	DMKCKF	DMKWRM							
ACNTNUM	000006	DMKACO	DMKCKF	DMKHVD	DMKWRM						
ACNTPGRD	000002	DMKACO	DMKCKF								
ACNTRFLG	000004	DMKACO	DMKCKF								
ACNTRMID	000008	DMKACO	DMKCKF								
ACNTRMTE	000004	DMKACO	DMKCKF								
ACNTSIZE	000010	DMKACO	DMKHVD	DMKJRL	DMKWRM						
ACNTSNA	000001	DMKACO									
ACNTSTOP	000018	DMKACO	DMKCKF								
ACNTTIME	000002	DMKACO	DMKCKF								
ACNTUSER	000008	DMKACO	DMKCKF	DMKHVD	DMKWRM						
ACNTVTIM	000002	DMKACO	DMKCKF								

LABEL	COUNT	REFERENCES									
ACNTVTT	000002	DMKACO	DMKCKF								
ACNTVVT	000002	DMKACO	DMKCKF								
ACOACCL	000001	DMKACO									
ACOACCM	000002	DMKACO									
ACOCHK	000001	DMKACO									
ACOEXIT	000005	DMKACO									
ACORETBL	000201	DMKATS	DMKBLD	DMKCCW	DMKCDB	DMKCDM	DMKCDS	DMKCFQ	DMKCFU	DMKCPP	DMKCPU
		DMKCPY	DMKDGD	DMKDGF	DMKDRD	DMKDRE	DMKFRT	DMKGRE	DMKHPT	DMKHPU	DMKHVD
		DMKHVE	DMKIDU	DMKMCC	DMKMCH	DMKMHV	DMKMNI	DMKPAG	DMKPGM	DMKPGS	DMKPMA
		DMKPRW	DMKPSA	DMKPST	DMKPTR	DMKPTS	DMKPTT	DMKQVM	DMKRPA	DMKSEG	DMKSEL
		DMKSPM	DMKSTA	DMKSTR	DMKSWA	DMKSWM	DMKUDU	DMKUNT	DMKVCN	DMKVDR	DMKVFC
		DMKVFD	DMKVFE	DMKVFS	DMKVMA	DMKVMC	DMKVRR	DMKVRS	DMKVSD	DMKWRM	
ACORETN	000002	DMKACO									
ACRLOCK	000007	DMKACR	DMKACS	DMKCCH	DMKCPO	DMKMCH	DMKMCT	DMKVSI			
ACTIAC	000004	DMKACO									
ACTIBF1R	000008	DMKACO									
ACTIBF1V	000005	DMKACO									
ACTIBF2R	000006	DMKACO									
ACTIBF2V	000005	DMKACO									
ACTIBLOK	000003	DMKACO	DMKCKF								
ACTICL	000004	DMKACO									
ACTICLAS	000002	DMKACO									
ACTICNT	000005	DMKACO									
ACTIDCUR	000017	DMKACO									
ACTIDISP	000007	DMKACO									
ACTIFLAG	000020	DMKACO	DMKCKF								
ACTIID	000003	DMKACO									
ACTILIMT	000004	DMKACO									
ACTIPCH	000005	DMKACO	DMKCKF								
ACTISFB	000009	DMKACO	DMKCKF								
ACTISFCK	000008	DMKACO	DMKCKF								
ACTIVTRQ	000018	DMKAPI	DMKCPP	DMKDSP	DMKEXT	DMKPMA	DMKQVM	DMKTRQ			
ACTSFB	000004	DMKCKR	DMKCKS								
ADDSFB	000026	DMKACO	DMKCKS	DMKMIA	DMKNLE	DMKSPL	DMKSPS	DMKTRT	DMKTRU	DMKVME	DMKVSD
		DMKVST	DMKWRM								
ADMKCPE	000001	DMKSAD									
ADSPCH	000332	DMKACO	DMKACR	DMKALG	DMKAPS	DMKAPT	DMKAPY	DMKATS	DMKBIO	DMKCAC	DMKCCH
		DMKCCW	DMKCFG	DMKCFM	DMKCFO	DMKCFQ	DMKCFR	DMKCKF	DMKCKV	DMKCNS	DMKCPB
		DMKCPI	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPW	DMKCPX	DMKCPZ	DMKCQC	DMKCQT
		DMKCRM	DMKCSB	DMKDAD	DMKDAS	DMKDAU	DMKDGD	DMKDGF	DMKDIA	DMKDIB	DMKDID
		DMKDRD	DMKDRE	DMKDSB	DMKDSP	DMKENT	DMKEXT	DMKGIO	DMKGRD	DMKGRE	DMKGRF
		DMKGRG	DMKGRT	DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKIDR
		DMKIOE	DMKIOF	DMKIOH	DMKIOS	DMKIOT	DMKIUA	DMKIUC	DMKIUJ	DMKIUN	DMKIUP
		DMKIUS	DMKLOC	DMKLOJ	DMKLOK	DMKLOM	DMKMCD	DMKMCH	DMKMCI	DMKMCT	DMKMHC
		DMKMID	DMKMNI	DMKMNL	DMKMON	DMKMOO	DMKMPO	DMKMSG	DMKNLD	DMKNLE	DMKPAG
		DMKPAH	DMKPGM	DMKPGS	DMKPGT	DMKPRG	DMKPRV	DMKPRW	DMKPTR	DMKPTS	DMKQCN
		DMKQCO	DMKQCP	DMKREI	DMKRGA	DMKRGB	DMKRNH	DMKRSF	DMKRSP	DMKRST	DMKSCH
		DMKSEL	DMKSEP	DMKSND	DMKSPK	DMKSPM	DMKSPS	DMKSSS	DMKSSU	DMKSTR	DMKSVC
		DMKSVD	DMKSWA	DMKTAP	DMKTCS	DMKTCT	DMKTHI	DMKTMR	DMKTPE	DMKTRD	DMKTRK
		DMKTRP	DMKTRQ	DMKTRT	DMKTRU	DMKTRX	DMKUDR	DMKUNT	DMKUSQ	DMKVAT	DMKVCA
		DMKVCB	DMKVCN	DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCU	DMKVCW	DMKVCX
		DMKVDA	DMKVDC	DMKVDE	DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVFC	DMKVFD	DMKVFE
		DMKVFR	DMKVFS	DMKVIO	DMKVMA	DMKVMC	DMKVSE	DMKVSG	DMKVSI	DMKVSJ	DMKVSP
		DMKVSQ	DMKVST	DMKWAI	DMKZTD						
ADTRANS	000015	DMKPRV									
AEXTSP	000055	DMKACO	DMKCCH	DMKCFP	DMKCLK	DMKCPI	DMKCPO	DMKCPP	DMKCPS	DMKCPU	DMKDSP
		DMKFRE	DMKIOS	DMKMCH	DMKMCT	DMKMHC	DMKMOO	DMKPTR	DMKPTT	DMKSCH	DMKSEL
		DMKSTK									

LABEL	COUNT	REFERENCES									
AFREE	000745	DMKACO	DMKACR	DMKACS	DMKALG	DMKALO	DMKAPI	DMKAPS	DMKAPT	DMKAPU	DMKAPW
		DMKAPX	DMKAPY	DMKAPZ	DMKATS	DMKBIO	DMKBLD	DMKBSC	DMKCAC	DMKCAO	DMKCCH
		DMKCCS	DMKCCW	DMKCDB	DMKCDM	DMKCFC	DMKCFD	DMKCFF	DMKCFH	DMKCFJ	DMKCFM
		DMKCFP	DMKCFQ	DMKCFR	DMKCFS	DMKCFT	DMKCFU	DMKCFV	DMKCFY	DMKCKR	DMKCKT
		DMKCKV	DMKCNS	DMKCPB	DMKCPI	DMKCPJ	DMKCPN	DMKCPO	DMKCPS	DMKCPT	DMKCPU
		DMKCPW	DMKCPZ	DMKCQC	DMKCQG	DMKCQH	DMKCQI	DMKCQP	DMKCQQ	DMKCQR	DMKCQS
		DMKCQT	DMKCQU	DMKCQY	DMKCRM	DMKCSB	DMKCSC	DMKCSF	DMKCSO	DMKCSP	DMKCSQ
		DMKCST	DMKCSU	DMKCSV	DMKCSW	DMKCSX	DMKCSY	DMKDAD	DMKDAS	DMKDAU	DMKDEF
		DMKDEG	DMKDEI	DMKDEX	DMKDGD	DMKDGF	DMKDIA	DMKDIB	DMKDID	DMKDRD	DMKDRE
		DMKDSB	DMKDSP	DMKEPS	DMKERM	DMKFRE	DMKGRA	DMKGRC	DMKGRF	DMKGRG	DMKGRI
		DMKGRT	DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKIDR	DMKIDU
		DMKIOE	DMKIOF	DMKIOG	DMKIOH	DMKIOS	DMKISM	DMKIUA	DMKIUB	DMKIUC	DMKIUG
		DMKIUJ	DMKIUN	DMKIUP	DMKIUS	DMKJRL	DMKLDOOE	DMKLNK	DMKLNM	DMKLOC	DMKLOG
		DMKLOH	DMKLOJ	DMKMCC	DMKMCD	DMKMCH	DMKMHC	DMKMHV	DMKMIA	DMKMNI	DMKMNJ
		DMKMNL	DMKMNT	DMKMOO	DMKMPO	DMKMSG	DMKMSW	DMKNEA	DMKNES	DMKNET	DMKNLD
		DMKNLE	DMKOPE	DMKPAG	DMKPEI	DMKPEL	DMKPEN	DMKPER	DMKPET	DMKPGM	DMKPGS
		DMKPGT	DMKPST	DMKPTR	DMKPTT	DMKQCN	DMKQCO	DMKQCP	DMKQCQ	DMKREI	DMKRGA
		DMKRGB	DMKRGC	DMKRGE	DMKRNH	DMKRSE	DMKRSF	DMKRSP	DMKRSQ	DMKRST	DMKSCH
		DMKSEL	DMKSND	DMKSPK	DMKSPL	DMKSPM	DMKSPR	DMKSPS	DMKSPT	DMKSRM	DMKSSS
		DMKSSU	DMKSTP	DMKSVD	DMKSWA	DMKSWM	DMKTAP	DMKTAQ	DMKTCS	DMKTHI	DMKTMR
		DMKTOD	DMKTPE	DMKTRA	DMKTRK	DMKTRP	DMKTRQ	DMKTRT	DMKTRU	DMKTRX	DMKTTX
		DMKTTY	DMKUDR	DMKUDU	DMKURS	DMKUSP	DMKUSQ	DMKVAT	DMKVAU	DMKVBM	DMKVCA
		DMKVCB	DMKVCH	DMKVCN	DMKVCP	DMKVCQ	DMKVCR	DMKVCT	DMKVCU	DMKVCV	DMKVCW
		DMKVCX	DMKVDA	DMKVDC	DMKVDD	DMKVDE	DMKVDG	DMKVDH	DMKVDR	DMKVDS	DMKVDT
		DMKVER	DMKVFC	DMKVFD	DMKVFE	DMKVFS	DMKVMA	DMKVMC	DMKVMD	DMKVME	DMKVMG
		DMKVRR	DMKVSD	DMKVSG	DMKVSI	DMKVSP	DMKVSQ	DMKVST	DMKVSW	DMKVSX	DMKWRM
		DMKXAB	DMKXST	DMKZTD							
AFREEP	000287	DMKACO	DMKACS	DMKALG	DMKAPS	DMKAPT	DMKATS	DMKBLD	DMKCCH	DMKCCS	DMKCCW
		DMKCDS	DMKCFG	DMKCFM	DMKCFO	DMKCFQ	DMKCFR	DMKCKF	DMKCKV	DMKCNS	DMKCPB
		DMKCPI	DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPU	DMKCPV	DMKCPW
		DMKCPX	DMKCQC	DMKCSO	DMKCSR	DMKCSW	DMKCSX	DMKDAD	DMKDAS	DMKDAU	DMKDGD
		DMKDIA	DMKDIB	DMKDID	DMKDRD	DMKDSB	DMKDSP	DMKEXT	DMKFRE	DMKGIO	DMKGRG
		DMKGRI	DMKHPT	DMKHPU	DMKHVC	DMKHVE	DMKHVF	DMKIOE	DMKIOF	DMKIOQ	DMKIOS
		DMKIOT	DMKIUA	DMKIUC	DMKIUJ	DMKIUN	DMKIUP	DMKIUS	DMKLOC	DMKLOJ	DMKLOK
		DMKMCC	DMKMCD	DMKMCH	DMKMCI	DMKMCT	DMKMHC	DMKMIA	DMKMID	DMKMNI	DMKMNL
		DMKMON	DMKMPO	DMKNLD	DMKNLE	DMKPGM	DMKPGS	DMKPGT	DMKPRV	DMKPTR	DMKPTS
		DMKQCO	DMKQCP	DMKQVM	DMKRGA	DMKRGB	DMKRGC	DMKRNH	DMKRPA	DMKRSE	DMKRSP
		DMKRST	DMKSEL	DMKSND	DMKSPK	DMKSPL	DMKSPM	DMKSSS	DMKSSU	DMKSTP	DMKSTR
		DMKSVC	DMKSWA	DMKSWM	DMKTAP	DMKTHI	DMKTPE	DMKTRD	DMKTRK	DMKTRT	DMKTRU
		DMKTRX	DMKUNT	DMKUSQ	DMKVAT	DMKVCA	DMKVCB	DMKVCN	DMKVCP	DMKVCU	DMKVCX
		DMKVDC	DMKVDD	DMKVDE	DMKVDR	DMKVDS	DMKVFC	DMKVFD	DMKVFE	DMKVFS	DMKVIO
		DMKVMA	DMKVMC	DMKVRR	DMKVSE	DMKVSG	DMKVSI	DMKVSJ	DMKVSP	DMKVSQ	DMKVST
		DMKZTD									
AFRET	000986	DMKACO	DMKACR	DMKACS	DMKAPS	DMKAPT	DMKAPV	DMKAPW	DMKAPX	DMKAPY	DMKAPZ
		DMKATS	DMKBIO	DMKBLD	DMKBSC	DMKCAC	DMKCAO	DMKCCH	DMKCCS	DMKCCW	DMKCDB
		DMKCDM	DMKCDS	DMKCFC	DMKCFD	DMKCFF	DMKCFG	DMKCFH	DMKCFM	DMKCFP	DMKCFQ
		DMKCFR	DMKCFS	DMKCFT	DMKCFU	DMKCFY	DMKCKV	DMKCNS	DMKCPB	DMKCPI	DMKCPJ
		DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPU	DMKCPW	DMKCPZ	DMKCQC	DMKCQG
		DMKCQH	DMKCQI	DMKCQP	DMKCQQ	DMKCQR	DMKCQS	DMKCQT	DMKCQU	DMKCQY	DMKCRM
		DMKCSB	DMKCSC	DMKCSO	DMKCSP	DMKCSQ	DMKCSR	DMKCST	DMKCSU	DMKCSV	DMKCSW
		DMKCSX	DMKCSY	DMKDAD	DMKDAS	DMKDAU	DMKDEF	DMKDEG	DMKDEI	DMKDEX	DMKDGD
		DMKDGF	DMKDIA	DMKDIB	DMKDID	DMKDIF	DMKDRD	DMKDRE	DMKDSB	DMKDSP	DMKEPS
		DMKERM	DMKFPS	DMKFRE	DMKGIO	DMKGRA	DMKGRC	DMKGRD	DMKGRF	DMKGRG	DMKGRI
		DMKGRT	DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKIDR	DMKIOE
		DMKIOF	DMKIOH	DMKIOQ	DMKIOS	DMKIOT	DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUG
		DMKIUJ	DMKIUN	DMKIUP	DMKIUS	DMKJRL	DMKLDOOE	DMKLNK	DMKLNM	DMKLOC	DMKLOG
		DMKLOH	DMKLOJ	DMKLOM	DMKMCC	DMKMCD	DMKMCH	DMKMHC	DMKMIA	DMKMNI	DMKMNL
		DMKMON	DMKMSG	DMKMSW	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKOPE	DMKPAH

LABEL	COUNT	REFERENCES									
		DMKPEI	DMKPEL	DMKPEN	DMKPET	DMKPGM	DMKPGS	DMKPGU	DMKPMA	DMKPST	DMKPTR
		DMKQCN	DMKQCO	DMKQCP	DMKQCQ	DMKQVM	DMKREI	DMKRGA	DMKRGB	DMKRGC	DMKRNH
		DMKRSE	DMKRSF	DMKRSP	DMKRST	DMKSEL	DMKSEP	DMKSND	DMKSPK	DMKSPL	DMKSPM
		DMKSPS	DMKSPT	DMKSRM	DMKSSS	DMKSSU	DMKSSV	DMKSVC	DMKSVD	DMKSWA	DMKSWM
		DMKTAP	DMKTAQ	DMKTCS	DMKTCT	DMKTHI	DMKTMR	DMKTPE	DMKTRA	DMKTRC	DMKTRD
		DMKTRK	DMKTRP	DMKTRQ	DMKTRT	DMKTRU	DMKTRX	DMKTTX	DMKUDR	DMKUDU	DMKUNT
		DMKURS	DMKUSP	DMKUSQ	DMKVAT	DMKVCA	DMKVCB	DMKVCH	DMKVCN	DMKVCP	DMKVCQ
		DMKVCR	DMKVCS	DMKVCT	DMKVCU	DMKVCV	DMKVCW	DMKVCX	DMKVDA	DMKVDC	DMKVDD
		DMKVDE	DMKVDG	DMKVDH	DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVFC	DMKVFD	DMKVFE
		DMKVIO	DMKVMA	DMKVMC	DMKVMD	DMKVME	DMKVMG	DMKVRR	DMKVSC	DMKVSD	DMKVSG
		DMKVSI	DMKVSP	DMKVSQ	DMKVST	DMKVSW	DMKWRM	DMKWRN	DMKXAB	DMKZTD	
ALARM	000104	DMKACO	DMKACR	DMKCCH	DMKCKH	DMKCKM	DMKCKN	DMKCKP	DMKCLK	DMKCNS	DMKCPI
		DMKCPJ	DMKCQP	DMKDAD	DMKDAS	DMKDAU	DMKDMQ	DMKDSB	DMKERM	DMKGRD	DMKGRH
		DMKJRL	DMKMCH	DMKMCT	DMKMID	DMKMSG	DMKMSW	DMKOPE	DMKOPR	DMKPGT	DMKRGB
		DMKRNH	DMKSAV	DMKTTY	DMKUDR	DMKURS	DMKVCN	DMKVCS	DMKVER	DMKWRM	
ALL	000039	DMKDMQ									
ALLCHANS	000009	DMKACR	DMKACS	DMKCPP	DMKIOG	DMKPMA					
ALOCALOC	000010	DMKMOO	DMKPGT	DMKSTP							
ALOCASGN	000001	DMKVDH									
ALOCBLOK	000127	DMKALO	DMKATS	DMKCKF	DMKCKT	DMKCKV	DMKCPX	DMKCPZ	DMKDAD	DMKDAS	DMKDAU
		DMKDMP	DMKIDU	DMKMOO	DMKPAG	DMKPGM	DMKPGS	DMKPGT	DMKPGU	DMKPST	DMKPTR
		DMKSEL	DMKSTP	DMKSTR	DMKSWA	DMKSWM	DMKTDK	DMKUSP	DMKVDG	DMKVDH	DMKWRM
		DMKXST	DMKZTD								
ALOCCHN	000044	DMKMOO	DMKPGT	DMKPST	DMKSTP	DMKTDK	DMKUSP	DMKVDG	DMKVDH	DMKXST	
ALOCCSR	000019	DMKMOO	DMKPGT	DMKPGU	DMKPST	DMKSEL	DMKVDG	DMKXST			
ALOCCYL1	000040	DMKMOO	DMKPAG	DMKPGT	DMKPGU	DMKPST	DMKTDK	DMKVDG	DMKVDH	DMKXST	
ALOCCYL2	000028	DMKCKF	DMKDMP	DMKPAG	DMKPGT	DMKPGU	DMKPST	DMKTDK	DMKVDG	DMKVDH	DMKXST
ALOCDLOC	000006	DMKPGU									
ALOCDOWN	000007	DMKPGT	DMKPGU	DMKVDG							
ALOCDU	000010	DMKCKF	DMKPGT	DMKVDG							
ALOCFLG	000074	DMKATS	DMKCKF	DMKDAD	DMKDAS	DMKDAU	DMKMOO	DMKPAG	DMKPGM	DMKPGS	DMKPGT
		DMKPGU	DMKPST	DMKSTR	DMKVDG	DMKVDH	DMKXST				
ALOCMALC	000009	DMKMOO	DMKPGM	DMKSTP	DMKSWM						
ALOCMAP	000050	DMKALO	DMKCKT	DMKCPZ	DMKIDU	DMKPGT	DMKPGU	DMKPST	DMKTDK	DMKVDG	DMKVDH
		DMKWRM	DMKXST								
ALOCMAX	000017	DMKALO	DMKCKT	DMKIDU	DMKPGT	DMKPST	DMKTDK	DMKVDG	DMKVDH	DMKXST	
ALOCMGIN	000004	DMKPGM	DMKSWM								
ALOCMGOU	000005	DMKPGM	DMKSTP	DMKSWM							
ALOCNPAG	000032	DMKALO	DMKCKT	DMKCKV	DMKIDU	DMKPGT	DMKPST	DMKVDG	DMKVDH	DMKWRM	DMKXST
ALOCPAGE	000027	DMKPGT	DMKPGU	DMKPST	DMKUSP	DMKVDG	DMKVDH	DMKXST			
ALOCPAVL	000011	DMKMOO	DMKPGT	DMKPST	DMKSTP	DMKVDG	DMKVDH	DMKXST			
ALOCPG	000021	DMKMOO	DMKPAG	DMKPGM	DMKPGT	DMKPGU	DMKSRM	DMKSTR	DMKVDG		
ALOCPM	000005	DMKPAG	DMKVDG								
ALOCPMAX	000012	DMKPGT	DMKPGU	DMKPST	DMKVDG	DMKVDH	DMKXST				
ALOCPNT	000038	DMKALO	DMKCKF	DMKDAD	DMKDAS	DMKDAU	DMKDMP	DMKIDU	DMKMOO	DMKPAG	DMKPGU
		DMKPST	DMKVDG	DMKVDH	DMKXST						
ALOCPP	000027	DMKATS	DMKMOO	DMKPAG	DMKPGM	DMKPGS	DMKPGT	DMKPGU	DMKSRM	DMKSTR	DMKVDG
ALOCPRD	000013	DMKPAG	DMKPGM	DMKPTR	DMKSTP	DMKSWM					
ALOCPREC	000015	DMKMOO	DMKPGT	DMKPGU	DMKUSP	DMKVDG					
ALOCPS	000022	DMKDAD	DMKDAS	DMKDAU	DMKPAG	DMKPGT	DMKPGU	DMKSRM	DMKSTP	DMKVDG	
ALOCPSIO	000016	DMKPAG	DMKPGM	DMKPTR	DMKSEL	DMKSWA	DMKSWM				
ALOCPUSE	000033	DMKMOO	DMKPGT	DMKPGU	DMKPST	DMKSTP	DMKVDG	DMKVDH	DMKXST		
ALOCPWRT	000009	DMKPAG	DMKSEL	DMKSTP	DMKSWA						
ALOCRCUU	000006	DMKALO	DMKPST	DMKVDH							
ALOCRDEV	000025	DMKALO	DMKIDU	DMKMOO	DMKPGT	DMKPGU	DMKPST	DMKTDK	DMKVDG	DMKVDH	DMKXST
ALOCRECS	000025	DMKCKF	DMKCKT	DMKCKV	DMKDMP	DMKIDU	DMKPGT	DMKPGU	DMKUSP	DMKVDG	DMKWRM
ALOCRSET	000006	DMKPGT	DMKPGU	DMKVDG							
ALOCRWRT	000005	DMKPST	DMKSEL	DMKVDG	DMKXST						

LABEL	COUNT	REFERENCES									
ALOCSREC	000002	DMKPGU									
ALOCSW	000035	DMKMOO	DMKPAG	DMKPGM	DMKPGT	DMKPGU	DMKPST	DMKSRM	DMKSWM	DMKVDG	DMKVDH
		DMKXST									
ALOCTDK	000006	DMKTDK	DMKVDH	DMKZTD							
ALOCTDSK	000008	DMKMOO	DMKTDK	DMKVDG	DMKVDH						
ALOCTMS	000009	DMKPAG	DMKPGT								
ALOCTMSI	000013	DMKPGT	DMKPST	DMKVDG	DMKXST						
ALOCUSED	000025	DMKMOO	DMKPGT	DMKPGU	DMKPST	DMKTDK	DMKVDG	DMKVDH	DMKXST	DMKZTD	
ALOKDS	000007	DMKAPI	DMKCFQ	DMKDSP	DMKMCT						
ALOKFR	000006	DMKFRE	DMKFRT	DMKMOO							
ALOKRM	000060	DMKCDS	DMKDGD	DMKFRE	DMKPAG	DMKPAH	DMKPGM	DMKPTR	DMKPTS	DMKPTT	DMKRPA
		DMKSEL	DMKSTR	DMKSWA	DMKSWM	DMKUSP	DMKVFD	DMKVFE	DMKVFS		
ALOKSP	000116	DMKACR	DMKACS	DMKAPI	DMKCCH	DMKCCS	DMKCDS	DMKCFQ	DMKCFR	DMKCKV	DMKCNS
		DMKCPM	DMKCPN	DMKCPO	DMKCPS	DMKCPT	DMKCPW	DMKCPZ	DMKCSO	DMKDEI	DMKDGD
		DMKDIA	DMKDIB	DMKDID	DMKDIF	DMKDSB	DMKDSP	DMKEXT	DMKFRE	DMKFRT	DMKGRI
		DMKIOQ	DMKIOS	DMKIOT	DMKLOH	DMKLOJ	DMKMCC	DMKMCT	DMKMNT	DMKMOO	DMKNES
		DMKNLD	DMKPAG	DMKPAH	DMKPGM	DMKPTR	DMKPTS	DMKPTT	DMKRGA	DMKRPA	DMKRSE
		DMKRSP	DMKSCH	DMKSEL	DMKSSS	DMKSST	DMKSSU	DMKSSV	DMKSTK	DMKSTR	DMKSWA
		DMKSWM	DMKTRQ	DMKUSP	DMKVCH	DMKVDA	DMKVDD	DMKVDE	DMKVDR	DMKVDS	DMKVDT
		DMKVFD	DMKVFE	DMKVFS	DMKVSJ						
ALOKSY	000069	DMKAPI	DMKCCH	DMKCCS	DMKCCW	DMKDGD	DMKDGF	DMKDSP	DMKEXT	DMKFRE	DMKFRT
		DMKIOS	DMKIOT	DMKLOK	DMKMCH	DMKMCT	DMKMHC	DMKPAG	DMKPRG	DMKPRV	DMKPTR
		DMKPTT	DMKSCH	DMKSEL	DMKSVC	DMKSVD	DMKTMR	DMKTRX	DMKVAT	DMKVAU	DMKVFR
		DMKVMA	DMKVSJ	DMKWAI							
ALOKTRL	000003	DMKAPI	DMKDSP								
ALOKVM	000006	DMKAPI	DMKCPP	DMKDSP	DMKMCH	DMKSEL					
ALPRTBLK	000008	DMKAPS	DMKAPU	DMKAPW	DMKAPY	DMKAPZ	DMKCKH				
ALTBLOK	000006	DMKHVF	DMKIUC	DMKSPL							
ALTSIZE	000016	DMKHVF									
ALTUSER	000008	DMKHVF	DMKIUC	DMKSPL							
ALTWRT	000003	DMKVCN									
AMCHAREA	000028	DMKACR	DMKCCH	DMKCFU	DMKCPO	DMKCPP	DMKCPU	DMKDID	DMKHVC	DMKIOG	DMKIOH
		DMKMCH	DMKMCI	DMKMHC	DMKMHV	DMKPRV	DMKVAT	DMKVAU			
ANYWRITE	000014	DMKQCN	DMKQCQ	DMKVCN							
AP	000362	DMKACO	DMKACR	DMKACS	DMKALG	DMKALO	DMKAPI	DMKAPS	DMKAPT	DMKAPU	DMKAPV
		DMKAPW	DMKAPX	DMKAPY	DMKAPZ	DMKBIO	DMKBLD	DMKCAO	DMKCCD	DMKCCF	DMKCCH
		DMKCCO	DMKCCS	DMKCCT	DMKCDS	DMKCFC	DMKCFF	DMKCFG	DMKCFH	DMKCFJ	DMKCFM
		DMKCFO	DMKCFP	DMKCFQ	DMKCFR	DMKCFS	DMKCFT	DMKCFU	DMKCFV	DMKCFW	DMKCFY
		DMKCKD	DMKCKF	DMKCKH	DMKCKM	DMKCKN	DMKCKP	DMKCKR	DMKCKS	DMKCKT	DMKCKV
		DMKCKW	DMKCNT	DMKCPB	DMKCPE	DMKCPI	DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPP
		DMKCPS	DMKCPT	DMKCPU	DMKCPV	DMKCPW	DMKCPX	DMKCPY	DMKCRM	DMKCSB	DMKCSC
		DMKCSF	DMKCSO	DMKCSP	DMKCSQ	DMKCSR	DMKCST	DMKCSU	DMKCSV	DMKCSW	DMKCSX
		DMKDEX	DMKDGD	DMKDRE	DMKDSB	DMKENT	DMKERP	DMKEXT	DMKHPS	DMKHPT	DMKIOE
		DMKIOG	DMKIOH	DMKIOJ	DMKIOQ	DMKIOS	DMKIOT	DMKLNM	DMKLOC	DMKLOG	DMKLOH
		DMKLOJ	DMKLOK	DMKLOM	DMKMCC	DMKMCD	DMKMCH	DMKMCI	DMKMNT	DMKMON	DMKMOO
		DMKMPO	DMKMSG	DMKNET	DMKNLD	DMKOPE	DMKPET	DMKPGM	DMKPGS	DMKPGT	DMKPRW
		DMKPSA	DMKPST	DMKPTR	DMKPTS	DMKPTT	DMKQCN	DMKQCO	DMKQCP	DMKRGA	DMKRGB
		DMKRGC	DMKRGD	DMKRNH	DMKRPA	DMKRPD	DMKRPI	DMKRPW	DMKRSE	DMKRSF	DMKRSP
		DMKRSQ	DMKSAV	DMKSBL	DMKSCH	DMKSCN	DMKSND	DMKSPK	DMKSPL	DMKSPM	DMKSPR
		DMKSPS	DMKSRM	DMKSSS	DMKSST	DMKSSU	DMKSSV	DMKSTA	DMKSTK	DMKSTP	DMKSVD
		DMKSWA	DMKTAQ	DMKTCS	DMKTMR	DMKTOD	DMKTRX	DMKTTX	DMKTTY	DMKTTZ	DMKUCB
		DMKUCC	DMKURS	DMKUSP	DMKVAU	DMKVBM	DMKVCA	DMKVCB	DMKVCH	DMKVCN	DMKVCP
		DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCU	DMKVCV	DMKVCW	DMKVER	DMKVFC	DMKVFD
		DMKVFE	DMKVFR	DMKVRS	DMKVSC	DMKVSD	DMKVSE	DMKVSF	DMKVSG	DMKVSI	DMKVSJ
		DMKVSP	DMKVSQ	DMKVSR	DMKVST	DMKVSU	DMKVSV	DMKVSW	DMKXAD	DMKXST	DMKZTD
APAGCP	000013	DMKCPI	DMKCPY	DMKPTR	DMKRST	DMKSVC					
APOINT	000001	DMKLD00E									
APSTALOC	000009	DMKMOO	DMKPGT	DMKPGU	DMKPST	DMKVDH	DMKXST				

LABEL	COUNT	REFERENCES									
APSTAT1	001111	DMKACO	DMKACR	DMKACS	DMKALG	DMKAPI	DMKATS	DMKBLD	DMKCAO	DMKCCH	DMKCCS
		DMKCCW	DMKCDB	DMKCDM	DMKCDS	DMKCFC	DMKCFF	DMKCFG	DMKCFM	DMKCFO	DMKCFP
		DMKCFQ	DMKCFR	DMKCFV	DMKCFY	DMKCKD	DMKCKF	DMKCKM	DMKCKV	DMKCLK	DMKCNS
		DMKCPI	DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPU	DMKCPV
		DMKCPW	DMKCPY	DMKCPZ	DMKCQP	DMKCQQ	DMKCQU	DMKCQY	DMKCSF	DMKCSO	DMKCSX
		DMKDAU	DMKDEI	DMKDGD	DMKDGF	DMKDIA	DMKDIB	DMKDID	DMKDIF	DMKDMQ	DMKDSB
		DMKDSP	DMKENT	DMKEXT	DMKFRE	DMKFRT	DMKGRI	DMKHPS	DMKHVD	DMKHVE	DMKIOG
		DMKIOQ	DMKIOS	DMKIOT	DMKIUA	DMKIUE	DMKIUL	DMKJRL	DMKLOG	DMKLOH	DMKLOJ
		DMKLOK	DMKLOM	DMKMCC	DMKMCD	DMKMCH	DMKMCI	DMKMCT	DMKMHC	DMKMIA	DMKMID
		DMKMNI	DMKMNJ	DMKMNT	DMKMON	DMKMOO	DMKMPO	DMKMSG	DMKMSW	DMKNEA	DMKNES
		DMKNET	DMKNLD	DMKOPE	DMKPAG	DMKPAH	DMKPGM	DMKPGS	DMKPMA	DMKPRG	DMKPRV
		DMKPRW	DMKPSA	DMKPST	DMKPTR	DMKPTS	DMKPTT	DMKQCN	DMKQCO	DMKQVM	DMKRGA
		DMKRGB	DMKRNH	DMKRPA	DMKRSE	DMKRSP	DMKSAD	DMKSCH	DMKSCN	DMKSEL	DMKSND
		DMKSPL	DMKSPM	DMKSPR	DMKSPS	DMKSSS	DMKSST	DMKSSU	DMKSSV	DMKSTK	DMKSTP
		DMKSTR	DMKSVC	DMKSVD	DMKSWA	DMKSWM	DMKTCS	DMKTCT	DMKTHI	DMKTMR	DMKTOD
		DMKTRQ	DMKTRX	DMKUDU	DMKUSP	DMKUSQ	DMKVAT	DMKVAU	DMKVBM	DMKVCB	DMKVCH
		DMKVCP	DMKVCS	DMKVCT	DMKVCU	DMKVCV	DMKVCW	DMKVCX	DMKVDA	DMKVDB	DMKVDC
		DMKVDD	DMKVDE	DMKVDF	DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVFC	DMKVFD	DMKVFE
		DMKVFR	DMKVFS	DMKVMA	DMKVMC	DMKVRR	DMKVSD	DMKVSG	DMKVSJ	DMKWAI	DMKWRM
APSTAT2	000054	DMKATS	DMKBLD	DMKCPP	DMKDSP	DMKFPS	DMKMCH	DMKMPO	DMKPGS	DMKPRV	DMKPRW
		DMKPTS	DMKRPA	DMKSEL	DMKSPM	DMKVAT	DMKVAU	DMKVMA			
APSTAT3	000004	DMKDSP	DMKEXT	DMKLOK							
APSTAT4	000063	DMKACR	DMKAPI	DMKCCH	DMKCLK	DMKCPO	DMKCPP	DMKCPU	DMKDSP	DMKENT	DMKEXT
		DMKIOG	DMKLOK	DMKMCH	DMKMCI	DMKMCT	DMKPRV	DMKVCH	DMKVFR		
APSTAT5	000011	DMKEXT	DMKLOK								
APTRAN	000326	DMKACO	DMKAPT	DMKATS	DMKBIO	DMKBLD	DMKCCH	DMKCCW	DMKCDB	DMKCDM	DMKCDS
		DMKCFC	DMKCFD	DMKCFF	DMKCFG	DMKCFH	DMKCFM	DMKCFS	DMKCFU	DMKCFV	DMKCKV
		DMKCNS	DMKCPB	DMKCPI	DMKCPM	DMKCPP	DMKCPS	DMKCPU	DMKCPW	DMKCPY	DMKCQS
		DMKCQY	DMKCSC	DMKCSO	DMKDGD	DMKDGF	DMKDRD	DMKDRE	DMKDSP	DMKERM	DMKGIO
		DMKGRA	DMKGRC	DMKGRF	DMKGRT	DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKHVD	DMKHVE
		DMKHVF	DMKIDU	DMKIOF	DMKIOG	DMKIOT	DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUL
		DMKLNK	DMKMCC	DMKMHV	DMKMIA	DMKMNJ	DMKNLD	DMKNLE	DMKOPE	DMKPEI	DMKPER
		DMKPGS	DMKPMA	DMKPRG	DMKPRV	DMKPRW	DMKPTR	DMKQCN	DMKQCO	DMKRGC	DMKRPA
		DMKRPD	DMKRSP	DMKRST	DMKSEG	DMKSEP	DMKSFB	DMKSNC	DMKSPK	DMKSPL	DMKSPT
		DMKSRM	DMKSSS	DMKSST	DMKSSV	DMKSVC	DMKSVD	DMKSWA	DMKTCS	DMKTCT	DMKTMR
		DMKTRC	DMKTRD	DMKTRK	DMKTRP	DMKTRU	DMKTRX	DMKTTX	DMKUDR	DMKUDU	DMKVAT
		DMKVAU	DMKVBM	DMKVCH	DMKVCN	DMKVCX	DMKVDR	DMKVER	DMKVIO	DMKVMC	DMKVMD
		DMKVME	DMKVSE	DMKVSG	DMKVSI	DMKVSJ	DMKVSP	DMKVSQ	DMKVSR	DMKVSX	DMKWRM
		DMKWRN	DMKXAB								
APTRLK	000166	DMKACO	DMKAPS	DMKAPT	DMKAPY	DMKATS	DMKBIO	DMKCCW	DMKCFC	DMKCFG	DMKCFH
		DMKCFQ	DMKCFR	DMKCFS	DMKCKV	DMKCPI	DMKCPM	DMKCPO	DMKCPP	DMKCPU	DMKCPW
		DMKCPX	DMKCQH	DMKCQI	DMKCQS	DMKCQY	DMKCRM	DMKCSC	DMKCSR	DMKCST	DMKCSW
		DMKCSY	DMKDGD	DMKDIB	DMKDRD	DMKDRE	DMKGRC	DMKHPS	DMKHPT	DMKHPU	DMKHVC
		DMKHVD	DMKHVE	DMKHVF	DMKIDR	DMKIOF	DMKIOG	DMKIOH	DMKIOT	DMKIUE	DMKIUJ
		DMKIUL	DMKLNK	DMKMCC	DMKMHV	DMKMIA	DMKMNI	DMKMNJ	DMKMSG	DMKNLD	DMKNLE
		DMKOPE	DMKPMA	DMKQCP	DMKREI	DMKRPD	DMKRSP	DMKRST	DMKSEP	DMKSFB	DMKSNC
		DMKSPK	DMKSPL	DMKSPT	DMKSSS	DMKSST	DMKSVC	DMKTAP	DMKTCS	DMKTCT	DMKTRC
		DMKTRP	DMKTRU	DMKTRX	DMKVBM	DMKVCH	DMKVCN	DMKVDR	DMKVMC	DMKVMD	DMKVME
		DMKVSD	DMKVSE	DMKVSG	DMKVSQ	DMKVSR	DMKWRM	DMKWRN	DMKXAB		
APUOPER	000867	DMKACO	DMKACR	DMKACS	DMKALG	DMKAPI	DMKATS	DMKBLD	DMKCAO	DMKCCH	DMKCCS
		DMKCCW	DMKCDB	DMKCDM	DMKCDS	DMKCFF	DMKCFG	DMKCFO	DMKCFP	DMKCFQ	DMKCFR
		DMKCFV	DMKCFY	DMKCKF	DMKCKV	DMKCLK	DMKCNS	DMKCPI	DMKCPJ	DMKCPM	DMKCPN
		DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPU	DMKCPV	DMKCPW	DMKCPY	DMKCPZ	DMKCQP
		DMKCQU	DMKCQY	DMKCSO	DMKCSX	DMKDEI	DMKDGD	DMKDGF	DMKDIA	DMKDIB	DMKDID
		DMKDIF	DMKDSB	DMKDSP	DMKENT	DMKEXT	DMKFRE	DMKFRT	DMKGRI	DMKHPS	DMKHVE
		DMKIOG	DMKIOQ	DMKIOS	DMKIOT	DMKIUA	DMKIUE	DMKIUL	DMKJRL	DMKLOG	DMKLOH
		DMKLOJ	DMKLOK	DMKLOM	DMKMCC	DMKMCD	DMKMCH	DMKMCI	DMKMCT	DMKMHC	DMKMIA
		DMKMID	DMKMNI	DMKMNJ	DMKMNT	DMKMON	DMKMOO	DMKMPO	DMKMSG	DMKMSW	DMKNEA

LABEL	COUNT	REFERENCES									
		DMKNES	DMKNET	DMKNLD	DMKOPE	DMKPAG	DMKPAH	DMKPGM	DMKPGS	DMKPMA	DMKPRG
		DMKPRV	DMKPRW	DMKPSA	DMKPST	DMKPTR	DMKPTS	DMKPTT	DMKQCN	DMKQCO	DMKQVM
		DMKRGA	DMKRGB	DMKRNH	DMKRPA	DMKRSE	DMKRSP	DMKSAD	DMKSCH	DMKSEL	DMKSND
		DMKSPL	DMKSPM	DMKSPR	DMKSPS	DMKSSS	DMKSST	DMKSSU	DMKSSV	DMKSTK	DMKSTP
		DMKSTR	DMKSVC	DMKSVD	DMKSWA	DMKSWM	DMKTCS	DMKTCT	DMKTHI	DMKTMR	DMKTOD
		DMKTRQ	DMKTRX	DMKUDU	DMKUSP	DMKUSQ	DMKVAT	DMKVAU	DMKVBM	DMKVCB	DMKVCH
		DMKVCP	DMKVCS	DMKVCT	DMKVCU	DMKVCV	DMKVCW	DMKVCX	DMKVDA	DMKVDB	DMKVDD
		DMKVDE	DMKVDF	DMKVDR	DMKVDS	DMKVDT	DMKVFC	DMKVFD	DMKVFE	DMKVFR	DMKVFS
		DMKVMA	DMKVMC	DMKVRR	DMKVSD	DMKVSG	DMKVSJ	DMKWAI	DMKWRM		
AQCNWT	000299	DMKACO	DMKACR	DMKBLD	DMKCAC	DMKCAO	DMKCCH	DMKCDB	DMKCDM	DMKCDS	DMKCFC
		DMKCFD	DMKCFF	DMKCFH	DMKCFM	DMKCFP	DMKCFS	DMKCFU	DMKCFV	DMKCFW	DMKCPB
		DMKCPJ	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPU	DMKCPV	DMKCPW	DMKCPY
		DMKCPZ	DMKCQC	DMKCQG	DMKCQH	DMKCQI	DMKCQP	DMKCQQ	DMKCQR	DMKCQS	DMKCQT
		DMKCQU	DMKCQY	DMKCSB	DMKCSF	DMKCSO	DMKDAD	DMKDAS	DMKDAU	DMKDEI	DMKDIA
		DMKDIB	DMKDIF	DMKDSB	DMKGRG	DMKJRL	DMKLNM	DMKLOM	DMKMCC	DMKMCD	DMKMCT
		DMKMIA	DMKMID	DMKMNI	DMKMNJ	DMKMSG	DMKMSW	DMKNEA	DMKNES	DMKNET	DMKNLD
		DMKNLE	DMKOPE	DMKPEI	DMKPEN	DMKPEQ	DMKPER	DMKPET	DMKPGM	DMKPGT	DMKPRG
		DMKPTR	DMKQCN	DMKQCP	DMKQVM	DMKRGC	DMKRNH	DMKSPL	DMKSPM	DMKSPR	DMKSPS
		DMKSRM	DMKSVD	DMKTHI	DMKTRA	DMKTRC	DMKTRD	DMKTRP	DMKTRU	DMKURS	DMKUSQ
		DMKVCB	DMKVCH	DMKVCN	DMKVDA	DMKVDB	DMKVDD	DMKVDR	DMKVDT	DMKVER	DMKVFC
		DMKVFD	DMKVFE	DMKVME	DMKWRM						
ARDCBLOK	000006	DMKCPW	DMKMNT								
ARIOCC	000002	DMKCKD	DMKQVM								
ARIOCH	000027	DMKACR	DMKACS	DMKCCH	DMKCKD	DMKCKM	DMKCPM	DMKCPO	DMKCPP	DMKCPS	DMKCPV
		DMKCQP	DMKENT	DMKGRG	DMKIOG	DMKIOQ	DMKIOS	DMKMNI	DMKMNT	DMKQVM	DMKSCN
		DMKSSP	DMKVCH								
ARIOCT	000030	DMKACR	DMKACS	DMKAPI	DMKCCH	DMKCFP	DMKCKD	DMKCKM	DMKCPM	DMKCPO	DMKCPP
		DMKCPS	DMKCPV	DMKENT	DMKIOG	DMKIOQ	DMKIOS	DMKMNT	DMKPMA	DMKSCN	DMKSSP
		DMKVCH	DMKVRR								
ARIOCU	000034	DMKACR	DMKACS	DMKCAC	DMKCAO	DMKCCH	DMKCKD	DMKCKM	DMKCPM	DMKCPO	DMKCPP
		DMKCPS	DMKCPV	DMKCPZ	DMKCQP	DMKDIF	DMKIOQ	DMKIOS	DMKMNI	DMKMNT	DMKNES
		DMKQVM	DMKSCN	DMKSSP	DMKVCH	DMKVRS					
ARIODC	000012	DMKDID	DMKDSP	DMKLNK	DMKLOJ	DMKMNL	DMKQVM	DMKSCN	DMKSSS	DMKSSV	DMKVRS
ARIODV	000060	DMKACR	DMKACS	DMKALO	DMKCCH	DMKCFS	DMKCFU	DMKCKD	DMKCKF	DMKCKH	DMKCKM
		DMKCKN	DMKCKV	DMKCPI	DMKCPJ	DMKCPP	DMKCPZ	DMKCSO	DMKDID	DMKDIF	DMKDSP
		DMKGRG	DMKLNK	DMKLOJ	DMKMNI	DMKMNL	DMKMNT	DMKMOO	DMKNES	DMKNET	DMKPGU
		DMKQVM	DMKSCN	DMKSPS	DMKSSP	DMKSSS	DMKSSV	DMKVCH	DMKVDA	DMKVDG	DMKVDT
		DMKVRS	DMKWRM								
ARIOPR	000006	DMKCKF	DMKCSB	DMKCSF	DMKCSO	DMKSPL					
ARIOPU	000007	DMKCKF	DMKCSB	DMKCSF	DMKCSO	DMKSPL					
ARIORD	000005	DMKCKF	DMKCSB	DMKCSO							
ARIOUC	000007	DMKCAC	DMKCAO	DMKCPZ	DMKQVM	DMKVRS					
ARPCHK1	000002	DMKSCH									
ARPCHK2	000002	DMKSCH									
ARSPAC	000004	DMKACO									
ARSPPR	000031	DMKAPS	DMKAPU	DMKAPV	DMKCKP	DMKCKR	DMKCPI	DMKCQH	DMKCSQ	DMKCST	DMKCSU
		DMKCSV	DMKCSX	DMKCSY	DMKSFB	DMKSPL	DMKSPS	DMKSPT	DMKUSP	DMKWRM	DMKXAB
ARSPPU	000017	DMKACO	DMKCKR	DMKCQH	DMKCSQ	DMKCSU	DMKCSV	DMKCSX	DMKSFB	DMKSPL	DMKSPS
		DMKSPT	DMKUSP	DMKWRM							
ARSPRD	000033	DMKCSQ	DMKCST	DMKCSU	DMKCSV	DMKCSX	DMKCSY	DMKDRD	DMKDRE	DMKSPK	DMKSPL
		DMKVSD									
ARSPTA	000003	DMKSPS									
ASCHN	000011	DMKCFG	DMKCFS	DMKCKM	DMKCPP	DMKCQY					
ASFBACO	000004	DMKCKF	DMKCPP								
ASLOGON	000005	DMKCFS	DMKLOH								
ASOFF	000005	DMKCFS	DMKDEG								
ASON	000004	DMKCFS									
ASVCLIST	000003	DMKCPI	DMKSAD								

LABEL	COUNT	REFERENCES									
ASYNELOG	000001	DMKCPJ									
ASYNFLOG	000001	DMKCPJ									
ASYSIUCV	000026	DMKIUA	DMKIUC	DMKIUE	DMKIUJ	DMKIUL	DMKIUN	DMKIUP			
ASYSLC	000032	DMKACO	DMKBLD	DMKCFT	DMKCFU	DMKCFV	DMKCKF	DMKLOC	DMKLOJ	DMKLOM	DMKUDR
		DMKUDU	DMKUSQ	DMKVCT	DMKVCW						
ASYSOP	000046	DMKCAC	DMKCKF	DMKCKT	DMKCLK	DMKCPI	DMKCPJ	DMKCPS	DMKDAD	DMKDAS	DMKDAU
		DMKERM	DMKEXT	DMKGRF	DMKJRL	DMKLOH	DMKMIA	DMKMNT	DMKMSG	DMKMSW	DMKOPE
		DMKQCN	DMKQVM	DMKSAD	DMKSPS	DMKUSQ	DMKVCH				
ASYSVM	000495	DMKACO	DMKACR	DMKACS	DMKALO	DMKAPI	DMKATS	DMKBLD	DMKCCH	DMKCFC	DMKCFF
		DMKCFG	DMKCFH	DMKCFM	DMKCFO	DMKCFP	DMKCFQ	DMKCFR	DMKCFS	DMKCFU	DMKCKF
		DMKCKV	DMKCNS	DMKCPB	DMKCPI	DMKCPJ	DMKCPM	DMKCPO	DMKCPP	DMKCPS	DMKCPU
		DMKCPV	DMKCPW	DMKCPX	DMKCPY	DMKCPZ	DMKCQP	DMKCQS	DMKCQT	DMKCQY	DMKCRM
		DMKCSC	DMKCSF	DMKCSO	DMKDAD	DMKDAS	DMKDAU	DMKDIB	DMKDID	DMKDIF	DMKDRD
		DMKDRE	DMKDSB	DMKDSP	DMKERM	DMKEXT	DMKFRE	DMKGRA	DMKGRC	DMKGRF	DMKGRT
		DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKIDR	DMKIDU	DMKIOE
		DMKIOG	DMKIOQ	DMKIOS	DMKIOT	DMKIUA	DMKIUJ	DMKIUP	DMKJRL	DMKLNK	DMKLOH
		DMKLOJ	DMKLOK	DMKMCC	DMKMCH	DMKMCT	DMKMHC	DMKMIA	DMKMID	DMKMNI	DMKMNJ
		DMKMNL	DMKMON	DMKMOO	DMKMSG	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKOPE	DMKPAG
		DMKPGM	DMKPGS	DMKPGT	DMKPGU	DMKPMA	DMKPST	DMKPTR	DMKPTS	DMKPTT	DMKQVM
		DMKRGA	DMKRGB	DMKRGC	DMKRNH	DMKRSP	DMKRST	DMKSCH	DMKSCN	DMKSEG	DMKSEL
		DMKSEP	DMKSNC	DMKSPK	DMKSPL	DMKSPM	DMKSPR	DMKSPS	DMKSPT	DMKSRM	DMKSSS
		DMKSST	DMKSSU	DMKSTA	DMKSTP	DMKSTR	DMKSVC	DMKSWA	DMKSWM	DMKTCS	DMKTCT
		DMKTHI	DMKTOD	DMKTRP	DMKTRQ	DMKTRT	DMKTRU	DMKTTX	DMKUDR	DMKUDU	DMKUSQ
		DMKVAT	DMKVBM	DMKVCH	DMKVCW	DMKVCX	DMKVDA	DMKVDC	DMKVDD	DMKVDE	DMKVDR
		DMKVFC	DMKVFD	DMKVFE	DMKVFR	DMKVFS	DMKVMA	DMKVMC	DMKVME	DMKVRR	DMKVSD
		DMKVSP	DMKVSQ	DMKVSR	DMKVSX	DMKWRM	DMKWRN	DMKXAB			
ATMRSN	000007	DMKCPP	DMKMOO	DMKSCH	DMKTMR						
ATTN	000089	DMKCFC	DMKCFM	DMKCKH	DMKCNS	DMKCPB	DMKDDR	DMKDIR	DMKDMP	DMKDSP	DMKFMT
		DMKGRF	DMKHPU	DMKIOS	DMKIOT	DMKNLD	DMKNLE	DMKRGA	DMKRGB	DMKRGC	DMKRNH
		DMKRSE	DMKRSP	DMKSSP	DMKSSS	DMKVCA	DMKVCB	DMKVCN	DMKVCP	DMKVIO	DMKVMI
		DMKVSI									
ATTNHIT	000002	DMKPET									
ATTPSA	000006	DMKDMQ									
ATTRPRHI	000008	DMKBOX									
ATTRSKIP	000002	DMKBOX									
ATTR457	000002	DMKBOX									
ATTR7	000006	DMKBOX									
AUTGO	000013	DMKENT	DMKMCC	DMKMCD	DMKMID	DMKMNI	DMKMNJ	DMKOPE			
AUTOIPL	000003	DMKCKH									
AUTOSPL	000005	DMKMCD	DMKMIA	DMKMNJ							
AVMREAL	000160	DMKBLD	DMKCCH	DMKCCT	DMKCDS	DMKCFD	DMKCFF	DMKCFG	DMKCFP	DMKCFQ	DMKCFS
		DMKCFV	DMKCKD	DMKCKH	DMKCKM	DMKCKP	DMKCPJ	DMKCPO	DMKCPP	DMKCPU	DMKCPY
		DMKCQP	DMKDMP	DMKDMQ	DMKDSP	DMKEXT	DMKFPS	DMKHVD	DMKHVE	DMKIOQ	DMKIOT
		DMKLOH	DMKLOJ	DMKMCC	DMKMCH	DMKMCT	DMKMHC	DMKMHV	DMKMNI	DMKMPO	DMKOPR
		DMKPGS	DMKPMA	DMKPRG	DMKPRV	DMKPRW	DMKPTR	DMKPTS	DMKQVM	DMKRPA	DMKSAV
		DMKSCH	DMKSPM	DMKTRA	DMKUNT	DMKVIO	DMKVRR	DMKVRS	DMKVSI	DMKVSJ	DMKVSR
BADCCW	000001	DMKCCD									
BADHEDNO	000002	DMKCCD									
BALRSAVE	000262	DMKAPI	DMKCFM	DMKCKD	DMKCKH	DMKCNT	DMKCPI	DMKCPV	DMKCSO	DMKCVT	DMKDDR
		DMKDGD	DMKDIA	DMKDIB	DMKDMP	DMKFRE	DMKFRT	DMKGRC	DMKGRH	DMKGRT	DMKIOC
		DMKIOJ	DMKIUE	DMKIUS	DMKPGT	DMKPGU	DMKPTR	DMKPTT	DMKQCO	DMKQCQ	DMKQVM
		DMKRET	DMKRGC	DMKRGD	DMKRGE	DMKRNH	DMKSCH	DMKSCN	DMKSTA	DMKUNT	DMKVAT
		DMKVCA	DMKVFR	DMKVIO	DMKVMA	DMKVRR	DMKVRS	DMKVSC			
BALRSAV2	000006	DMKAPI	DMKCPI	DMKSCH							
BALR0	000035	DMKDGD	DMKGRT	DMKIDU	DMKIOS	DMKIOT	DMKPGT	DMKPRW	DMKPSA	DMKPTT	DMKSCN
		DMKSEG	DMKVMA								
BALR1	000043	DMKAPI	DMKCKD	DMKCPM	DMKCVT	DMKDIF	DMKIDU	DMKMNT	DMKPGT	DMKPGU	DMKPSA
		DMKSCN	DMKVDS								

LABEL	COUNT	REFERENCES									
BALR10	000001	DMKGRC									
BALR11	000004	DMKQCO	DMKSCH								
BALR12	000001	DMKVAT									
BALR13	000001	DMKVAT									
BALR14	000010	DMKDGD	DMKTMR	DMKUNT	DMKVAT	DMKVCA					
BALR15	000017	DMKIUE	DMKPGT	DMKPSA	DMKPTR	DMKVCA	DMKVFR				
BALR2	000041	DMKCVT	DMKDMP	DMKDSP	DMKIOT	DMKPSA	DMKPTR	DMKSCN	DMKTMR	DMKVCA	DMKVDS
		DMKVMA									
BALR3	000004	DMKCNT	DMKMNT	DMKVCA							
BALR4	000010	DMKMNT	DMKQCQ	DMKVAT	DMKVAU						
BALR5	000004	DMKQCQ	DMKVAT								
BALR6	000005	DMKCNT	DMKGRT	DMKRGE							
BALR7	000002	DMKPSA									
BALR8	000007	DMKPGT	DMKSCN	DMKVSC							
BALR9	000006	DMKCNS	DMKDSP	DMKVCA							
BCTWAIT	000012	DMKACR	DMKACS	DMKCKD	DMKCPI	DMKCPM	DMKMNT	DMKOPR			
BFFCCPD	000013	DMKCKF	DMKTRT	DMKTRU							
BFFCLOSE	000008	DMKTRT									
BFFENTRY	000012	DMKCKF	DMKTRT	DMKTRU							
BFFFSIZE	000009	DMKCKF	DMKTRT	DMKTRU							
BFFPEND	000004	DMKTRT									
BFFREAL	000014	DMKCKF	DMKTRT	DMKTRU							
BFFSTAT	000014	DMKTRT									
BFFSTOP	000005	DMKTRT									
BFFVIRT	000003	DMKTRT	DMKTRU								
BIOBIRS	000007	DMKBIO	DMKCFQ								
BIOBKSZ	000005	DMKBIO									
BIOBLKND	000003	DMKBIO									
BIOBLKST	000002	DMKBIO									
BIOBLOK	000014	DMKBIO	DMKCFP	DMKCFQ							
BIODEVDA	000003	DMKBIO									
BIODEVDS	000002	DMKBIO									
BIOFBA	000003	DMKBIO									
BIOFLAG2	000002	DMKBIO									
BIONEXT	000009	DMKBIO	DMKCFP	DMKCFQ							
BIOOFFCP	000003	DMKBIO									
BIOOFFST	000004	DMKBIO									
BIOPARMA	000001	DMKBIO									
BIOPARML	000006	DMKBIO									
BIOPARMU	000001	DMKBIO									
BIOPATH	000006	DMKBIO									
BIORC	000002	DMKBIO									
BIOREAD	000002	DMKBIO									
BIORESET	000004	DMKBIO	DMKCFP	DMKCFQ							
BIORPS	000002	DMKBIO									
BIOSECDS	000001	DMKBIO									
BIOSIZE	000003	DMKBIO									
BIOSTAT1	000013	DMKBIO	DMKCFP	DMKCFQ							
BIOSVRD	000003	DMKBIO									
BIOVBLKS	000001	DMKBIO									
BIOVDEV	000001	DMKBIO									
BIOVDEVA	000002	DMKBIO	DMKCFQ								
BIO3344K	000002	DMKBIO									
BIO512BL	000005	DMKBIO									
BIRBIOBL	000002	DMKBIO									
BIRBLOK	000004	DMKBIO									
BIRCCWS	000003	DMKBIO									
BIRCMSRD	000002	DMKBIO									

LABEL	COUNT	REFERENCES									
BIRCMSWR	000004	DMKBIO									
BIREXEND	000001	DMKBIO									
BIREXIND	000003	DMKBIO									
BIREXOFF	000001	DMKBIO									
BIREXRD	000001	DMKBIO									
BIREXTLI	000001	DMKBIO									
BIREXWR	000001	DMKBIO									
BIRIDAL	000002	DMKBIO									
BIRIDAW1	000001	DMKBIO									
BIRIDAW2	000002	DMKBIO									
BIRIDAW3	000002	DMKBIO									
BIRLCBYT	000001	DMKBIO									
BIRLCIND	000003	DMKBIO									
BIRLCOFF	000001	DMKBIO									
BIRLCRD	000001	DMKBIO									
BIRLCWR	000001	DMKBIO									
BIRLOCLI	000001	DMKBIO									
BIRMSGID	000002	DMKBIO									
BIRPAGES	000002	DMKBIO									
BIRPAGE1	000001	DMKBIO									
BIRPAGE2	000001	DMKBIO									
BIRPARML	000001	DMKBIO									
BIRRWCNT	000001	DMKBIO									
BIRRWIDA	000001	DMKBIO									
BIRRWOP	000001	DMKBIO									
BIRRWRD	000001	DMKBIO									
BIRSECAR	000001	DMKBIO									
BIRSECOP	000004	DMKBIO									
BIRSECT	000002	DMKBIO									
BIRSIZE	000003	DMKBIO									
BIRSKCYL	000003	DMKBIO									
BIRSKDA	000001	DMKBIO									
BIRSKDAT	000001	DMKBIO									
BIRSKOP	000002	DMKBIO									
BIRSKTRK	000001	DMKBIO									
BIRSRCYL	000001	DMKBIO									
BIRSRDA	000001	DMKBIO									
BIRSRDAT	000001	DMKBIO									
BIRSROP	000001	DMKBIO									
BIRSRREC	000002	DMKBIO									
BIRSRTRK	000001	DMKBIO									
BIRTICAD	000001	DMKBIO									
BIRTRGCL	000002	DMKBIO									
BIRWCNT	000003	DMKBIO									
BIRWIDA	000001	DMKBIO									
BIRWOP	000002	DMKBIO									
BIRWRD	000002	DMKBIO									
BLANK	000228	DMKAPS	DMKAPU	DMKAPX	DMKAPY	DMKCAC	DMKCAO	DMKCKF	DMKCKM	DMKCPN	DMKCPT
		DMKCQC	DMKCQG	DMKCQH	DMKCQI	DMKCQQ	DMKCQR	DMKCQU	DMKCSO	DMKCSP	DMKCSQ
		DMKCSR	DMKCSU	DMKCSV	DMKCSW	DMKCSX	DMKCSY	DMKDEI	DMKDGD	DMKDIR	DMKDMQ
		DMKFMT	DMKGRI	DMKJRL	DMKLOG	DMKLOJ	DMKLOM	DMKMCD	DMKMCH	DMKMCT	DMKMSG
		DMKNEA	DMKPEQ	DMKQCN	DMKQCO	DMKRGB	DMKRGD	DMKRSP	DMKRST	DMKSAD	DMKSEP
		DMKSND	DMKSPL	DMKSPR	DMKSPS	DMKSSP	DMKTOD	DMKTRM	DMKTRR	DMKURS	DMKUSQ
		DMKVDA	DMKVDB	DMKVDC	DMKVDD	DMKVDF	DMKVDT	DMKVMD			
BLANKLEN	000002	DMKACO	DMKCKF								
BLANKS	000337	DMKACO	DMKACR	DMKAPU	DMKAPX	DMKCAC	DMKCAO	DMKCCW	DMKCDS	DMKCFC	DMKCFD
		DMKCFM	DMKCFO	DMKCFS	DMKCFU	DMKCFV	DMKCFY	DMKCKF	DMKCNS	DMKCPB	DMKCPU
		DMKCPW	DMKCQG	DMKCQH	DMKCQI	DMKCQP	DMKCQQ	DMKCQR	DMKCQT	DMKCQY	DMKCSB

LABEL	COUNT	REFERENCES									
		DMKCSO	DMKCSP	DMKCSQ	DMKCST	DMKCSU	DMKCSV	DMKCSX	DMKDAD	DMKDAS	DMKDAU
		DMKDDR	DMKDEF	DMKDEI	DMKDIA	DMKDIB	DMKDSB	DMKEPS	DMKERM	DMKEXT	DMKGRG
		DMKHVC	DMKHVF	DMKIDR	DMKIOS	DMKIOT	DMKJRL	DMKLDOOE	DMKLNK	DMKLNM	DMKLOG
		DMKLOH	DMKLOJ	DMKLOM	DMKMCC	DMKMCD	DMKMCI	DMKMIA	DMKMNJ	DMKMNT	DMKMON
		DMKMOO	DMKMSG	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKOPE	DMKOVR	DMKPEN
		DMKPEQ	DMKQCP	DMKQVM	DMKREI	DMKRGC	DMKRNH	DMKRPD	DMKSCN	DMKSEP	DMKSFB
		DMKSPK	DMKSPL	DMKSPM	DMKSPS	DMKSPT	DMKSRM	DMKSSS	DMKSST	DMKSSV	DMKTCT
		DMKTEE	DMKTHI	DMKTPE	DMKTRC	DMKTRD	DMKTRP	DMKTRR	DMKUDR	DMKUDU	DMKURS
		DMKUSQ	DMKVCB	DMKVCN	DMKVCP	DMKVCT	DMKVCU	DMKVDA	DMKVDC	DMKVDD	DMKVDE
		DMKVDF	DMKVDR	DMKVDT	DMKVMD	DMKVSD	DMKVSQ	DMKVSX			
BLKLEN	000008	DMKRGD	DMKSSP								
BLKMPX	000003	DMKCPI	DMKVSI								
BLKSIZE	000001	DMKBIO									
BOXBLOK	000007	DMKCFM	DMKGRT	DMKSEP	DMKVCX						
BOXINLNS	000007	DMKGRT	DMKVCX								
BOXINWTH	000004	DMKGRT	DMKVCX								
BOXLINES	000004	DMKGRT	DMKVCX								
BOXLOGO	000004	DMKGRT	DMKSEP	DMKVCX							
BOXWIDTH	000002	DMKGRT	DMKVCX								
BPFRESV	000002	DMKFRE									
BREAKSUP	000007	DMKDMQ									
BRING	000719	DMKACO	DMKAPT	DMKATS	DMKBIO	DMKCCH	DMKCCW	DMKCDB	DMKCDM	DMKCDS	DMKCFC
		DMKCFD	DMKCFF	DMKCFG	DMKCFH	DMKCFM	DMKCFS	DMKCFU	DMKCFV	DMKCKR	DMKCKS
		DMKCKV	DMKCNS	DMKCPB	DMKCPI	DMKCPM	DMKCPO	DMKCPP	DMKCPS	DMKCPU	DMKCPW
		DMKCPY	DMKCQI	DMKCQS	DMKCQY	DMKCSC	DMKCSO	DMKCST	DMKCSV	DMKCSW	DMKCSY
		DMKDGD	DMKDGF	DMKDRD	DMKDRE	DMKDSP	DMKERM	DMKGIO	DMKGRA	DMKGRC	DMKGRF
		DMKGRT	DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKIDU	DMKIOF
		DMKIOG	DMKIOH	DMKIOT	DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUL	DMKLNK	DMKMCC
		DMKMHV	DMKMIA	DMKMNJ	DMKNLD	DMKNLE	DMKOPE	DMKPEI	DMKPER	DMKPMA	DMKPRG
		DMKPRV	DMKPRW	DMKPTR	DMKQCN	DMKQCO	DMKRGC	DMKRPA	DMKRPD	DMKRSP	DMKRSQ
		DMKRST	DMKSEG	DMKSEP	DMKSFB	DMKSNC	DMKSPK	DMKSPL	DMKSPS	DMKSPT	DMKSRM
		DMKSSS	DMKSST	DMKSSV	DMKSVC	DMKSVD	DMKSWA	DMKTCS	DMKTCT	DMKTMR	DMKTOD
		DMKTRC	DMKTRD	DMKTRK	DMKTRP	DMKTRT	DMKTRU	DMKTRX	DMKTTX	DMKUDR	DMKUDU
		DMKVAT	DMKVAU	DMKVBM	DMKVCH	DMKVCN	DMKVCX	DMKVDD	DMKVDR	DMKVER	DMKVIO
		DMKVMC	DMKVMD	DMKVME	DMKVSE	DMKVSG	DMKVSI	DMKVSJ	DMKVSP	DMKVSQ	DMKVSX
		DMKWRM	DMKWRN	DMKXAB	DMKXAD						
BSCAUSER	000004	DMKRGA	DMKRGB								
BSCBLOK	000009	DMKBSC	DMKNET	DMKRGA	DMKRGB	DMKRGC	DMKRGD	DMKRGE			
BSCCNT	000015	DMKRGA	DMKRGB	DMKRGC							
BSCCOPY	000009	DMKRGA	DMKRGC								
BSCDCNT	000008	DMKRGA	DMKRGC								
BSCECCW1	000005	DMKRGA									
BSCECCW2	000005	DMKRGA									
BSCENQ	000008	DMKRGA	DMKRGC								
BSCETB	000006	DMKRGA	DMKRGC								
BSCFLAG	000052	DMKBSC	DMKRGA	DMKRGB	DMKRGC						
BSCFLAG1	000053	DMKBSC	DMKNET	DMKRGA	DMKRGB	DMKRGC					
BSCFORCE	000003	DMKRGA									
BSCHALT	000005	DMKBSC	DMKNET	DMKRGA	DMKRGB						
BSCIGN	000005	DMKRGA	DMKRGC								
BSCINBID	000004	DMKRGA	DMKRGB	DMKRGC							
BSCINDEX	000011	DMKRGA	DMKRGB	DMKRGC							
BSCLOG	000005	DMKRGA									
BSCOPIED	000005	DMKRGA	DMKRGC								
BSCPA1	000003	DMKRGA	DMKRGC								
BSCPCCW1	000009	DMKBSC	DMKRGA	DMKRGB							
BSCPCCW2	000006	DMKBSC	DMKRGA	DMKRGB							
BSCPCCW3	000003	DMKRGC									

LABEL	COUNT	REFERENCES									
BSCPCCW4	000006	DMKRGA	DMKRGB	DMKRGC							
BSCRCVD	000012	DMKBSC	DMKRGA	DMKRGB							
BSCREAD	000042	DMKBSC	DMKRGA	DMKRGB	DMKRGC						
BSCREGEN	000003	DMKRGA									
BSCRESP	000043	DMKBSC	DMKRGA	DMKRGB	DMKRGC						
BSCRPTR	000012	DMKRGA	DMKRGC								
BSCRROBN	000005	DMKRGB									
BSCRSTRT	000005	DMKBSC	DMKRGA	DMKRGC							
BSCRVI	000007	DMKBSC	DMKRGA	DMKRGC							
BSCSCAN	000004	DMKRGA	DMKRGB								
BSCSCCW1	000006	DMKRGB	DMKRGC								
BSCSCCW2	000007	DMKRGA	DMKRGB	DMKRGC							
BSCSCCW3	000004	DMKRGA	DMKRGB								
BSCSEL	000006	DMKRGB									
BSCSEND	000012	DMKRGA	DMKRGC								
BSCSENSE	000039	DMKRGA	DMKRGC	DMKRGE							
BSCSHUT	000005	DMKBSC	DMKNET	DMKRGB							
BSCSIZE	000002	DMKRGA	DMKRGB								
BSCSIZE1	000007	DMKRGA	DMKRGB	DMKRGC							
BSCSIZE2	000001	DMKRGB									
BSCSPTR	000035	DMKRGA	DMKRGB	DMKRGC							
BSCTMRQ	000006	DMKRGA									
BSCTSTRQ	000006	DMKRGA	DMKRGC								
BSCUCOPY	000006	DMKRGA	DMKRGC								
BSCUP	000025	DMKRGA	DMKRGB	DMKRGC							
BUFAPL	000003	DMKGRF									
BUFCNT	000089	DMKALG	DMKAPX	DMKAPY	DMKCFM	DMKCFU	DMKCFY	DMKCPT	DMKCST	DMKERM	DMKGRG
		DMKGRT	DMKHVE	DMKLOH	DMKMIA	DMKOPE	DMKPET	DMKQCO	DMKREI	DMKRGA	DMKRGC
		DMKRST	DMKSND	DMKSPC	DMKTRP	DMKUDU	DMKVCN	DMKVCR	DMKVCU	DMKVMD	
BUFFCV	000003	DMKCKF	DMKCKH								
BUFFDLM	000003	DMKCKF	DMKCKH								
BUFFER	000198	DMKALG	DMKAPX	DMKAPY	DMKCDM	DMKCFC	DMKCFG	DMKCFM	DMKCFO	DMKCFS	DMKCFU
		DMKCFV	DMKCFW	DMKCFY	DMKCPT	DMKCSB	DMKCSO	DMKCSP	DMKCSQ	DMKCST	DMKCSU
		DMKCSV	DMKCSX	DMKEPS	DMKERM	DMKGRF	DMKGRG	DMKGRT	DMKHVE	DMKLOH	DMKMIA
		DMKMSG	DMKOPE	DMKPET	DMKQCO	DMKREI	DMKRGA	DMKRGC	DMKRST	DMKSCN	DMKSND
		DMKSPC	DMKTRP	DMKUDU	DMKVCR	DMKVCU	DMKVDC	DMKVMD			
BUFFOPLG	000003	DMKCKF									
BUFFSTAT	000002	DMKCKH									
BUFFVER	000005	DMKCKH	DMKWRM								
BUFIN	000037	DMKAPX	DMKHVE	DMKLOH	DMKOPE	DMKREI	DMKSND	DMKVCN	DMKVFE		
BUFINLTH	000039	DMKAPX	DMKCFM	DMKCFY	DMKEPS	DMKERM	DMKGRF	DMKGRG	DMKGRH	DMKGRT	DMKHVC
		DMKHVE	DMKLOH	DMKREI	DMKRGC	DMKTTX	DMKTTY	DMKVCN	DMKVCR	DMKVCU	
BUFNORM	000005	DMKGRF									
BUFNXT	000099	DMKALG	DMKAPX	DMKAPY	DMKCDM	DMKCFC	DMKCFG	DMKCFM	DMKCFO	DMKCFS	DMKCFU
		DMKCFV	DMKCFY	DMKCPT	DMKCSO	DMKCST	DMKCSU	DMKCSV	DMKCSX	DMKEPS	DMKGRT
		DMKHVE	DMKLOH	DMKMIA	DMKMSG	DMKOPE	DMKPET	DMKREI	DMKRST	DMKSCN	DMKSND
		DMKSPC	DMKTRP	DMKUDU	DMKVCN	DMKVDC	DMKVFE	DMKVMD			
BUFSIZE	000086	DMKALG	DMKAPX	DMKCFM	DMKCNS	DMKCPJ	DMKEPS	DMKERM	DMKGRF	DMKGRG	DMKGRI
		DMKGRT	DMKHPT	DMKHVC	DMKHVE	DMKLOG	DMKLOH	DMKMIA	DMKOPE	DMKPET	DMKQCO
		DMKQVM	DMKREI	DMKRGA	DMKRGC	DMKRST	DMKSND	DMKTTX	DMKTTY	DMKUDU	DMKVCN
		DMKVCP	DMKVCR	DMKVCU	DMKVCV						
BUSOUT	000011	DMKDAD	DMKDDR	DMKRNH	DMKRSE						
BUSY	000112	DMKACS	DMKCKH	DMKCNS	DMKCPM	DMKCPZ	DMKCQT	DMKDDR	DMKDIR	DMKDMP	DMKDMQ
		DMKDSP	DMKEXT	DMKFMT	DMKIOQ	DMKIOS	DMKIOT	DMKMNT	DMKOPE	DMKOPR	DMKRNH
		DMKSAD	DMKSAV	DMKSSP	DMKVCA	DMKVCN	DMKVIO	DMKVMI	DMKVSI	DMKVSJ	
BYTBL	000003	DMKFRE									
CACHBLOK	000011	DMKMNL									
CACHCPXP	000006	DMKMNL									

LABEL	COUNT	REFERENCES									
CACHCUU1	000003	DMKMNL									
CACHCUU2	000005	DMKMNL									
CACHDPT	000005	DMKMNL									
CACHERRC	000006	DMKMNL									
CACHFCNT	000007	DMKMNL									
CACHICCW	000010	DMKMNL									
CACHIN1	000004	DMKMNL									
CACHIN2	000004	DMKMNL									
CACHIOBP	000004	DMKMNL									
CACHISSS	000002	DMKMNL									
CACHKX	000006	DMKMNL									
CACHNTDF	000001	DMKMNL									
CACHNTIM	000002	DMKMNL									
CACHNTMI	000001	DMKMNL									
CACHPDP	000003	DMKMNL									
CACHRCNT	000002	DMKMNL									
CACHRDVP	000004	DMKMNL									
CACHRPT	000008	DMKMNL									
CACHRSD2	000007	DMKMNL									
CACHSIZE	000003	DMKMNL									
CACHSTA1	000038	DMKMNL									
CACHSTOD	000002	DMKMNL									
CACHTRQP	000004	DMKMNL									
CACHWR1	000003	DMKMNL									
CACHWR2	000004	DMKMNL									
CACHWSD1	000007	DMKMNL									
CACHWSD2	000004	DMKMNL									
CACTDEV	000002	DMKRNH									
CACTLIN	000002	DMKNET	DMKRNH								
CACTLTR	000002	DMKNES	DMKRNH								
CALLCVT	000005	DMKACO	DMKCKF								
CALLER	000013	DMKIOQ	DMKIOS	DMKTEF							
CARDIN	000020	DMKDDR									
CAW	000141	DMKACS	DMKCKD	DMKCKH	DMKCKN	DMKCKP	DMKCNS	DMKCPI	DMKCPM	DMKCPZ	DMKDDR
		DMKDIR	DMKDMP	DMKDMQ	DMKFMT	DMKHVE	DMKIOS	DMKIOT	DMKLDOOE	DMKMNT	DMKOPE
		DMKOPR	DMKSAD	DMKSAV	DMKSSP	DMKTRD	DMKVMI	DMKVRR	DMKVRS	DMKVSI	DMKVSJ
CC	002329	DMKACO	DMKBSC	DMKCCD	DMKCCF	DMKCCO	DMKCCS	DMKCCT	DMKCCW	DMKCFQ	DMKCKD
		DMKCKH	DMKCKN	DMKCKP	DMKCNS	DMKCPM	DMKCPZ	DMKCSB	DMKCSC	DMKDAD	DMKDAS
		DMKDAU	DMKDDR	DMKDEX	DMKDGD	DMKDGF	DMKDIB	DMKDIR	DMKDMP	DMKDMQ	DMKDSB
		DMKFMT	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRH	DMKHPS	DMKIOS	DMKLDOOE	DMKMCC
		DMKMNI	DMKMNL	DMKMNT	DMKMON	DMKNLD	DMKNLE	DMKOPR	DMKOVR	DMKPAG	DMKPAH
		DMKPIA	DMKPIB	DMKRGA	DMKRGB	DMKRGC	DMKRGD	DMKRNH	DMKRSE	DMKRSF	DMKRSP
		DMKRST	DMKSAD	DMKSAV	DMKSEP	DMKSPK	DMKSPS	DMKSPT	DMKSSP	DMKSTP	DMKTAQ
		DMKTCS	DMKTCT	DMKTEF	DMKTPE	DMKTRK	DMKTTX	DMKTTY	DMKTTZ	DMKUCB	DMKUCC
		DMKUCS	DMKUDR	DMKVCA	DMKVCN	DMKVCV	DMKVDE	DMKVDR	DMKVDT	DMKVMI	DMKVSC
		DMKVSI	DMKVSP	DMKVSQ	DMKVST	DMKVSX	DMKZTD				
CCC	000092	DMKACS	DMKBSC	DMKCCH	DMKCNS	DMKCPI	DMKCPM	DMKCPW	DMKCPZ	DMKCQT	DMKDAD
		DMKDAS	DMKDAU	DMKDDR	DMKDMQ	DMKDSB	DMKDSP	DMKEIG	DMKGIO	DMKGRF	DMKHPS
		DMKHPT	DMKHVC	DMKIOE	DMKIOS	DMKIOT	DMKMNT	DMKMSW	DMKRSE	DMKRSF	DMKRSP
		DMKSAD	DMKSEV	DMKTAP	DMKTAQ	DMKTPE	DMKUNT	DMKVIO	DMKVSJ		
CCCPUID	000003	DMKCCH									
CCDESMD	000003	DMKDIF	DMKRNH								
CCDEVTYP	000001	DMKCCH									
CCHADDR	000002	DMKCCH									
CCHANID	000008	DMKCCH									
CCHCAV	000001	DMKCCH									
CCHCHCUA	000001	DMKCCH									
CCHCHNL	000012	DMKSEV	DMKSIX								

LABEL	COUNT	REFERENCES							
CCHCLOGL	000003	DMKCCH							
CCHCMDV	000010	DMKEIG	DMKSEV	DMKSIX					
CCHCNTB	000005	DMKSEV	DMKSIX						
CCHCPEX	000002	DMKCCH							
CCHCPU	000003	DMKSEV	DMKSIX						
CCHCUA	000002	DMKCCH							
CCHDAV	000010	DMKEIG	DMKSEV	DMKSIX					
CCHDI	000003	DMKEIG	DMKSEV	DMKSIX					
CCHHIO	000002	DMKCCH							
CCHINTB	000001	DMKCCH							
CCHINTFC	000007	DMKSEV	DMKSIX						
CCHIOH	000001	DMKCCH							
CCHLOG45	000002	DMKCCH							
CCHLOG60	000001	DMKSIX							
CCHLOG70	000001	DMKSEV							
CCHLOG80	000002	DMKCCH	DMKEIG						
CCHLOG81	000001	DMKCCH							
CCHRCV	000003	DMKCCH	DMKEIG						
CCHREC	000005	DMKCCH	DMKEIG	DMKSEV	DMKSIX				
CCHSIOB	000002	DMKCCH							
CCHSIZE	000002	DMKCCH							
CCHSIZE1	000001	DMKCCH							
CCHSNSB	000001	DMKCCH							
CCHSTG	000004	DMKSEV	DMKSIX						
CCHTIO	000002	DMKCCH							
CCHUSV	000005	DMKEIG	DMKSEV	DMKSIX					
CCMASK	000016	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS
CCPADDR	000001	DMKSNC							
CCPARM	000004	DMKNLD	DMKSNC						
CCPENTRY	000001	DMKNLD							
CCPMAXID	000001	DMKNLD							
CCPNAME	000003	DMKNLD	DMKSNC						
CCPPSIZE	000005	DMKNLD	DMKSNC						
CCPRESID	000002	DMKNLD							
CCPROGID	000003	DMKCCH							
CCPRSTAT	000001	DMKNLD							
CCPRSTEP	000001	DMKNLD							
CCPRSTYP	000001	DMKNLD							
CCPSIZE	000003	DMKNLD	DMKSNC						
CCPTNCP	000001	DMKNLD							
CCPTPEP	000001	DMKNLD							
CCPTYPE	000002	DMKNLD							
CCRECTYP	000002	DMKCCH							
CCSBUFCK	000001	DMKCCF							
CCSCKDON	000001	DMKCCF							
CCSHNSEN	000002	DMKCCF	DMKCCO						
CCSSENSE	000004	DMKCCD	DMKCCO	DMKCCT	DMKCCW				
CCSW1	000004	DMKCCH							
CCSW4	000008	DMKCCH							
CCS1PASS	000004	DMKCCH							
CCS4DGRD	000003	DMKACR	DMKACS	DMKMCH					
CCS4HARD	000005	DMKACR	DMKACS	DMKCCH	DMKMCH				
CCS4SOFT	000009	DMKACR	DMKACS	DMKCCH					
CCTCLCC	000002	DMKIUN	DMKIUP						
CCTCLPC	000002	DMKIUN	DMKIUP						
CCTCLPQ	000002	DMKIUN	DMKIUP						
CCTCLPR	000002	DMKIUN	DMKIUP						
CCTCLPS	000002	DMKIUN	DMKIUP						

LABEL	COUNT	REFERENCES									
CCTFLAG1	000014	DMKDSP	DMKIUA	DMKIUB	DMKIUN	DMKIUP					
CCTFLAG2	000029	DMKIUA	DMKIUB	DMKIUC	DMKIUJ	DMKIUN	DMKIUP	DMKIUS			
CCTFLAG3	000003	DMKIUB	DMKIUN	DMKIUP							
CCTFLCNT	000012	DMKDSP	DMKIUA	DMKIUC	DMKIUJ	DMKIUN	DMKIUS				
CCTICTRL	000005	DMKIUB	DMKIUN	DMKIUP							
CCTMSGCT	000009	DMKIUA	DMKIUJ	DMKIUN	DMKIUS						
CCTMXPDE	000007	DMKIUG	DMKIUJ								
CCTMXPDS	000004	DMKIUG	DMKIUJ								
CCTMXPID	000008	DMKIUA	DMKIUJ	DMKIUP							
CCTPDSEG	000018	DMKIUA	DMKIUG	DMKIUJ	DMKIUP	DMKIUS					
CCTPDSLO	000003	DMKIUA	DMKIUG								
CCTPNDCT	000008	DMKIUB	DMKIUC	DMKIUJ	DMKIUP						
CCTPNDRN	000003	DMKIUB									
CCTPNDRP	000004	DMKIUB									
CCTPNDSN	000005	DMKIUB	DMKIUS								
CCTPNDSP	000003	DMKIUB									
CCTRCVHD	000011	DMKIUA	DMKIUG	DMKIUJ	DMKIUL	DMKIUS					
CCTRCVTL	000009	DMKIUA	DMKIUG	DMKIUL	DMKIUS						
CCTRPYHD	000015	DMKIUA	DMKIUB	DMKIUG	DMKIUJ	DMKIUL	DMKIUN				
CCTRPYN	000007	DMKDSP	DMKIUA	DMKIUB	DMKIUN	DMKIUP					
CCTRPYP	000007	DMKDSP	DMKIUA	DMKIUB	DMKIUN	DMKIUP					
CCTRPYPR	000018	DMKIUA	DMKIUB	DMKIUG	DMKIUL						
CCTRPYTL	000016	DMKIUA	DMKIUB	DMKIUG	DMKIUL						
CCTSNDHD	000023	DMKIUA	DMKIUB	DMKIUE	DMKIUG	DMKIUJ	DMKIUN	DMKIUS			
CCTSNDN	000006	DMKIUA	DMKIUB	DMKIUN	DMKIUP						
CCTSNDP	000006	DMKIUA	DMKIUB	DMKIUN	DMKIUP						
CCTSNDPR	000022	DMKIUA	DMKIUB	DMKIUE	DMKIUG	DMKIUN	DMKIUS				
CCTSNDTL	000019	DMKIUA	DMKIUB	DMKIUE	DMKIUG	DMKIUN	DMKIUS				
CCWBD2	000001	DMKCCD									
CCWBD4	000002	DMKCCD	DMKCCO								
CCWCLEAR	000005	DMKCCD	DMKCCF	DMKCCO	DMKCCS	DMKCCT					
CCWCTL	000007	DMKCCF	DMKCCS	DMKCCW							
CCWCTLCM	000003	DMKCCD	DMKCCO	DMKCCS							
CCWFIRST	000003	DMKCCW									
CCWFORC	000004	DMKCCD	DMKCCF	DMKCCO	DMKCCS						
CCWGEN	000005	DMKCCD	DMKCCF	DMKCCO	DMKCCS	DMKCCT					
CCWIDAL1	000001	DMKCCO									
CCWIDASB	000001	DMKCCO									
CCWINV	000005	DMKCCD	DMKCCF	DMKCCO	DMKCCS	DMKCCT					
CCWMAN2	000003	DMKCCD	DMKCCF	DMKCCS							
CCWNOOP	000005	DMKCCD	DMKCCF	DMKCCO	DMKCCS	DMKCCT					
CCWNXT	000002	DMKCCD	DMKCCO								
CCWNX1	000002	DMKCCS									
CCWNX10	000001	DMKCCS									
CCWNX11	000004	DMKCCD	DMKCCF	DMKCCO	DMKCCT						
CCWNX13	000002	DMKCCD	DMKCCO								
CCWNX14	000001	DMKCCO									
CCWNX18	000001	DMKCCS									
CCWNX9	000001	DMKCCO									
CCWSRCH3	000001	DMKCCD									
CCWSUBR	000002	DMKCCW									
CCWTC	000004	DMKCCD	DMKCCF	DMKCCO	DMKCCT						
CCWUSIDA	000004	DMKCCD	DMKCCF	DMKCCO	DMKCCS						
CC0	000041	DMKCKD	DMKCKH	DMKCKM	DMKCKN	DMKCKP	DMKGRI	DMKIDR	DMKIUC	DMKIUJ	DMKIUS
		DMKSAD	DMKSAV	DMKTOD							
CC1	000023	DMKCKD	DMKCKH	DMKCKM	DMKCKN	DMKGRI	DMKIDR	DMKIUJ	DMKIUL	DMKIUS	DMKSAD
		DMKTOD									
CC2	000052	DMKCKD	DMKCKH	DMKCKM	DMKCKP	DMKGRI	DMKIDR	DMKIUA	DMKIUC	DMKIUJ	DMKIUL

LABEL	COUNT	REFERENCES									
		DMKIUP	DMKIUS	DMKSAD	DMKSEG	DMKTOD					
CC3	000048	DMKCKD	DMKCKH	DMKCKM	DMKCKN	DMKCKP	DMKDSP	DMKGRI	DMKIDR	DMKIUC	DMKIUJ
		DMKIUL	DMKIUP	DMKIUS	DMKSAD	DMKSEP	DMKSPT	DMKSRM	DMKTOD	DMKTRR	DMKVSI
		DMKVSJ									
CD	000200	DMKCCD	DMKCCF	DMKCCO	DMKCCS	DMKCCT	DMKCCW	DMKCNS	DMKCSC	DMKDAD	DMKDAS
		DMKDAU	DMKDDR	DMKDGD	DMKDIB	DMKDIR	DMKFMT	DMKGRD	DMKGRF	DMKHPS	DMKHPU
		DMKISM	DMKOPR	DMKPAG	DMKPAH	DMKRGA	DMKRGB	DMKRGC	DMKRGD	DMKRGE	DMKRSP
		DMKRSQ	DMKSAD	DMKSSP	DMKSTP	DMKTAP	DMKTAQ	DMKTPE	DMKTRK	DMKTTX	DMKTTY
		DMKTTZ	DMKUNT	DMKVCA	DMKVCN	DMKVCV	DMKVMI	DMKVSC	DMKVSI	DMKVSP	DMKVSQ
		DMKVSX									
CDC	000078	DMKACS	DMKBSC	DMKCCH	DMKCNS	DMKCQT	DMKDAD	DMKDAS	DMKDAU	DMKDMQ	DMKDSB
		DMKDSP	DMKGRF	DMKHVC	DMKIOE	DMKIOF	DMKIOS	DMKIOT	DMKMSW	DMKNLD	DMKNLE
		DMKRNH	DMKRSE	DMKRSF	DMKRSP	DMKSAD	DMKTAP	DMKTAQ	DMKTPE	DMKUNT	DMKVIO
		DMKVSJ									
CDCONF	000002	DMKCFG	DMKPMA								
CDCTLIN	000001	DMKNET									
CDDED	000003	DMKLNK	DMKPMA	DMKVDA							
CDDEF	000002	DMKDEF	DMKPMA								
CDISPLY	000001	DMKNES									
CDTIC	000011	DMKCCW									
CDVADD	000004	DMKDEF	DMKLNK	DMKVDA							
CE	000128	DMKCFC	DMKCKH	DMKCKN	DMKCKP	DMKCNS	DMKCPZ	DMKDDR	DMKDIB	DMKDID	DMKDIR
		DMKDMP	DMKDMQ	DMKFMT	DMKGRF	DMKHPS	DMKHPT	DMKHVC	DMKIOH	DMKIOS	DMKIOT
		DMKNLD	DMKOPE	DMKOPR	DMKPAH	DMKRGA	DMKRGE	DMKRSE	DMKRSP	DMKRST	DMKSAD
		DMKSAV	DMKSSP	DMKVCA	DMKVCB	DMKVCN	DMKVIO	DMKVMI	DMKVRS	DMKVSI	DMKVSJ
		DMKVSP	DMKVSX								
CFSTOP	000009	DMKCPS	DMKMCD	DMKMIA	DMKMNI	DMKMON					
CHANID	000006	DMKCPN	DMKIOG	DMKPRV							
CHC	000021	DMKBSC	DMKCNS	DMKDAD	DMKGRF	DMKHVC	DMKIOS	DMKIOT	DMKRNH	DMKRSE	DMKSAD
		DMKTAP	DMKTAQ	DMKTPE	DMKUNT						
CHECKVF	000002	DMKACO	DMKCKF								
CHEKISAM	000004	DMKCCD	DMKCCO	DMKCCW							
CHG	000004	DMKSEL									
CHGRDV	000004	DMKCSB	DMKCSO								
CHGREGS	000002	DMKTMR									
CHGSFB	000041	DMKACO	DMKAPV	DMKCKS	DMKCSQ	DMKCSV	DMKCSW	DMKCSY	DMKDMP	DMKRSP	DMKSPL
		DMKVSD	DMKXAB								
CHGSHQ	000008	DMKCSQ	DMKWRN								
CHNGMSG	000002	DMKTOD									
CIRCLEC	000001	DMKCCT									
CKCMASK	000004	DMKAPI	DMKCPI	DMKPMA							
CKPBITS	000003	DMKRNH									
CKPBKSZ	000001	DMKRNH									
CKPBLOK	000004	DMKRNH	DMKWRM								
CKPLD	000003	DMKCKP									
CKPLOAD	000356	DMKCKD	DMKCKF	DMKCKH	DMKCKM	DMKCKN	DMKCKP	DMKCKW	DMKERP	DMKSAV	
CKPNAME	000003	DMKRNH	DMKWRM								
CKPRMAX	000002	DMKRNH	DMKWRM								
CKPSIZE	000003	DMKRNH	DMKWRM								
CKPTLIST	000004	DMKCKF	DMKCPI	DMKRSP							
CL	000098	DMKCKF	DMKCKH	DMKCQU	DMKDAD	DMKDAS	DMKDAU	DMKDIR	DMKDSB	DMKJRL	DMKPEN
		DMKPEQ	DMKSBL	DMKTRP	DMKURS	DMKVDE					
CLASDASD	000331	DMKACO	DMKACR	DMKALO	DMKBIO	DMKCCD	DMKCCO	DMKCCS	DMKCCW	DMKCFG	DMKCFQ
		DMKCFR	DMKCKD	DMKCKF	DMKCKM	DMKCKP	DMKCPM	DMKCPN	DMKCPO	DMKCPS	DMKCPT
		DMKCPV	DMKCPW	DMKCQG	DMKCQP	DMKCQQ	DMKCQR	DMKCQT	DMKDDR	DMKDEF	DMKDEI
		DMKDGD	DMKDID	DMKDIR	DMKDMP	DMKDSB	DMKGIO	DMKHVE	DMKIOC	DMKIOE	DMKIOF
		DMKIOJ	DMKIOQ	DMKIOS	DMKIOT	DMKLNK	DMKLOJ	DMKMNI	DMKMNT	DMKMSW	DMKPAG
		DMKPMA	DMKPRV	DMKSAD	DMKSAV	DMKSCN	DMKSCO	DMKSSP	DMKSSS	DMKSST	DMKTRD

LABEL	COUNT	REFERENCES									
		DMKUDR	DMKVCH	DMKVDA	DMKVDC	DMKVDD	DMKVDE	DMKVDF	DMKVDR	DMKVDS	DMKVDT
		DMKVER	DMKVIO	DMKVMI	DMKVRR	DMKVSC	DMKVSI	DMKVSJ			
CLASFBA	000343	DMKACO	DMKACR	DMKALO	DMKATS	DMKBIO	DMKCCD	DMKCCO	DMKCCS	DMKCCW	DMKCFG
		DMKCFH	DMKCFQ	DMKCFR	DMKCFS	DMKCFU	DMKCKD	DMKCKF	DMKCKH	DMKCKM	DMKCKN
		DMKCKR	DMKCKS	DMKCKT	DMKCKV	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPV	DMKCPW
		DMKCQG	DMKCQP	DMKCQQ	DMKCQR	DMKCQT	DMKDDR	DMKDEF	DMKDID	DMKDIR	DMKDMP
		DMKDRD	DMKDRE	DMKDSB	DMKERP	DMKGIO	DMKHVD	DMKHVE	DMKHVF	DMKIDU	DMKIOC
		DMKIOE	DMKIOF	DMKIOG	DMKIOH	DMKIOJ	DMKIOQ	DMKIOS	DMKIOT	DMKLNK	DMKMNI
		DMKMNT	DMKMON	DMKMSW	DMKNLD	DMKNLE	DMKOPE	DMKPAG	DMKPGT	DMKPGU	DMKPMA
		DMKPRV	DMKRSP	DMKRSQ	DMKSAV	DMKSCN	DMKSCO	DMKSEG	DMKSNC	DMKSPK	DMKSPS
		DMKSSP	DMKTCS	DMKTDK	DMKTRD	DMKUDR	DMKUNT	DMKVCH	DMKVDA	DMKVDC	DMKVDD
		DMKVDE	DMKVDF	DMKVDG	DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVIO	DMKVMI	DMKVRR
		DMKVSC	DMKVSI	DMKVSX	DMKWRM	DMKWRN	DMKZTD				
CLASGRAF	000148	DMKACO	DMKBLD	DMKCCH	DMKCCW	DMKCFM	DMKCFQ	DMKCFR	DMKCFT	DMKCFY	DMKCKD
		DMKCKF	DMKCPO	DMKCPV	DMKCPW	DMKCQG	DMKCQP	DMKCQT	DMKDEF	DMKDIA	DMKDIB
		DMKDID	DMKDIF	DMKDIR	DMKDSP	DMKEPS	DMKEXT	DMKGRG	DMKHPS	DMKHVD	DMKHVE
		DMKIOF	DMKIOJ	DMKIOS	DMKIOT	DMKLOG	DMKMNT	DMKNEA	DMKOPE	DMKOPR	DMKQCN
		DMKQCO	DMKQCQ	DMKQVM	DMKSSP	DMKTRD	DMKUSQ	DMKVCH	DMKVCN	DMKVCR	DMKVCS
		DMKVDA	DMKVDC	DMKVDR	DMKVDS	DMKVIO	DMKVSI				
CLASSPEC	000106	DMKBLD	DMKCCO	DMKCCS	DMKCCT	DMKCCW	DMKCFQ	DMKCFR	DMKCFT	DMKCKD	DMKCPB
		DMKCPM	DMKCQG	DMKCQP	DMKCQQ	DMKCQS	DMKCQU	DMKCQY	DMKDEF	DMKDIB	DMKDID
		DMKDIR	DMKHVE	DMKIOH	DMKIOJ	DMKIOS	DMKIOT	DMKJRL	DMKLOG	DMKLOH	DMKLOM
		DMKMNT	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKQCN	DMKQCO	DMKQCP	DMKRNH	DMKSCN
		DMKTRD	DMKUSQ	DMKVCH	DMKVDA	DMKVDC	DMKVDR	DMKVDS	DMKVER	DMKVIO	DMKVMI
		DMKVSI	DMKVSJ	DMKWRM							
CLASTAPE	000175	DMKACS	DMKCCO	DMKCCS	DMKCCW	DMKCFQ	DMKCFR	DMKCFU	DMKCKD	DMKCPB	DMKCPM
		DMKCPN	DMKCPO	DMKCPS	DMKCPW	DMKCQG	DMKCQP	DMKCQR	DMKDDR	DMKDID	DMKDMP
		DMKDMQ	DMKDSB	DMKGIO	DMKHVC	DMKIOE	DMKIOF	DMKIOJ	DMKIOS	DMKIOT	DMKMCC
		DMKMNI	DMKMNT	DMKMSW	DMKSAD	DMKSPT	DMKVCH	DMKVDC	DMKVDD	DMKVDE	DMKVDR
		DMKVDS	DMKVMI	DMKVSC	DMKVSI						
CLASTERM	000225	DMKACO	DMKACR	DMKACS	DMKBLD	DMKCCT	DMKCCW	DMKCFC	DMKCFM	DMKCFQ	DMKCFR
		DMKCFT	DMKCFY	DMKCKD	DMKCKF	DMKCNS	DMKCPB	DMKCPJ	DMKCPM	DMKCPO	DMKCPV
		DMKCPW	DMKCQG	DMKCQP	DMKCQQ	DMKCQS	DMKCQU	DMKCQY	DMKCSP	DMKCSQ	DMKCSR
		DMKCST	DMKDDR	DMKDEF	DMKDIA	DMKDIB	DMKDIF	DMKDIR	DMKDSP	DMKEPS	DMKERM
		DMKEXT	DMKGRC	DMKGRT	DMKHVD	DMKHVE	DMKIDU	DMKIOC	DMKIOF	DMKIOJ	DMKIOS
		DMKIOT	DMKJRL	DMKLOG	DMKLOH	DMKLOJ	DMKLOM	DMKMNT	DMKNES	DMKNET	DMKOPE
		DMKOPR	DMKQCN	DMKQCO	DMKQCP	DMKQCQ	DMKQVM	DMKRGA	DMKSCN	DMKSSP	DMKTRD
		DMKUSQ	DMKVCH	DMKVCN	DMKVDA	DMKVDC	DMKVDE	DMKVDR	DMKVDS	DMKVIO	DMKVSC
		DMKVSI	DMKVSJ	DMKWRM							
CLASURI	000121	DMKACO	DMKCCO	DMKCCW	DMKCFQ	DMKCKH	DMKCPB	DMKCPO	DMKCPW	DMKCQG	DMKCQP
		DMKCSB	DMKCSO	DMKCSP	DMKCSQ	DMKCSR	DMKCST	DMKCSW	DMKCSX	DMKDEF	DMKDID
		DMKDIR	DMKDRD	DMKGIO	DMKHVE	DMKHVF	DMKIOF	DMKIOS	DMKIOT	DMKQVM	DMKRSE
		DMKRSP	DMKSCN	DMKSPL	DMKSSP	DMKTRD	DMKURS	DMKVCH	DMKVDA	DMKVDC	DMKVDD
		DMKVDR	DMKVDS	DMKVIO	DMKVMI	DMKVSI	DMKVSJ	DMKVSP	DMKVSR	DMKVST	DMKVSX
CLASURO	000179	DMKCCS	DMKCCW	DMKCFQ	DMKCFU	DMKCKF	DMKCKH	DMKCKS	DMKCKV	DMKCPB	DMKCPO
		DMKCPS	DMKCPW	DMKCQG	DMKCQP	DMKCSB	DMKCSF	DMKCSO	DMKCSP	DMKCSQ	DMKCSR
		DMKCST	DMKDEF	DMKDID	DMKDIR	DMKDMP	DMKGIO	DMKHVE	DMKIOE	DMKIOF	DMKIOJ
		DMKIOS	DMKIOT	DMKMNT	DMKQVM	DMKRSE	DMKRSP	DMKSAD	DMKSCN	DMKSSP	DMKTRD
		DMKURS	DMKVCH	DMKVDA	DMKVDC	DMKVDD	DMKVDR	DMKVDS	DMKVIO	DMKVSI	DMKVSJ
		DMKVSP	DMKVST	DMKXAB	DMKXAD						
CLCMD	000005	DMKMIA	DMKMNI	DMKMON							
CLEAR	000051	DMKCCH	DMKCPW	DMKCQT	DMKCSB	DMKCSF	DMKCSO	DMKCSP	DMKCSQ	DMKCSU	DMKCSV
		DMKCSX	DMKEXT	DMKIMG	DMKIOG	DMKIOS	DMKLDOOE	DMKMCH	DMKMCT	DMKNEA	DMKPEQ
		DMKRGA	DMKSPL	DMKSSS	DMKVFS						
CLEARBIT	000004	DMKTDK	DMKVDH	DMKVDS							
CLKCOMP	000002	DMKDMQ									
CLOCKCMP	000002	DMKSAD									
CLSUS	000011	DMKMIA	DMKMNI	DMKMON							

LABEL	COUNT	REFERENCES									
CMDBEG	000001	DMKMHV									
CMDRDC	000005	DMKCCO	DMKCCS	DMKCCW							
CMDREJ	000026	DMKCNS	DMKDAD	DMKDDR	DMKDIB	DMKHPS	DMKIOT	DMKRNH	DMKRSE	DMKTPE	DMKVCN
		DMKVSP	DMKVSX								
CNTLBTU	000005	DMKRNH									
CNTSDATA	000013	DMKMNL									
CNTSREC	000014	DMKMNL									
CNTSSIZE	000011	DMKMNL									
CNTSTDAT	000004	DMKMNL									
CNTSTOD	000004	DMKMNL									
CNTS2DAT	000011	DMKMNL									
CNTUNTIC	000004	DMKCCW									
CODE	000037	DMKACS	DMKCNS	DMKCPM	DMKCPZ	DMKDSP	DMKEXT	DMKFRE	DMKFRT	DMKIOS	DMKIOT
		DMKIUA	DMKLOK	DMKMCH	DMKMHC	DMKMNT	DMKPMA	DMKPRG	DMKPTS	DMKRNH	DMKSCH
		DMKSVC	DMKSVD	DMKSWA	DMKTED	DMKTEE	DMKVCV	DMKVCX	DMKVSJ		
COLON	000006	DMKCQC	DMKPEI	DMKPET	DMKSAD	DMKVMD					
COMPFES	000009	DMKEIG	DMKSEV	DMKSIX							
COMPSEL	000020	DMKEIG	DMKSEV	DMKSIX							
COMPSYS	000006	DMKCCH	DMKEIG	DMKSEV							
CONACTV	000036	DMKCFQ	DMKCNS	DMKDSP	DMKRGA	DMKRGB	DMKRGC	DMKRNH	DMKTTX	DMKVCR	DMKVCS
CONADDR	000096	DMKCNS	DMKCNT	DMKGRA	DMKGRE	DMKGRG	DMKGRI	DMKGRT	DMKMON	DMKQCN	DMKQCO
		DMKQCQ	DMKRGA	DMKRGB	DMKRGC	DMKRGE	DMKRNH	DMKTTX	DMKTTY	DMKVCQ	DMKVCR
		DMKVCS	DMKVCU	DMKVCV							
CONADDR2	000010	DMKQCO	DMKTTY	DMKVCR							
CONADDR3	000002	DMKVCR									
CONCCW1	000181	DMKCNS	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRH	DMKGRI	DMKQCO	DMKQVM	DMKRGA
		DMKRGB	DMKRGC	DMKRGD	DMKRNH	DMKTTX	DMKTTY	DMKVCV			
CONCCW2	000114	DMKCNS	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRH	DMKGRI	DMKQVM	DMKRGA	DMKRGB
		DMKRGC	DMKRGD	DMKRNH	DMKTTX	DMKTTY	DMKVCR				
CONCCW3	000070	DMKCNS	DMKDIF	DMKGRD	DMKGRF	DMKIOE	DMKNES	DMKNET	DMKRGA	DMKRGB	DMKRGC
		DMKRNH	DMKTTX	DMKTTY	DMKVCR						
CONCCW4	000033	DMKCNS	DMKGRD	DMKGRE	DMKGRF	DMKGRH	DMKQCO				
CONCMD	000001	DMKRGB									
CONCMD1	000003	DMKRGA	DMKRGB								
CONCNT	000167	DMKCNS	DMKCNT	DMKGRA	DMKGRC	DMKGRD	DMKGRE	DMKGRG	DMKGRH	DMKGRI	DMKGRT
		DMKMON	DMKQCN	DMKQCO	DMKQCQ	DMKRGA	DMKRGB	DMKRGC	DMKRGD	DMKRNH	DMKTTX
		DMKTTY	DMKVCQ	DMKVCR	DMKVCS	DMKVCU	DMKVCV				
CONCNTL	000035	DMKCNS	DMKQCO	DMKRGA	DMKRGB	DMKRGC	DMKRNH	DMKTTX	DMKTTY	DMKVCQ	DMKVCR
CONCNT2	000015	DMKQCO	DMKTTX	DMKTTY	DMKVCR						
CONCNT3	000001	DMKVCR									
CONCOMND	000009	DMKCNS	DMKGRE	DMKRNH	DMKTTX						
CONDATA	000179	DMKCNS	DMKDIF	DMKGRA	DMKGRC	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRH	DMKGRT
		DMKIOE	DMKNES	DMKQCN	DMKQCO	DMKQCQ	DMKRGA	DMKRGB	DMKRGC	DMKRGD	DMKRNH
		DMKTTX	DMKTTY	DMKTTZ	DMKVCQ	DMKVCR	DMKVCS				
CONDATAL	000002	DMKIUC									
CONDCLR	000001	DMKVCS									
CONDCNCL	000004	DMKGRD	DMKQCQ	DMKRGB	DMKVCS						
CONDCNT	000030	DMKDIF	DMKIOE	DMKRGA	DMKRNH						
CONDEST	000004	DMKRNH									
CONDIAG	000008	DMKQCN	DMKQCQ	DMKVCR	DMKVCS						
CONDLE	000002	DMKRGB									
CONDLE1	000010	DMKRGB	DMKRGC	DMKRGD							
CONDLN	000008	DMKGRD	DMKGRH	DMKQCQ	DMKRGB	DMKVCN					
CONDRFMT	000007	DMKGRD	DMKQCQ	DMKRGB	DMKVCN	DMKVCS					
CONDWC	000042	DMKGRD	DMKGRE	DMKGRH	DMKQCN	DMKQCQ	DMKRGB	DMKVCR	DMKVCS		
CONESCP	000021	DMKCNS	DMKRNH	DMKTTX	DMKTTY	DMKVCR					
CONEWA	000007	DMKGRE	DMKRGB	DMKVCS							
CONEWRT	000009	DMKGRD	DMKGRE	DMKRGB	DMKVCS						

LABEL	COUNT	REFERENCES									
CONEXT	000012	DMKIUC	DMKIUE	DMKIUJ	DMKIUS						
CONEXTR	000001	DMKRNH									
CONEXTSZ	000002	DMKIUC									
CONFGRPT	000001	DMKCPJ									
CONFLAG	000023	DMKCNS	DMKGRE	DMKRGD	DMKRNH	DMKTTX	DMKTTY	DMKVCV			
CONFLAGS	000049	DMKCNS	DMKGRD	DMKQCN	DMKQCO	DMKQCQ	DMKRGB	DMKTTX	DMKTTY	DMKVCR	DMKVCS
		DMKVCV									
CONFLAG2	000006	DMKGRF	DMKTTX	DMKTTY							
CONFLG2	000040	DMKGRE	DMKGRI	DMKMON	DMKQCN	DMKQCO	DMKQCQ	DMKRGA	DMKRGB	DMKRGD	DMKRGE
		DMKVCR	DMKVCS								
CONFMH5	000001	DMKIUC									
CONFSOP	000015	DMKCFM	DMKCFQ	DMKGRD	DMKGRF	DMKGRG	DMKQCO	DMKRGB	DMKRGC	DMKVCR	DMKVCS
CONFSS	000055	DMKCFM	DMKCFQ	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKQCN	DMKQCO	DMKQCQ	DMKRGA
		DMKRGB	DMKRGC	DMKRGD	DMKVCR	DMKVCS					
CONFSYNC	000001	DMKIUC									
CONFTPN	000001	DMKIUC									
CONIDATA	000001	DMKIUC									
CONILUNM	000002	DMKIUC									
CONIMODE	000002	DMKIUC									
CONLABEL	000025	DMKRGA	DMKRGB	DMKRGC	DMKRGD						
CONLED	000008	DMKGRI	DMKQCN	DMKQCQ	DMKRGB	DMKVCS					
CONLENTH	000008	DMKIUC	DMKIUE	DMKIUJ	DMKIUS						
CONLNCNT	000007	DMKGRD	DMKGRH	DMKQCO	DMKQCQ	DMKRGB					
CONLNRES	000003	DMKGRD	DMKQCQ	DMKRGB							
CONMORE	000024	DMKCNS	DMKQCO	DMKTTX	DMKTTY	DMKVCS	DMKVCV				
CONMSGBK	000002	DMKIUC	DMKIUS								
CONNCB	000032	DMKGRE	DMKMON	DMKQCN	DMKQCO	DMKRGA	DMKRGB	DMKRGD	DMKRGE	DMKVCR	DMKVCS
CONNECT	000003	DMKDRD	DMKIUC								
CONNEWL	000004	DMKQCQ									
CONOUTPT	000059	DMKCNS	DMKGRD	DMKGRE	DMKQCN	DMKQCO	DMKQCQ	DMKRGB	DMKRNH	DMKTTX	DMKTTY
		DMKVCQ	DMKVCR	DMKVCS	DMKVCV						
CONPARM	000173	DMKCNS	DMKCNT	DMKGRC	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRH	DMKGRI	DMKGRT
		DMKQCN	DMKQCO	DMKQCQ	DMKRGA	DMKRGB	DMKRGC	DMKRNH	DMKTTX	DMKTTY	DMKVCR
		DMKVCS	DMKVCU								
CONPARM2	000016	DMKGRC	DMKGRD	DMKGRG	DMKQCN	DMKQCQ	DMKRGB	DMKVCS			
CONPFDEL	000002	DMKTTX	DMKVCR								
CONPFWRT	000002	DMKTTX	DMKVCR								
CONPNT	000164	DMKCFM	DMKCFQ	DMKCNS	DMKDSP	DMKGRD	DMKGRE	DMKGRF	DMKGRI	DMKQCO	DMKQCQ
		DMKQVM	DMKRGA	DMKRGB	DMKRGC	DMKRNH	DMKTTX	DMKTTY	DMKVCR	DMKVCS	DMKVCV
		DMKVCW	DMKVCX								
CONPPA1	000003	DMKGRE	DMKRGB	DMKVCS							
CONP1LEN	000004	DMKIUC									
CONP2LEN	000004	DMKIUC									
CONRD	000002	DMKQCO	DMKVCR								
CONRESET	000004	DMKCFQ	DMKRGA	DMKRGC							
CONRESP	000034	DMKCFQ	DMKCNS	DMKGRI	DMKQCO	DMKQCQ	DMKRGA	DMKRGB	DMKRGC	DMKRNH	DMKVCQ
		DMKVCR	DMKVCS	DMKVCV							
CONRETN	000049	DMKCFQ	DMKCNS	DMKGRG	DMKGRI	DMKQCO	DMKQCQ	DMKRGA	DMKRGB	DMKRGC	DMKRNH
		DMKTTX	DMKVCQ	DMKVCR	DMKVCS	DMKVCU					
CONRMOD	000002	DMKGRF	DMKVCR								
CONRTAG	000003	DMKRNH									
CONRTRY	000012	DMKCNS	DMKRNH								
CONSBADR	000003	DMKRGB									
CONSFCNT	000004	DMKQCQ									
CONSOLE	000003	DMKDEF									
CONSPLT	000020	DMKCNS	DMKQCO	DMKQCQ	DMKRNH	DMKTTY	DMKVCR	DMKVCS	DMKVCV		
CONSRID	000014	DMKRNH									
CONSTAT	000203	DMKCFQ	DMKCNS	DMKDSP	DMKGRD	DMKGRE	DMKGRI	DMKQCN	DMKQCO	DMKQCQ	DMKRGA

LABEL	COUNT	REFERENCES									
		DMKRGB	DMKRGC	DMKRNH	DMKTTX	DMKTTY	DMKVCQ	DMKVCR	DMKVCS	DMKVCV	
CONSTX	000005	DMKRGA	DMKRGB								
CONSTX1	000006	DMKRGB	DMKRGC								
CONSYN	000001	DMKRGC									
CONSYNC	000017	DMKCNS	DMKGRD	DMKQCO	DMKRGB	DMKRNH	DMKTTX	DMKTTY	DMKVCS	DMKVCV	
CONSYNXP	000006	DMKRGA	DMKRGB								
CONSYN1	000005	DMKRGA	DMKRGB								
CONSYSR	000041	DMKDIF	DMKNES	DMKNET	DMKRNH						
CONTACT	000003	DMKNET	DMKRNH								
CONTASK	000238	DMKCFM	DMKCFQ	DMKCNS	DMKCNT	DMKDSP	DMKGRA	DMKGRC	DMKGRD	DMKGRE	DMKGRF
		DMKGRG	DMKGRH	DMKGRI	DMKGRT	DMKMON	DMKNES	DMKQCN	DMKQCO	DMKQCP	DMKQCQ
		DMKQVM	DMKRGA	DMKRGB	DMKRGC	DMKRGD	DMKRGE	DMKRNH	DMKTTX	DMKTTY	DMKTTZ
		DMKVCQ	DMKVCR	DMKVCS	DMKVCU	DMKVCV	DMKVCW	DMKVCX			
CONTCMD	000027	DMKRNH									
CONTGMXB	000008	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRI	DMKQCN	DMKQCQ	DMKVCN		
CONTGMXD	000003	DMKQCN	DMKQCQ	DMKVCN							
CONTINUE	000064	DMKACS	DMKCKW	DMKCPI	DMKCPT	DMKDDR	DMKDEF	DMKGRC	DMKGRG	DMKGRI	DMKHVE
		DMKHVF	DMKLD00E	DMKMCD	DMKMCH	DMKMID	DMKOPE	DMKOPR	DMKPTR	DMKRGC	DMKSAD
		DMKTAP	DMKTHI	DMKTPE	DMKTRR	DMKVAT	DMKVCB	DMKVDC	DMKVDF	DMKVER	DMKVSI
CONTROL	000019	DMKACO	DMKCCH	DMKFRE	DMKMCT	DMKPTR	DMKPTT	DMKSEL			
CONTSIZE	000087	DMKCNS	DMKGRA	DMKGRC	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRI	DMKGRT	DMKQCN
		DMKQCO	DMKQCQ	DMKRGA	DMKRGB	DMKRGC	DMKRNH	DMKTTX	DMKTTY	DMKVCQ	DMKVCR
		DMKVCU									
CONTSKSZ	000023	DMKCNS	DMKGRA	DMKGRC	DMKGRI	DMKQCN	DMKQCO	DMKQCQ	DMKQVM	DMKRGA	DMKRGB
		DMKRGC	DMKRNH	DMKTTX	DMKTTY	DMKVCQ	DMKVCR	DMKVCS	DMKVCV		
CONUSER	000026	DMKCFQ	DMKCNS	DMKGRC	DMKGRI	DMKQCN	DMKQCO	DMKQCQ	DMKRGA	DMKRGB	DMKRGC
		DMKRNH	DMKTTX	DMKTTY	DMKVCQ	DMKVCR					
CONVDEVB	000004	DMKQCN	DMKQCO	DMKQCQ							
CONVFIXL	000001	DMKIUC									
CONVLUNM	000001	DMKIUC									
CONVMARE	000001	DMKIUC									
CONVMODE	000001	DMKIUC									
CONVTOTL	000001	DMKIUC									
CONWCC	000003	DMKGRD	DMKRGB								
CONWORK	000007	DMKQCQ	DMKRGD								
CONWORK1	000012	DMKGRC	DMKQCN	DMKQCO	DMKQCQ						
CONWORK2	000016	DMKGRC	DMKGRD	DMKQCN	DMKQCO	DMKQCQ	DMKRGB				
CONWRT	000004	DMKGRE	DMKRGA	DMKVCS							
CONWRTRD	000013	DMKCNS	DMKQCO	DMKTTX	DMKTTY						
CONWSF	000003	DMKRGB	DMKRGD	DMKVCS							
CORBPNT	000037	DMKATS	DMKPTR	DMKPTS	DMKPTT	DMKSEL	DMKSTR	DMKSWA	DMKSWM		
CORCFLCK	000027	DMKATS	DMKBLD	DMKCPY	DMKHVE	DMKMCH	DMKPGS	DMKPTR	DMKRPA	DMKSEG	DMKVMA
CORCP	000038	DMKCPU	DMKDMP	DMKDMQ	DMKGRE	DMKHPT	DMKHPU	DMKMCC	DMKMNI	DMKPMA	DMKPST
		DMKPTR	DMKPTS	DMKPTT	DMKSEL	DMKSPM	DMKSTA	DMKVCN	DMKVFC	DMKVFD	DMKVFE
		DMKVMC	DMKVRR	DMKVSD	DMKWRM						
CORDISA	000027	DMKCDB	DMKCDM	DMKCDS	DMKCPY	DMKDMP	DMKDMQ	DMKHVD	DMKMCH	DMKPGS	DMKSTA
		DMKVRR	DMKVRS								
CORFLAG	000183	DMKATS	DMKBLD	DMKCCW	DMKCDB	DMKCDM	DMKCDS	DMKCPP	DMKCPU	DMKCPY	DMKDGD
		DMKDMP	DMKDMQ	DMKGRE	DMKHPT	DMKHPU	DMKHVD	DMKHVE	DMKMCC	DMKMCH	DMKMHV
		DMKMNI	DMKPGM	DMKPGS	DMKPMA	DMKPRW	DMKPSA	DMKPST	DMKPTR	DMKPTS	DMKPTT
		DMKRPA	DMKSEG	DMKSEL	DMKSPM	DMKSTA	DMKSTR	DMKSWA	DMKVCN	DMKVFC	DMKVFD
		DMKVFE	DMKVFS	DMKVMA	DMKVMC	DMKVRR	DMKVRS	DMKVSD	DMKWRM		
CORFLUSH	000020	DMKATS	DMKCCW	DMKDGD	DMKPGM	DMKPTS	DMKPTT	DMKSEL	DMKSWA	DMKVMA	
CORFPNT	000084	DMKATS	DMKBLD	DMKCPU	DMKCPY	DMKDMP	DMKDMQ	DMKFRT	DMKGRE	DMKHVE	DMKPGS
		DMKPMA	DMKPTR	DMKPTS	DMKPTT	DMKQCO	DMKRPA	DMKSEL	DMKSTA	DMKSTR	DMKSWA
		DMKSWM	DMKVFD	DMKVFE							
CORFREE	000014	DMKATS	DMKPGS	DMKPTR	DMKPTT	DMKSTA	DMKSTR	DMKVMA			
CORIOLCK	000027	DMKATS	DMKBLD	DMKCPP	DMKMCH	DMKPGS	DMKPTR	DMKRPA	DMKSEG	DMKSEL	DMKSTA

LABEL	COUNT	REFERENCES									
		DMKSTR	DMKVFS	DMKVMA							
CORLCNT	000017	DMKBLD	DMKMNI	DMKPTR	DMKRPA	DMKSEG	DMKSTA	DMKSTR	DMKSWA	DMKVFS	
CORPGPNT	000066	DMKATS	DMKBLD	DMKCDS	DMKCPP	DMKDGD	DMKFRT	DMKMCH	DMKPGS	DMKPTR	DMKPTS
		DMKPTT	DMKQVM	DMKRPA	DMKSEG	DMKSEL	DMKSTR	DMKSWM	DMKUNT	DMKVFS	DMKVMA
CORSHARE	000034	DMKATS	DMKCCW	DMKCDS	DMKCPP	DMKCPY	DMKDGD	DMKMHV	DMKPGS	DMKPRW	DMKPSA
		DMKPTR	DMKPTS	DMKPTT	DMKSEL						
CORSWAP	000010	DMKPTR	DMKPTS	DMKSEL	DMKSWA						
CORSWPNT	000060	DMKATS	DMKBLD	DMKCCW	DMKCDS	DMKCFU	DMKCPP	DMKDGD	DMKDRD	DMKDRE	DMKIDU
		DMKMCH	DMKPGS	DMKPRW	DMKPSA	DMKPTR	DMKPTS	DMKPTT	DMKQVM	DMKRPA	DMKSEG
		DMKSEL	DMKSTR	DMKSWA	DMKSWM	DMKUDU	DMKVMA				
CORTABLE	000172	DMKATS	DMKBLD	DMKCCW	DMKCDB	DMKCDM	DMKCDS	DMKCFU	DMKCPP	DMKCPU	DMKCPY
		DMKDGD	DMKDMP	DMKDMQ	DMKDRD	DMKDRE	DMKFRT	DMKGRE	DMKHPT	DMKHPU	DMKHVD
		DMKHVE	DMKIDU	DMKMCC	DMKMCH	DMKMHV	DMKMNI	DMKPAH	DMKPGM	DMKPGS	DMKPMA
		DMKPRW	DMKPSA	DMKPST	DMKPTR	DMKPTS	DMKPTT	DMKQCO	DMKQVM	DMKRPA	DMKSEG
		DMKSEL	DMKSPM	DMKSTA	DMKSTR	DMKSWA	DMKSWM	DMKUDU	DMKUNT	DMKVCN	DMKVFC
		DMKVFD	DMKVFE	DMKVFS	DMKVMA	DMKVMC	DMKVRR	DMKVRS	DMKVSD	DMKWRM	
CORVM	000062	DMKATS	DMKCPP	DMKCPY	DMKDMP	DMKDMQ	DMKHVE	DMKMCH	DMKPGS	DMKPST	DMKPTR
		DMKPTS	DMKPTT	DMKSEL	DMKSPM	DMKSWA	DMKSWM	DMKVFC	DMKVFS	DMKVMA	DMKVRR
		DMKVSD	DMKWRM								
COUNT	000024	DMKCSV	DMKCSW	DMKCSY	DMKERM	DMKFMT	DMKFRE	DMKFRT	DMKPEI	DMKPEN	DMKTRA
CPABEND	000020	DMKCPS	DMKDMP	DMKOPE	DMKPRG	DMKPSA	DMKSAD	DMKSVC			
CPAPRINP	000006	DMKCPP	DMKCPU	DMKEXT	DMKMCT						
CPAPRPND	000006	DMKDSP	DMKEXT	DMKMCT							
CPASTAVL	000013	DMKAPI	DMKCFO	DMKCFY	DMKCPI	DMKCPJ	DMKLOJ	DMKSVC			
CPASTON	000013	DMKAPI	DMKCFO	DMKCPI	DMKCQY	DMKLOJ	DMKSVC				
CPCCHLK	000005	DMKCCH	DMKDSP								
CPCLOSE	000001	DMKDDR									
CPCMD	000001	DMKSND									
CPCREG0	000099	DMKACR	DMKAPI	DMKCKD	DMKCLK	DMKCPI	DMKCPP	DMKCPU	DMKDMP	DMKDSP	DMKEXT
		DMKFPS	DMKIOT	DMKLOK	DMKMCH	DMKMPO	DMKPRG	DMKPRV	DMKPTT	DMKQVM	DMKSTA
		DMKSVD	DMKTMR	DMKTRC	DMKTRD	DMKTRX	DMKVAT	DMKVFC	DMKVFR	DMKVRS	
CPCREG6	000009	DMKAPI	DMKCFO	DMKCPI	DMKDSP						
CPCREG8	000046	DMKCPS	DMKDSP	DMKEXT	DMKIOS	DMKIOT	DMKMCC	DMKMCD	DMKMCH	DMKMIA	DMKMNI
		DMKMON	DMKMOO	DMKPRG	DMKQVM	DMKSVD					
CPDASAAV	000012	DMKAPI	DMKCFO	DMKCFS	DMKCPI	DMKDSP	DMKLOH				
CPDASAON	000014	DMKAPI	DMKCFO	DMKCFS	DMKCPI	DMKCQY	DMKDSP	DMKLOH			
CPEX	000009	DMKDSP	DMKVCA								
CPEXADD	000229	DMKACO	DMKACS	DMKALG	DMKAPS	DMKAPT	DMKATS	DMKCCH	DMKCCS	DMKCCW	DMKCDS
		DMKCFG	DMKCFM	DMKCFO	DMKCFR	DMKCKF	DMKCKV	DMKCNS	DMKCPJ	DMKCPM	DMKCPO
		DMKCPP	DMKCPS	DMKCPV	DMKCPW	DMKCPX	DMKCQC	DMKDAD	DMKDAS	DMKDAU	DMKDGD
		DMKDGF	DMKDIB	DMKDID	DMKDSB	DMKDSP	DMKEXT	DMKFRE	DMKGIO	DMKGRG	DMKHPS
		DMKHPT	DMKHPU	DMKHVE	DMKHVF	DMKIOE	DMKIOQ	DMKIOS	DMKIOT	DMKIUA	DMKIUC
		DMKIUJ	DMKIUN	DMKIUP	DMKIUS	DMKLOC	DMKLOJ	DMKLOK	DMKMCC	DMKMCD	DMKMCH
		DMKMCI	DMKMCT	DMKMHC	DMKMIA	DMKMID	DMKMNI	DMKMNL	DMKMON	DMKMPO	DMKMSG
		DMKPAG	DMKPAH	DMKPGM	DMKPGS	DMKPGT	DMKPRG	DMKPRV	DMKPTR	DMKPTS	DMKPTT
		DMKQCO	DMKQCP	DMKQVM	DMKRGA	DMKRGB	DMKRNH	DMKRPA	DMKRSP	DMKRST	DMKSCH
		DMKSEL	DMKSND	DMKSPK	DMKSPM	DMKSSS	DMKSSU	DMKSTP	DMKSTR	DMKSVC	DMKSVD
		DMKSWA	DMKSWM	DMKTAP	DMKTHI	DMKTMR	DMKTPE	DMKTRD	DMKTRP	DMKTRT	DMKTRU
		DMKTRX	DMKUSQ	DMKVAT	DMKVAU	DMKVCA	DMKVCB	DMKVCP	DMKVCU	DMKVCW	DMKVCX
		DMKVDR	DMKVDS	DMKVFC	DMKVFD	DMKVFE	DMKVFR	DMKVFS	DMKVIO	DMKVMA	DMKVMC
		DMKVSE	DMKVSG	DMKVSJ	DMKVSP	DMKVSQ	DMKVST	DMKZTD			
CPEXBLOK	000507	DMKACO	DMKACS	DMKALG	DMKAPS	DMKAPT	DMKATS	DMKCCH	DMKCCS	DMKCCW	DMKCDS
		DMKCFG	DMKCFM	DMKCFO	DMKCFQ	DMKCFR	DMKCKF	DMKCKV	DMKCNS	DMKCPB	DMKCPI
		DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPU	DMKCPV	DMKCPW
		DMKCPX	DMKCPZ	DMKCQC	DMKCQT	DMKDAD	DMKDAS	DMKDAU	DMKDGD	DMKDGF	DMKDIB
		DMKDID	DMKDSB	DMKDSP	DMKERP	DMKEXT	DMKFRE	DMKGIO	DMKGRG	DMKHPS	DMKHPT
		DMKHPU	DMKHVE	DMKHVF	DMKIOE	DMKIOF	DMKIOQ	DMKIOS	DMKIOT	DMKIUA	DMKIUC
		DMKIUJ	DMKIUN	DMKIUP	DMKIUS	DMKLNK	DMKLOC	DMKLOJ	DMKLOK	DMKMCC	DMKMCD

LABEL	COUNT	REFERENCES									
		DMKMCH	DMKMCI	DMKMCT	DMKMHC	DMKMIA	DMKMID	DMKMNI	DMKMNL	DMKMON	DMKMPO
		DMKMSG	DMKPAG	DMKPAH	DMKPGM	DMKPGS	DMKPGT	DMKPRG	DMKPRV	DMKPTR	DMKPTS
		DMKPTT	DMKQCO	DMKQCP	DMKQVM	DMKRGA	DMKRGB	DMKRNH	DMKRPA	DMKRSP	DMKRST
		DMKSCH	DMKSEL	DMKSND	DMKSPK	DMKSPM	DMKSSS	DMKSSU	DMKSSV	DMKSTK	DMKSTP
		DMKSTR	DMKSVC	DMKSVD	DMKSWA	DMKSWM	DMKTAP	DMKTAQ	DMKTHI	DMKTMR	DMKTPE
		DMKTRD	DMKTRP	DMKTRT	DMKTRU	DMKTRX	DMKUNT	DMKUSQ	DMKVAT	DMKVAU	DMKVCA
		DMKVCB	DMKVCP	DMKVCU	DMKVCW	DMKVCX	DMKVDA	DMKVDC	DMKVDE	DMKVDR	DMKVDS
		DMKVFC	DMKVFD	DMKVFE	DMKVFR	DMKVFS	DMKVIO	DMKVMA	DMKVMC	DMKVSC	DMKVSE
		DMKVSG	DMKVSJ	DMKVSP	DMKVSQ	DMKVST	DMKWAI	DMKZTD			
CPEXBPNT	000020	DMKCPP	DMKDSP	DMKFRE	DMKPAG	DMKPAH	DMKSTK				
CPEXDEFR	000004	DMKDSP	DMKSTK								
CPEXFPNT	000139	DMKCCW	DMKCDS	DMKCFR	DMKCPI	DMKCPM	DMKCPP	DMKCPU	DMKCPZ	DMKDAD	DMKDAS
		DMKDAU	DMKDGD	DMKDSP	DMKERP	DMKFRE	DMKGIO	DMKHPS	DMKIOE	DMKIOF	DMKIOT
		DMKIUA	DMKIUJ	DMKLOC	DMKMCT	DMKPAG	DMKPAH	DMKPGM	DMKPTR	DMKPTS	DMKPTT
		DMKRPA	DMKSEL	DMKSTR	DMKSVC	DMKSWA	DMKSWM	DMKTAP	DMKTAQ	DMKTPE	DMKUNT
		DMKUSQ	DMKVCA	DMKVCB	DMKVIO	DMKVSC	DMKVSP	DMKVST	DMKWAI	DMKZTD	
CPEXMISC	000059	DMKCCW	DMKCDS	DMKCFG	DMKCFR	DMKCPB	DMKCPZ	DMKDGD	DMKGIO	DMKHVE	DMKIOQ
		DMKIUA	DMKIUC	DMKIUJ	DMKIUP	DMKIUS	DMKMNL	DMKPAH	DMKPGM	DMKPTR	DMKPTS
		DMKPTT	DMKRPA	DMKSCH	DMKSEL	DMKSSS	DMKSTK	DMKSTR	DMKSVC	DMKSWA	DMKSWM
		DMKVSJ									
CPEXPRIO	000005	DMKDSP	DMKSTK								
CPEXPROC	000018	DMKCPP	DMKDSP	DMKFRE	DMKLOC	DMKLOK	DMKMCT	DMKSCH	DMKSTK	DMKSVC	
CPEXREGS	000105	DMKACO	DMKACS	DMKAPT	DMKATS	DMKCCH	DMKCCS	DMKCFG	DMKCFM	DMKCKF	DMKCKV
		DMKCNS	DMKCPM	DMKCPO	DMKCPP	DMKCPV	DMKCPW	DMKCPZ	DMKDAS	DMKDGD	DMKDGF
		DMKDID	DMKDSB	DMKDSP	DMKEXT	DMKHVE	DMKIOE	DMKIOF	DMKIOQ	DMKLOC	DMKLOK
		DMKMCD	DMKMCH	DMKMCI	DMKMCT	DMKMHC	DMKPRG	DMKPRV	DMKPTR	DMKPTT	DMKQCO
		DMKQCP	DMKQVM	DMKSND	DMKSPK	DMKSPM	DMKSSS	DMKSSU	DMKSSV	DMKSVD	DMKSWM
		DMKTAP	DMKTHI	DMKTMR	DMKTRX	DMKVAT	DMKVAU	DMKVCP	DMKVCU	DMKVCW	DMKVCX
		DMKVFC	DMKVFR	DMKVFS	DMKVMA	DMKVSE	DMKVSG	DMKVSJ			
CPEXR0	000141	DMKACO	DMKALG	DMKCCW	DMKCDS	DMKCFQ	DMKCFR	DMKCNS	DMKCPB	DMKCPN	DMKCPS
		DMKCPW	DMKCPX	DMKCQC	DMKCQT	DMKDAD	DMKDAS	DMKDAU	DMKDGD	DMKFRE	DMKGIO
		DMKGRG	DMKHPS	DMKHPT	DMKHPU	DMKIUA	DMKIUC	DMKIUJ	DMKIUN	DMKIUP	DMKIUS
		DMKLOJ	DMKMCC	DMKMIA	DMKMID	DMKMNI	DMKMNL	DMKMON	DMKMSG	DMKPAG	DMKPAH
		DMKPGM	DMKPGS	DMKPTR	DMKPTS	DMKRGA	DMKRPA	DMKSCH	DMKSEL	DMKSSS	DMKSSV
		DMKSTP	DMKSTR	DMKSWA	DMKTAP	DMKTPE	DMKTRP	DMKTRT	DMKTRU	DMKTRX	DMKUSQ
		DMKVAT	DMKVCA	DMKVCB	DMKVCP	DMKVDA	DMKVDC	DMKVDE	DMKVDR	DMKVDS	DMKVFD
		DMKVFE	DMKVMC	DMKZTD							
CPEXR1	000014	DMKCCW	DMKCPO	DMKGIO	DMKHVE	DMKLNK	DMKLOK	DMKSND	DMKSSV	DMKTRT	DMKVDA
		DMKVDR	DMKVIO	DMKVST	DMKZTD						
CPEXR10	000011	DMKCCH	DMKCNS	DMKDSP	DMKIUA	DMKIUC	DMKIUJ	DMKIUP	DMKIUS	DMKMIA	DMKMNL
		DMKVCP									
CPEXR11	000076	DMKACO	DMKACS	DMKCCH	DMKCFO	DMKCFR	DMKCPJ	DMKCPM	DMKCPO	DMKCPP	DMKCPW
		DMKCPX	DMKDID	DMKDSP	DMKEXT	DMKFRE	DMKHPT	DMKHPU	DMKIOE	DMKIOQ	DMKIUN
		DMKLOJ	DMKLOK	DMKMCC	DMKMHC	DMKMIA	DMKMID	DMKMNI	DMKMNL	DMKMPO	DMKPAG
		DMKPGT	DMKPTR	DMKPTS	DMKQCP	DMKSEL	DMKSND	DMKSSS	DMKSSV	DMKSTK	DMKSTP
		DMKVCP	DMKVCU	DMKVIO	DMKVMC	DMKVSP	DMKVSQ	DMKVST	DMKWAI	DMKZTD	
CPEXR12	000054	DMKACO	DMKALG	DMKCFM	DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPS	DMKCPV	DMKCPW
		DMKDAD	DMKDAS	DMKDAU	DMKDSP	DMKIUN	DMKLOJ	DMKMCC	DMKMIA	DMKMID	DMKMNI
		DMKMPO	DMKPTR	DMKQCO	DMKQCP	DMKQVM	DMKSND	DMKSPM	DMKSSS	DMKSTP	DMKSTR
		DMKSVC	DMKTAP	DMKTPE	DMKVCA	DMKVCP	DMKVCU	DMKVDS	DMKVIO	DMKVSQ	
CPEXR13	000021	DMKCCH	DMKDSP	DMKIOS	DMKIOT	DMKMNL	DMKSEL	DMKSSV	DMKSTR	DMKVCA	DMKVCP
		DMKVDE	DMKVIO	DMKZTD							
CPEXR14	000007	DMKCDS	DMKIUC	DMKIUJ	DMKLOK	DMKSPM	DMKSSS				
CPEXR15	000002	DMKCDS	DMKIOQ								
CPEXR2	000015	DMKCFO	DMKCFQ	DMKCPX	DMKCQT	DMKLOJ	DMKPTR	DMKSEL	DMKSND	DMKVAT	
CPEXR3	000004	DMKIUA	DMKIUJ	DMKSSS							
CPEXR4	000005	DMKIUA	DMKSTR	DMKVIO	DMKZTD						
CPEXR5	000016	DMKCDS	DMKCFO	DMKCPX	DMKDAD	DMKDAS	DMKIOS	DMKIUA	DMKPAG	DMKPAH	DMKVDE

LABEL	COUNT	REFERENCES									
CPEXR6	000008	DMKERP	DMKIOF	DMKIOQ	DMKZTD						
CPEXR7	000006	DMKCDS	DMKIOQ	DMKPAG	DMKPTT						
CPEXR8	000013	DMKCFM	DMKCFQ	DMKDSP	DMKLOJ	DMKMNL	DMKSND	DMKUSQ	DMKVDR	DMKVDS	DMKVSP
		DMKVST	DMKZTD								
CPEXR9	000011	DMKCFQ	DMKPAG	DMKPAH	DMKPGM	DMKPTR	DMKSTR				
CPEXSIZE	000342	DMKACO	DMKACS	DMKALG	DMKAPI	DMKAPS	DMKAPT	DMKATS	DMKBLD	DMKCCH	DMKCCS
		DMKCCW	DMKCDS	DMKCFG	DMKCFM	DMKCFO	DMKCFQ	DMKCFR	DMKCKF	DMKCKV	DMKCNS
		DMKCPI	DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPU	DMKCPV
		DMKCPW	DMKCPX	DMKCPZ	DMKCQC	DMKCQT	DMKDAD	DMKDAS	DMKDAU	DMKDGD	DMKDIB
		DMKDID	DMKDIF	DMKDSB	DMKDSP	DMKEXT	DMKFRE	DMKGIO	DMKGRG	DMKHPS	DMKHPT
		DMKHPU	DMKHVE	DMKHVF	DMKIOE	DMKIOF	DMKIOG	DMKIOQ	DMKIOS	DMKIOT	DMKIUA
		DMKIUC	DMKIUJ	DMKIUN	DMKIUP	DMKIUS	DMKLOC	DMKLOH	DMKLOJ	DMKLOK	DMKMCC
		DMKMCD	DMKMCH	DMKMCI	DMKMCT	DMKMHC	DMKMIA	DMKMID	DMKMNI	DMKMNL	DMKMON
		DMKMPO	DMKMSG	DMKPGM	DMKPGS	DMKPGT	DMKPRV	DMKPTR	DMKPTS	DMKPTT	DMKQCO
		DMKQCP	DMKQVM	DMKRGA	DMKRGB	DMKRNH	DMKRPA	DMKRSP	DMKRST	DMKSCH	DMKSEL
		DMKSND	DMKSPK	DMKSPM	DMKSSS	DMKSSU	DMKSSV	DMKSTP	DMKSTR	DMKSVC	DMKSWA
		DMKSWM	DMKTAP	DMKTHI	DMKTPE	DMKTRD	DMKTRP	DMKTRT	DMKTRU	DMKTRX	DMKUSQ
		DMKVAT	DMKVCA	DMKVCB	DMKVCP	DMKVCU	DMKVCW	DMKVCX	DMKVDA	DMKVDC	DMKVDE
		DMKVDR	DMKVDS	DMKVFC	DMKVFD	DMKVFE	DMKVFS	DMKVIO	DMKVMA	DMKVMC	DMKVSE
		DMKVSG	DMKVSJ	DMKVSP	DMKVSQ	DMKVST	DMKZTD				
CPEXTYPE	000009	DMKDSP	DMKSTK								
CPFRELK	000007	DMKDSP	DMKFRE								
CPID	000065	DMKCCH	DMKCKD	DMKCKF	DMKCKM	DMKCKP	DMKCLK	DMKCNS	DMKCPI	DMKCPJ	DMKCPP
		DMKCPS	DMKCVT	DMKDMP	DMKDMQ	DMKGRF	DMKGRG	DMKMCH	DMKMCT	DMKOPE	DMKOPR
		DMKPGT	DMKQCN	DMKSAD	DMKSAV	DMKTOD					
CPIEXLOG	000005	DMKCFC	DMKLOG	DMKLOH	DMKOPE						
CPINIT	000007	DMKCPI	DMKGRG								
CPINITD	000010	DMKAPI	DMKCFC	DMKCFM	DMKCPJ	DMKCPS	DMKCPT	DMKEXT	DMKIOT		
CPLOKFL	000006	DMKEXT	DMKLOK								
CPMCHLK	000002	DMKMCH									
CPMCHSE	000011	DMKDSP	DMKMCH	DMKPRV	DMKVFR						
CPMFAWIA	000011	DMKEXT	DMKLOK								
CPMICAVL	000013	DMKAPI	DMKCFO	DMKCFV	DMKCFY	DMKCPI	DMKLOJ				
CPMICON	000016	DMKAPI	DMKCFO	DMKCFY	DMKCPI	DMKCQU	DMKCQY	DMKDSP	DMKLOJ	DMKPRV	
CPPTLBR	000052	DMKATS	DMKBLD	DMKCPP	DMKDSP	DMKFPS	DMKMCH	DMKMPO	DMKPGS	DMKPRV	DMKPRW
		DMKPTS	DMKRPA	DMKSEL	DMKSPM	DMKVAT	DMKVAU	DMKVMA			
CPQVMCU	000004	DMKIOT	DMKQVM								
CPRSTPND	000003	DMKDSP	DMKMPO								
CPRUN	000030	DMKCFP	DMKDSP	DMKEXT	DMKIOS	DMKIOT	DMKPMA	DMKPRG	DMKPRV	DMKSCH	DMKSTK
		DMKSVD	DMKTMR								
CPSEGPRT	000018	DMKAPI	DMKCFF	DMKCPI	DMKDSP	DMKPRW	DMKPSA	DMKVAT	DMKVAU		
CPSHRLK	000020	DMKAPI	DMKCCW	DMKCPM	DMKCPO	DMKCPS	DMKCPU	DMKDGD	DMKDSP	DMKWAI	
CPSHUT	000005	DMKACS	DMKCPS	DMKCPW	DMKIOS						
CPSIMLTB	000003	DMKPRW	DMKSTA								
CPSPMODE	000094	DMKCCT	DMKCDS	DMKCFD	DMKCFG	DMKCFP	DMKCFQ	DMKCFV	DMKCKM	DMKCPI	DMKCPS
		DMKCPU	DMKCPY	DMKCQY	DMKDMP	DMKDSP	DMKEXT	DMKFPS	DMKFRE	DMKHVD	DMKIOQ
		DMKIOS	DMKIOT	DMKLOJ	DMKMCH	DMKMHC	DMKMPO	DMKPMA	DMKPRV	DMKPRW	DMKPSA
		DMKQVM	DMKSAD	DMKSPM	DMKTRA	DMKVAT	DMKVIO	DMKVRR	DMKVSI	DMKVSJ	
CPSTAT	000003	DMKIOS	DMKMOO								
CPSTATUS	000061	DMKAPI	DMKCFP	DMKCLK	DMKCPI	DMKDSP	DMKEXT	DMKIOS	DMKIOT	DMKMCH	DMKMCT
		DMKMPO	DMKPMA	DMKPRG	DMKPRV	DMKQVM	DMKSCH	DMKSTK	DMKSVC	DMKSVD	DMKTMR
CPSTAT2	000193	DMKAPI	DMKCCT	DMKCCW	DMKCDS	DMKCFD	DMKCFG	DMKCFO	DMKCFP	DMKCFQ	DMKCFS
		DMKCFV	DMKCFY	DMKCKM	DMKCPI	DMKCPJ	DMKCPM	DMKCPO	DMKCPP	DMKCPS	DMKCPU
		DMKCPY	DMKCQU	DMKCQY	DMKDGD	DMKDMP	DMKDSP	DMKEXT	DMKFPS	DMKFRE	DMKHVD
		DMKIOQ	DMKIOS	DMKIOT	DMKLOH	DMKLOJ	DMKMCH	DMKMHC	DMKMPO	DMKPMA	DMKPRG
		DMKPRV	DMKPRW	DMKPSA	DMKQVM	DMKSAD	DMKSPM	DMKSVC	DMKTRA	DMKVAT	DMKVAU
		DMKVIO	DMKVRR	DMKVSI	DMKVSJ	DMKWAI					
CPSTAT3	000033	DMKAPI	DMKCFO	DMKCFS	DMKCLK	DMKCPI	DMKCQY	DMKDSP	DMKLOH	DMKMOO	DMKMPO

LABEL	COUNT	REFERENCES									
CPSTAT4	000121	DMKACS	DMKAPI	DMKCCW	DMKCDB	DMKCDM	DMKCFC	DMKCFF	DMKCFH	DMKCFY	DMKCKM
		DMKCPI	DMKCPS	DMKCPU	DMKCPW	DMKCPY	DMKDGD	DMKDMP	DMKDSP	DMKFPS	DMKIOS
		DMKIOT	DMKLOG	DMKLOH	DMKLOJ	DMKMCH	DMKOPE	DMKPRV	DMKPRW	DMKPSA	DMKPTR
		DMKPTS	DMKQVM	DMKSEG	DMKSEL	DMKSPM	DMKSTA	DMKSTR	DMKVAT	DMKVAU	DMKVMA
		DMKVME									
CPSTAT5	000080	DMKACR	DMKACS	DMKCCH	DMKCFS	DMKCFU	DMKCKD	DMKCKF	DMKCKH	DMKCKM	DMKCKN
		DMKCKP	DMKCKV	DMKCPI	DMKCPJ	DMKCPP	DMKCPS	DMKCPZ	DMKCSO	DMKDIF	DMKDMP
		DMKDSP	DMKGRG	DMKIOG	DMKMCT	DMKMNI	DMKMNT	DMKMOO	DMKNES	DMKNET	DMKOPE
		DMKOPR	DMKPGU	DMKPRG	DMKSCN	DMKSPS	DMKTOD	DMKVCH	DMKVDA	DMKVDG	DMKVDT
		DMKVER	DMKWRM	DMKWRN							
CPSTAT6	000005	DMKIOG	DMKIOS	DMKVFC							
CPSUPER	000017	DMKAPI	DMKCLK	DMKCPI	DMKDSP	DMKEXT	DMKIOT	DMKMCT	DMKMPO	DMKPMA	DMKPRG
		DMKQVM	DMKSVC								
CPSYSLK	000004	DMKDSP	DMKEXT	DMKLOK							
CPTCH	000003	DMKIOG	DMKIOS								
CPTERMLK	000004	DMKMCT									
CPTIDLE	000008	DMKCLK	DMKDSP	DMKMOO							
CPTIONT	000003	DMKDSP									
CPTMASK	000002	DMKAPI	DMKPMA								
CPTPAGE	000004	DMKDSP									
CPTRAPER	000003	DMKCKF	DMKCKP								
CPUID	000077	DMKAPI	DMKCCH	DMKCKP	DMKCKV	DMKCPI	DMKCPJ	DMKCPU	DMKCPY	DMKCQY	DMKDMP
		DMKDMQ	DMKDSP	DMKEXT	DMKHVC	DMKIOE	DMKIOG	DMKIOH	DMKIOJ	DMKLOH	DMKLOK
		DMKMCH	DMKMHC	DMKMNI	DMKOPR	DMKPRV	DMKPST	DMKRSP	DMKSAD	DMKSCH	DMKSSP
		DMKSTA	DMKSTK	DMKVER	DMKVME	DMKWAI					
CPULOG	000003	DMKCPI	DMKDMP								
CPUMCELL	000007	DMKCPP	DMKCPU	DMKHVD	DMKIOG	DMKPRV					
CPUMODEL	000013	DMKAPI	DMKCPJ	DMKCPS	DMKCPU	DMKIOG	DMKMCH	DMKMNT	DMKOPE	DMKPRV	DMKSEP
CPUSER	000003	DMKMNT	DMKPRV	DMKSEP							
CPUTIMER	000002	DMKDMQ									
CPUTIMR	000004	DMKSAD									
CPUVERSN	000023	DMKCPI	DMKCPM	DMKCPN	DMKCPS	DMKCPW	DMKGRD	DMKGRF	DMKHVD	DMKIOF	DMKIOG
		DMKMCI	DMKMNT	DMKOPR	DMKPRV	DMKSAD					
CPVR	000003	DMKDMQ									
CPWAIT	000023	DMKAPI	DMKCLK	DMKCPI	DMKDSP	DMKEXT	DMKIOS	DMKIOT	DMKMCH	DMKMCT	DMKMOO
		DMKSCH	DMKSTK	DMKVCA							
CPXSTOR	000033	DMKCKM	DMKCPY	DMKDMP	DMKPTR	DMKPTS	DMKSEG	DMKSEL	DMKSTA	DMKSTR	DMKVAT
		DMKVAU	DMKVMA								
CPXTRANS	000004	DMKDSP	DMKPXA	DMKPXB							
CP370EAV	000013	DMKAPI	DMKCFO	DMKCFS	DMKCPI	DMKLOH					
CP370EON	000032	DMKAPI	DMKCFO	DMKCFS	DMKCPI	DMKCPP	DMKCPU	DMKCQY	DMKDSP	DMKFPS	DMKLOH
		DMKPRG	DMKPRW	DMKPSA	DMKVAT	DMKVAU					
CRBIT	000001	DMKMCH									
CRESCND	000001	DMKRNH									
CRESDQ	000001	DMKDIF									
CRESERL	000002	DMKRNH									
CRESIMD	000004	DMKDIF	DMKEXT	DMKNET	DMKRNH						
CRLOG	000006	DMKPMA	DMKSAD								
CRTAIO	000003	DMKGRF	DMKGRG								
CRTALRM	000010	DMKGRD	DMKGRE	DMKGRG	DMKGRH						
CRTAPL	000003	DMKGRF	DMKGRI								
CRTCARD	000005	DMKGRF	DMKGRG								
CRTDIAG	000008	DMKGRD	DMKGRE	DMKGRG							
CRTEXT	000026	DMKCFQ	DMKCFR	DMKDID	DMKDIF	DMKGRI	DMKHPT	DMKMNI	DMKQVM	DMKRGA	DMKRGB
		DMKVCT	DMKVCV	DMKVDA	DMKVDR	DMKVDS					
CRTEXTSZ	000012	DMKCFQ	DMKCFR	DMKDID	DMKGRI	DMKMNI	DMKRGA	DMKRGB	DMKVCV	DMKVDA	DMKVDR
		DMKVDS									
CRTFMT	000008	DMKGRD	DMKGRF	DMKGRG	DMKQVM						

LABEL	COUNT	REFERENCES									
CRTFSII	000028	DMKCFM	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKRGA	DMKRGB	DMKRGC		
CRTFSSA	000029	DMKCFM	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKRGA	DMKRGB	DMKRGC		
CRTPPA1	000014	DMKGRE	DMKGRF	DMKRGA	DMKRGB	DMKRGC					
CRTSIO	000006	DMKGRD	DMKGRH	DMKGRI							
CRTUSEWA	000013	DMKGRD	DMKGRE	DMKGRF	DMKGRG						
CRTWNG	000006	DMKGRD	DMKGRE	DMKGRG							
CSADDR	000008	DMKAPI	DMKCPI	DMKCPP	DMKMCT	DMKPRV					
CSETDSM	000002	DMKDIF	DMKRNH								
CSSFEAT	000009	DMKAPI	DMKCPI	DMKCPO	DMKCPU	DMKMCT	DMKPRV				
CSW	000521	DMKACS	DMKCCH	DMKCKD	DMKCKH	DMKCKN	DMKCKP	DMKCNS	DMKCPI	DMKCPM	DMKCPZ
		DMKDDR	DMKDGF	DMKDIR	DMKDMP	DMKDMQ	DMKDSP	DMKEIG	DMKEXT	DMKFMT	DMKGIO
		DMKIOS	DMKIOT	DMKLD00E	DMKMNT	DMKOPE	DMKOPR	DMKOVR	DMKSAD	DMKSAV	DMKSEV
		DMKSIX	DMKSSP	DMKTES	DMKTRA	DMKTRD	DMKVCN	DMKVIO	DMKVMI	DMKVRR	DMKVRS
		DMKVSJ	DMKVSP								
CSWLMEP	000002	DMKDIF	DMKNES								
CSWLNCP	000002	DMKDIB	DMKNES								
CTCADDR	000005	DMKCFC									
CTCLASS	000004	DMKCFC	DMKCQS								
CTENTRY	000006	DMKCFC	DMKCQS								
CTFALIAS	000010	DMKCFC	DMKCMD	DMKCQS							
CTFCHANG	000001	DMKCFC									
CTFLAG	000008	DMKCFC	DMKCQS								
CTFLAST	000011	DMKCFC	DMKCMD	DMKCQS							
CTFSUBCM	000022	DMKCFC									
CTL	000001	DMKLD00E									
CTLREGS	000004	DMKDMQ									
CTNAME	000016	DMKCFC	DMKCQS								
CTRMLTR	000003	DMKDIF	DMKNES	DMKRNH							
CTSIZE	000005	DMKCFC	DMKCQS								
CTTRUNC	000002	DMKCFC									
CTTYPE	000006	DMKCFC									
CUE	000077	DMKACS	DMKCFQ	DMKCKH	DMKCPM	DMKDDR	DMKDID	DMKDIR	DMKDMP	DMKDMQ	DMKDSP
		DMKEXT	DMKFMT	DMKIOS	DMKIOT	DMKMNT	DMKMON	DMKNLD	DMKNLE	DMKOPE	DMKPAH
		DMKQVM	DMKRSE	DMKSAD	DMKSAV	DMKSPM	DMKSSP	DMKTAP	DMKVIO	DMKVSI	DMKVSJ
CURNTPSW	000003	DMKSAD									
CURRSAVE	000004	DMKIMG									
C0	000286	DMKACR	DMKAPI	DMKCKD	DMKCKP	DMKCLK	DMKCPI	DMKCPP	DMKCPU	DMKDMP	DMKDMQ
		DMKDSP	DMKEXT	DMKFPS	DMKFRE	DMKIOT	DMKLOK	DMKMCH	DMKMPO	DMKPMA	DMKPRG
		DMKPRV	DMKPTT	DMKQVM	DMKSAD	DMKSTA	DMKSVD	DMKTMR	DMKTRC	DMKTRD	DMKTRX
		DMKVFC	DMKVFR	DMKVRS							
C1	000870	DMKACO	DMKAPI	DMKAPT	DMKATS	DMKBIO	DMKBLD	DMKCCD	DMKCCH	DMKCCS	DMKCCW
		DMKCDB	DMKCDM	DMKCDS	DMKCFC	DMKCFD	DMKCFF	DMKCFG	DMKCFH	DMKCFM	DMKCFS
		DMKCFU	DMKCFV	DMKCFY	DMKCKM	DMKCKV	DMKCKW	DMKCNS	DMKCPB	DMKCPI	DMKCPM
		DMKCPO	DMKCPP	DMKCPS	DMKCPU	DMKCPW	DMKCPY	DMKCQS	DMKCQY	DMKCSC	DMKCSO
		DMKDAD	DMKDAS	DMKDAU	DMKDGD	DMKDGF	DMKDRD	DMKDRE	DMKDSP	DMKERM	DMKEXT
		DMKFPS	DMKGIO	DMKGRA	DMKGRC	DMKGRF	DMKGRT	DMKHPS	DMKHPT	DMKHPU	DMKHVC
		DMKHVD	DMKHVE	DMKHVF	DMKIDU	DMKIOG	DMKIOT	DMKISM	DMKIUA	DMKIUB	DMKIUC
		DMKIUE	DMKIUL	DMKLNK	DMKMCC	DMKMCH	DMKMHV	DMKMIA	DMKMNI	DMKMNJ	DMKNLD
		DMKNLE	DMKOPE	DMKPEI	DMKPER	DMKPGS	DMKPMA	DMKPRG	DMKPRV	DMKPRW	DMKPSA
		DMKPTR	DMKPTT	DMKQCN	DMKQCO	DMKQVM	DMKRGC	DMKRPD	DMKRSP	DMKRST	DMKSAD
		DMKSCH	DMKSCN	DMKSEG	DMKSEL	DMKSEP	DMKSFB	DMKSNC	DMKSPK	DMKSPL	DMKSPT
		DMKSRM	DMKSSS	DMKSST	DMKSSV	DMKSTR	DMKSVC	DMKSVD	DMKSWA	DMKTCS	DMKTCT
		DMKTMR	DMKTOD	DMKTRA	DMKTRC	DMKTRD	DMKTRK	DMKTRP	DMKTRQ	DMKTRU	DMKTRX
		DMKTTX	DMKUDR	DMKUDU	DMKVAT	DMKVAU	DMKVBM	DMKVCH	DMKVCN	DMKVCX	DMKVDR
		DMKVER	DMKVFR	DMKVIO	DMKVMC	DMKVMD	DMKVME	DMKVRS	DMKVSE	DMKVSG	DMKVSI
		DMKVSJ	DMKVSP	DMKVSQ	DMKVSX	DMKWRM	DMKWRN	DMKXAB			
C11	000003	DMKDSP									
C13	000001	DMKMCH									

LABEL	COUNT	REFERENCES									
C14	000031	DMKAPI	DMKCFH	DMKCKM	DMKCPI	DMKCPJ	DMKDMP	DMKMCH	DMKMCI	DMKPRV	DMKQVM
		DMKSAD									
C15	000017	DMKCFH	DMKCKM	DMKDMP	DMKPMA	DMKPRV	DMKQVM				
C2	000125	DMKAPI	DMKCFH	DMKCKD	DMKCKM	DMKCKP	DMKCPP	DMKDDR	DMKDMP	DMKDMQ	DMKDSP
		DMKEXT	DMKFMT	DMKFRE	DMKIOT	DMKLD00E	DMKMCT	DMKMHC	DMKMNT	DMKOPE	DMKOPR
		DMKPMA	DMKSAD	DMKSAV	DMKSSP	DMKVRS					
C3	000001	DMKQVM									
C5	000001	DMKQVM									
C6	000070	DMKAPI	DMKCFO	DMKCKP	DMKCPI	DMKCPS	DMKDMP	DMKDSP	DMKEXT	DMKFPS	DMKFRE
		DMKIOT	DMKPMA	DMKPRG	DMKPRV	DMKQVM					
C7	000006	DMKDSP	DMKMCH	DMKPRG	DMKPRV						
C8	000055	DMKCPS	DMKDSP	DMKEXT	DMKIOT	DMKMCC	DMKMCD	DMKMIA	DMKMNI	DMKMON	DMKPRG
		DMKQVM	DMKSVD								
C9	000002	DMKDSP									
DAMAGRPT	000004	DMKCPI	DMKCPJ	DMKDMP	DMKPMA						
DASDCL	000010	DMKMCC	DMKMNI	DMKMNL	DMKMOO						
DATACHK	000020	DMKCNS	DMKDAD	DMKDAS	DMKDDR	DMKRSE	DMKRSF	DMKUNT	DMKVSP	DMKVSX	
DATAOUT	000001	DMKPEQ									
DATE	000044	DMKACO	DMKCKF	DMKCKM	DMKCVT	DMKMID	DMKSAD	DMKTOD	DMKVME		
DDRCUA1	000002	DMKVER									
DDRCUA2	000002	DMKVER									
DDRKEYN	000001	DMKVER									
DDRREC	000001	DMKVER									
DDRSIZE	000001	DMKVER									
DE	000202	DMKACO	DMKCFR	DMKCKH	DMKCKN	DMKCKP	DMKCNS	DMKCPB	DMKCPM	DMKCPW	DMKCPZ
		DMKCSF	DMKCSO	DMKCSR	DMKCSW	DMKCSX	DMKDDR	DMKDIB	DMKDID	DMKDIF	DMKDIR
		DMKDMP	DMKDMQ	DMKDSB	DMKFMT	DMKGRF	DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKIOH
		DMKIOQ	DMKIOS	DMKIOT	DMKLD00E	DMKMNT	DMKMON	DMKNLD	DMKNLE	DMKOPE	DMKOPR
		DMKPAH	DMKRGA	DMKRGB	DMKRGE	DMKRNH	DMKRSE	DMKRSP	DMKSAD	DMKSAV	DMKSPL
		DMKSPT	DMKSSP	DMKSSU	DMKTAP	DMKUNT	DMKVCA	DMKVCB	DMKVCN	DMKVDD	DMKVIO
		DMKVMI	DMKVRS	DMKVSI	DMKVSJ	DMKVSP	DMKVSX				
DEDTEST	000002	DMKACO	DMKCKF								
DEFER	000651	DMKACO	DMKAPT	DMKATS	DMKBIO	DMKBLD	DMKCCH	DMKCCW	DMKCDB	DMKCDM	DMKCDS
		DMKCFC	DMKCFD	DMKCFF	DMKCFG	DMKCFH	DMKCFM	DMKCFS	DMKCFU	DMKCFV	DMKCKV
		DMKCNS	DMKCPB	DMKCPI	DMKCPM	DMKCPO	DMKCPP	DMKCPS	DMKCPU	DMKCPW	DMKCPY
		DMKCQS	DMKCQY	DMKCSC	DMKCSO	DMKDGD	DMKDGF	DMKDRD	DMKDRE	DMKDSP	DMKERM
		DMKGIO	DMKGRA	DMKGRC	DMKGRF	DMKGRT	DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKHVD
		DMKHVE	DMKHVF	DMKIDU	DMKIOF	DMKIOG	DMKIOT	DMKIUA	DMKIUB	DMKIUC	DMKIUE
		DMKIUL	DMKLNK	DMKMCC	DMKMHV	DMKMIA	DMKMNJ	DMKNLD	DMKNLE	DMKOPE	DMKPEI
		DMKPER	DMKPGS	DMKPMA	DMKPRG	DMKPRV	DMKPRW	DMKPTR	DMKQCN	DMKQCO	DMKRGC
		DMKRPA	DMKRPD	DMKRSP	DMKRST	DMKSEG	DMKSEP	DMKSFB	DMKSNC	DMKSPK	DMKSPL
		DMKSPT	DMKSRM	DMKSSS	DMKSST	DMKSSV	DMKSVC	DMKSVD	DMKTCS	DMKTCT	DMKTMR
		DMKTOD	DMKTRC	DMKTRD	DMKTRK	DMKTRP	DMKTRU	DMKTRX	DMKTTX	DMKUDR	DMKUDU
		DMKVAT	DMKVAU	DMKVBM	DMKVCH	DMKVCN	DMKVCX	DMKVDR	DMKVER	DMKVIO	DMKVMC
		DMKVMD	DMKVME	DMKVSE	DMKVSG	DMKVSI	DMKVSJ	DMKVSP	DMKVSQ	DMKVSX	DMKWRM
		DMKWRN	DMKXAB	DMKXAD							
DEFINTVL	000002	DMKMCD	DMKMNI								
DEFRSENS	000002	DMKCCS	DMKCCW								
DELPAGES	000013	DMKBLD	DMKCFP	DMKDEG	DMKSTR	DMKUSQ					
DELSEGS	000008	DMKBLD	DMKDEG	DMKUSQ							
DELSFB	000008	DMKCKS	DMKCSQ	DMKSPK	DMKTRU						
DENDCC	000005	DMKCCD									
DESTRTCC	000004	DMKCCD									
DEVCARD	000002	DMKACO	DMKCKF								
DEVCCH	000005	DMKCCH									
DEVICE	000009	DMKCPB	DMKCPS	DMKLD00E	DMKOVR	DMKTEE	DMKTEF				
DEVTABLE	000005	DMKCCW									
DEXTENT	000003	DMKCCD	DMKCCW								

LABEL	COUNT	REFERENCES									
DFRCC	000002	DMKIOT									
DFRCC1	000003	DMKIOS	DMKIOT	DMKVIO							
DFRCC3	000002	DMKVIO									
DFRET	000059	DMKACO	DMKCFV	DMKCPS	DMKCQS	DMKCQT	DMKCQU	DMKCQY	DMKDAD	DMKDAS	DMKDAU
		DMKDIB	DMKDIF	DMKDSB	DMKERM	DMKJRL	DMKPGM	DMKQCN	DMKQCP	DMKRNH	DMKSPR
		DMKSVD	DMKTHI	DMKVDD							
DIAGCNT	000001	DMKCCO									
DIAGCNTL	000001	DMKCCW									
DIAGSNS	000003	DMKCCS									
DISCEOC	000002	DMKRNH									
DISCNCT	000001	DMKRNH									
DISCREC	000002	DMKACO	DMKCKF								
DISPMSG	000002	DMKACO									
DLE	000021	DMKRGD									
DMKACOCL	000002	DMKCPV									
DMKACODS	000004	DMKQCP	DMKUSQ								
DMKACODV	000010	DMKCPV	DMKDIB	DMKVDR							
DMKACOFF	000006	DMKCPV	DMKUSQ								
DMKACON	000002	DMKLOJ									
DMKACOQU	000008	DMKHVD	DMKJRL								
DMKACOSA	000004	DMKVCT	DMKVCX								
DMKACOTM	000006	DMKCPV	DMKCQY	DMKUSQ							
DMKACR	000001	DMKSYM									
DMKACRCO	000004	DMKCPO	DMKMCT								
DMKACRCT	000006	DMKCCH	DMKMCH	DMKVSI							
DMKACRC3	000001	DMKIOS									
DMKACRIO	000001	DMKACS									
DMKACSCV	000006	DMKACR	DMKCPO	DMKMCT							
DMKACSRF	000002	DMKACR									
DMKALGON	000004	DMKCFC	DMKOPE								
DMKALOCA	000001	DMKCPZ									
DMKALOCP	000002	DMKMNT									
DMKALODU	000001	DMKIDU									
DMKAPICK	000005	DMKCPI	DMKCPU								
DMKAPIPR	000004	DMKCPI	DMKCPU								
DMKAPSCN	000001	DMKIUB									
DMKAPSIL	000001	DMKIUB									
DMKAPSSV	000011	DMKAPT	DMKAPU	DMKAPV	DMKAPX	DMKIUB					
DMKAPTEP	000002	DMKAPS									
DMKAPUSE	000002	DMKAPS									
DMKAPVCL	000002	DMKAPS									
DMKAPWPG	000008	DMKAPT	DMKCSV	DMKCSW	DMKCSY						
DMKAPXMG	000002	DMKAPS									
DMKAPYSD	000018	DMKCPT	DMKCQP	DMKCQQ	DMKCSF	DMKCSO	DMKSND				
DMKAPZNO	000012	DMKAPV	DMKCSQ	DMKCSW	DMKCSX	DMKRSP	DMKSPL				
DMKATSCF	000014	DMKCDS	DMKCFD	DMKIUE	DMKIUL	DMKTRC	DMKVAT				
DMKBIOCN	000001	DMKIUB									
DMKBIOIL	000001	DMKIUB									
DMKBIORS	000002	DMKCFQ									
DMKBIOSV	000001	DMKIUB									
DMKBLDEC	000004	DMKCFS	DMKLOH								
DMKBLDRL	000010	DMKCFP	DMKDEG	DMKPGS	DMKSTR	DMKUSQ					
DMKBLDRT	000022	DMKCFF	DMKCFP	DMKDEG	DMKLOH	DMKPGS	DMKSEG	DMKSTR	DMKVBM		
DMKBLDVM	000016	DMKALG	DMKCNS	DMKDIB	DMKGRF	DMKRGA	DMKRNH	DMKVCP	DMKVRR		
DMKBOXMS	000004	DMKGRT	DMKVCX								
DMKBOXNS	000003	DMKCFM	DMKGRT	DMKVCX							
DMKBOXPR	000001	DMKSEP									
DMKBSCER	000001	DMKIOS									

LABEL	COUNT	REFERENCES				
DMKCACOF	000002	DMKCMD				
DMKCACON	000002	DMKCMD				
DMKCACQY	000002	DMKCMD				
DMKCAOWN	000002	DMKCMD				
DMKCCDAS	000002	DMKCCW				
DMKCCDSK	000001	DMKCCW				
DMKCCFBA	000002	DMKCCW				
DMKCCFBD	000002	DMKCCW				
DMKCCHCF	000003	DMKACS	DMKDID	DMKIOG		
DMKCCHER	000002	DMKDSP				
DMKCCHIS	000003	DMKIOS	DMKSYM			
DMKCCHMX	000001	DMKIOG				
DMKCCHNT	000003	DMKIOT	DMKSYM			
DMKCCHRF	000006	DMKDSP	DMKVIO	DMKVSJ		
DMKCCHRT	000003	DMKIOE	DMKSYM			
DMKCCHSZ	000001	DMKIOG				
DMKCCH60	000002	DMKIOG	DMKSYM			
DMKCCOCH	000003	DMKCCD	DMKCCS			
DMKCCODD	000002	DMKCCW				
DMKCCOMS	000002	DMKCCW				
DMKCCOTH	000002	DMKCCW				
DMKCCOTP	000002	DMKCCW				
DMKCCSEN	000010	DMKCCD	DMKCCF	DMKCCO	DMKCCT	DMKCCW
DMKCCSLK	000024	DMKCCD	DMKCCW			
DMKCCSRM	000008	DMKCCD	DMKCCF	DMKCCT	DMKCCW	
DMKCCTCN	000002	DMKCCW				
DMKCCTDL	000002	DMKCCW				
DMKCCTLC	000002	DMKCCW				
DMKCCTRM	000002	DMKCCW				
DMKCCWB1	000002	DMKAPI	DMKCPI			
DMKCCWB2	000002	DMKAPI	DMKCPI			
DMKCCWB3	000002	DMKAPI	DMKCPI			
DMKCCWB4	000002	DMKAPI	DMKCPI			
DMKCCWB5	000002	DMKAPI	DMKCPI			
DMKCCWB6	000002	DMKAPI	DMKCPI			
DMKCCWB7	000002	DMKAPI	DMKCPI			
DMKCCWB8	000002	DMKAPI	DMKCPI			
DMKCCWCN	000018	DMKCCD	DMKCCF	DMKCCO	DMKCCS	
DMKCCWCW	000001	DMKCCS				
DMKCCWGN	000002	DMKAPI	DMKCPI			
DMKCCWL1	000002	DMKAPI	DMKCPI			
DMKCCWL2	000002	DMKAPI	DMKCPI			
DMKCCWL3	000002	DMKAPI	DMKCPI			
DMKCCWL4	000002	DMKAPI	DMKCPI			
DMKCCWL5	000002	DMKAPI	DMKCPI			
DMKCCWRT	000012	DMKCCD	DMKCCF	DMKCCO	DMKCCS	DMKCCT
DMKCCWSB	000007	DMKSYM	DMKTRD	DMKTRK		
DMKCCWTR	000007	DMKGIO	DMKHVC	DMKSYM	DMKVSI	
DMKCCW0	000002	DMKAPI	DMKCPI			
DMKCCW1	000002	DMKAPI	DMKCPI			
DMKCDBDC	000002	DMKCFC				
DMKCDBDI	000002	DMKCFC				
DMKCDMDM	000002	DMKCFC				
DMKCDMDU	000002	DMKCFC				
DMKCDSCP	000002	DMKCFC				
DMKCDSTO	000002	DMKCFC				
DMKCFCCO	000002	DMKUDR				
DMKCFCMD	000002	DMKCFM				

LABEL	COUNT	REFERENCES									
DMKCFCTB	000001	DMKCQS									
DMKCFDAD	000002	DMKCFC									
DMKCFDLO	000002	DMKCFC									
DMKCFFSB	000002	DMKCFG									
DMKCFGCL	000002	DMKHVC									
DMKCFGII	000002	DMKLOG									
DMKCFGIP	000002	DMKCFC									
DMKCFGIR	000002	DMKREI									
DMKCFHSV	000002	DMKCFC									
DMKCFJBE	000002	DMKCFC									
DMKCFJRQ	000004	DMKCFC									
DMKCFJSL	000002	DMKCFC									
DMKCFMAT	000037	DMKCFJ	DMKCNS	DMKGRD	DMKGRF	DMKGRG	DMKRGB	DMKRGC	DMKRNH	DMKSND	DMKSYM
		DMKVCN	DMKVCP	DMKVCU							
DMKCFMBK	000073	DMKCFO	DMKCNS	DMKCPP	DMKCPS	DMKCPU	DMKDSP	DMKGRF	DMKGRG	DMKHVC	DMKMCH
		DMKMCT	DMKMPO	DMKPER	DMKPET	DMKPRG	DMKPTR	DMKRGA	DMKRGC	DMKRNH	DMKSND
		DMKSVD	DMKSYM	DMKTRC	DMKTRD	DMKVCN	DMKVCP	DMKVCR	DMKVCU	DMKVMA	
DMKCFMEN	000015	DMKGRG	DMKHVC	DMKOPE	DMKPET	DMKRGA	DMKSYM	DMKVCN	DMKVCU		
DMKCFMRU	000002	DMKVMA									
DMKCFMSD	000001	DMKSND									
DMKCFMWU	000001	DMKCFJ									
DMKCFOSA	000002	DMKCMD									
DMKCFOSC	000002	DMKCMD									
DMKCFOSF	000002	DMKCMD									
DMKCFOSP	000002	DMKCMD									
DMKCFOSQ	000002	DMKCMD									
DMKCFOSR	000002	DMKCMD									
DMKCFOS3	000002	DMKCMD									
DMKCFPRR	000030	DMKCFG	DMKCFO	DMKCFS	DMKCPB	DMKCPO	DMKDEG	DMKMCH	DMKMCT	DMKUSQ	
DMKCFQRD	000018	DMKCFP	DMKCPB	DMKCPS	DMKDEF	DMKDIB	DMKLNK	DMKNET	DMKVDR		
DMKCFREP	000004	DMKCFQ	DMKVDR								
DMKCFSAC	000002	DMKCMD									
DMKCFSAP	000002	DMKCMD									
DMKCFSAS	000004	DMKDEG	DMKLOH								
DMKCFSCC	000002	DMKCMD									
DMKCFSCP	000002	DMKCMD									
DMKCFSEC	000002	DMKCMD									
DMKCFSEM	000002	DMKCMD									
DMKCFSIM	000002	DMKCMD									
DMKCFSIS	000002	DMKCMD									
DMKCFSLE	000002	DMKCMD									
DMKCFSMG	000002	DMKCMD									
DMKCFSNT	000002	DMKCMD									
DMKCFSPX	000002	DMKCMD									
DMKCFSRN	000002	DMKCMD									
DMKCFSSA	000002	DMKCMD									
DMKCFSSM	000002	DMKCMD									
DMKCFSVC	000002	DMKCMD									
DMKCFSVS	000002	DMKCMD									
DMKCFSWG	000002	DMKCMD									
DMKCFS37	000002	DMKCMD									
DMKCFTRM	000002	DMKCFC									
DMKCFUDU	000002	DMKCMD									
DMKCFULO	000002	DMKCMD									
DMKCFUMI	000002	DMKCMD									
DMKCFUMO	000002	DMKCMD									
DMKCFUMW	000002	DMKCMD									
DMKCFUPA	000002	DMKCMD									

LABEL	COUNT	REFERENCES		
DMKCFURE	000002	DMKCMD		
DMKCFVMI	000002	DMKCMD		
DMKCFVSB	000002	DMKCMD		
DMKCFVSM	000002	DMKCMD		
DMKCFWEP	000003	DMKCFC	DMKDIR	
DMKCFYAG	000002	DMKCMD		
DMKCFYAS	000002	DMKCMD		
DMKCFYPF	000002	DMKLOH		
DMKCFYSA	000002	DMKCMD		
DMKCFYSC	000002	DMKCMD		
DMKCFYSM	000002	DMKCMD		
DMKCFYSP	000006	DMKCMD	DMKSPC	
DMKCKD	000001	DMKSAV		
DMKCKDCP	000002	DMKCKM		
DMKCKDEV	000002	DMKCKP		
DMKCKDGP	000002	DMKCKN		
DMKCKDSW	000012	DMKCKH	DMKCKM	DMKCKP
DMKCKDVA	000004	DMKCKH		
DMKCKF	000001	DMKSAV		
DMKCKFAD	000002	DMKCKM		
DMKCKFBL	000002	DMKERP		
DMKCKFCL	000002	DMKERP		
DMKCKFCN	000002	DMKCKD		
DMKCKFCQ	000002	DMKERP		
DMKCKFCV	000004	DMKCKH		
DMKCKFC2	000004	DMKCKD		
DMKCKFDL	000004	DMKCKH		
DMKCKFDV	000002	DMKCKP		
DMKCKFEP	000002	DMKERP		
DMKCKFES	000002	DMKERP		
DMKCKFID	000002	DMKCKP		
DMKCKFIL	000002	DMKCKP		
DMKCKFIP	000002	DMKERP		
DMKCKFIQ	000002	DMKERP		
DMKCKFLG	000002	DMKCKH		
DMKCKFLI	000004	DMKCKP		
DMKCKFLS	000004	DMKCKP		
DMKCKFMQ	000002	DMKERP		
DMKCKFMX	000002	DMKERP		
DMKCKFOC	000004	DMKCKM	DMKCKN	
DMKCKFOW	000006	DMKCKM	DMKCKN	
DMKCKFRC	000002	DMKCKP		
DMKCKFRU	000002	DMKCKH		
DMKCKFSF	000004	DMKCKH		
DMKCKFTE	000002	DMKCKD		
DMKCKFTP	000002	DMKCKP		
DMKCKFTR	000006	DMKCKN		
DMKCKFVM	000002	DMKCKM		
DMKCKFVR	000004	DMKCKD	DMKCKM	
DMKCKFVT	000002	DMKERP		
DMKCKFWA	000020	DMKCKH	DMKCKP	
DMKCKFWM	000002	DMKCKH		
DMKCKFWT	000002	DMKCKM		
DMKCKH	000001	DMKSAV		
DMKCKHAC	000008	DMKCKF		
DMKCKHDC	000001	DMKCKP		
DMKCKHDL	000022	DMKCKF	DMKCKP	
DMKCKHIN	000006	DMKCKF		

LABEL	COUNT	REFERENCES									
DMKCKHIO	000002	DMKCKP									
DMKCKHPR	000072	DMKCKF	DMKCKW								
DMKCKHST	000002	DMKCKF									
DMKCKHSZ	000008	DMKCKF									
DMKCKHWM	000002	DMKCKF									
DMKCKM	000001	DMKSAV									
DMKCKMDV	000002	DMKCKP									
DMKCKMSG	000018	DMKCKH	DMKCKN	DMKCKP							
DMKCKMSV	000002	DMKCKP									
DMKCKN	000001	DMKSAV									
DMKCKNBC	000002	DMKCKM									
DMKCKNBH	000002	DMKCKM									
DMKCKNEC	000002	DMKCKM									
DMKCKNEH	000002	DMKCKM									
DMKCKNIO	000028	DMKCKF	DMKCKM	DMKCKW	DMKERP						
DMKCKNND	000002	DMKSAV									
DMKCKNPS	000002	DMKCKM									
DMKCKNPW	000004	DMKCKM	DMKERP								
DMKCKNPX	000002	DMKCKM									
DMKCKNPY	000016	DMKCKM									
DMKCKNPZ	000004	DMKCKM									
DMKCKNRB	000037	DMKCKF	DMKCKH	DMKCKW							
DMKCKNRD	000010	DMKCKF	DMKCKM	DMKCKW	DMKERP						
DMKCKNTB	000004	DMKCKD	DMKCKF								
DMKCKP	000284	DMKCKD	DMKCKF	DMKCKH	DMKCKM	DMKCKN	DMKCKW	DMKERP	DMKPGU	DMKSAV	DMKSEG
DMKCKPCS	000002	DMKCKH									
DMKCKPDV	000002	DMKCKH									
DMKCKPER	000006	DMKCKF	DMKCKH	DMKCKW							
DMKCKPFL	000014	DMKCKF	DMKCKH								
DMKCKPIP	000002	DMKCKH									
DMKCKPLD	000002	DMKSAV									
DMKCKPLE	000001	DMKSAV									
DMKCKPLF	000001	DMKSAV									
DMKCKPLH	000001	DMKSAV									
DMKCKPLM	000001	DMKSAV									
DMKCKPLN	000002	DMKSAV									
DMKCKPLW	000001	DMKSAV									
DMKCKPMP	000014	DMKCKD	DMKCKM								
DMKCKPNC	000003	DMKSAV									
DMKCKPND	000002	DMKSAV									
DMKCKPRM	000012	DMKCKD	DMKCKH								
DMKCKPRS	000002	DMKSAV									
DMKCKPR2	000001	DMKSAV									
DMKCKPSN	000002	DMKCKH									
DMKCKPST	000003	DMKSAV									
DMKCKPT	000002	DMKSAV									
DMKCKPVR	000002	DMKCKM									
DMKCKRSY	000002	DMKCPJ									
DMKCKSCV	000006	DMKIDU	DMKVSE	DMKWRN							
DMKCKSPL	000072	DMKACO	DMKAPV	DMKCSB	DMKCSO	DMKCSQ	DMKCSV	DMKCSW	DMKCSY	DMKMIA	DMKNLE
		DMKRSP	DMKSPK	DMKSPL	DMKSPS	DMKTRT	DMKTRU	DMKVME	DMKVSD	DMKVST	DMKWRN
		DMKXAB									
DMKCKTIN	000003	DMKCKV									
DMKCKTMP	000002	DMKWRM									
DMKCKTSD	000034	DMKCFU	DMKCKV	DMKCSQ	DMKCSY	DMKDRD	DMKDRE	DMKMIA	DMKNLE	DMKRST	DMKSPK
		DMKSPS	DMKTRU	DMKVME	DMKVSE	DMKVSG					
DMKCKTSU	000010	DMKCKS	DMKCSQ	DMKCSY	DMKSPL	DMKVSG					
DMKCKTUU	000026	DMKACO	DMKCFU	DMKCSW	DMKDRD	DMKDRE	DMKIDU	DMKMIA	DMKNLE	DMKSPL	DMKSPS

LABEL	COUNT	REFERENCES							
		DMKTRT	DMKTRU	DMKVME					
DMKCKVWM	000002	DMKWRM							
DMKCKW	000001	DMKSAV							
DMKCKWHT	000002	DMKCKF							
DMKCKWSP	000002	DMKCKF							
DMKCLKAP	000001	DMKEXT							
DMKCLKCC	000001	DMKEXT							
DMKCLKCK	000005	DMKCPI	DMKCPU						
DMKCLKSC	000002	DMKEXT							
DMKCMD	000002	DMKCFC							
DMKCMDAT	000002	DMKCFC							
DMKCMDCA	000002	DMKCFC							
DMKCMDDE	000002	DMKCFC							
DMKCMDDG	000002	DMKCFC							
DMKCMDIG	000002	DMKCFC							
DMKCMDIN	000004	DMKCFC							
DMKCMDNO	000002	DMKCFC							
DMKCMDNR	000002	DMKCFC							
DMKCMDQA	000004	DMKCFC							
DMKCMDQG	000004	DMKCFC							
DMKCMDQP	000001	DMKCFC							
DMKCMDQQ	000002	DMKCFC							
DMKCMDQR	000002	DMKCFC							
DMKCMDQS	000002	DMKCFC							
DMKCMDQV	000001	DMKCFC							
DMKCMDSA	000002	DMKCFC							
DMKCMDSC	000002	DMKCFC							
DMKCMDSG	000002	DMKCFC							
DMKCMDSO	000002	DMKCFC							
DMKCMDSR	000002	DMKCFC							
DMKCNS	000001	DMKSYM							
DMKCNSEN	000004	DMKCPV	DMKOPE	DMKQVM					
DMKCNSIC	000001	DMKQCO							
DMKCNSIN	000002	DMKACS	DMKIOT						
DMKCNTED	000022	DMKCNS	DMKGRG	DMKGRT	DMKRGC	DMKRNH	DMKVCR	DMKVCU	
DMKCPBEX	000002	DMKCFC							
DMKCPBNR	000002	DMKCFC							
DMKCPBRS	000002	DMKCFC							
DMKCPBRW	000002	DMKCFC							
DMKCPBRY	000002	DMKCFC							
DMKCPBSR	000002	DMKCFC							
DMKCPE	000005	DMKFRE	DMKFRT	DMKLD00E	DMKPSA				
DMKCPEID	000008	DMKCQY	DMKDMP	DMKHVD	DMKMNI	DMKOPE	DMKSEP	DMKSYM	DMKVME
DMKCPEML	000004	DMKAPI	DMKCPI	DMKCPJ	DMKSYM				
DMKCPEND	000010	DMKCPI	DMKCPJ	DMKDMQ	DMKMNI	DMKSTA	DMKSYM		
DMKCPEPP	000004	DMKHVD	DMKMNI	DMKSYM					
DMKCPIBD	000003	DMKOPE	DMKSTA	DMKVRR					
DMKCPICA	000001	DMKCPJ							
DMKCPICD	000003	DMKCQY	DMKOPE	DMKSAV					
DMKCPIFL	000001	DMKSTA							
DMKCPIFT	000001	DMKALO							
DMKCPIF1	000003	DMKCPJ							
DMKCPILF	000001	DMKALO							
DMKCPIMS	000006	DMKALO	DMKCPJ	DMKCPZ	DMKMNT				
DMKCPINT	000002	DMKSAV	DMKSSP						
DMKCPINU	000001	DMKALO							
DMKCPIOL	000003	DMKCPJ	DMKOPE	DMKWRM					
DMKCPIRP	000002	DMKALO	DMKMNT						

LABEL	COUNT	REFERENCES	
DMKCPISH	000006	DMKALO	DMKMNT
DMKCPISW	000001	DMKSTA	
DMKCPITR	000002	DMKOPE	DMKSTA
DMKCPIVP	000002	DMKALO	DMKMNT
DMKCPIWC	000008	DMKCPJ	DMKOPE
DMKCPIWT	000005	DMKMNT	
DMKCPJNT	000002	DMKCPI	
DMKCPJST	000002	DMKCPI	DMKSTA
DMKCPJTA	000001	DMKSTA	
DMKCPMIO	000002	DMKCPU	
DMKCPNVY	000002	DMKCPT	
DMKCPOFF	000002	DMKCPU	
DMKCPPUP	000005	DMKCPO	DMKMCT
DMKCPSH	000002	DMKCFC	
DMKCPSSH	000002	DMKCFC	
DMKCPTNF	000002	DMKCFC	
DMKCPUVY	000002	DMKCPT	
DMKCPVAA	000002	DMKHVD	
DMKCPVAC	000004	DMKCFC	DMKUDU
DMKCPVAE	000004	DMKCPJ	DMKRNH
DMKCPVDS	000002	DMKCFC	
DMKCPVEN	000002	DMKCFC	
DMKCPWFB	000002	DMKCPN	
DMKCPWUN	000002	DMKCPS	
DMKCPWVF	000002	DMKCPT	
DMKCPX	000001	DMKIOT	
DMKCPXCK	000004	DMKTDK	DMKVDH
DMKCPXZT	000002	DMKIOT	DMKZTD
DMKCPYLK	000002	DMKCFC	
DMKCPYUL	000002	DMKCFC	
DMKCPZFD	000002	DMKMNT	
DMKCPZMG	000002	DMKMNT	
DMKCPZPG	000002	DMKCPT	
DMKCPZVR	000002	DMKMNT	
DMKCQCPT	000002	DMKCMD	
DMKCQGID	000002	DMKCMD	
DMKCQGQA	000002	DMKCMD	
DMKCQGQC	000002	DMKCMD	
DMKCQGQD	000002	DMKCMD	
DMKCQGQG	000002	DMKCMD	
DMKCQGQH	000002	DMKCMD	
DMKCQGQL	000002	DMKCMD	
DMKCQGQS	000002	DMKCMD	
DMKCQGQT	000002	DMKCMD	
DMKCQGQU	000002	DMKCMD	
DMKCQHFG	000002	DMKCMD	
DMKCQHFI	000004	DMKCPJ	DMKLOM
DMKCQHFS	000002	DMKCMD	
DMKCQHNG	000004	DMKCMD	
DMKCQHNS	000004	DMKCMD	
DMKCQHRG	000004	DMKCMD	
DMKCQHRS	000004	DMKCMD	
DMKCQHTG	000004	DMKCMD	
DMKCQHTS	000004	DMKCMD	
DMKCQIFR	000002	DMKCQH	
DMKCQPQA	000002	DMKCMD	
DMKCQPQD	000002	DMKCMD	
DMKCQPQG	000002	DMKCMD	

LABEL	COUNT	REFERENCES	
DMKCQPQL	000002	DMKCMD	
DMKCQPQP	000002	DMKCMD	
DMKCQPQS	000002	DMKCMD	
DMKCQPQT	000002	DMKCMD	
DMKCQPQU	000002	DMKCMD	
DMKCQPSC	000002	DMKCQQ	
DMKCQQID	000004	DMKCMD	DMKSPC
DMKCQQPT	000002	DMKCQP	
DMKCQQQL	000002	DMKCMD	
DMKCQQQS	000002	DMKCMD	
DMKCQQQT	000002	DMKCMD	
DMKCQRAF	000002	DMKCMD	
DMKCQRDP	000002	DMKCMD	
DMKCQRHD	000002	DMKCMD	
DMKCQRPG	000002	DMKCMD	
DMKCQRPR	000002	DMKCMD	
DMKCQRQD	000002	DMKCMD	
DMKCQSMI	000002	DMKCMD	
DMKCQSMW	000002	DMKCMD	
DMKCQSNA	000002	DMKCMD	
DMKCQSPS	000002	DMKCMD	
DMKCQSQC	000002	DMKCFC	
DMKCQSRE	000002	DMKCMD	
DMKCQSSC	000002	DMKCMD	
DMKCQTCL	000002	DMKCMD	
DMKCQTRE	000002	DMKCQP	
DMKCQTSN	000002	DMKCQP	
DMKCQTST	000002	DMKCMD	
DMKCQUSE	000002	DMKCMD	
DMKCQUST	000002	DMKCMD	
DMKCQUTE	000002	DMKCMD	
DMKCQYCA	000002	DMKCMD	
DMKCQYCL	000002	DMKCMD	
DMKCQYCP	000002	DMKCMD	
DMKCQYID	000001	DMKTOD	
DMKCQYIT	000001	DMKTOD	
DMKCQYLM	000002	DMKCMD	
DMKCQYPF	000002	DMKCMD	
DMKCQYRG	000002	DMKCFF	
DMKCQYSA	000002	DMKCMD	
DMKCQYSP	000002	DMKCMD	
DMKCQYS3	000002	DMKCMD	
DMKCQYTI	000002	DMKCMD	
DMKCQYUI	000002	DMKCMD	
DMKCQYUS	000002	DMKCMD	
DMKCQYVS	000002	DMKCMD	
DMKCRMCN	000001	DMKIUB	
DMKCRMIL	000001	DMKIUB	
DMKCRMQS	000001	DMKIUB	
DMKCRMSV	000001	DMKIUB	
DMKCSBLD	000002	DMKCFC	
DMKCSBLF	000006	DMKRSP	
DMKCSBSP	000008	DMKRSP	
DMKCSBVL	000002	DMKCFC	
DMKCSCLD	000002	DMKCSB	
DMKCSCLV	000002	DMKCSB	
DMKCSFBS	000002	DMKCFC	
DMKCSFFL	000002	DMKCFC	

LABEL	COUNT	REFERENCES									
DMKCSFRP	000002	DMKCFC									
DMKCSFSP	000002	DMKCFC									
DMKCSODR	000002	DMKCFC									
DMKCSOSD	000014	DMKAPV	DMKCPJ	DMKCSQ	DMKCSW	DMKCSX	DMKRSP	DMKSPL			
DMKCSOST	000002	DMKCFC									
DMKCSPSP	000002	DMKCFC									
DMKCSQCL	000002	DMKCFC									
DMKCSQFR	000002	DMKCFC									
DMKCSQHL	000002	DMKCFC									
DMKCSRGT	000002	DMKCSP									
DMKCSRPO	000002	DMKCSP									
DMKCSTAG	000002	DMKCFC									
DMKCSUCG	000002	DMKCFC									
DMKCSUCS	000002	DMKCFC									
DMKCSVOG	000002	DMKCFC									
DMKCSVOS	000002	DMKCFC									
DMKCSVPG	000002	DMKCFC									
DMKCSVPS	000002	DMKCFC									
DMKCSWCH	000002	DMKCSU									
DMKCSXTG	000002	DMKCFC									
DMKCSXTS	000002	DMKCFC									
DMKCSYTR	000002	DMKCSX									
DMKCVT	000001	DMKSYM									
DMKCVTAB	000112	DMKACO	DMKACS	DMKBLD	DMKCFJ	DMKCFQ	DMKCFR	DMKCFU	DMKCPO	DMKCQT	DMKDID
		DMKDIF	DMKDRD	DMKDRE	DMKENT	DMKEXT	DMKGRF	DMKIOE	DMKIOH	DMKIUJ	DMKIUS
		DMKJRL	DMKLOK	DMKLOM	DMKMCH	DMKMNI	DMKMNJ	DMKMNL	DMKMON	DMKMOO	DMKNEA
		DMKPGM	DMKPGS	DMKPTS	DMKQCP	DMKREI	DMKRGA	DMKRGB	DMKSCH	DMKSEL	DMKSPS
		DMKSTP	DMKTMR	DMKTRQ	DMKVCV	DMKVDA	DMKVDR	DMKVDS	DMKVDT	DMKVMC	DMKVRR
DMKCVTBD	000388	DMKCAC	DMKCDB	DMKCDM	DMKCFG	DMKCFV	DMKCKT	DMKCKV	DMKCPJ	DMKCPN	DMKCPO
		DMKCPT	DMKCQG	DMKCQH	DMKCQI	DMKCQP	DMKCQQ	DMKCQR	DMKCQS	DMKCQU	DMKCQY
		DMKCSU	DMKCSV	DMKCSX	DMKDEF	DMKDEG	DMKDIB	DMKDIF	DMKERM	DMKGRG	DMKHVF
		DMKIOE	DMKJRL	DMKLNM	DMKLOM	DMKMIA	DMKMNJ	DMKMSG	DMKMSW	DMKOPE	DMKPEI
		DMKPEN	DMKPEQ	DMKPER	DMKPST	DMKQCP	DMKRGC	DMKRSE	DMKSAV	DMKSEP	DMKSPL
		DMKSPR	DMKSPS	DMKSRM	DMKTCS	DMKTCT	DMKTHI	DMKTRT	DMKTTX	DMKURS	DMKUSQ
		DMKVCU	DMKVDE	DMKVDF	DMKVFC	DMKWRM	DMKXST				
DMKCVTBH	000984	DMKACO	DMKACR	DMKALO	DMKBLD	DMKCAC	DMKCAO	DMKCCH	DMKCDB	DMKCDM	DMKCDS
		DMKCFD	DMKCFG	DMKCFH	DMKCFO	DMKCFP	DMKCFQ	DMKCFR	DMKCFS	DMKCFT	DMKCFU
		DMKCFY	DMKCKF	DMKCNS	DMKCPB	DMKCPI	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT
		DMKCPU	DMKCPV	DMKCPW	DMKCPY	DMKCPZ	DMKCQC	DMKCQG	DMKCQP	DMKCQQ	DMKCQR
		DMKCQS	DMKCQT	DMKCQY	DMKCST	DMKDAD	DMKDAS	DMKDAU	DMKDEF	DMKDEI	DMKDIA
		DMKDIB	DMKDID	DMKDRD	DMKDRE	DMKDSB	DMKDSP	DMKIOH	DMKJRL	DMKLNK	DMKLNM
		DMKLOG	DMKLOH	DMKLOJ	DMKLOM	DMKMCD	DMKMCH	DMKMCT	DMKMNT	DMKMSW	DMKNEA
		DMKNES	DMKNET	DMKNLD	DMKNLE	DMKOPE	DMKPAH	DMKPEQ	DMKQCP	DMKRGA	DMKRNH
		DMKRSE	DMKRSP	DMKRST	DMKSAV	DMKSEP	DMKSSP	DMKSSS	DMKSSU	DMKSSV	DMKSVD
		DMKTCS	DMKTCT	DMKTHI	DMKTRC	DMKTRD	DMKTRX	DMKURS	DMKUSQ	DMKVCB	DMKVCH
		DMKVDA	DMKVDB	DMKVDD	DMKVDE	DMKVDF	DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVFC
		DMKVFD	DMKVFE	DMKVMA	DMKVMD	DMKVSE	DMKVSG	DMKVSP	DMKVST		
DMKCVTDB	000141	DMKCDB	DMKCDM	DMKCDS	DMKCFC	DMKCFG	DMKCFJ	DMKCFO	DMKCFT	DMKCFU	DMKCFV
		DMKCFY	DMKCQH	DMKCQY	DMKCSB	DMKCSF	DMKCSO	DMKCSP	DMKCST	DMKCSU	DMKCSV
		DMKCSX	DMKDEF	DMKDEG	DMKERM	DMKHVD	DMKMCC	DMKMCD	DMKMCI	DMKMSG	DMKNES
		DMKPEI	DMKPEN	DMKSPT	DMKSRM	DMKTRP	DMKVFD	DMKVFE			
DMKCVTDT	000056	DMKACO	DMKCFH	DMKCFU	DMKCQY	DMKHVC	DMKJRL	DMKLOM	DMKMIA	DMKMID	DMKMNI
		DMKMSG	DMKNLE	DMKQCN	DMKRNH	DMKRSP	DMKSEP	DMKSPL	DMKTOD	DMKTRT	DMKTRU
		DMKUSQ	DMKVME	DMKVSX							
DMKCVTHB	000232	DMKAPS	DMKCAC	DMKCAO	DMKCDB	DMKCDM	DMKCDS	DMKCFC	DMKCFD	DMKCFG	DMKCFJ
		DMKCFS	DMKCFU	DMKCFY	DMKCPB	DMKCPS	DMKCPT	DMKCPV	DMKCPY	DMKCQC	DMKCQG
		DMKCQQ	DMKCQT	DMKCSB	DMKCSF	DMKCSO	DMKCSQ	DMKCSR	DMKCST	DMKDEF	DMKDEG
		DMKDEI	DMKDIA	DMKDIB	DMKGRG	DMKLNK	DMKMCC	DMKMCD	DMKNEA	DMKNES	DMKNET

LABEL	COUNT	REFERENCES									
		DMKNLD	DMKNLE	DMKPEI	DMKRGC	DMKSPT	DMKSSP	DMKTRP	DMKVDC	DMKVER	DMKVFD
		DMKVFE	DMKVMD								
DMKCVUFP	000006	DMKCDB	DMKCDM	DMKVFD							
DMKDADER	000002	DMKIOS	DMKSYM								
DMKDASER	000002	DMKIOS	DMKSYM								
DMKDAUER	000001	DMKIOS									
DMKDDC	000001	DMKDDR									
DMKDDT	000001	DMKDDR									
DMKDEFDG	000002	DMKCFC									
DMKDEFDS	000002	DMKCFC									
DMKDEGIN	000006	DMKDEF									
DMKDEIMS	000002	DMKDEF									
DMKDEXIN	000002	DMKCCW									
DMKDGDA0	000002	DMKAPI	DMKCPI								
DMKDGDA1	000002	DMKAPI	DMKCPI								
DMKDGDA2	000002	DMKAPI	DMKCPI								
DMKDGDA3	000002	DMKAPI	DMKCPI								
DMKDGDA4	000002	DMKAPI	DMKCPI								
DMKDGDA5	000002	DMKAPI	DMKCPI								
DMKDGDA6	000002	DMKAPI	DMKCPI								
DMKDGDA8	000002	DMKAPI	DMKCPI								
DMKDGDA9	000002	DMKAPI	DMKCPI								
DMKDGDDK	000005	DMKAPI	DMKCPI	DMKHVC	DMKSYM						
DMKDGDUL	000002	DMKDGF									
DMKDGFA0	000002	DMKAPI	DMKCPI								
DMKDGFIN	000002	DMKDGD									
DMKDIAIR	000001	DMKIOT									
DMKDIAL	000002	DMKCFC									
DMKDIBCP	000002	DMKCFC									
DMKDIBDR	000010	DMKCFQ	DMKCFR	DMKDIF							
DMKDIBSM	000004	DMKCCW	DMKVCA								
DMKDIDEP	000005	DMKCFU	DMKCPJ	DMKCQS	DMKSYM						
DMKDIDLF	000002	DMKUSQ									
DMKDIDTR	000002	DMKCFU	DMKCPJ								
DMKDIFDI	000002	DMKDIA									
DMKDMPAA	000007	DMKAPI	DMKCPI	DMKCPP	DMKDMQ	DMKMCT	DMKSTA	DMKVRS			
DMKDMPAL	000017	DMKCFU	DMKCQR	DMKDRD	DMKDRE	DMKIDU	DMKPGT	DMKVDG	DMKVDH		
DMKDMPAS	000003	DMKCKF	DMKCPI	DMKRSP							
DMKDMPAU	000005	DMKCFU	DMKDRD	DMKDRE	DMKIDU						
DMKDMPBG	000001	DMKIDU									
DMKDMPCA	000001	DMKCPJ									
DMKDMPCP	000005	DMKCKP	DMKCPU	DMKPMA	DMKSPM	DMKVRR					
DMKDMPC2	000016	DMKAPI	DMKCFG	DMKCKF	DMKCPI	DMKCPP	DMKDMQ	DMKDSP	DMKEXT	DMKFRE	DMKIOT
		DMKMHC	DMKPMA	DMKRSP	DMKVRR	DMKVRS					
DMKDMPDK	000007	DMKPRG	DMKPSA	DMKSVC	DMKSYM						
DMKDMPDP	000001	DMKIDU									
DMKDMPDS	000005	DMKDMQ									
DMKDMPDT	000002	DMKMID	DMKTOD								
DMKDMPDV	000009	DMKCFU	DMKCQR	DMKDMQ	DMKDRD	DMKDRE	DMKIDU	DMKVDG			
DMKDMPG2	000001	DMKPMA									
DMKDMPLK	000001	DMKOPR									
DMKDMPMA	000007	DMKAPI	DMKCPI	DMKCPP	DMKDMQ	DMKMCT	DMKSTA				
DMKDMPMP	000002	DMKCPI	DMKVRS								
DMKDMPPA	000003	DMKAPI	DMKDMQ								
DMKDMPPX	000003	DMKAPI	DMKDMQ								
DMKDMPRC	000013	DMKCFU	DMKCKF	DMKCPI	DMKDRD	DMKDRE	DMKIDU	DMKPGT	DMKRSP	DMKVDG	
DMKDMPRS	000002	DMKCPS									
DMKDMPRY	000009	DMKCPI	DMKCPP	DMKCPS	DMKDMQ	DMKLOJ	DMKVRR				

LABEL	COUNT	REFERENCES									
DMKDMPSA	000006	DMKAPI	DMKCPI	DMKCPP	DMKMCT	DMKSTA	DMKVRS				
DMKDMPSD	000002	DMKCPI	DMKCPS								
DMKDMPSF	000010	DMKCFU	DMKDRD	DMKDRE	DMKIDU						
DMKDMPSI	000001	DMKIDU									
DMKDMPSW	000026	DMKCFU	DMKCQR	DMKDMQ	DMKPGT						
DMKDMPTD	000002	DMKMID	DMKTOD								
DMKDMPTR	000003	DMKCDB	DMKCDM	DMKDMQ							
DMKDMQEN	000002	DMKDMP									
DMKDMQGP	000009	DMKDMP									
DMKDMQLW	000001	DMKDMP									
DMKDMQMC	000001	DMKDMP									
DMKDMQPC	000001	DMKDMP									
DMKDMQSW	000004	DMKDMP									
DMKDMQTL	000003	DMKDMP									
DMKDNC	000001	DMKDDR									
DMKDNT	000001	DMKDDR									
DMKDRDDD	000004	DMKIDU	DMKSPK								
DMKDRDER	000002	DMKHVD									
DMKDRDMP	000002	DMKHVD									
DMKDRDSY	000002	DMKHVD									
DMKDSBRD	000006	DMKCPM	DMKCPW	DMKIOT	DMKSSU	DMKSYM	DMKVDD				
DMKDSBSD	000006	DMKCCS	DMKCPS	DMKSYM							
DMKDSP	000003	DMKWAI									
DMKDSPA	000019	DMKIOS	DMKIOT	DMKPRG	DMKPRV	DMKPRW	DMKSYM	DMKTMR	DMKTRX	DMKVFR	DMKVSJ
DMKDSPB	000030	DMKCFM	DMKCPJ	DMKEXT	DMKHVC	DMKMPO	DMKPMA	DMKPRG	DMKPRV	DMKQVM	DMKSVD
		DMKSYM	DMKTRX								
DMKDSPCH	000311	DMKACO	DMKACR	DMKAPS	DMKAPT	DMKAPY	DMKATS	DMKBIO	DMKCAC	DMKCCH	DMKCCW
		DMKCFG	DMKCFM	DMKCFO	DMKCFQ	DMKCFR	DMKCKF	DMKCKV	DMKCNS	DMKCPB	DMKCPI
		DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPW	DMKCPX	DMKCPZ	DMKCQC	DMKCQT	DMKCRM
		DMKCSB	DMKDAD	DMKDAS	DMKDAU	DMKDGD	DMKDGF	DMKDIA	DMKDIB	DMKDID	DMKDRD
		DMKDRE	DMKDSB	DMKENT	DMKEXT	DMKGIO	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRT
		DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKIDR	DMKIOE	DMKIOF
		DMKIOH	DMKIOS	DMKIOT	DMKIUA	DMKIUC	DMKIUJ	DMKIUN	DMKIUP	DMKIUS	DMKLOC
		DMKLOJ	DMKLOK	DMKLOM	DMKMCH	DMKMCH	DMKMCI	DMKMCT	DMKMHC	DMKMNI	DMKMNL
		DMKMON	DMKMOO	DMKMPO	DMKMSG	DMKNLD	DMKNLE	DMKPAG	DMKPAH	DMKPGM	DMKPGS
		DMKPGT	DMKPRG	DMKPRV	DMKPRW	DMKPSA	DMKPTR	DMKPTS	DMKQCN	DMKQCO	DMKQCP
		DMKREI	DMKRGA	DMKRGB	DMKRNH	DMKRSF	DMKRSP	DMKRST	DMKSCH	DMKSEL	DMKSEP
		DMKSPK	DMKSPS	DMKSSS	DMKSSU	DMKSTR	DMKSVC	DMKSVD	DMKSWA	DMKSYM	DMKTAP
		DMKTCS	DMKTCT	DMKTHI	DMKTMR	DMKTPE	DMKTRD	DMKTRK	DMKTRP	DMKTRQ	DMKTRT
		DMKTRU	DMKTRX	DMKUDR	DMKUNT	DMKUSQ	DMKVAT	DMKVCA	DMKVCB	DMKVCN	DMKVCP
		DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCU	DMKVCW	DMKVCX	DMKVDA	DMKVDC	DMKVDE
		DMKVDR	DMKVDS	DMKVDT	DMKVFC	DMKVFD	DMKVFE	DMKVFR	DMKVFS	DMKVIO	DMKVMA
		DMKVMC	DMKVSE	DMKVSG	DMKVSI	DMKVSJ	DMKVSP	DMKVST	DMKWAI	DMKZTD	
DMKDSPCK	000001	DMKMOO									
DMKDSPE	000003	DMKEXT	DMKSYM								
DMKDSPEC	000001	DMKSYM									
DMKDSPIS	000001	DMKVRS									
DMKDSPIT	000002	DMKMOO	DMKSYM								
DMKDSPNP	000050	DMKCKV	DMKCPP	DMKCPU	DMKCPY	DMKFRE	DMKFRT	DMKHVE	DMKMCC	DMKMIA	DMKMNI
		DMKMNJ	DMKMON	DMKMOO	DMKPGS	DMKPMA	DMKPST	DMKSCH	DMKSEL	DMKSTA	DMKSTP
		DMKSYM	DMKTHI	DMKVFC	DMKVFD	DMKVFE	DMKVMA	DMKVSD	DMKWRM		
DMKDSPN2	000013	DMKCPY	DMKMOO	DMKSCH	DMKSEL	DMKSTA	DMKSTP	DMKSYM	DMKTHI		
DMKDSPPT	000001	DMKMOO									
DMKDSPQI	000005	DMKIOT	DMKQVM	DMKVRR	DMKVRS						
DMKDSPQS	000004	DMKSRM	DMKSTP	DMKSYM							
DMKDSPRQ	000005	DMKCFQ	DMKCPI	DMKMCT	DMKSYM						
DMKDSPRU	000059	DMKCCH	DMKCCS	DMKDGD	DMKDGF	DMKEXT	DMKFRE	DMKLOK	DMKMCH	DMKMHC	DMKPAG
		DMKPRG	DMKPRV	DMKPTR	DMKPTT	DMKSEL	DMKSVC	DMKSVD	DMKSYM	DMKTMR	DMKTRX

LABEL	COUNT	REFERENCES									
		DMKVAT	DMKVAU	DMKVFR	DMKVSJ						
DMKDSPVM	000009	DMKIOT	DMKQVM	DMKVRR							
DMKDSPWA	000002	DMKSYM	DMKWAI								
DMKDSPWI	000002	DMKEXT									
DMKDSP0	000002	DMKAPI	DMKCPI								
DMKDSP1	000002	DMKAPI	DMKCPI								
DMKDSP2	000002	DMKAPI	DMKCPI								
DMKDSP3	000002	DMKAPI	DMKCPI								
DMKDSP4	000002	DMKAPI	DMKCPI								
DMKDSP5	000002	DMKAPI	DMKCPI								
DMKDSP6	000002	DMKAPI	DMKCPI								
DMKEIG80	000001	DMKIOG									
DMKENT	000001	DMKSYM									
DMKENTBS	000002	DMKMNI									
DMKENTEC	000001	DMKMNI									
DMKENTES	000002	DMKMNI									
DMKENTET	000001	DMKMNI									
DMKENTFI	000001	DMKMIA									
DMKENTGP	000001	DMKMNL									
DMKENTGQ	000001	DMKMNL									
DMKENTIM	000005	DMKMNL	DMKMOO	DMKSYM							
DMKENTKC	000001	DMKMID									
DMKENTSC	000001	DMKMNI									
DMKENTSK	000005	DMKIOS	DMKMCD								
DMKENTST	000001	DMKMNI									
DMKENTTB	000002	DMKMNI									
DMKENTTE	000002	DMKMNI									
DMKENTTI	000001	DMKMNI									
DMKENTUT	000003	DMKMCD	DMKMNI								
DMKENT62	000002	DMKMOO									
DMKENT63	000002	DMKMOO									
DMKEPSWD	000004	DMKLNK	DMKLOG								
DMKERMCP	000080	DMKCFC	DMKCFM	DMKCPI	DMKCPJ	DMKEPS	DMKIDU	DMKLOG	DMKOPE	DMKTOD	DMKUDR
		DMKUSQ	DMKWRM								
DMKERMSG	000462	DMKACO	DMKALG	DMKATS	DMKBLD	DMKCAC	DMKCAO	DMKCDB	DMKCDM	DMKCDS	DMKCFC
		DMKCFD	DMKCFF	DMKCFG	DMKCFH	DMKCFJ	DMKCFM	DMKCFO	DMKCFQ	DMKCFR	DMKCFS
		DMKCFT	DMKCFU	DMKCFV	DMKCFW	DMKCFY	DMKCKR	DMKCKS	DMKCKT	DMKCKV	DMKCNS
		DMKCPB	DMKCPJ	DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPU	DMKCPV	DMKCPY	DMKCPZ
		DMKCQC	DMKCQG	DMKCQH	DMKCQI	DMKCQP	DMKCQQ	DMKCQR	DMKCQS	DMKCQT	DMKCQY
		DMKCSB	DMKCSF	DMKCSO	DMKCSP	DMKCSQ	DMKCSR	DMKCST	DMKCSU	DMKCSV	DMKCSX
		DMKDAD	DMKDAS	DMKDEF	DMKDEG	DMKDEI	DMKDIA	DMKDIB	DMKDID	DMKDRD	DMKDRE
		DMKDSP	DMKGRE	DMKIOF	DMKIOG	DMKIOH	DMKJRL	DMKLNM	DMKLOG	DMKLOH	DMKLOJ
		DMKLOM	DMKMCC	DMKMCD	DMKMCI	DMKMIA	DMKMID	DMKMNI	DMKMNJ	DMKMON	DMKMPO
		DMKMSG	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKOPE	DMKPEI	DMKPEL	DMKPEN
		DMKPEQ	DMKPER	DMKPET	DMKPST	DMKQCO	DMKQVM	DMKRGA	DMKRNH	DMKRSE	DMKRSP
		DMKRST	DMKSEP	DMKSNC	DMKSND	DMKSPM	DMKSPS	DMKSPT	DMKSRM	DMKSSS	DMKSST
		DMKSSU	DMKSSV	DMKSWM	DMKTCS	DMKTCT	DMKTHI	DMKTRA	DMKTRP	DMKTRT	DMKTRU
		DMKTRX	DMKUDR	DMKVAT	DMKVCH	DMKVDE	DMKVDF	DMKVDG	DMKVDS	DMKVFC	DMKVFD
		DMKVFE	DMKVMA	DMKVMD	DMKVME	DMKVRR	DMKVSG	DMKVSP	DMKVST	DMKWRM	DMKWRN
		DMKXST									
DMKERP	000001	DMKSAV									
DMKERPCP	000002	DMKCKP									
DMKEXTIN	000002	DMKCPI	DMKSYM								
DMKEXTSP	000002	DMKPSA	DMKSYM								
DMKEXTST	000002	DMKCPO	DMKCPP								
DMKFCBLD	000002	DMKCSC									
DMKFPS	000002	DMKPRG	DMKSYM								
DMKFREAP	000006	DMKCPI	DMKCPP	DMKCPU	DMKPTR	DMKSTA	DMKSYM				

LABEL	COUNT	REFERENCES									
DMKFREDB	000001	DMKDSP									
DMKFREE	000741	DMKACO	DMKACR	DMKACS	DMKALG	DMKALO	DMKAPI	DMKAPS	DMKAPT	DMKAPU	DMKAPW
		DMKAPX	DMKAPY	DMKAPZ	DMKATS	DMKBIO	DMKBLD	DMKBSC	DMKCAC	DMKCAO	DMKCCH
		DMKCCS	DMKCCW	DMKCDB	DMKCDM	DMKCFC	DMKCFD	DMKCFF	DMKCFH	DMKCFJ	DMKCFM
		DMKCFP	DMKCFQ	DMKCFR	DMKCFS	DMKCFT	DMKCFU	DMKCFV	DMKCFY	DMKCKR	DMKCKT
		DMKCKV	DMKCNS	DMKCPB	DMKCPI	DMKCPJ	DMKCPN	DMKCPO	DMKCPS	DMKCPT	DMKCPU
		DMKCPW	DMKCPZ	DMKCQC	DMKCQG	DMKCQH	DMKCQI	DMKCQP	DMKCQQ	DMKCQR	DMKCQS
		DMKCQT	DMKCQU	DMKCQY	DMKCRM	DMKCSB	DMKCSC	DMKCSF	DMKCSO	DMKCSP	DMKCSQ
		DMKCST	DMKCSU	DMKCSV	DMKCSW	DMKCSX	DMKCSY	DMKDAD	DMKDAS	DMKDAU	DMKDEF
		DMKDEG	DMKDEI	DMKDEX	DMKDGD	DMKDGF	DMKDIA	DMKDIB	DMKDID	DMKDRD	DMKDRE
		DMKDSB	DMKDSP	DMKEPS	DMKERM	DMKFRT	DMKGRA	DMKGRC	DMKGRF	DMKGRG	DMKGRI
		DMKGRT	DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKIDR	DMKIDU
		DMKIOE	DMKIOF	DMKIOG	DMKIOH	DMKIOS	DMKISM	DMKIUA	DMKIUB	DMKIUC	DMKIUG
		DMKIUJ	DMKIUN	DMKIUP	DMKIUS	DMKJRL	DMKLNK	DMKLNM	DMKLOC	DMKLOG	DMKLOH
		DMKLOJ	DMKMCC	DMKMCD	DMKMCH	DMKMHC	DMKMHV	DMKMIA	DMKMNI	DMKMNJ	DMKMNL
		DMKMNT	DMKMPO	DMKMSG	DMKMSW	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKOPE
		DMKPAG	DMKPEI	DMKPEL	DMKPEN	DMKPER	DMKPET	DMKPGM	DMKPGS	DMKPGT	DMKPSA
		DMKPST	DMKPTR	DMKPTT	DMKQCN	DMKQCO	DMKQCP	DMKQCQ	DMKREI	DMKRGA	DMKRGB
		DMKRGC	DMKRGE	DMKRNH	DMKRSE	DMKRSF	DMKRSP	DMKRSQ	DMKRST	DMKSCH	DMKSEL
		DMKSND	DMKSPK	DMKSPL	DMKSPM	DMKSPR	DMKSPS	DMKSPT	DMKSRM	DMKSSS	DMKSSU
		DMKSTP	DMKSVD	DMKSWA	DMKSWM	DMKSYM	DMKTAP	DMKTAQ	DMKTCS	DMKTHI	DMKTMR
		DMKTOD	DMKTPE	DMKTRA	DMKTRK	DMKTRP	DMKTRQ	DMKTRT	DMKTRU	DMKTRX	DMKTTX
		DMKTTY	DMKUDR	DMKUDU	DMKURS	DMKUSP	DMKUSQ	DMKVAT	DMKVAU	DMKVBM	DMKVCA
		DMKVCB	DMKVCH	DMKVCN	DMKVCP	DMKVCQ	DMKVCR	DMKVCT	DMKVCU	DMKVCV	DMKVCW
		DMKVCX	DMKVDA	DMKVDC	DMKVDD	DMKVDE	DMKVDG	DMKVDH	DMKVDR	DMKVDS	DMKVDT
		DMKVER	DMKVFC	DMKVFD	DMKVFE	DMKVFS	DMKVMA	DMKVMC	DMKVMD	DMKVME	DMKVMG
		DMKVRR	DMKVSD	DMKVSG	DMKVSI	DMKVSP	DMKVSQ	DMKVST	DMKVSW	DMKVSX	DMKWRM
		DMKXAB	DMKXST	DMKZTD							
DMKFREEA	000005	DMKBLD	DMKSYM	DMKVRR							
DMKFREEP	000289	DMKACO	DMKACS	DMKALG	DMKAPI	DMKAPS	DMKAPT	DMKATS	DMKBLD	DMKCCH	DMKCCS
		DMKCCW	DMKCDS	DMKCFG	DMKCFM	DMKCFO	DMKCFQ	DMKCFR	DMKCKF	DMKCKV	DMKCNS
		DMKCPB	DMKCPI	DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPU	DMKCPV
		DMKCPW	DMKCPX	DMKCQC	DMKCSO	DMKCSR	DMKCSW	DMKCSX	DMKDAD	DMKDAS	DMKDAU
		DMKDGD	DMKDIA	DMKDIB	DMKDID	DMKDRD	DMKDSB	DMKDSP	DMKEXT	DMKGIO	DMKGRG
		DMKGRI	DMKHPT	DMKHPU	DMKHVC	DMKHVE	DMKHVF	DMKIOE	DMKIOF	DMKIOQ	DMKIOS
		DMKIOT	DMKIUA	DMKIUC	DMKIUJ	DMKIUN	DMKIUP	DMKIUS	DMKLOC	DMKLOJ	DMKLOK
		DMKMCC	DMKMCD	DMKMCH	DMKMCI	DMKMCT	DMKMHC	DMKMIA	DMKMID	DMKMNI	DMKMNL
		DMKMON	DMKMPO	DMKNLD	DMKNLE	DMKPGM	DMKPGS	DMKPGT	DMKPRV	DMKPSA	DMKPTR
		DMKPTS	DMKQCO	DMKQCP	DMKQVM	DMKRGA	DMKRGB	DMKRGC	DMKRNH	DMKRPA	DMKRSE
		DMKRSP	DMKRST	DMKSEL	DMKSND	DMKSPK	DMKSPL	DMKSPM	DMKSSS	DMKSSU	DMKSTP
		DMKSTR	DMKSVC	DMKSWA	DMKSWM	DMKSYM	DMKTAP	DMKTHI	DMKTPE	DMKTRD	DMKTRK
		DMKTRT	DMKTRU	DMKTRX	DMKUNT	DMKUSQ	DMKVAT	DMKVCA	DMKVCB	DMKVCN	DMKVCP
		DMKVCU	DMKVCX	DMKVDC	DMKVDD	DMKVDE	DMKVDR	DMKVDS	DMKVFC	DMKVFD	DMKVFE
		DMKVFS	DMKVIO	DMKVMA	DMKVMC	DMKVRR	DMKVSE	DMKVSG	DMKVSI	DMKVSJ	DMKVSP
		DMKVSQ	DMKVST	DMKZTD							
DMKFREHI	000011	DMKCPJ	DMKFRT	DMKMNI	DMKSTA	DMKSTP	DMKSYM	DMKUSP			
DMKFREHP	000002	DMKFRT	DMKSTA								
DMKFREH1	000001	DMKDGD									
DMKFRELG	000002	DMKCPI	DMKSYM								
DMKFRELN	000010	DMKFRT	DMKMOO	DMKSYM							
DMKFRELO	000010	DMKCPJ	DMKFRT	DMKMCH	DMKMNI	DMKSTA	DMKSYM	DMKUSP			
DMKFRELP	000007	DMKAPI	DMKCPP	DMKFRT	DMKSTA	DMKSVC	DMKSYM				
DMKFRELS	000005	DMKFRT	DMKMOO	DMKSYM							
DMKFREMT	000001	DMKFRT									
DMKFREMX	000002	DMKCPI	DMKSYM								
DMKFRENP	000006	DMKFRT	DMKMOO	DMKSYM	DMKUSP						
DMKFREPP	000002	DMKSTA	DMKSYM								
DMKFRERC	000029	DMKGRA	DMKGRG	DMKLOH	DMKQCN	DMKQVM	DMKSYM	DMKVCH	DMKVCN	DMKVDC	DMKVDS

LABEL	COUNT	REFERENCES									
		DMKVDT									
DMKFRESC	000002	DMKMOO	DMKSYM								
DMKFRESP	000002	DMKFRT									
DMKFRESV	000003	DMKFRT	DMKSVC	DMKSYM							
DMKFRESW	000001	DMKDSP									
DMKFRET	000986	DMKACO	DMKACR	DMKACS	DMKAPI	DMKAPS	DMKAPT	DMKAPV	DMKAPW	DMKAPX	DMKAPY
		DMKAPZ	DMKATS	DMKBIO	DMKBLD	DMKBSC	DMKCAC	DMKCAO	DMKCCH	DMKCCS	DMKCCW
		DMKCDB	DMKCDM	DMKCDS	DMKCFC	DMKCFD	DMKCFF	DMKCFG	DMKCFH	DMKCFM	DMKCFP
		DMKCFQ	DMKCFR	DMKCFS	DMKCFT	DMKCFU	DMKCFY	DMKCKV	DMKCNS	DMKCPB	DMKCPI
		DMKCPJ	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPU	DMKCPW	DMKCPZ	DMKCQC
		DMKCQG	DMKCQH	DMKCQI	DMKCQP	DMKCQQ	DMKCQR	DMKCQS	DMKCQT	DMKCQU	DMKCQY
		DMKCRM	DMKCSB	DMKCSC	DMKCSO	DMKCSP	DMKCSQ	DMKCSR	DMKCST	DMKCSU	DMKCSV
		DMKCSW	DMKCSX	DMKCSY	DMKDAD	DMKDAS	DMKDAU	DMKDEF	DMKDEG	DMKDEI	DMKDEX
		DMKDGD	DMKDGF	DMKDIA	DMKDIB	DMKDID	DMKDIF	DMKDRD	DMKDRE	DMKDSB	DMKDSP
		DMKEPS	DMKERM	DMKFPS	DMKGIO	DMKGRA	DMKGRC	DMKGRD	DMKGRF	DMKGRG	DMKGRI
		DMKGRT	DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKIDR	DMKIOE
		DMKIOF	DMKIOH	DMKIOQ	DMKIOS	DMKIOT	DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUG
		DMKIUJ	DMKIUN	DMKIUP	DMKIUS	DMKJRL	DMKLNK	DMKLNM	DMKLOC	DMKLOG	DMKLOH
		DMKLOJ	DMKLOM	DMKMCC	DMKMCD	DMKMCH	DMKMHC	DMKMIA	DMKMNI	DMKMNL	DMKMON
		DMKMSG	DMKMSW	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKOPE	DMKPAH	DMKPEI
		DMKPEL	DMKPEN	DMKPET	DMKPGM	DMKPGS	DMKPGU	DMKPMA	DMKPSA	DMKPST	DMKPTR
		DMKQCN	DMKQCO	DMKQCP	DMKQCQ	DMKQVM	DMKREI	DMKRGA	DMKRGB	DMKRGC	DMKRNH
		DMKRSE	DMKRSF	DMKRSP	DMKRST	DMKSEL	DMKSEP	DMKSND	DMKSPK	DMKSPL	DMKSPM
		DMKSPS	DMKSPT	DMKSRM	DMKSSS	DMKSSU	DMKSSV	DMKSVC	DMKSVD	DMKSWA	DMKSWM
		DMKSYM	DMKTAP	DMKTAQ	DMKTCS	DMKTCT	DMKTHI	DMKTMR	DMKTPE	DMKTRA	DMKTRC
		DMKTRD	DMKTRK	DMKTRP	DMKTRQ	DMKTRT	DMKTRU	DMKTRX	DMKTTX	DMKUDR	DMKUDU
		DMKUNT	DMKURS	DMKUSP	DMKUSQ	DMKVAT	DMKVCA	DMKVCB	DMKVCH	DMKVCN	DMKVCP
		DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCU	DMKVCV	DMKVCW	DMKVCX	DMKVDA	DMKVDC
		DMKVDD	DMKVDE	DMKVDG	DMKVDH	DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVFC	DMKVFD
		DMKVFE	DMKVIO	DMKVMA	DMKVMC	DMKVMD	DMKVME	DMKVMG	DMKVRR	DMKVSC	DMKVSD
		DMKVSG	DMKVSI	DMKVSP	DMKVSQ	DMKVST	DMKVSW	DMKWRM	DMKWRN	DMKXAB	DMKZTD
DMKFRETL	000003	DMKDSP	DMKSYM	DMKUNT							
DMKFRETS	000001	DMKMOO									
DMKFRET1	000002	DMKDSP	DMKUNT								
DMKFREXX	000002	DMKSCH									
DMKFRE14	000001	DMKCPI									
DMKFRE15	000001	DMKCPI									
DMKFRTLT	000002	DMKTMR	DMKTOD								
DMKFRTRS	000005	DMKCKR	DMKSYM	DMKTMR							
DMKFRTSN	000002	DMKFRE									
DMKFRTT	000001	DMKFRE									
DMKFRTTE	000004	DMKPTR	DMKUSP								
DMKFRTTR	000009	DMKPTR	DMKSTA	DMKSYM							
DMKGIOEX	000002	DMKHVC									
DMKGRAOT	000006	DMKGRI	DMKRGB	DMKRGC							
DMKGRCCP	000001	DMKOPE									
DMKGRCHL	000001	DMKOPE									
DMKGRCMR	000001	DMKOPE									
DMKGRCQR	000001	DMKRGA									
DMKGRCQY	000004	DMKGRF	DMKSYM	DMKVCP							
DMKGRCRU	000001	DMKOPE									
DMKGRCSV	000008	DMKHPT	DMKVCT	DMKVCW	DMKVCX						
DMKGRCUP	000015	DMKGRD	DMKGRG	DMKRGB	DMKRGC	DMKSYM					
DMKGRCVM	000001	DMKOPE									
DMKGRCWC	000002	DMKGRD	DMKRGB								
DMKGRDBR	000070	DMKGRE	DMKGRF	DMKGRG	DMKGRI						
DMKGRDCC	000022	DMKGRF	DMKGRG								
DMKGRDIC	000002	DMKCFM	DMKQCO								

LABEL	COUNT	REFERENCES				
DMKGRDSR	000003	DMKGRG				
DMKGRECL	000006	DMKGRD	DMKGRF	DMKGRG		
DMKGREFD	000001	DMKGRF				
DMKGRESM	000002	DMKGRD				
DMKGRF	000001	DMKSYM				
DMKGRFBR	000024	DMKGRD	DMKGRE	DMKGRG		
DMKGRFCF	000013	DMKGRD	DMKGRE	DMKGRG		
DMKGRFEN	000002	DMKCPV	DMKQVM			
DMKGRFIN	000007	DMKDID	DMKGRD	DMKGRI	DMKHPT	DMKHPU	DMKIOT
DMKGRFMT	000001	DMKGRT				
DMKGRGBK	000003	DMKGRF				
DMKGRGBR	000014	DMKGRD	DMKGRE	DMKGRF		
DMKGRGCL	000003	DMKGRF				
DMKGRGCR	000001	DMKGRF				
DMKGRGER	000002	DMKGRF				
DMKGRGPF	000002	DMKGRF				
DMKGRGTI	000003	DMKGRF				
DMKGRGTR	000001	DMKGRF				
DMKGRHIN	000023	DMKGRD	DMKGRF	DMKGRG	DMKSYM	
DMKGRISB	000006	DMKGRD	DMKGRE	DMKGRF	DMKGRG	
DMKGRT	000001	DMKSYM				
DMKGRTAB	000004	DMKGRG	DMKRGC			
DMKGRTAC	000002	DMKGRF	DMKRGC			
DMKGRTAI	000002	DMKGRF	DMKRGC			
DMKGRTB	000008	DMKGRC	DMKGRD	DMKRGA	DMKRGB	
DMKGRTDT	000002	DMKCFM				
DMKGRTFD	000002	DMKGRF	DMKRGB			
DMKGRTFM	000004	DMKGRF	DMKRGB			
DMKGRTFO	000001	DMKGRH				
DMKGRTPF	000005	DMKCFT	DMKCQU	DMKGRG	DMKRGC	
DMKGRTP6	000002	DMKGRF	DMKRGC			
DMKGRTTI	000001	DMKJRL				
DMKGRT12	000005	DMKGRC	DMKGRD	DMKGRG	DMKRGB	DMKRGC
DMKHPSDG	000002	DMKHVD				
DMKHPSHT	000004	DMKCFR	DMKVSJ			
DMKHPSQR	000002	DMKGRD				
DMKHPSQV	000002	DMKVSI				
DMKHPTDI	000010	DMKDIB	DMKEXT	DMKGRF	DMKUSQ	DMKVIO
DMKHPTEX	000002	DMKDSP				
DMKHPTRE	000004	DMKCFP	DMKHPS			
DMKHPUSR	000002	DMKHPS				
DMKHVCAL	000005	DMKPRV	DMKSYM	DMKVIO		
DMKHVCDE	000002	DMKCFC	DMKCQS			
DMKHVCDG	000006	DMKCFC	DMKCQS			
DMKHVCDI	000002	DMKMOO	DMKSYM			
DMKHVCDU	000008	DMKCFC	DMKCQS			
DMKHVCPC	000003	DMKDRD	DMKDRE	DMKSPS		
DMKHVCVI	000001	DMKVIO				
DMKHVDAL	000001	DMKHVC				
DMKHVDIP	000008	DMKCFG	DMKCFH	DMKPGS		
DMKHVDPP	000002	DMKCPJ				
DMKHVEAL	000001	DMKHVC				
DMKHVFAL	000001	DMKHVC				
DMKHVFCR	000002	DMKLOH				
DMKIDRCN	000001	DMKIUB				
DMKIDRFN	000002	DMKIUC				
DMKIDRFX	000002	DMKCRM				
DMKIDRIN	000002	DMKCRM				

LABEL	COUNT	REFERENCES						
DMKIDRSV	000001	DMKIUB						
DMKIDUMP	000002	DMKCPJ						
DMKIDUSF	000002	DMKCPJ						
DMKIOCVT	000009	DMKCKF	DMKCPI	DMKIOE	DMKIOF	DMKRSP		
DMKIOECC	000002	DMKCCH						
DMKIOECE	000007	DMKIOG	DMKIOH					
DMKIOECL	000007	DMKCKF	DMKCPI	DMKIOF	DMKIOG	DMKIOH	DMKRSP	
DMKIOECQ	000005	DMKCKF	DMKCPI	DMKIOF	DMKRSP			
DMKIOECT	000001	DMKIOG						
DMKIOEEP	000002	DMKIOF	DMKIOG					
DMKIOEES	000012	DMKCKF	DMKCPI	DMKIOF	DMKIOG	DMKIOH	DMKRSP	
DMKIOEEX	000002	DMKDAS						
DMKIOEFL	000002	DMKCPJ						
DMKIOEFM	000003	DMKHVD	DMKSYM					
DMKIOEFR	000006	DMKHVE	DMKIOG	DMKIOH				
DMKIOEHS	000004	DMKHVE	DMKIOG	DMKIOH				
DMKIOEID	000003	DMKCKF	DMKCPI	DMKRSP				
DMKIOEIF	000005	DMKIOG	DMKIOH					
DMKIOEIP	000005	DMKCKF	DMKCPI	DMKIOF	DMKRSP			
DMKIOEIQ	000006	DMKCKF	DMKCPI	DMKIOF	DMKRSP			
DMKIOEIR	000001	DMKCFU						
DMKIOEMC	000002	DMKMCH						
DMKIOEMI	000008	DMKCFQ	DMKCFR	DMKDID				
DMKIOEMQ	000005	DMKCKF	DMKCPI	DMKIOF	DMKRSP			
DMKIOEMX	000006	DMKCKF	DMKCPI	DMKIOF	DMKIOG	DMKIOH	DMKRSP	
DMKIOENI	000003	DMKIOF	DMKIOG	DMKIOH				
DMKIOENQ	000001	DMKIOF						
DMKIOERN	000004	DMKRGA	DMKRNH					
DMKIOERP	000001	DMKIOF						
DMKIOERQ	000002	DMKIOF						
DMKIOERR	000011	DMKCNS	DMKDAS	DMKGRF	DMKIOS	DMKSYM	DMKTAP	
DMKIOESD	000004	DMKDSB	DMKRSF					
DMKIOESQ	000001	DMKIOF						
DMKIOESR	000009	DMKCPS	DMKCPW	DMKNES	DMKNET			
DMKIOEST	000016	DMKBSC	DMKCNS	DMKDAS	DMKGRF	DMKRSE	DMKTAP	DMKTAQ
DMKIOETY	000002	DMKIOG	DMKIOH					
DMKIOEVQ	000001	DMKIOF						
DMKIOEVR	000002	DMKVER						
DMKIOFCN	000001	DMKIUB						
DMKIOFC1	000002	DMKIOE						
DMKIOFEP	000003	DMKCKF	DMKCPI	DMKRSP				
DMKIOFIN	000004	DMKIOE						
DMKIOFM1	000002	DMKIOE						
DMKIOFOB	000008	DMKIOE						
DMKIOFST	000002	DMKIOE						
DMKIOFSV	000001	DMKIUB						
DMKIOFVR	000002	DMKIOE						
DMKIOGAP	000005	DMKAPI	DMKCPJ	DMKCPU				
DMKIOGF1	000002	DMKIOE						
DMKIOGF2	000002	DMKIOE						
DMKIOHFR	000002	DMKIOG						
DMKIOJBL	000005	DMKCKF	DMKCPI	DMKIOF	DMKRSP			
DMKIOQDE	000004	DMKIOS						
DMKIOQDH	000002	DMKIOS						
DMKIOQDQ	000006	DMKIOS						
DMKIOQDU	000002	DMKIOS						
DMKIOQFC	000001	DMKIOS						
DMKIOQFP	000004	DMKIOS						

LABEL	COUNT	REFERENCES								
DMKIOQFX	000001	DMKIOS								
DMKIOQQD	000006	DMKIOS								
DMKIOQQU	000002	DMKIOS								
DMKIOQSK	000004	DMKIOS								
DMKIOQUS	000006	DMKIOS								
DMKIOSCB	000010	DMKCFQ	DMKIOQ	DMKIOT	DMKSPM	DMKVIO	DMKVSJ			
DMKIOSC1	000002	DMKIOT								
DMKIOSC3	000002	DMKIOT								
DMKIOSEN	000004	DMKIOT								
DMKIOSER	000002	DMKDSP	DMKIOT							
DMKIOSHA	000011	DMKCFR	DMKCPS	DMKDIA	DMKDID	DMKRGB	DMKSYM			
DMKIOSMQ	000007	DMKACS	DMKIOQ	DMKMNT	DMKSYM					
DMKIOSNM	000006	DMKIOQ	DMKMOO	DMKSYM						
DMKIOSQE	000002	DMKACR								
DMKIOSQR	000121	DMKBIO	DMKCAC	DMKCFQ	DMKCFR	DMKCNS	DMKCPB	DMKCPN	DMKCPS	DMKCPW	DMKCPZ
		DMKCQT	DMKCSB	DMKDIF	DMKDSB	DMKGRD	DMKIOH	DMKMNI	DMKMNL	DMKMON	DMKNLD
		DMKNLE	DMKPAG	DMKPAH	DMKRGB	DMKRNH	DMKRSF	DMKRSP	DMKRST	DMKSEP	DMKSPK
		DMKSPS	DMKSPT	DMKSYM	DMKTAP	DMKTCS	DMKTCT	DMKTRK	DMKUDR	DMKVDC	DMKVDE
		DMKVDR	DMKVDT	DMKZTD							
DMKIOSQV	000013	DMKDGD	DMKDGF	DMKGIO	DMKSYM	DMKTRK	DMKVSI	DMKVSJ			
DMKIOSRC	000002	DMKDSP	DMKIOT							
DMKIOSRH	000002	DMKIOT								
DMKIOSRQ	000002	DMKIOT								
DMKIOSRS	000002	DMKIOT								
DMKIOSRU	000002	DMKIOT								
DMKIOSRW	000004	DMKCFR	DMKCPB	DMKVDR						
DMKIOSST	000002	DMKIOT								
DMKIOSW1	000002	DMKDSP	DMKIOQ							
DMKIOTCT	000002	DMKMOO	DMKSYM							
DMKIOTIN	000005	DMKCPI	DMKMNT	DMKOPE	DMKOPR	DMKSYM				
DMKIOTRC	000003	DMKACS	DMKDID							
DMKISMTR	000002	DMKCCW								
DMKIUACP	000064	DMKAPS	DMKAPT	DMKAPU	DMKAPV	DMKAPW	DMKAPX	DMKAPY	DMKAPZ	DMKBIO	DMKCPJ
		DMKCRM	DMKIDR	DMKIOF	DMKMSG	DMKRPI	DMKVCP	DMKVCT	DMKVCV	DMKVCW	DMKVCX
		DMKVMG								
DMKIUACU	000007	DMKIUJ	DMKIUP	DMKPGS						
DMKIUAEP	000002	DMKPRV								
DMKIUAPD	000068	DMKIUB	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS	
DMKIUAQU	000024	DMKIUB	DMKIUE	DMKIUG	DMKIUL	DMKIUN	DMKIUS				
DMKIUBRK	000002	DMKDSP								
DMKIUBTB	000010	DMKIUC	DMKIUJ	DMKIUN	DMKIUP					
DMKIUCEP	000001	DMKIUA								
DMKIUEEP	000001	DMKIUA								
DMKIUELR	000002	DMKIUL								
DMKIUERC	000002	DMKIUS								
DMKIUGEP	000001	DMKIUA								
DMKIUGGP	000004	DMKIUC								
DMKIUJEP	000001	DMKIUA								
DMKIULEP	000001	DMKIUA								
DMKIULRP	000002	DMKIUS								
DMKIUNEP	000001	DMKIUA								
DMKIUNIN	000012	DMKIUA	DMKIUB	DMKIUC	DMKIUJ	DMKIUP	DMKIUS				
DMKIUPEP	000001	DMKIUA								
DMKIUSEP	000001	DMKIUA								
DMKIUSET	000004	DMKIUB	DMKIUL							
DMKIUSPD	000010	DMKIUE	DMKIUL							
DMKJRLIL	000002	DMKLNM								
DMKJRLLO	000004	DMKLOG								

LABEL	COUNT	REFERENCES									
DMKJRLQU	000002	DMKCMD									
DMKJRLSE	000002	DMKCMD									
DMKJRLSL	000002	DMKLNK									
DMKLNKIN	000002	DMKCFC									
DMKLNKSB	000002	DMKLOJ									
DMKLNKSS	000001	DMKSSS									
DMKLNMSG	000002	DMKLNK									
DMKLOCK	000010	DMKCFU	DMKLNK	DMKMSG							
DMKLOCKD	000086	DMKATS	DMKCFP	DMKCFU	DMKCKS	DMKDEF	DMKGRC	DMKHVE	DMKLNK	DMKLNM	DMKLOH
		DMKMSG	DMKPEI	DMKPEQ	DMKPER	DMKPET	DMKPGS	DMKTRA	DMKTRC	DMKTRD	DMKUDR
		DMKUDU	DMKUSP	DMKUSQ	DMKVCH	DMKVDA	DMKVDD	DMKVDE	DMKVDF		
DMKLOCKQ	000050	DMKATS	DMKCFP	DMKCKS	DMKDEF	DMKGRC	DMKHVE	DMKLNK	DMKLOH	DMKPEI	DMKPEQ
		DMKPER	DMKPGS	DMKTRA	DMKTRC	DMKTRD	DMKUDR	DMKUDU	DMKUSP	DMKUSQ	DMKVCH
		DMKVDA	DMKVDD								
DMKLOCRD	000152	DMKACO	DMKCFU	DMKCKT	DMKCQH	DMKCQI	DMKCSR	DMKCST	DMKCSV	DMKCSW	DMKCSX
		DMKCSY	DMKDRD	DMKDRE	DMKLOH	DMKMIA	DMKNLE	DMKRST	DMKSFB	DMKSPK	DMKSPL
		DMKSPS	DMKSPT	DMKTRT	DMKTRU	DMKVME	DMKVSD	DMKVSE	DMKVSW	DMKXAB	
DMKLOCRQ	000114	DMKACO	DMKCFU	DMKCKT	DMKCQH	DMKCQI	DMKCSR	DMKCST	DMKCSV	DMKCSW	DMKCSX
		DMKCSY	DMKDRD	DMKDRE	DMKLOH	DMKMIA	DMKNLE	DMKRST	DMKSFB	DMKSPK	DMKSPL
		DMKSPS	DMKSPT	DMKTRT	DMKTRU	DMKVME	DMKVSD	DMKVSE	DMKVSW	DMKXAB	
DMKLOGB	000002	DMKALG									
DMKLOGON	000002	DMKCFC									
DMKLOGOP	000004	DMKOPE	DMKVRR								
DMKLOHRC	000002	DMKLOG									
DMKLOJEP	000002	DMKLOG									
DMKLOKCT	000004	DMKDSP	DMKMOO	DMKSYM							
DMKLOKDF	000029	DMKBLD	DMKCCH	DMKCCS	DMKDGD	DMKDGF	DMKDSP	DMKEXT	DMKFRE	DMKIOS	DMKMCH
		DMKMCT	DMKMHC	DMKPRG	DMKPRV	DMKPTR	DMKPTT	DMKSEL	DMKSVD	DMKSYM	DMKTMR
		DMKTRX	DMKVAT	DMKVAU	DMKVFR	DMKVMA	DMKVSJ				
DMKLOKDS	000005	DMKCPP	DMKCPU	DMKMCT	DMKSYM						
DMKLOKFR	000008	DMKCPP	DMKCPU	DMKMCT	DMKMOO	DMKPSA	DMKSYM				
DMKLOKIO	000140	DMKACR	DMKACS	DMKCCH	DMKCCS	DMKCFQ	DMKCFR	DMKCKV	DMKCNS	DMKCPM	DMKCPN
		DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPU	DMKCPW	DMKCPZ	DMKCSO	DMKDEI	DMKDIA
		DMKDIB	DMKDID	DMKDIF	DMKDSB	DMKEXT	DMKFRE	DMKGRI	DMKIOS	DMKIOT	DMKLOH
		DMKLOJ	DMKMCC	DMKMCT	DMKMNT	DMKMOO	DMKNES	DMKNLD	DMKRGA	DMKRSE	DMKRSP
		DMKSSS	DMKSST	DMKSSU	DMKSSV	DMKSVC	DMKVCH	DMKVDA	DMKVDD	DMKVDE	DMKVDR
		DMKVDS	DMKVDT	DMKVSJ							
DMKLOKPS	000003	DMKMCT	DMKSYM								
DMKLOKRL	000014	DMKCPP	DMKCPU	DMKDSP	DMKMCT	DMKMOO	DMKSCH	DMKSYM			
DMKLOKRM	000007	DMKCPP	DMKCPU	DMKMCT	DMKMOO	DMKPSA					
DMKLOKSP	000002	DMKPSA	DMKSYM								
DMKLOKSW	000167	DMKACR	DMKALG	DMKCAO	DMKCFO	DMKCFQ	DMKCFR	DMKCNS	DMKCPJ	DMKCPN	DMKCPO
		DMKCPS	DMKCPU	DMKCPV	DMKCPY	DMKCSX	DMKDIB	DMKDIF	DMKFRE	DMKGRI	DMKHPS
		DMKIUA	DMKIUE	DMKIUL	DMKJRL	DMKLOG	DMKLOH	DMKLOJ	DMKLOM	DMKMIA	DMKMID
		DMKMSG	DMKMSW	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKPGM	DMKQCN	DMKQCO	DMKRGA
		DMKRGB	DMKRNH	DMKSEL	DMKSND	DMKSPL	DMKSPR	DMKSPS	DMKSSS	DMKSSV	DMKSTP
		DMKSTR	DMKSYM	DMKTCS	DMKTCT	DMKTRQ	DMKUDU	DMKUSQ	DMKVCB	DMKVCH	DMKVCP
		DMKVCS	DMKVCT	DMKVCU	DMKVCW	DMKVDA	DMKVDB	DMKVDD	DMKVDE	DMKVDF	DMKVDT
		DMKVMC									
DMKLOKSY	000032	DMKCPP	DMKCPU	DMKDSP	DMKEXT	DMKFRE	DMKIOS	DMKIOT	DMKMCT	DMKMOO	DMKPRV
		DMKPSA	DMKSTK	DMKSVC	DMKSVD	DMKSYM	DMKTMR	DMKTRX	DMKVMA	DMKVSJ	
DMKLOKTR	000014	DMKCPP	DMKCPU	DMKDSP	DMKEXT	DMKMCT	DMKMOO	DMKSYM	DMKTRQ		
DMKLOKVM	000002	DMKPSA	DMKSYM								
DMKLOMSG	000002	DMKLOJ									
DMKLOMSS	000001	DMKSSS									
DMKMCCCL	000008	DMKCFC	DMKMIA								
DMKMCDIN	000002	DMKMCC									
DMKMCDLI	000002	DMKMCC									

LABEL	COUNT	REFERENCES									
DMKMCDSE	000002	DMKMCC									
DMKMCDST	000002	DMKMCC									
DMKMCDTI	000002	DMKMCC									
DMKMCHIN	000002	DMKCPJ	DMKSYM								
DMKMCHLM	000001	DMKMCI									
DMKMCHSE	000002	DMKDSP									
DMKMCHST	000044	DMKACR	DMKCCH	DMKCKR	DMKCKS	DMKCKV	DMKCPI	DMKCPJ	DMKOPE	DMKPAH	DMKSEG
		DMKTOD	DMKVSE	DMKVSG	DMKWRM						
DMKMCIMS	000002	DMKCFU									
DMKMCTAF	000002	DMKCPO	DMKCPP								
DMKMCTFL	000004	DMKCPP									
DMKMCTFS	000003	DMKEXT	DMKSYM								
DMKMCTMA	000005	DMKEXT	DMKSYM								
DMKMCTPF	000002	DMKCPP	DMKCPU								
DMKMCTPR	000005	DMKDSP	DMKEXT	DMKSYM							
DMKMCTPT	000003	DMKMCH	DMKSYM								
DMKMCTST	000003	DMKMCH	DMKSYM								
DMKMCTVM	000001	DMKCPP									
DMKMHCCP	000010	DMKCPU	DMKPST	DMKVFC	DMKXST						
DMKMHCIN	000002	DMKEXT									
DMKMHCLK	000001	DMKMHV									
DMKMHCRE	000003	DMKCFP									
DMKMHCVM	000002	DMKMHV									
DMKMHVSM	000004	DMKHVC	DMKPRV								
DMKMIACC	000006	DMKMCC	DMKMNI	DMKMOO							
DMKMIADL	000002	DMKMCC									
DMKMIAEN	000004	DMKENT									
DMKMIAIN	000004	DMKENT									
DMKMIAKC	000002	DMKENT									
DMKMIAMU	000004	DMKMCC	DMKMCD								
DMKMIARO	000002	DMKMCC									
DMKMIAWO	000002	DMKMON									
DMKMIDNT	000002	DMKTRQ									
DMKMNIDK	000002	DMKMCC	DMKMIA								
DMKMNIFI	000004	DMKENT	DMKMON								
DMKMNISH	000008	DMKCPS	DMKMCD	DMKMIA							
DMKMNIST	000014	DMKMCC	DMKMCD	DMKMID	DMKOPE						
DMKMNITH	000002	DMKMCC									
DMKMNITR	000002	DMKMON									
DMKMNJDS	000002	DMKMIA									
DMKMNJGT	000004	DMKMCC									
DMKMNJSP	000002	DMKMCC									
DMKMNL	000002	DMKMCC	DMKMNI								
DMKMNLCP	000002	DMKENT									
DMKMNLFI	000004	DMKMCC	DMKMNI								
DMKMNLIN	000010	DMKCPZ	DMKMCC	DMKMNI	DMKVDA						
DMKMNLMR	000002	DMKMOO									
DMKMNLTQ	000002	DMKENT									
DMKMNTFB	000001	DMKALO									
DMKMNTIO	000002	DMKCPI									
DMKMONMI	000001	DMKMCC									
DMKMONPR	000040	DMKMNI	DMKMNL	DMKMOO							
DMKMOOTI	000001	DMKMNJ									
DMKMO000	000002	DMKMNI									
DMKMO040	000002	DMKMNI									
DMKMPOEX	000002	DMKEXT									
DMKMPOPX	000002	DMKPRV									
DMKMPORS	000002	DMKPSA									

LABEL	COUNT	REFERENCES
DMKMPOSP	000002	DMKPRV
DMKMSGAL	000002	DMKAPX
DMKMSGCN	000001	DMKIUB
DMKMSGC2	000001	DMKIUB
DMKMSGEC	000002	DMKCFC
DMKMSGIU	000002	DMKQCN
DMKMSGMA	000002	DMKCFC
DMKMSGMG	000002	DMKCFC
DMKMSGMS	000002	DMKAPX
DMKMSGNH	000002	DMKCFC
DMKMSGSM	000004	DMKAPX DMKCFC
DMKMSGSV	000001	DMKIUB
DMKMSGS2	000001	DMKIUB
DMKMSGWN	000002	DMKCFC
DMKMSWR	000021	DMKBSC DMKCNS DMKDAD DMKDAS DMKDAU DMKGRF DMKRSE DMKSYM DMKTAP DMKTAQ
		DMKTPE
DMKNEAAH	000002	DMKCMD
DMKNEADF	000002	DMKVDS
DMKNEADH	000002	DMKCMD
DMKNEARV	000002	DMKDEF
DMKNEAVR	000002	DMKLOJ
DMKNEMOP	000010	DMKPET DMKTRC DMKTRD
DMKNESDS	000002	DMKCMD
DMKNESEP	000002	DMKNET
DMKNESHD	000002	DMKCMD
DMKNESPL	000002	DMKCMD
DMKNESWN	000002	DMKNET
DMKNETAE	000002	DMKCPJ
DMKNETDB	000002	DMKCMD
DMKNETEN	000002	DMKCMD
DMKNETQU	000002	DMKCMD
DMKNETVA	000002	DMKCMD
DMKNLDR	000006	DMKCMD DMKCPJ DMKRNH
DMKNLEMP	000004	DMKCMD DMKRNH
DMKOPEAC	000002	DMKCPJ
DMKOPEDC	000002	DMKCPI
DMKOPEEM	000004	DMKCNS DMKGRF
DMKOPELO	000002	DMKCPJ
DMKOPERC	000002	DMKCPI
DMKOPRWT	000017	DMKCCH DMKCKF DMKCPI DMKDMQ DMKMCH DMKMCT DMKRSP DMKSAV DMKSYM
DMKPAGCC	000001	DMKMOO
DMKPAGDP	000002	DMKSRM
DMKPAGEX	000002	DMKPAH DMKUSP
DMKPAGHI	000001	DMKSTA
DMKPAGIO	000021	DMKCDS DMKPGM DMKPTR DMKRPA DMKSEL DMKSTR DMKSWA DMKSWM DMKSYM
DMKPAGLO	000001	DMKSTA
DMKPAGOU	000001	DMKPAH
DMKPAGPS	000003	DMKMOO DMKSTP DMKSYM
DMKPAGQ	000003	DMKPTR DMKPTS DMKSTR
DMKPAGRD	000002	DMKCPI DMKVDG
DMKPAGSK	000002	DMKPAH DMKUSP
DMKPAGSS	000002	DMKMOO DMKSYM
DMKPAGXS	000002	DMKPAH DMKUSP
DMKPAGXZ	000001	DMKUSP
DMKPAHIO	000001	DMKPAG
DMKPEINT	000002	DMKCFC
DMKPELCH	000004	DMKPEN
DMKPELCR	000006	DMKPEN DMKPER

LABEL	COUNT	REFERENCES									
DMKPELSD	000002	DMKPEI									
DMKPEND	000002	DMKPEI									
DMKPENDA	000006	DMKPER	DMKPET	DMKUSQ							
DMKPENGT	000002	DMKPEI									
DMKPENSV	000002	DMKPEI									
DMKPEQRY	000002	DMKCMD									
DMKPERIL	000010	DMKDSP	DMKMPO	DMKPRG	DMKPRV	DMKSVD					
DMKPETAB	000002	DMKPEI									
DMKPETAL	000002	DMKPER									
DMKPGMEP	000001	DMKSTP									
DMKPGMST	000002	DMKSRM									
DMKPGMUS	000002	DMKSWM									
DMKPGSPO	000016	DMKCFG	DMKCFP	DMKCPB	DMKDEG	DMKMCH	DMKMCT	DMKUSQ			
DMKPGSPP	000006	DMKCFF	DMKCFP	DMKUSQ							
DMKPGSPR	000006	DMKCFF	DMKCPY	DMKRPA							
DMKPGSPS	000006	DMKCFF	DMKCFG								
DMKPGSSS	000002	DMKHVC									
DMKPGTAE	000001	DMKTDK									
DMKPGTAN	000002	DMKTDK	DMKVDG								
DMKPGTAT	000001	DMKTDK									
DMKPGTAW	000002	DMKTDK	DMKVDG								
DMKPGTA0	000002	DMKTDK	DMKVDG								
DMKPGTA4	000002	DMKTDK	DMKVDG								
DMKPGTA5	000002	DMKTDK	DMKVDG								
DMKPGTCG	000002	DMKNLE									
DMKPGTCP	000004	DMKCQS	DMKPST	DMKVDH							
DMKPGTDC	000002	DMKPGM	DMKPGU								
DMKPGTDF	000004	DMKSRM	DMKSTP	DMKSYM							
DMKPGTDG	000002	DMKSPS									
DMKPGTDK	000003	DMKSRM	DMKSYM	DMKVDG							
DMKPGTDL	000005	DMKPST	DMKSRM	DMKVDG	DMKXST						
DMKPGTDM	000006	DMKMOO	DMKPST	DMKSRM	DMKSYM	DMKVDG	DMKXST				
DMKPGTDP	000003	DMKPGM	DMKSRM								
DMKPGTDS	000006	DMKCFU	DMKDRD	DMKDRE	DMKIDU						
DMKPGTDT	000006	DMKCFU	DMKDRD	DMKDRE							
DMKPGTGC	000002	DMKVDH									
DMKPGTMS	000001	DMKPGU									
DMKPGTMT	000003	DMKPGM									
DMKPGTMV	000009	DMKDIB	DMKDIF	DMKLOH	DMKPGM						
DMKPGTMX	000010	DMKPGM	DMKSTP	DMKSTR							
DMKPGTPB	000009	DMKCQS	DMKMOO	DMKPAG	DMKPGM	DMKPST	DMKPTR	DMKSEL	DMKXST		
DMKPGTPC	000003	DMKPGM	DMKPGU	DMKSYM							
DMKPGTPG	000013	DMKCDS	DMKCFG	DMKSEG	DMKSEL	DMKSST	DMKSYM				
DMKPGTPI	000006	DMKCQS	DMKMNI	DMKPST	DMKXST						
DMKPGTPL	000004	DMKSRM	DMKSYM	DMKVDG							
DMKPGTPM	000003	DMKPGM	DMKSWM								
DMKPGTPN	000012	DMKPGM	DMKPST	DMKSRM	DMKSTP	DMKSYM	DMKVDG	DMKXST			
DMKPGTPT	000004	DMKMOO	DMKPST	DMKSYM	DMKXST						
DMKPGTPU	000005	DMKMOO	DMKPGU	DMKPST	DMKXST						
DMKPGTQT	000002	DMKSRM	DMKSTP								
DMKPGTSB	000009	DMKCQS	DMKMOO	DMKPST	DMKPTR	DMKSWA	DMKXST				
DMKPGTSF	000005	DMKMOO	DMKPGM	DMKSWM	DMKSYM						
DMKPGTSG	000048	DMKACO	DMKCSV	DMKCSW	DMKCSY	DMKMIA	DMKRST	DMKSPL	DMKSPS	DMKTRT	DMKTRU
		DMKVME	DMKVSP	DMKVSQ	DMKXAB	DMKXAD					
DMKPGTSI	000006	DMKCQS	DMKMNI	DMKPST	DMKXST						
DMKPGTSL	000004	DMKPST	DMKSRM	DMKXST							
DMKPGTST	000004	DMKMOO	DMKPST	DMKSRM	DMKXST						
DMKPGTSU	000006	DMKMOO	DMKPGM	DMKPGU	DMKPST	DMKXST					

LABEL	COUNT	REFERENCES									
DMKPGTSW	000006	DMKSWA									
DMKPGTTM	000002	DMKSYM	DMKVDG								
DMKPGTTU	000006	DMKPGU	DMKSYM	DMKVDG							
DMKPGTVR	000002	DMKCFU	DMKIDU								
DMKPGT4A	000002	DMKTDK	DMKVDG								
DMKPGT5A	000002	DMKTDK	DMKVDG								
DMKPGT7A	000002	DMKTDK	DMKVDG								
DMKPGT8A	000002	DMKTDK	DMKVDG								
DMKPGT90	000002	DMKPGU	DMKVDG								
DMKPGUAL	000031	DMKATS	DMKPAG	DMKPGM	DMKPGS	DMKPTR	DMKSEL	DMKSTR	DMKSWA	DMKSWM	
DMKPGUBN	000003	DMKCKV	DMKSEG	DMKSYM							
DMKPGUDU	000008	DMKCFU	DMKDRD	DMKDRE	DMKIDU						
DMKPGUPP	000002	DMKSWA									
DMKPGUPR	000016	DMKATS	DMKCFG	DMKHVD	DMKPGM	DMKPGS	DMKRPA	DMKSEL			
DMKPGUSD	000043	DMKDRD	DMKDRE	DMKNLE	DMKRST	DMKSPK	DMKSPS	DMKTRU	DMKVDD	DMKVSP	DMKVSQ
		DMKXAB									
DMKPGUSP	000008	DMKCPP	DMKPGS	DMKRPA	DMKSSV						
DMKPGUSR	000002	DMKSPK									
DMKPGUSW	000014	DMKPTR	DMKSWA	DMKSWM							
DMKPGUVG	000162	DMKACO	DMKCFH	DMKCFS	DMKCKR	DMKCKS	DMKCKV	DMKCQI	DMKCQY	DMKCST	DMKCSV
		DMKCSW	DMKCSY	DMKDRD	DMKDRE	DMKGRC	DMKHVF	DMKIOF	DMKIOG	DMKIOH	DMKMCC
		DMKMIA	DMKNLD	DMKNLE	DMKOPE	DMKPMA	DMKPTT	DMKRSP	DMKRSQ	DMKRST	DMKSEP
		DMKSPK	DMKSPL	DMKSPT	DMKSSS	DMKSST	DMKTCS	DMKTCT	DMKTRU	DMKUDR	DMKUDU
		DMKVDD	DMKVME	DMKVSQ	DMKVSR	DMKVSW	DMKWRM	DMKWRN	DMKXAB	DMKXAD	
DMKPGUVR	000198	DMKCFH	DMKCFS	DMKCKR	DMKCKS	DMKCKV	DMKCQI	DMKCQY	DMKCST	DMKCSU	DMKCSV
		DMKCSW	DMKCSY	DMKDRD	DMKDRE	DMKGRC	DMKHVF	DMKIOF	DMKIOG	DMKIOH	DMKMIA
		DMKMNI	DMKNLD	DMKNLE	DMKOPE	DMKRSP	DMKRSQ	DMKRST	DMKSEP	DMKSPK	DMKSPS
		DMKSSS	DMKSST	DMKTCS	DMKTCT	DMKTRU	DMKUDR	DMKUDU	DMKVDD	DMKVME	DMKVSP
		DMKVSQ	DMKVST	DMKVSW	DMKWRM	DMKWRN	DMKXAB	DMKXAD			
DMKPIALD	000001	DMKCSC									
DMKPIBLD	000001	DMKCSC									
DMKPMA	000002	DMKCFG	DMKSYM								
DMKPMACD	000015	DMKCFG	DMKDEF	DMKLNK	DMKSYM	DMKVDA					
DMKPMACR	000003	DMKVSI									
DMKPMADT	000002	DMKLNK									
DMKPMAEW	000003	DMKEXT	DMKIOT	DMKSYM							
DMKPMAEX	000004	DMKCFP	DMKSYM								
DMKPMAIN	000003	DMKPRG	DMKSYM								
DMKPMAIO	000010	DMKDSP	DMKIOT	DMKMID	DMKPTR	DMKVRR					
DMKPMAI1	000005	DMKCPI	DMKSYM								
DMKPMAI2	000004	DMKCPI	DMKSEG	DMKSYM							
DMKPMAMC	000004	DMKMCH	DMKSYM								
DMKPMARW	000006	DMKDSP	DMKQVM	DMKVRS							
DMKPMASS	000003	DMKEXT									
DMKPMASW	000015	DMKDSP	DMKEXT	DMKMCH	DMKMCT	DMKSYM					
DMKPMATM	000006	DMKQVM	DMKVRR	DMKVRS							
DMKPMAWD	000008	DMKCCW	DMKPRV	DMKPRW							
DMKPMAWT	000004	DMKDSP	DMKSYM								
DMKPMAXF	000008	DMKPRV	DMKSYM								
DMKPMAXS	000007	DMKPRV	DMKPRW	DMKSYM							
DMKPRGCT	000003	DMKMOO	DMKPMA	DMKSYM							
DMKPRGC8	000018	DMKMCC	DMKMCD	DMKMIA	DMKMNI	DMKMNL	DMKMON	DMKSYM			
DMKPRGIN	000005	DMKAPI	DMKCPI	DMKSTA	DMKSYM	DMKTOD					
DMKPRGMC	000035	DMKCPS	DMKDMP	DMKENT	DMKMCC	DMKMCD	DMKMIA	DMKMID	DMKMNI	DMKMNJ	DMKMNL
		DMKMON	DMKMOO	DMKSYM							
DMKPRGMI	000003	DMKMCC	DMKMNI								
DMKPRGMO	000005	DMKMNI	DMKMNJ	DMKMOO							
DMKPRGRF	000003	DMKSVD	DMKSYM								

LABEL	COUNT	REFERENCES									
DMKPRGSM	000031	DMKDSP	DMKHVC	DMKIUA	DMKMPO	DMKPRV	DMKPRW	DMKRPD	DMKSYM	DMKTMR	DMKVAT
		DMKVAU	DMKVFR								
DMKPRGTI	000007	DMKMCD	DMKMNI	DMKMNJ	DMKMNL	DMKMOO					
DMKPRVCA	000001	DMKAPI									
DMKPRVCD	000002	DMKMOO	DMKSYM								
DMKPRVCE	000001	DMKMOO									
DMKPRVCH	000002	DMKMOO	DMKSYM								
DMKPRVCP	000002	DMKMOO	DMKSYM								
DMKPRVCS	000002	DMKMOO	DMKSYM								
DMKPRVCT	000002	DMKMOO	DMKSYM								
DMKPRVDI	000002	DMKMOO	DMKSYM								
DMKPRVEK	000002	DMKMOO	DMKSYM								
DMKPRVEP	000002	DMKMOO	DMKSYM								
DMKPRVIK	000002	DMKMOO	DMKSYM								
DMKPRVIP	000002	DMKMOO	DMKSYM								
DMKPRVLC	000002	DMKMOO	DMKSYM								
DMKPRVLG	000005	DMKPMA	DMKPRG	DMKSYM							
DMKPRVLP	000002	DMKMOO	DMKSYM								
DMKPRVLR	000002	DMKMOO	DMKSYM								
DMKPRVMA	000001	DMKPSA									
DMKPRVMN	000002	DMKMOO	DMKSYM								
DMKPRVMO	000002	DMKMOO	DMKSYM								
DMKPRVMS	000002	DMKMOO	DMKSYM								
DMKPRVNC	000002	DMKMOO	DMKSYM								
DMKPRVPB	000002	DMKMOO	DMKSYM								
DMKPRVPE	000002	DMKMOO	DMKSYM								
DMKPRVPT	000001	DMKMOO									
DMKPRVRR	000002	DMKMOO	DMKSYM								
DMKPRVTC	000002	DMKMOO	DMKSYM								
DMKPRVTE	000002	DMKMOO	DMKSYM								
DMKPRVTP	000001	DMKMOO									
DMKPRVXI	000002	DMKMOO	DMKPRW								
DMKPRVXR	000001	DMKPRW									
DMKPRVXS	000002	DMKMOO	DMKPRW								
DMKPRWIP	000003	DMKPRV	DMKSYM								
DMKPRWSK	000002	DMKAPI	DMKCPI								
DMKPRWTB	000002	DMKPRV									
DMKPRWTP	000003	DMKPRV	DMKSYM								
DMKPRWXK	000002	DMKPRV									
DMKPSA	000002	DMKLD00E									
DMKPSACC	000026	DMKCDS	DMKDGD	DMKDGF	DMKIUE	DMKIUL	DMKVCN	DMKVSR	DMKVSX		
DMKPSADU	000004	DMKAPI	DMKCLK	DMKCPI	DMKSYM						
DMKPSAER	000002	DMKEXT									
DMKPSAFC	000008	DMKHPU	DMKIUE	DMKVMC							
DMKPSAFP	000072	DMKBIO	DMKFPS	DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKIUC	DMKIUL	DMKPRV	DMKPRW
		DMKSFB	DMKTMR	DMKVER	DMKVMC	DMKVMD	DMKVME	DMKXAB			
DMKPSANX	000004	DMKEXT	DMKMOO	DMKSYM							
DMKPSAPO	000066	DMKCCH	DMKCCW	DMKCDB	DMKCDM	DMKCDS	DMKCPB	DMKDSP	DMKGIO	DMKHVD	DMKMPO
		DMKPMA	DMKPRG	DMKPRV	DMKPRW	DMKQVM	DMKTRK	DMKVCN	DMKVIO	DMKVME	DMKVSI
		DMKVSJ	DMKVSP								
DMKPSARR	000002	DMKTRC									
DMKPSARS	000006	DMKTRC									
DMKPSARX	000004	DMKTRC									
DMKPSASC	000040	DMKCDS	DMKCFD	DMKDGD	DMKHPU	DMKIUE	DMKIUL	DMKTRC	DMKVCN	DMKVMC	DMKVSR
		DMKVSX									
DMKPSASP	000090	DMKAPT	DMKBIO	DMKDRD	DMKDRE	DMKFPS	DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKIUA
		DMKIUE	DMKIUL	DMKPRV	DMKPRW	DMKSFB	DMKTMR	DMKTRC	DMKVMC	DMKXAB	
DMKPSAST	000002	DMKMPO	DMKSTA								

LABEL	COUNT	REFERENCES									
DMKPSTIN	000002	DMKIDU									
DMKPTRAN	000328	DMKACO	DMKAPT	DMKATS	DMKBIO	DMKBLD	DMKCCH	DMKCCW	DMKCDB	DMKCDM	DMKCDS
		DMKCFC	DMKCFD	DMKCFF	DMKCFG	DMKCFH	DMKCFM	DMKCFS	DMKCFU	DMKCFV	DMKCKV
		DMKCNS	DMKCPB	DMKCPI	DMKCPM	DMKCPO	DMKCPP	DMKCPS	DMKCPU	DMKCPW	DMKCPY
		DMKCQS	DMKCQY	DMKCSC	DMKCSO	DMKDGD	DMKDRD	DMKDRE	DMKERM	DMKGIO	DMKGRA
		DMKGRC	DMKGRF	DMKGRT	DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKHVD	DMKHVE	DMKHVF
		DMKIDU	DMKIOF	DMKIOG	DMKIOT	DMKISM	DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUL
		DMKLNK	DMKMCC	DMKMHV	DMKMIA	DMKMNJ	DMKNLD	DMKNLE	DMKOPE	DMKPEI	DMKPER
		DMKPGS	DMKPMA	DMKPRG	DMKPRV	DMKPRW	DMKPSA	DMKQCN	DMKQCO	DMKRGC	DMKRPA
		DMKRPD	DMKRSP	DMKRST	DMKSEG	DMKSEP	DMKSFB	DMKSNC	DMKSPK	DMKSPL	DMKSPT
		DMKSRM	DMKSSS	DMKSST	DMKSSV	DMKSVC	DMKSVD	DMKSWA	DMKSYM	DMKTCS	DMKTCT
		DMKTMR	DMKTOD	DMKTRC	DMKTRD	DMKTRK	DMKTRP	DMKTRU	DMKTRX	DMKTTX	DMKUDR
		DMKUDU	DMKVAT	DMKVAU	DMKVBM	DMKVCH	DMKVCN	DMKVCX	DMKVDR	DMKVER	DMKVIO
		DMKVMC	DMKVMD	DMKVME	DMKVSE	DMKVSG	DMKVSI	DMKVSJ	DMKVSP	DMKVSQ	DMKVSR
		DMKVSX	DMKWRM	DMKWRN	DMKXAB						
DMKPTRAQ	000018	DMKATS	DMKCPY	DMKHVE	DMKPGS	DMKSEL	DMKSTR	DMKSWA	DMKVMA		
DMKPTRCO	000003	DMKFRE	DMKSYM								
DMKPTRCS	000003	DMKMOO	DMKSEL	DMKSYM							
DMKPTRCT	000001	DMKSYM									
DMKPTREE	000003	DMKSEL	DMKSTP	DMKSYM							
DMKPTREF	000002	DMKSEL	DMKSYM								
DMKPTREP	000008	DMKPGM	DMKSTR	DMKSWM	DMKVFS						
DMKPTRER	000002	DMKSTP	DMKSTR								
DMKPTRES	000003	DMKSEL	DMKSTP	DMKSYM							
DMKPTRFA	000003	DMKCPI	DMKSEL	DMKSYM							
DMKPTRFC	000002	DMKMOO	DMKSYM								
DMKPTRFF	000003	DMKMOO	DMKSEL	DMKSYM							
DMKPTRFN	000007	DMKMOO	DMKPTT	DMKSEL	DMKSTA	DMKSTR	DMKSWM	DMKSYM			
DMKPTRFP	000003	DMKDSP	DMKSYM								
DMKPTRFQ	000005	DMKPTS	DMKPTT	DMKSEL	DMKSTR	DMKSYM					
DMKPTRFR	000041	DMKCCW	DMKCPU	DMKDGD	DMKFRE	DMKGRE	DMKMCH	DMKPMA	DMKPST	DMKSPM	DMKSYM
		DMKVCN	DMKVFC	DMKVFD	DMKVFE	DMKVRR	DMKVSD	DMKWRM			
DMKPTRF0	000002	DMKMOO	DMKSYM								
DMKPTRF1	000003	DMKPTT	DMKSTA	DMKSYM							
DMKPTRF2	000004	DMKPTT	DMKSEL	DMKSTA	DMKSYM						
DMKPTRIQ	000001	DMKSYM									
DMKPTRLE	000006	DMKBIO	DMKDGD								
DMKPTRLK	000070	DMKACO	DMKAPS	DMKAPT	DMKAPY	DMKATS	DMKBIO	DMKCCW	DMKCFG	DMKCFQ	DMKCFR
		DMKCPI	DMKCPP	DMKCPX	DMKCQH	DMKCQI	DMKCRM	DMKCSR	DMKCST	DMKCSW	DMKCSY
		DMKDGD	DMKDIB	DMKDRD	DMKDRE	DMKHPT	DMKHPU	DMKHVE	DMKIDR	DMKIOF	DMKIOH
		DMKIUJ	DMKMHV	DMKMIA	DMKMNI	DMKMSG	DMKPSA	DMKQCP	DMKREI	DMKRST	DMKSPK
		DMKSYM	DMKTAP	DMKTRC	DMKTRU	DMKVCN	DMKVMC	DMKVME	DMKVSD	DMKVSE	DMKVSG
		DMKXAB									
DMKPTRLX	000003	DMKAPI	DMKCPI	DMKSYM							
DMKPTRNF	000001	DMKSYM									
DMKPTRNS	000006	DMKMOO	DMKSEL	DMKSRM	DMKSTA	DMKSYM					
DMKPTRN2	000007	DMKMOO	DMKPTT	DMKSEL	DMKSTA	DMKSTR	DMKSWM	DMKSYM			
DMKPTROQ	000003	DMKSWA	DMKSWM	DMKSYM							
DMKPTRPL	000002	DMKSYM	DMKUNT								
DMKPTRPM	000001	DMKSYM									
DMKPTRPO	000002	DMKSEL	DMKSYM								
DMKPTRPQ	000002	DMKCPI	DMKSYM								
DMKPTRPR	000004	DMKMOO	DMKSTP	DMKSTR	DMKSYM						
DMKPTRPS	000031	DMKPAG	DMKPGM	DMKPGU	DMKSEL	DMKSWA	DMKSWM	DMKSYM	DMKVSE	DMKVSG	DMKWRM
DMKPTRQC	000002	DMKPTT	DMKSEL								
DMKPTRQ2	000004	DMKPTS	DMKPTT	DMKSEL	DMKSYM						
DMKPTRRC	000014	DMKCFO	DMKMOO	DMKSCH	DMKSEL	DMKSTP	DMKSYM	DMKUSQ			
DMKPTRRF	000004	DMKMOO	DMKSEL	DMKSTP	DMKSYM						

LABEL	COUNT	REFERENCES									
DMKPTRRM	000002	DMKSTA	DMKSYM								
DMKPTRRQ	000004	DMKPAG	DMKPGM	DMKSTR	DMKSYM						
DMKPTRRW	000002	DMKMOO	DMKSYM								
DMKPTRSC	000018	DMKATS	DMKCPP	DMKDSP	DMKMOO	DMKPGS	DMKPTS	DMKSCH	DMKSEL	DMKSTP	DMKSYM
		DMKVMA									
DMKPTRSS	000004	DMKMOO	DMKSEL	DMKSTP	DMKSYM						
DMKPTRSW	000007	DMKMOO	DMKSEL	DMKSWA	DMKSWM	DMKSYM					
DMKPTRS2	000006	DMKSEL	DMKSWA	DMKSWM	DMKSYM						
DMKPTRUL	000481	DMKACO	DMKAPS	DMKAPT	DMKAPY	DMKATS	DMKBIO	DMKCCW	DMKCFC	DMKCFG	DMKCFH
		DMKCFP	DMKCFQ	DMKCFR	DMKCFS	DMKCKR	DMKCKV	DMKCPI	DMKCPO	DMKCPP	DMKCPU
		DMKCPX	DMKCPY	DMKCQH	DMKCQI	DMKCQS	DMKCQY	DMKCRM	DMKCSB	DMKCSC	DMKCSR
		DMKCST	DMKCSV	DMKCSW	DMKCSY	DMKDGD	DMKDIB	DMKDRD	DMKDRE	DMKGRC	DMKHPT
		DMKHPU	DMKHVC	DMKHVE	DMKHVF	DMKIDR	DMKIOF	DMKIOH	DMKISM	DMKIUE	DMKIUJ
		DMKIUL	DMKLNK	DMKLOH	DMKMHC	DMKMIA	DMKMNI	DMKMSG	DMKNLD	DMKNLE	DMKPGS
		DMKQCP	DMKREI	DMKRPA	DMKRPD	DMKRST	DMKSEP	DMKSFB	DMKSNC	DMKSPK	DMKSPS
		DMKSPT	DMKSVC	DMKSYM	DMKTAP	DMKTRC	DMKTRP	DMKTRU	DMKTRX	DMKUNT	DMKVBM
		DMKVCH	DMKVCN	DMKVDD	DMKVDR	DMKVIO	DMKVMA	DMKVMC	DMKVMD	DMKVME	DMKVSD
		DMKVSE	DMKVSG	DMKVSP	DMKVSQ	DMKVST	DMKVSW	DMKWRN	DMKXAB		
DMKPTRUX	000003	DMKAPI	DMKCPI	DMKSYM							
DMKPTRU1	000005	DMKPTS	DMKSEL	DMKSWA	DMKSYM						
DMKPTRWQ	000006	DMKCDS	DMKPAG	DMKPGM	DMKRPA	DMKSEL	DMKSYM				
DMKPTSAD	000067	DMKATS	DMKCDB	DMKCDM	DMKCKF	DMKCPI	DMKCPP	DMKPAG	DMKPGS	DMKPTR	DMKRSP
		DMKSEL	DMKSTR	DMKSWA	DMKSWM	DMKVFD	DMKVFE	DMKVFS	DMKVMA		
DMKPTSAE	000034	DMKDAD	DMKDAS	DMKDAU	DMKPTR	DMKPTT	DMKSEL	DMKSTR	DMKSWM	DMKVAT	DMKVAU
		DMKVFS									
DMKPTSMW	000005	DMKCFU	DMKCPI	DMKCQS	DMKSEL						
DMKPTSPW	000018	DMKATS	DMKCFP	DMKCPP	DMKPGS	DMKUSQ	DMKVAT	DMKVDR	DMKVMC		
DMKPTSRS	000004	DMKPGS	DMKSCH								
DMKPTTAL	000004	DMKCPY	DMKSEG								
DMKPTTCL	000008	DMKPTR	DMKSTR	DMKVFS							
DMKPTTC1	000006	DMKDAD	DMKDAS	DMKDAU							
DMKPTTC2	000006	DMKDAD	DMKDAS	DMKDAU							
DMKPTTC3	000006	DMKDAD	DMKDAS	DMKDAU							
DMKPTTDM	000016	DMKPRV	DMKSTR								
DMKPTTFT	000078	DMKATS	DMKCFQ	DMKCPP	DMKCPU	DMKCPY	DMKDGD	DMKFRT	DMKMCH	DMKPGM	DMKPGS
		DMKPST	DMKPTR	DMKQCO	DMKRPA	DMKSEL	DMKSPM	DMKSWA	DMKSWM	DMKUNT	DMKVCN
		DMKVDR	DMKVFC	DMKVFD	DMKVFE	DMKVFS	DMKVMA				
DMKPTTHL	000003	DMKMOO	DMKSTP	DMKSYM							
DMKPTTLH	000001	DMKSTP									
DMKPTTPM	000018	DMKDGD	DMKDMQ	DMKPTR	DMKSEL	DMKVFD	DMKVFE				
DMKPXA	000009	DMKAPI	DMKFRE	DMKFRT	DMKLOK	DMKMOO	DMKPSA	DMKSYM			
DMKPXACX	000001	DMKMOO									
DMKPXADL	000003	DMKMOO	DMKPSA								
DMKPXADQ	000001	DMKPSA									
DMKPXAFC	000001	DMKMOO									
DMKPXAIP	000001	DMKMOO									
DMKPXAIS	000001	DMKMOO									
DMKPXAMC	000001	DMKMOO									
DMKPXAOP	000001	DMKMOO									
DMKPXAOS	000001	DMKMOO									
DMKPXAPC	000001	DMKMOO									
DMKPXARC	000001	DMKMOO									
DMKPXATL	000003	DMKMOO	DMKPSA								
DMKPXAVS	000002	DMKMOO									
DMKPXAWC	000001	DMKMOO									
DMKPXB	000006	DMKAPI	DMKFRE	DMKFRT	DMKMOO	DMKSYM					
DMKPXBCX	000001	DMKMOO									
DMKPXBDL	000002	DMKMOO									

LABEL	COUNT	REFERENCES									
DMKPXBFC	000001	DMKMOO									
DMKPXBIP	000001	DMKMOO									
DMKPXBIS	000001	DMKMOO									
DMKPXBMC	000001	DMKMOO									
DMKPXBND	000001	DMKLOK									
DMKPXBOP	000001	DMKMOO									
DMKPXBOS	000001	DMKMOO									
DMKPXBPC	000001	DMKMOO									
DMKPXBRC	000001	DMKMOO									
DMKPXBTL	000002	DMKMOO									
DMKPXBVS	000002	DMKMOO									
DMKPXBWC	000001	DMKMOO									
DMKQCNFT	000001	DMKFRT									
DMKQCNPL	000036	DMKCSU	DMKCSV	DMKCSX	DMKDEF	DMKDEG	DMKERM	DMKSPL	DMKTRT	DMKVCN	DMKVDD
DMKQCNWT	000326	DMKACO	DMKACR	DMKAPI	DMKBLD	DMKCAC	DMKCAO	DMKCCH	DMKCDB	DMKCDM	DMKCDS
		DMKCFC	DMKCFD	DMKCFF	DMKCFH	DMKCFM	DMKCFP	DMKCFS	DMKCFU	DMKCFV	DMKCFW
		DMKCLK	DMKCPB	DMKCPJ	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPU	DMKCPV
		DMKCPW	DMKCPY	DMKCPZ	DMKCQC	DMKCQG	DMKCQH	DMKCQI	DMKCQP	DMKCQQ	DMKCQR
		DMKCQS	DMKCQT	DMKCQU	DMKCQY	DMKCSB	DMKCSF	DMKCSO	DMKDAD	DMKDAS	DMKDAU
		DMKDEI	DMKDIA	DMKDIB	DMKDIF	DMKDSB	DMKGRG	DMKJRL	DMKLNM	DMKLOM	DMKMCC
		DMKMCD	DMKMCH	DMKMCI	DMKMCT	DMKMIA	DMKMID	DMKMNI	DMKMNJ	DMKMSG	DMKMSW
		DMKNEA	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKOPE	DMKPEI	DMKPEN	DMKPEQ	DMKPER
		DMKPET	DMKPGM	DMKPGT	DMKPRG	DMKPSA	DMKPTR	DMKQCP	DMKQVM	DMKRGC	DMKRNH
		DMKSPL	DMKSPM	DMKSPR	DMKSPS	DMKSRM	DMKSVD	DMKSYM	DMKTHI	DMKTRA	DMKTRC
		DMKTRD	DMKTRP	DMKTRU	DMKURS	DMKUSQ	DMKVCB	DMKVCH	DMKVCN	DMKVDA	DMKVDB
		DMKVDD	DMKVDR	DMKVDT	DMKVER	DMKVFC	DMKVFD	DMKVFE	DMKVME	DMKVRR	DMKWRM
DMKQCOCL	000049	DMKCFQ	DMKCNS	DMKDIA	DMKDIF	DMKEXT	DMKGRF	DMKGRI	DMKNES	DMKNLD	DMKRGA
		DMKRNH	DMKSYM	DMKUSQ	DMKVCW	DMKVCX					
DMKQCOCS	000001	DMKQCN									
DMKQCOET	000027	DMKCNS	DMKGRF	DMKGRI	DMKRGA	DMKRGB	DMKRGC	DMKRNH	DMKSYM	DMKVCS	DMKVCV
DMKQCOFT	000001	DMKFRT									
DMKQCONQ	000001	DMKQCN									
DMKQCOPL	000002	DMKVCN									
DMKQCORD	000029	DMKACO	DMKCFH	DMKCFM	DMKCFU	DMKCPJ	DMKEPS	DMKMSG	DMKMSW	DMKNLD	DMKNLE
		DMKOPE	DMKSYM	DMKTOD							
DMKQCOSY	000051	DMKAPI	DMKCFP	DMKCFT	DMKCFW	DMKCKR	DMKCKS	DMKCKT	DMKCKV	DMKCLK	DMKCPJ
		DMKDIA	DMKDIB	DMKLOH	DMKOPE	DMKSYM	DMKTOD	DMKWRM			
DMKQCPTO	000022	DMKCNS	DMKGRF	DMKNES	DMKNLD	DMKQCO	DMKRGA	DMKRNH	DMKVCW	DMKVCX	
DMKQCQED	000002	DMKQCN									
DMKQVMCU	000001	DMKIOT									
DMKQVMEP	000003	DMKCFC	DMKSYM								
DMKQVMRT	000001	DMKDSP									
DMKQVMTS	000003	DMKIOT	DMKVRR								
DMKREIPL	000012	DMKDSP	DMKPRG	DMKPTR	DMKVMA						
DMKRETGT	000008	DMKGRG	DMKRGC	DMKTTX	DMKVCU						
DMKRETPT	000010	DMKGRG	DMKRGC	DMKTTX	DMKVCR	DMKVCU					
DMKRGACC	000001	DMKBSC									
DMKRGAIN	000004	DMKIOT	DMKRGB	DMKRGC	DMKSYM						
DMKRGASP	000001	DMKRGC									
DMKRGATM	000001	DMKRGB									
DMKRGBCL	000006	DMKRGA	DMKRGC								
DMKRGBCO	000004	DMKRGA	DMKRGC								
DMKRGBEN	000011	DMKNES	DMKNET	DMKRGA	DMKSYM						
DMKRGBFM	000004	DMKGRT									
DMKRGBIC	000002	DMKQCO	DMKSYM								
DMKRGBMT	000004	DMKRGA	DMKRGC								
DMKRGBPL	000004	DMKRGA									
DMKRGBRE	000004	DMKRGA	DMKRGC								

LABEL	COUNT	REFERENCES									
DMKRGBSL	000008	DMKRGA									
DMKRGBSN	000004	DMKRGA									
DMKRGBUP	000010	DMKRGA	DMKRGC								
DMKRGC	000011	DMKRGA	DMKSYM								
DMKRGDOB	000003	DMKRGA	DMKSYM								
DMKRGDOI	000003	DMKRGB	DMKSYM								
DMKRGEIO	000010	DMKRGA	DMKVCW	DMKVCX							
DMKRGESK	000004	DMKRGA	DMKVCR								
DMKRIOCC	000001	DMKPSA									
DMKRIOCH	000004	DMKPSA	DMKQVM	DMKSSP	DMKSYM						
DMKRIOCN	000015	DMKCKF	DMKCPI	DMKDMQ	DMKGRF	DMKIOT	DMKMCT	DMKOPE	DMKOPR	DMKRSP	DMKSSP
		DMKSYM									
DMKRIOCT	000003	DMKCQP	DMKMNI	DMKPSA							
DMKRIOCU	000006	DMKCPI	DMKPSA	DMKSSP	DMKSYM						
DMKRIODC	000001	DMKPSA									
DMKRIODV	000013	DMKCPI	DMKDMP	DMKIOH	DMKMCT	DMKOPR	DMKPSA	DMKSSP	DMKSYM	DMKWRN	
DMKRIOPR	000009	DMKCPI	DMKCPO	DMKCQR	DMKDMP	DMKPSA	DMKSSP	DMKSYM	DMKWRN		
DMKRIOPU	000005	DMKCPI	DMKPSA	DMKSSP	DMKSYM	DMKWRN					
DMKRIORD	000004	DMKCPI	DMKPSA	DMKSSP	DMKSYM						
DMKRIORN	000020	DMKBLD	DMKCPI	DMKCPJ	DMKCQP	DMKCQT	DMKDIB	DMKIDU	DMKNEA	DMKNES	DMKNET
		DMKRNH	DMKSYM								
DMKRIOSF	000002	DMKCPI	DMKIOH								
DMKRIOUC	000001	DMKPSA									
DMKRNH	000001	DMKCPV									
DMKRNHIC	000002	DMKQCO	DMKSYM								
DMKRNHIN	000005	DMKIOT	DMKNLD	DMKNLE							
DMKRNHND	000028	DMKDIB	DMKDIF	DMKEXT	DMKNES	DMKNET					
DMKRNHTR	000003	DMKNES									
DMKRPAGT	000260	DMKAPT	DMKCFF	DMKCFG	DMKCFH	DMKCFS	DMKCKR	DMKCKS	DMKCKV	DMKCQI	DMKCQY
		DMKCST	DMKCSV	DMKCSW	DMKCSY	DMKDRD	DMKDRE	DMKHVD	DMKHVE	DMKHVF	DMKIOF
		DMKIOG	DMKIOH	DMKMIA	DMKNLD	DMKOPE	DMKRSP	DMKRSQ	DMKRST	DMKSEP	DMKSPK
		DMKSPL	DMKSPS	DMKSSS	DMKSST	DMKSYM	DMKTCS	DMKTCT	DMKTRT	DMKUDR	DMKUDU
		DMKVDD	DMKVME	DMKVSP	DMKVSQ	DMKVST	DMKVSW	DMKVSX	DMKWRM	DMKWRN	DMKXAB
		DMKXAD									
DMKRPAPT	000150	DMKACO	DMKCFG	DMKCFH	DMKCFS	DMKCFU	DMKCKR	DMKCKS	DMKCKV	DMKCST	DMKCSV
		DMKCSW	DMKCSY	DMKDRD	DMKDRE	DMKHVD	DMKHVF	DMKIDU	DMKIOF	DMKIOG	DMKIOH
		DMKMIA	DMKNLE	DMKOPE	DMKRST	DMKSEG	DMKSNC	DMKSPL	DMKSPS	DMKSST	DMKSYM
		DMKTRT	DMKUDU	DMKVME	DMKVSP	DMKVSQ	DMKWRM	DMKWRN	DMKXAB	DMKXAD	
DMKRPASV	000004	DMKVSE	DMKWRM								
DMKRPDEP	000002	DMKHVC									
DMKRPICN	000001	DMKIUB									
DMKRPIIL	000001	DMKIUB									
DMKRPIQS	000001	DMKIUB									
DMKRPIRA	000014	DMKCFC	DMKCSP	DMKCST	DMKCSX	DMKHVF	DMKLNK	DMKUSQ			
DMKRPIRM	000001	DMKIUB									
DMKRPISV	000001	DMKIUB									
DMKRPWEP	000004	DMKLOG									
DMKRSERR	000002	DMKRSP									
DMKRSFPB	000002	DMKRSE									
DMKRSFPR	000002	DMKRSE									
DMKRSFSD	000011	DMKCCS	DMKCPO	DMKCPS	DMKCPW	DMKRSE					
DMKRSPAC	000004	DMKCKF	DMKPSA	DMKSYM	DMKWRM						
DMKRSPCP	000005	DMKCKR	DMKCQH	DMKWRM	DMKWRN						
DMKRSPCR	000007	DMKCKR	DMKCQH	DMKVSD	DMKWRM						
DMKRSPCU	000005	DMKCKR	DMKCQH	DMKWRM	DMKWRN						
DMKRSPCV	000004	DMKCKF	DMKSYM	DMKWRM							
DMKRSPDC	000008	DMKCKR	DMKCKS	DMKCKV	DMKWRM						
DMKRSPDL	000015	DMKCKF	DMKSPK	DMKSYM	DMKURS	DMKWRM	DMKWRN				

LABEL	COUNT	REFERENCES									
DMKRSPDP	000005	DMKCKF	DMKVME								
DMKRSPEC	000003	DMKCKT									
DMKRSPER	000002	DMKIOS	DMKSYM								
DMKRSPEX	000004	DMKCSF	DMKCSO	DMKIOT	DMKSYM						
DMKRSPFC	000006	DMKCKR	DMKCKT	DMKVSG	DMKWRM						
DMKRSPFL	000007	DMKCKT	DMKSPS								
DMKRSPHE	000004	DMKCKW									
DMKRSPHL	000002	DMKCKW									
DMKRSPHQ	000011	DMKCKF	DMKCKV	DMKCQR	DMKCSQ	DMKCSY	DMKSPK	DMKSPL	DMKSYM	DMKWRM	DMKWRN
DMKRSPHT	000002	DMKCKW									
DMKRSPLP	000001	DMKPSA									
DMKRSPMN	000005	DMKCKF	DMKMCC	DMKMIA							
DMKRSPND	000006	DMKCKR	DMKCKT	DMKCKV	DMKWRN						
DMKRSPPC	000006	DMKCKR	DMKCKS	DMKCKT	DMKCKV	DMKWRM	DMKWRN				
DMKRSPPR	000012	DMKCKF	DMKCPI	DMKCQR	DMKPSA	DMKRSE	DMKSYM	DMKVSE	DMKVSG	DMKWRM	DMKWRN
DMKRSPPU	000008	DMKCKF	DMKCQR	DMKPSA	DMKSYM	DMKVSE	DMKVSG	DMKWRM	DMKWRN		
DMKRSPRD	000008	DMKCKF	DMKDMP	DMKPSA	DMKSYM	DMKWRM	DMKWRN				
DMKRSPSC	000014	DMKCKR	DMKCKS	DMKCKT	DMKCKV	DMKCSQ	DMKIDU	DMKVSE	DMKVSF	DMKWRM	DMKWRN
DMKRSPSF	000002	DMKCKF	DMKCPI								
DMKRSPSM	000007	DMKCKT	DMKVSF	DMKWRN							
DMKRSPSP	000002	DMKCKF	DMKPSA								
DMKRSPST	000013	DMKCKR	DMKCKS	DMKCKT	DMKCKV	DMKWRM	DMKWRN				
DMKRSPSV	000002	DMKCKW									
DMKRSPSW	000008	DMKCKR	DMKCKS	DMKCKT	DMKCKV	DMKWRN					
DMKRSPTA	000002	DMKCKV									
DMKRSPTC	000004	DMKCKV									
DMKRSPTP	000012	DMKCKR	DMKCKS	DMKCKT	DMKCKV	DMKWRM	DMKWRN				
DMKRSPTR	000007	DMKCKF	DMKTRT	DMKTRU							
DMKRSPWA	000002	DMKCKF	DMKCPJ								
DMKRSPWC	000004	DMKCPJ	DMKWRM								
DMKRSPWI	000004	DMKCKW									
DMKRSP83	000002	DMKRSE	DMKSYM								
DMKRSP9P	000002	DMKCKT									
DMKRSQDC	000002	DMKRSP									
DMKRSQFR	000002	DMKRSP									
DMKRSTIN	000002	DMKRSP									
DMKSAV	000014	DMKCKD	DMKCKP	DMKSEG	DMKSTA	DMKVRR	DMKVRS				
DMKSAVRS	000002	DMKCKP									
DMKSBLTR	000002	DMKSEP									
DMKSCHAL	000005	DMKMON	DMKSTP	DMKSYM	DMKTRQ						
DMKSCHAP	000002	DMKSTP	DMKSYM								
DMKSCHBS	000003	DMKMOO	DMKSTP								
DMKSCHCA	000003	DMKCPP	DMKSTP	DMKTHI							
DMKSCHCO	000002	DMKSTP	DMKTHI								
DMKSCHCT	000002	DMKMOO	DMKSYM								
DMKSCHCU	000003	DMKCPP	DMKSTP	DMKTHI							
DMKSCHDL	000063	DMKACO	DMKALG	DMKDSP	DMKHPT	DMKIOS	DMKPTR	DMKQCP	DMKRPA	DMKSTP	DMKSTR
		DMKSYM	DMKTRQ	DMKUSQ	DMKVCA	DMKVFR	DMKVMC	DMKVRR	DMKVSI		
DMKSCHEL	000002	DMKTHI									
DMKSCHET	000001	DMKSTP									
DMKSCHFS	000001	DMKSTP									
DMKSCHIB	000003	DMKSRM	DMKSYM								
DMKSCHIS	000003	DMKSTP	DMKTHI								
DMKSCHKA	000003	DMKSTP									
DMKSCHLI	000003	DMKCPP	DMKSTP	DMKTHI							
DMKSCHMF	000001	DMKMON									
DMKSCHMS	000004	DMKMOO	DMKPGM	DMKSTR							
DMKSCHNS	000003	DMKSTP	DMKTHI								

LABEL	COUNT	REFERENCES									
DMKSCHN1	000004	DMKMON	DMKMOO	DMKPTR	DMKSYM						
DMKSCHN2	000003	DMKMOO	DMKPTR	DMKSYM							
DMKSCHPB	000004	DMKSRM	DMKSTP	DMKSYM							
DMKSCHPG	000004	DMKCFU	DMKCQR	DMKSTP	DMKSYM						
DMKSCHPU	000003	DMKMON	DMKMOO	DMKSYM							
DMKSCHQB	000003	DMKSRM	DMKSTP								
DMKSCHQC	000003	DMKSRM	DMKSTP								
DMKSCHQT	000004	DMKMOO	DMKSTP	DMKTHI							
DMKSCHQ1	000011	DMKDSP	DMKFPS	DMKMON	DMKMOO	DMKPRG	DMKQVM	DMKSTP	DMKSYM	DMKTMR	DMKTRQ
DMKSCHQ2	000012	DMKDSP	DMKFPS	DMKMOO	DMKPRG	DMKQVM	DMKSTP	DMKSYM	DMKTHI	DMKTMR	DMKTRQ
DMKSCHQ3	000001	DMKMOO									
DMKSCHRL	000006	DMKDSP	DMKSTP	DMKSYM	DMKTHI	DMKTRQ					
DMKSCHRT	000103	DMKCFG	DMKCFJ	DMKCFM	DMKCFP	DMKCFQ	DMKCFR	DMKCFU	DMKCFY	DMKDIA	DMKDIB
		DMKDIF	DMKGRI	DMKHPT	DMKIOQ	DMKLOH	DMKLOJ	DMKMCC	DMKMCD	DMKMNI	DMKPMA
		DMKQCP	DMKQVM	DMKRGA	DMKRGB	DMKRGC	DMKSND	DMKSSU	DMKSYM	DMKTMR	DMKUSQ
		DMKVCR	DMKVCS	DMKVCT	DMKVCV	DMKVCX	DMKVRR				
DMKSCHSC	000001	DMKSTP									
DMKSCHSM	000001	DMKSTP									
DMKSCHST	000093	DMKCFJ	DMKCFQ	DMKCFR	DMKCFU	DMKCPJ	DMKCPO	DMKCQT	DMKDID	DMKDRD	DMKDRE
		DMKENT	DMKGRF	DMKIUJ	DMKIUS	DMKJRL	DMKMCH	DMKMHC	DMKMID	DMKMNI	DMKMNJ
		DMKMNL	DMKMOO	DMKPGS	DMKPMA	DMKQCP	DMKQVM	DMKREI	DMKRGA	DMKRGB	DMKSSU
		DMKSTP	DMKSYM	DMKTMR	DMKTOD	DMKVCV	DMKVDA	DMKVDR	DMKVDS	DMKVMC	DMKVRR
DMKSCHS1	000002	DMKSTP	DMKTHI								
DMKSCHS2	000002	DMKSTP	DMKTHI								
DMKSCHTQ	000011	DMKCPP	DMKDSP	DMKEXT	DMKPMA	DMKSYM	DMKTRQ				
DMKSCHUC	000001	DMKSTP									
DMKSCHWX	000002	DMKSRM									
DMKSCHW1	000003	DMKMON	DMKMOO	DMKSYM							
DMKSCHW2	000003	DMKMON	DMKMOO	DMKSYM							
DMKSCNAU	000158	DMKACO	DMKCAO	DMKCFD	DMKCFO	DMKCFU	DMKCFY	DMKCPV	DMKCPY	DMKCQG	DMKCQQ
		DMKCQR	DMKCQS	DMKCQU	DMKCQY	DMKCSP	DMKCSW	DMKCSX	DMKDIA	DMKDIB	DMKHVF
		DMKIUC	DMKJRL	DMKLNK	DMKLOH	DMKMIA	DMKMSG	DMKNEA	DMKPGM	DMKQCN	DMKQCO
		DMKRNH	DMKSND	DMKSPL	DMKSPR	DMKSPS	DMKSSS	DMKSSV	DMKSWM	DMKTCS	DMKTCT
		DMKTRP	DMKUDU	DMKUSQ	DMKVCH	DMKVDA	DMKVDC	DMKVDD	DMKVMC	DMKVSD	DMKVSG
DMKSCNDC	000014	DMKCKV	DMKRSP	DMKRSQ	DMKSPK	DMKVSX					
DMKSCNEP	000034	DMKCNS	DMKCPJ	DMKCPO	DMKCPS	DMKCPZ	DMKCQP	DMKCQQ	DMKDID	DMKEXT	DMKIDU
		DMKMCT	DMKMNL	DMKOPE	DMKVCH	DMKVDT					
DMKSCNFD	000750	DMKCAC	DMKCAO	DMKCDB	DMKCDM	DMKCDS	DMKCFC	DMKCFD	DMKCFG	DMKCFH	DMKCFJ
		DMKCFM	DMKCFO	DMKCFS	DMKCFT	DMKCFU	DMKCFV	DMKCFW	DMKCFY	DMKCPB	DMKCPS
		DMKCPT	DMKCPV	DMKCPY	DMKCQC	DMKCQG	DMKCQH	DMKCQP	DMKCQQ	DMKCQR	DMKCQS
		DMKCQT	DMKCQY	DMKCSB	DMKCSF	DMKCSO	DMKCSP	DMKCSQ	DMKCSR	DMKCST	DMKCSU
		DMKCSV	DMKCSX	DMKDEF	DMKDEG	DMKDEI	DMKDIA	DMKDIB	DMKEPS	DMKJRL	DMKLNK
		DMKLOG	DMKMCC	DMKMCD	DMKMCI	DMKMIA	DMKMNJ	DMKMSG	DMKNEA	DMKNES	DMKNET
		DMKNLD	DMKNLE	DMKPEI	DMKPEN	DMKPEQ	DMKPET	DMKQVM	DMKRST	DMKSND	DMKSPM
		DMKSPT	DMKSRM	DMKSWM	DMKTHI	DMKTRA	DMKTRP	DMKVCT	DMKVDC	DMKVFC	DMKVFD
		DMKVFE	DMKVMD								
DMKSCNMI	000002	DMKAPI	DMKMNT								
DMKSCNMU	000068	DMKACR	DMKACS	DMKCAC	DMKCAO	DMKCPJ	DMKCPM	DMKCPO	DMKCPS	DMKCPV	DMKCQP
		DMKCQR	DMKCQY	DMKDRD	DMKDRE	DMKIDU	DMKIOH	DMKNEA	DMKOPE	DMKPAG	DMKPAH
		DMKUDR	DMKVCH	DMKVSE	DMKVSG						
DMKSCNPH	000022	DMKACS	DMKCFU	DMKCPN	DMKCPW	DMKCQQ	DMKIOS	DMKIOT	DMKMNL	DMKMNT	DMKURS
DMKSCNRA	000058	DMKACS	DMKCAC	DMKCAO	DMKCFU	DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPS	DMKCPV
		DMKCPW	DMKCPZ	DMKCQP	DMKCQQ	DMKDID	DMKIDU	DMKMNL	DMKVCH	DMKVDT	
DMKSCNRD	000214	DMKACO	DMKACS	DMKALO	DMKBLD	DMKCAC	DMKCFQ	DMKCFR	DMKCFS	DMKCFU	DMKCKF
		DMKCKS	DMKCNS	DMKCPI	DMKCPO	DMKCPV	DMKCPZ	DMKCQG	DMKCQQ	DMKCQR	DMKCQS
		DMKCQT	DMKCQY	DMKCSB	DMKDIA	DMKDIB	DMKDID	DMKDIF	DMKDMQ	DMKDRD	DMKDRE
		DMKEXT	DMKGRI	DMKIOE	DMKIOQ	DMKIOS	DMKJRL	DMKLNK	DMKLNM	DMKLOH	DMKLOM
		DMKMNL	DMKMNT	DMKMOO	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKPAH	DMKPMA

LABEL	COUNT	REFERENCES									
		DMKQCP	DMKQVM	DMKRGA	DMKRGB	DMKRSP	DMKSEP	DMKSSV	DMKTCS	DMKTCT	DMKTHI
		DMKTRD	DMKURS	DMKUSQ	DMKVDD	DMKVDE	DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVSE
		DMKVSG									
DMKSCNRN	000046	DMKCPN	DMKCPO	DMKCPT	DMKCPV	DMKCQG	DMKCQP	DMKDIA	DMKDIB	DMKLOH	DMKLOM
		DMKMSW	DMKQCP	DMKTRD	DMKUSQ	DMKVDA	DMKVDD	DMKVDE	DMKVDF	DMKVDR	
DMKSCNRU	000260	DMKACR	DMKCAC	DMKCAO	DMKCCT	DMKCFD	DMKCFG	DMKCFU	DMKCKF	DMKCKR	DMKCKT
		DMKCKV	DMKCNS	DMKCPI	DMKCPS	DMKCPT	DMKCPV	DMKCPW	DMKCQP	DMKCQQ	DMKCQR
		DMKCQT	DMKCSB	DMKCSF	DMKCSO	DMKDEI	DMKDIA	DMKDIB	DMKDID	DMKDIF	DMKDMP
		DMKDMQ	DMKDSB	DMKGRF	DMKGRG	DMKHVE	DMKIOG	DMKIOH	DMKIOQ	DMKIOS	DMKIOT
		DMKLOJ	DMKLOM	DMKMCC	DMKMCD	DMKMNT	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKOPR
		DMKPAH	DMKPMA	DMKQVM	DMKRGA	DMKRNH	DMKRSP	DMKSEG	DMKSPT	DMKSSS	DMKSST
		DMKSSV	DMKSYM	DMKVCH	DMKVDA	DMKVDC	DMKVDD	DMKVDE	DMKVDF	DMKVDG	DMKVDS
		DMKVDT	DMKVRS	DMKVSJ	DMKWRM						
DMKSCNVD	000056	DMKACO	DMKALG	DMKCCW	DMKCFG	DMKCFP	DMKCFR	DMKCFT	DMKCPS	DMKCQQ	DMKCST
		DMKDEF	DMKDIA	DMKDIB	DMKGIO	DMKHVE	DMKLOJ	DMKPMA	DMKQVM	DMKSSU	DMKUNT
		DMKUSQ	DMKVIO	DMKVRS	DMKVST						
DMKSCNVN	000018	DMKCPS	DMKCQG	DMKCST	DMKDEF	DMKLNM	DMKLOJ	DMKTRD	DMKVDD		
DMKSCNVS	000042	DMKATS	DMKCFG	DMKCFH	DMKCPJ	DMKCPP	DMKHVD	DMKHVF	DMKLNK	DMKLOJ	DMKMNT
		DMKNLD	DMKOPE	DMKSEG	DMKSNC	DMKSSS	DMKTCS	DMKVDA			
DMKSCNVU	000216	DMKAPI	DMKBIO	DMKCCH	DMKCCW	DMKCFD	DMKCFG	DMKCFH	DMKCPB	DMKCPI	DMKCPN
		DMKCPS	DMKCQG	DMKCQP	DMKCQQ	DMKCSB	DMKCSQ	DMKCSR	DMKCST	DMKDEF	DMKDGD
		DMKDIA	DMKDIB	DMKDIF	DMKDRD	DMKDSP	DMKFPS	DMKGIO	DMKHVC	DMKHVE	DMKHVF
		DMKLNK	DMKLOJ	DMKLOM	DMKNEA	DMKNET	DMKNLD	DMKPMA	DMKPRV	DMKQCN	DMKQCO
		DMKRGA	DMKSSS	DMKTHI	DMKTRD	DMKUDR	DMKVCA	DMKVCB	DMKVCH	DMKVCN	DMKVDA
		DMKVDC	DMKVDD	DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVIO	DMKVRR	DMKVSI	DMKVSP
		DMKXAB									
DMKSCOLI	000004	DMKLNK	DMKVDS								
DMKSCONP	000006	DMKCPN	DMKCPW	DMKVDA							
DMKSEGPG	000002	DMKCPI									
DMKSEGWR	000002	DMKCPJ									
DMKSELCP	000004	DMKCPY	DMKSTA								
DMKSELCT	000002	DMKPTR									
DMKSELFD	000001	DMKDSP									
DMKSELFE	000002	DMKDSP	DMKPTR								
DMKSELSA	000001	DMKSTA									
DMKSELSB	000001	DMKSTA									
DMKSELSG	000001	DMKSTA									
DMKSELS1	000002	DMKMOO	DMKSYM								
DMKSELVR	000001	DMKCPY									
DMKSEPSP	000002	DMKRSP									
DMKSEPTL	000006	DMKRSP	DMKTCS								
DMKSEQDA	000003	DMKSEP									
DMKSEQLA	000004	DMKSEP									
DMKSEQSA	000003	DMKSEP									
DMKSEV70	000001	DMKIOG									
DMKSFBNS	000002	DMKHVF									
DMKSIX60	000001	DMKIOG									
DMKSNCP	000002	DMKHVD									
DMKSNDNH	000002	DMKCFC									
DMKSNTBL	000008	DMKATS	DMKCFG	DMKCFH	DMKCFS	DMKCPP	DMKCQY				
DMKSNTLA	000002	DMKHVF									
DMKSNTQN	000007	DMKCKV	DMKCPS	DMKCSO	DMKHVD	DMKRSP	DMKTCS	DMKWRM			
DMKSNTRN	000003	DMKNLD	DMKSNC								
DMKSPKDL	000047	DMKAPV	DMKCPO	DMKCPS	DMKCPW	DMKCSO	DMKCSV	DMKCSW	DMKRSP	DMKRST	DMKSPL
		DMKSPS	DMKVDS	DMKVME	DMKVSP	DMKVSQ	DMKVST	DMKVSW			
DMKSPLCR	000002	DMKRST									
DMKSPLCV	000004	DMKVSP	DMKVSQ								
DMKSPLOR	000002	DMKRST									

LABEL	COUNT	REFERENCES								
DMKSPLOV	000002	DMKVST								
DMKSPLSP	000002	DMKMIA								
DMKSPMEP	000002	DMKCFC								
DMKSPROT	000002	DMKSPS								
DMKSPSIO	000001	DMKSPT								
DMKSPTEP	000002	DMKCFC								
DMKSRMQS	000002	DMKCMD								
DMKSRMSS	000002	DMKCMD								
DMKSSSAS	000002	DMKVDA								
DMKSSSCA	000001	DMKSSV								
DMKSSSCV	000004	DMKSSV								
DMKSSSDE	000004	DMKCPW	DMKDEI							
DMKSSSEN	000001	DMKDSB								
DMKSSSHR	000005	DMKSST	DMKSSV							
DMKSSSLN	000002	DMKLNK								
DMKSSSL1	000002	DMKLOJ								
DMKSSSL2	000002	DMKLOJ								
DMKSSSL3	000002	DMKLOJ								
DMKSSSMQ	000020	DMKCFG	DMKCPB	DMKDGD	DMKDSB	DMKHVE	DMKLNK	DMKLOJ	DMKSSU	DMKSSV	DMKVDA
		DMKVSJ								
DMKSSSNS	000005	DMKSST	DMKSSV							
DMKSSSNV	000001	DMKSST								
DMKSSSRL	000004	DMKVDR								
DMKSSSVA	000002	DMKVDA								
DMKSSSVM	000004	DMKSSV	DMKUSQ	DMKVDR						
DMKSSTBL	000002	DMKSSV								
DMKSSTFV	000006	DMKSSS	DMKSSV							
DMKSSUCF	000002	DMKCFQ								
DMKSSUI1	000002	DMKIOS	DMKIOT							
DMKSSUI2	000001	DMKIOT								
DMKSSULO	000004	DMKUSQ	DMKVDR							
DMKSSVHV	000003	DMKHVC								
DMKSSVUS	000002	DMKUSQ								
DMKSTABD	000002	DMKOPE	DMKVRR							
DMKSTANT	000001	DMKCPI								
DMKSTDAT	000001	DMKSTP								
DMKSTKCP	000368	DMKACO	DMKACS	DMKALG	DMKAPT	DMKATS	DMKCCH	DMKCCS	DMKCCW	DMKCFM	DMKCFO
		DMKCFQ	DMKCFR	DMKCKV	DMKCPB	DMKCPJ	DMKCPM	DMKCPO	DMKCPP	DMKCPS	DMKCPV
		DMKCPW	DMKCPX	DMKCQC	DMKDAD	DMKDAS	DMKDAU	DMKDGD	DMKDIB	DMKDSB	DMKDSP
		DMKGRG	DMKHPS	DMKHPT	DMKHPU	DMKHVF	DMKIOE	DMKIOF	DMKIOS	DMKIOT	DMKIUA
		DMKIUC	DMKIUJ	DMKIUN	DMKIUP	DMKLNK	DMKLNM	DMKLOJ	DMKLOM	DMKMCC	DMKMHC
		DMKMIA	DMKMID	DMKMNI	DMKMNL	DMKMON	DMKMPO	DMKMSG	DMKPAG	DMKPAH	DMKPGM
		DMKPGS	DMKPGT	DMKPRV	DMKPTR	DMKPTS	DMKPTT	DMKQCO	DMKQCP	DMKQVM	DMKRGA
		DMKRGB	DMKRNH	DMKRSP	DMKRST	DMKSCH	DMKSEL	DMKSND	DMKSPK	DMKSPM	DMKSSS
		DMKSSU	DMKSSV	DMKSTP	DMKSTR	DMKSYM	DMKTAP	DMKTAQ	DMKTPE	DMKTRD	DMKTRP
		DMKTRT	DMKTRU	DMKTRX	DMKUNT	DMKUSQ	DMKVAT	DMKVCA	DMKVCB	DMKVCP	DMKVCU
		DMKVCW	DMKVCX	DMKVDA	DMKVDR	DMKVIO	DMKVMA	DMKVMC	DMKVSC	DMKVSE	DMKVSG
		DMKVSI	DMKVSP	DMKVSQ	DMKVST						
DMKSTKDE	000033	DMKCCS	DMKDGD	DMKDGF	DMKDSP	DMKPRG	DMKPRV	DMKPTR	DMKPTT	DMKSVD	DMKSYM
		DMKTMR	DMKTRX	DMKVAT	DMKVAU	DMKVFR	DMKVSJ				
DMKSTKIO	000147	DMKACO	DMKACS	DMKCFR	DMKCPM	DMKCPW	DMKCSF	DMKCSO	DMKCSR	DMKCSW	DMKCSX
		DMKDIB	DMKDID	DMKDIF	DMKEXT	DMKHPT	DMKHPU	DMKIOQ	DMKIOS	DMKIOT	DMKNLD
		DMKNLE	DMKPAG	DMKRGA	DMKRGB	DMKRGC	DMKRGE	DMKSPL	DMKSPT	DMKSSU	DMKSYM
		DMKTMR	DMKUNT	DMKVCA	DMKVCB	DMKVCN	DMKVCP	DMKVDD	DMKVIO	DMKVRR	DMKVSI
		DMKVSJ								
DMKSTKLF	000007	DMKFRE	DMKIOQ	DMKSVC	DMKSYM						
DMKSTKMP	000035	DMKACS	DMKCPJ	DMKCPO	DMKDID	DMKEXT	DMKIOT	DMKLOC	DMKMCT	DMKMHC	DMKPAH
		DMKPTR	DMKSYM	DMKVMA	DMKVMC						

LABEL	COUNT	REFERENCES									
DMKSTKOP	000047	DMKACO	DMKCFO	DMKCKF	DMKCPI	DMKCPJ	DMKDID	DMKLOC	DMKMCD	DMKMCH	DMKMCI
		DMKMHC	DMKPAH	DMKPTR	DMKRSP	DMKSYM	DMKTHI	DMKVFC	DMKVFD	DMKVFE	DMKVFS
		DMKVMC									
DMKSTKSW	000005	DMKLOK	DMKSVC	DMKSYM							
DMKSTPC2	000001	DMKCPU									
DMKSTPX	000006	DMKAPI	DMKCPJ	DMKCPU	DMKTRQ						
DMKSTRAN	000010	DMKPGM	DMKPTR	DMKSWM							
DMKSTRSM	000001	DMKSTP									
DMKSVCHI	000001	DMKSTA									
DMKSVCIN	000003	DMKAPI	DMKCPI	DMKSYM							
DMKSVCLO	000001	DMKSTA									
DMKSVCNO	000001	DMKCPI									
DMKSVCNS	000002	DMKCPI	DMKSYM								
DMKSVDIN	000002	DMKSVC									
DMKSWABS	000005	DMKMOO	DMKSRM	DMKSTP							
DMKSWABT	000003	DMKSRM	DMKSTP								
DMKSWAIL	000002	DMKMOO	DMKSYM								
DMKSWAIS	000004	DMKPGM	DMKPTR	DMKSTP	DMKSWM						
DMKSWAOP	000001	DMKSTP									
DMKSWAOS	000002	DMKSTP	DMKSWM								
DMKSWAPD	000001	DMKDSP									
DMKSWAPI	000002	DMKSCH									
DMKSWAPO	000002	DMKSEL									
DMKSWAPR	000008	DMKPGM	DMKPGS	DMKPTR	DMKRPA						
DMKSWAQ1	000002	DMKSRM									
DMKSWAQ2	000002	DMKSRM									
DMKSWART	000002	DMKSTP	DMKTHI								
DMKSWMIG	000002	DMKPGM									
DMKSWMUS	000002	DMKCFC									
DMKSYM	000007	DMKCFU	DMKDRD	DMKDRE	DMKIDU	DMKSEG					
DMKSYSAC	000003	DMKACO	DMKPSA								
DMKSYSAL	000001	DMKSAV									
DMKSYSAP	000014	DMKATS	DMKCDB	DMKCDM	DMKCDS	DMKCFF	DMKCPI	DMKCPU	DMKCPY	DMKMNI	DMKSTA
DMKSYSAT	000015	DMKENT	DMKMCC	DMKMCD	DMKMIA	DMKMID	DMKMNI	DMKMNJ	DMKMON	DMKMOO	DMKOPE
DMKSYSBF	000003	DMKMCC	DMKMIA								
DMKSYSCA	000002	DMKCKT	DMKCPJ								
DMKSYSCD	000001	DMKCFT									
DMKSYSCH	000002	DMKCKT	DMKDMP								
DMKSYSCK	000029	DMKACO	DMKCKF	DMKDIF	DMKDMP	DMKLOM	DMKNEA	DMKVDS	DMKVDT		
DMKSYSCL	000002	DMKMIA	DMKMNJ								
DMKSYSCO	000004	DMKCKR	DMKCKV	DMKCSR	DMKVDS						
DMKSYSCS	000016	DMKCCW	DMKDGD	DMKDMP	DMKDMQ	DMKFRE	DMKFRT	DMKPSA	DMKPTR	DMKPTT	DMKSEL
		DMKSTR	DMKSWA	DMKSWM	DMKSYM						
DMKSYSCT	000004	DMKHVE	DMKIOG								
DMKSYSCV	000001	DMKCPJ									
DMKSYSDP	000004	DMKPGT	DMKVDG								
DMKSYSDS	000002	DMKBLD	DMKCFV								
DMKSYSDT	000004	DMKCFU	DMKCKF	DMKLOM	DMKWRM						
DMKSYSDU	000007	DMKCFU	DMKDRD	DMKDRE	DMKIDU	DMKNLE	DMKSPS	DMKVMD			
DMKSYSDW	000009	DMKCFU	DMKCQY	DMKLOM	DMKMID	DMKTOD	DMKUSQ				
DMKSYSEA	000005	DMKCPJ	DMKHVE	DMKIOG							
DMKSYSEN	000004	DMKMCC	DMKMIA								
DMKSYSER	000008	DMKHVE	DMKIOG	DMKIOH							
DMKSYSES	000001	DMKCFT									
DMKSYSEV	000001	DMKCPJ									
DMKSYSFL	000010	DMKAPU	DMKAPV	DMKCKF	DMKCPI	DMKCSW	DMKRSP	DMKSEP	DMKSPK	DMKSPL	DMKVSQ
DMKSYSFP	000001	DMKSTA									
DMKSYSHE	000007	DMKCKF	DMKCPI	DMKRSP	DMKVSD	DMKVSE	DMKWRM				

LABEL	COUNT	REFERENCES									
DMKSYSHL	000004	DMKCKF	DMKCPI	DMKRSP	DMKWRM						
DMKSYSHT	000007	DMKCKF	DMKCPI	DMKRSP	DMKVSD	DMKVSE	DMKWRM				
DMKSYSID	000001	DMKCPI									
DMKSYSJR	000014	DMKALG	DMKGRT	DMKJRL	DMKLNK	DMKLNM	DMKLOG				
DMKSYSLB	000002	DMKLOC									
DMKSYSLC	000002	DMKPSA	DMKSYM								
DMKSYSLD	000001	DMKCFT									
DMKSYSLE	000002	DMKBLD	DMKCFT								
DMKSYSLG	000007	DMKCFU	DMKCKF	DMKCQY	DMKLOM	DMKWRM					
DMKSYSLL	000004	DMKBLD									
DMKSYSLR	000002	DMKLOC									
DMKSYSLU	000008	DMKCQY	DMKVCT	DMKVCW							
DMKSYSLW	000003	DMKCFU	DMKLOM								
DMKSYSMP	000004	DMKAPI	DMKCPI	DMKTHI							
DMKSYSMS	000001	DMKCFV									
DMKSYSMU	000006	DMKLOJ	DMKLOM								
DMKSYSMX	000005	DMKMCD	DMKMNJ	DMKMON	DMKMOO						
DMKSYSND	000007	DMKCQY	DMKDIB	DMKDIF	DMKMOO	DMKVRR					
DMKSYSNL	000001	DMKSAV									
DMKSYSNM	000015	DMKCQY	DMKGRF	DMKLOJ	DMKLOM	DMKMON	DMKMOO	DMKQCP	DMKSTP	DMKSYM	DMKUSQ
DMKSYSNP	000007	DMKAPI	DMKCPI	DMKCPP	DMKMNI	DMKSTA					
DMKSYSNU	000006	DMKSAV	DMKSEG	DMKSSP							
DMKSYSOC	000018	DMKACR	DMKCFS	DMKCKF	DMKCPI	DMKCPJ	DMKCPO	DMKCPZ	DMKCQY	DMKIDU	DMKMNT
		DMKMOO	DMKPAG	DMKRSP	DMKSCN	DMKSYM	DMKUDR	DMKVDA			
DMKSYSOP	000002	DMKPSA	DMKSYM								
DMKSYSOW	000036	DMKACR	DMKALO	DMKCFS	DMKCFU	DMKCKF	DMKCKV	DMKCPI	DMKCPJ	DMKCPO	DMKCPZ
		DMKCQY	DMKDRD	DMKDRE	DMKIDU	DMKMNT	DMKMON	DMKMOO	DMKPAG	DMKPAH	DMKPGU
		DMKRSP	DMKSCN	DMKSPS	DMKSYM	DMKUDR	DMKVDA	DMKVDG	DMKVSE	DMKVSG	DMKWRM
DMKSYSPC	000011	DMKCCD	DMKCCF	DMKCCO	DMKCCS	DMKCCT	DMKLOH	DMKMHV			
DMKSYSPE	000004	DMKPGT	DMKVDG								
DMKSYSPG	000004	DMKPGM	DMKPGT	DMKPST	DMKVDG						
DMKSYSPL	000004	DMKUDR	DMKUDU								
DMKSYSPM	000003	DMKPGM	DMKUSP	DMKVDG							
DMKSYSPP	000003	DMKPGM	DMKPGT	DMKXST							
DMKSYSPR	000003	DMKCSR	DMKMNT	DMKVDS							
DMKSYSPS	000001	DMKPGT									
DMKSYSPU	000003	DMKCSR	DMKMNT	DMKVDS							
DMKSYSRC	000010	DMKCPY	DMKCQP	DMKDMP	DMKIDU	DMKMCC	DMKMNI	DMKSTA	DMKUNT		
DMKSYSRM	000041	DMKCCH	DMKCCW	DMKCDB	DMKCDM	DMKCDS	DMKCKF	DMKCPI	DMKCPJ	DMKCPY	DMKCQP
		DMKDIB	DMKDMP	DMKERM	DMKFRT	DMKHVD	DMKIDU	DMKMCC	DMKMNI	DMKPTR	DMKRSP
		DMKSEG	DMKSTA	DMKSTP	DMKSYM	DMKTRD	DMKTRU	DMKTRX	DMKUNT	DMKVCA	
DMKSYSRS	000002	DMKSAV	DMKSYM								
DMKSYSRV	000014	DMKCCW	DMKCPJ	DMKCQP	DMKDMP	DMKMCC	DMKMNI	DMKPRW	DMKSEG	DMKSTA	DMKSYM
		DMKUNT									
DMKSYSSF	000002	DMKSEP	DMKVSQ								
DMKSYSSL	000002	DMKCKT									
DMKSYSSP	000035	DMKCKF	DMKCKT	DMKCPI	DMKCPY	DMKDMP	DMKDMQ	DMKLOG	DMKMOO	DMKPSA	DMKRSP
		DMKSCH	DMKSEG	DMKVSE	DMKVSG	DMKWRM					
DMKSYSSW	000014	DMKMOO	DMKPGT	DMKPST	DMKSRM	DMKSTP	DMKSWM	DMKSYM	DMKVDG	DMKVDH	DMKXST
DMKSYSSZ	000022	DMKBLD	DMKCPI	DMKLOG	DMKMOO	DMKPGT	DMKPST	DMKPTR	DMKSEL	DMKSTA	DMKSTP
		DMKSWA	DMKVBM	DMKVDG	DMKVDH	DMKXST					
DMKSYSTD	000004	DMKTDK	DMKVDH	DMKVDS							
DMKSYSTE	000008	DMKCPJ	DMKMCD	DMKMID	DMKMNI	DMKMNJ					
DMKSYSTI	000008	DMKCQY	DMKLOM	DMKMID	DMKTOD	DMKUSQ					
DMKSYSTM	000002	DMKCFU	DMKLOM								
DMKSYSTP	000010	DMKCKF	DMKCPI	DMKCPS	DMKDMP	DMKHVE	DMKRSP	DMKSAV	DMKSSP		
DMKSYSTR	000001	DMKSTA									
DMKSYSTS	000008	DMKCPJ	DMKMCD	DMKMID	DMKMNI	DMKMNJ					

LABEL	COUNT	REFERENCES									
DMKSYSTV	000006	DMKCFQ	DMKCFU	DMKCPJ	DMKCQS	DMKDID					
DMKSYSTZ	000006	DMKHVD	DMKIOE	DMKIOH	DMKSAV	DMKTOD					
DMKSYSUD	000013	DMKCPI	DMKUDR	DMKUDU							
DMKSYSUR	000009	DMKMIA	DMKMNJ	DMKOPE							
DMKSYSVL	000006	DMKALO	DMKCPI	DMKSAV	DMKSEG	DMKSYM					
DMKSYSVM	000009	DMKCKF	DMKCPI	DMKDMP	DMKDMQ	DMKPGT	DMKPSA	DMKRSP	DMKSYM		
DMKSYSWA	000002	DMKCPJ	DMKWRM								
DMKSYSWI	000013	DMKCKF	DMKCKR	DMKCKT	DMKCKV	DMKCPI	DMKRSP	DMKVSD	DMKVSE	DMKVSF	DMKVSG
		DMKWRM	DMKWRN								
DMKSYSWM	000005	DMKCKF	DMKCPI	DMKOPE	DMKRSP	DMKWRM					
DMKSYSWV	000002	DMKCPJ	DMKOPE								
DMKTAPER	000001	DMKIOS									
DMKTAPRL	000002	DMKCPM	DMKCPW								
DMKTAQRE	000002	DMKTAP									
DMKTAQRP	000002	DMKTAP									
DMKTAQSE	000002	DMKTAP									
DMKTBL	000001	DMKSYM									
DMKTBLCI	000002	DMKCNS	DMKTTX								
DMKTBLCO	000003	DMKCNS	DMKTTX	DMKTTY							
DMKTBLGR	000004	DMKCFT	DMKGRI	DMKRGB	DMKRGC						
DMKTBLGT	000003	DMKGRI	DMKRGB	DMKRGC							
DMKTBLPI	000002	DMKCNS	DMKTTX								
DMKTBLPO	000003	DMKCNS	DMKTTX	DMKTTY							
DMKTBLRG	000006	DMKGRI	DMKRGB	DMKRGC							
DMKTBLSF	000004	DMKQCO	DMKQCQ	DMKSEP	DMKVCN						
DMKTBLUP	000007	DMKCNT	DMKGRG	DMKRGC	DMKVCN	DMKVCU					
DMKTBMAI	000002	DMKGRF	DMKRGC								
DMKTBMAO	000001	DMKGRA									
DMKTBMMI	000002	DMKCNS	DMKTTX								
DMKTBMMO	000003	DMKCNS	DMKTTX	DMKTTY							
DMKTBMNI	000002	DMKCNS	DMKTTX								
DMKTBMNO	000003	DMKCNS	DMKTTX	DMKTTY							
DMKTBMTI	000002	DMKGRF	DMKRGC								
DMKTBMTO	000001	DMKGRA									
DMKTBMXI	000002	DMKGRF	DMKRGC								
DMKTBMXO	000001	DMKGRA									
DMKTBMZI	000002	DMKGRF	DMKRGC								
DMKTBMZO	000001	DMKGRA									
DMKTBNAE	000004	DMKCNS	DMKTTX								
DMKTBNAI	000020	DMKCNS	DMKTTX								
DMKTBNAO	000022	DMKCNS	DMKTTX	DMKTTY							
DMKTBNBE	000004	DMKCNS	DMKTTX								
DMKTBNEA	000006	DMKCNS	DMKTTX	DMKTTY							
DMKTBNEB	000006	DMKCNS	DMKTTX	DMKTTY							
DMKTCSCO	000002	DMKRSP									
DMKTCSET	000002	DMKRSP									
DMKTCSML	000006	DMKTCT									
DMKTCSSP	000004	DMKSEP									
DMKTCSTR	000006	DMKRSP									
DMKTCTET	000002	DMKTCS									
DMKTDKGT	000002	DMKVDS									
DMKTDKRL	000004	DMKVDR	DMKVDS								
DMKTEDAT	000004	DMKTEE	DMKTEF	DMKTEM	DMKTES						
DMKTEESF	000001	DMKTEM									
DMKTEFSF	000001	DMKTEE									
DMKTEMEP	000001	DMKTRR									
DMKTESUB	000002	DMKTEE	DMKTEF								
DMKTHIDP	000002	DMKCMD									

LABEL	COUNT	REFERENCES								
DMKTHIFA	000002	DMKCMD								
DMKTHIIO	000002	DMKCMD								
DMKTHILO	000002	DMKCMD								
DMKTHIPA	000002	DMKCMD								
DMKTHIPO	000002	DMKCMD								
DMKTHIQQ	000002	DMKCMD								
DMKTHIUG	000002	DMKCMD								
DMKTHIUS	000002	DMKCMD								
DMKTMR	000002	DMKPMA	DMKPRV							
DMKTMRCC	000001	DMKPRV								
DMKTMRCK	000001	DMKBLD								
DMKTMRFR	000001	DMKTOD								
DMKTMRPT	000024	DMKACO	DMKCKF	DMKCPI	DMKDSP	DMKFPS	DMKHVC	DMKMOO	DMKPRG	DMKRSP	DMKSVD
		DMKTHI	DMKVSX							
DMKTMRSN	000001	DMKPSA								
DMKTMRSP	000002	DMKPMA	DMKPRV							
DMKTMRTN	000003	DMKPRV	DMKSYM							
DMKTMRVT	000002	DMKEXT								
DMKTODIN	000002	DMKCPI								
DMKTODTB	000009	DMKALO	DMKCPI	DMKMNT						
DMKTPERP	000001	DMKIOS								
DMKTRACE	000002	DMKCFC								
DMKTRCEX	000004	DMKCPB	DMKDSP							
DMKTRCIO	000002	DMKDSP								
DMKTRCIT	000010	DMKCDS	DMKCFJ	DMKDSP	DMKSVD	DMKTRA				
DMKTRCND	000004	DMKCFG	DMKUSQ							
DMKTRCPB	000016	DMKCDS	DMKCFJ	DMKCFP	DMKPRV	DMKSVD	DMKTRA			
DMKTRCPG	000004	DMKDSP	DMKPRG							
DMKTRCPV	000002	DMKPRV								
DMKTRCSV	000002	DMKSVD								
DMKTRCSW	000002	DMKVIO								
DMKTRDSI	000012	DMKHPS	DMKIOS	DMKVIO	DMKVSI	DMKVSJ				
DMKTRDWT	000002	DMKVIO								
DMKTRKFP	000005	DMKSYM	DMKUNT	DMKVIO						
DMKTRKIN	000003	DMKDAS	DMKSYM							
DMKTRKVA	000003	DMKCCD	DMKSYM							
DMKTRMID	000002	DMKCNS								
DMKTRPST	000002	DMKCFC								
DMKTRQCP	000002	DMKBLD	DMKSYM							
DMKTRQIL	000001	DMKSTP								
DMKTRQMD	000002	DMKSYM	DMKTOD							
DMKTRQRT	000004	DMKSCH								
DMKTRQST	000006	DMKPGM	DMKSCH							
DMKTRQTI	000004	DMKSTP	DMKSYM							
DMKTRQ80	000003	DMKCFY	DMKLOH	DMKSYM						
DMKTRT	000001	DMKTRP								
DMKTRTCM	000001	DMKTRU								
DMKTRUAC	000002	DMKTRP								
DMKTRUST	000001	DMKTRT								
DMKTRXCP	000001	DMKTRU								
DMKTRXEX	000001	DMKTRU								
DMKTRXLK	000001	DMKTRU								
DMKTRXTT	000001	DMKTRU								
DMKTRXVT	000001	DMKTRU								
DMKTTXIN	000008	DMKCNS	DMKVCR	DMKVCS						
DMKTTXTR	000002	DMKTTY								
DMKTTYOP	000012	DMKCNS	DMKVCR	DMKVCS						
DMKTTZLF	000002	DMKTTX	DMKTTY							

LABEL	COUNT	REFERENCES									
DMKUCBLD	000001	DMKCSC									
DMKUCCLD	000001	DMKCSC									
DMKUCSLD	000001	DMKCSC									
DMKUDRBV	000002	DMKCPI									
DMKUDRDS	000002	DMKHVD									
DMKUDRFD	000004	DMKLNK									
DMKUDRFU	000060	DMKCAO	DMKCFT	DMKCFY	DMKCSP	DMKCSQ	DMKCSU	DMKCSV	DMKCSX	DMKDEG	DMKHVD
		DMKIUP	DMKLNK	DMKLOG	DMKLOH	DMKLOM	DMKMNJ	DMKOPE	DMKREI	DMKRPD	DMKRST
		DMKSPL	DMKTRP	DMKVCT	DMKVMD						
DMKUDRIA	000008	DMKIDR	DMKIUC								
DMKUDRMD	000022	DMKCFT	DMKCFY	DMKDEG	DMKHVD	DMKIUP	DMKLOG	DMKLOH	DMKOPE	DMKREI	DMKSPL
		DMKTRP									
DMKUDROV	000002	DMKCPI									
DMKUDRRD	000002	DMKLOJ									
DMKUDRRV	000042	DMKCFT	DMKCFY	DMKDEG	DMKHVD	DMKIUC	DMKIUP	DMKLNK	DMKLOG	DMKLOH	DMKLOJ
		DMKOPE	DMKREI	DMKSPL	DMKTRP						
DMKUDRXI	000004	DMKLOH	DMKREI								
DMKUDUMN	000002	DMKHVD									
DMKUNTFB	000001	DMKSYM									
DMKUNTFR	000021	DMKAPI	DMKCCW	DMKCFR	DMKCPI	DMKGIO	DMKHVC	DMKSYM	DMKVIO	DMKVSI	
DMKUNTF1	000001	DMKCPI									
DMKUNTIS	000003	DMKISM	DMKSYM								
DMKUNTRN	000009	DMKAPI	DMKCPI	DMKGIO	DMKSYM	DMKTRK	DMKVIO				
DMKUNTRS	000003	DMKCCS	DMKSYM								
DMKURSTA	000022	DMKCQP	DMKCSO	DMKRSP	DMKRST						
DMKVATAB	000019	DMKCDB	DMKCDM	DMKCDS	DMKDSP	DMKPRG	DMKPRV	DMKPRW	DMKSYM		
DMKVATAT	000003	DMKCFV	DMKDSP	DMKPRV							
DMKVATBC	000033	DMKCDS	DMKCFP	DMKCFV	DMKCPB	DMKDSP	DMKMPO	DMKQVM	DMKSPM	DMKSYM	DMKUSQ
		DMKVAU									
DMKVATEX	000007	DMKDSP	DMKPRV	DMKSYM	DMKTMR						
DMKVATMD	000022	DMKCDS	DMKCFG	DMKCFV	DMKCPB	DMKDSP	DMKSPM	DMKSYM	DMKVAU		
DMKVATOF	000001	DMKVRR									
DMKVATPF	000002	DMKPRG									
DMKVATPX	000003	DMKPRG	DMKSYM								
DMKVATSI	000018	DMKBLD	DMKMCH	DMKPGS	DMKPTS	DMKPTT	DMKRPA	DMKSEL			
DMKVATSX	000003	DMKPRG	DMKSYM								
DMKVATZP	000002	DMKAPI	DMKCPI								
DMKVATZS	000002	DMKAPI	DMKCPI								
DMKVAULA	000009	DMKPRG	DMKPRV	DMKPRW	DMKSVD	DMKSYM					
DMKVAURN	000019	DMKPEI	DMKPER	DMKPRV	DMKSVD	DMKSYM	DMKTMR	DMKTRC	DMKTRD	DMKVER	
DMKVAUZP	000002	DMKAPI	DMKCPI								
DMKVBMIN	000004	DMKHPS	DMKHVF								
DMKVBMVG	000002	DMKHVF									
DMKVBMVR	000002	DMKHVF									
DMKVCAST	000002	DMKVSI									
DMKVCBRD	000002	DMKCFQ									
DMKVCBRS	000006	DMKDEF	DMKDIB	DMKVDR							
DMKVCBSH	000002	DMKVSJ									
DMKVCBTS	000002	DMKVSI									
DMKVCHDC	000004	DMKVDA	DMKVDD								
DMKVCNEX	000003	DMKSYM	DMKVSI								
DMKVCNFT	000002	DMKFRT	DMKSYM								
DMKVCPIL	000002	DMKIUB	DMKSYM								
DMKVCQAT	000006	DMKVCP	DMKVCR	DMKVCS							
DMKVCQRE	000002	DMKVCS									
DMKVCQSR	000011	DMKVCR	DMKVCS	DMKVCT	DMKVCX						
DMKVCRMT	000001	DMKSYM									
DMKVCRNR	000010	DMKDSP	DMKVCP	DMKVCU	DMKVCX						

LABEL	COUNT	REFERENCES						
DMKVCRRD	000002	DMKQCO						
DMKVCSWT	000005	DMKQCO	DMKQCQ	DMKVCQ				
DMKVCTCH	000003	DMKCFW	DMKSYM					
DMKVCTDA	000003	DMKCPV						
DMKVCTEN	000003	DMKCPV						
DMKVCTER	000002	DMKCFT						
DMKVCTLO	000010	DMKDIB	DMKQCP	DMKUSQ				
DMKVCTTM	000001	DMKVCV						
DMKVCUIL	000002	DMKVCP						
DMKVCVCE	000082	DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCW	DMKVCX
DMKVCVEB	000046	DMKVCP	DMKVCR	DMKVCS	DMKVCT	DMKVCU	DMKVCX	
DMKVCVER	000011	DMKSYM	DMKVCR	DMKVCS	DMKVCU			
DMKVCVIN	000010	DMKVCQ	DMKVCR	DMKVCT				
DMKVCVIX	000024	DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCX		
DMKVCVKS	000008	DMKVCP	DMKVCW					
DMKVCVLD	000020	DMKVCP	DMKVCR	DMKVCS	DMKVCT	DMKVCU	DMKVCX	
DMKVCVLY	000014	DMKVCP	DMKVCU					
DMKVCVND	000016	DMKVCQ	DMKVCR	DMKVCW				
DMKVCVUT	000008	DMKVCQ	DMKVCS	DMKVCT				
DMKVCWCN	000001	DMKIUB						
DMKVCWQS	000001	DMKIUB						
DMKVCWRM	000001	DMKIUB						
DMKVCWSV	000001	DMKIUB						
DMKVCXD2	000012	DMKVCT	DMKVCW					
DMKVCXFU	000034	DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCV	DMKVCW
DMKVCXGF	000004	DMKVCP	DMKVCR					
DMKVCXGO	000006	DMKVCT	DMKVCW					
DMKVCXIO	000007	DMKSYM	DMKVCP	DMKVCQ	DMKVCR			
DMKVCXOR	000004	DMKVCR	DMKVCT					
DMKVCXOX	000006	DMKVCT	DMKVCW					
DMKVCXSA	000012	DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCU		
DMKVDAAA	000002	DMKCMD						
DMKVDAAC	000002	DMKCMD						
DMKVDAS1	000001	DMKSSS						
DMKVDAS2	000001	DMKSSS						
DMKVDBMD	000002	DMKVDA						
DMKVDCPS	000002	DMKVDA						
DMKVDCSC	000004	DMKVDA	DMKVDD					
DMKVDDDC	000002	DMKCMD						
DMKVDDDG	000002	DMKCMD						
DMKVDDDR	000002	DMKCMD						
DMKVDDDS	000002	DMKCMD						
DMKVDEDC	000002	DMKVDA						
DMKVDERR	000002	DMKVDA						
DMKVDFER	000002	DMKVDD						
DMKVDGAL	000004	DMKIDU	DMKVDA					
DMKVDHBB	000008	DMKVDG						
DMKVDHFR	000002	DMKVDG						
DMKVDHOR	000002	DMKIDU						
DMKVDHPG	000004	DMKVDG						
DMKVDREL	000012	DMKCFP	DMKNEA	DMKNLD	DMKUSQ	DMKVCH	DMKVDD	
DMKVDRES	000006	DMKNLD	DMKVIO					
DMKVDSAT	000011	DMKLOJ	DMKLOM	DMKVCH	DMKVDA	DMKVRR		
DMKVDSDF	000008	DMKDEF	DMKLOJ	DMKNEA				
DMKVDSLK	000004	DMKLNK						
DMKVDTPG	000011	DMKLOJ	DMKVCH	DMKVDA	DMKVRR			
DMKVERD	000001	DMKSVD						
DMKVERO	000001	DMKSVD						

LABEL	COUNT	REFERENCES								
DMKVFCQV	000002	DMKCMD								
DMKVFCVV	000002	DMKCPT								
DMKVFDDP	000002	DMKCDB								
DMKVFEST	000002	DMKCDS								
DMKVFRCH	000004	DMKSYM	DMKVFC	DMKVFS						
DMKVFRIA	000001	DMKVFC								
DMKVFRIM	000001	DMKMCH								
DMKVFRIN	000002	DMKPRG								
DMKVFRIP	000004	DMKAPI	DMKMOO	DMKVFC						
DMKVFRNA	000001	DMKVFC								
DMKVFRNC	000002	DMKMOO								
DMKVFRNM	000001	DMKMCH								
DMKVFRNO	000002	DMKMOO								
DMKVFRNP	000002	DMKAPI	DMKVFC							
DMKVFRNR	000001	DMKMOO								
DMKVFRNS	000001	DMKMOO								
DMKVFRSA	000002	DMKVFS								
DMKVFRSS	000003	DMKVFD	DMKVFE	DMKVFS						
DMKVFRSV	000018	DMKACO	DMKCKF	DMKCPO	DMKDMP	DMKEXT	DMKMOO	DMKPRV	DMKTHI	DMKVFC	DMKVFD
		DMKVFE	DMKVFS							
DMKVFRVL	000001	DMKVFS								
DMKVFSOS	000006	DMKVFD	DMKVFE	DMKVFR						
DMKVFSRS	000002	DMKPGS								
DMKVIOCL	000004	DMKDSP	DMKVSJ							
DMKVIOC1	000002	DMKVSI								
DMKVIOIN	000026	DMKACO	DMKCFR	DMKCSR	DMKCSW	DMKCSX	DMKDIF	DMKHPT	DMKHPU	DMKIOQ	DMKIOS
		DMKIOT	DMKRGA	DMKRGB	DMKRGC	DMKRGE	DMKSPL	DMKSPM	DMKSYM	DMKUNT	DMKVCA
		DMKVCB	DMKVCN	DMKVCP	DMKVSI	DMKVSJ				
DMKVIOMK	000015	DMKCFM	DMKCFQ	DMKCPB	DMKDSP	DMKSSS	DMKVCN	DMKVDC	DMKVSI	DMKVSJ	DMKVSP
DMKVIOXK	000002	DMKVSJ								
DMKVMASH	000039	DMKAPI	DMKATS	DMKCCW	DMKCDM	DMKCDS	DMKCPI	DMKDGD	DMKDGF	DMKDSP	DMKPGS
		DMKSYM	DMKUSQ	DMKVCN	DMKVSR	DMKVSX				
DMKVMASW	000010	DMKCPP	DMKLOK	DMKWAI						
DMKVMAS1	000007	DMKCFF	DMKCFH	DMKCPP	DMKCPU					
DMKVMAS2	000001	DMKCFF								
DMKVMCEX	000002	DMKDSP								
DMKVMCFC	000004	DMKHVC	DMKMSG							
DMKVMCUA	000002	DMKCFP								
DMKVMDEP	000002	DMKCFC								
DMKVMDIA	000002	DMKHVE								
DMKVMEDP	000002	DMKVMD								
DMKVMGCN	000001	DMKIUB								
DMKVMGIL	000001	DMKIUB								
DMKVMGSV	000001	DMKIUB								
DMKVMI	000004	DMKCFG	DMKRPA	DMKSEG						
DMKVRR	000011	DMKBLD	DMKCKF	DMKCPI	DMKDMP	DMKDMQ	DMKPMA	DMKRSP	DMKSYM		
DMKVRRDD	000004	DMKCPI	DMKMNT	DMKSYM						
DMKVRRIS	000006	DMKDMP	DMKDMQ	DMKOPR	DMKSYM					
DMKVRROP	000006	DMKDMP	DMKDMQ	DMKLOJ	DMKVRS					
DMKVRRRC	000004	DMKCPI	DMKSYM							
DMKVRRRS	000005	DMKCPJ	DMKSYM							
DMKVRSSV	000009	DMKDMP	DMKDMQ	DMKSTA	DMKSYM					
DMKVSC	000001	DMKSYM								
DMKVSCVR	000002	DMKVSI								
DMKVSDAD	000030	DMKACO	DMKCKR	DMKCSV	DMKCSY	DMKDRD	DMKMIA	DMKNLE	DMKSPL	DMKSPS	DMKTRT
		DMKVME	DMKVSW	DMKWRN						
DMKVSDDL	000018	DMKCSV	DMKCSY	DMKDRD	DMKSPS	DMKVSW	DMKWRN			
DMKVSDFH	000012	DMKLOH	DMKVSF	DMKVSG						

LABEL	COUNT	REFERENCES									
DMKVSEER	000014	DMKVSG									
DMKVSELK	000008	DMKVSG									
DMKVSETR	000044	DMKCKR	DMKCSY	DMKSPS	DMKVSD	DMKVSF	DMKVSG				
DMKVSFID	000014	DMKCQH	DMKCST	DMKDRD	DMKDRE	DMKSFB	DMKXAB				
DMKVSFNS	000016	DMKCQH	DMKCQI	DMKCSV	DMKCSW	DMKCSY	DMKSFB	DMKSPT			
DMKVSFNU	000040	DMKCQH	DMKCQI	DMKCSR	DMKCSV	DMKCSW	DMKCSY	DMKDRD	DMKDRE	DMKSFB	DMKSPT
		DMKVSW									
DMKVSGAI	000012	DMKCKT	DMKCSY	DMKRST	DMKSPL	DMKWRN					
DMKVSGRI	000028	DMKCFU	DMKCSY	DMKDRD	DMKDRE	DMKMIA	DMKNLE	DMKRST	DMKSPK	DMKSPL	DMKSPS
		DMKTRU	DMKVME	DMKWRN							
DMKVSGUM	000008	DMKCKR	DMKVSE								
DMKVSICH	000001	DMKVSJ									
DMKVSICI	000002	DMKMOO	DMKSYM								
DMKVSICT	000001	DMKMOO									
DMKVSICW	000002	DMKMOO	DMKSYM								
DMKVSIEX	000010	DMKHVC	DMKHVE	DMKPRV	DMKSYM	DMKVSJ					
DMKVSIFT	000001	DMKVSJ									
DMKVSISF	000002	DMKMOO	DMKSYM								
DMKVSISI	000002	DMKMOO	DMKSYM								
DMKVSITC	000002	DMKMOO	DMKSYM								
DMKVSITI	000002	DMKMOO	DMKSYM								
DMKVSIVS	000001	DMKPRV									
DMKVSJEX	000002	DMKVSI									
DMKVSJHD	000002	DMKMOO	DMKSYM								
DMKVSJHI	000002	DMKMOO	DMKSYM								
DMKVSP	000001	DMKVSV									
DMKVSPEX	000003	DMKSYM	DMKVSI								
DMKVSPPE	000002	DMKVSU									
DMKVSPST	000002	DMKVSU									
DMKVSPUS	000002	DMKVSU									
DMKVSPWA	000007	DMKSYM	DMKUSP	DMKVST							
DMKVSQPD	000011	DMKSYM	DMKVSP	DMKVST							
DMKVSRGC	000013	DMKSYM	DMKVSP	DMKVSX							
DMKVSRMD	000009	DMKSYM	DMKVSP	DMKVSX							
DMKVSTOP	000003	DMKSYM	DMKVSP								
DMKVSTPT	000018	DMKCDM	DMKPET	DMKRNH	DMKTRC	DMKTRD					
DMKVSTVP	000008	DMKQCO	DMKTTX	DMKTTY	DMKVCU						
DMKVSUCO	000011	DMKCFQ	DMKCPS	DMKCSQ	DMKCSR	DMKVDR					
DMKVSUCR	000008	DMKCFQ	DMKCSQ	DMKDRD	DMKVDR						
DMKVSVLD	000002	DMKVSP									
DMKVSWDC	000013	DMKHVF	DMKSYM	DMKVSP	DMKVSU	DMKVSX					
DMKVSWFC	000011	DMKSYM	DMKVSP	DMKVSU	DMKVSX						
DMKVSWOC	000002	DMKVSX									
DMKVSWOR	000004	DMKHVF	DMKVSP								
DMKVSWOT	000002	DMKVSX									
DMKVSWTO	000003	DMKSYM	DMKVSI								
DMKVSXCL	000002	DMKVSP									
DMKVSXRD	000002	DMKVSW									
DMKVSXSE	000002	DMKVSP									
DMKVSXSI	000002	DMKVSP									
DMKVSXSR	000002	DMKVSP									
DMKVSXTR	000002	DMKVSP									
DMKWAIST	000002	DMKDSP	DMKSYM								
DMKWAITA	000004	DMKAPI	DMKDSP	DMKSYM							
DMKWRM	000001	DMKLD00E									
DMKWRMST	000002	DMKCPJ									
DMKWRMSY	000002	DMKCPJ									
DMKWRNSB	000006	DMKCKV	DMKWRM								

LABEL	COUNT	REFERENCES				
DMKWRNWM	000004	DMKWRM				
DMKXABDG	000004	DMKAPT	DMKHVE			
DMKXADVS	000002	DMKVSP				
DMKXSTOR	000002	DMKPST				
DMKZTDDF	000002	DMKVDS				
DMKZTDST	000002	DMKCPX				
DMPABEND	000002	DMKDMP	DMKSAD			
DMPAP	000003	DMKDMQ				
DMPCKCOM	000002	DMKSAD	DMKVME			
DMPCPUID	000004	DMKDMP	DMKSAD	DMKVME		
DMPCPUTM	000001	DMKSAD				
DMPCRS	000003	DMKSAD	DMKVME			
DMPDMPID	000003	DMKVME				
DMPFLAG	000001	DMKDMP				
DMPFPRS	000003	DMKDMP	DMKSAD	DMKVME		
DMPGPRS	000005	DMKDMP	DMKSAD	DMKVME		
DMPINREC	000007	DMKDMP	DMKSAD	DMKVME		
DMPIPCS	000021	DMKDMP	DMKDMQ	DMKSAD	DMKVMD	DMKVME
DMPKEY	000002	DMKDMP				
DMPKYREC	000001	DMKDMP				
DMPLCORE	000003	DMKDMP	DMKSAD			
DMPMP	000009	DMKDMQ				
DMPPGMAP	000006	DMKDMP	DMKSAD	DMKVME		
DMPPGMP2	000004	DMKDMP	DMKSAD			
DMPPRFRG	000002	DMKDMP	DMKSAD			
DMPPROCA	000001	DMKDMP				
DMPPSW	000001	DMKVME				
DMPROC	000003	DMKDMQ				
DMPSMSG	000004	DMKDMQ				
DMPSMSGL	000002	DMKDMQ				
DMPSTAT	000021	DMKDMQ				
DMPSYSRM	000003	DMKDMP	DMKSAD	DMKVME		
DMPSYSRV	000003	DMKDMP	DMKSAD	DMKVME		
DMPTODCK	000004	DMKDMP	DMKSAD	DMKVME		
DMPUNI	000010	DMKDMQ				
DMPVMTYP	000002	DMKSAD	DMKVME			
DOLOCAL	000002	DMKACO	DMKCKF			
DORDEV	000006	DMKACO	DMKCKF			
DOTERMID	000002	DMKACO	DMKCKF			
DOUBLESP	000005	DMKDMQ				
DOVMTERM	000002	DMKACO	DMKCKF			
DPAPUOP	000006	DMKAPI	DMKCPP	DMKEXT	DMKPSA	
DPLEND	000004	DMKUDR	DMKUDU			
DPLENDFG	000004	DMKUDR	DMKUDU			
DPLHIGH	000005	DMKUDR				
DPLIST	000003	DMKUDR	DMKUDU			
DPLLOW	000005	DMKUDR				
DPLNGTH	000002	DMKUDR				
DPLSIZE	000005	DMKUDR	DMKUDU			
DPLSIZ2	000001	DMKUDR				
DPLVADD	000005	DMKUDR	DMKUDU			
DPMICON	000007	DMKAPI	DMKCFO	DMKCPI	DMKPSA	
DPOKTLB	000005	DMKAPI	DMKCPI	DMKDSP	DMKPSA	
DPSEGPRT	000003	DMKAPI	DMKCPI	DMKPSA		
DPSTAT	000019	DMKAPI	DMKCFO	DMKCPI	DMKCPP	DMKDSP	DMKEXT
DRHA	000001	DMKCCD				
DSPA	000004	DMKDSP	DMKPXA	DMKPXB		
DSPB	000006	DMKDSP	DMKPXA	DMKPXB		

LABEL	COUNT	REFERENCES									
DSPCH	000004	DMKDSP	DMKPXA	DMKPXB							
DSPRQ	000011	DMKAPI	DMKCPP	DMKDSP	DMKFRE	DMKPXA	DMKPXB	DMKWAI			
DSPRU	000004	DMKDSP	DMKPXA	DMKPXB							
DSPWA	000004	DMKPXA	DMKPXB	DMKWAI							
DUMP	000019	DMKCFC	DMKCKT	DMKCMD	DMKDDR	DMKDMQ	DMKRND	DMKSPT	DMKVMD	DMKWRM	
DUMPSAVE	000036	DMKCCH	DMKCKH	DMKCKN	DMKDMP	DMKMCT	DMKMPO	DMKPRG	DMKPSA	DMKSVC	
DWHA	000002	DMKCCD	DMKCCO								
D1	000005	DMKCKH	DMKDDR	DMKFMT							
D2	000010	DMKDDR	DMKFMT	DMKIOE	DMKSAV						
D28	000001	DMKFMT									
D3	000003	DMKDDR	DMKFMT	DMKPAG							
D4	000022	DMKCKH	DMKCKN	DMKDMP	DMKFMT	DMKFRT	DMKMNT	DMKPAG	DMKSAV	DMKSSP	DMKUNT
D5	000002	DMKDDR	DMKFMT								
D66LNCNT	000006	DMKGRD	DMKQCQ								
D66LNLEN	000006	DMKQCQ									
D8	000005	DMKCKN	DMKFMT	DMKLOH	DMKZTD						
ECBLOK	000155	DMKHVC									
ECKD	000004	DMKCKH	DMKCKP								
ECPSOP	000005	DMKFRE									
ECPSUBTB	000006	DMKFRE									
ECSWBYT3	000001	DMKCCH									
ECSWLOG	000010	DMKCCH	DMKIOG								
EDIT	000041	DMKCFH	DMKCFM	DMKCFU	DMKCNS	DMKCNT	DMKCPJ	DMKEPS	DMKGRG	DMKGRT	DMKMSW
		DMKNLD	DMKNLE	DMKOPE	DMKRGC	DMKRNH	DMKTOD	DMKTRR	DMKVCN	DMKVCR	DMKVCU
EGPR0	000002	DMKIMG									
EGPR15	000007	DMKIMG									
EGPR8	000001	DMKIMG									
EIGHT	000006	DMKLD00E	DMKSAD								
EMSINQSC	000003	DMKDSP									
EMSMASK	000009	DMKAPI	DMKDSP	DMKIOT	DMKLOK	DMKPMA					
EMSPCLKC	000003	DMKCLK	DMKEXT								
EMSPEND	000024	DMKCLK	DMKDSP	DMKEXT	DMKMCT						
EMSPEXT	000005	DMKDSP	DMKEXT								
EMSPQUI	000007	DMKDSP	DMKEXT	DMKMCT							
EMSPSHD	000003	DMKEXT									
EMSPSYNC	000004	DMKCLK	DMKEXT								
EMSPXEX	000003	DMKEXT									
EMSRCLKC	000001	DMKCLK									
EMSREC	000008	DMKCLK	DMKDSP	DMKEXT							
EMSREXT	000001	DMKDSP									
EMSRQUI	000001	DMKDSP									
EMSRSHD	000001	DMKEXT									
EMSRSYNC	000001	DMKCLK									
EMSRXEX	000001	DMKEXT									
ENDSIZES	000002	DMKFRE									
ENDWORK	000002	DMKCCW									
ENQ	000011	DMKBSC									
EQCHK	000020	DMKCQT	DMKDAD	DMKDDR	DMKFMT	DMKIOE	DMKRSE				
ERASWRT	000003	DMKVCN									
ERRBLOK	000017	DMKERP	DMKIOE	DMKIOF	DMKIOJ						
ERRCCNT	000002	DMKIOE	DMKIOJ								
ERRCCW	000005	DMKIOE	DMKIOJ								
ERRCODE	000003	DMKCQG	DMKVMD								
ERRCONT	000003	DMKIOJ									
ERRCORR	000003	DMKIOE	DMKIOJ								
ERRCPID	000006	DMKIOE	DMKIOJ								
ERRFLAG	000011	DMKIOE	DMKIOF								
ERRHEADR	000002	DMKIOE									

LABEL	COUNT	REFERENCES									
ERRIOB	000013	DMKIOE	DMKIOF	DMKIOJ							
ERRIOER	000003	DMKIOE	DMKIOF	DMKIOJ							
ERRKEEP	000005	DMKIOE	DMKIOF								
ERRKEY	000005	DMKIOE	DMKIOJ								
ERRMIOB	000006	DMKIOE	DMKIOJ								
ERRMIOER	000002	DMKIOE	DMKIOJ								
ERRMSG	000027	DMKACR	DMKCFH	DMKCFM	DMKCQI	DMKCQP	DMKDDR	DMKDIR	DMKERM	DMKNLD	DMKNLE
		DMKOVR	DMKPRG	DMKRNH	DMKTRR	DMKURS					
ERRMSIZE	000001	DMKIOE									
ERROBR	000008	DMKERP	DMKIOE	DMKIOF							
ERROR	000118	DMKMCD	DMKMNI	DMKMON							
ERRPARM	000004	DMKIOE	DMKIOF								
ERRPATH	000005	DMKIOF									
ERRSDR	000009	DMKIOE	DMKIOF	DMKIOJ							
ERRSIZE	000001	DMKIOE									
ERRSRDEV	000003	DMKERP	DMKIOE								
ERRSW2	000001	DMKIOE									
ERRVOLID	000006	DMKIOE	DMKIOJ								
ETB	000006	DMKRGD									
ETX	000020	DMKRGD									
EWA	000003	DMKGRG									
EXCLOCK	000004	DMKFRE	DMKFRT								
EXCOLOR	000009	DMKGRE									
EXDCCF	000001	DMKMCH									
EXDCNO	000001	DMKMCH									
EXDINSTO	000001	DMKMCH									
EXDINTTO	000001	DMKMCH									
EXDRESVD	000001	DMKMCH									
EXHAUST	000004	DMKMIA									
EXHILIGH	000009	DMKGRE									
EXNPSW	000016	DMKAPI	DMKCPI	DMKDSP	DMKEXT	DMKMCT	DMKMPO	DMKPMA	DMKSAD	DMKSAV	
EXOPSW	000029	DMKAPI	DMKDSP	DMKEXT	DMKMPO	DMKPMA	DMKSAD				
EXPSS	000003	DMKGRE									
EXSUBTOP	000006	DMKFRE	DMKFRT								
EXTMASK	000017	DMKACR	DMKAPI	DMKCFH	DMKCPI	DMKCPU	DMKEXT	DMKSVD	DMKTMR	DMKTRC	
EXTMODE	000097	DMKAPI	DMKCDS	DMKCFF	DMKCFG	DMKCFV	DMKCLK	DMKCPB	DMKCPI	DMKCPJ	DMKDMP
		DMKDMQ	DMKDSP	DMKFPS	DMKMHC	DMKPEI	DMKPEL	DMKPEN	DMKPER	DMKPET	DMKPMA
		DMKPRG	DMKPRV	DMKPRW	DMKPSA	DMKSAD	DMKSTA	DMKSVD	DMKTOD	DMKTRC	DMKTRD
		DMKVAT	DMKVER	DMKVMI							
EXTRACTL	000001	DMKPRV									
EXTVPORL	000012	DMKHVC									
FAILADD	000002	DMKCCH									
FAILCCW	000010	DMKCCH									
FAILCSW	000008	DMKCCH									
FAILECSW	000003	DMKCCH									
FAILSTAD	000002	DMKPMA									
FAKEATT	000003	DMKGRI									
FASTCPU	000002	DMKAPI	DMKCPI								
FF	000089	DMKAPI	DMKCFO	DMKCFV	DMKCLK	DMKCPI	DMKCPO	DMKCPP	DMKCPU	DMKMNT	DMKOPE
		DMKPRG	DMKPTR	DMKPTS	DMKPTT	DMKSEL	DMKSTA	DMKTRM	DMKVRR	DMKWRM	
FFS	000413	DMKACR	DMKACS	DMKAPI	DMKAPS	DMKAPT	DMKAPU	DMKATS	DMKBLD	DMKCCF	DMKCCH
		DMKCCS	DMKCCW	DMKCDB	DMKCDM	DMKCFC	DMKCFF	DMKCFG	DMKCFM	DMKCFP	DMKCFQ
		DMKCFR	DMKCKD	DMKCKF	DMKCKH	DMKCKP	DMKCKR	DMKCKS	DMKCKT	DMKCKV	DMKCPB
		DMKCPI	DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPT	DMKCPV	DMKCPY	DMKCPZ	DMKCQP
		DMKCRM	DMKCSQ	DMKCSR	DMKCST	DMKCSU	DMKCSV	DMKCSW	DMKCSX	DMKDAD	DMKDAS
		DMKDAU	DMKDEF	DMKDGD	DMKDGF	DMKDIA	DMKDIB	DMKDMQ	DMKDRD	DMKDSP	DMKEIG
		DMKEPS	DMKERP	DMKFRE	DMKGRC	DMKGRD	DMKGRF	DMKGRT	DMKHPS	DMKHVC	DMKHVE
		DMKIDR	DMKIOF	DMKIOG	DMKIOQ	DMKIOS	DMKIOT	DMKIUS	DMKJRL	DMKLNM	DMKLOH

LABEL	COUNT	REFERENCES									
		DMKLOJ	DMKMCC	DMKMCH	DMKMCT	DMKMIA	DMKMID	DMKMNI	DMKMNJ	DMKMNT	DMKMON
		DMKMPO	DMKMSG	DMKMSW	DMKNEA	DMKNES	DMKNLD	DMKOPE	DMKOPR	DMKPEQ	DMKPGS
		DMKPMA	DMKPRG	DMKPRV	DMKPST	DMKPTR	DMKQCN	DMKQCO	DMKRPA	DMKRSF	DMKRSP
		DMKSCH	DMKSCN	DMKSEL	DMKSEP	DMKSEV	DMKSIX	DMKSPK	DMKSPL	DMKSRM	DMKSSS
		DMKSST	DMKSTA	DMKSVD	DMKTAP	DMKTAQ	DMKTOD	DMKTRA	DMKTRC	DMKTRD	DMKTRK
		DMKTRP	DMKTRT	DMKTRX	DMKUDU	DMKUNT	DMKURS	DMKUSQ	DMKVBM	DMKVCA	DMKVCB
		DMKVCH	DMKVCN	DMKVCV	DMKVDB	DMKVDC	DMKVDD	DMKVDE	DMKVDF	DMKVDH	DMKVDR
		DMKVDS	DMKVDT	DMKVFC	DMKVIO	DMKVMA	DMKVMD	DMKVRR	DMKVRS	DMKVSD	DMKVSG
		DMKVSI	DMKVSP	DMKVSQ	DMKVST	DMKVSV	DMKWRM	DMKXAB	DMKXAD	DMKZTD	
FIRSTRCW	000014	DMKCCD	DMKCCO	DMKCCS	DMKCCW						
FLAG	000068	DMKCPZ	DMKCSW	DMKDAD	DMKDAS	DMKDEG	DMKDIR	DMKDMP	DMKERP	DMKFMT	
FLAGS	000040	DMKCQP	DMKCQQ	DMKCQT	DMKPEI	DMKPST	DMKSPT				
FLAG2	000005	DMKTRA	DMKVMI								
FNAME	000001	DMKACO									
FORMBLOK	000011	DMKAPU	DMKAPV	DMKCKF	DMKCKR	DMKCSR	DMKCSW	DMKMNT	DMKSEP	DMKSPL	DMKVDS
		DMKVSQ									
FORMFLAG	000002	DMKVSQ	DMKVSQ								
FORMNARR	000002	DMKSEP	DMKVSQ								
FORMNTRY	000007	DMKAPU	DMKAPV	DMKCKF	DMKCSW	DMKSEP	DMKSPL	DMKVSQ			
FORMOPER	000008	DMKAPU	DMKAPV	DMKCKF	DMKCKR	DMKCSW	DMKMNT	DMKSEP	DMKSPL		
FORMSEND	000007	DMKAPU	DMKAPV	DMKCKF	DMKCSW	DMKSEP	DMKSPL	DMKVSQ			
FORMUSER	000015	DMKAPU	DMKAPV	DMKCKF	DMKCKR	DMKCSR	DMKCSW	DMKSEP	DMKSPL	DMKVDS	DMKVSQ
FPREGS	000006	DMKDMQ									
FPRLOG	000007	DMKDMP	DMKPMA	DMKSAD							
FRADDR	000004	DMKFRE	DMKFRT								
FRBYTES	000002	DMKFRT									
FRECALLR	000001	DMKFRE									
FREDPACA	000004	DMKFRE									
FREE	000024	DMKFRE	DMKMOO	DMKPXA	DMKPXB						
FREEH	000007	DMKFRE									
FREELOWE	000001	DMKNMT									
FREENUM	000001	DMKMOO									
FREEP	000008	DMKFRE									
FREER0	000009	DMKFRE	DMKFRT								
FREER1	000008	DMKFRE									
FREER12	000001	DMKFRE									
FREER13	000001	DMKFRT									
FREER14	000003	DMKFRE	DMKFRT								
FREER15	000011	DMKFRE	DMKFRT								
FREESAVE	000031	DMKCKD	DMKCKF	DMKCKM	DMKCKP	DMKERP	DMKFRE	DMKFRT	DMKPTS	DMKVCA	
FREEWORK	000015	DMKFRE	DMKFRT	DMKPTR							
FREEXT	000014	DMKFRE	DMKFRT	DMKSVC							
FRELSNSS	000002	DMKFRE									
FRELSTTS	000005	DMKFRE									
FRENAME	000001	DMKFRE									
FRLKPROC	000006	DMKDSP	DMKFRE								
FROSIZE	000007	DMKFRE	DMKFRT								
FRPALLOC	000005	DMKFRE	DMKFRT	DMKSVC							
FRPCACHL	000001	DMKFRT									
FRPMCALR	000003	DMKFRE	DMKSVC								
FRPMEXT	000010	DMKFRE	DMKFRT	DMKSVC							
FRPMNAME	000007	DMKFRE	DMKSVC								
FRPMSIZE	000004	DMKFRE	DMKFRT	DMKSVC							
FRPMTAG	000008	DMKFRE	DMKFRT	DMKSVC							
FRPRETRN	000003	DMKFRT	DMKSVC								
FRPSIZE	000003	DMKFRE	DMKFRT	DMKSVC							
FRSIZE	000005	DMKFRE	DMKFRT								
FRTAG	000006	DMKFRE	DMKFRT	DMKSVC							

LABEL	COUNT	REFERENCES									
FSCBANIT	000001	DMKDIR									
FSCBBUFF	000005	DMKDIR									
FSCBD	000006	DMKDIR									
FSCBFN	000011	DMKDIR									
FSCBNOIT	000003	DMKDIR									
FSCBSIZE	000002	DMKDIR									
FSIZE	000010	DMKLDOOE									
FSREAD	000004	DMKVCN									
FSTD	000001	DMKDIR									
FSTFMODE	000001	DMKNMT									
FSTFNAME	000001	DMKNMT									
FSTLRECL	000003	DMKDIR									
FSTRECCT	000001	DMKDIR									
FSTRECFM	000001	DMKDIR									
FSWRITE	000004	DMKVCN									
FTRAWSF	000004	DMKVCN	DMKVCP								
FTRDIAL	000024	DMKBSC	DMKRGA	DMKRGB	DMKRGC	DMKVCN					
FTREXTSN	000007	DMKCCW	DMKDAS	DMKDSB	DMKIOE	DMKIOJ	DMKRSF	DMKSSP			
FTRFH	000007	DMKCPM	DMKMNT	DMKPAG	DMKVDE						
FTRNODRP	000002	DMKGRD	DMKGRF								
FTROPRDR	000001	DMKACO									
FTRRPS	000009	DMKBIO	DMKDAS	DMKDSB	DMKIOS	DMKMNT	DMKVDE				
FTRRSRL	000010	DMKCCS	DMKDAD	DMKDAS	DMKDAU	DMKDSB	DMKMNT	DMKTAP	DMKVDS		
FTRTYP1	000003	DMKNLD	DMKNLE								
FTRUCS	000002	DMKCSB	DMKSSP								
FTRVIRT	000004	DMKALO	DMKVDC	DMKVDE							
FTR2311B	000005	DMKDIR	DMKLNK	DMKSCO	DMKVER						
FTR2311T	000005	DMKDIR	DMKLNK	DMKSCO	DMKVER						
FTR3088	000006	DMKDEF	DMKDIR	DMKSCN	DMKVDS						
FTR3270	000003	DMKDIR	DMKVDS								
FTR3270E	000004	DMKDID	DMKGRD	DMKHVE							
FTR35MB	000019	DMKALO	DMKBIO	DMKCCD	DMKDAS	DMKDSB	DMKHVE	DMKLNK	DMKMNT	DMKPMA	DMKUNT
		DMKVDE	DMKVER	DMKVSC							
FTR4WCGM	000017	DMKDEF	DMKDEG	DMKDIR	DMKTCT	DMKVDS					
FTR70MB	000017	DMKDAS	DMKDSB	DMKLNK	DMKMNT	DMKPAG	DMKPMA	DMKSPK	DMKVDE	DMKVDG	DMKVSC
		DMKZTD									
FWDTIC	000003	DMKCCW									
FXDLOG	000022	DMKCCH	DMKCKD	DMKDMP	DMKIOG	DMKPMA	DMKQVM				
F0	000475	DMKACO	DMKACR	DMKAPT	DMKATS	DMKBIO	DMKBLD	DMKCCH	DMKCCW	DMKCDB	DMKCDM
		DMKCDS	DMKCFC	DMKCFD	DMKCFF	DMKCFG	DMKCFH	DMKCFM	DMKCFQ	DMKCFS	DMKCFU
		DMKCFV	DMKCFY	DMKCKD	DMKCKF	DMKCKP	DMKCKV	DMKCNS	DMKCPB	DMKCPI	DMKCPM
		DMKCPO	DMKCPP	DMKCPS	DMKCPU	DMKCPW	DMKCPY	DMKCQC	DMKCQS	DMKCQU	DMKCQY
		DMKCSC	DMKCSO	DMKCSW	DMKCSY	DMKDAU	DMKDRD	DMKDRE	DMKERM	DMKEXT	DMKFMT
		DMKFRE	DMKGIO	DMKGRA	DMKGRC	DMKGRF	DMKGRG	DMKGRI	DMKGRT	DMKHPS	DMKHPT
		DMKHPU	DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKIDU	DMKIOG	DMKIOQ	DMKIOT	DMKIUA
		DMKIUB	DMKIUC	DMKIUE	DMKIUG	DMKIUL	DMKIUN	DMKIUS	DMKJRL	DMKLNK	DMKLOH
		DMKLOJ	DMKMCC	DMKMCD	DMKMHV	DMKMIA	DMKMNJ	DMKMNL	DMKMON	DMKMSG	DMKNLD
		DMKNLE	DMKOPE	DMKOPR	DMKPEI	DMKPER	DMKPGM	DMKPGS	DMKPMA	DMKPRG	DMKPRV
		DMKPRW	DMKPTR	DMKPTS	DMKQCN	DMKQCO	DMKQCQ	DMKRGC	DMKRPD	DMKRSP	DMKRST
		DMKSCN	DMKSEG	DMKSEL	DMKSEP	DMKSFB	DMKSNC	DMKSPK	DMKSPL	DMKSPS	DMKSPT
		DMKSRM	DMKSSS	DMKSST	DMKSSV	DMKSTP	DMKSTR	DMKSVC	DMKSVD	DMKTAP	DMKTCS
		DMKTCT	DMKTMR	DMKTOD	DMKTRC	DMKTRD	DMKTRK	DMKTRP	DMKTRT	DMKTRU	DMKTRX
		DMKTTX	DMKUDR	DMKUDU	DMKUNT	DMKUSQ	DMKVAT	DMKVAU	DMKVBM	DMKVCH	DMKVCN
		DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCU	DMKVCV	DMKVCW	DMKVCX	DMKVDC
		DMKVDD	DMKVDR	DMKVDS	DMKVER	DMKVIO	DMKVMC	DMKVMD	DMKVME	DMKVSE	DMKVSG
		DMKVSI	DMKVSJ	DMKVSP	DMKVSQ	DMKVSR	DMKVSX	DMKWRM	DMKWRN	DMKXAB	DMKXAD
		DMKZTD									
F1	000626	DMKACO	DMKACR	DMKACS	DMKALG	DMKALO	DMKAPI	DMKATS	DMKBLD	DMKBSC	DMKCCF

LABEL	COUNT	REFERENCES									
		DMKCCH	DMKCCW	DMKCDB	DMKCDM	DMKCDS	DMKCFD	DMKCFF	DMKCFG	DMKCFH	DMKCFJ
		DMKCFP	DMKCFQ	DMKCFR	DMKCFT	DMKCFU	DMKCFV	DMKCFY	DMKCKF	DMKCKM	DMKCKR
		DMKCKV	DMKCNS	DMKCPB	DMKCPI	DMKCPJ	DMKCPN	DMKCPO	DMKCPP	DMKCPT	DMKCPU
		DMKCPY	DMKCQG	DMKCQH	DMKCQI	DMKCQP	DMKCQQ	DMKCQR	DMKCQS	DMKCQT	DMKCQY
		DMKCSB	DMKCSC	DMKCSF	DMKCSP	DMKCSQ	DMKCSR	DMKCST	DMKCSU	DMKCSV	DMKCSW
		DMKCSX	DMKCSY	DMKCVT	DMKCVU	DMKDAD	DMKDAS	DMKDAU	DMKDGD	DMKDIB	DMKDID
		DMKDIF	DMKDMP	DMKDRD	DMKDRE	DMKDSP	DMKENT	DMKERM	DMKEXT	DMKFPS	DMKFRT
		DMKGRC	DMKGRE	DMKGRF	DMKGRG	DMKGRT	DMKHPS	DMKHPU	DMKHVC	DMKHVD	DMKHVE
		DMKHVF	DMKIDU	DMKIOE	DMKIOG	DMKIOH	DMKIOQ	DMKIOT	DMKIUA	DMKIUE	DMKIUJ
		DMKIUS	DMKLNK	DMKLNM	DMKLOG	DMKLOJ	DMKLOK	DMKLOM	DMKMCC	DMKMCD	DMKMCH
		DMKMHC	DMKMIA	DMKMNI	DMKMNJ	DMKMNL	DMKMOO	DMKMPO	DMKMSG	DMKNEA	DMKNES
		DMKNLD	DMKNLE	DMKOPE	DMKOPR	DMKPAG	DMKPEI	DMKPEN	DMKPEQ	DMKPER	DMKPET
		DMKPGM	DMKPGS	DMKPGT	DMKPMA	DMKPRG	DMKPRV	DMKPST	DMKPTR	DMKPTS	DMKPTT
		DMKQCO	DMKQCQ	DMKQVM	DMKRGA	DMKRGB	DMKRGC	DMKRGD	DMKRGE	DMKRNH	DMKRPA
		DMKRSE	DMKRSP	DMKSAV	DMKSBL	DMKSCH	DMKSEG	DMKSEL	DMKSIX	DMKSNC	DMKSPL
		DMKSPM	DMKSPS	DMKSPT	DMKSRM	DMKSST	DMKSSU	DMKSTA	DMKSTP	DMKSVD	DMKSWA
		DMKSWM	DMKTAP	DMKTAQ	DMKTCS	DMKTCT	DMKTHI	DMKTMR	DMKTOD	DMKTPE	DMKTRA
		DMKTRD	DMKTRK	DMKTRP	DMKTRQ	DMKTRT	DMKTRU	DMKTRX	DMKTTX	DMKUNT	DMKURS
		DMKUSQ	DMKVAT	DMKVAU	DMKVCA	DMKVCH	DMKVCN	DMKVCP	DMKVCS	DMKVCV	DMKVDA
		DMKVDC	DMKVDE	DMKVDG	DMKVDR	DMKVER	DMKVFC	DMKVFR	DMKVMA	DMKVMC	DMKVMD
		DMKVME	DMKVMG	DMKVRR	DMKVRS	DMKVSC	DMKVSD	DMKVSI	DMKVSP	DMKVSQ	DMKVST
		DMKVSV	DMKVSX	DMKWRM	DMKWRN	DMKXAB	DMKXST				
F10	000054	DMKCCD	DMKCDB	DMKCDM	DMKCNS	DMKCSO	DMKCVT	DMKDAD	DMKDAS	DMKDAU	DMKDID
		DMKGRC	DMKIOH	DMKIOS	DMKIUC	DMKMSW	DMKPAG	DMKQCQ	DMKSPK	DMKSTA	DMKVDG
		DMKVMD	DMKVSC	DMKXAB							
F15	000107	DMKATS	DMKBLD	DMKBSC	DMKCCD	DMKCCF	DMKCCO	DMKCCT	DMKCCW	DMKCDB	DMKCDS
		DMKCFF	DMKCFR	DMKCNS	DMKCPP	DMKCPZ	DMKDAS	DMKDEG	DMKDGD	DMKDID	DMKFPS
		DMKHPS	DMKIOF	DMKIOJ	DMKMNL	DMKMPO	DMKMSG	DMKPEI	DMKPEQ	DMKPER	DMKPGS
		DMKPRV	DMKPRW	DMKPSA	DMKPTR	DMKQVM	DMKRSE	DMKTAP	DMKTRC	DMKTRD	DMKTTX
		DMKUNT	DMKUSQ	DMKVCH	DMKVDT	DMKVER	DMKVSC	DMKVSP			
F16	000101	DMKATS	DMKBLD	DMKCCD	DMKCCF	DMKCCH	DMKCCT	DMKCCW	DMKCDB	DMKCDS	DMKCFF
		DMKCFG	DMKCFP	DMKCFU	DMKCNS	DMKCPB	DMKCPP	DMKCPU	DMKDAD	DMKDAS	DMKDAU
		DMKDEG	DMKDGD	DMKDGF	DMKHVC	DMKHVD	DMKIOS	DMKISM	DMKMPO	DMKPGM	DMKPGS
		DMKPRV	DMKPTR	DMKPTS	DMKPTT	DMKRNH	DMKSCH	DMKSEG	DMKSEL	DMKSPM	DMKSST
		DMKSTR	DMKTAP	DMKTAQ	DMKTCT	DMKTRC	DMKTRD	DMKUNT	DMKVAT	DMKVDS	DMKVDT
		DMKVFE	DMKVIO	DMKVMA	DMKXAB						
F2	000262	DMKACR	DMKACS	DMKAPY	DMKBIO	DMKCCD	DMKCCH	DMKCCW	DMKCDB	DMKCDM	DMKCDS
		DMKCFF	DMKCFG	DMKCFH	DMKCFJ	DMKCFO	DMKCFP	DMKCFS	DMKCFT	DMKCFU	DMKCFV
		DMKCFY	DMKCKT	DMKCNS	DMKCPI	DMKCPN	DMKCPO	DMKCPP	DMKCPT	DMKCPY	DMKCQH
		DMKCQP	DMKCQQ	DMKCQR	DMKCQT	DMKCQY	DMKCSB	DMKCSF	DMKCSO	DMKCSP	DMKCSQ
		DMKCSR	DMKCST	DMKCSU	DMKCSV	DMKCSX	DMKDAD	DMKDAS	DMKDEF	DMKDEG	DMKDID
		DMKEPS	DMKERM	DMKEXT	DMKGRC	DMKGRD	DMKGRG	DMKGRT	DMKHVC	DMKHVD	DMKHVE
		DMKIOS	DMKISM	DMKLNK	DMKLNM	DMKLOG	DMKMCC	DMKMCD	DMKMCI	DMKMIA	DMKMNJ
		DMKMON	DMKMOO	DMKMPO	DMKMSG	DMKNEA	DMKNES	DMKNLE	DMKPAG	DMKPEI	DMKPEL
		DMKPEN	DMKPEQ	DMKPGM	DMKPMA	DMKPSA	DMKPTR	DMKQCN	DMKQCO	DMKQCQ	DMKRGA
		DMKRGB	DMKRGC	DMKRSP	DMKSAV	DMKSCH	DMKSEG	DMKSEL	DMKSND	DMKSPK	DMKSPT
		DMKSRM	DMKSTP	DMKTAQ	DMKTCS	DMKTCT	DMKTOD	DMKTRA	DMKTRC	DMKTRD	DMKTRP
		DMKTRU	DMKTRX	DMKTTX	DMKUDU	DMKVAT	DMKVAU	DMKVCA	DMKVCH	DMKVCN	DMKVCV
		DMKVDC	DMKVDE	DMKVDF	DMKVMD	DMKVSI	DMKVSP	DMKVST	DMKVSV	DMKWRM	
F20	000009	DMKCPI	DMKHVC	DMKHVD	DMKRSE	DMKVDD					
F24	000016	DMKCDB	DMKCDM	DMKCFT	DMKCSU	DMKHVD	DMKIOJ	DMKRST	DMKSCH	DMKVER	DMKVFC
		DMKVFD	DMKVFE	DMKVSQ							
F240	000031	DMKCCW	DMKCVT	DMKCVU	DMKDIA	DMKDIB	DMKFPS	DMKMNT	DMKMPO	DMKPEI	DMKPRV
		DMKPRW	DMKPSA	DMKTRD	DMKUDU	DMKUNT	DMKVAT	DMKVCA	DMKVCB	DMKVDS	DMKVDT
		DMKVSI									
F255	000090	DMKACO	DMKAPV	DMKBLD	DMKCAC	DMKCCH	DMKCFF	DMKCFM	DMKCFQ	DMKCFT	DMKCFV
		DMKCFY	DMKCKF	DMKCKN	DMKCKT	DMKCKV	DMKCPU	DMKCPZ	DMKCQG	DMKCQU	DMKCSF
		DMKCSP	DMKCSU	DMKDEF	DMKDIA	DMKDIB	DMKDRD	DMKDSP	DMKERM	DMKEXT	DMKGRA

LABEL	COUNT	REFERENCES									
		DMKHVD	DMKHVE	DMKIDU	DMKIOE	DMKIOF	DMKIOG	DMKLOG	DMKLOH	DMKMCH	DMKMCT
		DMKNEA	DMKNES	DMKNET	DMKQCN	DMKQCO	DMKRGB	DMKRGC	DMKRGD	DMKRNH	DMKRSP
		DMKRSQ	DMKSCN	DMKSTA	DMKTDK	DMKTRD	DMKUNT	DMKVAT	DMKVAU	DMKVCN	DMKVDR
		DMKVDS	DMKVSQ	DMKVSR							
F256	000070	DMKATS	DMKCFC	DMKCFF	DMKCFH	DMKCFO	DMKCKT	DMKCKV	DMKCNT	DMKCPP	DMKDAS
		DMKDRD	DMKDRE	DMKEXT	DMKGRI	DMKHVC	DMKHVD	DMKHVF	DMKIOE	DMKIOS	DMKMNT
		DMKNLD	DMKNLE	DMKQCO	DMKQCQ	DMKRGB	DMKRNH	DMKSCH	DMKSNC	DMKTCS	DMKTDK
		DMKVBM	DMKVCN	DMKVMA	DMKVSP	DMKVSQ	DMKVSR	DMKVSV	DMKWRM		
F3	000244	DMKAPS	DMKAPX	DMKCAC	DMKCAO	DMKCCD	DMKCCW	DMKCDB	DMKCDM	DMKCFD	DMKCFF
		DMKCFG	DMKCFH	DMKCFJ	DMKCFO	DMKCFT	DMKCFU	DMKCFW	DMKCFY	DMKCKT	DMKCPB
		DMKCPI	DMKCPT	DMKCPV	DMKCPY	DMKCQC	DMKCQG	DMKCQH	DMKCQP	DMKCQQ	DMKCSB
		DMKCSF	DMKCSO	DMKCSP	DMKCSQ	DMKCSR	DMKCST	DMKCSU	DMKCSV	DMKCSX	DMKDAD
		DMKDEF	DMKDEG	DMKDGD	DMKDIA	DMKDIB	DMKDID	DMKDSP	DMKEXT	DMKGRG	DMKGRT
		DMKHVC	DMKHVD	DMKHVE	DMKLNM	DMKLOH	DMKLOJ	DMKLOM	DMKMCC	DMKMCD	DMKMCH
		DMKMCT	DMKMIA	DMKMON	DMKMOO	DMKMSG	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKNLE
		DMKPAG	DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPET	DMKQCQ	DMKRGB	DMKRGC	DMKRSE
		DMKRSP	DMKSAV	DMKSPK	DMKSPL	DMKSPM	DMKSPT	DMKSRM	DMKSTP	DMKTAP	DMKTCS
		DMKTCT	DMKTHI	DMKTOD	DMKTRA	DMKTRD	DMKTRK	DMKTRP	DMKTTX	DMKUDU	DMKVCH
		DMKVCN	DMKVCU	DMKVDC	DMKVDE	DMKVDH	DMKVFD	DMKVFE	DMKVIO	DMKVMD	DMKVSP
		DMKVSQ	DMKVST	DMKVSV	DMKVSW	DMKVSX					
F4	000286	DMKACS	DMKAPU	DMKATS	DMKBLD	DMKCCD	DMKCCO	DMKCCT	DMKCCW	DMKCDB	DMKCDM
		DMKCDS	DMKCFC	DMKCFD	DMKCFF	DMKCFG	DMKCFH	DMKCFO	DMKCFQ	DMKCFS	DMKCFT
		DMKCFU	DMKCKM	DMKCKT	DMKCNS	DMKCPB	DMKCPI	DMKCPJ	DMKCPN	DMKCPP	DMKCPS
		DMKCPT	DMKCPU	DMKCPV	DMKCPY	DMKCQH	DMKCQQ	DMKCQS	DMKCQY	DMKCSB	DMKCSF
		DMKCSO	DMKCSP	DMKCST	DMKCSU	DMKCSV	DMKCSW	DMKCSX	DMKCVU	DMKDEF	DMKDGD
		DMKDID	DMKEPS	DMKERM	DMKEXT	DMKFPS	DMKGRF	DMKGRI	DMKHPS	DMKHPT	DMKHVC
		DMKHVD	DMKHVE	DMKHVF	DMKIOE	DMKIOF	DMKIOJ	DMKISM	DMKIUA	DMKIUN	DMKJRL
		DMKLNK	DMKLNM	DMKLOG	DMKLOH	DMKLOJ	DMKLOM	DMKMCC	DMKMCD	DMKMIA	DMKMON
		DMKMOO	DMKMSG	DMKMSW	DMKNEA	DMKNES	DMKNET	DMKNLE	DMKOPE	DMKPAG	DMKPEL
		DMKPEN	DMKPEQ	DMKPRV	DMKPRW	DMKPST	DMKRGB	DMKRNH	DMKRPA	DMKRSE	DMKRSP
		DMKSAV	DMKSEG	DMKSEP	DMKSPK	DMKSPM	DMKSPT	DMKSRM	DMKSST	DMKSSV	DMKSTA
		DMKSVD	DMKTAQ	DMKTHI	DMKTMR	DMKTOD	DMKTRD	DMKTRP	DMKUNT	DMKUSQ	DMKVCH
		DMKVCN	DMKVCQ	DMKVCU	DMKVDC	DMKVDE	DMKVDH	DMKVER	DMKVMD	DMKVRR	DMKVSC
		DMKVSP	DMKVSQ	DMKVSR	DMKVST	DMKVSV					
F4095	000097	DMKBLD	DMKCCD	DMKCCO	DMKCCW	DMKCDB	DMKCDM	DMKCDS	DMKCFG	DMKCPJ	DMKCPY
		DMKCSC	DMKDAD	DMKDAS	DMKDAU	DMKDGD	DMKFPS	DMKGRG	DMKHVC	DMKHVD	DMKHVF
		DMKIOE	DMKIOF	DMKIUA	DMKIUE	DMKIUL	DMKLNK	DMKMNI	DMKMON	DMKPER	DMKPMA
		DMKPRV	DMKPRW	DMKPTR	DMKQCN	DMKRGC	DMKRSP	DMKSCN	DMKSTA	DMKSVC	DMKTCS
		DMKTMR	DMKTRT	DMKTRX	DMKVCN	DMKVER	DMKVFR	DMKVIO	DMKVMA	DMKVME	DMKVSF
		DMKVSI	DMKVSQ	DMKVSR	DMKVSV	DMKVSX					
F4096	000361	DMKAPI	DMKAPT	DMKATS	DMKBIO	DMKCCH	DMKCCS	DMKCCW	DMKCDB	DMKCDM	DMKCDS
		DMKCFF	DMKCFG	DMKCFH	DMKCFP	DMKCFS	DMKCKF	DMKCKH	DMKCKM	DMKCKP	DMKCKR
		DMKCKS	DMKCKV	DMKCKW	DMKCPI	DMKCPP	DMKCPS	DMKCPU	DMKCPY	DMKCQY	DMKCSC
		DMKCSO	DMKCSQ	DMKDAD	DMKDAS	DMKDAU	DMKDGD	DMKDRD	DMKDRE	DMKDSP	DMKFPS
		DMKFRE	DMKFRT	DMKGRC	DMKGRE	DMKGRF	DMKGRI	DMKHPS	DMKHPT	DMKHPU	DMKHVC
		DMKHVD	DMKHVE	DMKHVF	DMKIDU	DMKLOH	DMKMCH	DMKMNT	DMKMOO	DMKMPO	DMKNLD
		DMKNLE	DMKPGS	DMKPMA	DMKPRV	DMKPRW	DMKPSA	DMKPST	DMKPTR	DMKPTT	DMKQVM
		DMKRSP	DMKSAV	DMKSCN	DMKSEG	DMKSEL	DMKSNC	DMKSPL	DMKSPM	DMKSST	DMKSTA
		DMKSTR	DMKTCS	DMKTRK	DMKTRT	DMKTRU	DMKTRX	DMKUDR	DMKVAT	DMKVBM	DMKVCN
		DMKVDG	DMKVFC	DMKVFD	DMKVFE	DMKVFS	DMKVIO	DMKVMD	DMKVME	DMKVRR	DMKVRS
		DMKVSC	DMKVSF	DMKVSP	DMKVSQ	DMKVSR	DMKVST	DMKVSV	DMKWRM	DMKXAB	DMKXST
F5	000084	DMKCDS	DMKCFO	DMKCFT	DMKCFU	DMKCPT	DMKCPV	DMKCQS	DMKCSU	DMKDEF	DMKDEG
		DMKDGD	DMKDRD	DMKDRE	DMKEXT	DMKGRE	DMKHVC	DMKIUA	DMKLNM	DMKLOG	DMKMCC
		DMKMCD	DMKMIA	DMKMNJ	DMKMNL	DMKMOO	DMKMSG	DMKNLE	DMKPAH	DMKPEI	DMKPEL
		DMKPEQ	DMKPRV	DMKRSE	DMKSCN	DMKSPT	DMKTAP	DMKTAQ	DMKTMR	DMKTOD	DMKVDA
		DMKVDE	DMKVDF	DMKVFC	DMKVMD	DMKVME	DMKVSP	DMKVST			
F6	000107	DMKAPS	DMKCDB	DMKCDM	DMKCDS	DMKCFD	DMKCFH	DMKCFJ	DMKCFU	DMKCKT	DMKCPB
		DMKCQP	DMKCSO	DMKCSU	DMKCSV	DMKCSX	DMKDGD	DMKEXT	DMKFPS	DMKHVC	DMKHVE

LABEL	COUNT	REFERENCES									
		DMKIOH	DMKIUA	DMKIUG	DMKLNM	DMKLOG	DMKLOH	DMKMCH	DMKMOO	DMKMSG	DMKMSW
		DMKNEA	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKPAG	DMKPEI	DMKPEL	DMKPEQ	DMKPRV
		DMKPTS	DMKQVM	DMKRGA	DMKRGC	DMKRST	DMKSCH	DMKSPT	DMKSWM	DMKTAP	DMKTPE
		DMKTRK	DMKUNT	DMKVDC	DMKVDE	DMKVDF	DMKVMD	DMKVME	DMKVSP	DMKVST	
F60	000043	DMKACO	DMKCFU	DMKCFU	DMKCKM	DMKCQS	DMKCVT	DMKFPS	DMKHVC	DMKHVE	DMKIOE
		DMKIUA	DMKMCD	DMKMID	DMKNET	DMKPRV	DMKPRW	DMKPSA	DMKRNH	DMKSVD	DMKTHI
		DMKTMR	DMKTOD	DMKTRC	DMKTRD	DMKVST					
F7	000084	DMKBLD	DMKBSC	DMKCCF	DMKCCH	DMKCCW	DMKCDS	DMKCFF	DMKCFG	DMKCFU	DMKCFY
		DMKCPI	DMKCPP	DMKCPT	DMKCPU	DMKCQG	DMKCSQ	DMKCSU	DMKDIB	DMKHPT	DMKHPU
		DMKHVF	DMKIOE	DMKIOF	DMKIOG	DMKIOJ	DMKLNK	DMKLOG	DMKLOJ	DMKMCD	DMKMCH
		DMKMOO	DMKMSG	DMKNES	DMKNET	DMKPEI	DMKPRV	DMKQCO	DMKRSE	DMKSCN	DMKSEV
		DMKSIX	DMKSPT	DMKTMR	DMKTOD	DMKTRM	DMKUDU	DMKUNT	DMKVCA	DMKVCN	DMKVDA
		DMKVDC	DMKVDT	DMKVER	DMKVMC	DMKVSP	DMKVST				
F8	000362	DMKACS	DMKATS	DMKBLD	DMKBSC	DMKCAC	DMKCCD	DMKCCF	DMKCCT	DMKCCW	DMKCDS
		DMKCFC	DMKCFF	DMKCFH	DMKCFO	DMKCFQ	DMKCFS	DMKCFT	DMKCFU	DMKCFV	DMKCFW
		DMKCFY	DMKCNS	DMKCPJ	DMKCPN	DMKCPP	DMKCPT	DMKCPV	DMKCQH	DMKCQP	DMKCQQ
		DMKCQR	DMKCQT	DMKCQY	DMKCSB	DMKCSF	DMKCSO	DMKCSP	DMKCSQ	DMKCST	DMKCSU
		DMKCSV	DMKCSW	DMKCSX	DMKDAD	DMKDAS	DMKDAU	DMKDEF	DMKDEG	DMKDEI	DMKDGD
		DMKDGF	DMKDID	DMKDRD	DMKEPS	DMKEXT	DMKGRC	DMKGRF	DMKGRI	DMKHVC	DMKHVD
		DMKIOE	DMKIOJ	DMKIOS	DMKIOT	DMKISM	DMKJRL	DMKLNK	DMKLOG	DMKLOJ	DMKMCC
		DMKMCD	DMKMCI	DMKMHC	DMKMNJ	DMKMOO	DMKMSG	DMKMSW	DMKNEA	DMKNES	DMKNET
		DMKNLD	DMKNLE	DMKOPE	DMKPAG	DMKPEI	DMKPEN	DMKPEQ	DMKPGS	DMKPMA	DMKPRV
		DMKPRW	DMKPSA	DMKPST	DMKPTT	DMKQCO	DMKQVM	DMKRGA	DMKRNH	DMKRSE	DMKRSF
		DMKRSP	DMKRST	DMKSCH	DMKSEL	DMKSEV	DMKSIX	DMKSPK	DMKSPL	DMKSPT	DMKSRM
		DMKSSS	DMKSST	DMKSSV	DMKSTA	DMKSTR	DMKSWA	DMKSWM	DMKTAP	DMKTAQ	DMKTHI
		DMKTOD	DMKTPE	DMKTRA	DMKTRC	DMKTRD	DMKTRK	DMKTRP	DMKTTY	DMKUDR	DMKVAT
		DMKVCN	DMKVCQ	DMKVCT	DMKVDA	DMKVDC	DMKVDD	DMKVDE	DMKVDF	DMKVDS	DMKVER
		DMKVFE	DMKVIO	DMKVMA	DMKVMD	DMKVSC	DMKVSI	DMKVSP	DMKXST	DMKZTD	
F9	000023	DMKACS	DMKCAC	DMKCCD	DMKCCO	DMKCFC	DMKCPT	DMKCPY	DMKEXT	DMKGRD	DMKHVE
		DMKMCD	DMKMSG	DMKQCN	DMKQCQ	DMKSEP	DMKUNT	DMKVDC	DMKVDG		
GRACDAPL	000003	DMKGRI									
GRACDTXT	000004	DMKGRI									
GRADEV77	000003	DMKGRI									
GRADEV78	000005	DMKGRI									
GRAFDEV	000007	DMKDIF	DMKQCN								
GRAFESC	000003	DMKGRI									
GRCALRM	000009	DMKGRD	DMKGRG								
GRCCLRIN	000007	DMKGRD	DMKGRG								
GRCCLROT	000005	DMKGRG									
GRCCPRD	000006	DMKGRD									
GRCECOL	000006	DMKGRD	DMKGRG								
GRCEHLT	000006	DMKGRD	DMKGRG								
GRCHOLD	000006	DMKGRG									
GRCINH	000007	DMKGRD									
GRCMORE	000006	DMKGRG									
GRCMXDLN	000007	DMKGRD	DMKGRF								
GRCMXLEN	000006	DMKGRG									
GRCNAC	000006	DMKGRG									
GRCRUN	000020	DMKGRF	DMKGRG								
GRCSFIN	000018	DMKGRD	DMKGRG								
GRCSFSTA	000015	DMKGRD	DMKGRG								
GRCVMRD	000006	DMKGRD									
GREBI	000004	DMKGRD	DMKGRE	DMKGRF	DMKGRG						
GREBT	000004	DMKGRD	DMKGRE	DMKGRF	DMKGRG						
GRECR	000004	DMKGRD	DMKGRE	DMKGRF	DMKGRG						
GREC1	000004	DMKGRD	DMKGRE	DMKGRF	DMKGRG						
GREC2	000002	DMKGRD	DMKGRF								
GREDATNR	000003	DMKGRE	DMKGRF	DMKGRG							

LABEL	COUNT	REFERENCES			
GREDBCCW	000003	DMKGRE	DMKGRF	DMKGRG	
GREDCIOS	000003	DMKGRE	DMKGRF	DMKGRG	
GREDCK1D	000003	DMKGRE	DMKGRF	DMKGRG	
GREDCVMP	000003	DMKGRE	DMKGRF	DMKGRG	
GREDFIOB	000001	DMKGRF			
GREDIIOB	000002	DMKGRF	DMKGRI		
GREDNCTL	000003	DMKGRE	DMKGRF	DMKGRG	
GREDNSEL	000003	DMKGRE	DMKGRF	DMKGRG	
GREDRD77	000001	DMKGRG			
GREDRIOB	000004	DMKGRE	DMKGRF	DMKGRG	DMKGRI
GREDRSCP	000003	DMKGRE	DMKGRF	DMKGRG	
GREDWBOK	000003	DMKGRE	DMKGRF	DMKGRG	
GREFC	000002	DMKGRE	DMKGRG		
GREFCCLR	000001	DMKGRG			
GREFI	000004	DMKGRD	DMKGRE	DMKGRF	DMKGRG
GREFL	000003	DMKGRD	DMKGRE	DMKGRF	
GREFLTST	000001	DMKGRG			
GREFR	000003	DMKGRE	DMKGRF	DMKGRG	
GREFS	000002	DMKGRE	DMKGRF		
GREFSINT	000001	DMKGRG			
GREFSMOR	000002	DMKGRD	DMKGRG		
GREFSM77	000001	DMKGRG			
GREFSREJ	000001	DMKGRG			
GREFSR77	000001	DMKGRG			
GREFSWNG	000001	DMKGRG			
GREFSW77	000001	DMKGRG			
GREFT	000003	DMKGRE	DMKGRF	DMKGRG	
GREFTFMT	000001	DMKGRD			
GREFUTCP	000001	DMKGRE			
GREGB77C	000001	DMKGRE			
GREGC	000004	DMKGRD	DMKGRE	DMKGRF	DMKGRG
GREGCMOR	000001	DMKGRD			
GREGGCLR	000001	DMKGRD			
GREGGCLT	000001	DMKGRF			
GREGGRUN	000001	DMKGRD			
GREGL	000003	DMKGRD	DMKGRE	DMKGRF	
GREGN	000003	DMKGRD	DMKGRE	DMKGRF	
GREGS	000003	DMKDMQ			
GREGSBUF	000001	DMKGRF			
GREGTHLD	000001	DMKGRD			
GREIC	000004	DMKGRD	DMKGRE	DMKGRF	DMKGRG
GREIS	000004	DMKGRD	DMKGRE	DMKGRF	DMKGRG
GRERT	000004	DMKGRD	DMKGRE	DMKGRF	DMKGRG
GRESR	000004	DMKGRD	DMKGRE	DMKGRF	DMKGRG
GREST	000003	DMKGRD	DMKGRE	DMKGRF	
GRESV	000004	DMKGRD	DMKGRE	DMKGRF	DMKGRG
GREWS	000003	DMKGRD	DMKGRE	DMKGRG	
GRLOG	000005	DMKDMP	DMKPMA	DMKSAD	
GRTABLEN	000002	DMKGRG			
GRTBLOK	000007	DMKGRD			
GRTHILEN	000001	DMKQCQ			
GRTSALEN	000007	DMKQCN			
GT16MEG	000005	DMKCCW			
G16CHAN	000003	DMKIOG	DMKVER		
HADAPOLL	000003	DMKCCT	DMKCCW		
HADISAM	000007	DMKCCD	DMKCCO	DMKCCW	
HADRCGEN	000005	DMKCCD	DMKCCF	DMKCCW	
HADUTIC	000004	DMKCCW			

LABEL	COUNT	REFERENCES					
HALFPAGE	000010	DMKDMP					
HALF1	000006	DMKTRP	DMKTRR	DMKTRT			
HALF1ENT	000011	DMKCQC	DMKTRP	DMKTRR	DMKTRT		
HALF1EN1	000009	DMKCQC	DMKTRP	DMKTRR			
HALF1EN2	000004	DMKTRP	DMKTRR				
HALF1RLN	000005	DMKCQC	DMKTRP	DMKTRR			
HALF1SZE	000018	DMKCQC	DMKTRP	DMKTRR	DMKTRT	DMKTRX	
HALF1VAL	000019	DMKCQC	DMKTRP	DMKTRR	DMKTRT		
HALF2	000002	DMKTRP	DMKTRR				
HARDSTOP	000003	DMKCPI	DMKCPJ	DMKDMP			
HBOUND	000004	DMKFRE					
HCBLOK	000027	DMKCPU	DMKMHC	DMKMHV	DMKPST	DMKVFC	DMKXST
HCBPNT	000005	DMKMHC					
HCCVU	000001	DMKVFC					
HCDVU	000001	DMKVFC					
HCFLAG	000009	DMKMHC					
HCFPNT	000013	DMKMHC					
HCIRTN	000001	DMKMHC					
HCMSFACT	000001	DMKMHC					
HCMSFCW	000008	DMKCPU	DMKMHC	DMKMHV	DMKPST	DMKVFC	DMKXST
HCMSFDB	000015	DMKCPU	DMKMHC	DMKMHV	DMKPST	DMKVFC	DMKXST
HCMSFDBV	000003	DMKMHC	DMKMHV				
HCMSFSYS	000003	DMKMHC					
HCREG13	000008	DMKMHC					
HCRMAP	000002	DMKPST	DMKXST				
HCRSCP	000002	DMKMHC	DMKPST				
HCSIZE	000012	DMKMHC	DMKMHV	DMKPST	DMKVFC	DMKXST	
HCVMBLK	000006	DMKMHC					
HCVMREQ	000004	DMKMHC					
HCVRX	000002	DMKMHC					
HEADER	000012	DMKCQY	DMKDDR	DMKMCC	DMKSAD	DMKSRM	DMKVMD
HILIGHT	000006	DMKCFM	DMKGRG	DMKMSG	DMKRGC		
HIOCCH	000006	DMKCCH	DMKSEV	DMKSIX			
HLDADDR1	000004	DMKTCS	DMKTCT				
HLDAREA	000085	DMKTCS	DMKTCT				
HLDCCUU	000005	DMKTCT					
HLDCCW1	000015	DMKTCS	DMKTCT				
HLDCCW2	000016	DMKTCS	DMKTCT				
HLDCCW3	000009	DMKTCS	DMKTCT				
HLDCCW4	000005	DMKTCS	DMKTCT				
HLDCCW5	000005	DMKTCS	DMKTCT				
HLDCCW6	000005	DMKTCS	DMKTCT				
HLDCCW7	000002	DMKTCS					
HLDCHARS	000023	DMKTCS	DMKTCT				
HLDCHAR1	000012	DMKTCS	DMKTCT				
HLDCHAR2	000012	DMKTCS	DMKTCT				
HLDCHAR3	000012	DMKTCS	DMKTCT				
HLDCPY	000006	DMKTCS					
HLDFCB	000025	DMKTCS					
HLDFLAG	000004	DMKTCS					
HLDFLSHC	000010	DMKTCS					
HLDHLPU	000005	DMKTCT					
HLDMCHR	000004	DMKTCS					
HLDMDFY	000005	DMKTCS					
HLDMNAM	000010	DMKTCT					
HLDSFID	000005	DMKTCT					
HLDSIZE	000023	DMKTCS					
HLDSTCPY	000005	DMKTCS					

LABEL	COUNT	REFERENCES									
HMAXSIZE	000005	DMKFRE									
HWORD	000011	DMKFRE									
H7	000001	DMKDIR									
IC	000034	DMKBOX	DMKGRE	DMKGRG	DMKGRI						
IDA	000104	DMKCCD	DMKCCF	DMKCCO	DMKCCS	DMKCCT	DMKCCW	DMKCKN	DMKDAD	DMKDAS	DMKDAU
		DMKDGD	DMKDIB	DMKDMP	DMKGRE	DMKGRF	DMKHPS	DMKHPU	DMKISM	DMKPAG	DMKQCO
		DMKRGD	DMKTAQ	DMKTCS	DMKTCT	DMKTRD	DMKUNT	DMKVCA	DMKVCB	DMKVCN	DMKVSC
		DMKVSP	DMKVSX								
IDAENTRY	000007	DMKCFQ	DMKRGD	DMKVCN	DMKVCR	DMKVCV	DMKVDR				
IDAEWRD1	000011	DMKCFQ	DMKRGD	DMKVCN	DMKVCR	DMKVCV	DMKVDR				
IDAEWRD2	000005	DMKVCN	DMKVCR	DMKVCV							
IDAHCNT	000005	DMKVCN									
IDAHCURR	000008	DMKRGD	DMKVCN								
IDAHEAD	000037	DMKCFQ	DMKGRE	DMKQCN	DMKRGA	DMKRGB	DMKRGD	DMKRGE	DMKVCN	DMKVCR	DMKVCV
		DMKVDR									
IDAHPAGE	000005	DMKCFQ	DMKVCN	DMKVCV	DMKVDR						
IDAHSIZD	000004	DMKCFQ	DMKVCN	DMKVDR							
IDAHSTRT	000036	DMKCFQ	DMKGRE	DMKQCN	DMKRGA	DMKRGB	DMKRGD	DMKVCN	DMKVCR	DMKVCV	DMKVDR
IDAHWRK1	000020	DMKCFQ	DMKRGA	DMKRGE	DMKVCN	DMKVCR	DMKVDR				
IDAHWRK2	000012	DMKRGA	DMKRGE	DMKVCN	DMKVCR						
IDLEWAIT	000019	DMKAPI	DMKCLK	DMKCPI	DMKDSP	DMKMON	DMKMOO	DMKSTP			
IETYPMNP	000001	DMKIUN									
IETYPMP	000001	DMKIUN									
IETYPRNP	000001	DMKIUA									
IETYPRP	000001	DMKIUA									
IFCC	000097	DMKBSC	DMKCCH	DMKCNS	DMKCPI	DMKCPM	DMKCPW	DMKCPZ	DMKCQT	DMKDAD	DMKDAS
		DMKDAU	DMKDDR	DMKDID	DMKDMQ	DMKDSB	DMKDSP	DMKEIG	DMKGIO	DMKGRF	DMKHVC
		DMKIOE	DMKIOS	DMKIOT	DMKMNT	DMKMSW	DMKRSE	DMKRSF	DMKRSP	DMKSAD	DMKSEV
		DMKSIX	DMKTAP	DMKTAQ	DMKTPE	DMKUNT	DMKVIO	DMKVSJ			
IGBLAME	000027	DMKEIG	DMKSEV	DMKSIX							
IGPRGFLG	000009	DMKCCH	DMKSEV	DMKSIX							
IGTERMSQ	000048	DMKCCH	DMKEIG	DMKSEV	DMKSIX						
IGVALIDB	000029	DMKCCH	DMKEIG	DMKSEV	DMKSIX						
IL	000061	DMKBIO	DMKCNS	DMKDAD	DMKDAS	DMKDIB	DMKGIO	DMKHPS	DMKHPU	DMKHVC	DMKIOS
		DMKIOT	DMKNLD	DMKNLE	DMKPAH	DMKRNH	DMKRST	DMKSPS	DMKTAQ	DMKTPE	DMKTRK
		DMKVCA	DMKVCN	DMKVIO	DMKVMI	DMKVSP	DMKVSX				
IMGBUFLN	000003	DMKCSB	DMKCSC								
IMGCCWDS	000001	DMKCSB									
IMGCCWLN	000001	DMKCSB									
IMGFCBF1	000002	DMKCSB									
IMGFCBPI	000003	DMKCSB									
IMGFCBTI	000002	DMKCSB									
IMGF3211	000002	DMKCSB									
IMGHDR	000003	DMKCSB	DMKCSC								
IMGHDRSD	000003	DMKCSB	DMKCSC								
IMGHDRSZ	000008	DMKCSC									
IMGNAME	000003	DMKCSB	DMKCSC								
IMGNEXT	000003	DMKCSC									
INBUFF	000003	DMKOVR									
INCRBY1	000011	DMKFRE									
INFILE	000005	DMKOVR									
INHIBIT	000038	DMKACO	DMKCNS	DMKEPS	DMKGRD	DMKGRG	DMKLOG	DMKQCN	DMKQCQ	DMKRGA	DMKRGB
		DMKRGC	DMKRNH	DMKTTX	DMKTTY	DMKVCN	DMKVCR				
INMSFBLK	000003	DMKDSP	DMKEXT	DMKMHC							
INPUT	000032	DMKDDR	DMKRND	DMKSSP							
INTBLK	000009	DMKTRR	DMKTRT								
INTCODE	000001	DMKTRT									
INTCTRL	000003	DMKTRR	DMKTRT								

LABEL	COUNT	REFERENCES									
INTERCCH	000005	DMKCCH	DMKEIG	DMKSEV							
INTEX	000019	DMKDSP	DMKEXT	DMKMPO	DMKPMA						
INTEXF	000009	DMKDSP	DMKEXT	DMKMPO	DMKPMA						
INTKFLIN	000003	DMKCKH	DMKCKP								
INTLTH	000006	DMKTRR	DMKTRT								
INTMASK	000001	DMKCPI									
INTMC	000005	DMKMCH	DMKPMA								
INTMTYPE	000003	DMKTRR	DMKTRT								
INTNULL	000002	DMKTRR									
INTPR	000049	DMKCKP	DMKCPI	DMKCPY	DMKPGS	DMKPMA	DMKPRG	DMKPRV	DMKPRW	DMKSAD	DMKSTA
		DMKVAT									
INTPRL	000023	DMKDMP	DMKDSP	DMKFPS	DMKPMA	DMKPRG	DMKPRV				
INTRC	000007	DMKMCH	DMKPMA								
INTREQ	000067	DMKCNS	DMKCPI	DMKCPM	DMKCPS	DMKCQT	DMKDAD	DMKDDR	DMKDIB	DMKDID	DMKDMP
		DMKFMT	DMKGRF	DMKGRG	DMKHPS	DMKHPT	DMKIOS	DMKIOT	DMKMNT	DMKMSW	DMKNLD
		DMKNLE	DMKRNH	DMKRSE	DMKSAV	DMKVCA	DMKVCN	DMKVIO	DMKVSI	DMKVSP	DMKVSX
INTSVC	000008	DMKSVC	DMKSVD								
INTSVCL	000021	DMKCKP	DMKSVC	DMKSVD	DMKTRC						
INTTCP	000005	DMKTRR	DMKTRX								
INTTIO	000047	DMKCCH	DMKCKH	DMKCKN	DMKCKP	DMKDMP	DMKDMQ	DMKDSP	DMKIOT	DMKMNT	DMKOPE
		DMKOPR	DMKSAD	DMKSSU	DMKVMI	DMKVRR	DMKVRS				
INTTMACH	000001	DMKTRR									
INTTVG	000005	DMKTRR	DMKTRX								
INTTVT	000005	DMKTRR	DMKTRX								
INTTYPE	000012	DMKTRR	DMKTRT								
INVCCW	000005	DMKCCD	DMKCCO	DMKCCS							
INVCCW1	000004	DMKCCD	DMKCCF	DMKCCO	DMKCCS						
INVLD	000002	DMKCDB	DMKCDM								
IOBALTSK	000008	DMKCCD	DMKDAS	DMKTRK	DMKUNT	DMKVIO					
IOBASN	000002	DMKCPN									
IOBASNCT	000009	DMKCPN	DMKCPW	DMKTPE							
IOBBADCH	000010	DMKCPN	DMKCPT								
IOBBPNT	000037	DMKACS	DMKCFQ	DMKCPP	DMKDSP	DMKIOQ	DMKIOS	DMKIOT	DMKNLD	DMKPAG	DMKQVM
		DMKSSU	DMKVRR								
IOBCAW	000310	DMKACS	DMKBIO	DMKBSC	DMKCAC	DMKCCH	DMKCCW	DMKCFQ	DMKCFR	DMKCNS	DMKCPB
		DMKCPN	DMKCPS	DMKCPW	DMKCPZ	DMKCQT	DMKCSB	DMKDAD	DMKDAS	DMKDAU	DMKDEX
		DMKDGD	DMKDGF	DMKDIB	DMKDID	DMKDSB	DMKGIO	DMKGRD	DMKGRF	DMKGRG	DMKGRH
		DMKGRI	DMKHPS	DMKHPT	DMKHVC	DMKIOH	DMKIOS	DMKIOT	DMKISM	DMKMCC	DMKMNI
		DMKMNL	DMKMON	DMKNLD	DMKNLE	DMKPAG	DMKPAH	DMKQVM	DMKRGA	DMKRGB	DMKRGC
		DMKRNH	DMKRSE	DMKRSF	DMKRSP	DMKRST	DMKSEP	DMKSPK	DMKSPT	DMKTAP	DMKTAQ
		DMKTCS	DMKTCT	DMKTPE	DMKTRD	DMKTRK	DMKUDR	DMKUNT	DMKVCA	DMKVDE	DMKVDR
		DMKVDT	DMKVIO	DMKVSI	DMKZTD						
IOBCCH	000006	DMKCCH									
IOBCC1	000025	DMKCNS	DMKDGF	DMKDIB	DMKIOQ	DMKIOS	DMKIOT	DMKNLD	DMKNLE	DMKRNH	DMKRSE
		DMKRSP	DMKRST	DMKVCA	DMKVDE						
IOBCC2	000004	DMKIOH	DMKIOS	DMKVIO							
IOBCC3	000062	DMKACS	DMKCAC	DMKCFQ	DMKCNS	DMKCPN	DMKCPW	DMKCPZ	DMKCQT	DMKDGF	DMKDIB
		DMKDSB	DMKGIO	DMKIOQ	DMKIOS	DMKIOT	DMKNLD	DMKNLE	DMKPAG	DMKRGA	DMKRNH
		DMKRSE	DMKRSF	DMKSSU	DMKTAP	DMKTRK	DMKUNT	DMKVCA	DMKVDC	DMKVDE	DMKVIO
		DMKVSI	DMKZTD								
IOBCLN	000004	DMKCCW	DMKGIO	DMKUNT	DMKVSI						
IOBCLRIO	000010	DMKDID	DMKIOS	DMKIOT							
IOBCOPY	000006	DMKGRF	DMKGRG								
IOBCP	000096	DMKACR	DMKACS	DMKCCH	DMKCPS	DMKCPZ	DMKCSF	DMKCSO	DMKDAD	DMKDAS	DMKDAU
		DMKDIA	DMKDID	DMKHPS	DMKHPT	DMKHPU	DMKIOE	DMKIOQ	DMKIOS	DMKIOT	DMKMNL
		DMKNLD	DMKNLE	DMKPAG	DMKRGB	DMKRNH	DMKRSP	DMKRST	DMKSPK	DMKTAP	DMKTAQ
		DMKTPE	DMKTRK	DMKUDR	DMKVDC	DMKVDE	DMKVDT	DMKZTD			
IOBCSW	000575	DMKACO	DMKACS	DMKBIO	DMKBSC	DMKCCH	DMKCCS	DMKCFR	DMKCNS	DMKCPM	DMKCPW

LABEL	COUNT	REFERENCES									
		DMKCPZ	DMKCQT	DMKCSB	DMKCSF	DMKCSO	DMKCSR	DMKCSW	DMKCSX	DMKDAD	DMKDAS
		DMKDAU	DMKDGF	DMKDIB	DMKDID	DMKDIF	DMKDSB	DMKDSP	DMKGIO	DMKGRC	DMKGRE
		DMKGRF	DMKGRG	DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKIOH	DMKIOQ	DMKIOS	DMKIOT
		DMKMON	DMKNLD	DMKNLE	DMKPAG	DMKPAH	DMKQVM	DMKRGA	DMKRGB	DMKRGC	DMKRGE
		DMKRNH	DMKRSE	DMKRSP	DMKRST	DMKSEP	DMKSPL	DMKSPM	DMKSPS	DMKSPT	DMKSSU
		DMKTAP	DMKTAQ	DMKTRC	DMKTRD	DMKTRK	DMKUNT	DMKVCA	DMKVCB	DMKVCN	DMKVCP
		DMKVDC	DMKVDD	DMKVDE	DMKVIO	DMKVRR	DMKVRS	DMKVSI	DMKVSJ	DMKZTD	
IOBCTRQ	000008	DMKIOQ	DMKSSU								
IOBCUBSY	000030	DMKCFQ	DMKIOQ	DMKIOS	DMKIOT	DMKSPM	DMKVIO	DMKVSJ			
IOBCUE	000004	DMKSPM	DMKVIO								
IOBCYL	000040	DMKBIO	DMKCCD	DMKCCF	DMKCCO	DMKCCW	DMKDGD	DMKIOQ	DMKIOS	DMKMON	DMKPAG
		DMKSPK	DMKSSU	DMKZTD							
IOBDIAG	000003	DMKCCW	DMKVIO								
IOBDYNP	000003	DMKCPN	DMKCPW	DMKTPE							
IOBERCNT	000006	DMKIOS									
IOBERP	000066	DMKBSC	DMKCNS	DMKDAD	DMKDAS	DMKDAU	DMKGRI	DMKIOQ	DMKIOS	DMKIOT	DMKRSE
		DMKRSF	DMKRSP	DMKTAP	DMKTAQ	DMKTPE					
IOBFATAL	000161	DMKACS	DMKBIO	DMKBSC	DMKCAC	DMKCCH	DMKCFQ	DMKCNS	DMKCPN	DMKCPS	DMKCPW
		DMKCQT	DMKCSB	DMKCSO	DMKDAD	DMKDAS	DMKDAU	DMKDGF	DMKDID	DMKDSB	DMKGIO
		DMKGRF	DMKIOE	DMKIOH	DMKIOJ	DMKIOQ	DMKIOS	DMKIOT	DMKMNI	DMKMNL	DMKMON
		DMKPAG	DMKPAH	DMKRGA	DMKRGB	DMKRSE	DMKRSF	DMKRSP	DMKRST	DMKSEP	DMKSPK
		DMKSPS	DMKSSU	DMKTAP	DMKTAQ	DMKTCS	DMKTCT	DMKTPE	DMKTRK	DMKUDR	DMKVDE
		DMKVIO	DMKZTD								
IOBFCNS	000005	DMKDID	DMKGRF								
IOBFH	000007	DMKIOQ	DMKIOS	DMKMON	DMKPAG						
IOBFLAG	000338	DMKACR	DMKACS	DMKBSC	DMKCCH	DMKCCO	DMKCCW	DMKCFR	DMKCNS	DMKCPS	DMKCPZ
		DMKCSB	DMKCSF	DMKCSO	DMKDAD	DMKDAS	DMKDAU	DMKDGD	DMKDGF	DMKDIA	DMKDIB
		DMKDID	DMKDSP	DMKGIO	DMKGRI	DMKHPS	DMKHPT	DMKHPU	DMKIOE	DMKIOH	DMKIOQ
		DMKIOS	DMKIOT	DMKMNI	DMKMNL	DMKMON	DMKNLD	DMKNLE	DMKPAG	DMKRGB	DMKRNH
		DMKRSE	DMKRSF	DMKRSP	DMKRST	DMKSEP	DMKSPK	DMKSPS	DMKSPT	DMKSSU	DMKTAP
		DMKTAQ	DMKTCS	DMKTCT	DMKTPE	DMKTRK	DMKUDR	DMKUNT	DMKVCA	DMKVDC	DMKVDE
		DMKVDR	DMKVDT	DMKVIO	DMKVSC	DMKVSI	DMKZTD				
IOBFLT	000006	DMKIOQ	DMKSSU								
IOBFPNT	000097	DMKACR	DMKACS	DMKCFQ	DMKCPP	DMKDID	DMKDSP	DMKHPS	DMKHPT	DMKHPU	DMKIOQ
		DMKIOS	DMKIOT	DMKNLD	DMKPAG	DMKQVM	DMKSSU	DMKSTK	DMKVIO	DMKVRR	DMKVRS
		DMKVSC	DMKVSJ	DMKWAI							
IOBFUNCT	000002	DMKCPN	DMKCPW								
IOBHIO	000041	DMKCCH	DMKCFR	DMKDID	DMKIOS	DMKIOT	DMKVIO	DMKVSJ			
IOBHVC	000038	DMKACS	DMKCCH	DMKCCW	DMKCFR	DMKCPS	DMKDGD	DMKDGF	DMKDID	DMKGIO	DMKIOE
		DMKIOQ	DMKIOS	DMKIOT	DMKTAP	DMKTRK					
IOBIMSTK	000004	DMKCPS	DMKIOS								
IOBIOER	000238	DMKACS	DMKBIO	DMKBSC	DMKCCH	DMKCFQ	DMKCFR	DMKCNS	DMKCPN	DMKCPS	DMKCPW
		DMKCPZ	DMKCQT	DMKDAD	DMKDAS	DMKDAU	DMKDGF	DMKDIA	DMKDIB	DMKDID	DMKDSB
		DMKGIO	DMKGRF	DMKGRI	DMKHPS	DMKHPT	DMKIOE	DMKIOS	DMKIOT	DMKMNI	DMKMNL
		DMKMON	DMKNLD	DMKNLE	DMKRGA	DMKRGB	DMKRGE	DMKRNH	DMKRSE	DMKRSF	DMKRSP
		DMKSPS	DMKTAP	DMKTAQ	DMKTPE	DMKTRK	DMKVCA	DMKVCN	DMKVDC	DMKVDE	DMKVIO
IOBIOLOK	000004	DMKIOS									
IOBIRA	000132	DMKACO	DMKBIO	DMKCAC	DMKCFQ	DMKCFR	DMKCNS	DMKCPB	DMKCPM	DMKCPN	DMKCPS
		DMKCPW	DMKCPZ	DMKCQT	DMKCSB	DMKCSF	DMKCSO	DMKCSR	DMKCSW	DMKCSX	DMKDGD
		DMKDIA	DMKDID	DMKDIF	DMKDSB	DMKDSP	DMKGIO	DMKGRI	DMKHPT	DMKHPU	DMKIOH
		DMKIOQ	DMKIOS	DMKIOT	DMKISM	DMKMNI	DMKMNL	DMKNLD	DMKNLE	DMKPAG	DMKRGA
		DMKRGB	DMKRGC	DMKRGE	DMKRNH	DMKRSF	DMKRSP	DMKSEP	DMKSPK	DMKSPL	DMKSPM
		DMKSPT	DMKSSU	DMKTAP	DMKTCS	DMKTCT	DMKTRK	DMKUDR	DMKUNT	DMKVCA	DMKVCB
		DMKVCN	DMKVCP	DMKVDC	DMKVDD	DMKVDE	DMKVDR	DMKVDT	DMKVSI	DMKVSJ	DMKZTD
IOBLDCCW	000003	DMKVDR									
IOBLDRUN	000002	DMKIOS	DMKVDR								
IOBLDSZ	000004	DMKIOS	DMKVDR								
IOBLDTXT	000002	DMKVDR									

LABEL	COUNT	REFERENCES									
IOBLINK	000116	DMKACO	DMKACR	DMKACS	DMKCAC	DMKCFR	DMKCNS	DMKCPM	DMKCPN	DMKCPS	DMKCPW
		DMKCPZ	DMKCSB	DMKCSF	DMKCSO	DMKCSR	DMKCSW	DMKCSX	DMKDGF	DMKDIA	DMKDID
		DMKDSB	DMKGIO	DMKGRD	DMKGRF	DMKGRG	DMKGRI	DMKHPS	DMKHPT	DMKHPU	DMKIOH
		DMKIOQ	DMKIOS	DMKIOT	DMKMNL	DMKNLD	DMKNLE	DMKPAG	DMKQVM	DMKRGA	DMKRGB
		DMKRGC	DMKRGE	DMKRNH	DMKRSP	DMKRST	DMKSEP	DMKSPL	DMKSPM	DMKSSU	DMKTAP
		DMKTCS	DMKTCT	DMKUNT	DMKVCA	DMKVCB	DMKVCN	DMKVCP	DMKVDD	DMKVDT	DMKVIO
		DMKVRR	DMKVSI	DMKZTD							
IOBLOK	000685	DMKACO	DMKACR	DMKACS	DMKBIO	DMKBSC	DMKCAC	DMKCCD	DMKCCF	DMKCCH	DMKCCO
		DMKCCS	DMKCCW	DMKCFQ	DMKCFR	DMKCKD	DMKCNS	DMKCNT	DMKCPB	DMKCPM	DMKCPN
		DMKCPP	DMKCPS	DMKCPT	DMKCPW	DMKCPZ	DMKCQT	DMKCSB	DMKCSC	DMKCSF	DMKCSO
		DMKCSR	DMKCSW	DMKCSX	DMKDAD	DMKDAS	DMKDAU	DMKDEX	DMKDGD	DMKDGF	DMKDIA
		DMKDIB	DMKDID	DMKDIF	DMKDSB	DMKDSP	DMKGIO	DMKGRC	DMKGRD	DMKGRE	DMKGRF
		DMKGRG	DMKGRH	DMKGRI	DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKIOE	DMKIOH	DMKIOQ
		DMKIOS	DMKIOT	DMKISM	DMKLOH	DMKMCC	DMKMNI	DMKMNL	DMKMON	DMKMSW	DMKNLD
		DMKNLE	DMKOPE	DMKPAG	DMKPAH	DMKQVM	DMKRGA	DMKRGB	DMKRGC	DMKRGD	DMKRGE
		DMKRNH	DMKRSE	DMKRSF	DMKRSP	DMKRSQ	DMKRST	DMKSEP	DMKSPK	DMKSPL	DMKSPM
		DMKSPS	DMKSPT	DMKSSU	DMKSTK	DMKTAP	DMKTAQ	DMKTCS	DMKTCT	DMKTPE	DMKTRC
		DMKTRD	DMKTRK	DMKTTX	DMKTTY	DMKUDR	DMKUNT	DMKUSP	DMKVCA	DMKVCB	DMKVCN
		DMKVCP	DMKVDA	DMKVDC	DMKVDD	DMKVDE	DMKVDG	DMKVDR	DMKVDT	DMKVIO	DMKVRR
		DMKVRS	DMKVSC	DMKVSI	DMKVSJ	DMKWAI	DMKZTD				
IOBMID	000006	DMKCFR	DMKDID	DMKIOS							
IOBMINI	000024	DMKACS	DMKDID	DMKIOQ	DMKIOS	DMKPAG					
IOBMISC	000196	DMKCCW	DMKCFQ	DMKCFR	DMKCNS	DMKCPB	DMKCPN	DMKCPS	DMKCPT	DMKCPW	DMKCPZ
		DMKCSB	DMKCSF	DMKCSO	DMKDAD	DMKDAS	DMKDGD	DMKDGF	DMKDIA	DMKDSP	DMKGIO
		DMKGRF	DMKGRG	DMKGRH	DMKGRI	DMKHVC	DMKIOT	DMKISM	DMKMCC	DMKMNI	DMKMNL
		DMKMON	DMKNLD	DMKNLE	DMKPAG	DMKPAH	DMKRGA	DMKRGB	DMKRNH	DMKRSF	DMKRSP
		DMKRST	DMKSEP	DMKSPS	DMKSPT	DMKTAP	DMKTCS	DMKTCT	DMKUDR	DMKUNT	DMKUSP
		DMKVDA	DMKVDE	DMKVDG	DMKVIO	DMKVSI	DMKZTD				
IOBMISC2	000185	DMKBIO	DMKBSC	DMKCCW	DMKCFQ	DMKCFR	DMKCNS	DMKCPS	DMKCPW	DMKCQT	DMKCSB
		DMKDGD	DMKDID	DMKGIO	DMKGRD	DMKGRF	DMKGRG	DMKHVC	DMKMNI	DMKMON	DMKNLD
		DMKNLE	DMKPAG	DMKPAH	DMKRGA	DMKRGB	DMKRGC	DMKRNH	DMKRSE	DMKRSP	DMKSEP
		DMKSPK	DMKTCS	DMKTCT	DMKTRK	DMKUDR	DMKVDC	DMKVDE	DMKVIO		
IOBMQDTO	000008	DMKACS	DMKIOQ								
IOBMSG	000004	DMKCFR									
IOBMSIZE	000002	DMKMNT									
IOBOERR	000003	DMKACS	DMKCCH								
IOBOFF	000006	DMKIOQ	DMKIOS	DMKPAG							
IOBPAG	000012	DMKDAD	DMKDAS	DMKDSP	DMKIOQ	DMKIOS	DMKIOT	DMKMON	DMKPAG		
IOBPATH	000004	DMKIOS									
IOBPATHF	000021	DMKACS	DMKCPN	DMKCPW	DMKIOQ	DMKIOS	DMKMNL	DMKTPE	DMKVDT	DMKVSJ	
IOBPGID	000004	DMKCPN	DMKCPW								
IOBPMINT	000003	DMKIOT	DMKVRR	DMKVRS							
IOBPROC	000019	DMKCPN	DMKCPW	DMKCPZ	DMKDID	DMKIOQ	DMKIOS	DMKMNL	DMKTPE	DMKVSJ	
IOBPST	000003	DMKIOS	DMKVIO								
IOBPVM	000004	DMKIOS	DMKIOT	DMKVSI							
IOBQDTO	000015	DMKACS	DMKIOQ	DMKIOS	DMKIOT						
IOBQSTAT	000006	DMKCQT	DMKIOE	DMKIOS	DMKIOT						
IOBRADD	000145	DMKACR	DMKACS	DMKCAC	DMKCCH	DMKCKD	DMKCNS	DMKCPM	DMKCPN	DMKCPS	DMKCPW
		DMKCPZ	DMKCSB	DMKCSF	DMKCSO	DMKDAD	DMKDAS	DMKDAU	DMKDIA	DMKDID	DMKDSB
		DMKDSP	DMKGRF	DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKIOE	DMKIOH	DMKIOQ	DMKIOS
		DMKIOT	DMKMNL	DMKMSW	DMKNLD	DMKNLE	DMKPAG	DMKPAH	DMKQVM	DMKRGA	DMKRNH
		DMKRSE	DMKRSP	DMKRST	DMKSSU	DMKTAP	DMKTRC	DMKTRD	DMKVDD	DMKVDT	DMKVRR
		DMKVRS	DMKVSJ								
IOBRCAW	000141	DMKBSC	DMKCCH	DMKDAD	DMKDAS	DMKDAU	DMKDIB	DMKDIF	DMKIOS	DMKIOT	DMKNLD
		DMKNLE	DMKRNH	DMKRSE	DMKRSF	DMKRSP	DMKTAP	DMKTAQ	DMKTPE	DMKTRK	DMKVSI
		DMKVSJ									
IOBRCNT	000158	DMKBSC	DMKDAD	DMKDAS	DMKDAU	DMKDID	DMKGRF	DMKHPS	DMKHPT	DMKIOT	DMKNLD
		DMKNLE	DMKRNH	DMKRSE	DMKRSP	DMKRST	DMKTAP	DMKTAQ	DMKTPE	DMKTRK	

LABEL	COUNT	REFERENCES									
IOBREL	000006	DMKDAD	DMKDAS	DMKDSB							
IOBRELCU	000016	DMKCCO	DMKIOQ	DMKIOS	DMKVDR	DMKVIO	DMKVSC	DMKVSI			
IOBREMOT	000009	DMKRGC	DMKRGE	DMKVCN	DMKVCP	DMKVIO					
IOBRES	000024	DMKCFR	DMKCNS	DMKCSB	DMKDID	DMKIOS	DMKIOT	DMKUNT	DMKVCA	DMKVIO	
IOBRESGN	000002	DMKCPW									
IOBRESRV	000002	DMKCCS	DMKVIO								
IOBRETRY	000008	DMKACS	DMKIOS	DMKIOT							
IOBRREL	000002	DMKCFR	DMKIOS								
IOBRSTRT	000118	DMKBSC	DMKCCH	DMKCSB	DMKDAD	DMKDAS	DMKDAU	DMKDIB	DMKIOQ	DMKIOS	DMKIOT
		DMKNLD	DMKNLE	DMKRGB	DMKRNH	DMKRSE	DMKRSF	DMKRSP	DMKRST	DMKSEP	DMKSSU
		DMKTAP	DMKTAQ	DMKTCS	DMKTCT	DMKTPE	DMKTRK	DMKVCA			
IOBRTCT	000002	DMKIOS									
IOBRTCT2	000005	DMKIOS									
IOBRUNLD	000001	DMKCPS									
IOBRUN1	000003	DMKCQT									
IOBSENS	000005	DMKGRF	DMKGRG								
IOBSENSE	000006	DMKCCS	DMKCCW	DMKDEX	DMKVIO	DMKVSI					
IOBSETP	000003	DMKCPN									
IOBSIMS	000003	DMKIOS									
IOBSIOEX	000004	DMKIOS	DMKIOT								
IOBSIOF	000008	DMKIOQ	DMKIOS	DMKIOT	DMKTRK	DMKVIO	DMKVSI				
IOBSIZE	000339	DMKACO	DMKACS	DMKBIO	DMKCAC	DMKCCS	DMKCFQ	DMKCFR	DMKCNS	DMKCPB	DMKCPM
		DMKCPN	DMKCPO	DMKCPS	DMKCPT	DMKCPW	DMKCPZ	DMKCQT	DMKCSB	DMKCSF	DMKCSO
		DMKCSR	DMKCSW	DMKCSX	DMKDAD	DMKDAS	DMKDAU	DMKDGD	DMKDGF	DMKDIA	DMKDID
		DMKDIF	DMKDMQ	DMKDSB	DMKDSP	DMKGIO	DMKGRD	DMKGRF	DMKGRI	DMKHPS	DMKHPT
		DMKHPU	DMKHVC	DMKIOE	DMKIOH	DMKIOQ	DMKIOS	DMKIOT	DMKMCC	DMKMNI	DMKMNL
		DMKMON	DMKNLD	DMKNLE	DMKPAG	DMKPAH	DMKQVM	DMKRGA	DMKRGB	DMKRGC	DMKRGE
		DMKRNH	DMKRSE	DMKRSP	DMKRST	DMKSEP	DMKSPL	DMKSPM	DMKSPS	DMKSSU	DMKTAP
		DMKTCS	DMKTCT	DMKTRK	DMKUDR	DMKUNT	DMKVCA	DMKVCB	DMKVCN	DMKVCP	DMKVDA
		DMKVDC	DMKVDD	DMKVDE	DMKVDG	DMKVDR	DMKVDT	DMKVIO	DMKVRR	DMKVSC	DMKVSI
		DMKVSJ	DMKZTD								
IOBSNSE	000003	DMKCPS									
IOBSNSIO	000045	DMKIOQ	DMKIOS	DMKIOT							
IOBSPEC	000165	DMKACS	DMKBSC	DMKCCH	DMKCFR	DMKCNS	DMKCPM	DMKCPS	DMKCPW	DMKDAD	DMKDAS
		DMKDAU	DMKDID	DMKDSB	DMKGRD	DMKGRF	DMKGRG	DMKHPS	DMKHPT	DMKHPU	DMKIOQ
		DMKIOS	DMKIOT	DMKNLD	DMKNLE	DMKPAG	DMKRGA	DMKRGB	DMKRGC	DMKRGE	DMKRNH
		DMKRSP	DMKSPL	DMKSPM	DMKSSU	DMKTAP	DMKTRK	DMKVCA	DMKVCP	DMKVDC	DMKVDD
		DMKVDE	DMKVIO	DMKVSI	DMKVSJ						
IOBSPEC2	000080	DMKACS	DMKCCS	DMKCCW	DMKDAD	DMKDAS	DMKDSB	DMKGIO	DMKIOQ	DMKIOS	DMKIOT
		DMKMON	DMKPAG	DMKUNT	DMKVIO	DMKVSI					
IOBSPEC3	000038	DMKACS	DMKCCH	DMKCCS	DMKCCW	DMKCFR	DMKDEX	DMKIOQ	DMKIOS	DMKIOT	DMKSPM
		DMKVIO	DMKVSI	DMKVSJ							
IOBSPEC4	000034	DMKCCW	DMKCFR	DMKDID	DMKIOQ	DMKIOS	DMKIOT	DMKVIO	DMKVRR	DMKVRS	DMKVSI
		DMKVSJ									
IOBSPEC5	000035	DMKCCS	DMKCFR	DMKCPN	DMKCPW	DMKCQT	DMKDID	DMKGRF	DMKIOE	DMKIOS	DMKIOT
		DMKRGC	DMKRGE	DMKTPE	DMKVCN	DMKVCP	DMKVIO				
IOBSPLT	000011	DMKIOS	DMKIOT								
IOBSPM	000003	DMKIOQ	DMKIOS	DMKVSI							
IOBSTAT	000339	DMKACS	DMKBIO	DMKBSC	DMKCAC	DMKCCD	DMKCCH	DMKCFQ	DMKCNS	DMKCPN	DMKCPS
		DMKCPW	DMKCPZ	DMKCQT	DMKCSB	DMKCSO	DMKDAD	DMKDAS	DMKDAU	DMKDGF	DMKDIB
		DMKDID	DMKDSB	DMKGIO	DMKGRF	DMKHPS	DMKIOE	DMKIOH	DMKIOQ	DMKIOS	DMKIOT
		DMKMNI	DMKMNL	DMKMON	DMKNLD	DMKNLE	DMKPAG	DMKPAH	DMKRGA	DMKRGB	DMKRNH
		DMKRSE	DMKRSF	DMKRSP	DMKRST	DMKSEP	DMKSPK	DMKSPS	DMKSSU	DMKTAP	DMKTAQ
		DMKTCS	DMKTCT	DMKTPE	DMKTRD	DMKTRK	DMKUDR	DMKUNT	DMKVCA	DMKVDC	DMKVDE
		DMKVDT	DMKVIO	DMKVSI	DMKVSJ	DMKZTD					
IOBTIO	000050	DMKACS	DMKCCH	DMKCFR	DMKCPW	DMKDSB	DMKIOS	DMKIOT	DMKNLD	DMKNLE	DMKTAP
		DMKVDC	DMKVDE	DMKVIO	DMKVSI						
IOBUASN	000003	DMKCPW									

LABEL	COUNT	REFERENCES									
IOBUC	000010	DMKIOS	DMKIOT								
IOBUNREL	000004	DMKCCS	DMKCCW	DMKUNT	DMKVIO						
IOBUNSL	000067	DMKACS	DMKBSC	DMKCCH	DMKCFR	DMKCNS	DMKCPM	DMKCPW	DMKDAD	DMKDAS	DMKDAU
		DMKDID	DMKGRD	DMKGRF	DMKHPT	DMKHPU	DMKIOQ	DMKIOS	DMKIOT	DMKNLD	DMKNLE
		DMKRGA	DMKRGB	DMKRGC	DMKRGE	DMKRNH	DMKRSP	DMKSPL	DMKSPM	DMKSSU	DMKTAP
		DMKVCA	DMKVCP	DMKVDD	DMKVIO						
IOBUSER	000145	DMKACO	DMKACR	DMKACS	DMKBIO	DMKCAC	DMKCCH	DMKCFQ	DMKCFR	DMKCNS	DMKCPB
		DMKCPM	DMKCPN	DMKCPS	DMKCPW	DMKCPZ	DMKCQT	DMKCSB	DMKCSF	DMKCSO	DMKCSR
		DMKCSW	DMKCSX	DMKDIA	DMKDIB	DMKDID	DMKDIF	DMKDSB	DMKDSP	DMKGRD	DMKGRF
		DMKGRI	DMKHPT	DMKHPU	DMKIOH	DMKIOQ	DMKIOS	DMKIOT	DMKLOH	DMKMCC	DMKMNL
		DMKNLD	DMKNLE	DMKOPE	DMKPAG	DMKQVM	DMKRGA	DMKRGB	DMKRGC	DMKRGE	DMKRNH
		DMKRSF	DMKSPK	DMKSPL	DMKSPT	DMKSSU	DMKSTK	DMKTAP	DMKTRK	DMKUDR	DMKUNT
		DMKVCA	DMKVCB	DMKVCN	DMKVCP	DMKVDC	DMKVDD	DMKVDE	DMKVDR	DMKVDT	DMKVIO
		DMKVSI	DMKVSJ	DMKWAI	DMKZTD						
IOBVADD	000055	DMKACO	DMKCFQ	DMKCFR	DMKCPS	DMKCSR	DMKCSW	DMKCSX	DMKDIF	DMKHPS	DMKHPT
		DMKHPU	DMKIOQ	DMKIOS	DMKIOT	DMKRGA	DMKRGB	DMKRGC	DMKRGE	DMKSPL	DMKSSU
		DMKTRC	DMKUNT	DMKVCA	DMKVCB	DMKVCN	DMKVCP	DMKVIO	DMKVSI	DMKVSJ	
IOBVCUE	000011	DMKIOQ	DMKIOS	DMKIOT	DMKVSI	DMKVSJ					
IOBVDVIO	000002	DMKVIO									
IOBVHIO	000005	DMKIOS	DMKVIO	DMKVSJ							
IOBWRAP	000003	DMKCCW	DMKVIO								
IOBXTRA	000012	DMKCPN	DMKCPS	DMKCPT	DMKCPW						
IOELPNTR	000007	DMKCCH	DMKCPJ	DMKEIG	DMKIOG						
IOERACT	000011	DMKBSC	DMKDAD	DMKDAS	DMKDAU	DMKMSW	DMKRSE	DMKTAP	DMKTPE		
IOERADR	000035	DMKDAD	DMKDAS	DMKDAU	DMKIOJ	DMKMSW	DMKTAQ	DMKTRK			
IOERALTR	000004	DMKDAS	DMKTRK								
IOERBLOK	000452	DMKACS	DMKBSC	DMKCCH	DMKCCS	DMKCCW	DMKCFQ	DMKCFR	DMKCNS	DMKCPN	DMKCPS
		DMKCPW	DMKCPZ	DMKCQT	DMKDAD	DMKDAS	DMKDAU	DMKDGD	DMKDGF	DMKDIA	DMKDIB
		DMKDID	DMKDSB	DMKDSP	DMKEIG	DMKGIO	DMKGRF	DMKGRI	DMKHPS	DMKHPT	DMKHPU
		DMKIOE	DMKIOF	DMKIOJ	DMKIOS	DMKIOT	DMKMNL	DMKMSW	DMKNLD	DMKNLE	DMKQVM
		DMKRGA	DMKRGB	DMKRGE	DMKRNH	DMKRSE	DMKRSF	DMKRSP	DMKSEV	DMKSIX	DMKTAP
		DMKTAQ	DMKTPE	DMKTRK	DMKUNT	DMKVCA	DMKVCB	DMKVCN	DMKVDC	DMKVDE	DMKVIO
		DMKVRR	DMKVSC	DMKVSI							
IOERBSR	000025	DMKTAP	DMKTAQ	DMKTPE							
IOERB80	000002	DMKCCH									
IOERCAL	000005	DMKDAS									
IOERCAN	000050	DMKBSC	DMKDAD	DMKDAS	DMKDAU	DMKTAP	DMKTAQ	DMKTPE			
IOERCCH	000006	DMKACS	DMKCCH	DMKDID							
IOERCCRA	000020	DMKBSC	DMKCCH	DMKCNS	DMKDAD	DMKDAS	DMKDAU	DMKDSB	DMKIOE	DMKIOS	DMKIOT
		DMKRSE	DMKRSF	DMKRSP	DMKTAP	DMKTAQ	DMKTPE				
IOERCCRL	000016	DMKBSC	DMKCCH	DMKDAD	DMKDAS	DMKDAU	DMKDSB	DMKIOE	DMKRSE	DMKRSF	DMKRSP
		DMKTAP	DMKTAQ	DMKTPE							
IOERCCUA	000003	DMKACS	DMKCCH	DMKDID							
IOERCCW	000016	DMKCNS	DMKDIB	DMKIOS	DMKVCA	DMKVCB					
IOERCEMD	000029	DMKDAD	DMKDAS	DMKDAU	DMKIOE	DMKRSE	DMKTAP	DMKTRK			
IOERCHID	000012	DMKACS	DMKCCH	DMKDID							
IOERCLN	000006	DMKTAP	DMKTAQ								
IOERCLOG	000002	DMKACS	DMKCCH								
IOERCNCL	000002	DMKMSW									
IOERCPEX	000022	DMKDAD	DMKDAS	DMKDAU	DMKTAP	DMKTAQ	DMKTPE				
IOERCPID	000012	DMKACS	DMKCCH	DMKDID	DMKDSB	DMKIOE	DMKIOF	DMKIOS	DMKRSF		
IOERCSW	000173	DMKACS	DMKBSC	DMKCCH	DMKCQT	DMKDAD	DMKDAS	DMKDAU	DMKDIB	DMKDID	DMKDSB
		DMKGIO	DMKGRF	DMKHPS	DMKHPT	DMKIOE	DMKIOF	DMKIOJ	DMKIOS	DMKIOT	DMKMSW
		DMKRGE	DMKRSE	DMKRSF	DMKRSP	DMKTAP	DMKTAQ	DMKTPE	DMKTRK	DMKVCA	DMKVCB
		DMKVCN	DMKVIO	DMKVSI							
IOERCYLR	000002	DMKUNT									
IOERDASD	000012	DMKDAD	DMKDAS	DMKDAU	DMKMSW						
IOERDATA	000478	DMKBSC	DMKCCS	DMKCNS	DMKCPW	DMKCQT	DMKDAD	DMKDAS	DMKDAU	DMKDGF	DMKDIB

LABEL	COUNT	REFERENCES									
		DMKDID	DMKDSB	DMKDSP	DMKGIO	DMKGRF	DMKHPS	DMKHPT	DMKIOE	DMKIOF	DMKIOJ
		DMKIOS	DMKIOT	DMKMSW	DMKNLD	DMKNLE	DMKRGE	DMKRNH	DMKRSE	DMKRSF	DMKRSP
		DMKTAP	DMKTAQ	DMKTPE	DMKTRK	DMKUNT	DMKVCA	DMKVCB	DMKVCN	DMKVDC	DMKVDE
		DMKVIO	DMKVRR	DMKVSI							
IOERDEC	000006	DMKDAD	DMKDAS	DMKDAU	DMKMSW						
IOERDEF	000006	DMKDAD									
IOERDEPD	000004	DMKRSE	DMKRSP								
IOERDERD	000005	DMKRSE	DMKRSP								
IOERDW	000039	DMKBSC	DMKDAD	DMKDAS	DMKDAU	DMKDSB	DMKTAP	DMKTAQ	DMKTPE	DMKTRK	
IOERECF	000037	DMKDAD	DMKDAS	DMKDAU							
IOERECSW	000009	DMKACS	DMKCCH	DMKDID	DMKRSE	DMKVIO					
IOERERG	000012	DMKTAP	DMKTAQ	DMKTPE							
IOERERP	000004	DMKRSE	DMKRSP								
IOERETN	000122	DMKAPT	DMKCFG	DMKCFH	DMKCFS	DMKCFU	DMKCPI	DMKCPS	DMKCPU	DMKCSC	DMKCSO
		DMKDRD	DMKDRE	DMKERM	DMKGRC	DMKHPS	DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKIDU
		DMKMCC	DMKMIA	DMKNLD	DMKNLE	DMKPTR	DMKRPA	DMKRSP	DMKRST	DMKSEG	DMKSEP
		DMKSNC	DMKSPK	DMKSSS	DMKSST	DMKSVC	DMKTCS	DMKUDR	DMKUDU	DMKVMD	DMKVME
		DMKVSE	DMKVSG	DMKVSP	DMKVSQ	DMKVSX	DMKXAB				
IOERETRY	000006	DMKDAU	DMKMSW	DMKRSE							
IOEREXT	000081	DMKACS	DMKBSC	DMKCCH	DMKCCW	DMKCFQ	DMKCFR	DMKCNS	DMKCPN	DMKCPS	DMKCPW
		DMKCQT	DMKDAD	DMKDAS	DMKDAU	DMKDGD	DMKDGF	DMKDIA	DMKDID	DMKDSB	DMKGIO
		DMKGRI	DMKHPT	DMKIOE	DMKIOJ	DMKIOS	DMKIOT	DMKMNL	DMKNLD	DMKNLE	DMKQVM
		DMKRGA	DMKRGB	DMKRNH	DMKRSE	DMKRSF	DMKRSP	DMKTAP	DMKTAQ	DMKTPE	DMKTRK
		DMKVDC	DMKVDE	DMKVIO	DMKVSC						
IOERFIXP	000001	DMKTPE									
IOERFLG1	000096	DMKDAD	DMKDAS	DMKDAU	DMKMSW	DMKRSE	DMKRSF	DMKRSP	DMKTAP	DMKTAQ	DMKTPE
IOERFLG2	000156	DMKBSC	DMKDAD	DMKDAS	DMKDAU	DMKIOE	DMKRSE	DMKTAP	DMKTAQ	DMKTPE	DMKTRK
		DMKUNT									
IOERFLG3	000045	DMKBSC	DMKCNS	DMKDAD	DMKDAS	DMKDAU	DMKGRF	DMKIOE	DMKIOF	DMKIOS	DMKRSE
		DMKTAP	DMKTAQ	DMKTPE	DMKTRK						
IOERFLG4	000003	DMKDAS	DMKIOE								
IOERFLM	000005	DMKIOE	DMKTAP								
IOERFSR	000019	DMKTAP	DMKTAQ	DMKTPE							
IOERHA	000008	DMKDAD	DMKDAS								
IOERIGN	000004	DMKMSW	DMKRSE								
IOERIGNR	000008	DMKDAD	DMKDAS	DMKDAU	DMKMSW	DMKTAP	DMKTAQ				
IOERIND3	000091	DMKBSC	DMKDAD	DMKDAS	DMKDAU	DMKMSW	DMKRSE	DMKTAP	DMKTAQ	DMKTPE	
IOERIND4	000021	DMKDAD	DMKDAS	DMKDAU	DMKMSW	DMKTAP	DMKTAQ				
IOERINFO	000054	DMKBSC	DMKDAD	DMKDAS	DMKDAU	DMKMSW	DMKRSE	DMKTAP	DMKTAQ	DMKTPE	
IOERLEN	000026	DMKCCS	DMKCQT	DMKDID	DMKDSB	DMKHPS	DMKHPT	DMKIOE	DMKIOJ	DMKIOS	DMKIOT
		DMKMSW	DMKRGE	DMKRSF	DMKUNT	DMKVCA	DMKVCB	DMKVCN	DMKVSI		
IOERLG45	000001	DMKCCH									
IOERLOC	000087	DMKBSC	DMKDAD	DMKDAS	DMKDAU	DMKDSB	DMKTAP	DMKTAQ	DMKTPE	DMKTRK	
IOERLOGL	000006	DMKACS	DMKCCH	DMKDID							
IOERMSG	000005	DMKDAD	DMKDAS	DMKDAU	DMKTAP						
IOERMSW	000032	DMKBSC	DMKDAD	DMKDAS	DMKDAU	DMKTAP	DMKTAQ	DMKTPE			
IOERNOLG	000005	DMKIOS	DMKTPE								
IOERNUM	000126	DMKBSC	DMKCNS	DMKDAD	DMKDAS	DMKDAU	DMKGRF	DMKMSW	DMKRSE	DMKTAP	DMKTAQ
		DMKTPE									
IOERORA	000014	DMKTAP	DMKTAQ								
IOERPEND	000027	DMKDAD	DMKDAS	DMKDAU	DMKMSW	DMKRSE	DMKTAP	DMKTAQ	DMKTPE		
IOERPNT	000035	DMKCFQ	DMKCQT	DMKDAD	DMKDAS	DMKDAU	DMKDGF	DMKDID	DMKGIO	DMKIOE	DMKRSE
		DMKRSF	DMKTAP	DMKTRK							
IOERQUE	000003	DMKDAS	DMKIOE								
IOERRBK	000012	DMKTAP	DMKTAQ								
IOERRDRO	000005	DMKDAS	DMKTRK								
IOERREAD	000020	DMKBSC	DMKCNS	DMKDAD	DMKDAS	DMKDAU	DMKGRF	DMKIOF	DMKRSE	DMKTAP	DMKTAQ
IOERREW	000003	DMKTAP	DMKTAQ								

LABEL	COUNT	REFERENCES									
IOERSIZE	000142	DMKACS	DMKBIO	DMKBSC	DMKCCH	DMKCCW	DMKCFQ	DMKCFR	DMKCNS	DMKCPN	DMKCPS
		DMKCPW	DMKCPZ	DMKCQT	DMKDAD	DMKDAS	DMKDAU	DMKDGD	DMKDGF	DMKDIA	DMKDIB
		DMKDID	DMKDSB	DMKGIO	DMKGRF	DMKGRI	DMKHPS	DMKHPT	DMKIOE	DMKIOJ	DMKIOS
		DMKIOT	DMKMNI	DMKMNL	DMKMON	DMKNLD	DMKNLE	DMKQVM	DMKRGA	DMKRGB	DMKRGE
		DMKRNH	DMKRSE	DMKRSF	DMKRSP	DMKSPS	DMKTAP	DMKTAQ	DMKTPE	DMKTRK	DMKVCA
		DMKVCB	DMKVCN	DMKVDC	DMKVDE	DMKVIO	DMKVRR	DMKVSC	DMKVSI		
IOERSNSZ	000019	DMKDAD	DMKDAS	DMKDAU	DMKDSB	DMKIOS	DMKVDE				
IOERSPID	000004	DMKTPE									
IOERSTAT	000035	DMKDAD	DMKDAS	DMKDAU	DMKTAP						
IOERSTRT	000008	DMKDAD	DMKDAS	DMKDAU	DMKMSW	DMKTAP	DMKTAQ				
IOERSUPP	000006	DMKTAQ									
IOERS80	000002	DMKCCH									
IOERTEMP	000003	DMKIOE	DMKTPE								
IOERVLD	000003	DMKTAQ									
IOERVOL1	000009	DMKDAD	DMKDAS	DMKDAU							
IOERVSER	000022	DMKDAD	DMKDAS	DMKDAU	DMKDSB	DMKIOE	DMKIOJ	DMKTPE			
IOERWRK	000014	DMKTAQ	DMKTRK								
IOERXERP	000003	DMKRSE	DMKRSF								
IOER2860	000003	DMKCCH	DMKDID								
IOER2870	000002	DMKCCH	DMKDID								
IOKEEP	000072	DMKAPT	DMKCFG	DMKCFH	DMKCFS	DMKCFU	DMKCKV	DMKCPY	DMKCQY	DMKCSC	DMKCSO
		DMKDRD	DMKDRE	DMKERM	DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKPTR	DMKRSP	DMKSEP
		DMKSPT	DMKSRM	DMKTCS	DMKTRP	DMKTRU	DMKUDU	DMKVBM	DMKVMD	DMKVME	DMKWRM
IOLOG	000001	DMKCPJ									
IOMASK	000018	DMKAPI	DMKCFM	DMKDMP	DMKDMQ	DMKDSP	DMKEXT	DMKLD00E	DMKSAD	DMKSVD	DMKTRC
		DMKVRR	DMKVRS								
IONPSW	000036	DMKCKH	DMKCKN	DMKCKP	DMKDMP	DMKDMQ	DMKDSP	DMKFMT	DMKMNT	DMKOPE	DMKOPR
		DMKSAD	DMKSAV	DMKSSP	DMKVRS						
IONTWAIT	000011	DMKAPI	DMKCPI	DMKDSP	DMKMON	DMKSTP					
IOOPSW	000049	DMKCCH	DMKCKH	DMKDSP	DMKFMT	DMKIOT	DMKMNT	DMKSAD	DMKSAV	DMKSSP	DMKVRS
IPLADDR	000002	DMKVMI									
IPLCCW1	000013	DMKCPI	DMKDMP	DMKDSP	DMKMPO	DMKVMI					
IPLPSW	000035	DMKCKP	DMKCPI	DMKDMP	DMKLD00E	DMKMIA	DMKMON	DMKMOO	DMKSAD	DMKVMI	
IPLREQ	000005	DMKNLD	DMKNLE	DMKRNH							
IPUADDR	000070	DMKAPI	DMKCCH	DMKCFO	DMKCFP	DMKCFY	DMKCKD	DMKCPI	DMKCPO	DMKCPP	DMKCPS
		DMKCPU	DMKCQP	DMKCQQ	DMKCQY	DMKDMP	DMKDSP	DMKFPS	DMKHVD	DMKIOG	DMKIOS
		DMKMCH	DMKMCI	DMKMCT	DMKMNT	DMKMPO	DMKPMA	DMKPRV	DMKTHI	DMKURS	DMKVDC
		DMKVDT	DMKVFC								
IPUADDRX	000083	DMKACO	DMKACS	DMKAPI	DMKCCH	DMKCFO	DMKCKD	DMKCLK	DMKCPI	DMKCPO	DMKCPP
		DMKCPS	DMKCPU	DMKCQP	DMKCQQ	DMKCQY	DMKDMP	DMKDSP	DMKEXT	DMKFRE	DMKIOS
		DMKMCH	DMKMCI	DMKMCT	DMKMNT	DMKMOO	DMKPTR	DMKPTT	DMKSCH	DMKSEL	DMKSTK
		DMKTHI	DMKURS	DMKVDC	DMKVFC						
IREALVF	000002	DMKVFC									
IRMAND	000003	DMKCFU	DMKIOE								
IRMBIT1	000002	DMKCFU	DMKIOE								
IRMBIT2	000002	DMKCFU	DMKIOE								
IRMBLOK	000004	DMKCFU	DMKIOE								
IRMBYT1	000002	DMKCFU	DMKIOE								
IRMBYT2	000002	DMKCFU	DMKIOE								
IRMFLG	000006	DMKCFU	DMKIOE								
IRMLMT	000003	DMKCFU	DMKIOE								
IRMLMTCT	000003	DMKIOE									
IRMMAXCT	000005	DMKIOE									
IRMOR	000003	DMKCFU	DMKIOE								
IRMRLADD	000003	DMKCFU	DMKIOE								
IRMSIZE	000005	DMKCFU	DMKIOE								
IUCBFAD1	000012	DMKIUA	DMKIUB	DMKIUC	DMKIUP						
IUCBFAD2	000002	DMKIUP									

LABEL	COUNT	REFERENCES									
IUCBFLN1	000002	DMKIUP									
IUCBFLN2	000002	DMKIUP									
IUCCBFA1	000007	DMKIUB	DMKIUC	DMKIUP							
IUCDWRD	000002	DMKIUP									
IUCLINK1	000006	DMKIUA									
IUCMSG	000005	DMKIUN	DMKIUP	DMKIUS							
IUCMXCN	000005	DMKIUC	DMKIUG	DMKIUP							
IUCPNDHD	000012	DMKIUB	DMKIUC	DMKIUJ	DMKIUP						
IUCPNDTL	000013	DMKIUB	DMKIUC	DMKIUJ	DMKIUP						
IUCSIZE	000002	DMKIUP									
IUCTOTCN	000015	DMKIUC	DMKIUJ								
IUCVBLOK	000071	DMKDSP	DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUP
		DMKIUS	DMKPGS								
IUCVCNT	000004	DMKCPP	DMKIUN								
IUCVCODE	000001	DMKDSP									
IUCVCPEX	000016	DMKIUA	DMKIUC	DMKIUJ	DMKIUP	DMKIUS	DMKPGS				
IUCVMASK	000001	DMKDSP									
IUCVMB	000013	DMKIUA	DMKIUC	DMKIUE	DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS		
IUSAVE	000065	DMKIUA	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS	
IUSBUFF	000002	DMKIUP									
IUSCC	000001	DMKIUN									
IUSCCODE	000008	DMKIUA	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUP		
IUSCPENT	000020	DMKIUA	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS	
IUSFCODE	000005	DMKIUA									
IUSFLAGS	000002	DMKIUA									
IUSFLAG2	000020	DMKIUA	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS	
IUSINSTR	000002	DMKIUA									
IUSIPSIZ	000001	DMKIUP									
IUSIUCV	000009	DMKIUA	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUP		
IUSIUCV2	000004	DMKIUA	DMKIUN	DMKIUP							
IUSLEN1	000003	DMKIUA									
IUSLEN2	000002	DMKIUA									
IUSMASK	000002	DMKIUN									
IUSMSGBK	000010	DMKIUE	DMKIUG	DMKIUL	DMKIUN	DMKIUS					
IUSMXCN	000001	DMKIUP									
IUSPAGE1	000006	DMKIUA	DMKIUC	DMKIUN							
IUSPAGE2	000002	DMKIUA									
IUSPARMS	000001	DMKIUA									
IUSPATH	000013	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS		
IUSRCODE	000004	DMKIUA									
IUSSIZE	000005	DMKIUA									
IUSVMBK	000002	DMKIUA									
IXBLOK	000052	DMKAPY	DMKCRM	DMKIDR	DMKIOF	DMKIUA	DMKIUC	DMKIUJ	DMKLOH	DMKMSG	DMKVCP
		DMKVCU	DMKVCV	DMKVCX							
IXEXBLOK	000028	DMKIDR	DMKIUA	DMKIUC	DMKIUJ	DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCV	
IXIRA	000014	DMKAPY	DMKCRM	DMKIDR	DMKIOF	DMKIUA	DMKIUC	DMKIUJ	DMKMSG	DMKVCV	DMKVCX
IXNEXT	000017	DMKIDR	DMKIUC	DMKIUJ							
IXREGS	000008	DMKAPY	DMKIDR	DMKIOF	DMKIUA	DMKMSG	DMKVCV	DMKVCX			
IXR0	000003	DMKCRM									
IXR1	000001	DMKCRM									
IXR11	000004	DMKCRM	DMKIDR	DMKIUA	DMKLOH						
IXR12	000004	DMKIDR	DMKVCV	DMKVCX							
IXR13	000001	DMKVCV									
IXSIZE	000026	DMKAPY	DMKCRM	DMKIDR	DMKIOF	DMKIUJ	DMKMSG	DMKVCP	DMKVCQ	DMKVCT	DMKVCU
		DMKVCV	DMKVCX								
JPSCBLOK	000015	DMKLNK	DMKLNM	DMKLOG							
JPSFLAGS	000019	DMKLNK	DMKLNM	DMKLOG							
JPSLNKDS	000002	DMKLNK	DMKLNM								

LABEL	COUNT	REFERENCES									
JPSLOGDS	000002	DMKLOG									
KEEPSEGS	000014	DMKBLD	DMKCFF	DMKCFP	DMKPGS						
KEYCHG	000009	DMKPTR	DMKPTS	DMKSEL							
KEYMASK	000001	DMKCPI									
KEYREF	000009	DMKPTS	DMKSEL								
LANGADDR	000006	DMKCPI	DMKERM	DMKHVF							
LANGBLOK	000014	DMKCPI	DMKCQT	DMKERM	DMKHVF	DMKLOH					
LANGDBCS	000002	DMKERM									
LANGFLAG	000003	DMKCPI	DMKERM								
LANGHIGH	000002	DMKCPI	DMKERM								
LANGLANG	000006	DMKCQT	DMKHVF	DMKLOH							
LANGLOCK	000003	DMKHVF									
LANGLOW	000002	DMKCPI	DMKERM								
LANGNEXT	000007	DMKCPI	DMKHVF	DMKLOH							
LANGNTRY	000004	DMKCPI	DMKERM	DMKHVF							
LANGNTSZ	000006	DMKCPI	DMKERM	DMKHVF							
LANGPAGE	000007	DMKCPI	DMKERM	DMKHVF							
LANGSIZE	000006	DMKCPI	DMKERM	DMKHVF							
LANGVMB	000002	DMKCPI	DMKERM								
LANGVMBK	000004	DMKCPI	DMKERM	DMKHVF							
LAP370E	000026	DMKDMP	DMKDSP	DMKFPS	DMKPRG	DMKSVD	DMKVAT				
LASTALOC	000001	DMKTDK									
LASTMSG	000001	DMKLDOOE									
LASTUSER	000054	DMKAPI	DMKATS	DMKBLD	DMKCDM	DMKCPI	DMKCPP	DMKDIB	DMKDIF	DMKDSP	DMKFRE
		DMKIOS	DMKIOT	DMKLOK	DMKPAG	DMKPGS	DMKSEL	DMKSWA	DMKUSQ		
LINE	000031	DMKDEF	DMKDMQ	DMKFCB	DMKSAD						
LINECT	000004	DMKDDR									
LINESUP	000005	DMKDMQ									
LINKJRL	000005	DMKLNK									
LINKJRLI	000006	DMKLNK	DMKLNM								
LISTSIZE	000001	DMKCKF									
LLTTY	000001	DMKBLD									
LL2741	000001	DMKBLD									
LL3066	000001	DMKBLD									
LMSG	000001	DMKSRM									
LOAD	000003	DMKCMD	DMKSPT								
LOADFLG1	000016	DMKCSB	DMKCSC								
LOADHCPY	000004	DMKCSB	DMKCSC								
LOADHEAD	000008	DMKCSB	DMKCSC								
LOADIMAG	000011	DMKCSB	DMKCSC								
LOADIML1	000003	DMKCSB	DMKCSC								
LOADIML2	000003	DMKCSB	DMKCSC								
LOADNAME	000004	DMKCSB	DMKCSC								
LOADPAG1	000010	DMKCSB	DMKCSC								
LOADPAG2	000013	DMKCSB	DMKCSC								
LOADPARM	000011	DMKCSB	DMKCSC								
LOADSIZE	000002	DMKCSB									
LOADSPAC	000002	DMKCSB	DMKCSC								
LOADSZDW	000003	DMKCSB									
LOADUCS	000007	DMKCSB	DMKCSC								
LOADVPAG	000005	DMKCSC									
LOAD2CCW	000005	DMKCSB	DMKCSC								
LOCCNT	000001	DMKIMG									
LOCGRAF	000002	DMKACO									
LOCK	000657	DMKACO	DMKACR	DMKACS	DMKALG	DMKALO	DMKAPI	DMKAPS	DMKAPT	DMKAPU	DMKAPV
		DMKAPW	DMKAPX	DMKAPY	DMKAPZ	DMKATS	DMKBIO	DMKBLD	DMKBSC	DMKCAC	DMKCAO
		DMKCCD	DMKCCF	DMKCCH	DMKCCO	DMKCCS	DMKCCT	DMKCCW	DMKCDB	DMKCDM	DMKCDS
		DMKCFC	DMKCFF	DMKCFG	DMKCFH	DMKCFJ	DMKCFM	DMKCFO	DMKCFP	DMKCFQ	DMKCFR

LABEL	COUNT	REFERENCES									
		DMKCFS	DMKCFT	DMKCFU	DMKCFV	DMKCFW	DMKCFY	DMKCKD	DMKCKF	DMKCKH	DMKCKM
		DMKCKN	DMKCKP	DMKCKR	DMKCKS	DMKCKT	DMKCKV	DMKCKW	DMKCLK	DMKCNS	DMKCNT
		DMKCPB	DMKCPI	DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPU
		DMKCPV	DMKCPW	DMKCPX	DMKCPY	DMKCPZ	DMKCQC	DMKCQS	DMKCQY	DMKCRM	DMKCSB
		DMKCSC	DMKCSF	DMKCSO	DMKCSP	DMKCSQ	DMKCSR	DMKCST	DMKCSU	DMKCSV	DMKCSW
		DMKCSX	DMKCSY	DMKCVT	DMKCVU	DMKDAD	DMKDAS	DMKDAU	DMKDEF	DMKDEG	DMKDEI
		DMKDEX	DMKDGD	DMKDGF	DMKDIA	DMKDIB	DMKDID	DMKDIF	DMKDMP	DMKDMQ	DMKDRD
		DMKDRE	DMKDSB	DMKDSP	DMKENT	DMKEPS	DMKERM	DMKERP	DMKEXT	DMKFPS	DMKFRE
		DMKFRT	DMKGIO	DMKGRA	DMKGRC	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRH	DMKGRI
		DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKIDR	DMKIDU	DMKIOE
		DMKIOF	DMKIOG	DMKIOH	DMKIOJ	DMKIOQ	DMKIOS	DMKIOT	DMKIUA	DMKIUB	DMKIUC
		DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS	DMKJRL	DMKLNK	DMKLNM
		DMKLOC	DMKLOG	DMKLOH	DMKLOJ	DMKLOK	DMKLOM	DMKMCC	DMKMCD	DMKMCH	DMKMCI
		DMKMCT	DMKMHC	DMKMHV	DMKMIA	DMKMID	DMKMNI	DMKMNJ	DMKMNL	DMKMNT	DMKMON
		DMKMOO	DMKMPO	DMKMSG	DMKMSW	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKOPE
		DMKOPR	DMKPAG	DMKPAH	DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPER	DMKPET	DMKPGM
		DMKPGS	DMKPGT	DMKPGU	DMKPIA	DMKPIB	DMKPMA	DMKPRG	DMKPRV	DMKPRW	DMKPSA
		DMKPST	DMKPTR	DMKPTS	DMKPTT	DMKQCN	DMKQCO	DMKQCP	DMKQCQ	DMKQVM	DMKREI
		DMKRGA	DMKRGB	DMKRGC	DMKRGD	DMKRGE	DMKRNH	DMKRPA	DMKRPD	DMKRPI	DMKRPW
		DMKRSE	DMKRSF	DMKRSP	DMKRSQ	DMKRST	DMKSAV	DMKSBL	DMKSCH	DMKSCN	DMKSCO
		DMKSEG	DMKSEL	DMKSEP	DMKSFB	DMKSNC	DMKSND	DMKSPK	DMKSPL	DMKSPM	DMKSPR
		DMKSPS	DMKSPT	DMKSRM	DMKSSS	DMKSST	DMKSSU	DMKSSV	DMKSTA	DMKSTK	DMKSTP
		DMKSTR	DMKSVC	DMKSVD	DMKSWA	DMKSWM	DMKSYM	DMKTAP	DMKTAQ	DMKTCS	DMKTCT
		DMKTDK	DMKTEE	DMKTHI	DMKTMR	DMKTOD	DMKTPE	DMKTRA	DMKTRC	DMKTRD	DMKTRK
		DMKTRM	DMKTRP	DMKTRQ	DMKTRT	DMKTRU	DMKTRX	DMKTTX	DMKTTY	DMKTTZ	DMKUCB
		DMKUCC	DMKUCS	DMKUDR	DMKUDU	DMKUNT	DMKURS	DMKUSP	DMKUSQ	DMKVAT	DMKVAU
		DMKVBM	DMKVCA	DMKVCB	DMKVCH	DMKVCN	DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCT
		DMKVCU	DMKVCV	DMKVCW	DMKVCX	DMKVDA	DMKVDB	DMKVDC	DMKVDD	DMKVDE	DMKVDF
		DMKVDG	DMKVDH	DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVFC	DMKVFD	DMKVFE	DMKVFR
		DMKVFS	DMKVIO	DMKVMA	DMKVMC	DMKVMD	DMKVME	DMKVMG	DMKVMI	DMKVRR	DMKVRS
		DMKVSC	DMKVSD	DMKVSE	DMKVSF	DMKVSG	DMKVSI	DMKVSJ	DMKVSP	DMKVSQ	DMKVST
		DMKVSU	DMKVSV	DMKVSW	DMKVSX	DMKWAI	DMKWRM	DMKWRN	DMKXAB	DMKXAD	DMKXST
		DMKZTD									
LOCKSAV	000286	DMKACR	DMKACS	DMKBLD	DMKCCS	DMKCDS	DMKCFQ	DMKCFR	DMKCNS	DMKDEI	DMKDGD
		DMKDGF	DMKDIA	DMKDIB	DMKDIF	DMKDSB	DMKDSP	DMKEXT	DMKFRE	DMKFRT	DMKGRI
		DMKIOQ	DMKIOS	DMKLOH	DMKLOJ	DMKMCH	DMKMNT	DMKMOO	DMKNLD	DMKPGM	DMKPRV
		DMKPTR	DMKPTS	DMKPTT	DMKRPA	DMKSCH	DMKSEL	DMKSSS	DMKSST	DMKSSU	DMKSSV
		DMKSTR	DMKSVC	DMKSWA	DMKSWM	DMKTRQ	DMKTRX	DMKUSP	DMKVAT	DMKVAU	DMKVCH
		DMKVDA	DMKVDD	DMKVDE	DMKVDR	DMKVDS	DMKVDT	DMKVFD	DMKVFE	DMKVFS	DMKVMA
		DMKVSJ									
LOCR	000003	DMKCCD	DMKCCO	DMKCCW							
LOCRCCHH	000006	DMKCCD	DMKCCO								
LOCSCCHH	000002	DMKCCD									
LOCSRR	000001	DMKCCD									
LOCWORK	000002	DMKCCW									
LOGDROP	000027	DMKCNS	DMKGRD	DMKGRF	DMKGRG	DMKQCN	DMKRGA	DMKRGB	DMKRNH	DMKUSQ	DMKVCS
LOGHOLD	000024	DMKCNS	DMKDIB	DMKGRD	DMKGRF	DMKGRG	DMKQCN	DMKRGA	DMKRGB	DMKRNH	DMKUSQ
		DMKVCS									
LOGMBLOK	000014	DMKCFU	DMKCQY	DMKLOM							
LOGMLEN	000005	DMKCFU	DMKCQY	DMKLOM							
LOGMNXT	000017	DMKCFU	DMKCQY	DMKLOM							
LOGMSEQ	000004	DMKCFU									
LOGMSIZB	000002	DMKCFU									
LOGMSIZD	000004	DMKCFU									
LOGMTXT	000005	DMKCFU	DMKCQY	DMKLOM							
LOGONJRL	000007	DMKLOG									
LOKREQ	000012	DMKDSP	DMKFRE	DMKSTK							
LOKSAVE	000021	DMKCKW	DMKLOK	DMKVAT							

LABEL	COUNT	REFERENCES									
LOKSAV2	000010	DMKLOK									
LOKSYS	000003	DMKPTT									
LOWCORE	000005	DMKDMQ									
LPRAN38D	000002	DMKAPU									
LPRBE38D	000002	DMKAPU									
LPRCHGE	000001	DMKAPU									
LPRCLASS	000002	DMKAPU									
LPRCNVD	000002	DMKAPU									
LPRCNVRT	000001	DMKAPU									
LPRCPXAD	000002	DMKAPS	DMKAPT								
LPRCTIOP	000005	DMKAPT									
LPRDEFLT	000002	DMKAPU									
LPRDEST1	000002	DMKAPU									
LPRFLASH	000004	DMKAPU									
LPRFLG1	000007	DMKAPU									
LPRFLG1D	000004	DMKAPU									
LPRFLG2	000010	DMKAPU									
LPRFLG3	000007	DMKAPS	DMKAPT	DMKAPU	DMKAPZ						
LPRFORM	000004	DMKAPU									
LPRIDLE	000003	DMKAPU	DMKAPZ								
LPRLSELD	000002	DMKAPU									
LPRLSPLC	000007	DMKAPS	DMKAPT	DMKAPU	DMKAPV	DMKAPW	DMKCKH				
LPRNAME	000003	DMKAPS	DMKAPY								
LPRNCNVD	000002	DMKAPU									
LPRNEXT	000011	DMKAPS	DMKAPU	DMKAPW	DMKAPY	DMKAPZ	DMKCKH				
LPRNO38D	000002	DMKAPU									
LPRN5AD	000002	DMKAPU									
LPRPATH	000006	DMKAPS	DMKAPW	DMKAPY	DMKAPZ						
LPRREADB	000001	DMKAPT									
LPRRECBF	000003	DMKAPV									
LPRSELDA	000001	DMKAPU									
LPRSESID	000001	DMKAPU									
LPRSEVER	000004	DMKAPS	DMKAPT								
LPRTBLOK	000018	DMKAPS	DMKAPT	DMKAPU	DMKAPV	DMKAPW	DMKAPY	DMKAPZ	DMKCKH		
LPRTSIZE	000003	DMKAPS									
LPRUSRID	000001	DMKAPS									
LPRXAB	000003	DMKAPT									
LPR5AD	000002	DMKAPU									
LPUADDR	000561	DMKACR	DMKACS	DMKAPI	DMKATS	DMKBLD	DMKCCH	DMKCCS	DMKCCW	DMKCDS	DMKCFF
		DMKCFG	DMKCFP	DMKCFQ	DMKCFR	DMKCFS	DMKCFY	DMKCKV	DMKCLK	DMKCNS	DMKCPI
		DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPU	DMKCPW	DMKCPZ
		DMKCQP	DMKCQQ	DMKCQU	DMKCSO	DMKDEI	DMKDGD	DMKDGF	DMKDIA	DMKDIB	DMKDID
		DMKDIF	DMKDSB	DMKDSP	DMKENT	DMKEXT	DMKFRE	DMKFRT	DMKGRI	DMKIOG	DMKIOJ
		DMKIOQ	DMKIOS	DMKIOT	DMKLOC	DMKLOH	DMKLOJ	DMKLOK	DMKMCC	DMKMCH	DMKMCT
		DMKMHC	DMKMNI	DMKMNT	DMKMOO	DMKMPO	DMKNES	DMKNLD	DMKOPE	DMKPAG	DMKPAH
		DMKPGM	DMKPMA	DMKPRG	DMKPRV	DMKPTR	DMKPTS	DMKPTT	DMKRGA	DMKRPA	DMKRSE
		DMKRSP	DMKSCH	DMKSEL	DMKSPM	DMKSSS	DMKSST	DMKSSU	DMKSSV	DMKSTK	DMKSTR
		DMKSVC	DMKSVD	DMKSWA	DMKSWM	DMKTMR	DMKTOD	DMKTRQ	DMKTRX	DMKURS	DMKUSP
		DMKVAT	DMKVAU	DMKVCH	DMKVDA	DMKVDD	DMKVDE	DMKVDR	DMKVDS	DMKVDT	DMKVFD
		DMKVFE	DMKVFR	DMKVFS	DMKVMA	DMKVMC	DMKVRR	DMKVSJ	DMKWAI		
LPUADDRX	000102	DMKACS	DMKAPI	DMKCFS	DMKCFY	DMKCLK	DMKCNS	DMKCPI	DMKCPJ	DMKCPO	DMKCPP
		DMKCPS	DMKCPU	DMKDSB	DMKDSP	DMKENT	DMKFRE	DMKIOG	DMKIOQ	DMKLOH	DMKMCT
		DMKMNI	DMKMNT	DMKMOO	DMKOPE	DMKPMA	DMKPTR	DMKSCH	DMKSTK	DMKSVC	DMKTOD
		DMKVCH	DMKVFR								
LSPCONVT	000002	DMKAPU	DMKAPV								
LSPID	000004	DMKAPU	DMKAPW								
LSPIOPND	000009	DMKAPS	DMKAPT	DMKAPV	DMKAPW						
LSPLCTL	000013	DMKAPS	DMKAPT	DMKAPU	DMKAPV	DMKAPW	DMKCKH				

LABEL	COUNT	REFERENCES									
LSPLFLG1	000019	DMKAPS	DMKAPT	DMKAPU	DMKAPV	DMKAPW	DMKCKH				
LSPLNEXT	000015	DMKAPS	DMKAPT	DMKAPU	DMKAPV	DMKAPW	DMKCKH				
LSPLPRTB	000001	DMKAPU									
LSPLSIZE	000005	DMKAPS	DMKAPU	DMKAPV	DMKAPW						
LSPMSK	000003	DMKFRE									
LSPPRINT	000004	DMKAPS	DMKAPU	DMKAPV	DMKCKH						
LSPPURGE	000004	DMKAPT	DMKAPW								
LSPSFBLK	000006	DMKAPS	DMKAPT	DMKAPU	DMKAPV	DMKCKH					
LSPSIZE	000006	DMKFRE	DMKFRT								
LSPSIZE$	000002	DMKFRE	DMKFRT								
LSPSPLNK	000007	DMKAPT	DMKAPU								
L1	000048	DMKDAD	DMKDAS	DMKDDR	DMKDGD	DMKDIR	DMKDMP	DMKDRD	DMKDRE	DMKFMT	DMKHVE
		DMKLNK	DMKPAG	DMKPMA	DMKRGE	DMKTDK	DMKTRK				
L10	000001	DMKDDR									
L1024	000001	DMKFMT									
L15	000001	DMKCPX									
L16	000047	DMKCPX	DMKDDR	DMKDGD	DMKDGF	DMKDIR	DMKDMP	DMKFMT	DMKPAG	DMKZTD	
L2	000059	DMKCAC	DMKCFG	DMKCFT	DMKCPI	DMKDAD	DMKDAS	DMKDDR	DMKDGD	DMKDID	DMKDMP
		DMKFPS	DMKHVE	DMKPRV	DMKRGE	DMKSSS	DMKTRK	DMKVSC	DMKVSI		
L20	000002	DMKDGD	DMKFMT								
L2048	000010	DMKCFG	DMKCPU	DMKPTR	DMKSTA						
L24	000001	DMKDAS									
L3	000032	DMKCFT	DMKCKH	DMKCKN	DMKDAS	DMKDEF	DMKDEI	DMKDMP	DMKDRD	DMKDRE	DMKFMT
		DMKMCH	DMKMNT	DMKPAG	DMKSAV	DMKSSP	DMKTRK				
L4	000072	DMKCCD	DMKCFG	DMKCFT	DMKCKH	DMKCPI	DMKCPS	DMKCQG	DMKDAD	DMKDAS	DMKDDR
		DMKDEI	DMKDGD	DMKDGF	DMKDIR	DMKDMP	DMKFMT	DMKHVE	DMKPAG	DMKPRV	DMKPTR
		DMKSAV	DMKSRM	DMKSSS	DMKTRK	DMKVDD					
L400	000001	DMKFMT									
L4096	000008	DMKDIR	DMKFMT								
L5	000040	DMKCKH	DMKCKN	DMKDDR	DMKDGD	DMKDMP	DMKFMT	DMKMNT	DMKSAV	DMKSSP	
L6	000009	DMKCPI	DMKDGD	DMKDGF	DMKFMT						
L7	000009	DMKCAO	DMKDDR	DMKDEF	DMKDEI	DMKDIR	DMKFMT	DMKSAV			
L8	000142	DMKAPI	DMKAPS	DMKCAO	DMKCKH	DMKCKN	DMKCLK	DMKCQP	DMKDAD	DMKDDR	DMKDEI
		DMKDMP	DMKFMT	DMKLOH	DMKMCH	DMKPAG	DMKPGU	DMKPMA	DMKSAV	DMKTOD	DMKTRK
		DMKZTD									
L80	000001	DMKFMT									
L9	000002	DMKDDR	DMKDEI								
MACRO	000001	DMKDMP									
MAINHIGH	000004	DMKIMG									
MAINPSA	000006	DMKDMQ									
MASKLINK	000001	DMKLNK									
MASKLOG	000002	DMKLOG									
MATCH	000002	DMKFPS	DMKHVF								
MAXCSIZE	000001	DMKFRE									
MAXHSIZ	000002	DMKFRE	DMKFRT								
MAXLEN	000021	DMKFMT	DMKSSP								
MAXSIZE	000012	DMKFRE									
MAXSPSIZ	000002	DMKFRE									
MCCPUID	000001	DMKMCH									
MCFXDLOG	000014	DMKMCH									
MCHAREA	000021	DMKACR	DMKCCH	DMKCFU	DMKCPO	DMKCPP	DMKCPU	DMKDID	DMKHVC	DMKIOG	DMKIOH
		DMKMCH	DMKMCI	DMKMCT	DMKMHC	DMKMHV	DMKPRV	DMKVAT	DMKVAU		
MCHCPEX	000005	DMKCPP	DMKCPU	DMKIOG	DMKMCH						
MCHDAMLN	000001	DMKIOG									
MCHEK	000035	DMKAPI	DMKCLK	DMKCPI	DMKDMP	DMKDMQ	DMKMCH	DMKPMA	DMKSAD	DMKSTA	DMKSVD
		DMKTOD									
MCHFIX	000006	DMKCPP	DMKCPU	DMKIOG	DMKMCH						
MCHFLAG0	000007	DMKMCH									

LABEL	COUNT	REFERENCES									
MCHFLAG1	000012	DMKACR	DMKMCH	DMKMCT							
MCHFLAG3	000024	DMKMCH	DMKMCT								
MCHFLAG4	000002	DMKMCH									
MCHFLAG5	000002	DMKMCH									
MCHFLAG6	000001	DMKMCH									
MCHFLAG7	000037	DMKMCH	DMKMCT								
MCHFSAR	000007	DMKMCH									
MCHLEN	000001	DMKMCH									
MCHLEN1	000004	DMKCPP	DMKCPU	DMKIOG	DMKMCH						
MCHMODEL	000050	DMKCCH	DMKCFU	DMKCPO	DMKDID	DMKHVC	DMKIOG	DMKIOH	DMKMCH	DMKMCI	DMKMHC
		DMKMHV	DMKPRV	DMKVAT	DMKVAU						
MCHPDAR1	000009	DMKMCH									
MCHPDAR6	000003	DMKMCH									
MCHPDAR7	000013	DMKMCH									
MCHPROCA	000001	DMKIOG									
MCHP1IDE	000001	DMKMCH									
MCHP1IKE	000002	DMKMCH									
MCHP1SDE	000001	DMKMCH									
MCHP1SKE	000002	DMKMCH									
MCHP1STD	000004	DMKMCH									
MCHP6CBA	000003	DMKMCH									
MCHREC	000008	DMKCPP	DMKCPU	DMKIOG	DMKMCH						
MCHRESEV	000002	DMKMCH									
MCH0HDWR	000002	DMKMCH									
MCH0QUIT	000001	DMKMCH									
MCH0SFTR	000001	DMKMCH									
MCH0TERM	000001	DMKMCH									
MCH0USAD	000001	DMKMCH									
MCH1BUFF	000001	DMKMCH									
MCH1COST	000002	DMKMCH									
MCH1GERR	000002	DMKACR	DMKMCH								
MCH1IOTO	000001	DMKMCH									
MCH1MAIN	000002	DMKMCH									
MCH1PROC	000002	DMKMCH									
MCH1TODC	000002	DMKMCH	DMKMCT								
MCH3BCST	000003	DMKMCH	DMKMCT								
MCH3DATA	000002	DMKMCH									
MCH3DGRD	000004	DMKMCH									
MCH3HARD	000004	DMKMCH									
MCH3INTE	000002	DMKMCH									
MCH3PASS	000002	DMKMCH									
MCH3PROT	000001	DMKMCH									
MCH3SOFT	000003	DMKMCH									
MCH3SOLD	000004	DMKMCH									
MCH4BURE	000001	DMKMCH									
MCH4REPA	000001	DMKMCH									
MCH5IFSA	000002	DMKMCH									
MCH7CHTM	000003	DMKMCH									
MCH7EXIT	000002	DMKMCH									
MCH7GSTR	000002	DMKMCH									
MCH7IOEM	000002	DMKMCH									
MCH7OPSW	000016	DMKMCH	DMKMCT								
MCH7PURG	000002	DMKMCH									
MCH7RSRE	000002	DMKMCH									
MCH7SMCR	000003	DMKMCH									
MCH7SUP	000003	DMKMCH									
MCH7SYST	000008	DMKMCH									
MCH7TRQ	000002	DMKMCH									

LABEL	COUNT	REFERENCES
MCH7VEQR	000002	DMKMCH
MCH7VRTM	000004	DMKMCH
MCNPSW	000026	DMKACR DMKCCH DMKCKP DMKCPJ DMKIOG DMKMCH DMKPMA DMKSAD DMKSAV DMKSSP
MCOLDPW	000002	DMKMCH
MCOPSW	000010	DMKMCH DMKPMA DMKSAD
MCPROGID	000002	DMKMCH
MCREC	000002	DMKMCH
MCRECORD	000001	DMKMCH
MCRECTYP	000001	DMKMCH
MCSWONE	000001	DMKMCH
MCSWTWO	000002	DMKMCH
MCS1PASS	000001	DMKMCH
MCS2HARD	000001	DMKMCH
MDRCUA1	000003	DMKVER
MDRKEYN	000002	DMKVER
MDRREC	000004	DMKVER
MDRSENS	000003	DMKVER
MDRSIZE	000001	DMKVER
MDRSIZE1	000001	DMKVER
MDRSWS3	000002	DMKVER
MDRVOL	000001	DMKVER
MEMO1	000034	DMKCCD DMKCCF DMKCCO DMKCCW
MEMO2	000044	DMKCCD DMKCCF DMKCCO DMKCCS DMKCCT DMKCCW
MEMO3	000014	DMKCCD DMKCCS DMKCCT DMKCCW
MESSAGE	000004	DMKCFC DMKCKM DMKMCI
MFAMASK	000012	DMKACR DMKAPI DMKCPI DMKCPU DMKDSP DMKEXT DMKIOT DMKLOK DMKPMA
MFASAVE	000003	DMKMCT
MICBLOK	000080	DMKAPI DMKBLD DMKCDS DMKCFF DMKCFG DMKCFO DMKCFP DMKCFS DMKCFY DMKCPB
		DMKCPI DMKCPP DMKCPU DMKDSP DMKFPS DMKIOT DMKIUN DMKLOJ DMKMCH DMKPGS
		DMKPMA DMKPRV DMKPTR DMKQVM DMKRPA DMKSAD DMKSCH DMKSEL DMKSPM DMKSTR
		DMKSWA DMKTMR DMKTRA DMKTRQ DMKVAT DMKVRR
MICCREG	000005	DMKCFS DMKCFY DMKLOJ
MICCREG0	000010	DMKCDS DMKCFF DMKCFP DMKCFS DMKCFY DMKLOJ DMKPRV
MICDASA	000010	DMKAPI DMKCFG DMKCFO DMKCFS DMKCPI DMKLOJ DMKVAT DMKVRR
MICDIAG	000001	DMKCPI
MICEVMA	000004	DMKCFS DMKCFY DMKLOJ
MICEVMA2	000018	DMKCFP DMKCFS DMKCPB DMKDSP DMKFPS DMKLOJ DMKQVM DMKSCH DMKSPM DMKTMR
		DMKTRQ DMKVAT DMKVRR
MICEVMA3	000033	DMKAPI DMKCFG DMKCFO DMKCFS DMKCFY DMKCPI DMKDSP DMKLOJ DMKPMA DMKSPM
		DMKVAT DMKVRR
MICEVMA4	000005	DMKCFS DMKPMA
MICFSSE	000011	DMKCFG DMKCFO DMKCFS DMKLOJ DMKVAT DMKVRR
MICIOSW	000003	DMKPMA
MICIPTP2	000003	DMKDSP DMKVAT
MICISKE	000005	DMKAPI DMKCFS DMKCFY DMKCPI DMKDSP
MICLCTL2	000001	DMKVAT
MICLPSW	000001	DMKCPI
MICLRA2	000003	DMKDSP DMKVAT
MICPEND	000005	DMKDSP DMKFPS DMKIUN
MICPMAMP	000005	DMKCFG DMKSPM DMKVRR
MICPMAV	000005	DMKCFG DMKPMA DMKVRR
MICPMMSK	000005	DMKCFG DMKIOT DMKVRR
MICPMPSA	000005	DMKCFG DMKCPP DMKCPU DMKSAD DMKVRR
MICPTLB	000001	DMKLOJ
MICPTLB2	000003	DMKDSP DMKVAT
MICRRBE	000004	DMKCFS DMKCFY DMKDSP
MICRSEG	000004	DMKBLD DMKCFS DMKCFY DMKLOJ
MICSCSP	000001	DMKCPI

LABEL	COUNT	REFERENCES									
MICSIO	000003	DMKCPI	DMKLOJ								
MICSIZE	000024	DMKBLD	DMKCFS	DMKCFY	DMKLOJ	DMKPMA	DMKUSQ				
MICSKYMD	000004	DMKAPI	DMKCFY	DMKCPI	DMKLOJ						
MICSPT	000009	DMKCFP	DMKCPB	DMKFPS	DMKQVM	DMKSCH	DMKTMR	DMKTRQ			
MICSPT4	000002	DMKCFS									
MICSSKE	000003	DMKCFS	DMKCFY	DMKDSP							
MICSTBVR	000009	DMKCFS	DMKDSP	DMKLOJ	DMKSPM	DMKVAT	DMKVRR				
MICSTPT	000001	DMKCPI									
MICSTSM	000001	DMKCPI									
MICSTSM2	000004	DMKLOJ	DMKSPM	DMKVAT							
MICSVC4	000002	DMKCFS									
MICTCH	000002	DMKCPI									
MICVIP	000004	DMKDSP	DMKFPS								
MICVPFR2	000003	DMKDSP	DMKVAT								
MICVPSW	000004	DMKCFS	DMKCFY	DMKIUN	DMKLOJ						
MICVTMR	000020	DMKCFG	DMKCFY	DMKLOJ	DMKMCH	DMKPGS	DMKPTR	DMKRPA	DMKSEL	DMKSTR	DMKSWA
		DMKTRA									
MICWORK	000003	DMKCFS	DMKCFY	DMKLOJ							
MIHCE	000001	DMKDID									
MIHCPID	000001	DMKIOE									
MIHCUA1	000003	DMKIOE	DMKVER								
MIHCUA2	000001	DMKIOE									
MIHDASD	000001	DMKDID									
MIHDE	000002	DMKCFQ	DMKCFR								
MIHDEVT	000001	DMKIOE									
MIHGRAF	000001	DMKDID									
MIHINT	000004	DMKIOE									
MIHKEYN	000002	DMKIOE	DMKVER								
MIHMISC	000002	DMKCFQ	DMKDID								
MIHMSG	000009	DMKCFQ	DMKCFR	DMKDID							
MIHMSGCS	000006	DMKCFQ	DMKCFR	DMKDID							
MIHMSGDV	000003	DMKCFQ	DMKCFR	DMKDID							
MIHMSGDW	000006	DMKCFQ	DMKCFR	DMKDID							
MIHMSGID	000006	DMKCFQ	DMKCFR	DMKDID							
MIHMSGLN	000006	DMKCFQ	DMKCFR	DMKDID							
MIHREC	000003	DMKIOE	DMKVER								
MIHSIZE	000002	DMKIOE	DMKVER								
MIHSWS3	000001	DMKIOE									
MIHTAPE	000001	DMKDID									
MIHUR	000001	DMKDID									
MIHUSER	000001	DMKIOE									
MIHVOL	000002	DMKIOE	DMKVER								
MINLEAVE	000001	DMKFRE									
MNBHDLEN	000002	DMKMCC	DMKMIA								
MNCHDAT1	000004	DMKENT									
MNCHDAT2	000004	DMKENT									
MNCHDT11	000004	DMKENT									
MNCHDT22	000004	DMKENT									
MNCHLIST	000004	DMKENT	DMKMOO								
MNCHSAMP	000003	DMKENT									
MNCHSAM1	000006	DMKENT	DMKMOO								
MNCHSAM2	000001	DMKMOO									
MNCHSIZ	000003	DMKENT	DMKMOO								
MNCHSIZE	000007	DMKENT	DMKMNI	DMKMOO							
MNCHSZ	000010	DMKENT	DMKMNI								
MNCHZMP	000001	DMKMNI									
MNCLDAST	000003	DMKMNI	DMKMNL	DMKMOO							
MNCLINST	000005	DMKPRV	DMKPRW								

LABEL	COUNT	REFERENCES				
MNCLPERF	000004	DMKMNI	DMKMON	DMKMOO		
MNCLRESP	000007	DMKGRG	DMKQCN	DMKQCO	DMKRGC	
MNCLSCH	000003	DMKSCH				
MNCLSEEK	000001	DMKIOS				
MNCLSWAP	000009	DMKPTR	DMKPTS	DMKSWA		
MNCLSYS	000001	DMKMON				
MNCLUSER	000002	DMKMNI	DMKMOO			
MNCOAEL	000001	DMKSCH				
MNCOAQ	000001	DMKSCH				
MNCOBRD	000001	DMKQCO				
MNCOCYL	000001	DMKIOS				
MNCODA	000001	DMKMON				
MNCODAS	000001	DMKMOO				
MNCODASH	000001	DMKMNI				
MNCODEIS	000001	DMKPTR				
MNCODEOS	000001	DMKSWA				
MNCODLSI	000001	DMKSWA				
MNCODLSO	000001	DMKPTS				
MNCODQ	000001	DMKSCH				
MNCODSIS	000002	DMKPTR				
MNCODSOS	000003	DMKSWA				
MNCOERD	000005	DMKGRG	DMKQCO	DMKRGC		
MNCOSIM	000005	DMKPRV	DMKPRW			
MNCOSUS	000001	DMKMON				
MNCOSYS	000001	DMKMOO				
MNCOTH	000001	DMKMNI				
MNCOTT	000001	DMKMNI				
MNCOUSER	000002	DMKMNI	DMKMOO			
MNCOWRIT	000001	DMKQCN				
MNCUBSY	000005	DMKENT				
MNDEVLEN	000004	DMKENT	DMKMNI			
MNDEVLMP	000002	DMKMNI				
MNDEVLST	000011	DMKENT	DMKMCC	DMKMNI	DMKMON	DMKMOO
MNDVBSY	000001	DMKENT				
MNDVBSY2	000004	DMKENT				
MNDVCNT	000011	DMKENT	DMKMNI	DMKMON	DMKMOO	
MNDVHDRL	000001	DMKMNI				
MNDVLEN	000006	DMKENT	DMKMNI	DMKMON	DMKMOO	
MNDVMORE	000010	DMKENT	DMKMCC	DMKMNI	DMKMOO	
MNDVSIZE	000004	DMKMCC	DMKMNI			
MNHCLASS	000001	DMKMON				
MNHCODE	000001	DMKMON				
MNHDR	000001	DMKMON				
MNHDRLEN	000004	DMKMON				
MNHRECSZ	000001	DMKMON				
MNHTOD	000001	DMKMON				
MNRDEVB	000005	DMKENT	DMKMNI	DMKMON	DMKMOO	
MN000	000002	DMKMOO				
MN000ATT	000001	DMKMOO				
MN000CDC	000001	DMKMOO				
MN000CPA	000001	DMKMOO				
MN000EXT	000001	DMKMOO				
MN000INT	000001	DMKMOO				
MN000ISD	000001	DMKMOO				
MN000LEN	000002	DMKMON	DMKMOO			
MN000PAP	000001	DMKMOO				
MN000PAS	000001	DMKMOO				
MN000PBP	000001	DMKMOO				

LABEL	COUNT	REFERENCES	
MN000PBS	000001	DMKMOO	
MN000PFP	000001	DMKMOO	
MN000PFS	000001	DMKMOO	
MN000PPA	000001	DMKMOO	
MN000PPC	000001	DMKMOO	
MN000PSI	000001	DMKMOO	
MN000PTP	000001	DMKMOO	
MN000PTS	000001	DMKMOO	
MN000Q1E	000002	DMKMOO	
MN000Q2E	000001	DMKMOO	
MN000RSV	000001	DMKMOO	
MN000SRC	000001	DMKMOO	
MN000SSI	000001	DMKMOO	
MN000VFO	000002	DMKMOO	
MN000VFV	000001	DMKMOO	
MN000VRC	000002	DMKMOO	
MN000VSC	000002	DMKMOO	
MN000WID	000001	DMKMOO	
MN001	000002	DMKMOO	
MN001LEN	000002	DMKMON	DMKMOO
MN001NXR	000001	DMKMOO	
MN001WID	000001	DMKMOO	
MN002	000001	DMKMOO	
MN002IBM	000002	DMKMOO	
MN002IBS	000001	DMKMOO	
MN002LEN	000001	DMKMOO	
MN002MNS	000002	DMKMOO	
MN002Q1A	000002	DMKMOO	
MN002Q1B	000001	DMKMOO	
MN002Q1C	000001	DMKMOO	
MN002Q1D	000001	DMKMOO	
MN002Q11	000002	DMKMOO	
MN002Q2A	000001	DMKMOO	
MN002Q2B	000001	DMKMOO	
MN002Q2C	000001	DMKMOO	
MN002Q2D	000001	DMKMOO	
MN002Q21	000003	DMKMOO	
MN002SQT	000001	DMKMOO	
MN002SQ3	000001	DMKMOO	
MN003	000001	DMKMOO	
MN003CDM	000001	DMKMOO	
MN003CIE	000001	DMKMOO	
MN003CIP	000001	DMKMOO	
MN003CMG	000001	DMKMOO	
MN003CRE	000001	DMKMOO	
MN003CSE	000001	DMKMOO	
MN003CSR	000001	DMKMOO	
MN003CTP	000001	DMKMOO	
MN003LEN	000001	DMKMOO	
MN003NSW	000001	DMKMOO	
MN003PNP	000001	DMKMOO	
MN003PNS	000001	DMKMOO	
MN003SGF	000001	DMKMOO	
MN003SWP	000001	DMKMOO	
MN004	000001	DMKMOO	
MN004HL	000001	DMKMOO	
MN004LEN	000001	DMKMOO	
MN004NP	000001	DMKMOO	

LABEL	COUNT	REFERENCES
MN004N2	000001	DMKMOO
MN005	000001	DMKMOO
MN005ADR	000003	DMKMOO
MN005CCN	000004	DMKMOO
MN005CIU	000002	DMKMOO
MN005DIR	000001	DMKMOO
MN005FLG	000001	DMKMOO
MN005LCN	000003	DMKMOO
MN005LEN	000001	DMKMOO
MN005MAL	000002	DMKMOO
MN005NOA	000002	DMKMOO
MN005PAL	000001	DMKMOO
MN005SER	000002	DMKMOO
MN005SZ	000001	DMKMOO
MN005TYP	000003	DMKMOO
MN006	000001	DMKMOO
MN006END	000001	DMKMOO
MN006IL	000003	DMKMOO
MN006LEN	000001	DMKMOO
MN006S1	000003	DMKMOO
MN0061C	000003	DMKMOO
MN007	000001	DMKMOO
MN007AS1	000001	DMKMOO
MN007AS2	000001	DMKMOO
MN007B1	000002	DMKMOO
MN007DPB	000001	DMKMOO
MN007FC1	000002	DMKMOO
MN007FC2	000003	DMKMOO
MN007LEN	000002	DMKMOO
MN007LSB	000002	DMKMOO
MN007NO1	000002	DMKMOO
MN007PSB	000004	DMKMOO
MN007S1	000005	DMKMOO
MN097	000001	DMKMNI
MN097APL	000002	DMKMNI
MN097CPL	000002	DMKMNI
MN097CPP	000001	DMKMNI
MN097CPU	000002	DMKMNI
MN097CR8	000001	DMKMNI
MN097DAT	000001	DMKMNI
MN097DPA	000003	DMKMNI
MN097FSS	000001	DMKMNI
MN097LEN	000001	DMKMNI
MN097LEV	000001	DMKMNI
MN097LSN	000003	DMKMNI
MN097MOD	000003	DMKMNI
MN097NUC	000001	DMKMNI
MN097PCP	000001	DMKMNI
MN097PCS	000001	DMKMNI
MN097PSS	000001	DMKMNI
MN097TIM	000001	DMKMNI
MN097TTS	000001	DMKMNI
MN097UID	000001	DMKMNI
MN097VR	000001	DMKMNI
MN098	000001	DMKMNI
MN098LEN	000001	DMKMNI
MN098UID	000001	DMKMNI
MN099	000001	DMKMON

LABEL	COUNT	REFERENCES
MN099CNT	000001	DMKMON
MN099LEN	000001	DMKMON
MN099TOD	000001	DMKMON
MN10X	000001	DMKMON
MN10XADD	000002	DMKMON
MN10XLEN	000001	DMKMON
MN10XUID	000001	DMKMON
MN10YCNT	000001	DMKMON
MN10YIO	000001	DMKMON
MN10YLEN	000001	DMKMON
MN20X	000001	DMKMON
MN20XNPP	000001	DMKMON
MN20XPRC	000001	DMKMON
MN20XQNM	000011	DMKMON
MN20XQ1E	000001	DMKMON
MN20XQ1N	000001	DMKMON
MN20XQ2E	000001	DMKMON
MN20XQ2N	000001	DMKMON
MN20XSWS	000001	DMKMON
MN20XUID	000001	DMKMON
MN20XWSS	000001	DMKMON
MN20YTTI	000001	DMKMON
MN20YVTI	000001	DMKMON
MN202APR	000001	DMKMON
MN202IOC	000001	DMKMON
MN202LEN	000001	DMKMON
MN202LPR	000001	DMKMON
MN202PGR	000001	DMKMON
MN202PRI	000001	DMKMON
MN202PST	000001	DMKMON
MN202QDP	000001	DMKMON
MN202QDR	000001	DMKMON
MN202QDT	000001	DMKMON
MN202REF	000002	DMKMON
MN202RES	000001	DMKMON
MN202RV2	000002	DMKMON
MN202SFG	000001	DMKMON
MN202VMR	000001	DMKMON
MN202VOT	000001	DMKMON
MN202VRL	000001	DMKMON
MN202VVT	000003	DMKMON
MN203LEN	000001	DMKMON
MN204LEN	000001	DMKMON
MN204PRI	000001	DMKMON
MN30X	000001	DMKMON
MN30XCYL	000001	DMKMON
MN30XEPG	000001	DMKMON
MN30XFLG	000001	DMKMON
MN30XLEN	000001	DMKMON
MN30XSSZ	000001	DMKMON
MN30XUID	000001	DMKMON
MN30XUPG	000001	DMKMON
MN30XVOL	000004	DMKMON
MN30XVPG	000001	DMKMON
MN300	000001	DMKMON
MN300END	000001	DMKMON
MN300LEN	000001	DMKMON
MN300RSP	000001	DMKMON

LABEL	COUNT	REFERENCES
MN300SST	000001	DMKMON
MN300TS	000003	DMKMON
MN300UID	000001	DMKMON
MN305	000001	DMKMON
MN305LEN	000001	DMKMON
MN305PN	000001	DMKMON
MN305PO	000003	DMKMON
MN305RSP	000001	DMKMON
MN305SN	000001	DMKMON
MN305SO	000001	DMKMON
MN305UID	000001	DMKMON
MN305WCT	000002	DMKMON
MN3880DV	000010	DMKMNL
MN38801L	000002	DMKMNL
MN4RSV1	000001	DMKMOO
MN400	000001	DMKMOO
MN400ACT	000001	DMKMOO
MN400ARC	000001	DMKMOO
MN400INT	000001	DMKMOO
MN400IOC	000001	DMKMOO
MN400LEN	000001	DMKMOO
MN400LPR	000001	DMKMOO
MN400MHL	000001	DMKMOO
MN400MLH	000001	DMKMOO
MN400MWS	000001	DMKMOO
MN400PDR	000001	DMKMOO
MN400PGR	000001	DMKMOO
MN400PSO	000003	DMKMOO
MN400PSP	000001	DMKMOO
MN400PUP	000001	DMKMOO
MN400PUS	000001	DMKMOO
MN400QDP	000001	DMKMOO
MN400QDR	000001	DMKMOO
MN400QDT	000002	DMKMOO
MN400RES	000001	DMKMOO
MN400RST	000001	DMKMOO
MN400SST	000001	DMKMOO
MN400SWI	000001	DMKMOO
MN400TTI	000001	DMKMOO
MN400UID	000001	DMKMOO
MN400UPR	000001	DMKMOO
MN400VOT	000001	DMKMOO
MN400VRC	000001	DMKMOO
MN400VRL	000001	DMKMOO
MN400VSC	000001	DMKMOO
MN400VTI	000001	DMKMOO
MN400VVT	000003	DMKMOO
MN400WCT	000003	DMKMOO
MN400WSS	000001	DMKMOO
MN410	000001	DMKMOO
MN410HWM	000001	DMKMOO
MN410LEN	000001	DMKMOO
MN410SST	000001	DMKMOO
MN500	000001	DMKMON
MN500INS	000001	DMKMON
MN500LEN	000001	DMKMON
MN500OVH	000001	DMKMON
MN500UID	000001	DMKMON

LABEL	COUNT	REFERENCES	
MN500VAD	000002	DMKMON	
MN600ADD	000006	DMKMNI	DMKMOO
MN600CNT	000002	DMKMNI	DMKMOO
MN600DEV	000002	DMKMNI	DMKMOO
MN600DLN	000005	DMKMNI	DMKMOO
MN600HDR	000003	DMKMNI	DMKMOO
MN600HLN	000006	DMKMNI	DMKMOO
MN600MAX	000001	DMKMNI	
MN600MXS	000001	DMKMNI	
MN600NUM	000003	DMKMNI	DMKMOO
MN600SER	000002	DMKMNI	DMKMOO
MN600TY	000002	DMKMNI	DMKMOO
MN602ADD	000003	DMKENT	
MN602CUB	000001	DMKENT	
MN602CUQ	000001	DMKENT	
MN602DEV	000001	DMKENT	
MN602DLN	000003	DMKENT	DMKMOO
MN602DVQ	000001	DMKENT	
MN602DV2	000001	DMKENT	
MN602HDR	000001	DMKENT	
MN602HLN	000002	DMKENT	DMKMOO
MN602MLN	000002	DMKENT	DMKMOO
MN602SAM	000001	DMKENT	
MN603CBI	000001	DMKENT	
MN603CB2	000001	DMKENT	
MN603CB3	000001	DMKENT	
MN603CB4	000001	DMKENT	
MN603CH	000001	DMKENT	
MN603CQI	000001	DMKENT	
MN603CQ2	000001	DMKENT	
MN603CQ3	000001	DMKENT	
MN603CQ4	000001	DMKENT	
MN603LNG	000001	DMKMOO	
MN603LNM	000001	DMKMOO	
MN605	000040	DMKMNL	
MN605BD	000003	DMKMNL	
MN605BW	000003	DMKMNL	
MN605DBU	000003	DMKMNL	
MN605DUO	000003	DMKMNL	
MN605LN	000003	DMKMNL	
MN605NPH	000003	DMKMNL	
MN605NPO	000003	DMKMNL	
MN605NPR	000003	DMKMNL	
MN605PMC	000003	DMKMNL	
MN605PMO	000003	DMKMNL	
MN605RS1	000003	DMKMNL	
MN605RS2	000003	DMKMNL	
MN605RS3	000003	DMKMNL	
MN605SIO	000003	DMKMNL	
MN605SIR	000003	DMKMNL	
MN605SPH	000003	DMKMNL	
MN605SPO	000003	DMKMNL	
MN605SPR	000003	DMKMNL	
MN605SRH	000003	DMKMNL	
MN700	000001	DMKMON	
MN700ADD	000001	DMKMON	
MN700CCY	000002	DMKMON	
MN700CHR	000003	DMKMON	

LABEL	COUNT	REFERENCES						
MN700CYL	000003	DMKMON						
MN700DIR	000002	DMKMON						
MN700LEN	000001	DMKMON						
MN700OPC	000001	DMKMON						
MN700PRO	000001	DMKMON						
MN700QCH	000001	DMKMON						
MN700QCU	000001	DMKMON						
MN700QDV	000001	DMKMON						
MN700RS1	000001	DMKMON						
MN700RS2	000001	DMKMON						
MN700UID	000001	DMKMON						
MN802CLN	000001	DMKMON						
MN802CNT	000001	DMKMON						
MN802CTR	000001	DMKMON						
MN802DEV	000001	DMKMON						
MN802DLN	000002	DMKMON						
MN802NAU	000001	DMKMON						
MN802NPP	000001	DMKMON						
MN802NUM	000001	DMKMON						
MN802PGR	000001	DMKMON						
MN802PGW	000001	DMKMON						
MN802PRB	000001	DMKMON						
MN802WID	000001	DMKMON						
MN802WIO	000001	DMKMON						
MN802WPG	000001	DMKMON						
MODEL135	000005	DMKCCH	DMKDID	DMKIOG	DMKMCH			
MODEL138	000001	DMKIOG						
MODEL145	000004	DMKCCH	DMKIOG	DMKMCH				
MODEL148	000001	DMKIOG						
MODEL155	000004	DMKCCH	DMKIOG	DMKMCH				
MODEL158	000001	DMKIOG						
MODEL165	000005	DMKCCH	DMKIOG	DMKMCH				
MODEL168	000001	DMKIOG						
MODESET	000010	DMKDDR	DMKVME					
MODESWT	000001	DMKNES						
MODNAME	000003	DMKIMG						
MODNOP	000001	DMKCCO						
MOD3031	000003	DMKCCH	DMKIOG	DMKMCH				
MOD3032	000001	DMKIOG						
MOD3033	000004	DMKCFU	DMKIOG	DMKIOH	DMKMCH			
MOD3081	000013	DMKCFU	DMKCPO	DMKIOG	DMKMCH	DMKMHC	DMKMHV	DMKVAT	DMKVAU
MOD3083	000001	DMKIOG						
MOD3084	000001	DMKIOG						
MOD3090	000010	DMKCCH	DMKCFU	DMKCPO	DMKHVC	DMKIOG	DMKMCH	DMKMHV	DMKPRV
MOD4321	000001	DMKIOG						
MOD4331	000004	DMKCCH	DMKIOG	DMKMCH				
MOD4341	000001	DMKIOG						
MOD4381	000005	DMKCCH	DMKIOG	DMKMCH	DMKMHV			
MOD9081	000001	DMKIOG						
MOD9083	000001	DMKIOG						
MOD9190	000001	DMKIOG						
MONAIOB	000012	DMKCPS	DMKDMQ	DMKMCC	DMKMIA	DMKMNI	DMKMON	
MONARDB	000007	DMKCPS	DMKDMQ	DMKMCC	DMKMNI	DMKMON		
MONATRB	000006	DMKMCC	DMKMNI	DMKMNJ				
MONBUFAC	000010	DMKMCC	DMKMIA	DMKMON				
MONBUFAV	000006	DMKMCC	DMKMON					
MONBUFIO	000007	DMKMIA	DMKMON					
MONBUF1	000013	DMKMCC	DMKMIA	DMKMNI	DMKMNJ	DMKMON		

LABEL	COUNT	REFERENCES									
MONBUF1V	000001	DMKMNI									
MONCAD	000002	DMKDMQ									
MONCHPTR	000012	DMKENT	DMKMCC	DMKMCD	DMKMNI	DMKMOO					
MONCLASS	000016	DMKMNI	DMKMNL	DMKMON	DMKMOO	DMKPRG					
MONCLKSA	000002	DMKMON									
MONCLOCK	000002	DMKMON									
MONCODE	000034	DMKMNI	DMKMNL	DMKMON	DMKMOO	DMKPRG	DMKTRX				
MONCOM	000029	DMKCPS	DMKDMQ	DMKENT	DMKMCC	DMKMCD	DMKMIA	DMKMNI	DMKMNJ	DMKMON	DMKMOO
MONCRSLT	000006	DMKMCC	DMKMON								
MONCURBF	000008	DMKMCC	DMKMIA	DMKMNI	DMKMON						
MONDAS	000014	DMKMIA									
MONDASA	000006	DMKMIA									
MONDASB	000006	DMKMIA									
MONDVLST	000013	DMKENT	DMKMCC	DMKMNI	DMKMON	DMKMOO					
MONDVNUM	000002	DMKMCC	DMKMNI								
MONEX	000005	DMKMIA	DMKMNI	DMKMON							
MONFLAG1	000034	DMKCPS	DMKMCC	DMKMCD	DMKMIA	DMKMNI	DMKMON	DMKMOO			
MONFLAG2	000013	DMKDMQ	DMKMIA	DMKMNI	DMKMON						
MONFLAG3	000050	DMKCPS	DMKDMQ	DMKMCC	DMKMCD	DMKMIA	DMKMNI	DMKMON			
MONIOBF	000023	DMKCPS	DMKMCC	DMKMCD	DMKMIA	DMKMNI	DMKMON				
MONIOSLT	000004	DMKMCC	DMKMON								
MONLSTBK	000001	DMKMON									
MONNEXT	000015	DMKMCC	DMKMIA	DMKMNI	DMKMON						
MONPDLY	000003	DMKMON									
MONREGS	000008	DMKMON	DMKPRG								
MONSACT	000004	DMKMOO									
MONSAVE1	000001	DMKMON									
MONSAVE2	000001	DMKMON									
MONSFB	000006	DMKMIA	DMKMNJ								
MONSIZE	000003	DMKMCC	DMKMNI								
MONSLMT	000005	DMKMCD	DMKMON	DMKMOO							
MONSPLCT	000008	DMKMIA	DMKMNJ	DMKMON							
MONSUSCK	000002	DMKMON									
MONSUSCT	000010	DMKMNI	DMKMON								
MONSYSVM	000003	DMKMOO									
MONTPERR	000003	DMKMON									
MONUSER	000008	DMKCPS	DMKMCC	DMKMCD	DMKMIA	DMKMNI	DMKMON				
MONUTRB	000010	DMKENT	DMKMCC	DMKMCD	DMKMNI	DMKMOO					
MON1BUF	000008	DMKMCC	DMKMIA	DMKMON							
MP	000362	DMKACO	DMKACR	DMKACS	DMKALG	DMKALO	DMKAPI	DMKAPS	DMKAPT	DMKAPU	DMKAPV
		DMKAPW	DMKAPX	DMKAPY	DMKAPZ	DMKBIO	DMKBLD	DMKCAO	DMKCCD	DMKCCF	DMKCCH
		DMKCCO	DMKCCS	DMKCCT	DMKCDS	DMKCFC	DMKCFF	DMKCFG	DMKCFH	DMKCFJ	DMKCFM
		DMKCFO	DMKCFP	DMKCFQ	DMKCFR	DMKCFS	DMKCFT	DMKCFU	DMKCFV	DMKCFW	DMKCFY
		DMKCKD	DMKCKF	DMKCKH	DMKCKM	DMKCKN	DMKCKP	DMKCKR	DMKCKS	DMKCKT	DMKCKV
		DMKCKW	DMKCNT	DMKCPB	DMKCPE	DMKCPI	DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPP
		DMKCPS	DMKCPT	DMKCPU	DMKCPV	DMKCPW	DMKCPX	DMKCPY	DMKCRM	DMKCSB	DMKCSC
		DMKCSF	DMKCSO	DMKCSP	DMKCSQ	DMKCSR	DMKCST	DMKCSU	DMKCSV	DMKCSW	DMKCSX
		DMKDEX	DMKDGD	DMKDRE	DMKDSB	DMKENT	DMKERP	DMKEXT	DMKHPS	DMKHPT	DMKIOE
		DMKIOG	DMKIOH	DMKIOJ	DMKIOQ	DMKIOS	DMKIOT	DMKLNM	DMKLOC	DMKLOG	DMKLOH
		DMKLOJ	DMKLOK	DMKLOM	DMKMCC	DMKMCD	DMKMCH	DMKMCI	DMKMNT	DMKMON	DMKMOO
		DMKMPO	DMKMSG	DMKNET	DMKNLD	DMKOPE	DMKPET	DMKPGM	DMKPGS	DMKPGT	DMKPRW
		DMKPSA	DMKPST	DMKPTR	DMKPTS	DMKPTT	DMKQCN	DMKQCO	DMKQCP	DMKRGA	DMKRGB
		DMKRGC	DMKRGD	DMKRNH	DMKRPA	DMKRPD	DMKRPI	DMKRPW	DMKRSE	DMKRSF	DMKRSP
		DMKRSQ	DMKSAV	DMKSBL	DMKSCH	DMKSCN	DMKSND	DMKSPK	DMKSPL	DMKSPM	DMKSPR
		DMKSPS	DMKSRM	DMKSSS	DMKSST	DMKSSU	DMKSSV	DMKSTA	DMKSTK	DMKSTP	DMKSVD
		DMKSWA	DMKTAQ	DMKTCS	DMKTMR	DMKTOD	DMKTRX	DMKTTX	DMKTTY	DMKTTZ	DMKUCB
		DMKUCC	DMKURS	DMKUSP	DMKVAU	DMKVBM	DMKVCA	DMKVCB	DMKVCH	DMKVCN	DMKVCP
		DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCU	DMKVCV	DMKVCW	DMKVER	DMKVFC	DMKVFD

LABEL	COUNT	REFERENCES									
		DMKVFE	DMKVFR	DMKVRS	DMKVSC	DMKVSD	DMKVSE	DMKVSF	DMKVSG	DMKVSI	DMKVSJ
		DMKVSP	DMKVSQ	DMKVSR	DMKVST	DMKVSU	DMKVSV	DMKVSW	DMKXAD	DMKXST	DMKZTD
MPFEAT	000012	DMKATS	DMKCFF	DMKCPI	DMKCPU	DMKCPY	DMKFRT	DMKMCC	DMKPRV	DMKQVM	DMKSPM
MPGEND	000016	DMKCPI	DMKCQP	DMKCQQ	DMKIOQ	DMKMCC	DMKMNI	DMKSCN			
MPUOPER	000053	DMKAPI	DMKCDB	DMKCDM	DMKCDS	DMKCKD	DMKCPO	DMKCPP	DMKCPS	DMKCPU	DMKCPZ
		DMKCQP	DMKCQQ	DMKENT	DMKEXT	DMKHVD	DMKIOG	DMKIOQ	DMKIOS	DMKMCT	DMKMNI
		DMKMNT	DMKMON	DMKMOO	DMKSAD	DMKSCN	DMKTHI	DMKVCH	DMKVDC		
MSFBLOK	000011	DMKCPU	DMKMHC	DMKMHV							
MSFCFLGS	000002	DMKMHV									
MSFDATA	000004	DMKMHC	DMKMHV								
MSFDB1	000001	DMKMHV									
MSFDB2	000001	DMKMHV									
MSFDB3	000001	DMKMHV									
MSFINFO	000001	DMKMHV									
MSFLNG	000004	DMKCPU	DMKMHV								
MSFRSP	000010	DMKCPU	DMKMHC	DMKMHV							
MSFR01	000002	DMKMHC	DMKMHV								
MSFR41	000001	DMKMHV									
MSF00	000001	DMKMHV									
MSGAALEN	000001	DMKIUL									
MSGAANAX	000001	DMKIUL									
MSGAANPX	000001	DMKIUL									
MSGAATOT	000001	DMKIUL									
MSGABLEN	000001	DMKIUE									
MSGABTOT	000001	DMKIUE									
MSGADDR	000002	DMKCSF	DMKCSO								
MSGAIINV	000001	DMKIUE									
MSGAITRN	000001	DMKIUE									
MSGALIST	000004	DMKIUL	DMKIUN								
MSGANSAD	000009	DMKIUL	DMKIUN	DMKIUS							
MSGANSLN	000019	DMKIUA	DMKIUB	DMKIUE	DMKIUL	DMKIUN	DMKIUS				
MSGAPPC	000018	DMKIUB	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUS		
MSGAPRMD	000001	DMKIUL									
MSGARCAX	000002	DMKIUE									
MSGARCPX	000002	DMKIUE									
MSGARJCT	000001	DMKIUG									
MSGARLST	000006	DMKIUE	DMKIUL								
MSGARPAX	000002	DMKIUL									
MSGARPLE	000002	DMKIUL	DMKIUS								
MSGARPPX	000002	DMKIUL									
MSGASNAX	000001	DMKIUE									
MSGASNPX	000001	DMKIUE									
MSGASVRD	000002	DMKIUJ									
MSGATINV	000001	DMKIUL									
MSGATTRN	000001	DMKIUL									
MSGAUDIT	000009	DMKIUA	DMKIUB	DMKIUG	DMKIUL	DMKIUS					
MSGAUDT1	000008	DMKIUE	DMKIUG	DMKIUL	DMKIUS						
MSGAUDT2	000014	DMKIUE	DMKIUJ	DMKIUL							
MSGAUDT3	000008	DMKIUE	DMKIUL								
MSGBLIST	000004	DMKIUE	DMKIUN								
MSGBLOK	000055	DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS
MSGCPIO	000001	DMKQCN									
MSGCTLS	000012	DMKIUB	DMKIUG	DMKIUL	DMKIUN	DMKIUS					
MSGCTLT	000012	DMKIUB	DMKIUE	DMKIUG	DMKIUN	DMKIUS					
MSGDESC	000011	DMKIUB	DMKIUE	DMKIUG	DMKIUN	DMKIUS					
MSGEMSG	000002	DMKQCN									
MSGERROR	000026	DMKIUA	DMKIUB	DMKIUJ	DMKIUL	DMKIUS					
MSGFLAGS	000094	DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUS	

LABEL	COUNT	REFERENCES									
MSGFLAG2	000028	DMKIUB	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUS		
MSGFLAG3	000032	DMKIUA	DMKIUB	DMKIUE	DMKIUG	DMKIUL	DMKIUN	DMKIUP	DMKIUS		
MSGFPNT	000070	DMKIUA	DMKIUB	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUS		
MSGID	000025	DMKIUB	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN				
MSGIMSG	000002	DMKQCN									
MSGKEY	000006	DMKIUE	DMKIUL	DMKIUN	DMKIUS						
MSGMASK1	000008	DMKIUB	DMKIUE	DMKIUG	DMKIUL	DMKIUN					
MSGMSG	000001	DMKMSG									
MSGNOFL	000003	DMKIUA	DMKIUN	DMKIUS							
MSGNORPY	000010	DMKIUB	DMKIUC	DMKIUE	DMKIUG	DMKIUN	DMKIUS				
MSGPARTL	000006	DMKIUE	DMKIUS								
MSGPRM	000007	DMKIUB	DMKIUE	DMKIUL	DMKIUN						
MSGPRMD	000015	DMKIUB	DMKIUE	DMKIUG	DMKIUL	DMKIUN	DMKIUS				
MSGPRTY	000010	DMKIUA	DMKIUB	DMKIUL	DMKIUN	DMKIUS					
MSGPURGE	000034	DMKIUA	DMKIUB	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUS		
MSGSCCLS	000015	DMKIUB	DMKIUG	DMKIUJ	DMKIUL	DMKIUN					
MSGSCIF	000001	DMKQCN									
MSGSCPID	000023	DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUS	
MSGSIZE	000054	DMKIUA	DMKIUN	DMKIUP	DMKIUS						
MSGSMSG	000001	DMKMSG									
MSGSNDAD	000010	DMKIUC	DMKIUE	DMKIUN	DMKIUS						
MSGSNDLN	000021	DMKIUB	DMKIUC	DMKIUE	DMKIUJ	DMKIUN	DMKIUS				
MSGSNDOP	000028	DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUJ	DMKIUL	DMKIUN	DMKIUS		
MSGTAG	000006	DMKIUA	DMKIUB	DMKIUG	DMKIUL	DMKIUN					
MSGTGCLS	000014	DMKIUB	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN				
MSGTGPID	000031	DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUS	
MSGUSED	000008	DMKIUA	DMKIUN	DMKIUP	DMKIUS						
MSGVMIO	000001	DMKQCN									
MSGWHTRC	000017	DMKIUA	DMKIUB	DMKIUE	DMKIUJ	DMKIUL	DMKIUS				
MSGWNG	000001	DMKMSG									
MSSFINTR	000002	DMKDSP	DMKEXT								
MSSFMASK	000003	DMKAPI	DMKCPI	DMKPMA							
MSSPRES	000022	DMKCFG	DMKCFQ	DMKCPI	DMKCPW	DMKDEI	DMKDGD	DMKHVE	DMKLNK	DMKLOJ	DMKSST
		DMKSSV	DMKUSQ	DMKVDA	DMKVDR	DMKVSJ					
MVSA370E	000007	DMKCFG	DMKDSP	DMKPRG	DMKQVM	DMKVRR					
M3880CUU	000002	DMKMNL									
M3880FLG	000006	DMKMNL									
M3880M13	000001	DMKMNL									
M3880NAC	000001	DMKMNL									
M3880RCU	000002	DMKMNL									
M3880RLN	000003	DMKMNL									
M3880SD2	000002	DMKMNL									
M3880SG	000001	DMKMNL									
M3880SSC	000005	DMKMNL									
M3880SSS	000001	DMKMNL									
M3880TOD	000003	DMKMNL									
M3880TOT	000001	DMKMNL									
NAME	000021	DMKTRP	DMKTRR								
NAMEENV	000005	DMKTRP	DMKTRR								
NAMEMIN	000002	DMKTRP	DMKTRR								
NAMENTRY	000003	DMKTRP	DMKTRR								
NAMEROUT	000002	DMKTRP	DMKTRR								
NAMESIZE	000004	DMKTRP	DMKTRR								
NAMNOPRO	000006	DMKTRP									
NAMPROC	000006	DMKTRP									
NAM2NUM	000001	DMKTRR									
NAM2WORD	000007	DMKTRP	DMKTRR								
NCPNAME	000002	DMKNLD	DMKSNC								

LABEL	COUNT	REFERENCES									
NCPPAGCT	000002	DMKNLD	DMKSNC								
NCPPNT	000006	DMKNLD	DMKSNC								
NCPSTART	000002	DMKNLD	DMKSNC								
NCPTBL	000003	DMKNLD	DMKSNC								
NCPVOL	000004	DMKNLD	DMKSNC								
NDISK	000001	DMKSAV									
NEEDSEEK	000016	DMKCCD	DMKCCF	DMKCCW							
NEWPAGES	000019	DMKBLD	DMKCFF	DMKDEG	DMKLOH	DMKSEG	DMKSTR	DMKVBM			
NEWSEGS	000011	DMKBLD	DMKCFP	DMKDEG	DMKLOH	DMKSEG	DMKVBM				
NICADFF	000007	DMKGRC	DMKGRT	DMKRGB							
NICADVF	000046	DMKGRC	DMKHVE	DMKNET	DMKRGA	DMKRGB	DMKRGC	DMKVDR			
NICAINH	000004	DMKRGB	DMKRGC								
NICALRM	000010	DMKRGA	DMKRGB	DMKRGC							
NICAPL	000013	DMKCFT	DMKCQU	DMKRGA	DMKRGB	DMKRGC	DMKVCN				
NICATOF	000006	DMKCFT	DMKCQU	DMKRNH							
NICATRB	000041	DMKCFM	DMKCFQ	DMKDIF	DMKLOH	DMKRGA	DMKRGB	DMKRGC			
NICATTN	000008	DMKRNH									
NICAWSF	000007	DMKGRC	DMKRGB	DMKRGC	DMKRGD	DMKVCN					
NICBLOK	000087	DMKACO	DMKBLD	DMKCFM	DMKCFQ	DMKCFT	DMKCFY	DMKCKD	DMKCKF	DMKCPJ	DMKCQG
		DMKCQT	DMKCQU	DMKDEF	DMKDIA	DMKDIB	DMKDIF	DMKERM	DMKEXT	DMKGRC	DMKGRT
		DMKHVD	DMKHVE	DMKLOG	DMKLOH	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKQCN	DMKQCO
		DMKRGA	DMKRGB	DMKRGC	DMKRGD	DMKRGE	DMKRNH	DMKSCN	DMKVCN	DMKVDR	DMKVDS
		DMKWRM									
NICCARD	000005	DMKRGC									
NICCDCNT	000004	DMKRGB									
NICCIBM	000006	DMKBLD	DMKDIA	DMKNES	DMKNET	DMKNLD	DMKRNH				
NICCORD	000007	DMKQCN	DMKRGA	DMKRGB	DMKRGC						
NICCPNA	000005	DMKRGA	DMKRGC								
NICDED	000001	DMKRNH									
NICDIAG	000010	DMKRGA	DMKRGB	DMKRGC							
NICDISA	000036	DMKCKD	DMKCPJ	DMKCQT	DMKDIB	DMKDIF	DMKNEA	DMKNES	DMKNET	DMKRGC	DMKRNH
		DMKWRM									
NICDISB	000016	DMKCKD	DMKNET	DMKRGA	DMKRGB	DMKRGC	DMKRNH	DMKVDR			
NICDMSG	000004	DMKDIB	DMKRGA	DMKRGC							
NICDTYPE	000038	DMKACO	DMKCFM	DMKCFT	DMKCFY	DMKCKF	DMKDIA	DMKHVE	DMKNEA	DMKNET	DMKRGB
		DMKRGC	DMKVCN	DMKVDS							
NICDXSC	000018	DMKCQG	DMKCQT	DMKDIB	DMKDIF	DMKHVE	DMKNET	DMKRGA	DMKRGC		
NICD3275	000010	DMKDIA	DMKHVE	DMKNEA	DMKRGB	DMKRGC					
NICD3276	000002	DMKHVE	DMKRGB								
NICD3277	000017	DMKCFM	DMKCFT	DMKCFY	DMKDIA	DMKRGB	DMKRGC	DMKVCN	DMKVDS		
NICD3278	000008	DMKCFT	DMKCFY	DMKDIA	DMKRGB	DMKVCN	DMKVDS				
NICD3279	000002	DMKCFM									
NICD3284	000007	DMKNEA	DMKNET	DMKRGB	DMKRGC	DMKVCN					
NICECOL	000006	DMKHVE	DMKRGB	DMKRGC							
NICEHLT	000006	DMKHVE	DMKRGB	DMKRGC							
NICENAB	000035	DMKCKD	DMKCQT	DMKDIB	DMKNEA	DMKNES	DMKNET	DMKRGA	DMKRGC	DMKRNH	DMKVDR
		DMKWRM									
NICEPAD	000010	DMKDIA	DMKDIF	DMKNEA	DMKNES	DMKNET	DMKNLD				
NICEPMD	000015	DMKDIB	DMKDIF	DMKNES	DMKNET	DMKNLD	DMKRNH				
NICERLK	000004	DMKRNH									
NICEWO	000004	DMKGRC	DMKRGB								
NICFLAG	000110	DMKCFT	DMKCKD	DMKCQT	DMKCQU	DMKDIB	DMKDIF	DMKLOG	DMKNEA	DMKNES	DMKNET
		DMKNLD	DMKRGA	DMKRGB	DMKRGC	DMKRNH	DMKVDR	DMKWRM			
NICFMT	000011	DMKNET	DMKRGA	DMKRGB	DMKRGC	DMKVDR					
NICGRAF	000002	DMKBLD	DMKHVD								
NICHOLD	000008	DMKRGA	DMKRGB	DMKRGC							
NICHT	000013	DMKGRC	DMKGRT	DMKQCN	DMKRGB	DMKRGC	DMKVCN				
NICLBSC	000002	DMKNES	DMKNET								

LABEL	COUNT	REFERENCES									
NICLGDRP	000004	DMKRGA									
NICLGRP	000008	DMKCKD	DMKNET	DMKRGA	DMKVCN	DMKWRM					
NICLINE	000013	DMKCKD	DMKDIA	DMKNES	DMKNET	DMKRNH					
NICLLEN	000006	DMKBLD	DMKCFT	DMKCQU	DMKHVE						
NICLOGDT	000003	DMKRGC									
NICLTRC	000013	DMKDIF	DMKNES	DMKRNH							
NICMDL	000002	DMKACO	DMKCKF								
NICMORE	000009	DMKRGA	DMKRGB	DMKRGC							
NICMTA	000002	DMKRNH									
NICNAME	000019	DMKBLD	DMKCPJ	DMKCQT	DMKDIF	DMKEXT	DMKNET	DMKNLD	DMKRGA	DMKRGC	DMKRNH
NICNTRL	000042	DMKRGA	DMKRGB	DMKRGC	DMKRNH						
NICOPRDR	000001	DMKACO									
NICPOLL	000004	DMKRGA	DMKRGB								
NICPSS	000001	DMKHVE									
NICPSUP	000012	DMKCFT	DMKLOG	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKRNH			
NICPT	000002	DMKGRC	DMKHVE								
NICQDONE	000002	DMKRGB	DMKRGC								
NICQPNT	000091	DMKCFQ	DMKDIF	DMKNES	DMKQCO	DMKRGA	DMKRGB	DMKRGC	DMKRNH		
NICQREP	000002	DMKGRC	DMKHVE								
NICQRY	000021	DMKNET	DMKRGA	DMKRGB	DMKRGC	DMKVDR					
NICRATTD	000006	DMKNET	DMKRGA	DMKRGB	DMKVDR						
NICRATTN	000009	DMKNET	DMKRGA	DMKRGB	DMKRGC	DMKVCN	DMKVDR				
NICRCNT	000012	DMKRNH									
NICRDED	000021	DMKCQT	DMKDIB	DMKDIF	DMKHVE	DMKNEA	DMKNET	DMKRGA	DMKRGB	DMKRGC	DMKVCN
		DMKVDR									
NICREAD	000010	DMKRGA	DMKRGB	DMKRGC							
NICRFLG	000064	DMKCQG	DMKCQT	DMKDIB	DMKDIF	DMKGRC	DMKHVE	DMKNEA	DMKNET	DMKRGA	DMKRGB
		DMKRGC	DMKRGD	DMKVCN	DMKVDR						
NICRFLG1	000007	DMKRGA	DMKRGC								
NICROPER	000003	DMKNEA	DMKVDR								
NICRSPL	000016	DMKCQG	DMKNEA	DMKNET	DMKRGA	DMKRGC	DMKSCN				
NICRUNN	000023	DMKNEA	DMKNET	DMKRGA	DMKRGB	DMKRGC					
NICSELT	000003	DMKRGA	DMKRGB								
NICSESN	000012	DMKDIB	DMKDIF	DMKNES	DMKNET	DMKRNH					
NICSIO	000005	DMKRGA	DMKRGB								
NICSIZE	000083	DMKACO	DMKACR	DMKCFM	DMKCFQ	DMKCFT	DMKCFY	DMKCKD	DMKCKF	DMKCPJ	DMKCPW
		DMKCQG	DMKCQT	DMKCQU	DMKDEF	DMKDIA	DMKDIB	DMKDIF	DMKERM	DMKEXT	DMKHVD
		DMKHVE	DMKLOG	DMKLOH	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKQCN	DMKQCO	DMKRGA
		DMKRGB	DMKRGC	DMKRNH	DMKSCN	DMKVCN	DMKVDR	DMKVDS	DMKWRM		
NICSTAT	000147	DMKCKD	DMKCPJ	DMKCQT	DMKDIA	DMKDIB	DMKDIF	DMKLOH	DMKNEA	DMKNES	DMKNET
		DMKNLD	DMKRGA	DMKRGB	DMKRGC	DMKRNH	DMKWRM				
NICSWEP	000007	DMKDIA	DMKDIB	DMKNES	DMKNLD						
NICTELE	000010	DMKDIF	DMKNET	DMKRNH							
NICTERM	000019	DMKBLD	DMKCKD	DMKNEA	DMKNET	DMKNLD	DMKRGA	DMKRNH	DMKWRM		
NICTEXT	000014	DMKCFT	DMKCQU	DMKRGA	DMKRGB	DMKRGC	DMKVCN				
NICTMAT	000006	DMKACO	DMKCKF	DMKDIB	DMKDIF	DMKNEA	DMKVDR				
NICTMCD	000041	DMKCFT	DMKCQU	DMKHVE	DMKRGA	DMKRGB	DMKRGC	DMKVCN			
NICTRQ	000015	DMKDIF	DMKLOH	DMKRGA	DMKRGB	DMKRGC					
NICTYPE	000076	DMKACO	DMKBLD	DMKCKD	DMKCKF	DMKCQG	DMKDIA	DMKDIF	DMKHVD	DMKNEA	DMKNES
		DMKNET	DMKNLD	DMKRGA	DMKRGB	DMKRGC	DMKRNH	DMKSCN	DMKVCN	DMKWRM	
NICUSER	000060	DMKBLD	DMKCQT	DMKDIB	DMKDIF	DMKEXT	DMKLOH	DMKNEA	DMKNES	DMKNET	DMKNLD
		DMKRGA	DMKRGB	DMKRGC	DMKRNH	DMKVDR					
NICUSEWA	000014	DMKRGA	DMKRGB	DMKRGC							
NICVDEVB	000019	DMKDEF	DMKDIB	DMKDIF	DMKNEA	DMKNET	DMKRGA	DMKRGB	DMKRGC	DMKRGE	DMKVCN
		DMKVDR									
NICWSF	000010	DMKRGA	DMKRGB	DMKRGC							
NICWTH	000016	DMKBLD	DMKERM	DMKGRC	DMKGRT	DMKHVE	DMKQCN	DMKRGB	DMKRGC	DMKVCN	
NIC14AD	000001	DMKHVE									

LABEL	COUNT	REFERENCES									
NIC14B	000003	DMKRGB									
NIC3274	000002	DMKRGB	DMKVCN								
NIC3275	000003	DMKRGA	DMKRGC								
NIC3276	000001	DMKRGB									
NLSNAME	000001	DMKHVF									
NLSNEXT	000006	DMKHVF									
NLSPGCT	000003	DMKHVF									
NLSSTRT	000003	DMKHVF									
NLSTBL	000005	DMKHVF									
NLSVOL	000001	DMKHVF									
NOADD	000006	DMKCKM	DMKCPO	DMKCPT	DMKPMA	DMKVDT					
NOAI	000002	DMKACO	DMKCKF								
NOALCARY	000035	DMKSCH	DMKSTP	DMKTRQ							
NOAUTO	000029	DMKCFM	DMKCFU	DMKCNS	DMKCPJ	DMKNLD	DMKNLE	DMKOPE	DMKOPR	DMKRNH	DMKTOD
		DMKTTY	DMKVCN	DMKVCS							
NOCARRY1	000003	DMKACO	DMKCKF								
NOCODE	000005	DMKEXT									
NOCOPY	000002	DMKSPL	DMKVSD								
NOINVPTE	000004	DMKPTT	DMKVFD	DMKVFE							
NOMC	000004	DMKGRG	DMKQCN	DMKRGC							
NOMODEL	000001	DMKIOG									
NOP	000018	DMKCCO									
NOPRINT	000001	DMKFMT									
NOPTLB	000012	DMKFPS	DMKVAT								
NORESTR	000002	DMKPTR	DMKPTT								
NORET	000601	DMKACO	DMKACR	DMKALO	DMKBLD	DMKCAC	DMKCAO	DMKCCH	DMKCDB	DMKCDM	DMKCDS
		DMKCFC	DMKCFD	DMKCFF	DMKCFH	DMKCFM	DMKCFS	DMKCFU	DMKCFV	DMKCFW	DMKCPB
		DMKCPI	DMKCPJ	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPU	DMKCPV	DMKCPW
		DMKCPY	DMKCPZ	DMKCQC	DMKCQG	DMKCQH	DMKCQI	DMKCQP	DMKCQQ	DMKCQR	DMKCQS
		DMKCQT	DMKCQU	DMKCQY	DMKCSB	DMKCSF	DMKCSO	DMKCSV	DMKCSX	DMKDAD	DMKDAS
		DMKDAU	DMKDEI	DMKDIB	DMKDIF	DMKDSB	DMKERM	DMKEXT	DMKGRG	DMKIDU	DMKJRL
		DMKLNM	DMKLOM	DMKMCC	DMKMCD	DMKMCH	DMKMCI	DMKMCT	DMKMIA	DMKMID	DMKMNI
		DMKMNJ	DMKMNT	DMKMSG	DMKMSW	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKOPE
		DMKPEI	DMKPEN	DMKPGM	DMKPGT	DMKPRG	DMKPTR	DMKQCN	DMKQCP	DMKQCQ	DMKQVM
		DMKRGC	DMKRNH	DMKSPL	DMKSPM	DMKSPR	DMKSPS	DMKSRM	DMKSVD	DMKTHI	DMKTOD
		DMKTRA	DMKTRC	DMKTRD	DMKTRP	DMKTRU	DMKUDR	DMKURS	DMKUSQ	DMKVCB	DMKVCH
		DMKVCN	DMKVDA	DMKVDB	DMKVDD	DMKVDR	DMKVDT	DMKVER	DMKVFC	DMKVFD	DMKVFE
		DMKVME	DMKVRR								
NORLSE	000009	DMKCFG	DMKRPA								
NOSIGP	000004	DMKPTR	DMKPTT	DMKSEL							
NOSLBORO	000015	DMKSCH	DMKSTP								
NOTERM	000009	DMKCPV	DMKCQG	DMKMCT	DMKVCH						
NOTIME	000059	DMKCFC	DMKCFM	DMKCLK	DMKCPI	DMKCPJ	DMKCQI	DMKGRG	DMKLOG	DMKMSG	DMKMSW
		DMKOPE	DMKQCN	DMKQCO	DMKQCQ	DMKRGC	DMKTOD	DMKURS	DMKUSQ	DMKVCN	
NOTIPL	000002	DMKCKM	DMKCPI								
NOTMID	000003	DMKACO	DMKCKF								
NOTRESP	000024	DMKACO	DMKCFH	DMKCFM	DMKCSV	DMKCSX	DMKEPS	DMKGRD	DMKGRG	DMKMID	DMKMSG
		DMKNLD	DMKNLE	DMKQCN	DMKQCO	DMKQCQ	DMKRGC	DMKSPL			
NOTUSED	000011	DMKBSC	DMKCKT	DMKDAD	DMKDAS	DMKDAU	DMKDIR	DMKDSB	DMKRSF	DMKSAD	DMKTAP
		DMKTAQ									
NPRCNT	000013	DMKCKV	DMKCPS	DMKCSO	DMKHVD	DMKRSP	DMKWRM				
NPRNAME	000008	DMKCKV	DMKCPS	DMKCSO	DMKHVD	DMKRSP	DMKTCS	DMKWRM			
NPRPAGCT	000003	DMKHVD									
NPRPNT	000021	DMKCKV	DMKCPS	DMKCSO	DMKHVD	DMKRSP	DMKTCS	DMKWRM			
NPRSTART	000002	DMKHVD	DMKTCS								
NPRTBL	000009	DMKCKV	DMKCPS	DMKCSO	DMKHVD	DMKRSP	DMKTCS	DMKWRM			
NPRVOL	000002	DMKHVD	DMKTCS								
NUCON	000002	DMKIMG	DMKNMT								

LABEL	COUNT	REFERENCES									
NUMCAPLS	000002	DMKFRE									
NUMPOOLS	000005	DMKMOO	DMKPXA	DMKPXB							
NUMSTEAL	000001	DMKFRE									
OBRBLKLN	000001	DMKIOJ									
OBRCCHS	000002	DMKIOJ	DMKVER								
OBRCORL	000001	DMKIOJ									
OBRCPIDN	000002	DMKIOJ	DMKVER								
OBRCSWN	000001	DMKIOJ									
OBRCUA	000003	DMKVER									
OBRCUAIN	000004	DMKIOJ	DMKVER								
OBRCUAPR	000004	DMKIOJ	DMKVER								
OBRDDCNT	000006	DMKIOJ									
OBRDEMNT	000001	DMKVER									
OBRDEVSH	000015	DMKIOC	DMKIOE								
OBRDEVTN	000015	DMKIOC	DMKVER								
OBREOD	000001	DMKIOJ									
OBRFBSNS	000005	DMKVER									
OBRFCCWN	000001	DMKIOJ									
OBRHAN	000003	DMKIOJ	DMKVER								
OBRHSIZE	000001	DMKVER									
OBRIORTY	000001	DMKIOJ									
OBRKEYN	000012	DMKIOJ	DMKVER								
OBRLSIZE	000001	DMKVER									
OBRLSKN	000005	DMKIOJ	DMKVER								
OBRPBN	000001	DMKIOJ									
OBRPGMN	000002	DMKIOJ	DMKVER								
OBRRECN	000008	DMKIOC	DMKIOE	DMKIOJ	DMKVER						
OBRSDRCT	000008	DMKIOJ									
OBRSDRSH	000001	DMKIOJ									
OBRSENSN	000007	DMKVER									
OBRSHOBR	000010	DMKIOC	DMKIOE	DMKIOJ							
OBRSNSCT	000004	DMKIOJ									
OBRSSDR1	000001	DMKIOJ									
OBRSSIZE	000001	DMKVER									
OBRSWSN	000023	DMKIOC	DMKIOE	DMKIOJ	DMKVER						
OBRTEMP	000002	DMKIOJ									
OBRVOLN	000003	DMKIOJ	DMKVER								
OBR2SIZE	000001	DMKVER									
OBR3SIZE	000001	DMKVER									
OBR33SNS	000019	DMKIOJ	DMKVER								
OFF	000020	DMKCAC	DMKCFU	DMKCMD	DMKCQC	DMKCSV	DMKCSX	DMKDSP	DMKLOK	DMKMCH	DMKMCT
		DMKSPT	DMKSRM	DMKVAT							
OFFLPROC	000006	DMKCPO	DMKVCH								
OLDKEYOP	000007	DMKCKP	DMKFPS	DMKPRV	DMKSTA						
OLDVMSEG	000014	DMKBLD	DMKCFF	DMKCFP	DMKPGS						
ON	000015	DMKCAC	DMKCMD	DMKCPT	DMKCPU	DMKCQC	DMKCQY	DMKRND	DMKVAT		
ONEENT	000005	DMKRGA	DMKRGD	DMKVCN	DMKVCV						
OPERATOR	000134	DMKACR	DMKCAC	DMKCCH	DMKCFP	DMKCLK	DMKCPJ	DMKCPP	DMKCPY	DMKDAD	DMKDAS
		DMKDAU	DMKDIB	DMKDIF	DMKDSB	DMKERM	DMKGRD	DMKLOM	DMKMCH	DMKMCI	DMKMCT
		DMKMIA	DMKMSW	DMKNEA	DMKNLD	DMKNLE	DMKOPE	DMKPGT	DMKQCN	DMKQCP	DMKRNH
		DMKSPR	DMKUDR	DMKURS	DMKUSQ	DMKVCH	DMKVDA	DMKVDB	DMKVDD	DMKVDR	DMKVDT
		DMKVER	DMKVRR	DMKWRM							
OPNSFB	000020	DMKACO	DMKCKR	DMKCKV	DMKMIA	DMKTRT	DMKTRU	DMKVST			
OPPRSTRT	000015	DMKPTT	DMKSEL								
ORIGSEL	000006	DMKTRP	DMKTRR								
OVERLAP	000003	DMKPEL									
OWNDLIST	000042	DMKACR	DMKALO	DMKCFS	DMKCFU	DMKCKF	DMKCKM	DMKCKN	DMKCKV	DMKCPI	DMKCPJ
		DMKCPO	DMKCQY	DMKDRD	DMKDRE	DMKIDU	DMKMNT	DMKMON	DMKPAG	DMKPAH	DMKPGU

LABEL	COUNT	REFERENCES									
		DMKSCN	DMKSPK	DMKSPS	DMKUDR	DMKVDA	DMKVDG	DMKVSE	DMKVSG	DMKWRM	
OWNDRDEV	000035	DMKACR	DMKALO	DMKCFS	DMKCFU	DMKCKF	DMKCKM	DMKCKN	DMKCKV	DMKCPI	DMKCPJ
		DMKCPO	DMKCQY	DMKDRD	DMKDRE	DMKIDU	DMKPAG	DMKPAH	DMKPGU	DMKSCN	DMKSPS
		DMKUDR	DMKVDA	DMKVDG	DMKVSE	DMKVSG	DMKWRM				
OWNDVSER	000013	DMKCFS	DMKCKM	DMKCKN	DMKCKV	DMKCPJ	DMKCQY	DMKMNT	DMKMON	DMKSCN	DMKUDR
		DMKVDA									
OWNID	000003	DMKCAO	DMKCSV	DMKCSX							
O2ENTRY	000003	DMKCQC	DMKTRP	DMKTRR							
O2LENGTH	000002	DMKTRP	DMKTRR								
O2MAXLEN	000001	DMKCQC									
O2NAME	000002	DMKTRP	DMKTRR								
O2SIZE	000003	DMKCQC	DMKTRP	DMKTRR							
O2TABEND	000002	DMKTRP	DMKTRR								
O2TABLE	000003	DMKTRP	DMKTRR								
O3DISP	000002	DMKTRP	DMKTRR								
O3ENTRY	000004	DMKTRP	DMKTRR								
O3HEXLTH	000004	DMKTRP	DMKTRR								
O3MCHFLD	000003	DMKTRP	DMKTRR								
O3NUMSZ	000003	DMKTRP	DMKTRR								
O3NUMOO	000006	DMKTRP	DMKTRR								
O3SIZE	000005	DMKCQC	DMKTRP	DMKTRR							
O3TABEND	000001	DMKTRR									
O3TABLE	000006	DMKTRP	DMKTRR								
O3TRPEND	000001	DMKTRP									
O4ENTRY	000002	DMKTRR									
O4FMTADR	000002	DMKTRR									
O4FORMAT	000002	DMKTRR									
O4MINLEN	000001	DMKTRR									
O4NAME	000001	DMKTRR									
O4SIZE	000002	DMKTRR									
O4TABEND	000002	DMKTRR									
O4TABLE	000005	DMKTRP	DMKTRR								
O4VALUE	000002	DMKTRR									
PAGACT	000031	DMKATS	DMKCFF	DMKPGS	DMKPTR	DMKPTS	DMKVMA				
PAGBMP	000040	DMKATS	DMKBLD	DMKCFF	DMKCPP	DMKCPU	DMKFRT	DMKPGS	DMKPTR		
PAGCORE	000209	DMKATS	DMKBLD	DMKCCW	DMKCDS	DMKCFF	DMKCFG	DMKCKM	DMKCKW	DMKCPP	DMKDAD
		DMKDAS	DMKDAU	DMKHVD	DMKMCH	DMKPGS	DMKPMA	DMKPSA	DMKPTR	DMKPTS	DMKPTT
		DMKRPA	DMKSEG	DMKSEL	DMKSTR	DMKSWA	DMKSWM	DMKVFR	DMKVMA		
PAGECCWS	000006	DMKPAG	DMKPAH								
PAGECUR	000004	DMKCPP	DMKMCC	DMKMNI							
PAGECYL	000031	DMKMON	DMKPAG								
PAGEFBAT	000004	DMKPAG	DMKPAH								
PAGEFSNS	000001	DMKPAG									
PAGEFTIC	000002	DMKPAG									
PAGEHEAD	000022	DMKPAG									
PAGEIDA1	000004	DMKPAG									
PAGEIDA2	000002	DMKPAG									
PAGEIOB	000024	DMKPAG	DMKPAH								
PAGELOAD	000004	DMKCPP	DMKSTP	DMKTHI							
PAGELOCA	000005	DMKPAG	DMKPAH								
PAGELOCB	000001	DMKPAG									
PAGELOCD	000004	DMKPAG	DMKPAH								
PAGELOCN	000004	DMKPAG	DMKPAH								
PAGELOCO	000003	DMKPAG									
PAGELOCW	000001	DMKPAG									
PAGEND	000008	DMKMCC	DMKMIA	DMKMNI	DMKMON						
PAGENUME	000005	DMKPAG									
PAGENXT	000001	DMKMNI									

LABEL	COUNT	REFERENCES									
PAGEPARM	000008	DMKPAG									
PAGEPRI	000006	DMKPAG									
PAGERATE	000004	DMKCPP	DMKSTP	DMKTHI							
PAGERCD	000010	DMKPAG									
PAGERW	000003	DMKPAG	DMKPAH								
PAGESECT	000011	DMKPAG									
PAGESEEK	000005	DMKPAG	DMKPAH								
PAGESIZE	000024	DMKPAG	DMKSTA	DMKUSP							
PAGESK	000029	DMKPAG	DMKPAH								
PAGESLOT	000011	DMKPAG									
PAGESNS	000022	DMKPAG	DMKPAH								
PAGESRCD	000029	DMKPAG									
PAGESRCH	000009	DMKPAG									
PAGESS	000020	DMKPAG	DMKPAH								
PAGETYPE	000010	DMKPAG	DMKPAH								
PAGEWAIT	000011	DMKAPI	DMKCPI	DMKDSP	DMKMON	DMKSTP					
PAGEXDED	000002	DMKPAG									
PAGEXECC	000001	DMKPAG									
PAGEXEHH	000001	DMKPAG									
PAGEXFER	000011	DMKPAG	DMKPAH								
PAGEXLR	000006	DMKPAG	DMKPAH								
PAGEXLRD	000001	DMKPAG									
PAGEXOP	000002	DMKPAG									
PAGEXRW	000003	DMKPAG	DMKPAH								
PAGEXSHC	000001	DMKPAG									
PAGEXSHH	000001	DMKPAG									
PAGEXSIZ	000002	DMKPAG	DMKUSP								
PAGEXSKC	000002	DMKMON	DMKPAG								
PAGEXSKH	000001	DMKPAG									
PAGEXSLT	000006	DMKPAG									
PAGEXSL1	000002	DMKPAG									
PAGEXSNS	000004	DMKPAG	DMKPAH								
PAGEXSTR	000003	DMKPAG									
PAGEXST2	000002	DMKPAG									
PAGEXTIC	000005	DMKPAG	DMKPAH								
PAGE2K	000006	DMKCFV	DMKPRV	DMKPTT	DMKVAT						
PAGE4K	000008	DMKCPI	DMKFPS	DMKPMA							
PAGINVAL	000044	DMKATS	DMKCDS	DMKCKM	DMKCPP	DMKDAD	DMKDAS	DMKDAU	DMKMCH	DMKPGM	DMKPMA
		DMKPTR	DMKPTS	DMKPTT	DMKRPA	DMKSEL	DMKSTR	DMKSWA	DMKSWM	DMKVMA	
PAGPGSWP	000006	DMKBLD	DMKCFS	DMKCFY	DMKLOJ	DMKPMA	DMKSTR				
PAGRBITS	000016	DMKATS	DMKPGM	DMKPGS	DMKPTR	DMKSEL	DMKSTR				
PAGREF	000019	DMKMCH	DMKPGS	DMKPTR	DMKPTS	DMKRPA	DMKSEL	DMKSTR	DMKSWA	DMKSWM	
PAGSHR	000024	DMKATS	DMKCCW	DMKCFF	DMKCFG	DMKPGS	DMKPSA	DMKPTR	DMKPTS	DMKPTT	DMKSEL
PAGSWARG	000002	DMKPAG									
PAGSWCC	000001	DMKPAG									
PAGSWCHN	000002	DMKPAG	DMKPAH								
PAGSWCSZ	000003	DMKPAG									
PAGSWDA1	000002	DMKPAG									
PAGSWDA2	000001	DMKPAG									
PAGSWDSZ	000002	DMKPAG									
PAGSWEXT	000002	DMKPAG	DMKPAH								
PAGSWHH	000001	DMKPAG									
PAGSWNOP	000001	DMKPAG									
PAGSWP	000007	DMKATS	DMKBLD	DMKCFF	DMKCFG	DMKSTR	DMKVAT				
PAGSWR	000001	DMKPAG									
PAGSWRW	000001	DMKPAG									
PAGSWSCH	000003	DMKPAG									
PAGSWTIC	000001	DMKPAG									

LABEL	COUNT	REFERENCES									
PAGTABLE	000061	DMKATS	DMKBLD	DMKCFF	DMKCFG	DMKCPP	DMKCPU	DMKPGM	DMKPGS	DMKPTR	DMKPTS
		DMKPTT	DMKSEL	DMKSTR	DMKVAT	DMKVMA					
PAGTBSIZ	000001	DMKSTR									
PAGTONLY	000015	DMKBLD	DMKPGS	DMKSTR							
PAGTOT	000016	DMKATS	DMKCFF	DMKCPU	DMKPGM	DMKPGS					
PAGTSWP	000027	DMKATS	DMKCFF	DMKCPP	DMKPGS	DMKPTR	DMKSEL				
PAGXIDA1	000002	DMKPAG									
PAGXIDA2	000001	DMKPAG									
PARMSIZE	000001	DMKHVC									
PCHCHN	000009	DMKACO	DMKCKS	DMKCKV	DMKCSY	DMKSPL	DMKSPS	DMKWRM			
PCI	000039	DMKDAD	DMKDAS	DMKDSP	DMKHVC	DMKIOS	DMKIOT	DMKPAH	DMKRNH	DMKRSE	DMKTRK
		DMKVCA	DMKVCB	DMKVCN	DMKVIO	DMKVSI	DMKVSP	DMKVSX			
PCIF	000013	DMKCCW	DMKDGD	DMKPAG	DMKPAH	DMKVCA	DMKVCB	DMKVCN	DMKVIO	DMKVSP	DMKVSX
PDALLOCD	000019	DMKIUC	DMKIUE	DMKIUJ	DMKIUS						
PDAPLOCK	000012	DMKIUJ	DMKIUS								
PDAPMSGP	000007	DMKIUB	DMKIUN	DMKIUS							
PDAPPC	000024	DMKIUA	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS	
PDAPPCFL	000062	DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS
PDAPPEND	000005	DMKIUA	DMKIUJ	DMKIUS							
PDAPRECQ	000004	DMKIUS									
PDAPREQS	000003	DMKIUA	DMKIUS								
PDAPSNDQ	000006	DMKIUS									
PDAPSYCF	000003	DMKIUC	DMKIUS								
PDAVAIL	000003	DMKIUG	DMKIUJ	DMKIUP							
PDCNTRL	000017	DMKIUB	DMKIUC	DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS			
PDCPXQ	000005	DMKIUA	DMKIUJ								
PDENT	000099	DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS
PDENTMAX	000002	DMKIUG									
PDENTMIN	000003	DMKIUG									
PDFCNCD	000008	DMKIUC	DMKIUJ	DMKIUN	DMKIUP						
PDFLAGS	000083	DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS
PDFLAGS2	000013	DMKIUE	DMKIUJ	DMKIUL	DMKIUS						
PDINUSE	000003	DMKIUA									
PDLRCINV	000013	DMKIUE	DMKIUJ	DMKIUL	DMKIUS						
PDLRECL	000024	DMKIUE	DMKIUJ	DMKIUL	DMKIUS						
PDMSGCT	000010	DMKIUA	DMKIUJ	DMKIUN	DMKIUS						
PDMSGLIM	000004	DMKIUC	DMKIUN								
PDPEND1	000006	DMKIUC	DMKIUJ	DMKIUS							
PDPEND2	000006	DMKIUC	DMKIUJ	DMKIUS							
PDPRMD	000005	DMKIUC	DMKIUL	DMKIUN							
PDPRTY	000005	DMKIUC	DMKIUN								
PDSEND	000011	DMKIUC	DMKIUJ	DMKIUN	DMKIUP						
PDSEVERD	000024	DMKIUA	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUP	DMKIUS		
PDSINV	000003	DMKIUA	DMKIUG								
PDSIZE	000006	DMKIUG	DMKIUJ								
PDSTALLC	000004	DMKIUC	DMKIUJ	DMKIUS							
PDSTATE	000041	DMKIUB	DMKIUC	DMKIUE	DMKIUJ	DMKIUN	DMKIUS				
PDSTCONF	000008	DMKIUJ	DMKIUS								
PDSTCONN	000012	DMKIUC	DMKIUE	DMKIUJ	DMKIUS						
PDSTRECV	000016	DMKIUB	DMKIUC	DMKIUJ	DMKIUN	DMKIUS					
PDSTSEND	000015	DMKIUB	DMKIUC	DMKIUJ	DMKIUS						
PDSTSEVR	000006	DMKIUJ	DMKIUS								
PDTGIUCV	000025	DMKIUA	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS	
PDTGPID	000033	DMKIUC	DMKIUG	DMKIUJ	DMKIUN	DMKIUP	DMKIUS				
PDVALID	000005	DMKIUA	DMKIUC	DMKIUG							
PDZERO	000003	DMKIUJ									
PENDCONN	000008	DMKIUC	DMKIUJ								
PERADD	000004	DMKDSP	DMKPMA	DMKPRG							

LABEL	COUNT	REFERENCES									
PERADDR	000011	DMKPER	DMKPET	DMKPRG	DMKPRV	DMKPRW	DMKTMR				
PERANYTH	000010	DMKPEI	DMKPEN								
PERAPPND	000006	DMKPEN									
PERBLIP	000003	DMKPEI	DMKPER								
PERBLOK	000020	DMKDSP	DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPER	DMKPET	DMKPRG	DMKPRV	DMKPRW
		DMKSVD	DMKTMR								
PERBUF	000050	DMKPEI	DMKPEN	DMKPEQ	DMKPER	DMKPET					
PERCDE	000007	DMKPER									
PERCHAIN	000031	DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPER	DMKPET				
PERCHANG	000002	DMKPEN									
PERCLEAR	000002	DMKPEN									
PERCMDPT	000005	DMKPEI	DMKPEL								
PERCNTCH	000016	DMKPEL	DMKPEN								
PERCODE	000004	DMKDSP	DMKPMA	DMKPRG							
PERCOUNT	000015	DMKPEL	DMKPEN								
PERCR10	000001	DMKPEL									
PERCR11	000001	DMKPEL									
PERCR9	000017	DMKDSP	DMKPEI	DMKPEL	DMKPER	DMKPRV	DMKPRW	DMKTMR			
PERCTACT	000010	DMKPEI	DMKPEN	DMKPEQ	DMKPER						
PERCTEND	000002	DMKPEN									
PERDATA	000007	DMKPEI									
PERDATON	000011	DMKPER	DMKPET	DMKPRG	DMKPRV	DMKSVD					
PERENDIT	000002	DMKPEN									
PERERROR	000006	DMKPEI									
PEREX	000017	DMKPER	DMKPET								
PEREXADD	000005	DMKPER	DMKPET								
PEREXMOD	000003	DMKPER	DMKPET								
PERFCL	000008	DMKMCC	DMKMIA	DMKMNI	DMKMOO						
PERFLAG	000021	DMKPEI	DMKPEN	DMKPEQ	DMKPER	DMKPET	DMKPRG	DMKPRV	DMKSVD		
PERGALT	000003	DMKPER									
PERGPRP	000006	DMKPEI	DMKPEN	DMKPER							
PERGPRS	000009	DMKMPO	DMKPRV	DMKPRW							
PERHITS	000009	DMKPEI	DMKPEN	DMKPEQ	DMKPER						
PERINST	000043	DMKPER	DMKPET								
PERINTO	000004	DMKPEI									
PERLNEND	000003	DMKPEI									
PERMODE	000008	DMKDSP	DMKPEL	DMKPEN	DMKPMA	DMKSVD	DMKTRC				
PEROFF	000005	DMKPEL									
PERON	000007	DMKPEL									
PEROPNOT	000006	DMKPER	DMKPET								
PEROPQU	000008	DMKPER									
PEROPTN	000001	DMKPEI									
PEROP1	000032	DMKPER	DMKPET								
PEROP2	000007	DMKPER	DMKPET								
PERPASCT	000005	DMKPEI	DMKPEL								
PERPASSP	000004	DMKPEI	DMKPEL								
PERPUSED	000016	DMKPEI									
PERREGSV	000002	DMKPEI	DMKPEN								
PERRNGTB	000006	DMKPEI	DMKPEL								
PERSALT	000011	DMKPRV	DMKPRW	DMKTMR							
PERSAVED	000015	DMKPEN	DMKPEQ	DMKPET							
PERSCADD	000001	DMKPEI									
PERSCAN	000013	DMKPEI	DMKPEN								
PERSCDLN	000006	DMKPEN									
PERSEQP	000001	DMKPET									
PERSEQT	000001	DMKPET									
PERSIZE	000004	DMKPEI	DMKPEN								
PERSTLEN	000002	DMKPER									

LABEL	COUNT	REFERENCES					
PERSTPCT	000007	DMKPEI	DMKPEL				
PERSTPSP	000011	DMKPEI	DMKPEL				
PERTBAK	000006	DMKPEI	DMKPEN	DMKPER	DMKPET		
PERTBLEN	000005	DMKPEI	DMKPEN	DMKPER			
PERTMPCH	000011	DMKPEI	DMKPEL	DMKPEN			
PERTOTAL	000020	DMKPEI	DMKPEL	DMKPEN			
PERWKFLG	000048	DMKPEI	DMKPEL	DMKPEN			
PERWKFL2	000023	DMKPEI	DMKPEN				
PERWKLEN	000001	DMKPEI					
PERWORK	000002	DMKPEI					
PERWRKCT	000021	DMKPEI	DMKPEL	DMKPEN			
PESBLOK	000015	DMKPEN	DMKPEQ				
PESCHAIN	000009	DMKPEN	DMKPEQ				
PESCOUNT	000010	DMKPEN					
PESELIM	000003	DMKPEN					
PESFLAG	000003	DMKPEN					
PESNAME	000005	DMKPEN	DMKPEQ				
PESNEXT	000015	DMKPEN	DMKPEQ				
PESSIZE	000005	DMKPEN					
PEXBLOK	000053	DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPER	DMKPET
PEXBR	000009	DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPER	DMKPET
PEXCHCMP	000002	DMKPEL					
PEXCMND	000012	DMKPEL	DMKPEN	DMKPEQ	DMKPET		
PEXDATA	000020	DMKPEI	DMKPEL	DMKPEQ	DMKPER	DMKPET	
PEXDINV	000006	DMKPEI	DMKPER				
PEXDLEN	000018	DMKPEI	DMKPEL	DMKPEQ	DMKPER	DMKPET	
PEXELIM	000007	DMKPEI	DMKPEL	DMKPEN			
PEXFLAGO	000033	DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPER	DMKPET
PEXFLAGT	000025	DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPER	DMKPET
PEXFROM	000013	DMKPEL	DMKPEQ	DMKPER			
PEXFROM2	000002	DMKPEI					
PEXGPR	000012	DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPER	DMKPET
PEXGREG	000019	DMKPEI	DMKPEL	DMKPEQ	DMKPER		
PEXGSUC	000002	DMKPER	DMKPET				
PEXINST	000011	DMKPEI	DMKPEL	DMKPEN	DMKPER		
PEXINTO	000015	DMKPEI	DMKPEL	DMKPEQ	DMKPER	DMKPET	
PEXINTO1	000004	DMKPEI	DMKPER				
PEXLEN	000007	DMKPEI	DMKPEL	DMKPEN			
PEXMASK	000007	DMKPEI	DMKPEL	DMKPEN	DMKPER	DMKPET	
PEXMECMP	000002	DMKPEL					
PEXNEXT	000044	DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPER	DMKPET
PEXPASS	000007	DMKPEL	DMKPEQ	DMKPER			
PEXPASSN	000003	DMKPER					
PEXPRINT	000008	DMKPEI	DMKPEL	DMKPET			
PEXRANGE	000002	DMKPEL					
PEXRUN	000013	DMKPEI	DMKPEL	DMKPEQ	DMKPET		
PEXSECND	000008	DMKPEI	DMKPEL	DMKPEQ	DMKPER		
PEXSIZE	000003	DMKPEI					
PEXSTEP	000010	DMKPEL	DMKPEQ	DMKPET			
PEXSTEPN	000002	DMKPET					
PEXSTORE	000013	DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPER	DMKPET
PEXSUCC	000005	DMKPER	DMKPET				
PEXTERM	000012	DMKPEI	DMKPEL	DMKPER	DMKPET		
PEXTHIRD	000008	DMKPEI	DMKPEL	DMKPEQ	DMKPER		
PFDATA	000023	DMKCFY	DMKCQY	DMKGRG	DMKGRT	DMKRGC	DMKTTX	DMKVCU
PFDCMD	000011	DMKCFY	DMKCQY	DMKGRG	DMKRGC	DMKVCU	
PFDCPAD4	000006	DMKGRG	DMKRGC				
PFDCPYAD	000001	DMKVCU					

LABEL	COUNT	REFERENCES									
PFDCPYSP	000003	DMKCFY	DMKGRG	DMKRGC							
PFDTEXT	000002	DMKTTX									
PFDVAL	000006	DMKCFY	DMKCQY	DMKGRT	DMKVCU						
PFKADDR	000013	DMKCFY	DMKCQY	DMKGRG	DMKGRT	DMKRGC	DMKTTX	DMKUSQ	DMKVCU		
PFKCOUNT	000005	DMKCFC	DMKCFY	DMKCQY	DMKUSQ						
PFKDWDS	000003	DMKCFY	DMKUSQ								
PFKFLAG	000014	DMKCFY	DMKCQY	DMKGRG	DMKRGC	DMKTTX	DMKVCU				
PFKIMM	000007	DMKCFY	DMKCQY	DMKGRG	DMKRGC	DMKTTX	DMKVCU				
PFKLNG	000015	DMKCFY	DMKCQY	DMKGRG	DMKGRT	DMKRGC	DMKTTX	DMKVCU			
PFKRET	000007	DMKCFY	DMKCQY	DMKGRG	DMKRGC	DMKTTX	DMKVCU				
PFKSIZE	000003	DMKCFY	DMKUSQ								
PFKTABLE	000015	DMKCFY	DMKCQY	DMKGRG	DMKGRT	DMKRGC	DMKTTX	DMKUSQ	DMKVCU		
PFKTBLSZ	000004	DMKCFY	DMKUSQ								
PFTAB	000025	DMKCFY	DMKCQY	DMKGRT	DMKVCU						
PFTABS	000003	DMKCQY	DMKGRT	DMKVCU							
PGADDR	000002	DMKDSP	DMKVAT								
PGBLOK	000003	DMKCFP	DMKDSP	DMKVAT							
PGBSIZE	000003	DMKCFP	DMKDSP	DMKVAT							
PGIDADDR	000002	DMKMNT									
PGIDCPU	000001	DMKMNT									
PGIDMOD	000001	DMKMNT									
PGIDTOD	000002	DMKMNT									
PGPNT	000003	DMKCFP	DMKDSP	DMKVAT							
PGREAD	000015	DMKAPI	DMKCPI	DMKMON	DMKMOO	DMKPGM	DMKPTR	DMKSTP	DMKSTR		
PGSRATIO	000004	DMKCPP	DMKSTP	DMKTHI							
PGWAITIM	000007	DMKCPU	DMKDSP	DMKSTP							
PGWRITE	000009	DMKMON	DMKMOO	DMKPGM	DMKSEL	DMKSTP					
PIB	000001	DMKPIB									
PLALARM	000004	DMKQCN	DMKQCQ								
PLDFRET	000007	DMKQCN	DMKQCQ	DMKVDD							
PLDIAG	000004	DMKQCN	DMKQCQ								
PLEDIT	000001	DMKQCO									
PLERRMSG	000004	DMKQCN									
PLFLAG1	000038	DMKCSU	DMKCSV	DMKCSX	DMKDEF	DMKDEG	DMKERM	DMKQCN	DMKQCO	DMKQCQ	DMKSPL
		DMKTRT	DMKVCN	DMKVDD							
PLFRET	000017	DMKCSU	DMKCSV	DMKCSX	DMKDEF	DMKERM	DMKQCN	DMKQCO	DMKSPL	DMKTRT	DMKVDD
PLHILITE	000008	DMKQCN	DMKQCQ								
PLIMSG	000019	DMKCSU	DMKCSV	DMKCSX	DMKDEF	DMKDEG	DMKQCN	DMKSPL	DMKTRT	DMKVDD	
PLINHIBT	000008	DMKQCN	DMKQCO								
PLIST	000072	DMKCSU	DMKCSV	DMKCSX	DMKDEF	DMKDEG	DMKERM	DMKQCN	DMKQCO	DMKQCQ	DMKSPL
		DMKTRT	DMKVCN	DMKVDD							
PLLED	000006	DMKERM	DMKQCN	DMKQCQ	DMKVCN						
PLLOGDRP	000007	DMKQCN	DMKQCO	DMKQCQ							
PLLOGHLD	000007	DMKQCN	DMKQCO	DMKQCQ							
PLNCB	000007	DMKQCN	DMKQCO								
PLNOAUTO	000002	DMKQCN	DMKQCQ								
PLNORESP	000003	DMKQCN	DMKQCO	DMKQCQ							
PLNORET	000012	DMKCSU	DMKDEF	DMKDEG	DMKQCN	DMKQCQ	DMKTRT	DMKVDD			
PLNOTIME	000009	DMKQCN									
PLOPERTR	000005	DMKQCN	DMKQCO	DMKQCQ							
PLPRIOR	000004	DMKQCN	DMKQCO	DMKQCQ							
PLR2	000037	DMKCSU	DMKCSV	DMKCSX	DMKDEF	DMKDEG	DMKERM	DMKQCN	DMKQCO	DMKSPL	DMKTRT
		DMKVCN	DMKVDD								
PLSECUSR	000006	DMKQCN									
PLSIZE	000039	DMKCSU	DMKCSV	DMKCSX	DMKDEF	DMKDEG	DMKERM	DMKQCN	DMKQCO	DMKSPL	DMKTRT
		DMKVDD									
PLSVR2	000017	DMKQCN	DMKQCO	DMKQCQ							
PLSVR20	000005	DMKQCN	DMKQCQ								

LABEL	COUNT	REFERENCES									
PLSVR21	000011	DMKQCN	DMKQCO								
PLSVR22	000027	DMKQCN	DMKQCO	DMKQCQ							
PLSVR23	000010	DMKQCN	DMKQCO								
PLUCASE	000001	DMKQCO									
PLVIRDVD	000011	DMKQCN	DMKQCO	DMKQCQ							
PLVMGNIO	000015	DMKQCN	DMKQCO	DMKQCQ							
PLWRTRD	000003	DMKQCO									
PMAAVAIL	000026	DMKAPI	DMKCFG	DMKCPP	DMKCPS	DMKCPU	DMKDMP	DMKDSP	DMKFRE	DMKHVD	DMKPMA
		DMKQVM	DMKSAD	DMKSEG	DMKSPM	DMKVRR					
PMAGUEST	000004	DMKEXT	DMKPMA	DMKPRG							
PMAMODE	000041	DMKCCH	DMKCFG	DMKCFQ	DMKCKP	DMKCPO	DMKCPP	DMKCPU	DMKDID	DMKDMP	DMKEXT
		DMKIOT	DMKMCH	DMKMCT	DMKMHC	DMKPMA	DMKQVM	DMKSPM	DMKVRR		
PMAON	000012	DMKCFG	DMKMCH	DMKMCT	DMKPMA	DMKQVM	DMKSAD	DMKVRR			
PMASTAT	000073	DMKAPI	DMKCCH	DMKCFG	DMKCFQ	DMKCKP	DMKCPO	DMKCPP	DMKCPS	DMKCPU	DMKDID
		DMKDMP	DMKDSP	DMKEXT	DMKFRE	DMKHVD	DMKIOT	DMKMCH	DMKMCT	DMKMHC	DMKPMA
		DMKPRG	DMKQVM	DMKSAD	DMKSEG	DMKSPM	DMKVRR				
PMSGLIM	000004	DMKCRM	DMKIDR								
PNTR	000023	DMKFRE									
POFFLINE	000027	DMKACR	DMKAPI	DMKCPO	DMKCPP	DMKCPU	DMKENT	DMKIOG	DMKMCT	DMKVCH	
POWEROFF	000009	DMKCKH	DMKCKP	DMKCPJ	DMKCPS	DMKOPE	DMKTOD				
PPMAP	000002	DMKHVD									
PPMBSEPP	000001	DMKHVD									
PPMHPO	000008	DMKHVD									
PPMHPO1	000001	DMKHVD									
PPMHPO2	000001	DMKHVD									
PPMHPO25	000001	DMKHVD									
PPMHPO3	000001	DMKHVD									
PPMHPO32	000001	DMKHVD									
PPMHPO34	000001	DMKHVD									
PPMHPO36	000001	DMKHVD									
PPMHPO4	000001	DMKHVD									
PPMHPO42	000001	DMKHVD									
PPMHPO5	000001	DMKHVD									
PPMHP2	000005	DMKHVD									
PPMM21	000001	DMKHVD									
PPMPP	000007	DMKHVD									
PPMSCH	000001	DMKHVD									
PPMSEPP	000001	DMKHVD									
PPMVDLE	000001	DMKHVD									
PPMVMSP1	000001	DMKHVD									
PPMVMSP2	000001	DMKHVD									
PPMVMSP3	000001	DMKHVD									
PPMVMSP4	000001	DMKHVD									
PPMVMSP5	000001	DMKHVD									
PRECCW	000022	DMKCCD	DMKCCO	DMKCCS	DMKCCW						
PRECTL	000002	DMKCCD	DMKCCO								
PREFIXA	000264	DMKACS	DMKAPI	DMKATS	DMKBLD	DMKCCH	DMKCDB	DMKCDM	DMKCDS	DMKCFF	DMKCFG
		DMKCFS	DMKCFY	DMKCKD	DMKCKF	DMKCNS	DMKCPI	DMKCPJ	DMKCPM	DMKCPN	DMKCPP
		DMKCPS	DMKCPU	DMKCPW	DMKCPY	DMKCPZ	DMKCQC	DMKCQS	DMKCQT	DMKCQY	DMKDSP
		DMKENT	DMKEXT	DMKFRE	DMKFRT	DMKGRF	DMKHVD	DMKHVF	DMKIOF	DMKIOQ	DMKIOS
		DMKIOT	DMKIUA	DMKIUC	DMKIUJ	DMKIUN	DMKIUP	DMKLOH	DMKLOJ	DMKLOK	DMKMCC
		DMKMCH	DMKMCT	DMKMHC	DMKMIA	DMKMID	DMKMNI	DMKMNT	DMKMON	DMKOPE	DMKPAG
		DMKPMA	DMKPRG	DMKPSA	DMKPTR	DMKPTS	DMKPTT	DMKRGC	DMKRNH	DMKSAD	DMKSCH
		DMKSEL	DMKSPM	DMKSTA	DMKSTP	DMKSVC	DMKSVD	DMKSWA	DMKTHI	DMKTOD	DMKTRP
		DMKTRT	DMKTRU	DMKTRX	DMKVBM	DMKVCT	DMKVCV	DMKVCW	DMKVCX	DMKVIO	DMKVMG
		DMKVRR	DMKVSJ	DMKWAI							
PREFIXB	000212	DMKACO	DMKACR	DMKACS	DMKAPI	DMKBLD	DMKCCH	DMKCDB	DMKCDM	DMKCDS	DMKCFG
		DMKCFS	DMKCFV	DMKCFY	DMKCKD	DMKCKF	DMKCPI	DMKCPJ	DMKCPM	DMKCPO	DMKCPP

LABEL	COUNT	REFERENCES									
		DMKCPS	DMKCPU	DMKCPY	DMKCQU	DMKCQY	DMKDSP	DMKENT	DMKEXT	DMKFPS	DMKFRE
		DMKFRT	DMKHVD	DMKIOG	DMKIOJ	DMKIOQ	DMKIOS	DMKIUP	DMKLOH	DMKLOJ	DMKLOK
		DMKMCC	DMKMCD	DMKMCH	DMKMCT	DMKMIA	DMKMID	DMKMNI	DMKMNT	DMKMON	DMKMOO
		DMKMPO	DMKPAG	DMKPGS	DMKPMA	DMKPRG	DMKPRV	DMKPRW	DMKPSA	DMKPST	DMKPTS
		DMKRPA	DMKSAD	DMKSCH	DMKSEL	DMKSPM	DMKSSV	DMKSTK	DMKSTP	DMKSVC	DMKTHI
		DMKTOD	DMKTRP	DMKTRQ	DMKTRT	DMKTRU	DMKVAT	DMKVAU	DMKVCH	DMKVDC	DMKVER
		DMKVFC	DMKVFD	DMKVFE	DMKVFR	DMKVFS	DMKVME	DMKVRR	DMKVSJ		
PREFIXVL	000003	DMKSAD									
PREFLAG	000001	DMKCCD									
PREFRG	000011	DMKDMQ									
PREPRD	000002	DMKCCO	DMKCCW								
PREVRCW	000002	DMKCCS	DMKCCW								
PRGC	000032	DMKBSC	DMKCNS	DMKDAD	DMKDAS	DMKDAU	DMKDIB	DMKGRF	DMKHPS	DMKHVC	DMKIOS
		DMKIOT	DMKRNH	DMKRSE	DMKSAD	DMKTAP	DMKTAQ	DMKTPE	DMKUNT	DMKVCA	DMKVCN
		DMKVSP	DMKVSX								
PRIMEHDR	000024	DMKAPI	DMKCPP	DMKFRE	DMKFRT	DMKMOO	DMKSTA	DMKSVC			
PRIMEHI	000010	DMKAPI	DMKCPP	DMKFRE	DMKFRT	DMKSTA	DMKSVC				
PRIMELO	000009	DMKAPI	DMKCPP	DMKFRE	DMKFRT	DMKSTA	DMKSVC				
PRIORITY	000042	DMKACO	DMKCNS	DMKCPO	DMKCPS	DMKCQR	DMKDAS	DMKDIB	DMKGRD	DMKGRI	DMKMCT
		DMKMSG	DMKRGB	DMKRNH	DMKTTX	DMKUSQ	DMKVCN	DMKVCS			
PRNPSW	000140	DMKAPI	DMKCKD	DMKCKP	DMKCPI	DMKDMP	DMKDSP	DMKPGS	DMKPMA	DMKPRG	DMKPRV
		DMKPRW	DMKSAD	DMKSAV	DMKSSP	DMKSTA	DMKTOD	DMKVRR	DMKVRS		
PROBMODE	000027	DMKDSP	DMKFPS	DMKIUA	DMKMCH	DMKMPO	DMKPRG	DMKPRV	DMKSVC	DMKSVD	DMKTEE
		DMKTEM									
PROBSTRT	000016	DMKCFP	DMKDSP	DMKSCH	DMKTMR						
PROBTIME	000012	DMKAPI	DMKCFP	DMKCPI	DMKDSP	DMKMON	DMKSCH	DMKTMR			
PROB370E	000005	DMKDSP	DMKPRG								
PROCIO	000058	DMKAPI	DMKCCH	DMKCDB	DMKCDM	DMKCDS	DMKCFG	DMKCFP	DMKCPI	DMKCPJ	DMKCPP
		DMKCPU	DMKCQP	DMKCSF	DMKDAU	DMKDSP	DMKENT	DMKEXT	DMKFRE	DMKHVD	DMKIOT
		DMKMCH	DMKMCT	DMKMHC	DMKMNI	DMKMON	DMKMOO	DMKPMA	DMKPRV	DMKSCN	DMKTHI
		DMKVCH	DMKVDC	DMKVRR							
PROCIPL	000102	DMKACS	DMKAPI	DMKATS	DMKCCH	DMKCDB	DMKCDM	DMKCDS	DMKCFF	DMKCKD	DMKCKM
		DMKCLK	DMKCPI	DMKCPJ	DMKCPP	DMKCPS	DMKCPU	DMKCPZ	DMKCQP	DMKCQY	DMKDID
		DMKDMQ	DMKDSB	DMKDSP	DMKENT	DMKEXT	DMKHVD	DMKIOG	DMKIOQ	DMKLOH	DMKMCH
		DMKMNI	DMKMNT	DMKMON	DMKMOO	DMKMPO	DMKPGS	DMKPMA	DMKPRV	DMKSCN	DMKSEL
		DMKSTP	DMKSVC	DMKTHI	DMKTOD	DMKVER	DMKVFC	DMKVFR	DMKVMA	DMKVRR	
PROCSCHK	000003	DMKCLK									
PROPSW	000080	DMKCKP	DMKDMP	DMKDSP	DMKFMT	DMKFPS	DMKIUA	DMKMON	DMKMPO	DMKPMA	DMKPRG
		DMKPRV	DMKSAD	DMKSTA							
PROTATRQ	000004	DMKCFY	DMKREI								
PROTBEG	000003	DMKREI	DMKVMA								
PROTBLOK	000018	DMKCFG	DMKCFY	DMKDSP	DMKHVE	DMKLOH	DMKPRG	DMKPTR	DMKREI	DMKVMA	
PROTBUFF	000005	DMKCFG	DMKHVE	DMKREI							
PROTCAW	000002	DMKCCW									
PROTDPSW	000001	DMKDSP									
PROTERR	000032	DMKCFG	DMKDSP	DMKHVE	DMKPRG	DMKPTR	DMKREI	DMKVMA			
PROTEXTL	000001	DMKDSP									
PROTFLAG	000010	DMKDSP	DMKPRG	DMKPTR	DMKREI	DMKVMA					
PROTPAGE	000002	DMKHVE	DMKPTR								
PROTPALT	000002	DMKHVE	DMKVMA								
PROTPGAD	000001	DMKVMA									
PROTPRGL	000001	DMKPRG									
PROTPSW	000003	DMKDSP	DMKPRG								
PROTRCNT	000008	DMKDSP	DMKPRG	DMKPTR	DMKREI	DMKVMA					
PROTREI	000008	DMKDSP	DMKPRG	DMKPTR	DMKREI	DMKVMA					
PROTREIL	000006	DMKDSP	DMKPRG	DMKPTR	DMKVMA						
PROTSIZE	000005	DMKCFY	DMKLOH								
PROTSYSN	000001	DMKVMA									

LABEL	COUNT	REFERENCES									
PROTTRAN	000002	DMKHVE	DMKPRG								
PRTC	000027	DMKBSC	DMKCNS	DMKDAD	DMKDAS	DMKDAU	DMKDIB	DMKGRF	DMKHVC	DMKIOS	DMKIOT
		DMKRNH	DMKRSE	DMKSAD	DMKTAP	DMKTAQ	DMKTPE	DMKUNT	DMKVCA	DMKVCN	DMKVSP
PRTCHN	000010	DMKCKR	DMKCKS	DMKCKV	DMKCSY	DMKSPL	DMKSPS	DMKVST	DMKWRM		
PRVCOMND	000010	DMKCCW									
PRVFLAG	000011	DMKCCD	DMKCCF	DMKCCS	DMKCCW						
PSA	001180	DMKACO	DMKACR	DMKACS	DMKALG	DMKALO	DMKAPI	DMKAPS	DMKAPT	DMKAPU	DMKAPV
		DMKAPW	DMKAPX	DMKAPY	DMKAPZ	DMKATS	DMKBIO	DMKBLD	DMKBSC	DMKCAC	DMKCAO
		DMKCCD	DMKCCF	DMKCCH	DMKCCO	DMKCCS	DMKCCT	DMKCCW	DMKCDB	DMKCDM	DMKCDS
		DMKCFC	DMKCFD	DMKCFF	DMKCFG	DMKCFH	DMKCFJ	DMKCFM	DMKCFO	DMKCFP	DMKCFQ
		DMKCFR	DMKCFS	DMKCFT	DMKCFU	DMKCFV	DMKCFW	DMKCFY	DMKCKD	DMKCKF	DMKCKH
		DMKCKM	DMKCKN	DMKCKP	DMKCKR	DMKCKS	DMKCKT	DMKCKV	DMKCKW	DMKCLK	DMKCNS
		DMKCNT	DMKCPB	DMKCPI	DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT
		DMKCPU	DMKCPV	DMKCPW	DMKCPX	DMKCPY	DMKCPZ	DMKCQC	DMKCQG	DMKCQH	DMKCQI
		DMKCQP	DMKCQQ	DMKCQR	DMKCQS	DMKCQT	DMKCQU	DMKCQY	DMKCRM	DMKCSB	DMKCSC
		DMKCSF	DMKCSO	DMKCSP	DMKCSQ	DMKCSR	DMKCST	DMKCSU	DMKCSV	DMKCSW	DMKCSX
		DMKCSY	DMKCVT	DMKCVU	DMKDAD	DMKDAS	DMKDAU	DMKDEF	DMKDEG	DMKDEI	DMKDEX
		DMKDGD	DMKDGF	DMKDIA	DMKDIB	DMKDID	DMKDIF	DMKDMP	DMKDMQ	DMKDRD	DMKDRE
		DMKDSB	DMKDSP	DMKEIG	DMKENT	DMKEPS	DMKERM	DMKERP	DMKEXT	DMKFMT	DMKFPS
		DMKFRE	DMKFRT	DMKGIO	DMKGRA	DMKGRC	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRH
		DMKGRI	DMKGRT	DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKIDR
		DMKIDU	DMKIOC	DMKIOE	DMKIOF	DMKIOG	DMKIOH	DMKIOJ	DMKIOQ	DMKIOS	DMKIOT
		DMKISM	DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUP
		DMKIUS	DMKJRL	DMKLNK	DMKLNM	DMKLOC	DMKLOG	DMKLOH	DMKLOJ	DMKLOK	DMKLOM
		DMKMCC	DMKMCD	DMKMCH	DMKMCI	DMKMCT	DMKMHC	DMKMHV	DMKMIA	DMKMID	DMKMNI
		DMKMNJ	DMKMNL	DMKMNT	DMKMON	DMKMOO	DMKMPO	DMKMSG	DMKMSW	DMKNEA	DMKNES
		DMKNET	DMKNLD	DMKNLE	DMKOPE	DMKOPR	DMKPAG	DMKPAH	DMKPEI	DMKPEL	DMKPEN
		DMKPEQ	DMKPER	DMKPET	DMKPGM	DMKPGS	DMKPGT	DMKPGU	DMKPMA	DMKPRG	DMKPRV
		DMKPRW	DMKPSA	DMKPST	DMKPTR	DMKPTS	DMKPTT	DMKQCN	DMKQCO	DMKQCP	DMKQCQ
		DMKQVM	DMKREI	DMKRET	DMKRGA	DMKRGB	DMKRGC	DMKRGD	DMKRGE	DMKRNH	DMKRPA
		DMKRPD	DMKRSE	DMKRSF	DMKRSP	DMKRSQ	DMKRST	DMKSAD	DMKSAV	DMKSBL	DMKSCH
		DMKSCN	DMKSCO	DMKSEG	DMKSEL	DMKSEP	DMKSEV	DMKSFB	DMKSIX	DMKSNC	DMKSND
		DMKSPK	DMKSPL	DMKSPM	DMKSPR	DMKSPS	DMKSPT	DMKSRM	DMKSSP	DMKSSS	DMKSST
		DMKSSU	DMKSSV	DMKSTA	DMKSTK	DMKSTP	DMKSTR	DMKSVC	DMKSVD	DMKSWA	DMKSWM
		DMKTAP	DMKTAQ	DMKTCS	DMKTCT	DMKTDK	DMKTHI	DMKTMR	DMKTOD	DMKTPE	DMKTRA
		DMKTRC	DMKTRD	DMKTRK	DMKTRM	DMKTRP	DMKTRQ	DMKTRT	DMKTRU	DMKTRX	DMKTTX
		DMKTTY	DMKUDR	DMKUDU	DMKUNT	DMKURS	DMKUSP	DMKUSQ	DMKVAT	DMKVAU	DMKVBM
		DMKVCA	DMKVCB	DMKVCH	DMKVCN	DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCU
		DMKVCV	DMKVCW	DMKVCX	DMKVDA	DMKVDB	DMKVDC	DMKVDD	DMKVDE	DMKVDF	DMKVDG
		DMKVDH	DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVFC	DMKVFD	DMKVFE	DMKVFR	DMKVFS
		DMKVIO	DMKVMA	DMKVMC	DMKVMD	DMKVME	DMKVMG	DMKVMI	DMKVRR	DMKVRS	DMKVSC
		DMKVSD	DMKVSE	DMKVSF	DMKVSG	DMKVSI	DMKVSJ	DMKVSP	DMKVSQ	DMKVSR	DMKVST
		DMKVSU	DMKVSV	DMKVSW	DMKVSX	DMKWAI	DMKWRM	DMKWRN	DMKXAB	DMKXAD	DMKXST
		DMKZTD									
PSACPXBP	000014	DMKCPI	DMKCPP	DMKCPU	DMKSVC						
PSADSPRQ	000013	DMKAPI	DMKCPI	DMKCPP	DMKDSP	DMKSTK	DMKVSJ				
PSAEVMA	000006	DMKCFS	DMKCFY	DMKCPI	DMKLOJ						
PSAEXT	000042	DMKAPI	DMKCPP	DMKDSP	DMKFRE	DMKFRT	DMKPAG	DMKPGM	DMKPSA	DMKPTR	DMKSCH
		DMKSEL	DMKSTK	DMKSWA	DMKSWM	DMKTRQ	DMKWAI				
PSAEXTX	000020	DMKAPI	DMKCPP	DMKDSP	DMKFRE	DMKSCH	DMKSTK	DMKTRQ			
PSAIOSW	000005	DMKIOS	DMKIOT								
PSALANG	000013	DMKBLD	DMKCPI	DMKCPP	DMKHVF	DMKLOH	DMKOPE	DMKVBM			
PSAMSS	000022	DMKCFG	DMKCFQ	DMKCPI	DMKCPW	DMKDEI	DMKDGD	DMKHVE	DMKLNK	DMKLOJ	DMKSST
		DMKSSV	DMKUSQ	DMKVDA	DMKVDR	DMKVSJ					
PSAPGID	000015	DMKACS	DMKCKP	DMKCPI	DMKCPM	DMKCPN	DMKCPP	DMKCPW	DMKMNT	DMKTPE	
PSASVCCT	000004	DMKMOO	DMKSVD								
PSASYSID	000009	DMKCNS	DMKCPI	DMKCQY	DMKOPE						
PSBCLR2	000002	DMKCPI									

LABEL	COUNT	REFERENCES									
PSECLR2	000001	DMKCPI									
PSENDCLR	000003	DMKAPI	DMKCPI								
PSTARTSV	000008	DMKSAV									
PSTPINPP	000008	DMKPAG	DMKPGM	DMKPTR	DMKPXA	DMKPXB					
PSTPINSW	000008	DMKPGM	DMKPTR	DMKPXA	DMKPXB	DMKSWM					
PSTPOUPP	000006	DMKPAG	DMKPXA	DMKPXB	DMKSEL						
PSTPOUSW	000004	DMKPXA	DMKPXB	DMKSWA							
PSTRCBYT	000008	DMKPST	DMKXST								
PSTRCPAG	000004	DMKPST	DMKXST								
PSTRCPG$	000006	DMKPGT	DMKPGU	DMKPST	DMKVDH						
PSW	000002	DMKLD00E									
PSWCC1	000001	DMKCFG									
PSWCC2	000003	DMKCFF	DMKCFG	DMKHVF							
PSWCC3	000001	DMKHVF									
PTRCACH	000006	DMKMNL									
PTRDVCT	000005	DMKMNL									
PTRENLN	000009	DMKMNL									
PTRLIST	000004	DMKMNL									
PTRNTRY	000013	DMKMNL									
PTRRDEV	000004	DMKMNL									
PTRSIZE	000002	DMKMNL									
PURGESTO	000014	DMKFPS	DMKVAT								
PWDALOG	000001	DMKJRL									
PWDCHAIN	000006	DMKGRT	DMKJRL								
PWDDATE	000002	DMKJRL									
PWDFLAGS	000002	DMKJRL									
PWDIBLOK	000004	DMKGRT	DMKJRL								
PWDINVCT	000007	DMKJRL									
PWDITRQ	000007	DMKGRT	DMKJRL								
PWDLGCNT	000006	DMKGRT	DMKJRL								
PWDLOG	000002	DMKJRL									
PWDLUNAM	000002	DMKJRL									
PWDSIZE	000003	DMKJRL									
PWDTERMA	000002	DMKJRL									
PWDUSRID	000002	DMKJRL									
PWDVSMNM	000002	DMKJRL									
PWTPAGES	000005	DMKCPU	DMKDSP								
PXA	000070	DMKAPI	DMKCPP	DMKDSP	DMKFRE	DMKFRT	DMKMOO	DMKPAG	DMKPGM	DMKPTR	DMKSCH
		DMKSEL	DMKSTK	DMKSWA	DMKSWM	DMKTRQ	DMKWAI				
PXAEND	000002	DMKPXA	DMKPXB								
QDISK	000001	DMKSAV									
QUANTUM	000017	DMKDSP	DMKEXT	DMKFPS	DMKPRG	DMKQVM	DMKSVD				
QUANTUMR	000017	DMKDSP	DMKEXT	DMKFPS	DMKIOT	DMKMCH	DMKMPO	DMKPMA	DMKPRG	DMKQVM	DMKSVD
Q1DROP	000006	DMKMON	DMKSCH								
RANGE	000121	DMKCAC	DMKCAO	DMKCDB	DMKCDM	DMKCPN	DMKCPT	DMKCQG	DMKCQP	DMKCQQ	DMKCQT
		DMKFMT	DMKMCD	DMKPEI	DMKPEL	DMKPEQ	DMKVDA	DMKVDC	DMKVDD	DMKVDE	DMKVDF
RAPF	000002	DMKPRG									
RB	000003	DMKGRD	DMKGRE	DMKVCR							
RCADDRCK	000002	DMKIUE	DMKIUL								
RCANSBND	000003	DMKIUL	DMKIUN	DMKIUS							
RCAPPC	000007	DMKIUE	DMKIUG	DMKIUL	DMKIUN	DMKIUP					
RCBADDIR	000002	DMKIUC									
RCBADFCN	000001	DMKIUC									
RCBADLIM	000002	DMKIUC									
RCBUFBND	000003	DMKIUE	DMKIUN	DMKIUS							
RCCOMSRV	000003	DMKIUC	DMKIUJ	DMKIUS							
RCEXTLEN	000001	DMKIUC									
RCHADD	000041	DMKACR	DMKACS	DMKCCH	DMKCFP	DMKCKD	DMKCKH	DMKCKN	DMKCPN	DMKDID	DMKDMQ

LABEL	COUNT	REFERENCES									
		DMKDSB	DMKENT	DMKIOQ	DMKIOS	DMKMNI	DMKMNL	DMKMNT	DMKMON	DMKMOO	DMKQVM
		DMKSCN	DMKSSS								
RCHASBFR	000001	DMKIUP									
RCHBLOK	000081	DMKACR	DMKACS	DMKCCH	DMKCFP	DMKCFU	DMKCKD	DMKCKH	DMKCKM	DMKCKN	DMKCNS
		DMKCPM	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPV	DMKCPW	DMKCPZ	DMKCQP
		DMKCQQ	DMKDID	DMKDIF	DMKDMQ	DMKDSB	DMKENT	DMKEXT	DMKIOG	DMKIOQ	DMKIOS
		DMKIOT	DMKMNI	DMKMNL	DMKMNT	DMKMON	DMKMOO	DMKNES	DMKPRV	DMKQVM	DMKSCN
		DMKSCO	DMKSSP	DMKSSS	DMKURS	DMKVCH	DMKVDA	DMKVDT			
RCHBMX	000002	DMKIOS	DMKQVM								
RCHBUSY	000019	DMKACS	DMKCNS	DMKIOQ	DMKIOS	DMKIOT	DMKQVM				
RCHCUTBL	000035	DMKACR	DMKACS	DMKCCH	DMKCKD	DMKCKM	DMKCPM	DMKCPO	DMKCPP	DMKCPS	DMKCPV
		DMKCQP	DMKDIF	DMKIOQ	DMKIOS	DMKMNI	DMKMNT	DMKNES	DMKSCN	DMKSSP	DMKVCH
RCHDED	000007	DMKCCH	DMKCFP	DMKCPN	DMKVCH	DMKVDA					
RCHDISA	000011	DMKACR	DMKACS	DMKCPW	DMKIOG	DMKPRV	DMKVCH				
RCHFIOB	000009	DMKACS	DMKIOQ	DMKIOS							
RCHMPX	000006	DMKIOQ	DMKIOS								
RCHPEND	000004	DMKCPM	DMKCPO	DMKIOQ	DMKIOS						
RCHPROC	000021	DMKCPM	DMKCPN	DMKCPP	DMKCPW	DMKCPZ	DMKCQP	DMKCQQ	DMKEXT	DMKIOQ	DMKIOS
		DMKMNL	DMKMNT	DMKMON	DMKURS	DMKVCH					
RCHQCNT	000012	DMKENT	DMKIOQ	DMKIOS	DMKMON						
RCHRSTQ	000007	DMKIOQ	DMKIOS								
RCHSCED	000002	DMKIOS	DMKQVM								
RCHSEL	000003	DMKIOS									
RCHSIZE	000002	DMKQVM	DMKSSP								
RCHSTAT	000045	DMKACR	DMKACS	DMKCCH	DMKCFP	DMKCNS	DMKCPM	DMKCPN	DMKCPO	DMKCPW	DMKIOG
		DMKIOQ	DMKIOS	DMKIOT	DMKPRV	DMKQVM	DMKVCH	DMKVDA			
RCHSTIDC	000006	DMKCCH	DMKCPN	DMKDID	DMKIOG	DMKPRV					
RCHTYPE	000014	DMKIOG	DMKIOQ	DMKIOS	DMKQVM						
RCH370	000003	DMKIOG	DMKIOS								
RCIINVLN	000001	DMKIUL									
RCINVCNF	000002	DMKIUJ	DMKIUS								
RCINVCON	000003	DMKIUJ	DMKIUS								
RCINVLUN	000001	DMKIUC									
RCINVMOD	000001	DMKIUC									
RCINVREC	000002	DMKIUJ	DMKIUS								
RCINVSCD	000001	DMKIUJ									
RCINVSEV	000002	DMKIUJ	DMKIUS								
RCINVSND	000001	DMKIUS									
RCINVSRV	000001	DMKIUC									
RCMSGCT	000005	DMKIUJ	DMKIUN	DMKIUS	DMKVCV	DMKVCX					
RCMSGLEN	000005	DMKIUE	DMKIUL	DMKIUN	DMKIUS						
RCNEGLEN	000002	DMKIUE	DMKIUL								
RCNLOG	000004	DMKIUC									
RCNODATA	000005	DMKIUE	DMKIUG	DMKIUL							
RCNONAPP	000002	DMKIUJ	DMKIUS								
RCNOPATH	000012	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS		
RCNOPRMD	000002	DMKIUL	DMKIUN								
RCNOPRTY	000001	DMKIUN									
RCNOSEND	000002	DMKIUN	DMKRPI								
RCNTRGIU	000003	DMKIUC									
RCPRMLST	000002	DMKIUL	DMKIUN								
RCPROTCK	000002	DMKIUE	DMKIUL								
RCPTHSVD	000002	DMKIUC									
RCPURGED	000004	DMKIUE	DMKIUL								
RCRECVSH	000002	DMKIUE	DMKIUL								
RCSNDLST	000004	DMKIUE	DMKIUL								
RCSNDOP	000002	DMKIUJ	DMKIUS								
RCSYCLVL	000001	DMKIUS									

LABEL	COUNT	REFERENCES									
RCTINVLN	000001	DMKIUE									
RCTOTLEN	000002	DMKIUE	DMKIUL								
RCTRUNC	000003	DMKIUJ	DMKIUL	DMKIUS							
RCTTRUNC	000001	DMKIUE									
RCUADD	000033	DMKACS	DMKCAC	DMKCCH	DMKCKD	DMKCKH	DMKCKN	DMKDID	DMKDMQ	DMKDSB	DMKENT
		DMKIOQ	DMKMNI	DMKMNL	DMKMNT	DMKMON	DMKMOO	DMKQVM	DMKSCN	DMKSSP	DMKSSS
RCUBLOK	000156	DMKACR	DMKACS	DMKCAC	DMKCAO	DMKCCH	DMKCCO	DMKCFQ	DMKCFU	DMKCKD	DMKCKH
		DMKCKM	DMKCKN	DMKCNS	DMKCPM	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPV
		DMKCPW	DMKCPZ	DMKCQP	DMKCQQ	DMKDIB	DMKDID	DMKDIF	DMKDMQ	DMKDSB	DMKENT
		DMKGRG	DMKIOC	DMKIOQ	DMKIOS	DMKIOT	DMKMNI	DMKMNL	DMKMNT	DMKMON	DMKMOO
		DMKNES	DMKNLD	DMKPRV	DMKQVM	DMKSCN	DMKSCO	DMKSSP	DMKSSS	DMKURS	DMKVCH
		DMKVDA	DMKVDE	DMKVDS	DMKVDT	DMKVRR	DMKVRS	DMKVSC	DMKVSJ		
RCUBUSY	000022	DMKACS	DMKCNS	DMKENT	DMKIOQ	DMKIOS	DMKIOT	DMKQVM			
RCUCACH	000022	DMKCAC	DMKCAO	DMKCPZ	DMKMNT	DMKVDE	DMKVRR	DMKVRS			
RCUCHA	000049	DMKACR	DMKACS	DMKCAO	DMKCKD	DMKCKH	DMKCKN	DMKCPM	DMKCPO	DMKCPS	DMKCPT
		DMKCQP	DMKCQQ	DMKDID	DMKDMQ	DMKDSB	DMKENT	DMKIOQ	DMKIOS	DMKMNL	DMKMON
		DMKMOO	DMKPRV	DMKQVM	DMKSCN	DMKSCO	DMKSSP	DMKSSS	DMKURS	DMKVDA	
RCUCHB	000005	DMKCPN	DMKMNL	DMKPRV	DMKSSP	DMKURS					
RCUCHC	000004	DMKMNL	DMKPRV	DMKSCO	DMKSSP						
RCUCHCNT	000012	DMKCQQ	DMKIOQ	DMKMNT	DMKSCN	DMKVCH					
RCUCHD	000007	DMKACR	DMKCPM	DMKCPO	DMKMNL	DMKPRV	DMKSCN	DMKSSP			
RCUCUBSY	000015	DMKCFQ	DMKIOQ	DMKIOS	DMKIOT	DMKQVM	DMKVSJ				
RCUDISA	000022	DMKACR	DMKACS	DMKCFU	DMKCPM	DMKCPN	DMKCPW	DMKCPZ	DMKCQQ	DMKDID	DMKIOQ
		DMKIOT	DMKMNT	DMKNES	DMKNLD	DMKSCN	DMKVCH	DMKVRR			
RCUDVTBL	000040	DMKACR	DMKACS	DMKCAC	DMKCAO	DMKCCH	DMKCKD	DMKCKM	DMKCPM	DMKCPO	DMKCPP
		DMKCPS	DMKCPV	DMKCPZ	DMKCQP	DMKDIB	DMKDIF	DMKGRG	DMKMNI	DMKMNT	DMKNES
		DMKNLD	DMKSCN	DMKSSP	DMKVCH	DMKVDT					
RCUFIOB	000016	DMKACS	DMKDID	DMKIOQ	DMKIOS	DMKIOT					
RCUOWNER	000013	DMKCAC	DMKCAO	DMKCCO	DMKVRR	DMKVRS	DMKVSC				
RCUPRIME	000070	DMKACR	DMKACS	DMKCAO	DMKCCO	DMKCFQ	DMKCKD	DMKCKH	DMKCKM	DMKCKN	DMKCPM
		DMKCPO	DMKCPS	DMKCPT	DMKCQP	DMKCQQ	DMKDID	DMKDMQ	DMKDSB	DMKENT	DMKGRG
		DMKIOQ	DMKIOS	DMKIOT	DMKMNI	DMKMNL	DMKMNT	DMKMON	DMKMOO	DMKPRV	DMKQVM
		DMKSCN	DMKSCO	DMKSSS	DMKURS	DMKVCH	DMKVDA	DMKVDE	DMKVDT	DMKVRR	DMKVSC
RCUQCNT	000024	DMKDID	DMKENT	DMKIOQ	DMKIOS	DMKIOT	DMKMON				
RCURSTQ	000007	DMKIOQ	DMKIOS								
RCUSCED	000017	DMKACS	DMKCNS	DMKIOQ	DMKIOS	DMKIOT	DMKQVM				
RCUSENIO	000013	DMKIOQ	DMKIOS	DMKIOT							
RCUSENSE	000009	DMKIOQ	DMKIOS								
RCUSHRD	000007	DMKIOQ	DMKIOS	DMKVDS							
RCUSIZE	000012	DMKCAC	DMKCAO	DMKCPZ	DMKMNT	DMKQVM	DMKSSP	DMKVDT	DMKVRS		
RCUSTAT	000080	DMKACR	DMKACS	DMKCFU	DMKCNS	DMKCPM	DMKCPN	DMKCPW	DMKCPZ	DMKCQQ	DMKDID
		DMKENT	DMKIOQ	DMKIOS	DMKIOT	DMKMNT	DMKNES	DMKNLD	DMKQVM	DMKSCN	DMKVCH
		DMKVRR									
RCUSUB	000069	DMKACR	DMKACS	DMKCAO	DMKCCO	DMKCFQ	DMKCKD	DMKCKH	DMKCKM	DMKCKN	DMKCPM
		DMKCPO	DMKCPS	DMKCPT	DMKCPZ	DMKCQP	DMKCQQ	DMKDID	DMKDMQ	DMKDSB	DMKENT
		DMKGRG	DMKIOQ	DMKIOS	DMKIOT	DMKMNI	DMKMNL	DMKMNT	DMKMON	DMKMOO	DMKPRV
		DMKQVM	DMKSCN	DMKSCO	DMKSSS	DMKURS	DMKVCH	DMKVDA	DMKVDE	DMKVDT	DMKVRR
		DMKVSC									
RCUTYPE	000107	DMKACR	DMKACS	DMKCAC	DMKCAO	DMKCCO	DMKCFQ	DMKCKD	DMKCKH	DMKCKM	DMKCKN
		DMKCPM	DMKCPO	DMKCPS	DMKCPT	DMKCPZ	DMKCQP	DMKCQQ	DMKDID	DMKDMQ	DMKDSB
		DMKENT	DMKGRG	DMKIOC	DMKIOQ	DMKIOS	DMKIOT	DMKMNI	DMKMNL	DMKMNT	DMKMON
		DMKMOO	DMKPRV	DMKQVM	DMKSCN	DMKSCO	DMKSSP	DMKSSS	DMKURS	DMKVCH	DMKVDA
		DMKVDE	DMKVDS	DMKVDT	DMKVRR	DMKVRS	DMKVSC				
RCU2701	000002	DMKIOC									
RCU2702	000002	DMKIOC									
RCU2703	000002	DMKIOC									
RCU3880	000002	DMKIOS									
RCWADDR	000070	DMKCCD	DMKCCF	DMKCCO	DMKCCS	DMKCCT	DMKCCW	DMKDEX	DMKDGD	DMKDGF	DMKDIB

LABEL	COUNT	REFERENCES
		DMKTRK DMKUNT DMKVCA DMKVCB
RCWCCNT	000025	DMKCCD DMKCCO DMKCCS DMKCCW DMKCFR DMKCPB DMKDEX DMKIOS DMKISM DMKTRK
		DMKUNT DMKVDR DMKVIO
RCWCCW	000102	DMKCCD DMKCCF DMKCCO DMKCCS DMKCCT DMKCCW DMKCFR DMKCPB DMKDEX DMKDGD
		DMKDGF DMKDIB DMKHVC DMKIOS DMKISM DMKTRD DMKTRK DMKUNT DMKVCA DMKVCB
		DMKVDR DMKVIO
RCWCNT	000021	DMKCCD DMKCCO DMKCCW DMKDGD DMKDIB DMKTRK DMKUNT DMKVCA DMKVCB
RCWCOMND	000130	DMKCCD DMKCCF DMKCCO DMKCCS DMKCCT DMKCCW DMKDAD DMKDAS DMKDEX DMKDGD
		DMKDGF DMKDIB DMKTRK DMKUNT DMKVCA DMKVCB
RCWCTL	000054	DMKCCD DMKCCF DMKCCO DMKCCS DMKCCT DMKCCW DMKDEX DMKDGD DMKDIB DMKHVC
		DMKTRK DMKUNT DMKVCA DMKVCB
RCWFLAG	000112	DMKCCD DMKCCF DMKCCO DMKCCS DMKCCT DMKCCW DMKDEX DMKDGD DMKDIB DMKTRK
		DMKUNT DMKVCA DMKVCB
RCWGEN	000018	DMKCCD DMKCCF DMKCCW DMKDEX DMKHVC DMKTRD DMKTRK DMKUNT
RCWHEAD	000013	DMKCCS DMKCCW DMKCFR DMKCPB DMKDAD DMKHVC DMKTRK DMKVDR DMKVIO
RCWHMR	000008	DMKCCD DMKCCF DMKCCS DMKCCW DMKUNT
RCWINVL	000011	DMKCCW DMKDIB DMKTRK DMKVCA DMKVCB
RCWIO	000012	DMKCCW DMKDGD DMKISM DMKUNT
RCWISAM	000002	DMKCCD DMKCCO
RCWPNT	000021	DMKCCD DMKCCS DMKCCW DMKDAD DMKDEX DMKHVC DMKISM DMKTRD DMKTRK DMKUNT
		DMKVIO
RCWRCNT	000021	DMKCCD DMKCCS DMKCCW DMKCFR DMKDAD DMKDEX DMKHVC DMKISM DMKTRD DMKTRK
		DMKUNT DMKVIO
RCWREL	000023	DMKCCD DMKCCF DMKCCO DMKCCS DMKCCT DMKCCW DMKDEX DMKTRK
RCWSHR	000007	DMKCCW DMKDGD DMKUNT
RCWTASK	000074	DMKCCD DMKCCO DMKCCS DMKCCW DMKCFR DMKCPB DMKDAD DMKDEX DMKHVC DMKIOS
		DMKISM DMKTRD DMKTRK DMKUNT DMKVDR DMKVIO
RCWVCAW	000016	DMKCCS DMKCCW DMKHVC DMKISM DMKTRD DMKUNT DMKVIO
RCWVCNT	000007	DMKCCS DMKCCW DMKHVC DMKTRD
RCW2311	000007	DMKCCD DMKCCF DMKCCS DMKCCW DMKUNT
RC100	000001	DMKVDA
RC104	000001	DMKVDA
RC2MANY	000001	DMKIUC
RC2MANYT	000001	DMKIUC
RC32	000004	DMKLNK DMKVDC DMKVDS DMKVDT
RC40	000007	DMKCQP DMKCQQ DMKDEF DMKVDC DMKVDS
RDATA	000003	DMKCCO
RDBUFLN	000004	DMKRNH
RDBUFNO	000003	DMKRNH
RDCALTCL	000001	DMKUNT
RDCALTRK	000001	DMKUNT
RDCBLKAP	000017	DMKBIO DMKCCF DMKCCW DMKCPW DMKMNT DMKPAG DMKSPK DMKTDK DMKUNT DMKZTD
RDCBLKCE	000001	DMKUNT
RDCBLKCG	000007	DMKCPW DMKMNT DMKUNT
RDCBLKFA	000012	DMKCCF DMKCPW DMKDAU DMKMNT DMKUNT DMKVDG
RDCBLKMA	000016	DMKBIO DMKCCF DMKCPW DMKDAU DMKMNT DMKUNT DMKVDG
RDCBLKMX	000005	DMKCPW DMKLNK DMKMNT DMKPMA DMKVSC
RDCBLKPG	000008	DMKCPW DMKDMP DMKMNT DMKPAG DMKPAH DMKSPK
RDCBLOK	000048	DMKALO DMKBIO DMKCCF DMKCCW DMKCKM DMKCPW DMKDAU DMKDMP DMKDSB DMKHVE
		DMKIOC DMKIOJ DMKLNK DMKMNT DMKPAG DMKPAH DMKPGU DMKPMA DMKRSP DMKRSQ
		DMKSPK DMKTDK DMKUNT DMKVDG DMKVSC DMKVSX DMKZTD
RDCCKD	000003	DMKCPW DMKMNT
RDCDIAG	000001	DMKUNT
RDCDIAGN	000001	DMKUNT
RDCDVCLS	000002	DMKIOC
RDCDVCYL	000001	DMKUNT
RDCDVTRK	000001	DMKUNT
RDCFBA	000002	DMKCPW DMKMNT

LABEL	COUNT	REFERENCES									
RDCFEAT	000001	DMKDSB									
RDCFLAG	000005	DMKCPW	DMKMNT								
RDCFPNT	000004	DMKCPW	DMKMNT								
RDCLENG	000005	DMKCCW	DMKCPW	DMKMNT							
RDCLENGC	000004	DMKCPW	DMKMNT								
RDCLENGF	000004	DMKCPW	DMKMNT								
RDCMDR	000001	DMKIOJ									
RDCOBR	000002	DMKIOC									
RDCPAGAP	000015	DMKALO	DMKCKM	DMKCPW	DMKHVE	DMKMNT	DMKPGU	DMKRSP	DMKRSQ	DMKVDG	DMKVSX
RDCPAGCG	000002	DMKCPW	DMKMNT								
RDCPAGFA	000003	DMKALO	DMKCPW	DMKMNT							
RDCPAGMA	000003	DMKALO	DMKCPW	DMKMNT							
RDCPAGXT	000004	DMKMNT	DMKPAG	DMKSPK							
RDCPRIM	000002	DMKBIO	DMKUNT								
RDCRECSZ	000003	DMKCPW	DMKMNT	DMKVDG							
RDCRRLSE	000001	DMKDSB									
RDCSIZE	000004	DMKCPW	DMKMNT								
RDCSTART	000005	DMKCCW	DMKCPW	DMKMNT							
RDCTYPE	000002	DMKCPW	DMKMNT								
RDEVACTV	000031	DMKCNS	DMKDIB	DMKGRD	DMKGRF	DMKGRG	DMKQCO	DMKQVM	DMKTTY		
RDEVADD	000093	DMKACO	DMKACR	DMKACS	DMKCCH	DMKCCS	DMKCFT	DMKCKD	DMKCKF	DMKCKH	DMKCKN
		DMKCPI	DMKCPM	DMKCPO	DMKCPT	DMKCPW	DMKCQG	DMKCQP	DMKCQQ	DMKCQS	DMKCQY
		DMKDIB	DMKDID	DMKDIF	DMKDMQ	DMKDSB	DMKDSP	DMKENT	DMKHPS	DMKHPT	DMKHPU
		DMKIOQ	DMKJRL	DMKLOJ	DMKMNI	DMKMNL	DMKMNT	DMKMON	DMKMOO	DMKNES	DMKNLD
		DMKPAG	DMKSCN	DMKSSP	DMKSSS	DMKVCH	DMKVDA	DMKVDC	DMKVDD	DMKVDF	DMKVDG
		DMKVDR	DMKVDT								
RDEVADFF	000007	DMKGRC	DMKGRD	DMKGRT	DMKHPS						
RDEVADVF	000045	DMKGRC	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKHVE	DMKRGA	DMKVCP	DMKVCX	
RDEVAINH	000004	DMKGRD	DMKGRG								
RDEVAIOB	000104	DMKACS	DMKCFQ	DMKCFR	DMKCKD	DMKCNS	DMKCPS	DMKCSO	DMKDAD	DMKDAS	DMKDAU
		DMKDIA	DMKDID	DMKDIF	DMKDSP	DMKGRD	DMKGRF	DMKHPS	DMKHPT	DMKHPU	DMKIOS
		DMKIOT	DMKLOH	DMKNLD	DMKOPE	DMKQVM	DMKRGB	DMKRSE	DMKVSJ		
RDEVAIRA	000034	DMKCFM	DMKCFQ	DMKDIF	DMKGRF	DMKGRG	DMKGRI	DMKHPT	DMKLOH	DMKQVM	DMKUSQ
		DMKVCR	DMKVCS	DMKVCT	DMKVCV	DMKVCX					
RDEVALGN	000003	DMKRSP									
RDEVALLN	000028	DMKALO	DMKCKF	DMKCKT	DMKCKV	DMKCPZ	DMKDAD	DMKDAS	DMKDAU	DMKDMP	DMKIDU
		DMKMOO	DMKPAG	DMKPGU	DMKVDG	DMKWRM					
RDEVALT	000023	DMKACR	DMKACS	DMKCCS	DMKCPM	DMKCPN	DMKCPT	DMKCPW	DMKCPZ	DMKMNT	DMKQVM
		DMKVCH	DMKVRR	DMKVSC							
RDEVAOF	000009	DMKACR	DMKCPO	DMKCQQ	DMKIOQ	DMKMNL	DMKMNT	DMKSCN			
RDEVAPLI	000004	DMKTTX	DMKVCQ								
RDEVAPLO	000008	DMKCFT	DMKCNS	DMKTTX	DMKTTY	DMKVCQ					
RDEVAPLP	000017	DMKCFT	DMKCQU	DMKGRF	DMKGRG	DMKGRI	DMKQCQ	DMKVCN	DMKVCR	DMKVCU	DMKVCV
RDEVASGN	000013	DMKACS	DMKCPM	DMKCPN	DMKCPW	DMKMNT					
RDEVASTB	000009	DMKBLD	DMKCFT	DMKCNS	DMKCQU	DMKNES	DMKTTX	DMKVCP			
RDEVATNC	000008	DMKCNS									
RDEVATOF	000012	DMKCFT	DMKCNS	DMKCQU	DMKOPE	DMKTRM	DMKTTY	DMKVCT			
RDEVATSW	000008	DMKCNS	DMKGRF	DMKLOG	DMKLOJ						
RDEVATT	000029	DMKCPN	DMKCQG	DMKCQP	DMKDEF	DMKDIB	DMKDIF	DMKHPT	DMKHPU	DMKIOT	DMKNLD
		DMKRGE	DMKSSS	DMKVCH	DMKVCN	DMKVCP	DMKVDD	DMKVDR	DMKVDS	DMKVDT	DMKVRS
RDEVAUTO	000008	DMKCKD	DMKCPJ	DMKCQP	DMKNET	DMKNLE	DMKRNH	DMKWRM			
RDEVAVM1	000004	DMKBLD	DMKCFT	DMKTTX							
RDEVAVM2	000005	DMKCFT	DMKCNS	DMKCQU	DMKNES	DMKVCP					
RDEVBACK	000012	DMKCSF	DMKRSE	DMKRSP							
RDEVBASE	000009	DMKCCT	DMKCPV	DMKDIB	DMKDIF	DMKEXT	DMKNES	DMKNET	DMKNLD		
RDEVBLOK	000688	DMKACO	DMKACR	DMKACS	DMKALO	DMKATS	DMKBIO	DMKBLD	DMKBSC	DMKCAC	DMKCAO
		DMKCCD	DMKCCF	DMKCCH	DMKCCO	DMKCCS	DMKCCT	DMKCCW	DMKCFC	DMKCFG	DMKCFH
		DMKCFM	DMKCFQ	DMKCFR	DMKCFS	DMKCFT	DMKCFU	DMKCFW	DMKCFY	DMKCKD	DMKCKF

LABEL	COUNT	REFERENCES									
		DMKCKH	DMKCKM	DMKCKN	DMKCKR	DMKCKS	DMKCKT	DMKCKV	DMKCNS	DMKCNT	DMKCPB
		DMKCPI	DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPV	DMKCPW
		DMKCPX	DMKCPZ	DMKCQG	DMKCQI	DMKCQP	DMKCQQ	DMKCQR	DMKCQS	DMKCQT	DMKCQU
		DMKCQY	DMKCSB	DMKCSC	DMKCSF	DMKCSO	DMKDAD	DMKDAS	DMKDAU	DMKDEF	DMKDEI
		DMKDGD	DMKDGF	DMKDIA	DMKDIB	DMKDID	DMKDIF	DMKDMP	DMKDMQ	DMKDRD	DMKDRE
		DMKDSB	DMKDSP	DMKENT	DMKEPS	DMKERM	DMKEXT	DMKGIO	DMKGRC	DMKGRD	DMKGRE
		DMKGRF	DMKGRG	DMKGRH	DMKGRI	DMKGRT	DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKHVD
		DMKHVE	DMKHVF	DMKIDU	DMKIOC	DMKIOE	DMKIOF	DMKIOG	DMKIOJ	DMKIOQ	DMKIOS
		DMKIOT	DMKJRL	DMKLNK	DMKLOG	DMKLOH	DMKLOJ	DMKLOM	DMKMCC	DMKMNI	DMKMNL
		DMKMNT	DMKMON	DMKMOO	DMKMSW	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKOPE
		DMKOPR	DMKPAG	DMKPAH	DMKPGT	DMKPGU	DMKPMA	DMKPRV	DMKQCN	DMKQCO	DMKQCP
		DMKQCQ	DMKQVM	DMKRGA	DMKRGB	DMKRGC	DMKRGE	DMKRNH	DMKRSE	DMKRSF	DMKRSP
		DMKRSQ	DMKRST	DMKSCN	DMKSCO	DMKSEG	DMKSEP	DMKSNC	DMKSPK	DMKSPL	DMKSPS
		DMKSPT	DMKSSP	DMKSSS	DMKSST	DMKSSU	DMKSSV	DMKTAP	DMKTAQ	DMKTCS	DMKTCT
		DMKTDK	DMKTPE	DMKTRD	DMKTRK	DMKTRM	DMKTTX	DMKTTY	DMKUDR	DMKUNT	DMKURS
		DMKUSQ	DMKVCH	DMKVCN	DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCU	DMKVCV
		DMKVCW	DMKVCX	DMKVDA	DMKVDB	DMKVDC	DMKVDD	DMKVDE	DMKVDF	DMKVDG	DMKVDH
		DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVIO	DMKVRR	DMKVRS	DMKVSC	DMKVSI	DMKVSJ
		DMKVSX	DMKWRM	DMKWRN	DMKZTD						
RDEVBOF	000009	DMKACR	DMKCPO	DMKCQQ	DMKIOQ	DMKMNL	DMKMNT	DMKSCN			
RDEVBPAG	000008	DMKCAC	DMKCPW	DMKCPZ	DMKCQG	DMKCQP	DMKLOJ				
RDEVBSC	000010	DMKBSC	DMKNET	DMKRGA	DMKRGB						
RDEVBUZY	000057	DMKCCH	DMKCFQ	DMKCNS	DMKCPB	DMKCPO	DMKCPW	DMKCSO	DMKDID	DMKENT	DMKGRF
		DMKHPS	DMKHPT	DMKHPU	DMKIOQ	DMKIOS	DMKIOT	DMKPAG	DMKQVM	DMKRNH	DMKRSE
		DMKRSP	DMKVCH	DMKVDA							
RDEVBZCH	000015	DMKACS	DMKCNS	DMKCSO	DMKIOS	DMKIOT	DMKQVM				
RDEVCC3	000004	DMKIOT	DMKZTD								
RDEVCFCB	000005	DMKCSO	DMKRSP	DMKURS							
RDEVCFLT	000004	DMKDSB	DMKMNT	DMKSSU							
RDEVCKDX	000006	DMKCCD	DMKCCW	DMKMNT	DMKSSP	DMKVDE					
RDEVCKP	000006	DMKCKH	DMKCKS	DMKWRM							
RDEVCKPT	000007	DMKRNH	DMKWRM								
RDEVCLAS	000011	DMKCFQ	DMKCKH	DMKCKR	DMKCKS	DMKCKV	DMKCSO	DMKRSP	DMKSSP	DMKURS	DMKWRM
RDEVCMDK	000024	DMKCCO	DMKCCW	DMKDGD	DMKMNL	DMKMNT	DMKVDA	DMKVDE	DMKVSC		
RDEVCODE	000035	DMKALO	DMKATS	DMKCFG	DMKCFH	DMKCFS	DMKCKF	DMKCKM	DMKCKT	DMKCPP	DMKCPZ
		DMKCQY	DMKHVD	DMKHVE	DMKHVF	DMKIDU	DMKIOG	DMKNLD	DMKOPE	DMKPAG	DMKPGT
		DMKSEG	DMKSNC	DMKTCS	DMKVDG	DMKWRM					
RDEVCON	000113	DMKCFM	DMKCFQ	DMKCNS	DMKDIF	DMKDSP	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRI
		DMKNES	DMKQCO	DMKQVM	DMKRGA	DMKRNH	DMKTTX	DMKTTY	DMKVCQ	DMKVCR	DMKVCS
		DMKVCV	DMKVCW	DMKVCX							
RDEVCORD	000026	DMKCFT	DMKDIA	DMKDIB	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKOPR	DMKQCN	DMKTTX
		DMKTTY									
RDEVCORR	000003	DMKCNS	DMKTRM								
RDEVCPEX	000004	DMKIOT	DMKZTD								
RDEVCPIO	000003	DMKIOT	DMKZTD								
RDEVCPNA	000005	DMKGRF	DMKGRG								
RDEVCRDC	000009	DMKBIO	DMKCCO	DMKCCS	DMKCPW	DMKMNT	DMKVDE				
RDEVCSW	000006	DMKACS	DMKCFQ	DMKCFR	DMKDID	DMKIOT					
RDEVCTL	000046	DMKACS	DMKCNS	DMKDIB	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRH	DMKQVM	DMKVDS
RDEVCTRS	000017	DMKCPS	DMKCPW	DMKHPS	DMKIOE	DMKIOF	DMKIOJ	DMKNES	DMKNET		
RDEVCUA	000062	DMKACR	DMKACS	DMKCAO	DMKCCO	DMKCFQ	DMKCKD	DMKCKH	DMKCKN	DMKCPM	DMKCPO
		DMKCPS	DMKCPT	DMKCPZ	DMKCQP	DMKCQQ	DMKDIB	DMKDID	DMKDIF	DMKDMQ	DMKDSB
		DMKENT	DMKGRG	DMKHPS	DMKIOC	DMKIOQ	DMKIOS	DMKMNI	DMKMNL	DMKMNT	DMKMON
		DMKMOO	DMKNES	DMKNLD	DMKPRV	DMKQVM	DMKSCN	DMKSCO	DMKSSP	DMKSSS	DMKURS
		DMKVCH	DMKVDA	DMKVDE	DMKVRR	DMKVSC					
RDEVCUB	000035	DMKACS	DMKCAC	DMKCAO	DMKCFQ	DMKCKD	DMKCPZ	DMKCQQ	DMKDAD	DMKDAS	DMKDAU
		DMKDID	DMKDMQ	DMKDSB	DMKIOQ	DMKIOS	DMKMNL	DMKMNT	DMKSCN	DMKSCO	DMKTAP
		DMKVCH	DMKVDE	DMKVRR							

LABEL	COUNT	REFERENCES									
RDEVCUP	000072	DMKALO	DMKCAC	DMKCAO	DMKCFH	DMKCKM	DMKCKN	DMKCPT	DMKCPW	DMKCPZ	DMKCQP
		DMKCQQ	DMKDAS	DMKDEF	DMKDGD	DMKDSB	DMKGIO	DMKLNK	DMKLOJ	DMKMNL	DMKMNT
		DMKPAG	DMKVCH	DMKVDA	DMKVDB	DMKVDC	DMKVDD	DMKVDE	DMKVDF	DMKVDG	DMKVRR
RDEVCURP	000003	DMKTCS	DMKVDR								
RDEVCU11	000002	DMKCPZ	DMKMNT								
RDEVCU2	000010	DMKCPZ	DMKMNL	DMKVDT							
RDEVCYL	000005	DMKDIB	DMKDIF	DMKIOQ	DMKIOS	DMKMON					
RDEVDED	000113	DMKCFQ	DMKCFU	DMKCKD	DMKCKH	DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPS	DMKCPV
		DMKCPW	DMKCPZ	DMKCQP	DMKCQT	DMKCSB	DMKCSF	DMKCSO	DMKDAS	DMKDAU	DMKDEI
		DMKDIB	DMKDIF	DMKDSB	DMKDSP	DMKGRG	DMKHPT	DMKHPU	DMKIOQ	DMKIOS	DMKIOT
		DMKLNK	DMKLOJ	DMKMCC	DMKMNL	DMKMSW	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKNLE
		DMKQVM	DMKRGB	DMKRNH	DMKRSP	DMKRST	DMKSCN	DMKSPT	DMKSSS	DMKSST	DMKVCH
		DMKVDA	DMKVDC	DMKVDD	DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVRR	DMKVRS	
RDEVDELP	000018	DMKCKH	DMKCPO	DMKCPS	DMKCPW	DMKRSE	DMKRSP	DMKVDR	DMKVDS		
RDEVDFCB	000006	DMKCSO	DMKRSP	DMKURS							
RDEVDIIP	000007	DMKDIA	DMKDIF	DMKGRG	DMKHPS	DMKIOT					
RDEVDISA	000118	DMKACR	DMKACS	DMKCAC	DMKCFU	DMKCKH	DMKCKN	DMKCKR	DMKCKV	DMKCNS	DMKCPJ
		DMKCPM	DMKCPN	DMKCPO	DMKCPS	DMKCPT	DMKCPV	DMKCPW	DMKCPZ	DMKCQP	DMKCQT
		DMKCSB	DMKCSF	DMKCSO	DMKDAD	DMKDAS	DMKDAU	DMKDEI	DMKDIB	DMKDID	DMKDMP
		DMKDSB	DMKGRG	DMKIOS	DMKIOT	DMKLNK	DMKLOJ	DMKMCC	DMKMNI	DMKMNL	DMKMNT
		DMKNEA	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKQVM	DMKRGA	DMKRGB	DMKRNH	DMKRSP
		DMKSCN	DMKSPT	DMKSSS	DMKSST	DMKVCH	DMKVDA	DMKVDC	DMKVDD	DMKVDE	DMKVDS
		DMKVDT	DMKVRR	DMKWRM							
RDEVDISB	000033	DMKACR	DMKCKD	DMKCNS	DMKCPV	DMKCQP	DMKDIB	DMKGRF	DMKNES	DMKNET	DMKRGA
		DMKRGB	DMKVDA								
RDEVDRAN	000036	DMKCKH	DMKCKR	DMKCKV	DMKCPO	DMKCPW	DMKCQP	DMKCSB	DMKCSO	DMKQVM	DMKRSP
		DMKRST	DMKURS	DMKVCH	DMKVDA	DMKVDC	DMKVDS	DMKWRM			
RDEVDROP	000009	DMKCNS	DMKDID	DMKGRF	DMKQCO						
RDEVDUPL	000004	DMKMNT									
RDEVECKD	000022	DMKCCD	DMKCCW	DMKCKM	DMKCKN	DMKCKP	DMKDGD	DMKDGF	DMKDMP	DMKMON	DMKPAG
		DMKVDG	DMKZTD								
RDEVECOL	000014	DMKGRC	DMKGRD	DMKGRE	DMKGRG	DMKHVE	DMKQCN	DMKVCP	DMKVCX		
RDEVEHLT	000014	DMKGRC	DMKGRD	DMKGRE	DMKGRG	DMKHVE	DMKQCN	DMKVCP	DMKVCX		
RDEVENAB	000055	DMKACR	DMKCKD	DMKCNS	DMKCPO	DMKCPV	DMKCPW	DMKCQP	DMKGRF	DMKGRG	DMKHPS
		DMKNES	DMKNET	DMKNLD	DMKOPE	DMKRGA	DMKRGB	DMKVCH	DMKVDA	DMKVDC	DMKVDS
		DMKVRR	DMKWRM								
RDEVEPDV	000024	DMKDIA	DMKDIB	DMKDIF	DMKNES	DMKNLD					
RDEVEPLN	000006	DMKCPO	DMKCPW	DMKDIF	DMKNES	DMKNLD	DMKVDS				
RDEVEPMD	000007	DMKCNS	DMKDIB	DMKDIF	DMKNES	DMKNLD					
RDEVERR	000003	DMKGRF									
RDEVEWO	000005	DMKGRC	DMKGRD	DMKHPS							
RDEVEXTN	000052	DMKCKH	DMKCKS	DMKCKV	DMKCPW	DMKCSB	DMKCSF	DMKCSO	DMKMNT	DMKRSP	DMKSEP
		DMKTCS	DMKTCT	DMKURS	DMKWRM						
RDEVFCNS	000005	DMKCNS	DMKGRF	DMKUSQ							
RDEVFIOB	000033	DMKACS	DMKCFQ	DMKCPI	DMKCPS	DMKDID	DMKDSP	DMKHPS	DMKHPT	DMKHPU	DMKIOQ
		DMKIOS	DMKIOT	DMKNLD	DMKPAG	DMKSSU	DMKVDG				
RDEVFLAG	000448	DMKACR	DMKALO	DMKBLD	DMKCFC	DMKCFG	DMKCFH	DMKCFM	DMKCFT	DMKCFU	DMKCKD
		DMKCKH	DMKCKR	DMKCKV	DMKCNS	DMKCPJ	DMKCPM	DMKCPO	DMKCPV	DMKCPW	DMKCPZ
		DMKCQP	DMKCQQ	DMKCSB	DMKCSF	DMKCSO	DMKDAD	DMKDAS	DMKDAU	DMKDEI	DMKDIB
		DMKDIF	DMKDSB	DMKEXT	DMKGRD	DMKGRF	DMKGRG	DMKHPS	DMKHVD	DMKHVF	DMKIOQ
		DMKIOS	DMKIOT	DMKLNK	DMKLOG	DMKLOJ	DMKLOM	DMKMCC	DMKMNI	DMKMNL	DMKMNT
		DMKMON	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKOPE	DMKPAG	DMKQCO	DMKQVM	DMKRGA
		DMKRGB	DMKRNH	DMKRSE	DMKRSP	DMKRST	DMKSCN	DMKSEP	DMKSNC	DMKSPS	DMKSPT
		DMKSSS	DMKSST	DMKSSV	DMKTCS	DMKTRM	DMKTTY	DMKURS	DMKVCH	DMKVCT	DMKVDA
		DMKVDC	DMKVDD	DMKVDG	DMKVDH	DMKVDR	DMKVDS	DMKVDT	DMKVRR	DMKVRS	DMKWRM
RDEVFOFF	000005	DMKACR	DMKCPM	DMKCPN	DMKCPW						
RDEVFORC	000004	DMKDID	DMKGRF								
RDEVFSEP	000014	DMKCKH	DMKCKS	DMKCKV	DMKCSO	DMKTCS	DMKURS	DMKWRM			

LABEL	COUNT	REFERENCES									
RDEVFTR	000153	DMKACO	DMKALO	DMKBIO	DMKBSC	DMKCCD	DMKCCO	DMKCCS	DMKCCW	DMKCFQ	DMKCPW
		DMKCQP	DMKCSB	DMKDAD	DMKDAS	DMKDAU	DMKDEI	DMKDGD	DMKDID	DMKDSB	DMKGRD
		DMKGRF	DMKHVE	DMKIOE	DMKIOJ	DMKIOS	DMKIOT	DMKLNK	DMKLOJ	DMKMNT	DMKNLD
		DMKNLE	DMKPAG	DMKPMA	DMKRGA	DMKRGB	DMKRGC	DMKRSF	DMKSCN	DMKSPK	DMKSSP
		DMKSSS	DMKSST	DMKTAP	DMKTCT	DMKUNT	DMKVCN	DMKVDA	DMKVDC	DMKVDE	DMKVDG
		DMKVDS	DMKVER	DMKVRR	DMKVRS	DMKVSC	DMKZTD				
RDEVFTR2	000012	DMKCPM	DMKGRC	DMKMNT	DMKPAG	DMKVCN	DMKVCP	DMKVDE			
RDEVFTR3	000012	DMKCPZ	DMKMNL	DMKMNT	DMKVDT						
RDEVGRF1	000031	DMKGRD	DMKGRF	DMKGRG	DMKGRI						
RDEVGRTY	000002	DMKOPR	DMKSSP								
RDEVHIO	000020	DMKCNS	DMKDIB	DMKEXT	DMKGRF						
RDEVHOLD	000009	DMKGRD	DMKGRF	DMKGRG	DMKGRH						
RDEVHSTR	000008	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRI					
RDEVHT	000025	DMKDIA	DMKGRC	DMKGRD	DMKGRF	DMKGRG	DMKGRT	DMKHPS	DMKQCN	DMKSSP	DMKVCN
		DMKVCP									
RDEVIDNT	000011	DMKCFC	DMKCFM	DMKCNS	DMKOPE	DMKTRM					
RDEVIMAG	000013	DMKCKH	DMKCKS	DMKCKV	DMKCPS	DMKCSO	DMKRSP	DMKTCS	DMKURS	DMKWRM	
RDEVIOAT	000013	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRI					
RDEVIOBL	000012	DMKCPI	DMKIOQ	DMKIOS	DMKPAG	DMKVDG					
RDEVIOCP	000015	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRI					
RDEVIOCT	000005	DMKIOS	DMKMNI	DMKMON	DMKMOO						
RDEVIODE	000009	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRI					
RDEVIOER	000056	DMKBSC	DMKCFQ	DMKCFR	DMKCPS	DMKCSO	DMKDAD	DMKDAS	DMKDAU	DMKDID	DMKGRF
		DMKIOS	DMKIOT	DMKNLD	DMKRSE	DMKRSF	DMKRSP	DMKTAP	DMKTPE	DMKTRK	
RDEVIRM	000007	DMKCFU	DMKIOE	DMKNES	DMKNLD						
RDEVISPL	000005	DMKRSP									
RDEVLCEP	000011	DMKACR	DMKCKD	DMKCPW	DMKCQP	DMKDIB	DMKNLD	DMKRNH			
RDEVLDBG	000004	DMKCSO	DMKRSP	DMKURS							
RDEVLDMD	000004	DMKCSO	DMKRSP	DMKURS							
RDEVLGDT	000003	DMKGRF									
RDEVLIOB	000007	DMKACS	DMKCPI	DMKHPS	DMKIOQ	DMKVDG					
RDEVLLEN	000015	DMKBLD	DMKCFT	DMKCQU	DMKHPS	DMKHVE	DMKSSP	DMKVCP	DMKVCS	DMKVCV	
RDEVLNCP	000023	DMKACR	DMKCKD	DMKCPW	DMKCQP	DMKDIB	DMKNES	DMKNET	DMKNLD	DMKRNH	DMKVDS
RDEVLNKS	000021	DMKCPM	DMKCPO	DMKCPW	DMKCQP	DMKDEI	DMKSCO	DMKSSS	DMKTDK	DMKVDE	DMKVDF
		DMKVDR	DMKVDS								
RDEVLOAD	000005	DMKCSO	DMKRSP	DMKSEP							
RDEVLOG	000013	DMKCNS	DMKCPV	DMKGRF	DMKVCP	DMKVDA					
RDEVLOGC	000001	DMKUSQ									
RDEVLOW	000014	DMKCCD	DMKDGD	DMKDGF	DMKPAG	DMKVDG					
RDEVLTRM	000004	DMKGRF	DMKUSQ								
RDEVMAX	000031	DMKACR	DMKCKD	DMKCPJ	DMKCPW	DMKCQT	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKRGA
		DMKRGB	DMKRGC	DMKRNH	DMKVDS	DMKWRM					
RDEVMAXP	000001	DMKRSP									
RDEVMDL	000080	DMKACO	DMKALO	DMKBIO	DMKCCD	DMKCCW	DMKCKF	DMKCKM	DMKCKN	DMKCPW	DMKDGD
		DMKDGF	DMKDMP	DMKGRF	DMKGRG	DMKHPS	DMKHVE	DMKIOC	DMKIOJ	DMKLNK	DMKMNT
		DMKMON	DMKNES	DMKNLD	DMKNLE	DMKPAG	DMKPMA	DMKSPK	DMKSSP	DMKUNT	DMKVCP
		DMKVDE	DMKVDG	DMKVER	DMKVSC	DMKZTD					
RDEVMD00	000003	DMKCPW	DMKMNT	DMKPMA							
RDEVMD02	000011	DMKALO	DMKCCD	DMKDMP	DMKHVE	DMKPAG	DMKPMA	DMKVDE	DMKVDG	DMKVSC	
RDEVMD13	000003	DMKMNL	DMKMNT	DMKVDE							
RDEVMID	000020	DMKACS	DMKCFR	DMKDID	DMKIOS	DMKIOT					
RDEVMORE	000010	DMKGRF	DMKGRG	DMKGRH							
RDEVMOUT	000025	DMKCPM	DMKCPW	DMKCQP	DMKDAS	DMKDAU	DMKDSB	DMKSCN	DMKVDA	DMKVDD	DMKVDR
		DMKVDS	DMKVDT								
RDEVMO00	000004	DMKCPW	DMKMNT								
RDEVMO04	000004	DMKCPW	DMKIOC	DMKIOJ	DMKMNT						
RDEVNAME	000023	DMKCKN	DMKCPV	DMKGRG	DMKHPS	DMKIOC	DMKMNT	DMKMOO	DMKMSW	DMKSSP	
RDEVNATH	000008	DMKALO	DMKCPN	DMKCPW	DMKMNT	DMKVDA					

LABEL	COUNT	REFERENCES									
RDEVNCP	000010	DMKCKD	DMKCQP	DMKNLD	DMKRNH	DMKWRM					
RDEVNDLF	000007	DMKCNS	DMKTTX	DMKTTY							
RDEVNICL	000078	DMKACO	DMKACR	DMKCFM	DMKCFQ	DMKCFT	DMKCFY	DMKCKD	DMKCKF	DMKCPJ	DMKCPW
		DMKCQG	DMKCQT	DMKCQU	DMKDEF	DMKDIA	DMKDIB	DMKERM	DMKEXT	DMKHVD	DMKHVE
		DMKLOG	DMKLOH	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKQCN	DMKQCO	DMKRGA	DMKRGB
		DMKRGC	DMKRNH	DMKSCN	DMKVCN	DMKVDR	DMKVDS	DMKWRM			
RDEVNOCR	000011	DMKTTX	DMKTTY								
RDEVNOHD	000004	DMKCNS									
RDEVNOLF	000010	DMKTTX	DMKTTY								
RDEVNOW	000009	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRI					
RDEVNRDY	000067	DMKCFQ	DMKCNS	DMKCPJ	DMKCPM	DMKCPN	DMKCPS	DMKCPW	DMKCQP	DMKCQT	DMKCSO
		DMKDAD	DMKDAS	DMKDAU	DMKDIB	DMKDID	DMKIDU	DMKIOS	DMKIOT	DMKNES	DMKNET
		DMKNLD	DMKNLE	DMKRGA	DMKRGB	DMKRNH	DMKRSE	DMKRSP	DMKRST	DMKSST	DMKTAP
		DMKTPE	DMKURS	DMKVCN	DMKVDE	DMKVDS					
RDEVOFF	000004	DMKIOS	DMKIOT								
RDEVOVLY	000004	DMKCSO	DMKRSP	DMKURS	DMKVDR						
RDEVOWN	000047	DMKALO	DMKCFG	DMKCFH	DMKCPM	DMKCPW	DMKCPZ	DMKCQP	DMKCQQ	DMKDAD	DMKDAS
		DMKDAU	DMKDSB	DMKHVD	DMKHVF	DMKLOJ	DMKMNL	DMKNLD	DMKSNC	DMKSSS	DMKSST
		DMKTCS	DMKVCH	DMKVDA	DMKVDC	DMKVDD	DMKVDG	DMKVDH	DMKVDR	DMKVDS	DMKVDT
RDEVPBYP	000003	DMKCPT									
RDEVPCHG	000017	DMKBLD	DMKCFT	DMKTTX	DMKTTY						
RDEVPCNT	000005	DMKCNS									
RDEVPDLY	000003	DMKNES	DMKRGA								
RDEVPEND	000028	DMKACR	DMKCFU	DMKCPO	DMKCPT	DMKCPV	DMKCPZ	DMKCQT	DMKDIA	DMKDID	DMKMCC
		DMKNES	DMKNET	DMKNLD	DMKNLE	DMKRGB	DMKRNH	DMKSPT	DMKVDA	DMKVDS	DMKVDT
RDEVPIOB	000011	DMKCFQ	DMKDID	DMKIOQ	DMKIOS	DMKIOT					
RDEVPPAG	000032	DMKCAC	DMKCPM	DMKCPT	DMKCPW	DMKCPZ	DMKCQG	DMKCQP	DMKCQQ	DMKDSB	DMKLOJ
		DMKMNL	DMKMNT	DMKVCH	DMKVDA	DMKVDC	DMKVDD	DMKVDE	DMKVDT		
RDEVPRDV	000054	DMKALO	DMKCKM	DMKCKN	DMKCPT	DMKCPW	DMKCPZ	DMKCQG	DMKCQP	DMKCQQ	DMKDSB
		DMKLOJ	DMKMNT	DMKPAG	DMKVDA	DMKVDD	DMKVDE	DMKVDG	DMKVDT		
RDEVPREP	000014	DMKCNS	DMKDIB	DMKQVM	DMKVDA						
RDEVPRFG	000032	DMKCKH	DMKCKS	DMKCKV	DMKCNS	DMKCSO	DMKGRF	DMKRSP	DMKSEP	DMKTCT	DMKURS
		DMKUSQ	DMKWRM								
RDEVPROC	000033	DMKACS	DMKCKD	DMKCNS	DMKCPP	DMKDID	DMKENT	DMKEXT	DMKIOS	DMKIOT	DMKRSF
		DMKTPE	DMKVSJ								
RDEVPS	000052	DMKACO	DMKBLD	DMKCFQ	DMKCFR	DMKCKF	DMKCQG	DMKCQP	DMKCQQ	DMKCQS	DMKCQY
		DMKDIA	DMKDIB	DMKDID	DMKEXT	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKHPS	DMKJRL
		DMKLOH	DMKPRV	DMKQCP	DMKSCN	DMKUSQ	DMKVDA	DMKVDS	DMKVIO	DMKVSI	DMKVSJ
RDEVPSS	000005	DMKGRC	DMKGRE	DMKHVE	DMKVCP						
RDEVPSUP	000020	DMKBLD	DMKCFT	DMKCNS	DMKHPS	DMKLOG	DMKTTX	DMKTTY	DMKVCT		
RDEVPT	000003	DMKGRC	DMKHVE	DMKVCP							
RDEVPTHS	000072	DMKACR	DMKACS	DMKCFU	DMKCPM	DMKCPN	DMKCPO	DMKCPT	DMKCPW	DMKCPZ	DMKCQQ
		DMKDID	DMKDIF	DMKIOQ	DMKMNL	DMKMNT	DMKNES	DMKNLD	DMKRSP	DMKSCN	DMKURS
		DMKVRR	DMKVRS	DMKWRM							
RDEVPTH1	000004	DMKACR	DMKCPM	DMKCPO	DMKVRR						
RDEVPTH2	000004	DMKACR	DMKCPM	DMKCPO	DMKVRR						
RDEVPTH3	000004	DMKACR	DMKCPM	DMKCPO	DMKVRR						
RDEVPTH4	000004	DMKACR	DMKCPM	DMKCPO	DMKVRR						
RDEVPTH5	000004	DMKACR	DMKCPM	DMKCPO	DMKVRR						
RDEVPTH6	000004	DMKACR	DMKCPM	DMKCPO	DMKVRR						
RDEVPTH7	000004	DMKACR	DMKCPM	DMKCPO	DMKVRR						
RDEVPTH8	000004	DMKACR	DMKCPM	DMKCPO	DMKVRR						
RDEVPTTC	000010	DMKCNS	DMKNES	DMKNLD	DMKOPE	DMKTRM	DMKVCU	DMKVCW			
RDEVPURG	000008	DMKCSO	DMKRSP	DMKTCS	DMKTCT	DMKURS					
RDEVQBSY	000011	DMKCFQ	DMKDID	DMKIOQ	DMKIOS	DMKIOT					
RDEVQCNT	000017	DMKACS	DMKENT	DMKIOQ	DMKIOS	DMKIOT	DMKMON	DMKPAG	DMKSSU		
RDEVQIOB	000013	DMKCFQ	DMKCPS	DMKDID	DMKIOQ	DMKIOS	DMKIOT				
RDEVQREP	000010	DMKGRC	DMKHPT	DMKHVE	DMKVCT	DMKVCW	DMKVCX				

LABEL	COUNT	REFERENCES									
RDEVQRY	000010	DMKGRF									
RDEVQUED	000004	DMKCKM	DMKCKN								
RDEVRCNT	000012	DMKCNS									
RDEVRCVY	000017	DMKCPO	DMKCPW	DMKDIB	DMKNES	DMKNLD	DMKNLE	DMKRNH	DMKVCH	DMKVDC	DMKVDS
RDEVRDC	000043	DMKALO	DMKBIO	DMKCCF	DMKCCW	DMKCKM	DMKCPW	DMKDAU	DMKDMP	DMKDSB	DMKHVE
		DMKIOC	DMKIOJ	DMKLNK	DMKMNT	DMKPAG	DMKPAH	DMKPGU	DMKPMA	DMKRSP	DMKRSQ
		DMKSPK	DMKTDK	DMKUNT	DMKVDG	DMKVSC	DMKVSX	DMKZTD			
RDEVRDY	000003	DMKIOS	DMKIOT								
RDEVREAD	000016	DMKGRD	DMKGRF	DMKGRG	DMKVCP	DMKVCR	DMKVCU				
RDEVREST	000004	DMKCNS									
RDEVREW	000002	DMKCPS									
RDEVRRES	000021	DMKACR	DMKCCS	DMKCFR	DMKCPM	DMKCPN	DMKCPW	DMKCPZ	DMKDAD	DMKDAS	DMKDAU
		DMKDSB									
RDEVRSTA	000018	DMKDID	DMKIOS	DMKIOT							
RDEVRSTR	000018	DMKCSF	DMKRSE	DMKRSP							
RDEVRSVD	000020	DMKCPO	DMKCPW	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKRGA	DMKRGB	DMKRNH	DMKVCH
		DMKVDC	DMKVDS								
RDEVRUN	000023	DMKDIB	DMKGRD	DMKGRF	DMKGRG	DMKGRH	DMKOPE				
RDEVSADN	000005	DMKCCT	DMKCNS	DMKIOC	DMKNES						
RDEVSCHD	000046	DMKACS	DMKCFQ	DMKCNS	DMKCPO	DMKCPW	DMKCSO	DMKDID	DMKDSP	DMKGRF	DMKHPU
		DMKIOQ	DMKIOS	DMKIOT	DMKPAG	DMKQVM	DMKRNH	DMKVCH			
RDEVSCRL	000008	DMKBLD	DMKCFT	DMKCQU	DMKTTX	DMKTTY					
RDEVSDR	000003	DMKNES									
RDEVSEL	000017	DMKCPW	DMKDSB	DMKLNK	DMKSSS	DMKSST	DMKSSV				
RDEVSENS	000006	DMKIOS	DMKIOT								
RDEVSEP	000012	DMKCKH	DMKCSO	DMKRSP	DMKSEP	DMKURS	DMKWRM				
RDEVSEPF	000004	DMKSEP	DMKTCT								
RDEVSER	000104	DMKACR	DMKALO	DMKCFG	DMKCKT	DMKCPW	DMKCPZ	DMKCQG	DMKCQP	DMKDAD	DMKDAS
		DMKDAU	DMKDEI	DMKDSB	DMKHVC	DMKIOE	DMKIOS	DMKLNK	DMKLOJ	DMKMNI	DMKMNT
		DMKMOO	DMKSCN	DMKSSS	DMKSST	DMKSSV	DMKTPE	DMKVDA	DMKVDD	DMKVDE	DMKVDG
		DMKVDH	DMKVDT	DMKVER	DMKVRR	DMKVRS	DMKWRM				
RDEVSIZE	000032	DMKCCS	DMKCPI	DMKDID	DMKDSP	DMKHPS	DMKHPT	DMKLNK	DMKLOJ	DMKMNL	DMKPAG
		DMKQVM	DMKSCN	DMKSSP	DMKSSS	DMKSSV	DMKVCT	DMKVCW	DMKVCX	DMKVDG	DMKVRS
RDEVSKUP	000004	DMKIOQ	DMKMON								
RDEVSLOW	000009	DMKCQP	DMKNES	DMKRNH							
RDEVSNA	000071	DMKBLD	DMKCFC	DMKCFM	DMKCFQ	DMKCFT	DMKCFW	DMKCQI	DMKCQU	DMKDIA	DMKDIB
		DMKGRC	DMKJRL	DMKLOG	DMKLOH	DMKLOM	DMKMON	DMKQCN	DMKQCO	DMKQCP	DMKQCQ
		DMKRGE	DMKTRD	DMKUSQ	DMKVCN	DMKVCT	DMKVCW	DMKVCX	DMKVDS		
RDEVSNRB	000101	DMKACO	DMKBLD	DMKCFM	DMKCFT	DMKCKF	DMKCQG	DMKCQQ	DMKCQS	DMKCQT	DMKCQU
		DMKCQY	DMKDIA	DMKDIB	DMKDIF	DMKDSP	DMKJRL	DMKLOG	DMKLOH	DMKLOM	DMKQCO
		DMKQCP	DMKQCQ	DMKTRD	DMKUSQ	DMKVCN	DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCU
		DMKVCV	DMKVCW	DMKVCX	DMKVDS						
RDEVSPAC	000004	DMKCSF	DMKCSO	DMKRSP							
RDEVSPL	000044	DMKCKH	DMKCKR	DMKCPO	DMKCPW	DMKCSB	DMKCSF	DMKCSO	DMKRSE	DMKRSP	DMKRST
		DMKSPL	DMKURS	DMKVCH	DMKVDA	DMKVDC	DMKVDS	DMKWRM	DMKWRN		
RDEVSPT	000005	DMKIOT	DMKSPS	DMKSPT							
RDEVSTAT	000344	DMKACR	DMKACS	DMKCAC	DMKCCH	DMKCFQ	DMKCFU	DMKCKD	DMKCKH	DMKCKN	DMKCKR
		DMKCKS	DMKCKV	DMKCNS	DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPS	DMKCPT	DMKCPV
		DMKCPW	DMKCPZ	DMKCQP	DMKCQT	DMKCSB	DMKCSF	DMKCSO	DMKDAD	DMKDAS	DMKDAU
		DMKDEI	DMKDIA	DMKDIB	DMKDID	DMKDIF	DMKDMP	DMKDSB	DMKDSP	DMKGRG	DMKHPT
		DMKHPU	DMKIDU	DMKIOE	DMKIOQ	DMKIOS	DMKIOT	DMKLNK	DMKLOJ	DMKMCC	DMKMNI
		DMKMNL	DMKMNT	DMKMSW	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKQVM	DMKRGA
		DMKRGB	DMKRGC	DMKRNH	DMKRSE	DMKRSP	DMKRST	DMKSCN	DMKSPT	DMKSSS	DMKSST
		DMKTAP	DMKTPE	DMKURS	DMKVCH	DMKVCN	DMKVDA	DMKVDC	DMKVDD	DMKVDE	DMKVDR
		DMKVDS	DMKVDT	DMKVER	DMKVRR	DMKVRS	DMKWRM	DMKZTD			
RDEVSTA2	000072	DMKACR	DMKACS	DMKCCS	DMKCNS	DMKCPM	DMKCPN	DMKCPT	DMKCPW	DMKCPZ	DMKCSO
		DMKDID	DMKGRF	DMKIOS	DMKIOT	DMKMNT	DMKNLD	DMKQCO	DMKQVM	DMKRSP	DMKTCS
		DMKTCT	DMKURS	DMKVCH	DMKVRR	DMKVRS	DMKVSC				

LABEL	COUNT	REFERENCES									
RDEVSTA3	000084	DMKACO	DMKBLD	DMKCCD	DMKCCW	DMKCFQ	DMKCFR	DMKCKF	DMKCNS	DMKCPT	DMKCQG
		DMKCQP	DMKCQQ	DMKCQS	DMKCQY	DMKDIA	DMKDIB	DMKDID	DMKDIF	DMKEXT	DMKGRD
		DMKGRE	DMKGRF	DMKGRG	DMKHPS	DMKIOS	DMKIOT	DMKJRL	DMKLOG	DMKLOH	DMKLOJ
		DMKMNT	DMKPRV	DMKQCP	DMKSCN	DMKSSP	DMKTAP	DMKUSQ	DMKVDA	DMKVDE	DMKVDS
		DMKVIO	DMKVSI	DMKVSJ	DMKZTD						
RDEVSTA4	000160	DMKACR	DMKACS	DMKCCH	DMKCCS	DMKCFQ	DMKCFR	DMKCNS	DMKCPB	DMKCPM	DMKCPN
		DMKCPO	DMKCPW	DMKCPZ	DMKCSO	DMKDAD	DMKDAS	DMKDAU	DMKDID	DMKDIF	DMKDSB
		DMKDSP	DMKENT	DMKGRF	DMKHPS	DMKHPT	DMKHPU	DMKIOQ	DMKIOS	DMKIOT	DMKNES
		DMKNLD	DMKPAG	DMKQVM	DMKRNH	DMKRSE	DMKRSP	DMKVCH	DMKVDA		
RDEVSTA5	000139	DMKACS	DMKALO	DMKCAC	DMKCAO	DMKCFH	DMKCKM	DMKCKN	DMKCPM	DMKCPN	DMKCPS
		DMKCPT	DMKCPW	DMKCPZ	DMKCQG	DMKCQP	DMKCQQ	DMKDAS	DMKDEF	DMKDGD	DMKDSB
		DMKGIO	DMKGRF	DMKIOS	DMKIOT	DMKLNK	DMKLOJ	DMKMNL	DMKMNT	DMKPAG	DMKVCH
		DMKVDA	DMKVDB	DMKVDC	DMKVDD	DMKVDE	DMKVDF	DMKVDG	DMKVDT	DMKVRR	
RDEVSTA6	000052	DMKBIO	DMKCCO	DMKCCS	DMKCCW	DMKCPW	DMKDGD	DMKDSB	DMKGRF	DMKMNL	DMKMNT
		DMKNES	DMKSSU	DMKUSQ	DMKVCH	DMKVDA	DMKVDE	DMKVRR	DMKVRS	DMKVSC	
RDEVSTA7	000013	DMKDID	DMKGRF	DMKIOS	DMKIOT	DMKMNL	DMKMNT	DMKVDE			
RDEVSTMD	000004	DMKCFR	DMKIOS	DMKTAP							
RDEVSYNC	000009	DMKCNS									
RDEVSYS	000045	DMKALO	DMKCFU	DMKCPM	DMKCPO	DMKCPW	DMKCPZ	DMKCQP	DMKCQQ	DMKDAS	DMKDAU
		DMKDSB	DMKLOJ	DMKMCC	DMKMNI	DMKSPS	DMKSPT	DMKSSS	DMKSST	DMKVCH	DMKVDA
		DMKVDC	DMKVDD	DMKVDH	DMKVDR	DMKVDS	DMKVDT				
RDEVTALY	000006	DMKVCH									
RDEVTAPL	000007	DMKCFT	DMKCQU	DMKQCQ	DMKTTX	DMKVCQ					
RDEVTBTU	000003	DMKNES	DMKRNH								
RDEVTCTL	000002	DMKNES									
RDEVTERM	000016	DMKCSF	DMKCSO	DMKRSE	DMKRSP						
RDEVTEXT	000015	DMKCFT	DMKCQU	DMKGRF	DMKGRI	DMKVCN	DMKVCR	DMKVCU	DMKVCV		
RDEVTFLG	000192	DMKACS	DMKBLD	DMKCFT	DMKCNS	DMKCPV	DMKCQU	DMKDIB	DMKDIF	DMKGRD	DMKGRE
		DMKGRF	DMKGRG	DMKGRH	DMKGRI	DMKHPT	DMKLOH	DMKNLD	DMKOPE	DMKQCQ	DMKQVM
		DMKTRM	DMKTTX	DMKTTY	DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCU	DMKVCV
		DMKVCX	DMKVDA	DMKVDS							
RDEVTMAT	000010	DMKACO	DMKCKF	DMKDIB	DMKDIF	DMKVDR	DMKVDS	DMKVDT			
RDEVTMCD	000079	DMKBLD	DMKCFT	DMKCNS	DMKCQU	DMKGRF	DMKGRG	DMKGRI	DMKHVE	DMKIOC	DMKNES
		DMKNLD	DMKOPE	DMKQCQ	DMKTRM	DMKTTX	DMKTTY	DMKVCN	DMKVCP	DMKVCQ	DMKVCR
		DMKVCU	DMKVCV	DMKVCW							
RDEVTRQ	000029	DMKDIF	DMKGRD	DMKGRF	DMKGRG	DMKGRI	DMKHPT	DMKLOH	DMKVCR	DMKVCS	DMKVCT
		DMKVCV	DMKVCX								
RDEVTTYB	000018	DMKCNS	DMKTTX	DMKTTY							
RDEVTYPC	000745	DMKACO	DMKACR	DMKACS	DMKALO	DMKATS	DMKBLD	DMKCCH	DMKCCS	DMKCCT	DMKCCW
		DMKCFC	DMKCFG	DMKCFH	DMKCFM	DMKCFS	DMKCFT	DMKCFU	DMKCFY	DMKCKD	DMKCKF
		DMKCKH	DMKCKM	DMKCKN	DMKCKS	DMKCKT	DMKCKV	DMKCNS	DMKCPJ	DMKCPM	DMKCPN
		DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPV	DMKCPW	DMKCQG	DMKCQP	DMKCQQ	DMKCQR
		DMKCQS	DMKCQT	DMKCQU	DMKCQY	DMKCSB	DMKCSF	DMKCSO	DMKDEI	DMKDIA	DMKDIB
		DMKDID	DMKDIF	DMKDMP	DMKDMQ	DMKDRD	DMKDRE	DMKDSB	DMKDSP	DMKEPS	DMKERM
		DMKEXT	DMKGRC	DMKGRG	DMKGRT	DMKHPS	DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKIDU
		DMKIOC	DMKIOE	DMKIOF	DMKIOG	DMKIOJ	DMKIOQ	DMKIOS	DMKIOT	DMKJRL	DMKLNK
		DMKLOG	DMKLOH	DMKLOJ	DMKLOM	DMKMCC	DMKMNI	DMKMNT	DMKMON	DMKMOO	DMKMSW
		DMKNES	DMKNET	DMKNLD	DMKNLE	DMKOPE	DMKOPR	DMKPAG	DMKPGT	DMKPGU	DMKQCN
		DMKQCO	DMKQCP	DMKQCQ	DMKQVM	DMKRGA	DMKRNH	DMKRSE	DMKRSP	DMKRSQ	DMKSCN
		DMKSEG	DMKSNC	DMKSPK	DMKSPS	DMKSPT	DMKSSP	DMKSSS	DMKSST	DMKTCS	DMKTDK
		DMKUDR	DMKURS	DMKUSQ	DMKVCH	DMKVCN	DMKVCP	DMKVCR	DMKVCS	DMKVDA	DMKVDC
		DMKVDD	DMKVDE	DMKVDF	DMKVDG	DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVRR	DMKVSC
		DMKVSX	DMKWRM	DMKZTD							
RDEVTYPE	000963	DMKACO	DMKACR	DMKACS	DMKALO	DMKATS	DMKBLD	DMKCCS	DMKCCT	DMKCFC	DMKCFG
		DMKCFH	DMKCFM	DMKCFQ	DMKCFS	DMKCFT	DMKCFU	DMKCFY	DMKCKD	DMKCKF	DMKCKH
		DMKCKN	DMKCKS	DMKCKT	DMKCKV	DMKCNS	DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPP
		DMKCPS	DMKCPT	DMKCPV	DMKCPW	DMKCQG	DMKCQP	DMKCQQ	DMKCQS	DMKCQU	DMKCQY
		DMKCSB	DMKCSC	DMKCSF	DMKCSO	DMKDAD	DMKDAS	DMKDAU	DMKDIA	DMKDIB	DMKDID

LABEL	COUNT	REFERENCES									
		DMKDIF	DMKDMP	DMKDMQ	DMKDRD	DMKDRE	DMKDSB	DMKGRD	DMKGRF	DMKGRG	DMKGRI
		DMKHPS	DMKHPT	DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKIOC	DMKIOE	DMKIOF	DMKIOG
		DMKIOJ	DMKIOS	DMKIOT	DMKJRL	DMKLNK	DMKLOG	DMKLOH	DMKLOM	DMKMCC	DMKMNT
		DMKMSW	DMKNES	DMKNLD	DMKNLE	DMKOPE	DMKOPR	DMKPAG	DMKPGT	DMKQCN	DMKQCO
		DMKQCP	DMKQCQ	DMKQVM	DMKRGA	DMKRSE	DMKRSF	DMKRSP	DMKRST	DMKSEG	DMKSEP
		DMKSNC	DMKSPK	DMKSPS	DMKSPT	DMKSSP	DMKSSS	DMKTAP	DMKTAQ	DMKTCS	DMKTCT
		DMKTDK	DMKURS	DMKUSQ	DMKVCH	DMKVCN	DMKVCP	DMKVCQ	DMKVCR	DMKVCU	DMKVDA
		DMKVDC	DMKVDE	DMKVDF	DMKVDG	DMKVDH	DMKVDR	DMKVDS	DMKVER	DMKWRM	DMKZTD
RDEVUNF	000004	DMKCPW									
RDEVUNSN	000007	DMKIOS	DMKIOT								
RDEVUSC8	000007	DMKBLD	DMKCFT	DMKCNS	DMKNES	DMKVCP					
RDEVUSER	000105	DMKACS	DMKBLD	DMKCCT	DMKCFU	DMKCKD	DMKCKH	DMKCNS	DMKCPI	DMKCPN	DMKCPO
		DMKCPT	DMKCPV	DMKCPZ	DMKCQP	DMKCQS	DMKCQT	DMKCSB	DMKCSF	DMKCSO	DMKDIA
		DMKDIF	DMKDMP	DMKEXT	DMKGRD	DMKGRF	DMKGRI	DMKHPS	DMKHPT	DMKHPU	DMKIOT
		DMKLOH	DMKMCC	DMKMNT	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKOPE	DMKOPR	DMKPAG
		DMKQVM	DMKSPS	DMKSPT	DMKSSS	DMKUSQ	DMKVCH	DMKVCP	DMKVCV	DMKVCW	DMKVDA
		DMKVDD	DMKVDE	DMKVDF	DMKVDG	DMKVDR	DMKVDS	DMKVDT	DMKVRR	DMKVRS	
RDEVVMNT	000008	DMKDSB	DMKLOJ	DMKMNT	DMKSSV	DMKVDA	DMKVDR				
RDEVVM2	000007	DMKBLD	DMKCFT	DMKCNS	DMKNES	DMKVCP					
RDEVVM2I	000004	DMKTTX	DMKVCQ								
RDEVWAII	000023	DMKRGA	DMKRGB	DMKRGC							
RDEVWAIT	000008	DMKNES	DMKRNH								
RDEVWIOB	000021	DMKGRD	DMKGRF	DMKGRG	DMKGRI						
RDEVWSF	000006	DMKGRE	DMKGRF	DMKGRG							
RDEVWTH	000022	DMKBLD	DMKERM	DMKGRC	DMKGRD	DMKGRF	DMKGRG	DMKGRT	DMKHPS	DMKHVE	DMKQCN
		DMKSSP	DMKVCN	DMKVCP							
RDEVXSEP	000008	DMKCKH	DMKCKS	DMKCKV	DMKCSO	DMKTCS	DMKURS	DMKWRM			
RDEV12B	000006	DMKGRC	DMKHPS								
RDEV14AD	000002	DMKGRC	DMKHVE								
RDEV14B	000011	DMKGRC	DMKGRD	DMKGRT							
RDEV3101	000016	DMKBLD	DMKCFT	DMKCQU	DMKTTX	DMKTTY					
RDEV333V	000011	DMKIOS	DMKIOT	DMKLOJ	DMKLOM	DMKPAG	DMKVDA	DMKVDR	DMKVRR		
RDEV741D	000004	DMKGRD	DMKGRE	DMKGRF							
RDIDX	000054	DMKACR	DMKACS	DMKCCH	DMKCFS	DMKCFU	DMKCKD	DMKCKF	DMKCKH	DMKCKM	DMKCKN
		DMKCKV	DMKCPI	DMKCPJ	DMKCPP	DMKCPZ	DMKCSO	DMKDIF	DMKDMP	DMKGRG	DMKMCT
		DMKMNI	DMKMNT	DMKMOO	DMKNES	DMKNET	DMKOPR	DMKPGU	DMKSCN	DMKSPS	DMKVCH
		DMKVDA	DMKVDG	DMKVDT	DMKWRM	DMKWRN					
RDPTQRYG	000002	DMKGRF	DMKGRG								
RDRCHN	000030	DMKACO	DMKCKR	DMKCKS	DMKDMP	DMKMIA	DMKNLE	DMKSPL	DMKSPS	DMKTRT	DMKTRU
		DMKVME	DMKVSD	DMKVSE	DMKWRM						
READ	000133	DMKCCD	DMKCKP	DMKDAU	DMKDDR	DMKDGD	DMKDIR	DMKFMT	DMKLD00E	DMKPAG	DMKSPK
READMOD	000001	DMKVCN									
READNRM	000002	DMKRNH									
RECBAK	000015	DMKPGT	DMKPGU	DMKPST	DMKUSP	DMKVDG	DMKVDH	DMKXST			
RECBLOK	000101	DMKCKF	DMKCKM	DMKCKT	DMKCKV	DMKDMP	DMKIDU	DMKMOO	DMKPGT	DMKPGU	DMKPST
		DMKRSP	DMKRSQ	DMKSPK	DMKUSP	DMKVDG	DMKVDH	DMKVSX	DMKWRM	DMKXST	
RECBUFF	000007	DMKCKF	DMKCKH	DMKWRM							
RECCCPD	000004	DMKIOF	DMKIOG								
RECCYL	000070	DMKCKF	DMKCKM	DMKCKT	DMKCKV	DMKDMP	DMKIDU	DMKMOO	DMKPGT	DMKPGU	DMKPST
		DMKRSP	DMKRSQ	DMKVDG	DMKVDH	DMKVSX	DMKWRM	DMKXST			
RECDUMP	000001	DMKCKF									
RECFLAG1	000021	DMKERP	DMKIOF	DMKIOG	DMKIOH						
RECFLAG2	000004	DMKIOG									
RECFLG	000004	DMKCKF									
RECFSZ	000003	DMKCKF	DMKWRM								
RECFULL	000001	DMKCKF									
RECMAP	000083	DMKCKT	DMKCKV	DMKDMP	DMKIDU	DMKPGT	DMKPGU	DMKPST	DMKRSP	DMKRSQ	DMKVDH
		DMKVSX	DMKXST								

LABEL	COUNT	REFERENCES									
RECMAX	000034	DMKCKF	DMKCKT	DMKCKV	DMKDMP	DMKIDU	DMKPGT	DMKPGU	DMKPST	DMKVDH	DMKXST
RECMODE	000005	DMKIOG	DMKMCH	DMKMCI							
RECNXT	000018	DMKERP	DMKIOF	DMKIOG	DMKIOH						
RECOVRPT	000008	DMKCPJ	DMKMCH	DMKMCI							
RECPAG	000008	DMKERP	DMKIOF	DMKIOG	DMKIOH	DMKIOJ					
RECPAGDN	000001	DMKIOG									
RECPAGFA	000007	DMKIOG	DMKIOH								
RECPAGFL	000009	DMKERP	DMKIOF	DMKIOG	DMKIOH						
RECPAGFM	000003	DMKIOG									
RECPAGFR	000002	DMKIOG									
RECPAGIU	000003	DMKIOF	DMKIOG								
RECPNT	000066	DMKCKF	DMKCKT	DMKCKV	DMKDMP	DMKIDU	DMKPGT	DMKPGU	DMKPST	DMKRSP	DMKRSQ
		DMKSPK	DMKUSP	DMKVDG	DMKVDH	DMKVSX	DMKWRM	DMKXST			
RECQMSG	000010	DMKIUB	DMKIUE	DMKIUL	DMKIUN	DMKIUS					
RECSIZE	000040	DMKCKT	DMKIDU	DMKPGT	DMKPGU	DMKPST	DMKRSP	DMKRSQ	DMKSPK	DMKUSP	DMKVDH
		DMKVSX	DMKWRM	DMKXST							
RECUSED	000050	DMKCKF	DMKCKT	DMKCKV	DMKDMP	DMKIDU	DMKPGT	DMKPGU	DMKPST	DMKRSP	DMKRSQ
		DMKVDG	DMKVDH	DMKVSX	DMKWRM	DMKXST					
REGNCODE	000002	DMKPMA									
REGSAV	000006	DMKFMT	DMKSAV								
REMOTE	000007	DMKACO	DMKCKF								
RENAME	000001	DMKIMG									
REPCNT	000001	DMKERM									
REQUSER	000001	DMKSWM									
RESET	000027	DMKCFC	DMKCNS	DMKCPI	DMKEXT	DMKFMT	DMKMCT	DMKMID	DMKPEQ	DMKQCN	DMKTRP
		DMKTRR	DMKVDE	DMKVDS	DMKVMD						
RESINCHA	000002	DMKCCS	DMKCCW								
RETBUF	000006	DMKCFY	DMKQCP	DMKRET							
RETDATA	000004	DMKRET									
RETDLEN	000004	DMKCFY	DMKQCP	DMKRET							
RETLEN1	000012	DMKRET									
RETLEN2	000004	DMKRET									
RETL132	000001	DMKCFY									
RETNXTD	000004	DMKRET									
RETNXTS	000004	DMKRET									
RETRY	000092	DMKACO	DMKCKF								
RETRYSW	000003	DMKCNS									
REW	000001	DMKGRC									
REWIND	000007	DMKCFC	DMKCPB	DMKSPT	DMKTAP	DMKTAQ					
RHTBLOK	000029	DMKCKR	DMKCKT	DMKCKV	DMKCKW	DMKVSD	DMKVSE	DMKVSF	DMKVSG	DMKWRM	DMKWRN
RHTECNT	000005	DMKCKW	DMKVSD	DMKWRM							
RHTFLAG	000002	DMKVSE	DMKVSG								
RHTINDEX	000005	DMKVSD	DMKVSE								
RHTLEN	000009	DMKVSD	DMKVSE								
RHTNUMEN	000002	DMKVSD	DMKVSE								
RHTPAGNO	000004	DMKVSD	DMKVSE								
RHTRSRV	000016	DMKVSD	DMKVSE	DMKVSG							
RHTSFB1	000008	DMKVSD	DMKVSE								
RHTSPRF	000002	DMKVSE	DMKVSG								
RHTSPU1	000028	DMKVSD	DMKVSE	DMKVSG							
RHTSPU3	000001	DMKVSE									
RHTVIRT	000008	DMKCKR	DMKCKT	DMKCKV	DMKCKW	DMKVSF	DMKWRN				
RHTVRFY	000001	DMKWRM									
RHXADDR	000004	DMKCKW	DMKWRM								
RHXTABLE	000003	DMKCKW	DMKWRM								
RM	000053	DMKGRD	DMKVCR								
ROUTE	000006	DMKDIR									
RPYQMSG	000007	DMKIUB	DMKIUE	DMKIUG	DMKIUL	DMKIUS					

LABEL	COUNT	REFERENCES									
RQLKBASE	000002	DMKDSP	DMKWAI								
RQLOCK	000015	DMKAPI	DMKDSP	DMKFRE	DMKPXA	DMKPXB	DMKSTK	DMKTRQ			
RSPBF1DC	000003	DMKRSP									
RSPBF1IO	000009	DMKCSO	DMKRSP								
RSPBF1VL	000011	DMKRSP									
RSPBF2DC	000003	DMKRSP									
RSPBF2IO	000008	DMKCSO	DMKRSP								
RSPBF2VL	000009	DMKRSP									
RSPCLPRT	000004	DMKRSP									
RSPDEFER	000002	DMKRSP									
RSPDPAGE	000028	DMKRSP	DMKRST	DMKSPL	DMKTCS						
RSPDPAG2	000007	DMKRSP									
RSPERR	000007	DMKRST									
RSPFLAG1	000045	DMKCSO	DMKRSP								
RSPFLAG2	000087	DMKCSF	DMKCSO	DMKRSP	DMKRST						
RSPIMIDL	000003	DMKRSP									
RSPLCTL	000025	DMKCKH	DMKCSF	DMKCSO	DMKRSP	DMKRSQ	DMKRST	DMKSPL	DMKTCS	DMKTCT	DMKURS
RSPMISC	000006	DMKCSF	DMKRSP								
RSPRPAGE	000047	DMKRSP	DMKRST	DMKSPL	DMKTCS	DMKTCT					
RSPRPAG2	000011	DMKRSP	DMKTCS	DMKTCT							
RSPRSTRT	000007	DMKRSP									
RSPSEP	000012	DMKRSP									
RSPSFBLK	000029	DMKCKH	DMKCSF	DMKCSO	DMKRSP	DMKRST	DMKSPL	DMKTCT	DMKURS		
RSPSFLOK	000072	DMKCSF	DMKCSO	DMKRSP							
RSPSIZE	000013	DMKRSP	DMKRST	DMKSPL							
RSPSWAP	000010	DMKRST									
RSPVPAGE	000012	DMKRSP	DMKRST	DMKSPL	DMKTCS						
RSPVPAG2	000026	DMKRSP									
RSPVPG2	000003	DMKRST									
RSPXAUTO	000009	DMKCKV	DMKCSO	DMKRSP	DMKURS	DMKWRM					
RSPXBLOK	000055	DMKCKH	DMKCKS	DMKCKV	DMKCPW	DMKCSB	DMKCSF	DMKCSO	DMKMNT	DMKRSP	DMKSEP
		DMKTCS	DMKTCT	DMKURS	DMKWRM						
RSPXCHR	000003	DMKTCS	DMKTCT								
RSPXCHR1	000002	DMKTCS	DMKTCT								
RSPXCHR2	000002	DMKTCS	DMKTCT								
RSPXCHR3	000002	DMKTCS	DMKTCT								
RSPXCMOD	000003	DMKTCS	DMKTCT								
RSPXDEST	000010	DMKCKH	DMKCKS	DMKCSO	DMKRSP	DMKWRM					
RSPXDST1	000002	DMKRSP	DMKURS								
RSPXDST2	000002	DMKMNT	DMKURS								
RSPXDST3	000002	DMKMNT	DMKURS								
RSPXDST4	000001	DMKURS									
RSPXFCB	000012	DMKCKH	DMKCKS	DMKCKV	DMKCSB	DMKRSP	DMKTCS	DMKTCT	DMKWRM		
RSPXFILE	000004	DMKRSP									
RSPXFLAG	000053	DMKCKH	DMKCKS	DMKCKV	DMKCPW	DMKCSB	DMKCSF	DMKCSO	DMKRSP	DMKURS	DMKWRM
RSPXFMNT	000008	DMKCPW	DMKCSO	DMKRSP							
RSPXFOLD	000005	DMKCKV	DMKCSB	DMKWRM							
RSPXFORM	000014	DMKCKH	DMKCKS	DMKCKV	DMKCSO	DMKMNT	DMKRSP	DMKURS	DMKWRM		
RSPXFPND	000012	DMKCPW	DMKCSO	DMKRSP	DMKURS						
RSPXINDX	000007	DMKCKH	DMKCKS	DMKCKV	DMKCSB	DMKWRM					
RSPXMCHR	000002	DMKTCS									
RSPXNOPL	000003	DMKRSP									
RSPXOTRC	000004	DMKTCT									
RSPXPMNT	000007	DMKCPW	DMKCSO	DMKRSP	DMKURS						
RSPXRECT	000013	DMKCSO	DMKRSP	DMKURS							
RSPXSEQ	000006	DMKRSP	DMKSEP	DMKURS							
RSPXSETU	000009	DMKCKV	DMKCSO	DMKRSP	DMKURS	DMKWRM					
RSPXSFIL	000013	DMKCSF	DMKCSO	DMKRSP	DMKURS						

LABEL	COUNT	REFERENCES									
RSPXSIZE	000002	DMKMNT									
RSPXVTRC	000004	DMKRSP	DMKTCS								
RSRTNPSW	000022	DMKAPI	DMKCCH	DMKCKD	DMKDMP	DMKDMQ	DMKMCT	DMKQVM	DMKVRS		
RSRTOPSW	000002	DMKMPO									
RTCODE0	000002	DMKEIG									
RTCODE1	000004	DMKEIG	DMKSEV	DMKSIX							
RTCODE2	000004	DMKEIG	DMKSEV	DMKSIX							
RTCODE3	000007	DMKEIG	DMKSEV	DMKSIX							
RTCODE4	000010	DMKEIG	DMKSEV	DMKSIX							
RTCODE5	000004	DMKEIG	DMKSEV	DMKSIX							
RTCODE7	000004	DMKSEV									
RTNBLOCK	000001	DMKRGD									
RTNNOCTL	000002	DMKRGD									
RTYPE	000003	DMKQCN	DMKQCQ								
RUN	000005	DMKCFJ	DMKCMD	DMKSVD	DMKTEE	DMKTRA					
RUNCRINV	000007	DMKDSP	DMKPRG								
RUNCR0	000072	DMKAPI	DMKCPI	DMKDSP	DMKEXT	DMKFPS	DMKIOT	DMKPMA	DMKPRG	DMKPRV	DMKSTA
		DMKSVD	DMKTRX	DMKVFR							
RUNCR1	000020	DMKAPI	DMKCPI	DMKDSP	DMKFPS	DMKPMA	DMKPRG	DMKPRV	DMKSVD		
RUNPSW	000048	DMKAPI	DMKDSP	DMKEXT	DMKFPS	DMKPMA	DMKPRG	DMKSVD	DMKWAI		
RUNUSER	000050	DMKAPI	DMKCDS	DMKCFP	DMKCPI	DMKDIB	DMKDIF	DMKDSP	DMKEXT	DMKIOS	DMKIOT
		DMKLOH	DMKMCH	DMKMCT	DMKMPO	DMKPMA	DMKPRG	DMKSCH	DMKSEL	DMKSVD	DMKTHI
		DMKUSQ	DMKVCA								
RUN370E	000005	DMKDSP	DMKPTT								
R0	017418	DMKACO	DMKACR	DMKACS	DMKALG	DMKALO	DMKAPI	DMKAPS	DMKAPT	DMKAPU	DMKAPV
		DMKAPW	DMKAPX	DMKAPY	DMKAPZ	DMKATS	DMKBIO	DMKBLD	DMKBSC	DMKCAC	DMKCAO
		DMKCCD	DMKCCF	DMKCCH	DMKCCO	DMKCCS	DMKCCT	DMKCCW	DMKCDB	DMKCDM	DMKCDS
		DMKCFC	DMKCFD	DMKCFF	DMKCFG	DMKCFH	DMKCFJ	DMKCFM	DMKCFO	DMKCFP	DMKCFQ
		DMKCFR	DMKCFS	DMKCFT	DMKCFU	DMKCFV	DMKCFW	DMKCFY	DMKCKD	DMKCKF	DMKCKH
		DMKCKM	DMKCKN	DMKCKP	DMKCKR	DMKCKS	DMKCKT	DMKCKV	DMKCKW	DMKCLK	DMKCNS
		DMKCNT	DMKCPB	DMKCPI	DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT
		DMKCPU	DMKCPV	DMKCPW	DMKCPX	DMKCPY	DMKCPZ	DMKCQC	DMKCQG	DMKCQH	DMKCQI
		DMKCQP	DMKCQQ	DMKCQR	DMKCQS	DMKCQT	DMKCQU	DMKCQY	DMKCRM	DMKCSB	DMKCSC
		DMKCSF	DMKCSO	DMKCSP	DMKCSQ	DMKCSR	DMKCST	DMKCSU	DMKCSV	DMKCSW	DMKCSX
		DMKCSY	DMKCVT	DMKCVU	DMKDAD	DMKDAS	DMKDAU	DMKDDC	DMKDDR	DMKDEF	DMKDEG
		DMKDEI	DMKDEX	DMKDGD	DMKDGF	DMKDIA	DMKDIB	DMKDID	DMKDIF	DMKDIR	DMKDMP
		DMKDMQ	DMKDNC	DMKDRD	DMKDRE	DMKDSB	DMKDSP	DMKEIG	DMKENT	DMKEPS	DMKERM
		DMKERP	DMKEXT	DMKFMT	DMKFPS	DMKFRE	DMKFRT	DMKGIO	DMKGRA	DMKGRC	DMKGRD
		DMKGRE	DMKGRF	DMKGRG	DMKGRH	DMKGRI	DMKGRT	DMKHPS	DMKHPT	DMKHPU	DMKHVC
		DMKHVD	DMKHVE	DMKHVF	DMKIDR	DMKIDU	DMKIMG	DMKIOC	DMKIOE	DMKIOF	DMKIOG
		DMKIOH	DMKIOJ	DMKIOQ	DMKIOS	DMKIOT	DMKISM	DMKIUA	DMKIUB	DMKIUC	DMKIUE
		DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS	DMKJRL	DMKLDOOE	DMKLNK	DMKLNM
		DMKLOC	DMKLOG	DMKLOH	DMKLOJ	DMKLOK	DMKLOM	DMKMCC	DMKMCD	DMKMCH	DMKMCI
		DMKMCT	DMKMHC	DMKMHV	DMKMIA	DMKMID	DMKMNI	DMKMNJ	DMKMNL	DMKMNT	DMKMON
		DMKMOO	DMKMPO	DMKMSG	DMKMSW	DMKNEA	DMKNEM	DMKNES	DMKNET	DMKNLD	DMKNLE
		DMKNMT	DMKOPE	DMKOPR	DMKPAG	DMKPAH	DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPER
		DMKPET	DMKPGM	DMKPGS	DMKPGT	DMKPGU	DMKPMA	DMKPRG	DMKPRV	DMKPRW	DMKPSA
		DMKPST	DMKPTR	DMKPTS	DMKPTT	DMKQCN	DMKQCO	DMKQCP	DMKQCQ	DMKQVM	DMKREI
		DMKRET	DMKRGA	DMKRGB	DMKRGC	DMKRGD	DMKRGE	DMKRND	DMKRNH	DMKRPA	DMKRPD
		DMKRPI	DMKRPW	DMKRSE	DMKRSF	DMKRSP	DMKRSQ	DMKRST	DMKSAD	DMKSAV	DMKSBL
		DMKSCH	DMKSCN	DMKSCO	DMKSEG	DMKSEL	DMKSEP	DMKSEV	DMKSFB	DMKSIX	DMKSNC
		DMKSND	DMKSPC	DMKSPK	DMKSPL	DMKSPM	DMKSPR	DMKSPS	DMKSPT	DMKSRM	DMKSSP
		DMKSSS	DMKSST	DMKSSU	DMKSSV	DMKSTA	DMKSTK	DMKSTP	DMKSTR	DMKSVC	DMKSVD
		DMKSWA	DMKSWM	DMKTAP	DMKTAQ	DMKTCS	DMKTCT	DMKTDK	DMKTES	DMKTHI	DMKTMR
		DMKTOD	DMKTPE	DMKTRA	DMKTRC	DMKTRD	DMKTRK	DMKTRM	DMKTRP	DMKTRQ	DMKTRR
		DMKTRT	DMKTRU	DMKTRX	DMKTTX	DMKTTY	DMKUDR	DMKUDU	DMKUNT	DMKURS	DMKUSP
		DMKUSQ	DMKVAT	DMKVAU	DMKVBM	DMKVCA	DMKVCB	DMKVCH	DMKVCN	DMKVCP	DMKVCQ
		DMKVCR	DMKVCS	DMKVCT	DMKVCU	DMKVCV	DMKVCW	DMKVCX	DMKVDA	DMKVDB	DMKVDC

LABEL	COUNT	REFERENCES									
		DMKVDD	DMKVDE	DMKVDF	DMKVDG	DMKVDH	DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVFC
		DMKVFD	DMKVFE	DMKVFR	DMKVFS	DMKVIO	DMKVMA	DMKVMC	DMKVMD	DMKVME	DMKVMG
		DMKVMI	DMKVRR	DMKVRS	DMKVSC	DMKVSD	DMKVSE	DMKVSF	DMKVSG	DMKVSI	DMKVSJ
		DMKVSP	DMKVSQ	DMKVST	DMKVSU	DMKVSV	DMKVSW	DMKVSX	DMKWAI	DMKWRM	DMKWRN
		DMKXAB	DMKXAD	DMKXST	DMKZTD						
R1	031954	DMKACO	DMKACR	DMKACS	DMKALG	DMKALO	DMKAPI	DMKAPS	DMKAPT	DMKAPU	DMKAPV
		DMKAPW	DMKAPX	DMKAPY	DMKAPZ	DMKATS	DMKBIO	DMKBLD	DMKBSC	DMKCAC	DMKCAO
		DMKCCD	DMKCCF	DMKCCH	DMKCCO	DMKCCS	DMKCCT	DMKCCW	DMKCDB	DMKCDM	DMKCDS
		DMKCFC	DMKCFD	DMKCFF	DMKCFG	DMKCFH	DMKCFJ	DMKCFM	DMKCFO	DMKCFP	DMKCFQ
		DMKCFR	DMKCFS	DMKCFT	DMKCFU	DMKCFV	DMKCFW	DMKCFY	DMKCKD	DMKCKF	DMKCKH
		DMKCKM	DMKCKN	DMKCKP	DMKCKR	DMKCKS	DMKCKT	DMKCKV	DMKCKW	DMKCLK	DMKCNS
		DMKCNT	DMKCPB	DMKCPI	DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT
		DMKCPU	DMKCPV	DMKCPW	DMKCPX	DMKCPY	DMKCPZ	DMKCQC	DMKCQG	DMKCQH	DMKCQI
		DMKCQP	DMKCQQ	DMKCQR	DMKCQS	DMKCQT	DMKCQU	DMKCQY	DMKCRM	DMKCSB	DMKCSC
		DMKCSF	DMKCSO	DMKCSP	DMKCSQ	DMKCSR	DMKCST	DMKCSU	DMKCSV	DMKCSW	DMKCSX
		DMKCSY	DMKCVT	DMKCVU	DMKDAD	DMKDAS	DMKDAU	DMKDDC	DMKDDR	DMKDEF	DMKDEG
		DMKDEI	DMKDEX	DMKDGD	DMKDGF	DMKDIA	DMKDIB	DMKDID	DMKDIF	DMKDIR	DMKDMP
		DMKDMQ	DMKDNC	DMKDRD	DMKDRE	DMKDSB	DMKDSP	DMKEIG	DMKENT	DMKEPS	DMKERM
		DMKERP	DMKEXT	DMKFMT	DMKFPS	DMKFRE	DMKFRT	DMKGIO	DMKGRA	DMKGRC	DMKGRD
		DMKGRE	DMKGRF	DMKGRG	DMKGRH	DMKGRI	DMKGRT	DMKHPS	DMKHPT	DMKHPU	DMKHVC
		DMKHVD	DMKHVE	DMKHVF	DMKIDR	DMKIDU	DMKIMG	DMKIOE	DMKIOF	DMKIOG	DMKIOH
		DMKIOJ	DMKIOQ	DMKIOS	DMKIOT	DMKISM	DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUG
		DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS	DMKJRL	DMKLDOOE	DMKLNK	DMKLNM	DMKLOC
		DMKLOG	DMKLOH	DMKLOJ	DMKLOK	DMKLOM	DMKMCC	DMKMCD	DMKMCH	DMKMCI	DMKMCT
		DMKMHC	DMKMHV	DMKMIA	DMKMID	DMKMNI	DMKMNJ	DMKMNL	DMKMNT	DMKMON	DMKMOO
		DMKMPO	DMKMSG	DMKMSW	DMKNEA	DMKNEM	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKNMT
		DMKOPE	DMKOPR	DMKOVR	DMKPAG	DMKPAH	DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPER
		DMKPET	DMKPGM	DMKPGS	DMKPGT	DMKPGU	DMKPMA	DMKPRG	DMKPRV	DMKPRW	DMKPSA
		DMKPST	DMKPTR	DMKPTS	DMKPTT	DMKQCN	DMKQCO	DMKQCP	DMKQCQ	DMKQVM	DMKREI
		DMKRET	DMKRGA	DMKRGB	DMKRGC	DMKRGD	DMKRGE	DMKRND	DMKRNH	DMKRPA	DMKRPD
		DMKRPI	DMKRPW	DMKRSE	DMKRSF	DMKRSP	DMKRSQ	DMKRST	DMKSAD	DMKSAV	DMKSBL
		DMKSCH	DMKSCN	DMKSCO	DMKSEG	DMKSEL	DMKSEP	DMKSEV	DMKSFB	DMKSIX	DMKSNC
		DMKSND	DMKSPC	DMKSPK	DMKSPL	DMKSPM	DMKSPR	DMKSPS	DMKSPT	DMKSRM	DMKSSP
		DMKSSS	DMKSST	DMKSSU	DMKSSV	DMKSTA	DMKSTK	DMKSTP	DMKSTR	DMKSVC	DMKSVD
		DMKSWA	DMKSWM	DMKTAP	DMKTAQ	DMKTCS	DMKTCT	DMKTDK	DMKTEE	DMKTEF	DMKTEM
		DMKTES	DMKTHI	DMKTMR	DMKTOD	DMKTPE	DMKTRA	DMKTRC	DMKTRD	DMKTRK	DMKTRM
		DMKTRP	DMKTRQ	DMKTRR	DMKTRT	DMKTRU	DMKTRX	DMKTTX	DMKTTY	DMKUDR	DMKUDU
		DMKUNT	DMKURS	DMKUSP	DMKUSQ	DMKVAT	DMKVAU	DMKVBM	DMKVCA	DMKVCB	DMKVCH
		DMKVCN	DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCU	DMKVCV	DMKVCW	DMKVCX
		DMKVDA	DMKVDB	DMKVDC	DMKVDD	DMKVDE	DMKVDF	DMKVDG	DMKVDH	DMKVDR	DMKVDS
		DMKVDT	DMKVER	DMKVFC	DMKVFD	DMKVFE	DMKVFR	DMKVFS	DMKVIO	DMKVMA	DMKVMC
		DMKVMD	DMKVME	DMKVMG	DMKVMI	DMKVRR	DMKVRS	DMKVSC	DMKVSD	DMKVSE	DMKVSF
		DMKVSG	DMKVSI	DMKVSJ	DMKVSP	DMKVSQ	DMKVST	DMKVSV	DMKVSW	DMKVSX	DMKWAI
		DMKWRM	DMKWRN	DMKXAB	DMKXAD	DMKXST	DMKZTD				
R10	005725	DMKACO	DMKACR	DMKACS	DMKALG	DMKAPI	DMKAPS	DMKAPT	DMKAPU	DMKAPV	DMKAPY
		DMKATS	DMKBIO	DMKBLD	DMKBSC	DMKCAC	DMKCCD	DMKCCF	DMKCCH	DMKCCO	DMKCCS
		DMKCCW	DMKCDB	DMKCDM	DMKCDS	DMKCFD	DMKCFF	DMKCFG	DMKCFH	DMKCFO	DMKCFP
		DMKCFQ	DMKCFR	DMKCFS	DMKCFT	DMKCFU	DMKCFV	DMKCFW	DMKCFY	DMKCKD	DMKCKF
		DMKCKH	DMKCKP	DMKCKR	DMKCKV	DMKCNS	DMKCNT	DMKCPB	DMKCPI	DMKCPM	DMKCPN
		DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPU	DMKCPV	DMKCPW	DMKCPX	DMKCPY	DMKCPZ
		DMKCQC	DMKCQG	DMKCQH	DMKCQI	DMKCQP	DMKCQQ	DMKCQR	DMKCQS	DMKCQT	DMKCQY
		DMKCRM	DMKCSB	DMKCSC	DMKCSF	DMKCSO	DMKCSQ	DMKCSR	DMKCST	DMKCSV	DMKCSW
		DMKCSX	DMKCSY	DMKCVU	DMKDAD	DMKDAS	DMKDAU	DMKDDC	DMKDDR	DMKDEF	DMKDEG
		DMKDEI	DMKDEX	DMKDGD	DMKDGF	DMKDIA	DMKDIB	DMKDID	DMKDIF	DMKDIR	DMKDMP
		DMKDMQ	DMKDNC	DMKDRD	DMKDRE	DMKDSB	DMKDSP	DMKENT	DMKERM	DMKERP	DMKEXT
		DMKFMT	DMKFPS	DMKFRE	DMKFRT	DMKGIO	DMKGRA	DMKGRC	DMKGRD	DMKGRE	DMKGRF
		DMKGRG	DMKGRH	DMKGRI	DMKGRT	DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKHVD	DMKHVE
		DMKHVF	DMKIDR	DMKIDU	DMKIOC	DMKIOE	DMKIOF	DMKIOG	DMKIOH	DMKIOJ	DMKIOQ

LABEL	COUNT	REFERENCES									
		DMKIOS	DMKIOT	DMKISM	DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUJ	DMKIUL	DMKIUP
		DMKIUS	DMKJRL	DMKLD00E	DMKLNK	DMKLNM	DMKLOC	DMKLOG	DMKLOH	DMKLOJ	DMKLOK
		DMKLOM	DMKMCC	DMKMCD	DMKMCH	DMKMCT	DMKMHC	DMKMHV	DMKMIA	DMKMID	DMKMNI
		DMKMNJ	DMKMNL	DMKMNT	DMKMON	DMKMOO	DMKMPO	DMKMSG	DMKMSW	DMKNEA	DMKNES
		DMKNET	DMKNLD	DMKNLE	DMKOPE	DMKOPR	DMKOVR	DMKPAG	DMKPAH	DMKPEI	DMKPEL
		DMKPEN	DMKPEQ	DMKPER	DMKPET	DMKPGM	DMKPGS	DMKPGT	DMKPGU	DMKPMA	DMKPRG
		DMKPRV	DMKPRW	DMKPST	DMKPTR	DMKPTS	DMKPTT	DMKQCN	DMKQCO	DMKQCP	DMKQCQ
		DMKQVM	DMKREI	DMKRGA	DMKRGB	DMKRGC	DMKRGD	DMKRGE	DMKRND	DMKRNH	DMKRSE
		DMKRSF	DMKRSP	DMKRSQ	DMKRST	DMKSAD	DMKSAV	DMKSCH	DMKSCO	DMKSEG	DMKSEL
		DMKSEP	DMKSFB	DMKSNC	DMKSND	DMKSPK	DMKSPL	DMKSPM	DMKSPS	DMKSPT	DMKSRM
		DMKSSP	DMKSSS	DMKSST	DMKSSU	DMKSSV	DMKSTA	DMKSTK	DMKSTP	DMKSTR	DMKSVD
		DMKSWA	DMKSWM	DMKTAP	DMKTAQ	DMKTCS	DMKTCT	DMKTDK	DMKTEE	DMKTEF	DMKTEM
		DMKTES	DMKTHI	DMKTMR	DMKTPE	DMKTRC	DMKTRD	DMKTRK	DMKTRP	DMKTRQ	DMKTRR
		DMKTRT	DMKTRU	DMKTRX	DMKTTX	DMKTTY	DMKUDR	DMKUDU	DMKUNT	DMKURS	DMKUSP
		DMKUSQ	DMKVAT	DMKVAU	DMKVBM	DMKVCA	DMKVCB	DMKVCH	DMKVCN	DMKVCP	DMKVCQ
		DMKVCR	DMKVCS	DMKVCT	DMKVCU	DMKVCV	DMKVCW	DMKVCX	DMKVDA	DMKVDB	DMKVDC
		DMKVDD	DMKVDE	DMKVDF	DMKVDG	DMKVDH	DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVFC
		DMKVFD	DMKVFE	DMKVFR	DMKVFS	DMKVIO	DMKVMA	DMKVMC	DMKVMD	DMKVME	DMKVMI
		DMKVRR	DMKVRS	DMKVSC	DMKVSE	DMKVSF	DMKVSI	DMKVSJ	DMKVSP	DMKVSQ	DMKVST
		DMKVSV	DMKWAI	DMKWRM	DMKWRN	DMKXAB	DMKXAD	DMKXST	DMKZTD		
R11	003867	DMKACO	DMKACR	DMKALG	DMKALO	DMKAPI	DMKAPS	DMKAPT	DMKAPU	DMKAPV	DMKAPW
		DMKAPX	DMKAPY	DMKAPZ	DMKATS	DMKBIO	DMKBLD	DMKBSC	DMKCAC	DMKCAO	DMKCCD
		DMKCCF	DMKCCH	DMKCCO	DMKCCS	DMKCCT	DMKCCW	DMKCDB	DMKCDM	DMKCDS	DMKCFC
		DMKCFD	DMKCFF	DMKCFG	DMKCFH	DMKCFJ	DMKCFM	DMKCFO	DMKCFP	DMKCFQ	DMKCFR
		DMKCFS	DMKCFT	DMKCFU	DMKCFV	DMKCFW	DMKCFY	DMKCKD	DMKCKF	DMKCKH	DMKCKM
		DMKCKN	DMKCKP	DMKCKR	DMKCKS	DMKCKT	DMKCKV	DMKCLK	DMKCNS	DMKCNT	DMKCPB
		DMKCPI	DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPU	DMKCPV
		DMKCPW	DMKCPX	DMKCPY	DMKCPZ	DMKCQC	DMKCQG	DMKCQH	DMKCQI	DMKCQP	DMKCQQ
		DMKCQR	DMKCQS	DMKCQT	DMKCQU	DMKCQY	DMKCRM	DMKCSB	DMKCSC	DMKCSF	DMKCSO
		DMKCSP	DMKCSQ	DMKCSR	DMKCST	DMKCSU	DMKCSV	DMKCSW	DMKCSX	DMKCSY	DMKCVU
		DMKDAD	DMKDAS	DMKDAU	DMKDDC	DMKDDR	DMKDEF	DMKDEG	DMKDEI	DMKDEX	DMKDGD
		DMKDGF	DMKDIA	DMKDIB	DMKDID	DMKDIF	DMKDIR	DMKDMP	DMKDMQ	DMKDNC	DMKDRD
		DMKDRE	DMKDSB	DMKDSP	DMKENT	DMKEPS	DMKERM	DMKEXT	DMKFMT	DMKFPS	DMKFRE
		DMKFRT	DMKGIO	DMKGRA	DMKGRC	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRI	DMKGRT
		DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKIDR	DMKIDU	DMKIOE
		DMKIOF	DMKIOG	DMKIOH	DMKIOQ	DMKIOS	DMKIOT	DMKISM	DMKIUA	DMKIUB	DMKIUC
		DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS	DMKJRL	DMKLNK	DMKLNM
		DMKLOC	DMKLOG	DMKLOH	DMKLOJ	DMKLOK	DMKLOM	DMKMCC	DMKMCD	DMKMCH	DMKMCI
		DMKMCT	DMKMHC	DMKMHV	DMKMIA	DMKMID	DMKMNI	DMKMNJ	DMKMNL	DMKMNT	DMKMON
		DMKMOO	DMKMPO	DMKMSG	DMKMSW	DMKNEA	DMKNEM	DMKNES	DMKNET	DMKNLD	DMKNLE
		DMKOPE	DMKOVR	DMKPAG	DMKPAH	DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPER	DMKPET
		DMKPGM	DMKPGS	DMKPGT	DMKPGU	DMKPMA	DMKPRG	DMKPRV	DMKPRW	DMKPSA	DMKPST
		DMKPTR	DMKPTS	DMKPTT	DMKQCN	DMKQCO	DMKQCP	DMKQCQ	DMKQVM	DMKREI	DMKRET
		DMKRGA	DMKRGB	DMKRGC	DMKRGD	DMKRGE	DMKRND	DMKRNH	DMKRPA	DMKRPD	DMKRPI
		DMKRPW	DMKRSE	DMKRSF	DMKRSP	DMKRSQ	DMKRST	DMKSAD	DMKSAV	DMKSBL	DMKSCH
		DMKSCN	DMKSCO	DMKSEG	DMKSEL	DMKSEP	DMKSFB	DMKSNC	DMKSND	DMKSPC	DMKSPK
		DMKSPL	DMKSPM	DMKSPR	DMKSPS	DMKSPT	DMKSRM	DMKSSP	DMKSSS	DMKSST	DMKSSU
		DMKSSV	DMKSTA	DMKSTK	DMKSTP	DMKSTR	DMKSVC	DMKSVD	DMKSWA	DMKSWM	DMKTAP
		DMKTAQ	DMKTCS	DMKTCT	DMKTDK	DMKTEE	DMKTEF	DMKTEM	DMKTES	DMKTHI	DMKTMR
		DMKTOD	DMKTPE	DMKTRA	DMKTRC	DMKTRD	DMKTRK	DMKTRM	DMKTRP	DMKTRQ	DMKTRR
		DMKTRT	DMKTRU	DMKTRX	DMKTTX	DMKTTY	DMKUDR	DMKUDU	DMKUNT	DMKURS	DMKUSP
		DMKUSQ	DMKVAT	DMKVAU	DMKVBM	DMKVCA	DMKVCB	DMKVCH	DMKVCN	DMKVCP	DMKVCQ
		DMKVCR	DMKVCS	DMKVCT	DMKVCU	DMKVCV	DMKVCW	DMKVCX	DMKVDA	DMKVDB	DMKVDC
		DMKVDD	DMKVDE	DMKVDF	DMKVDG	DMKVDH	DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVFC
		DMKVFD	DMKVFE	DMKVFR	DMKVFS	DMKVIO	DMKVMA	DMKVMC	DMKVMD	DMKVME	DMKVMG
		DMKVMI	DMKVRR	DMKVRS	DMKVSC	DMKVSD	DMKVSE	DMKVSF	DMKVSG	DMKVSI	DMKVSJ
		DMKVSP	DMKVSQ	DMKVST	DMKVSU	DMKVSW	DMKVSX	DMKWAI	DMKWRM	DMKWRN	DMKXAB
		DMKXAD	DMKXST	DMKZTD							

LABEL	COUNT	REFERENCES									
R12	006079	DMKACO	DMKACR	DMKACS	DMKALG	DMKALO	DMKAPI	DMKAPS	DMKAPT	DMKAPU	DMKAPV
		DMKAPW	DMKAPX	DMKAPY	DMKAPZ	DMKATS	DMKBIO	DMKBLD	DMKBSC	DMKCAC	DMKCAO
		DMKCCD	DMKCCF	DMKCCH	DMKCCO	DMKCCS	DMKCCT	DMKCCW	DMKCDB	DMKCDM	DMKCDS
		DMKCFC	DMKCFD	DMKCFF	DMKCFG	DMKCFH	DMKCFJ	DMKCFM	DMKCFO	DMKCFP	DMKCFQ
		DMKCFR	DMKCFS	DMKCFT	DMKCFU	DMKCFV	DMKCFW	DMKCFY	DMKCKD	DMKCKF	DMKCKH
		DMKCKM	DMKCKN	DMKCKP	DMKCKR	DMKCKS	DMKCKT	DMKCKV	DMKCKW	DMKCLK	DMKCNS
		DMKCNT	DMKCPB	DMKCPI	DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT
		DMKCPU	DMKCPV	DMKCPW	DMKCPX	DMKCPY	DMKCPZ	DMKCQC	DMKCQG	DMKCQH	DMKCQI
		DMKCQP	DMKCQQ	DMKCQR	DMKCQS	DMKCQT	DMKCQU	DMKCQY	DMKCRM	DMKCSB	DMKCSC
		DMKCSF	DMKCSO	DMKCSP	DMKCSQ	DMKCSR	DMKCST	DMKCSU	DMKCSV	DMKCSW	DMKCSX
		DMKCSY	DMKCVU	DMKDAD	DMKDAS	DMKDAU	DMKDDC	DMKDDR	DMKDEF	DMKDEG	DMKDEI
		DMKDEX	DMKDGD	DMKDGF	DMKDIA	DMKDIB	DMKDID	DMKDIF	DMKDIR	DMKDMP	DMKDMQ
		DMKDNC	DMKDRD	DMKDRE	DMKDSB	DMKDSP	DMKEIG	DMKENT	DMKEPS	DMKERM	DMKERP
		DMKEXT	DMKFMT	DMKFPS	DMKFRE	DMKFRT	DMKGIO	DMKGRA	DMKGRC	DMKGRD	DMKGRE
		DMKGRF	DMKGRG	DMKGRH	DMKGRI	DMKGRT	DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKHVD
		DMKHVE	DMKHVF	DMKIDR	DMKIDU	DMKIMG	DMKIOC	DMKIOE	DMKIOF	DMKIOG	DMKIOH
		DMKIOJ	DMKIOQ	DMKIOS	DMKIOT	DMKISM	DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUG
		DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS	DMKJRL	DMKLD00E	DMKLNK	DMKLNM	DMKLOC
		DMKLOG	DMKLOH	DMKLOJ	DMKLOK	DMKLOM	DMKMCC	DMKMCD	DMKMCH	DMKMCI	DMKMCT
		DMKMHC	DMKMHV	DMKMIA	DMKMID	DMKMNI	DMKMNJ	DMKMNL	DMKMNT	DMKMON	DMKMOO
		DMKMPO	DMKMSG	DMKMSW	DMKNEA	DMKNEM	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKNMT
		DMKOPE	DMKOPR	DMKOVR	DMKPAG	DMKPAH	DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPER
		DMKPET	DMKPGM	DMKPGS	DMKPGT	DMKPGU	DMKPMA	DMKPRG	DMKPRV	DMKPRW	DMKPSA
		DMKPST	DMKPTR	DMKPTS	DMKPTT	DMKQCN	DMKQCO	DMKQCP	DMKQCQ	DMKQVM	DMKREI
		DMKRET	DMKRGA	DMKRGB	DMKRGC	DMKRGD	DMKRGE	DMKRND	DMKRNH	DMKRPA	DMKRPD
		DMKRPI	DMKRPW	DMKRSE	DMKRSF	DMKRSP	DMKRSQ	DMKRST	DMKSAD	DMKSAV	DMKSBL
		DMKSCH	DMKSCO	DMKSEG	DMKSEL	DMKSEP	DMKSEV	DMKSFB	DMKSIX	DMKSNC	DMKSND
		DMKSPC	DMKSPK	DMKSPL	DMKSPM	DMKSPR	DMKSPS	DMKSPT	DMKSRM	DMKSSP	DMKSSS
		DMKSST	DMKSSU	DMKSSV	DMKSTA	DMKSTD	DMKSTK	DMKSTP	DMKSTR	DMKSVC	DMKSVD
		DMKSWA	DMKSWM	DMKTAP	DMKTAQ	DMKTCS	DMKTCT	DMKTDK	DMKTEE	DMKTEF	DMKTEM
		DMKTES	DMKTHI	DMKTMR	DMKTOD	DMKTPE	DMKTRA	DMKTRC	DMKTRD	DMKTRK	DMKTRM
		DMKTRP	DMKTRQ	DMKTRR	DMKTRT	DMKTRU	DMKTRX	DMKTTX	DMKTTY	DMKUDR	DMKUDU
		DMKUNT	DMKURS	DMKUSP	DMKUSQ	DMKVAT	DMKVAU	DMKVBM	DMKVCA	DMKVCB	DMKVCH
		DMKVCN	DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCU	DMKVCV	DMKVCW	DMKVCX
		DMKVDA	DMKVDB	DMKVDC	DMKVDD	DMKVDE	DMKVDF	DMKVDG	DMKVDH	DMKVDR	DMKVDS
		DMKVDT	DMKVER	DMKVFC	DMKVFD	DMKVFE	DMKVFR	DMKVFS	DMKVIO	DMKVMA	DMKVMC
		DMKVMD	DMKVME	DMKVMG	DMKVMI	DMKVRR	DMKVRS	DMKVSC	DMKVSD	DMKVSE	DMKVSF
		DMKVSG	DMKVSI	DMKVSJ	DMKVSP	DMKVSQ	DMKVST	DMKVSU	DMKVSV	DMKVSW	DMKVSX
		DMKWAI	DMKWRM	DMKWRN	DMKXAB	DMKXAD	DMKXST	DMKZTD			
R13	002005	DMKACO	DMKACR	DMKACS	DMKALG	DMKALO	DMKAPI	DMKAPS	DMKAPT	DMKAPU	DMKAPV
		DMKAPW	DMKAPX	DMKAPY	DMKAPZ	DMKATS	DMKBIO	DMKBLD	DMKBSC	DMKCAC	DMKCAO
		DMKCCD	DMKCCF	DMKCCH	DMKCCO	DMKCCS	DMKCCT	DMKCCW	DMKCDB	DMKCDM	DMKCDS
		DMKCFC	DMKCFD	DMKCFF	DMKCFG	DMKCFH	DMKCFJ	DMKCFM	DMKCFO	DMKCFP	DMKCFQ
		DMKCFR	DMKCFS	DMKCFT	DMKCFU	DMKCFV	DMKCFW	DMKCFY	DMKCKD	DMKCKR	DMKCKS
		DMKCKT	DMKCKV	DMKCLK	DMKCNS	DMKCNT	DMKCPB	DMKCPI	DMKCPJ	DMKCPM	DMKCPN
		DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPU	DMKCPV	DMKCPW	DMKCPX	DMKCPY	DMKCPZ
		DMKCQC	DMKCQG	DMKCQH	DMKCQI	DMKCQP	DMKCQQ	DMKCQR	DMKCQS	DMKCQT	DMKCQU
		DMKCQY	DMKCRM	DMKCSB	DMKCSC	DMKCSF	DMKCSO	DMKCSP	DMKCSQ	DMKCSR	DMKCST
		DMKCSU	DMKCSV	DMKCSW	DMKCSX	DMKCSY	DMKCVU	DMKDAD	DMKDAS	DMKDAU	DMKDDC
		DMKDDR	DMKDEF	DMKDEG	DMKDEI	DMKDEX	DMKDGD	DMKDGF	DMKDIA	DMKDIB	DMKDID
		DMKDIF	DMKDIR	DMKDMP	DMKDMQ	DMKDNC	DMKDRD	DMKDRE	DMKDSB	DMKDSP	DMKEIG
		DMKENT	DMKEPS	DMKERM	DMKEXT	DMKFMT	DMKFPS	DMKFRE	DMKFRT	DMKGIO	DMKGRA
		DMKGRC	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRI	DMKGRT	DMKHPS	DMKHPT	DMKHPU
		DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKIDR	DMKIDU	DMKIMG	DMKIOE	DMKIOF	DMKIOG
		DMKIOH	DMKIOQ	DMKIOS	DMKIOT	DMKISM	DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUG
		DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS	DMKJRL	DMKLD00E	DMKLNK	DMKLNM	DMKLOC
		DMKLOG	DMKLOH	DMKLOJ	DMKLOM	DMKMCC	DMKMCD	DMKMCH	DMKMCI	DMKMHC	DMKMHV
		DMKMIA	DMKMID	DMKMNI	DMKMNJ	DMKMNL	DMKMNT	DMKMON	DMKMOO	DMKMPO	DMKMSG

LABEL	COUNT	REFERENCES									
		DMKMSW	DMKNEA	DMKNEM	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKOPE	DMKOVR	DMKPAG
		DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPER	DMKPET	DMKPGM	DMKPGS	DMKPGU	DMKPMA
		DMKPRG	DMKPRV	DMKPST	DMKPTR	DMKPTS	DMKPTT	DMKQCN	DMKQCO	DMKQCP	DMKQCQ
		DMKQVM	DMKREI	DMKRGA	DMKRGB	DMKRGC	DMKRGE	DMKRND	DMKRNH	DMKRPA	DMKRPD
		DMKRPI	DMKRPW	DMKRSE	DMKRSF	DMKRSP	DMKRSQ	DMKRST	DMKSAD	DMKSAV	DMKSBL
		DMKSCH	DMKSCO	DMKSEG	DMKSEL	DMKSEP	DMKSEV	DMKSFB	DMKSIX	DMKSNC	DMKSND
		DMKSPC	DMKSPK	DMKSPL	DMKSPM	DMKSPR	DMKSPS	DMKSPT	DMKSRM	DMKSSP	DMKSSS
		DMKSST	DMKSSU	DMKSSV	DMKSTP	DMKSTR	DMKSVC	DMKSVD	DMKSWA	DMKSWM	DMKTAP
		DMKTAQ	DMKTCS	DMKTCT	DMKTDK	DMKTEE	DMKTEF	DMKTEM	DMKTES	DMKTHI	DMKTMR
		DMKTOD	DMKTPE	DMKTRA	DMKTRC	DMKTRD	DMKTRK	DMKTRM	DMKTRP	DMKTRQ	DMKTRR
		DMKTRU	DMKTRX	DMKTTX	DMKTTY	DMKUDR	DMKUDU	DMKUNT	DMKURS	DMKUSP	DMKUSQ
		DMKVAT	DMKVAU	DMKVBM	DMKVCA	DMKVCB	DMKVCH	DMKVCN	DMKVCP	DMKVCQ	DMKVCR
		DMKVCS	DMKVCT	DMKVCU	DMKVCV	DMKVCW	DMKVCX	DMKVDA	DMKVDB	DMKVDC	DMKVDD
		DMKVDE	DMKVDF	DMKVDG	DMKVDH	DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVFC	DMKVFD
		DMKVFE	DMKVFR	DMKVFS	DMKVMA	DMKVMC	DMKVMD	DMKVME	DMKVMG	DMKVMI	DMKVRR
		DMKVSC	DMKVSD	DMKVSE	DMKVSF	DMKVSG	DMKVSI	DMKVSJ	DMKVSP	DMKVSQ	DMKVST
		DMKVSU	DMKVSV	DMKVSW	DMKVSX	DMKWAI	DMKWRM	DMKWRN	DMKXAB	DMKXAD	DMKXST
		DMKZTD									
R14	017582	DMKACO	DMKACR	DMKACS	DMKALG	DMKALO	DMKAPI	DMKAPS	DMKAPT	DMKAPU	DMKAPV
		DMKAPW	DMKAPX	DMKAPY	DMKAPZ	DMKATS	DMKBIO	DMKBLD	DMKBSC	DMKCAC	DMKCAO
		DMKCCD	DMKCCF	DMKCCH	DMKCCO	DMKCCS	DMKCCT	DMKCCW	DMKCDB	DMKCDM	DMKCDS
		DMKCFC	DMKCFD	DMKCFF	DMKCFG	DMKCFH	DMKCFJ	DMKCFM	DMKCFO	DMKCFP	DMKCFQ
		DMKCFR	DMKCFS	DMKCFT	DMKCFU	DMKCFV	DMKCFW	DMKCFY	DMKCKD	DMKCKF	DMKCKH
		DMKCKM	DMKCKN	DMKCKP	DMKCKR	DMKCKS	DMKCKT	DMKCKV	DMKCKW	DMKCLK	DMKCNS
		DMKCNT	DMKCPB	DMKCPI	DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT
		DMKCPU	DMKCPV	DMKCPW	DMKCPX	DMKCPY	DMKCPZ	DMKCQC	DMKCQG	DMKCQH	DMKCQI
		DMKCQP	DMKCQQ	DMKCQR	DMKCQS	DMKCQT	DMKCQU	DMKCQY	DMKCRM	DMKCSB	DMKCSC
		DMKCSF	DMKCSO	DMKCSP	DMKCSQ	DMKCSR	DMKCST	DMKCSU	DMKCSV	DMKCSW	DMKCSX
		DMKCSY	DMKCVT	DMKDAD	DMKDAS	DMKDAU	DMKDDC	DMKDDR	DMKDEF	DMKDEG	DMKDEI
		DMKDEX	DMKDGD	DMKDGF	DMKDIA	DMKDIB	DMKDID	DMKDIF	DMKDIR	DMKDMP	DMKDMQ
		DMKDNC	DMKDRD	DMKDRE	DMKDSB	DMKDSP	DMKEIG	DMKENT	DMKEPS	DMKERM	DMKERP
		DMKEXT	DMKFMT	DMKFPS	DMKFRE	DMKFRT	DMKGIO	DMKGRA	DMKGRC	DMKGRD	DMKGRE
		DMKGRF	DMKGRG	DMKGRH	DMKGRI	DMKGRT	DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKHVD
		DMKHVE	DMKHVF	DMKIDR	DMKIDU	DMKIMG	DMKIOC	DMKIOE	DMKIOF	DMKIOG	DMKIOH
		DMKIOJ	DMKIOQ	DMKIOS	DMKIOT	DMKISM	DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUG
		DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS	DMKJRL	DMKLDOOE	DMKLNK	DMKLNM	DMKLOC
		DMKLOG	DMKLOH	DMKLOJ	DMKLOK	DMKLOM	DMKMCC	DMKMCD	DMKMCH	DMKMCI	DMKMCT
		DMKMHC	DMKMHV	DMKMIA	DMKMID	DMKMNI	DMKMNJ	DMKMNL	DMKMNT	DMKMON	DMKMOO
		DMKMPO	DMKMSG	DMKMSW	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKNMT	DMKOPE
		DMKOPR	DMKOVR	DMKPAG	DMKPAH	DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPER	DMKPET
		DMKPGM	DMKPGS	DMKPGT	DMKPGU	DMKPMA	DMKPRG	DMKPRV	DMKPRW	DMKPSA	DMKPST
		DMKPTR	DMKPTS	DMKPTT	DMKQCN	DMKQCO	DMKQCP	DMKQCQ	DMKQVM	DMKREI	DMKRET
		DMKRGA	DMKRGB	DMKRGC	DMKRGD	DMKRGE	DMKRND	DMKRNH	DMKRPA	DMKRPD	DMKRSE
		DMKRSF	DMKRSP	DMKRSQ	DMKRST	DMKSAD	DMKSAV	DMKSBL	DMKSCH	DMKSCN	DMKSEG
		DMKSEL	DMKSEP	DMKSEV	DMKSFB	DMKSIX	DMKSNC	DMKSND	DMKSPK	DMKSPL	DMKSPM
		DMKSPR	DMKSPS	DMKSPT	DMKSRM	DMKSSP	DMKSSS	DMKSST	DMKSSU	DMKSSV	DMKSTA
		DMKSTK	DMKSTP	DMKSTR	DMKSVC	DMKSVD	DMKSWA	DMKSWM	DMKTAP	DMKTAQ	DMKTCS
		DMKTCT	DMKTDK	DMKTEE	DMKTEF	DMKTEM	DMKTES	DMKTHI	DMKTMR	DMKTOD	DMKTPE
		DMKTRA	DMKTRC	DMKTRD	DMKTRK	DMKTRP	DMKTRQ	DMKTRR	DMKTRT	DMKTRU	DMKTRX
		DMKTTX	DMKTTY	DMKUDR	DMKUDU	DMKUNT	DMKURS	DMKUSP	DMKUSQ	DMKVAT	DMKVAU
		DMKVBM	DMKVCA	DMKVCB	DMKVCH	DMKVCN	DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCT
		DMKVCU	DMKVCV	DMKVCW	DMKVCX	DMKVDA	DMKVDB	DMKVDC	DMKVDD	DMKVDE	DMKVDF
		DMKVDG	DMKVDH	DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVFC	DMKVFD	DMKVFE	DMKVFR
		DMKVFS	DMKVIO	DMKVMA	DMKVMC	DMKVMD	DMKVME	DMKVMG	DMKVMI	DMKVRR	DMKVRS
		DMKVSC	DMKVSD	DMKVSE	DMKVSF	DMKVSG	DMKVSI	DMKVSJ	DMKVSP	DMKVSQ	DMKVST
		DMKVSV	DMKVSW	DMKVSX	DMKWAI	DMKWRM	DMKWRN	DMKXAB	DMKXAD	DMKXST	DMKZTD
R15	026616	DMKACO	DMKACR	DMKACS	DMKALG	DMKALO	DMKAPI	DMKAPS	DMKAPT	DMKAPU	DMKAPV
		DMKAPW	DMKAPX	DMKAPY	DMKAPZ	DMKATS	DMKBIO	DMKBLD	DMKBSC	DMKCAC	DMKCAO

LABEL	COUNT	REFERENCES									
		DMKCCD	DMKCCF	DMKCCH	DMKCCO	DMKCCS	DMKCCT	DMKCCW	DMKCDB	DMKCDM	DMKCDS
		DMKCFC	DMKCFD	DMKCFF	DMKCFG	DMKCFH	DMKCFJ	DMKCFM	DMKCFO	DMKCFP	DMKCFQ
		DMKCFR	DMKCFS	DMKCFT	DMKCFU	DMKCFV	DMKCFW	DMKCFY	DMKCKD	DMKCKF	DMKCKH
		DMKCKM	DMKCKN	DMKCKP	DMKCKR	DMKCKS	DMKCKT	DMKCKV	DMKCKW	DMKCLK	DMKCNS
		DMKCNT	DMKCPB	DMKCPI	DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT
		DMKCPU	DMKCPV	DMKCPW	DMKCPX	DMKCPY	DMKCPZ	DMKCQC	DMKCQG	DMKCQH	DMKCQI
		DMKCQP	DMKCQQ	DMKCQR	DMKCQS	DMKCQT	DMKCQU	DMKCQY	DMKCRM	DMKCSB	DMKCSC
		DMKCSF	DMKCSO	DMKCSP	DMKCSQ	DMKCSR	DMKCST	DMKCSU	DMKCSV	DMKCSW	DMKCSX
		DMKCSY	DMKCVT	DMKDAD	DMKDAS	DMKDAU	DMKDDC	DMKDDR	DMKDEF	DMKDEG	DMKDEI
		DMKDEX	DMKDGD	DMKDGF	DMKDIA	DMKDIB	DMKDID	DMKDIF	DMKDIR	DMKDMP	DMKDMQ
		DMKDNC	DMKDRD	DMKDRE	DMKDSB	DMKDSP	DMKEIG	DMKENT	DMKEPS	DMKERM	DMKERP
		DMKEXT	DMKFMT	DMKFPS	DMKFRE	DMKFRT	DMKGIO	DMKGRA	DMKGRC	DMKGRD	DMKGRE
		DMKGRF	DMKGRG	DMKGRH	DMKGRI	DMKGRT	DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKHVD
		DMKHVE	DMKHVF	DMKIDR	DMKIDU	DMKIMG	DMKIOC	DMKIOE	DMKIOF	DMKIOG	DMKIOH
		DMKIOJ	DMKIOQ	DMKIOS	DMKIOT	DMKISM	DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUG
		DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS	DMKJRL	DMKLDOOE	DMKLNK	DMKLNM	DMKLOC
		DMKLOG	DMKLOH	DMKLOJ	DMKLOK	DMKLOM	DMKMCC	DMKMCD	DMKMCH	DMKMCI	DMKMCT
		DMKMHC	DMKMHV	DMKMIA	DMKMID	DMKMNI	DMKMNJ	DMKMNL	DMKMNT	DMKMON	DMKMOO
		DMKMPO	DMKMSG	DMKMSW	DMKNEA	DMKNEM	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKNMT
		DMKOPE	DMKOPR	DMKOVR	DMKPAG	DMKPAH	DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPER
		DMKPET	DMKPGM	DMKPGS	DMKPGT	DMKPGU	DMKPMA	DMKPRG	DMKPRV	DMKPRW	DMKPSA
		DMKPST	DMKPTR	DMKPTS	DMKPTT	DMKQCN	DMKQCO	DMKQCP	DMKQCQ	DMKQVM	DMKREI
		DMKRET	DMKRGA	DMKRGB	DMKRGC	DMKRGD	DMKRGE	DMKRND	DMKRNH	DMKRPA	DMKRPD
		DMKRPI	DMKRSE	DMKRSF	DMKRSP	DMKRSQ	DMKRST	DMKSAD	DMKSAV	DMKSBL	DMKSCH
		DMKSCN	DMKSEG	DMKSEL	DMKSEP	DMKSEV	DMKSFB	DMKSIX	DMKSNC	DMKSND	DMKSPC
		DMKSPK	DMKSPL	DMKSPM	DMKSPR	DMKSPS	DMKSPT	DMKSRM	DMKSSP	DMKSSS	DMKSST
		DMKSSU	DMKSSV	DMKSTA	DMKSTK	DMKSTP	DMKSTR	DMKSVC	DMKSVD	DMKSWA	DMKSWM
		DMKTAP	DMKTAQ	DMKTCS	DMKTCT	DMKTDK	DMKTEE	DMKTEF	DMKTEM	DMKTES	DMKTHI
		DMKTMR	DMKTOD	DMKTPE	DMKTRA	DMKTRC	DMKTRD	DMKTRK	DMKTRP	DMKTRQ	DMKTRR
		DMKTRT	DMKTRU	DMKTRX	DMKTTX	DMKTTY	DMKUDR	DMKUDU	DMKUNT	DMKURS	DMKUSP
		DMKUSQ	DMKVAT	DMKVAU	DMKVBM	DMKVCA	DMKVCB	DMKVCH	DMKVCN	DMKVCP	DMKVCQ
		DMKVCR	DMKVCS	DMKVCT	DMKVCU	DMKVCV	DMKVCW	DMKVCX	DMKVDA	DMKVDB	DMKVDC
		DMKVDD	DMKVDE	DMKVDF	DMKVDG	DMKVDH	DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVFC
		DMKVFD	DMKVFE	DMKVFR	DMKVFS	DMKVIO	DMKVMA	DMKVMC	DMKVMD	DMKVME	DMKVMG
		DMKVMI	DMKVRR	DMKVRS	DMKVSC	DMKVSD	DMKVSE	DMKVSF	DMKVSG	DMKVSI	DMKVSJ
		DMKVSP	DMKVSQ	DMKVST	DMKVSU	DMKVSV	DMKVSW	DMKVSX	DMKWAI	DMKWRM	DMKWRN
		DMKXAB	DMKXAD	DMKXST	DMKZTD						
R2	022358	DMKACO	DMKACR	DMKACS	DMKALG	DMKALO	DMKAPI	DMKAPS	DMKAPT	DMKAPU	DMKAPV
		DMKAPX	DMKAPY	DMKATS	DMKBIO	DMKBLD	DMKBSC	DMKCAC	DMKCAO	DMKCCD	DMKCCH
		DMKCCO	DMKCCS	DMKCCW	DMKCDB	DMKCDM	DMKCDS	DMKCFC	DMKCFD	DMKCFF	DMKCFG
		DMKCFH	DMKCFJ	DMKCFM	DMKCFO	DMKCFP	DMKCFQ	DMKCFR	DMKCFS	DMKCFT	DMKCFU
		DMKCFV	DMKCFW	DMKCFY	DMKCKD	DMKCKF	DMKCKH	DMKCKM	DMKCKN	DMKCKP	DMKCKR
		DMKCKS	DMKCKT	DMKCKV	DMKCKW	DMKCLK	DMKCNS	DMKCNT	DMKCPB	DMKCPI	DMKCPJ
		DMKCPM	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPU	DMKCPV	DMKCPW	DMKCPX
		DMKCPY	DMKCPZ	DMKCQC	DMKCQG	DMKCQH	DMKCQI	DMKCQP	DMKCQQ	DMKCQR	DMKCQS
		DMKCQT	DMKCQU	DMKCQY	DMKCRM	DMKCSB	DMKCSC	DMKCSF	DMKCSO	DMKCSP	DMKCSQ
		DMKCSR	DMKCST	DMKCSU	DMKCSV	DMKCSW	DMKCSX	DMKCSY	DMKCVT	DMKCVU	DMKDAD
		DMKDAS	DMKDAU	DMKDDC	DMKDDR	DMKDEF	DMKDEG	DMKDEI	DMKDEX	DMKDGD	DMKDGF
		DMKDIA	DMKDIB	DMKDID	DMKDIF	DMKDIR	DMKDMP	DMKDMQ	DMKDNC	DMKDRD	DMKDRE
		DMKDSB	DMKDSP	DMKEIG	DMKENT	DMKEPS	DMKERM	DMKERP	DMKEXT	DMKFMT	DMKFPS
		DMKFRE	DMKFRT	DMKGIO	DMKGRA	DMKGRC	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRH
		DMKGRI	DMKGRT	DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKIDR
		DMKIDU	DMKIMG	DMKIOE	DMKIOF	DMKIOG	DMKIOH	DMKIOJ	DMKIOQ	DMKIOS	DMKIOT
		DMKISM	DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUP
		DMKIUS	DMKJRL	DMKLDOOE	DMKLNK	DMKLNM	DMKLOC	DMKLOG	DMKLOH	DMKLOJ	DMKLOK
		DMKLOM	DMKMCC	DMKMCD	DMKMCH	DMKMCI	DMKMCT	DMKMHC	DMKMHV	DMKMIA	DMKMID
		DMKMNI	DMKMNJ	DMKMNL	DMKMNT	DMKMON	DMKMOO	DMKMPO	DMKMSG	DMKMSW	DMKNEA
		DMKNEM	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKNMT	DMKOPE	DMKOPR	DMKOVR	DMKPAG

LABEL	COUNT	REFERENCES									
		DMKPAH	DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPER	DMKPET	DMKPGM	DMKPGS	DMKPGT
		DMKPGU	DMKPMA	DMKPRG	DMKPRV	DMKPRW	DMKPSA	DMKPST	DMKPTR	DMKPTS	DMKPTT
		DMKQCN	DMKQCO	DMKQCP	DMKQCQ	DMKQVM	DMKREI	DMKRET	DMKRGA	DMKRGB	DMKRGC
		DMKRGD	DMKRGE	DMKRND	DMKRNH	DMKRPA	DMKRPD	DMKRSE	DMKRSP	DMKRSQ	DMKRST
		DMKSAD	DMKSAV	DMKSCH	DMKSCN	DMKSCO	DMKSEG	DMKSEL	DMKSEP	DMKSEV	DMKSFB
		DMKSIX	DMKSNC	DMKSND	DMKSPK	DMKSPL	DMKSPM	DMKSPR	DMKSPS	DMKSPT	DMKSRM
		DMKSSP	DMKSSS	DMKSST	DMKSSU	DMKSSV	DMKSTA	DMKSTK	DMKSTP	DMKSTR	DMKSVC
		DMKSVD	DMKSWA	DMKSWM	DMKTAP	DMKTAQ	DMKTCS	DMKTCT	DMKTDK	DMKTES	DMKTHI
		DMKTMR	DMKTOD	DMKTPE	DMKTRA	DMKTRC	DMKTRD	DMKTRK	DMKTRM	DMKTRP	DMKTRQ
		DMKTRR	DMKTRT	DMKTRU	DMKTRX	DMKTTX	DMKTTY	DMKUDR	DMKUDU	DMKUNT	DMKURS
		DMKUSP	DMKUSQ	DMKVAT	DMKVAU	DMKVBM	DMKVCA	DMKVCB	DMKVCH	DMKVCN	DMKVCP
		DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCU	DMKVCV	DMKVCW	DMKVCX	DMKVDA	DMKVDB
		DMKVDC	DMKVDD	DMKVDE	DMKVDF	DMKVDG	DMKVDH	DMKVDR	DMKVDS	DMKVDT	DMKVER
		DMKVFC	DMKVFD	DMKVFE	DMKVFR	DMKVFS	DMKVIO	DMKVMA	DMKVMC	DMKVMD	DMKVME
		DMKVMG	DMKVMI	DMKVRR	DMKVRS	DMKVSC	DMKVSD	DMKVSE	DMKVSF	DMKVSG	DMKVSI
		DMKVSJ	DMKVSP	DMKVSQ	DMKVST	DMKVSV	DMKVSW	DMKVSX	DMKWAI	DMKWRM	DMKWRN
		DMKXAB	DMKXAD	DMKXST	DMKZTD						
R3	015012	DMKACO	DMKACR	DMKACS	DMKALG	DMKALO	DMKAPI	DMKAPS	DMKAPT	DMKAPU	DMKAPV
		DMKAPW	DMKAPX	DMKAPY	DMKATS	DMKBIO	DMKBLD	DMKBSC	DMKCAC	DMKCAO	DMKCCD
		DMKCCF	DMKCCH	DMKCCO	DMKCCS	DMKCCW	DMKCDB	DMKCDM	DMKCDS	DMKCFC	DMKCFD
		DMKCFF	DMKCFG	DMKCFH	DMKCFJ	DMKCFM	DMKCFO	DMKCFP	DMKCFQ	DMKCFR	DMKCFS
		DMKCFT	DMKCFU	DMKCFV	DMKCFW	DMKCFY	DMKCKD	DMKCKF	DMKCKH	DMKCKM	DMKCKN
		DMKCKP	DMKCKR	DMKCKS	DMKCKT	DMKCKV	DMKCKW	DMKCNS	DMKCNT	DMKCPB	DMKCPI
		DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPU	DMKCPV	DMKCPW
		DMKCPX	DMKCPY	DMKCPZ	DMKCQC	DMKCQG	DMKCQH	DMKCQI	DMKCQP	DMKCQQ	DMKCQR
		DMKCQS	DMKCQT	DMKCQU	DMKCQY	DMKCRM	DMKCSB	DMKCSF	DMKCSO	DMKCSP	DMKCSQ
		DMKCSR	DMKCST	DMKCSU	DMKCSV	DMKCSW	DMKCSX	DMKCSY	DMKCVT	DMKDAD	DMKDAS
		DMKDAU	DMKDDC	DMKDDR	DMKDEF	DMKDEG	DMKDEI	DMKDEX	DMKDGD	DMKDGF	DMKDIA
		DMKDIB	DMKDID	DMKDIF	DMKDIR	DMKDMP	DMKDMQ	DMKDNC	DMKDRD	DMKDRE	DMKDSB
		DMKDSP	DMKEIG	DMKENT	DMKEPS	DMKERM	DMKERP	DMKEXT	DMKFMT	DMKFPS	DMKFRE
		DMKFRT	DMKGIO	DMKGRA	DMKGRC	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRH	DMKGRI
		DMKGRT	DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKIDR	DMKIDU
		DMKIMG	DMKIOE	DMKIOF	DMKIOG	DMKIOH	DMKIOJ	DMKIOQ	DMKIOS	DMKIOT	DMKISM
		DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS
		DMKJRL	DMKLDOOE	DMKLNK	DMKLNM	DMKLOC	DMKLOG	DMKLOH	DMKLOJ	DMKLOK	DMKLOM
		DMKMCC	DMKMCD	DMKMCH	DMKMCI	DMKMCT	DMKMHC	DMKMHV	DMKMIA	DMKMID	DMKMNI
		DMKMNJ	DMKMNL	DMKMNT	DMKMON	DMKMOO	DMKMPO	DMKMSG	DMKMSW	DMKNEA	DMKNEM
		DMKNES	DMKNET	DMKNLD	DMKNLE	DMKNMT	DMKOPE	DMKOPR	DMKOVR	DMKPAG	DMKPAH
		DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPER	DMKPET	DMKPGM	DMKPGS	DMKPGT	DMKPGU
		DMKPMA	DMKPRG	DMKPRV	DMKPRW	DMKPST	DMKPTR	DMKPTS	DMKPTT	DMKQCN	DMKQCO
		DMKQCP	DMKQCQ	DMKQVM	DMKREI	DMKRET	DMKRGA	DMKRGB	DMKRGC	DMKRGD	DMKRGE
		DMKRND	DMKRNH	DMKRPA	DMKRPD	DMKRSE	DMKRSF	DMKRSP	DMKRSQ	DMKRST	DMKSAD
		DMKSAV	DMKSBL	DMKSCH	DMKSCN	DMKSCO	DMKSEG	DMKSEL	DMKSEP	DMKSEV	DMKSFB
		DMKSIX	DMKSNC	DMKSND	DMKSPK	DMKSPL	DMKSPM	DMKSPR	DMKSPS	DMKSPT	DMKSRM
		DMKSSP	DMKSSS	DMKSST	DMKSSU	DMKSSV	DMKSTA	DMKSTK	DMKSTP	DMKSTR	DMKSVC
		DMKSVD	DMKSWA	DMKSWM	DMKTAP	DMKTAQ	DMKTCS	DMKTCT	DMKTDK	DMKTEM	DMKTES
		DMKTHI	DMKTMR	DMKTOD	DMKTPE	DMKTRA	DMKTRC	DMKTRD	DMKTRK	DMKTRM	DMKTRP
		DMKTRQ	DMKTRR	DMKTRT	DMKTRU	DMKTRX	DMKTTX	DMKTTY	DMKUDR	DMKUDU	DMKUNT
		DMKUSP	DMKUSQ	DMKVAT	DMKVAU	DMKVBM	DMKVCA	DMKVCB	DMKVCH	DMKVCN	DMKVCP
		DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCU	DMKVCV	DMKVCW	DMKVCX	DMKVDA	DMKVDB
		DMKVDC	DMKVDD	DMKVDE	DMKVDF	DMKVDG	DMKVDH	DMKVDR	DMKVDS	DMKVDT	DMKVER
		DMKVFC	DMKVFD	DMKVFE	DMKVFR	DMKVFS	DMKVIO	DMKVMA	DMKVMC	DMKVMD	DMKVME
		DMKVMG	DMKVMI	DMKVRR	DMKVRS	DMKVSC	DMKVSD	DMKVSE	DMKVSF	DMKVSG	DMKVSI
		DMKVSJ	DMKVSP	DMKVSQ	DMKVST	DMKVSV	DMKVSX	DMKWAI	DMKWRM	DMKWRN	DMKXAB
		DMKXAD	DMKXST	DMKZTD							
R4	012277	DMKACO	DMKACR	DMKACS	DMKALG	DMKALO	DMKAPI	DMKAPS	DMKAPT	DMKAPU	DMKAPV
		DMKAPW	DMKAPX	DMKAPY	DMKATS	DMKBIO	DMKBLD	DMKBSC	DMKCAC	DMKCAO	DMKCCD
		DMKCCF	DMKCCH	DMKCCO	DMKCCS	DMKCCW	DMKCDB	DMKCDM	DMKCDS	DMKCFC	DMKCFD

LABEL	COUNT	REFERENCES									
		DMKCFF	DMKCFG	DMKCFH	DMKCFJ	DMKCFM	DMKCFO	DMKCFP	DMKCFQ	DMKCFR	DMKCFS
		DMKCFT	DMKCFU	DMKCFV	DMKCFW	DMKCFY	DMKCKD	DMKCKF	DMKCKH	DMKCKM	DMKCKN
		DMKCKP	DMKCKR	DMKCKS	DMKCKT	DMKCKV	DMKCKW	DMKCNS	DMKCNT	DMKCPB	DMKCPI
		DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPU	DMKCPV	DMKCPW
		DMKCPX	DMKCPY	DMKCPZ	DMKCQC	DMKCQG	DMKCQH	DMKCQI	DMKCQP	DMKCQQ	DMKCQR
		DMKCQS	DMKCQT	DMKCQU	DMKCQY	DMKCRM	DMKCSB	DMKCSC	DMKCSF	DMKCSO	DMKCSP
		DMKCSQ	DMKCSR	DMKCST	DMKCSU	DMKCSV	DMKCSW	DMKCSX	DMKCSY	DMKDAD	DMKDAS
		DMKDAU	DMKDDC	DMKDDR	DMKDEF	DMKDEG	DMKDEI	DMKDEX	DMKDGD	DMKDGF	DMKDIA
		DMKD!B	DMKDID	DMKDIF	DMKDIR	DMKDMP	DMKDMQ	DMKDNC	DMKDRD	DMKDRE	DMKDSB
		DMKDSP	DMKEIG	DMKENT	DMKEPS	DMKERM	DMKERP	DMKEXT	DMKFMT	DMKFPS	DMKFRE
		DMKFRT	DMKGIO	DMKGRA	DMKGRC	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRH	DMKGRT
		DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKIDR	DMKIDU	DMKIMG
		DMKIOC	DMKIOE	DMKIOF	DMKIOG	DMKIOH	DMKIOJ	DMKIOQ	DMKIOS	DMKIOT	DMKISM
		DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS
		DMKJRL	DMKLDOOE	DMKLNK	DMKLNM	DMKLOC	DMKLOG	DMKLOH	DMKLOJ	DMKLOK	DMKLOM
		DMKMCC	DMKMCD	DMKMCH	DMKMCI	DMKMCT	DMKMHC	DMKMHV	DMKMIA	DMKMID	DMKMNI
		DMKMNJ	DMKMNL	DMKMNT	DMKMON	DMKMOO	DMKMPO	DMKMSG	DMKMSW	DMKNEA	DMKNEM
		DMKNES	DMKNET	DMKNLD	DMKNLE	DMKNMT	DMKOPE	DMKOPR	DMKOVR	DMKPAG	DMKPAH
		DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPER	DMKPET	DMKPGM	DMKPGS	DMKPGT	DMKPGU
		DMKPMA	DMKPRG	DMKPRV	DMKPRW	DMKPST	DMKPTR	DMKPTS	DMKPTT	DMKQCN	DMKQCO
		DMKQCP	DMKQCQ	DMKQVM	DMKREI	DMKRET	DMKRGA	DMKRGB	DMKRGC	DMKRGD	DMKRGE
		DMKRND	DMKRNH	DMKRPD	DMKRSE	DMKRSF	DMKRSP	DMKRSQ	DMKRST	DMKSAD	DMKSAV
		DMKSBL	DMKSCH	DMKSCN	DMKSCO	DMKSEG	DMKSEL	DMKSEP	DMKSEV	DMKSFB	DMKSIX
		DMKSNC	DMKSND	DMKSPK	DMKSPL	DMKSPM	DMKSPR	DMKSPS	DMKSPT	DMKSRM	DMKSSP
		DMKSSS	DMKSST	DMKSSU	DMKSSV	DMKSTA	DMKSTK	DMKSTP	DMKSTR	DMKSVC	DMKSVD
		DMKSWA	DMKSWM	DMKTAP	DMKTAQ	DMKTCS	DMKTCT	DMKTDK	DMKTEE	DMKTEF	DMKTEM
		DMKTES	DMKTHI	DMKTMR	DMKTOD	DMKTPE	DMKTRA	DMKTRC	DMKTRD	DMKTRK	DMKTRM
		DMKTRP	DMKTRQ	DMKTRR	DMKTRT	DMKTRX	DMKTTX	DMKTTY	DMKUDR	DMKUDU	DMKUNT
		DMKURS	DMKUSP	DMKUSQ	DMKVAT	DMKVAU	DMKVBM	DMKVCA	DMKVCB	DMKVCH	DMKVCN
		DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCU	DMKVCV	DMKVCW	DMKVCX	DMKVDA
		DMKVDB	DMKVDC	DMKVDD	DMKVDE	DMKVDF	DMKVDG	DMKVDH	DMKVDR	DMKVDS	DMKVDT
		DMKVER	DMKVFC	DMKVFD	DMKVFE	DMKVFR	DMKVFS	DMKVIO	DMKVMA	DMKVMC	DMKVMD
		DMKVME	DMKVMG	DMKVMI	DMKVRR	DMKVRS	DMKVSC	DMKVSD	DMKVSE	DMKVSF	DMKVSG
		DMKVSI	DMKVSJ	DMKVSP	DMKVSQ	DMKVST	DMKVSV	DMKVSX	DMKWAI	DMKWRM	DMKWRN
		DMKXAB	DMKXAD	DMKXST	DMKZTD						
R5	010893	DMKACO	DMKACR	DMKACS	DMKALO	DMKAPI	DMKAPS	DMKAPT	DMKAPU	DMKAPV	DMKAPW
		DMKAPX	DMKAPY	DMKATS	DMKBIO	DMKBLD	DMKBSC	DMKCAC	DMKCAO	DMKCCH	DMKCCS
		DMKCCW	DMKCDB	DMKCDM	DMKCDS	DMKCFC	DMKCFD	DMKCFF	DMKCFG	DMKCFH	DMKCFJ
		DMKCFM	DMKCFO	DMKCFP	DMKCFQ	DMKCFR	DMKCFS	DMKCFT	DMKCFU	DMKCFV	DMKCFW
		DMKCFY	DMKCKD	DMKCKF	DMKCKH	DMKCKM	DMKCKN	DMKCKP	DMKCKR	DMKCKT	DMKCKV
		DMKCKW	DMKCNS	DMKCNT	DMKCPB	DMKCPI	DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPP
		DMKCPS	DMKCPT	DMKCPU	DMKCPV	DMKCPW	DMKCPY	DMKCPZ	DMKCQC	DMKCQG	DMKCQH
		DMKCQI	DMKCQP	DMKCQQ	DMKCQR	DMKCQS	DMKCQT	DMKCQU	DMKCQY	DMKCRM	DMKCSB
		DMKCSC	DMKCSO	DMKCSQ	DMKCSR	DMKCST	DMKCSU	DMKCSV	DMKCSW	DMKCSX	DMKCSY
		DMKCVU	DMKDAD	DMKDAS	DMKDAU	DMKDDC	DMKDDR	DMKDEF	DMKDEG	DMKDEI	DMKDEX
		DMKDGD	DMKDGF	DMKDIA	DMKDIB	DMKDID	DMKDIF	DMKDIR	DMKDMP	DMKDMQ	DMKDNC
		DMKDRD	DMKDRE	DMKDSB	DMKDSP	DMKENT	DMKEPS	DMKERM	DMKERP	DMKEXT	DMKFMT
		DMKFPS	DMKFRE	DMKFRT	DMKGIO	DMKGRA	DMKGRC	DMKGRD	DMKGRE	DMKGRF	DMKGRG
		DMKGRH	DMKGRI	DMKGRT	DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKHVD	DMKHVE	DMKHVF
		DMKIDR	DMKIDU	DMKIMG	DMKIOC	DMKIOE	DMKIOF	DMKIOG	DMKIOH	DMKIOJ	DMKIOQ
		DMKIOS	DMKIOT	DMKISM	DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL
		DMKIUN	DMKIUP	DMKIUS	DMKJRL	DMKLDOOE	DMKLNK	DMKLNM	DMKLOC	DMKLOG	DMKLOH
		DMKLOJ	DMKLOK	DMKLOM	DMKMCC	DMKMCD	DMKMCH	DMKMCI	DMKMHC	DMKMHV	DMKMIA
		DMKMID	DMKMNI	DMKMNJ	DMKMNL	DMKMNT	DMKMON	DMKMOO	DMKMPO	DMKMSG	DMKMSW
		DMKNEA	DMKNEM	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKNMT	DMKOPE	DMKOPR	DMKOVR
		DMKPAG	DMKPAH	DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPER	DMKPET	DMKPGM	DMKPGS
		DMKPGT	DMKPGU	DMKPMA	DMKPRG	DMKPRV	DMKPRW	DMKPST	DMKPTR	DMKPTS	DMKPTT
		DMKQCN	DMKQCO	DMKQCP	DMKQCQ	DMKQVM	DMKREI	DMKRET	DMKRGA	DMKRGB	DMKRGC

LABEL	COUNT	REFERENCES									
		DMKRGD	DMKRGE	DMKRNH	DMKRPA	DMKRPD	DMKRPI	DMKRSE	DMKRSF	DMKRSP	DMKRSQ
		DMKRST	DMKSAD	DMKSAV	DMKSCH	DMKSCN	DMKSCO	DMKSEG	DMKSEL	DMKSEP	DMKSFB
		DMKSNC	DMKSND	DMKSPK	DMKSPL	DMKSPM	DMKSPR	DMKSPS	DMKSRM	DMKSSP	DMKSSS
		DMKSST	DMKSSU	DMKSSV	DMKSTA	DMKSTK	DMKSTP	DMKSTR	DMKSVD	DMKSWA	DMKSWM
		DMKTAP	DMKTAQ	DMKTCS	DMKTCT	DMKTDK	DMKTEE	DMKTEF	DMKTEM	DMKTES	DMKTHI
		DMKTMR	DMKTOD	DMKTPE	DMKTRA	DMKTRC	DMKTRD	DMKTRK	DMKTRM	DMKTRP	DMKTRQ
		DMKTRR	DMKTRT	DMKTRU	DMKTRX	DMKTTX	DMKTTY	DMKUDR	DMKUDU	DMKUNT	DMKURS
		DMKUSP	DMKUSQ	DMKVAT	DMKVAU	DMKVBM	DMKVCA	DMKVCB	DMKVCH	DMKVCN	DMKVCP
		DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCU	DMKVCV	DMKVCW	DMKVCX	DMKVDA	DMKVDB
		DMKVDC	DMKVDD	DMKVDE	DMKVDF	DMKVDG	DMKVDH	DMKVDR	DMKVDS	DMKVDT	DMKVER
		DMKVFC	DMKVFD	DMKVFE	DMKVFR	DMKVIO	DMKVMA	DMKVMC	DMKVMD	DMKVME	DMKVMG
		DMKVMI	DMKVRR	DMKVRS	DMKVSC	DMKVSD	DMKVSE	DMKVSF	DMKVSG	DMKVSI	DMKVSJ
		DMKVSP	DMKVSQ	DMKVST	DMKVSV	DMKVSW	DMKVSX	DMKWAI	DMKWRM	DMKWRN	DMKXAB
		DMKXAD	DMKXST	DMKZTD							
R6	008272	DMKACO	DMKACR	DMKACS	DMKALO	DMKAPI	DMKAPS	DMKAPT	DMKAPU	DMKAPV	DMKAPY
		DMKATS	DMKBIO	DMKBLD	DMKBSC	DMKCAO	DMKCCD	DMKCCF	DMKCCH	DMKCCO	DMKCCS
		DMKCCT	DMKCCW	DMKCDB	DMKCDM	DMKCDS	DMKCFC	DMKCFD	DMKCFF	DMKCFG	DMKCFH
		DMKCFJ	DMKCFM	DMKCFO	DMKCFP	DMKCFQ	DMKCFR	DMKCFS	DMKCFT	DMKCFU	DMKCFV
		DMKCFW	DMKCFY	DMKCKD	DMKCKF	DMKCKH	DMKCKM	DMKCKN	DMKCKP	DMKCKR	DMKCKS
		DMKCKT	DMKCKV	DMKCKW	DMKCNS	DMKCNT	DMKCPB	DMKCPI	DMKCPJ	DMKCPM	DMKCPN
		DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPU	DMKCPV	DMKCPW	DMKCPX	DMKCPY	DMKCPZ
		DMKCQC	DMKCQG	DMKCQH	DMKCQI	DMKCQP	DMKCQQ	DMKCQR	DMKCQS	DMKCQT	DMKCQU
		DMKCQY	DMKCRM	DMKCSB	DMKCSC	DMKCSO	DMKCSQ	DMKCSR	DMKCST	DMKCSU	DMKCSV
		DMKCSW	DMKCSX	DMKCSY	DMKCVU	DMKDAD	DMKDAS	DMKDAU	DMKDDC	DMKDDR	DMKDEF
		DMKDEG	DMKDEI	DMKDEX	DMKDGD	DMKDGF	DMKDIA	DMKDIB	DMKDID	DMKDIF	DMKDIR
		DMKDMP	DMKDMQ	DMKDNC	DMKDRD	DMKDRE	DMKDSB	DMKDSP	DMKENT	DMKERM	DMKERP
		DMKEXT	DMKFMT	DMKFPS	DMKFRE	DMKFRT	DMKGIO	DMKGRA	DMKGRC	DMKGRD	DMKGRE
		DMKGRF	DMKGRG	DMKGRH	DMKGRI	DMKGRT	DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKHVD
		DMKHVE	DMKHVF	DMKIDR	DMKIDU	DMKIMG	DMKIOC	DMKIOE	DMKIOF	DMKIOG	DMKIOH
		DMKIOJ	DMKIOQ	DMKIOS	DMKIOT	DMKISM	DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUG
		DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS	DMKJRL	DMKLDOOE	DMKLNK	DMKLNM	DMKLOC
		DMKLOG	DMKLOH	DMKLOK	DMKLOM	DMKMCC	DMKMCD	DMKMCH	DMKMCI	DMKMCT	DMKMHC
		DMKMHV	DMKMIA	DMKMID	DMKMNI	DMKMNJ	DMKMNL	DMKMNT	DMKMON	DMKMOO	DMKMPO
		DMKMSG	DMKMSW	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKNMT	DMKOPR	DMKOVR
		DMKPAG	DMKPAH	DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPER	DMKPET	DMKPGM	DMKPGS
		DMKPGT	DMKPGU	DMKPMA	DMKPRG	DMKPRV	DMKPRW	DMKPST	DMKPTR	DMKPTS	DMKPTT
		DMKQCN	DMKQCO	DMKQCP	DMKQCQ	DMKQVM	DMKREI	DMKRET	DMKRGA	DMKRGB	DMKRGC
		DMKRGD	DMKRGE	DMKRND	DMKRNH	DMKRPD	DMKRSE	DMKRSF	DMKRSP	DMKRSQ	DMKRST
		DMKSAD	DMKSAV	DMKSBL	DMKSCH	DMKSCN	DMKSCO	DMKSEG	DMKSEL	DMKSEP	DMKSFB
		DMKSNC	DMKSND	DMKSPK	DMKSPL	DMKSPM	DMKSPR	DMKSPS	DMKSPT	DMKSRM	DMKSSP
		DMKSSS	DMKSST	DMKSSV	DMKSTA	DMKSTK	DMKSTP	DMKSTR	DMKSVD	DMKSWA	DMKSWM
		DMKTAP	DMKTAQ	DMKTCS	DMKTCT	DMKTDK	DMKTEE	DMKTES	DMKTHI	DMKTMR	DMKTOD
		DMKTRC	DMKTRD	DMKTRK	DMKTRP	DMKTRQ	DMKTRR	DMKTRT	DMKTRU	DMKTRX	DMKTTX
		DMKTTY	DMKUDR	DMKUDU	DMKUNT	DMKURS	DMKUSP	DMKUSQ	DMKVAT	DMKVAU	DMKVCA
		DMKVCB	DMKVCH	DMKVCN	DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCU	DMKVCV
		DMKVCW	DMKVCX	DMKVDA	DMKVDB	DMKVDC	DMKVDD	DMKVDE	DMKVDF	DMKVDG	DMKVDH
		DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVFC	DMKVFD	DMKVFE	DMKVFR	DMKVFS	DMKVIO
		DMKVMA	DMKVMC	DMKVMD	DMKVME	DMKVMG	DMKVMI	DMKVRR	DMKVSC	DMKVSD	DMKVSE
		DMKVSF	DMKVSG	DMKVSI	DMKVSJ	DMKVSP	DMKVSQ	DMKVST	DMKVSU	DMKVSV	DMKVSW
		DMKVSX	DMKWAI	DMKWRM	DMKWRN	DMKXAB	DMKXAD	DMKXST	DMKZTD		
R7	008835	DMKACO	DMKACR	DMKACS	DMKALG	DMKAPS	DMKAPT	DMKAPU	DMKAPV	DMKAPW	DMKAPX
		DMKAPY	DMKATS	DMKBIO	DMKBLD	DMKBSC	DMKCAC	DMKCAO	DMKCCD	DMKCCF	DMKCCH
		DMKCCO	DMKCCS	DMKCCT	DMKCCW	DMKCDB	DMKCDM	DMKCDS	DMKCFC	DMKCFD	DMKCFF
		DMKCFG	DMKCFH	DMKCFM	DMKCFO	DMKCFP	DMKCFQ	DMKCFR	DMKCFS	DMKCFT	DMKCFU
		DMKCFV	DMKCFW	DMKCFY	DMKCKD	DMKCKF	DMKCKM	DMKCKN	DMKCKR	DMKCKS	DMKCKT
		DMKCKV	DMKCKW	DMKCNS	DMKCNT	DMKCPB	DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPP
		DMKCPS	DMKCPT	DMKCPU	DMKCPV	DMKCPW	DMKCPX	DMKCPY	DMKCPZ	DMKCQC	DMKCQG
		DMKCQP	DMKCQQ	DMKCQR	DMKCQS	DMKCQT	DMKCQU	DMKCQY	DMKCRM	DMKCSB	DMKCSC

LABEL	COUNT	REFERENCES									
		DMKCSF	DMKCSO	DMKCSQ	DMKCSR	DMKCST	DMKCSU	DMKCSV	DMKCSW	DMKCSX	DMKCSY
		DMKCVU	DMKDAD	DMKDAS	DMKDAU	DMKDDC	DMKDDR	DMKDEF	DMKDEG	DMKDEX	DMKDGD
		DMKDIA	DMKDIB	DMKDID	DMKDIF	DMKDIR	DMKDMP	DMKDMQ	DMKDNC	DMKDRD	DMKDRE
		DMKDSB	DMKDSP	DMKENT	DMKERM	DMKERP	DMKEXT	DMKFMT	DMKFPS	DMKFRE	DMKFRT
		DMKGIO	DMKGRA	DMKGRC	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRI	DMKGRT	DMKHPS
		DMKHPT	DMKHPU	DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKIDR	DMKIDU	DMKIMG	DMKIOE
		DMKIOF	DMKIOG	DMKIOH	DMKIOJ	DMKIOQ	DMKIOS	DMKIOT	DMKISM	DMKIUA	DMKIUC
		DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS	DMKJRL	DMKLDOOE	DMKLNK
		DMKLNM	DMKLOC	DMKLOG	DMKLOH	DMKLOJ	DMKLOM	DMKMCC	DMKMCD	DMKMCH	DMKMCT
		DMKMHV	DMKMIA	DMKMID	DMKMNI	DMKMNJ	DMKMNL	DMKMNT	DMKMON	DMKMOO	DMKMPO
		DMKMSG	DMKMSW	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKNMT	DMKOPR	DMKOVR
		DMKPAG	DMKPAH	DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPER	DMKPET	DMKPGM	DMKPGS
		DMKPGT	DMKPGU	DMKPMA	DMKPRG	DMKPRV	DMKPRW	DMKPSA	DMKPST	DMKPTR	DMKPTS
		DMKPTT	DMKQCN	DMKQCO	DMKQCQ	DMKQVM	DMKREI	DMKRET	DMKRGA	DMKRGB	DMKRGC
		DMKRGD	DMKRGE	DMKRND	DMKRNH	DMKRPA	DMKRSE	DMKRSF	DMKRSP	DMKRSQ	DMKRST
		DMKSAD	DMKSAV	DMKSBL	DMKSCH	DMKSCN	DMKSCO	DMKSEG	DMKSEL	DMKSEP	DMKSFB
		DMKSNC	DMKSND	DMKSPK	DMKSPL	DMKSPM	DMKSPS	DMKSPT	DMKSRM	DMKSSP	DMKSSS
		DMKSST	DMKSSV	DMKSTA	DMKSTK	DMKSTP	DMKSTR	DMKSVC	DMKSVD	DMKSWA	DMKSWM
		DMKTAP	DMKTAQ	DMKTCS	DMKTCT	DMKTDK	DMKTES	DMKTHI	DMKTMR	DMKTOD	DMKTPE
		DMKTRC	DMKTRD	DMKTRK	DMKTRP	DMKTRQ	DMKTRR	DMKTRT	DMKTRU	DMKTRX	DMKTTX
		DMKTTY	DMKUDR	DMKUDU	DMKUNT	DMKURS	DMKUSP	DMKUSQ	DMKVAT	DMKVAU	DMKVCA
		DMKVCB	DMKVCH	DMKVCN	DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCU	DMKVCV
		DMKVCW	DMKVCX	DMKVDA	DMKVDB	DMKVDC	DMKVDD	DMKVDE	DMKVDF	DMKVDG	DMKVDH
		DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVFC	DMKVFD	DMKVFE	DMKVFR	DMKVFS	DMKVIO
		DMKVMA	DMKVMC	DMKVMD	DMKVME	DMKVMG	DMKVMI	DMKVRR	DMKVRS	DMKVSC	DMKVSD
		DMKVSE	DMKVSF	DMKVSG	DMKVSI	DMKVSJ	DMKVSP	DMKVSQ	DMKVST	DMKVSU	DMKVSW
		DMKVSX	DMKWAI	DMKWRM	DMKWRN	DMKXAB	DMKXAD	DMKXST	DMKZTD		
R8	006535	DMKACO	DMKACR	DMKACS	DMKALG	DMKALO	DMKAPI	DMKAPS	DMKAPT	DMKAPU	DMKAPV
		DMKAPY	DMKAPZ	DMKATS	DMKBIO	DMKBLD	DMKBSC	DMKCAC	DMKCAO	DMKCCD	DMKCCF
		DMKCCH	DMKCCO	DMKCCS	DMKCCT	DMKCCW	DMKCDB	DMKCDM	DMKCDS	DMKCFC	DMKCFD
		DMKCFF	DMKCFG	DMKCFH	DMKCFM	DMKCFO	DMKCFP	DMKCFQ	DMKCFR	DMKCFS	DMKCFT
		DMKCFU	DMKCFV	DMKCFW	DMKCFY	DMKCKD	DMKCKF	DMKCKH	DMKCKM	DMKCKN	DMKCKR
		DMKCKS	DMKCKT	DMKCKV	DMKCKW	DMKCNS	DMKCNT	DMKCPB	DMKCPI	DMKCPJ	DMKCPM
		DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPU	DMKCPV	DMKCPW	DMKCPX	DMKCPY
		DMKCPZ	DMKCQC	DMKCQG	DMKCQH	DMKCQI	DMKCQP	DMKCQQ	DMKCQR	DMKCQS	DMKCQT
		DMKCQU	DMKCQY	DMKCRM	DMKCSB	DMKCSC	DMKCSF	DMKCSO	DMKCSP	DMKCSQ	DMKCSR
		DMKCST	DMKCSU	DMKCSV	DMKCSW	DMKCSX	DMKCSY	DMKCVU	DMKDAD	DMKDAS	DMKDAU
		DMKDDC	DMKDDR	DMKDEF	DMKDEI	DMKDEX	DMKDGD	DMKDGF	DMKDIA	DMKDIB	DMKDID
		DMKDIF	DMKDIR	DMKDMP	DMKDMQ	DMKDNC	DMKDRD	DMKDRE	DMKDSB	DMKDSP	DMKENT
		DMKEPS	DMKERM	DMKERP	DMKEXT	DMKFMT	DMKFPS	DMKFRE	DMKFRT	DMKGIO	DMKGRA
		DMKGRC	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRH	DMKGRI	DMKGRT	DMKHPS	DMKHPT
		DMKHPU	DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKIDR	DMKIDU	DMKIMG	DMKIOC	DMKIOE
		DMKIOF	DMKIOG	DMKIOH	DMKIOJ	DMKIOQ	DMKIOS	DMKIOT	DMKISM	DMKIUA	DMKIUB
		DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS	DMKJRL	DMKLDOOE
		DMKLNK	DMKLNM	DMKLOC	DMKLOG	DMKLOH	DMKLOJ	DMKLOM	DMKMCC	DMKMCD	DMKMCH
		DMKMCT	DMKMHC	DMKMHV	DMKMIA	DMKMID	DMKMNI	DMKMNJ	DMKMNL	DMKMNT	DMKMON
		DMKMOO	DMKMPO	DMKMSG	DMKMSW	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKNMT
		DMKOPE	DMKOPR	DMKOVR	DMKPAG	DMKPAH	DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPER
		DMKPET	DMKPGM	DMKPGS	DMKPGT	DMKPGU	DMKPMA	DMKPRG	DMKPRV	DMKPRW	DMKPST
		DMKPTR	DMKPTS	DMKPTT	DMKQCN	DMKQCO	DMKQCP	DMKQCQ	DMKQVM	DMKREI	DMKRET
		DMKRGA	DMKRGB	DMKRGC	DMKRGD	DMKRGE	DMKRNH	DMKRSE	DMKRSF	DMKRSP	DMKRSQ
		DMKRST	DMKSAD	DMKSAV	DMKSBL	DMKSCH	DMKSCN	DMKSCO	DMKSEG	DMKSEL	DMKSEP
		DMKSFB	DMKSNC	DMKSND	DMKSPK	DMKSPL	DMKSPM	DMKSPS	DMKSPT	DMKSRM	DMKSSP
		DMKSSS	DMKSST	DMKSSU	DMKSSV	DMKSTA	DMKSTK	DMKSTP	DMKSTR	DMKSVC	DMKSVD
		DMKSWA	DMKTAP	DMKTAQ	DMKTCS	DMKTCT	DMKTDK	DMKTEE	DMKTEF	DMKTEM	DMKTES
		DMKTHI	DMKTMR	DMKTOD	DMKTPE	DMKTRC	DMKTRD	DMKTRK	DMKTRM	DMKTRP	DMKTRQ
		DMKTRR	DMKTRT	DMKTRX	DMKTTX	DMKTTY	DMKUDR	DMKUDU	DMKUNT	DMKURS	DMKUSP
		DMKUSQ	DMKVAT	DMKVAU	DMKVCA	DMKVCB	DMKVCH	DMKVCN	DMKVCP	DMKVCQ	DMKVCR

LABEL	COUNT	REFERENCES									
		DMKVCS	DMKVCT	DMKVCU	DMKVCV	DMKVCW	DMKVCX	DMKVDA	DMKVDB	DMKVDC	DMKVDD
		DMKVDE	DMKVDF	DMKVDG	DMKVDH	DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVFC	DMKVFD
		DMKVFE	DMKVFS	DMKVIO	DMKVMA	DMKVMC	DMKVMD	DMKVME	DMKVMI	DMKVRR	DMKVRS
		DMKVSC	DMKVSD	DMKVSE	DMKVSF	DMKVSG	DMKVSI	DMKVSJ	DMKVSP	DMKVSQ	DMKVST
		DMKVSU	DMKVSV	DMKVSW	DMKVSX	DMKWAI	DMKWRM	DMKWRN	DMKXAB	DMKXAD	DMKXST
		DMKZTD									
R9	006154	DMKACO	DMKACR	DMKACS	DMKALG	DMKAPI	DMKAPS	DMKAPT	DMKAPU	DMKAPV	DMKAPX
		DMKAPY	DMKATS	DMKBIO	DMKBLD	DMKBSC	DMKCAC	DMKCAO	DMKCCH	DMKCCS	DMKCCW
		DMKCDM	DMKCDS	DMKCFC	DMKCFF	DMKCFG	DMKCFH	DMKCFM	DMKCFO	DMKCFP	DMKCFQ
		DMKCFR	DMKCFS	DMKCFU	DMKCFV	DMKCFW	DMKCFY	DMKCKD	DMKCKF	DMKCKH	DMKCKM
		DMKCKN	DMKCKR	DMKCKS	DMKCKV	DMKCKW	DMKCNS	DMKCNT	DMKCPI	DMKCPJ	DMKCPM
		DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPU	DMKCPV	DMKCPW	DMKCPY	DMKCPZ
		DMKCQG	DMKCQH	DMKCQI	DMKCQP	DMKCQQ	DMKCQR	DMKCQS	DMKCQT	DMKCQU	DMKCQY
		DMKCRM	DMKCSB	DMKCSF	DMKCSO	DMKCSP	DMKCSQ	DMKCSR	DMKCST	DMKCSU	DMKCSV
		DMKCSW	DMKCSX	DMKCSY	DMKCVU	DMKDAD	DMKDAS	DMKDAU	DMKDDC	DMKDDR	DMKDEF
		DMKDEX	DMKDGD	DMKDGF	DMKDIB	DMKDID	DMKDIF	DMKDIR	DMKDMP	DMKDMQ	DMKDNC
		DMKDRD	DMKDRE	DMKDSP	DMKEIG	DMKENT	DMKEPS	DMKERM	DMKERP	DMKEXT	DMKFMT
		DMKFPS	DMKFRE	DMKFRT	DMKGIO	DMKGRA	DMKGRC	DMKGRD	DMKGRE	DMKGRF	DMKGRG
		DMKGRI	DMKGRT	DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKIDR
		DMKIDU	DMKIMG	DMKIOC	DMKIOE	DMKIOF	DMKIOG	DMKIOH	DMKIOJ	DMKIOS	DMKIOT
		DMKISM	DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUP	DMKIUS
		DMKJRL	DMKLD00E	DMKLNK	DMKLNM	DMKLOC	DMKLOG	DMKLOH	DMKLOJ	DMKLOK	DMKLOM
		DMKMCC	DMKMCD	DMKMCH	DMKMCI	DMKMCT	DMKMHC	DMKMHV	DMKMIA	DMKMID	DMKMNI
		DMKMNL	DMKMNT	DMKMON	DMKMOO	DMKMPO	DMKMSG	DMKMSW	DMKNES	DMKNET	DMKNLD
		DMKNLE	DMKNMT	DMKOPE	DMKOPR	DMKOVR	DMKPAG	DMKPAH	DMKPEQ	DMKPER	DMKPET
		DMKPGM	DMKPGS	DMKPGT	DMKPGU	DMKPMA	DMKPRG	DMKPRV	DMKPRW	DMKPST	DMKPTR
		DMKPTS	DMKPTT	DMKQCN	DMKQCO	DMKQCQ	DMKQVM	DMKREI	DMKRGA	DMKRGB	DMKRGC
		DMKRGD	DMKRGE	DMKRNH	DMKRPA	DMKRSE	DMKRSF	DMKRSP	DMKRSQ	DMKRST	DMKSAD
		DMKSAV	DMKSCH	DMKSCN	DMKSCO	DMKSEG	DMKSEL	DMKSEP	DMKSEV	DMKSFB	DMKSIX
		DMKSNC	DMKSND	DMKSPC	DMKSPK	DMKSPL	DMKSPM	DMKSPR	DMKSPS	DMKSRM	DMKSSP
		DMKSSS	DMKSST	DMKSSV	DMKSTA	DMKSTP	DMKSTR	DMKSVD	DMKSWA	DMKSWM	DMKTAP
		DMKTAQ	DMKTCS	DMKTCT	DMKTDK	DMKTEE	DMKTEF	DMKTES	DMKTHI	DMKTMR	DMKTOD
		DMKTRA	DMKTRC	DMKTRD	DMKTRK	DMKTRP	DMKTRQ	DMKTRR	DMKTRT	DMKTRU	DMKTRX
		DMKTTX	DMKTTY	DMKUDR	DMKUDU	DMKUNT	DMKURS	DMKUSP	DMKUSQ	DMKVAT	DMKVAU
		DMKVCA	DMKVCB	DMKVCH	DMKVCN	DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCU
		DMKVCV	DMKVCW	DMKVCX	DMKVDA	DMKVDB	DMKVDC	DMKVDD	DMKVDE	DMKVDF	DMKVDG
		DMKVDH	DMKVDS	DMKVDT	DMKVER	DMKVFC	DMKVFE	DMKVFR	DMKVFS	DMKVIO	DMKVMA
		DMKVMC	DMKVMD	DMKVME	DMKVMG	DMKVMI	DMKVSC	DMKVSE	DMKVSF	DMKVSG	DMKVSI
		DMKVSJ	DMKVSP	DMKVSQ	DMKVST	DMKVSU	DMKVSV	DMKVSW	DMKVSX	DMKWAI	DMKWRM
		DMKXAB	DMKXAD	DMKXST	DMKZTD						
SAS	000014	DMKCDS	DMKDSP	DMKEXT	DMKFPS	DMKIOT	DMKPRG	DMKPSA	DMKSVD		
SASPF	000002	DMKPRG	DMKVAT								
SAVCNT	000002	DMKCKW									
SAVCREGS	000014	DMKCFF	DMKCFH	DMKCKM							
SAVDATE	000005	DMKCFF	DMKCFH	DMKCKM	DMKCQY						
SAVEAR	000007	DMKIMG	DMKRND								
SAVEAREA	000526	DMKACO	DMKALG	DMKALO	DMKAPI	DMKAPS	DMKAPT	DMKAPU	DMKAPV	DMKAPW	DMKAPX
		DMKAPY	DMKAPZ	DMKATS	DMKBIO	DMKBLD	DMKBSC	DMKCAC	DMKCAO	DMKCCD	DMKCCF
		DMKCCH	DMKCCO	DMKCCS	DMKCCT	DMKCCW	DMKCDB	DMKCDM	DMKCDS	DMKCFC	DMKCFD
		DMKCFF	DMKCFG	DMKCFH	DMKCFJ	DMKCFM	DMKCFO	DMKCFP	DMKCFQ	DMKCFR	DMKCFS
		DMKCFT	DMKCFU	DMKCFV	DMKCFW	DMKCFY	DMKCKR	DMKCKS	DMKCKT	DMKCKV	DMKCLK
		DMKCNS	DMKCPB	DMKCPJ	DMKCPM	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPU
		DMKCPV	DMKCPW	DMKCPX	DMKCPY	DMKCPZ	DMKCQC	DMKCQG	DMKCQH	DMKCQI	DMKCQP
		DMKCQQ	DMKCQR	DMKCQS	DMKCQT	DMKCQU	DMKCQY	DMKCRM	DMKCSB	DMKCSC	DMKCSF
		DMKCSO	DMKCSP	DMKCSQ	DMKCSR	DMKCST	DMKCSU	DMKCSV	DMKCSW	DMKCSX	DMKCSY
		DMKCVU	DMKDAD	DMKDAS	DMKDAU	DMKDEF	DMKDEG	DMKDEI	DMKDEX	DMKDGD	DMKDIA
		DMKDIB	DMKDID	DMKDIF	DMKDRD	DMKDRE	DMKDSB	DMKEIG	DMKENT	DMKEPS	DMKERM
		DMKGIO	DMKGRA	DMKGRC	DMKGRG	DMKGRI	DMKGRT	DMKHPS	DMKHPT	DMKHPU	DMKHVD

LABEL	COUNT	REFERENCES									
		DMKHVE	DMKHVF	DMKIDR	DMKIDU	DMKIOE	DMKIOF	DMKIOG	DMKIOH	DMKIOQ	DMKIOS
		DMKIOT	DMKISM	DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN
		DMKIUP	DMKIUS	DMKJRL	DMKLNK	DMKLNM	DMKLOC	DMKLOG	DMKLOH	DMKLOJ	DMKLOM
		DMKMCC	DMKMCD	DMKMCI	DMKMHC	DMKMHV	DMKMIA	DMKMID	DMKMNI	DMKMNJ	DMKMNL
		DMKMNT	DMKMON	DMKMOO	DMKMSG	DMKMSW	DMKNEA	DMKNEM	DMKNES	DMKNET	DMKNLD
		DMKNLE	DMKOPE	DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPER	DMKPET	DMKPGM	DMKPGS
		DMKPMA	DMKPST	DMKPTR	DMKPTS	DMKPTT	DMKQCN	DMKQCO	DMKQCP	DMKQCQ	DMKQVM
		DMKREI	DMKRGA	DMKRGB	DMKRGC	DMKRGE	DMKRNH	DMKRPA	DMKRPD	DMKRPI	DMKRPW
		DMKRSE	DMKRSF	DMKRSP	DMKRSQ	DMKRST	DMKSBL	DMKSCO	DMKSEG	DMKSEL	DMKSEP
		DMKSEV	DMKSFB	DMKSIX	DMKSNC	DMKSND	DMKSPC	DMKSPK	DMKSPL	DMKSPM	DMKSPR
		DMKSPS	DMKSPT	DMKSRM	DMKSSP	DMKSSS	DMKSST	DMKSSU	DMKSSV	DMKSTP	DMKSTR
		DMKSVC	DMKSWA	DMKSWM	DMKTAP	DMKTAQ	DMKTCS	DMKTCT	DMKTDK	DMKTHI	DMKTOD
		DMKTPE	DMKTRA	DMKTRC	DMKTRD	DMKTRK	DMKTRM	DMKTRP	DMKTRQ	DMKTRU	DMKTRX
		DMKTTX	DMKTTY	DMKUDR	DMKUDU	DMKUNT	DMKURS	DMKUSP	DMKUSQ	DMKVAT	DMKVAU
		DMKVBM	DMKVCA	DMKVCB	DMKVCH	DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCU
		DMKVCV	DMKVCW	DMKVCX	DMKVDA	DMKVDB	DMKVDC	DMKVDD	DMKVDE	DMKVDF	DMKVDG
		DMKVDH	DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVFC	DMKVFD	DMKVFE	DMKVFR	DMKVFS
		DMKVMA	DMKVMC	DMKVMD	DMKVME	DMKVMG	DMKVRR	DMKVSC	DMKVSD	DMKVSE	DMKVSF
		DMKVSG	DMKVSP	DMKVSQ	DMKVSR	DMKVST	DMKVSU	DMKVSW	DMKVSX	DMKWRM	DMKWRN
		DMKXAB	DMKXAD	DMKXST	DMKZTD						
SAVEPROC	000012	DMKPTR	DMKSVC								
SAVEREGS	001334	DMKACO	DMKALG	DMKALO	DMKAPI	DMKAPS	DMKAPT	DMKAPU	DMKAPV	DMKAPW	DMKAPX
		DMKAPY	DMKAPZ	DMKATS	DMKBIO	DMKBLD	DMKBSC	DMKCAC	DMKCAO	DMKCCH	DMKCCS
		DMKCCW	DMKCDB	DMKCDM	DMKCDS	DMKCFC	DMKCFD	DMKCFF	DMKCFG	DMKCFH	DMKCFJ
		DMKCFM	DMKCFO	DMKCFP	DMKCFQ	DMKCFR	DMKCFS	DMKCFT	DMKCFU	DMKCFV	DMKCFW
		DMKCFY	DMKCKR	DMKCKS	DMKCKT	DMKCKV	DMKCLK	DMKCPB	DMKCPJ	DMKCPM	DMKCPN
		DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPU	DMKCPV	DMKCPW	DMKCPX	DMKCPY	DMKCPZ
		DMKCQC	DMKCQG	DMKCQH	DMKCQI	DMKCQP	DMKCQQ	DMKCQR	DMKCQS	DMKCQT	DMKCQU
		DMKCQY	DMKCRM	DMKCSB	DMKCSC	DMKCSF	DMKCSO	DMKCSP	DMKCSQ	DMKCSR	DMKCST
		DMKCSU	DMKCSV	DMKCSW	DMKCSX	DMKCSY	DMKCVU	DMKDAD	DMKDAS	DMKDAU	DMKDEF
		DMKDEG	DMKDEI	DMKDEX	DMKDGD	DMKDIA	DMKDIB	DMKDID	DMKDIF	DMKDRD	DMKDRE
		DMKDSB	DMKENT	DMKEPS	DMKERM	DMKGIO	DMKGRA	DMKGRC	DMKGRT	DMKHPS	DMKHPT
		DMKHPU	DMKHVD	DMKHVE	DMKHVF	DMKIDR	DMKIDU	DMKIOE	DMKIOF	DMKIOG	DMKIOH
		DMKIOS	DMKISM	DMKIUA	DMKIUB	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN
		DMKIUP	DMKIUS	DMKJRL	DMKLNK	DMKLNM	DMKLOC	DMKLOG	DMKLOH	DMKLOJ	DMKLOM
		DMKMCC	DMKMCD	DMKMCI	DMKMHC	DMKMHV	DMKMIA	DMKMID	DMKMNI	DMKMNJ	DMKMNL
		DMKMNT	DMKMON	DMKMOO	DMKMSG	DMKMSW	DMKNEA	DMKNEM	DMKNES	DMKNET	DMKNLD
		DMKNLE	DMKOPE	DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPER	DMKPET	DMKPGM	DMKPGS
		DMKPMA	DMKPST	DMKPTR	DMKPTS	DMKPTT	DMKQCN	DMKQCO	DMKQCP	DMKQVM	DMKREI
		DMKRGB	DMKRGE	DMKRNH	DMKRPA	DMKRPD	DMKRPI	DMKRPW	DMKRSE	DMKRSF	DMKRSP
		DMKRSQ	DMKRST	DMKSBL	DMKSCO	DMKSEG	DMKSEL	DMKSEP	DMKSFB	DMKSNC	DMKSND
		DMKSPC	DMKSPK	DMKSPL	DMKSPM	DMKSPR	DMKSPT	DMKSRM	DMKSSS	DMKSST	DMKSSU
		DMKSSV	DMKSTP	DMKSTR	DMKSWA	DMKSWM	DMKTAP	DMKTAQ	DMKTCS	DMKTCT	DMKTDK
		DMKTHI	DMKTOD	DMKTPE	DMKTRA	DMKTRC	DMKTRD	DMKTRK	DMKTRM	DMKTRP	DMKTRU
		DMKTTX	DMKTTY	DMKUDR	DMKUDU	DMKUNT	DMKURS	DMKUSP	DMKUSQ	DMKVAT	DMKVAU
		DMKVBM	DMKVCA	DMKVCB	DMKVCH	DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCU
		DMKVCV	DMKVCW	DMKVCX	DMKVDA	DMKVDB	DMKVDC	DMKVDD	DMKVDE	DMKVDF	DMKVDG
		DMKVDH	DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVFC	DMKVFD	DMKVFE	DMKVFR	DMKVFS
		DMKVMC	DMKVMD	DMKVME	DMKVMG	DMKVRR	DMKVSD	DMKVSE	DMKVSF	DMKVSG	DMKVSP
		DMKVSQ	DMKVST	DMKVSU	DMKVSW	DMKVSX	DMKWRM	DMKWRN	DMKXAB	DMKXAD	DMKXST
		DMKZTD									
SAVERET	000021	DMKDDR	DMKDIR								
SAVERETN	000051	DMKCFG	DMKCFJ	DMKCFQ	DMKCPB	DMKDIF	DMKIOS	DMKIUA	DMKLNK	DMKLOC	DMKLOG
		DMKPGM	DMKPTR	DMKQCO	DMKSSS	DMKSTP	DMKSVC	DMKVAT	DMKVCA	DMKVCP	DMKVCQ
		DMKVCR	DMKVCS	DMKVER							
SAVERTN	000003	DMKSVC	DMKVSQ								
SAVERO	000166	DMKCKS	DMKCKT	DMKCNS	DMKCQG	DMKCQQ	DMKCQR	DMKCQY	DMKDAD	DMKDAS	DMKDAU
		DMKDEG	DMKDIA	DMKDIB	DMKDRD	DMKDRE	DMKEPS	DMKERM	DMKGRG	DMKGRT	DMKHVD

LABEL	COUNT	REFERENCES									
		DMKHVE	DMKHVF	DMKIUA	DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUP	DMKIUS
		DMKJRL	DMKLNK	DMKLNM	DMKLOH	DMKMON	DMKMSW	DMKNEM	DMKPGM	DMKPMA	DMKPTR
		DMKQCN	DMKQCO	DMKRGC	DMKRNH	DMKSCO	DMKSFB	DMKSNC	DMKSTP	DMKSTR	DMKSVC
		DMKTAP	DMKTDK	DMKTPE	DMKTRC	DMKTRD	DMKUDR	DMKVAU	DMKVBM	DMKVCA	DMKVCQ
		DMKVCR	DMKVCT	DMKVCV	DMKVDA	DMKVDC	DMKVFE	DMKVSD	DMKVSF	DMKVSG	DMKVST
SAVER1	000231	DMKACR	DMKBLD	DMKCCW	DMKCFC	DMKCFR	DMKCKS	DMKCKT	DMKCNS	DMKCPO	DMKCPP
		DMKCPU	DMKCPZ	DMKCQQ	DMKCSB	DMKCSC	DMKDEG	DMKERM	DMKHPU	DMKHVF	DMKIDR
		DMKIOE	DMKIUA	DMKIUG	DMKIUJ	DMKIUP	DMKLNK	DMKLNM	DMKMHC	DMKMIA	DMKMON
		DMKMSG	DMKPGM	DMKPGS	DMKPMA	DMKPTR	DMKPTT	DMKQCN	DMKQCO	DMKRNH	DMKRPA
		DMKSSS	DMKSTP	DMKSTR	DMKTDK	DMKTRC	DMKTRD	DMKTRK	DMKVAU	DMKVBM	DMKVCB
		DMKVCP	DMKVCV	DMKVCX	DMKVDE	DMKVDF	DMKVDR	DMKVDS	DMKVDT	DMKVFC	DMKVFE
		DMKVSD	DMKVSE	DMKVST	DMKVSU						
SAVER10	000050	DMKACO	DMKBLD	DMKCCD	DMKCCF	DMKCCH	DMKCCO	DMKCCS	DMKCCW	DMKHPU	DMKLNK
		DMKLOG	DMKLOJ	DMKSEP	DMKVBM	DMKVCA	DMKVCH	DMKVDC	DMKVDE	DMKVDG	DMKVDS
		DMKVFE									
SAVER11	000165	DMKALG	DMKAPI	DMKBLD	DMKCAO	DMKCFM	DMKCFO	DMKCFQ	DMKCFR	DMKCPN	DMKCPO
		DMKCPP	DMKCPS	DMKCPU	DMKCPY	DMKCQG	DMKCQQ	DMKCSR	DMKCSU	DMKCSX	DMKDAD
		DMKDAS	DMKDAU	DMKDIA	DMKDIB	DMKDIF	DMKEPS	DMKHPS	DMKHPT	DMKHVF	DMKIOQ
		DMKIOS	DMKLNK	DMKLOG	DMKLOH	DMKLOJ	DMKMID	DMKMSG	DMKMSW	DMKNES	DMKNLD
		DMKPGM	DMKQCN	DMKQCO	DMKSCO	DMKSEL	DMKSND	DMKSPK	DMKSPL	DMKSPM	DMKSSS
		DMKSTR	DMKSWA	DMKTCS	DMKTCT	DMKTHI	DMKUNT	DMKVCA	DMKVCB	DMKVCH	DMKVCP
		DMKVCS	DMKVCT	DMKVDA	DMKVDB	DMKVDD	DMKVDE	DMKVDT	DMKVMC		
SAVER12	000034	DMKCCS	DMKCCW	DMKCFQ	DMKCPV	DMKDGD	DMKIDR	DMKLNK	DMKMCC	DMKMCD	DMKPGM
		DMKPGS	DMKPTR	DMKSTP	DMKSVC	DMKVAT	DMKVCA	DMKVER			
SAVER13	000046	DMKCFF	DMKCFR	DMKCPN	DMKCPW	DMKCQT	DMKCSY	DMKDIF	DMKHPU	DMKLNK	DMKLNM
		DMKPGM	DMKPGS	DMKPTR	DMKSEL	DMKSTP	DMKSVC	DMKTTX	DMKVAT	DMKVCU	DMKVDB
		DMKVDC	DMKVDE	DMKVDF	DMKVDS						
SAVER14	000002	DMKCFU									
SAVER2	000417	DMKACO	DMKALG	DMKBLD	DMKCCW	DMKCDB	DMKCDS	DMKCFC	DMKCFM	DMKCFS	DMKCKR
		DMKCKS	DMKCKV	DMKCNS	DMKCPT	DMKCQG	DMKCQH	DMKCQP	DMKCQQ	DMKCSR	DMKCSW
		DMKCSY	DMKDEF	DMKDEG	DMKDEI	DMKDGD	DMKDIA	DMKDIB	DMKDRD	DMKDRE	DMKERM
		DMKGIO	DMKGRI	DMKHVF	DMKIUG	DMKJRL	DMKLNK	DMKLNM	DMKLOG	DMKLOH	DMKLOJ
		DMKLOM	DMKMHC	DMKMIA	DMKMNJ	DMKMSG	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKNLE
		DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPET	DMKPGS	DMKPMA	DMKPTR	DMKPTT	DMKQCN
		DMKQCO	DMKQCQ	DMKREI	DMKRGA	DMKRGB	DMKRGC	DMKRNH	DMKRPA	DMKRSQ	DMKSEL
		DMKSFB	DMKSNC	DMKSPT	DMKSSS	DMKSST	DMKSTR	DMKSWA	DMKTRA	DMKTRC	DMKTRD
		DMKTRP	DMKTTX	DMKUDR	DMKVAU	DMKVCH	DMKVCQ	DMKVCR	DMKVCS	DMKVCU	DMKVDA
		DMKVDC	DMKVDD	DMKVDE	DMKVDG	DMKVDS	DMKVDT	DMKVMC	DMKVSD	DMKVSE	DMKVSF
		DMKVSQ	DMKVST	DMKVSU	DMKWRM						
SAVER3	000044	DMKCFC	DMKCPN	DMKCSW	DMKERM	DMKLNK	DMKLOH	DMKPGS	DMKPTR	DMKPTT	DMKQCN
		DMKQCQ	DMKRPA	DMKSCO	DMKSST	DMKSTR	DMKTTX	DMKUDU	DMKVAU	DMKVDT	
SAVER4	000016	DMKCCH	DMKCFC	DMKCFQ	DMKCPN	DMKQCN	DMKQCO	DMKQCQ	DMKTRC	DMKVDA	DMKVDE
		DMKVDG									
SAVER5	000023	DMKAPT	DMKAPV	DMKCFG	DMKCPN	DMKHPT	DMKHVE	DMKLOH	DMKLOJ	DMKMON	DMKSST
		DMKTRC	DMKVCB	DMKVDT	DMKVMC						
SAVER6	000036	DMKACO	DMKCCH	DMKCFF	DMKCFG	DMKCSR	DMKCSW	DMKDRD	DMKDRE	DMKGRA	DMKHVE
		DMKHVF	DMKIOF	DMKLNK	DMKLOH	DMKRSF	DMKSCO	DMKSNC	DMKSST	DMKVCV	DMKVDE
		DMKVMC	DMKVSW	DMKVSX							
SAVER7	000028	DMKCCH	DMKCSW	DMKLNK	DMKLOH	DMKMSW	DMKPGM	DMKPTR	DMKQCN	DMKRSF	DMKSPL
		DMKTRK	DMKVCQ	DMKVCV	DMKVDE	DMKVSD	DMKVSG	DMKVSP	DMKVSQ	DMKVST	DMKVSW
		DMKVSX									
SAVER8	000076	DMKBLD	DMKCCH	DMKCCS	DMKCCW	DMKCFQ	DMKCFR	DMKCKS	DMKCPZ	DMKDIB	DMKGRT
		DMKHPT	DMKLNK	DMKQCN	DMKQCO	DMKSCO	DMKSEP	DMKSPL	DMKTCT	DMKTDK	DMKTRD
		DMKTRK	DMKUNT	DMKVDG	DMKVDS	DMKVDT	DMKVSP	DMKVSQ	DMKVST	DMKVSW	DMKVSX
		DMKZTD									
SAVER9	000041	DMKALG	DMKCCW	DMKCPT	DMKCPW	DMKCQP	DMKCQQ	DMKCSR	DMKGRT	DMKNEA	DMKNES
		DMKNET	DMKPGM	DMKSND	DMKSPL	DMKUSQ	DMKVCV	DMKVCX	DMKVDR	DMKVSP	DMKVSQ
		DMKVST	DMKVSW	DMKZTD							

LABEL	COUNT	REFERENCES									
SAVESIZE	000036	DMKCPI	DMKIUA	DMKLNK	DMKLNM	DMKQCO	DMKSTA	DMKSVC	DMKSWA	DMKVCP	DMKVCQ
		DMKVCR	DMKVCS	DMKVCT	DMKVDA						
SAVETRAC	000007	DMKSVC									
SAVEVAC	000005	DMKACO	DMKCKF								
SAVEWRK1	002746	DMKACO	DMKALG	DMKAPX	DMKAPY	DMKATS	DMKBLD	DMKCAC	DMKCAO	DMKCCH	DMKCCW
		DMKCDB	DMKCDM	DMKCDS	DMKCFC	DMKCFD	DMKCFF	DMKCFG	DMKCFH	DMKCFO	DMKCFP
		DMKCFQ	DMKCFR	DMKCFS	DMKCFT	DMKCFU	DMKCFV	DMKCFY	DMKCKR	DMKCKS	DMKCKT
		DMKCKV	DMKCPN	DMKCPO	DMKCPS	DMKCPT	DMKCPU	DMKCPV	DMKCPW	DMKCPX	DMKCPY
		DMKCPZ	DMKCQC	DMKCQG	DMKCQH	DMKCQI	DMKCQP	DMKCQQ	DMKCQS	DMKCQT	DMKCQY
		DMKCSB	DMKCSF	DMKCSO	DMKCSP	DMKCSQ	DMKCST	DMKCSU	DMKCSV	DMKCSW	DMKCSX
		DMKDAD	DMKDAS	DMKDAU	DMKDEF	DMKDEG	DMKDEI	DMKDEX	DMKDIA	DMKDIB	DMKDIF
		DMKDRD	DMKDRE	DMKEIG	DMKEPS	DMKERM	DMKGRC	DMKHPS	DMKHPU	DMKHVF	DMKIDR
		DMKIOQ	DMKIOS	DMKIOT	DMKIUB	DMKIUC	DMKIUE	DMKIUJ	DMKIUL	DMKIUN	DMKIUP
		DMKIUS	DMKJRL	DMKLNK	DMKLNM	DMKLOC	DMKLOG	DMKLOH	DMKLOJ	DMKLOM	DMKMCC
		DMKMCD	DMKMIA	DMKMON	DMKMSG	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKPEI
		DMKPEL	DMKPEN	DMKPEQ	DMKPER	DMKPET	DMKPGM	DMKPGS	DMKPMA	DMKPST	DMKPTR
		DMKPTT	DMKQCN	DMKQCO	DMKQCP	DMKQCQ	DMKRPA	DMKRSE	DMKRSF	DMKSEL	DMKSEP
		DMKSEV	DMKSFB	DMKSIX	DMKSNC	DMKSPL	DMKSPT	DMKSRM	DMKSSS	DMKSST	DMKSSU
		DMKSTP	DMKSTR	DMKSWM	DMKTAP	DMKTAQ	DMKTCS	DMKTCT	DMKTDK	DMKTHI	DMKTRA
		DMKTRC	DMKTRD	DMKTRK	DMKTRP	DMKTRU	DMKTTX	DMKTTY	DMKUDR	DMKUNT	DMKUSP
		DMKUSQ	DMKVAU	DMKVCA	DMKVCB	DMKVCH	DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCT
		DMKVCU	DMKVCV	DMKVCW	DMKVCX	DMKVDA	DMKVDB	DMKVDC	DMKVDD	DMKVDE	DMKVDG
		DMKVDH	DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVFC	DMKVFD	DMKVFE	DMKVFR	DMKVFS
		DMKVMC	DMKVMD	DMKVME	DMKVSE	DMKVSG	DMKVSU	DMKWRM	DMKXAB	DMKXAD	DMKXST
		DMKZTD									
SAVEWRK2	001566	DMKACO	DMKAPI	DMKATS	DMKBIO	DMKBLD	DMKCAC	DMKCAO	DMKCCH	DMKCCW	DMKCDB
		DMKCDM	DMKCDS	DMKCFC	DMKCFD	DMKCFF	DMKCFG	DMKCFH	DMKCFJ	DMKCFM	DMKCFO
		DMKCFQ	DMKCFR	DMKCFS	DMKCFT	DMKCFU	DMKCFV	DMKCFY	DMKCKS	DMKCKT	DMKCPB
		DMKCPM	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPU	DMKCPV	DMKCPW	DMKCPX
		DMKCPY	DMKCQC	DMKCQG	DMKCQH	DMKCQI	DMKCQP	DMKCQQ	DMKCQR	DMKCQT	DMKCQY
		DMKCSB	DMKCSF	DMKCSO	DMKCSP	DMKCSQ	DMKCSR	DMKCST	DMKCSU	DMKCSV	DMKCSW
		DMKCSX	DMKDAD	DMKDAS	DMKDAU	DMKDEF	DMKDEG	DMKDEI	DMKDEX	DMKDGD	DMKDIA
		DMKDIB	DMKDIF	DMKDRD	DMKDRE	DMKERM	DMKGRC	DMKHPS	DMKHPT	DMKHPU	DMKHVD
		DMKHVE	DMKHVF	DMKIDR	DMKIOE	DMKIOF	DMKIOG	DMKIOQ	DMKIOS	DMKIOT	DMKIUB
		DMKIUC	DMKIUE	DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS	DMKLNK	DMKLNM	DMKLOG
		DMKLOH	DMKMCI	DMKMID	DMKMON	DMKMSG	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKNLE
		DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPET	DMKPGM	DMKPGS	DMKPMA	DMKPST	DMKPTR
		DMKPTT	DMKQCN	DMKQCO	DMKQCQ	DMKQVM	DMKRPA	DMKRPD	DMKRSE	DMKRSF	DMKRSQ
		DMKRST	DMKSEL	DMKSEP	DMKSNC	DMKSPL	DMKSPM	DMKSPT	DMKSRM	DMKSSS	DMKSST
		DMKSSU	DMKSSV	DMKSTP	DMKSTR	DMKSVC	DMKSWA	DMKSWM	DMKTCS	DMKTCT	DMKTDK
		DMKTHI	DMKTRA	DMKTRC	DMKTRD	DMKTRK	DMKTRP	DMKTTX	DMKUDR	DMKUDU	DMKUNT
		DMKUSP	DMKVAU	DMKVBM	DMKVCA	DMKVCB	DMKVCH	DMKVCP	DMKVCQ	DMKVCR	DMKVCS
		DMKVCT	DMKVCU	DMKVCV	DMKVCW	DMKVDA	DMKVDC	DMKVDD	DMKVDE	DMKVDG	DMKVDH
		DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVFC	DMKVFE	DMKVFR	DMKVMC	DMKVMD	DMKVME
		DMKVSD	DMKVSE	DMKVSF	DMKVSG	DMKVSP	DMKVST	DMKWRM	DMKXAB	DMKXAD	DMKXST
		DMKZTD									
SAVEWRK3	000628	DMKACO	DMKAPI	DMKAPX	DMKATS	DMKBIO	DMKCAC	DMKCAO	DMKCDB	DMKCDM	DMKCDS
		DMKCFC	DMKCFD	DMKCFF	DMKCFR	DMKCFS	DMKCFT	DMKCFU	DMKCKR	DMKCKS	DMKCKV
		DMKCPB	DMKCPM	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPV	DMKCPW	DMKCPX
		DMKCPZ	DMKCQG	DMKCQP	DMKCQQ	DMKCQS	DMKCQT	DMKCQY	DMKCSO	DMKCST	DMKCSW
		DMKDAD	DMKDAS	DMKDAU	DMKDEF	DMKDEG	DMKDEI	DMKDEX	DMKDIA	DMKDIB	DMKDRD
		DMKDRE	DMKGRC	DMKHPS	DMKHPT	DMKHPU	DMKHVD	DMKHVE	DMKHVF	DMKIDR	DMKIOQ
		DMKIOT	DMKIUC	DMKIUE	DMKIUJ	DMKIUL	DMKLNK	DMKLOG	DMKLOH	DMKMCC	DMKMCD
		DMKMCI	DMKMIA	DMKMSG	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKPEI	DMKPEL
		DMKPEN	DMKPEQ	DMKPER	DMKPET	DMKPGM	DMKPGS	DMKPMA	DMKPTR	DMKQCN	DMKQCO
		DMKRPA	DMKRSE	DMKRSQ	DMKSEL	DMKSNC	DMKSSS	DMKSST	DMKSSU	DMKSSV	DMKSTR
		DMKSWA	DMKSWM	DMKTCS	DMKTCT	DMKTDK	DMKTHI	DMKTRC	DMKTRK	DMKTRP	DMKUNT
		DMKURS	DMKUSP	DMKVCB	DMKVCT	DMKVCV	DMKVCW	DMKVDA	DMKVDB	DMKVDC	DMKVDD

LABEL	COUNT	REFERENCES									
		DMKVDE	DMKVDG	DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVFC	DMKVFD	DMKVMD	DMKVME
		DMKVSG	DMKWRM	DMKXAB	DMKXAD	DMKXST	DMKZTD				
SAVEWRK4	000724	DMKACO	DMKATS	DMKBIO	DMKCAC	DMKCAO	DMKCCW	DMKCDB	DMKCDM	DMKCDS	DMKCFC
		DMKCFD	DMKCFG	DMKCFH	DMKCFO	DMKCFS	DMKCFT	DMKCFU	DMKCFV	DMKCFY	DMKCKR
		DMKCKS	DMKCKT	DMKCPB	DMKCPM	DMKCPN	DMKCPO	DMKCPP	DMKCPT	DMKCPU	DMKCPV
		DMKCPW	DMKCPY	DMKCQC	DMKCQG	DMKCQH	DMKCQI	DMKCQP	DMKCQQ	DMKCQY	DMKCSB
		DMKCSF	DMKCSO	DMKCSQ	DMKCST	DMKCSU	DMKCSV	DMKCSX	DMKDAD	DMKDAS	DMKDAU
		DMKDEF	DMKDEG	DMKDEI	DMKDGD	DMKDIA	DMKDIB	DMKDRD	DMKDRE	DMKERM	DMKHPT
		DMKHPU	DMKHVD	DMKHVF	DMKIDR	DMKIOQ	DMKIOS	DMKIUA	DMKIUC	DMKIUE	DMKIUL
		DMKIUP	DMKLNK	DMKLNM	DMKLOG	DMKLOH	DMKMCD	DMKMCI	DMKMSG	DMKNEA	DMKNES
		DMKNET	DMKNLD	DMKNLE	DMKPEI	DMKPEL	DMKPEQ	DMKPER	DMKPET	DMKPGM	DMKPGS
		DMKPST	DMKPTR	DMKQCO	DMKQCQ	DMKRSF	DMKRSQ	DMKSEP	DMKSNC	DMKSPT	DMKSRM
		DMKSSS	DMKSST	DMKSSU	DMKSTP	DMKSTR	DMKSWM	DMKTAP	DMKTAQ	DMKTCS	DMKTCT
		DMKTDK	DMKTHI	DMKTRC	DMKTRD	DMKTRK	DMKTRP	DMKUDR	DMKUSP	DMKUSQ	DMKVCB
		DMKVCH	DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCV	DMKVCW	DMKVCX	DMKVDE
		DMKVDG	DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVFC	DMKVFD	DMKVFE	DMKVMD	DMKVME
		DMKVSD	DMKWRM	DMKXAB	DMKXST	DMKZTD					
SAVEWRK5	000552	DMKATS	DMKCAC	DMKCAO	DMKCCH	DMKCDB	DMKCDM	DMKCDS	DMKCFC	DMKCFD	DMKCFF
		DMKCFG	DMKCFH	DMKCFO	DMKCFS	DMKCFU	DMKCFV	DMKCFY	DMKCKS	DMKCKT	DMKCKV
		DMKCPB	DMKCPM	DMKCPN	DMKCPS	DMKCPT	DMKCPW	DMKCQG	DMKCQQ	DMKCQY	DMKCSB
		DMKCSO	DMKCSP	DMKCSQ	DMKCST	DMKCSU	DMKCSV	DMKCSW	DMKCSX	DMKDAD	DMKDAS
		DMKDEG	DMKDEI	DMKDIA	DMKDIB	DMKDIF	DMKDRD	DMKERM	DMKHPU	DMKHVE	DMKHVF
		DMKIDR	DMKIUA	DMKIUB	DMKIUE	DMKIUL	DMKLNK	DMKLNM	DMKLOH	DMKLOJ	DMKLOM
		DMKMCC	DMKMID	DMKMSG	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKPEI	DMKPEL
		DMKPEQ	DMKPER	DMKPET	DMKPGM	DMKPGS	DMKPTR	DMKQCN	DMKQCO	DMKRSE	DMKSEP
		DMKSNC	DMKSSS	DMKSST	DMKSSU	DMKSTR	DMKSWA	DMKSWM	DMKTCS	DMKTCT	DMKTHI
		DMKTRC	DMKTRK	DMKTRP	DMKUNT	DMKURS	DMKUSQ	DMKVCH	DMKVCP	DMKVCQ	DMKVCR
		DMKVCS	DMKVCT	DMKVCU	DMKVCV	DMKVCW	DMKVCX	DMKVDA	DMKVDB	DMKVDC	DMKVDD
		DMKVDE	DMKVDG	DMKVDS	DMKVDT	DMKVER	DMKVFC	DMKVFD	DMKVFE	DMKVMC	DMKVMD
		DMKVME	DMKVSG	DMKXAB	DMKXST	DMKZTD					
SAVEWRK6	000523	DMKACO	DMKATS	DMKBIO	DMKCAC	DMKCAO	DMKCCH	DMKCDB	DMKCDM	DMKCFC	DMKCFF
		DMKCFG	DMKCFH	DMKCFM	DMKCFQ	DMKCFR	DMKCFS	DMKCKR	DMKCKS	DMKCKT	DMKCKV
		DMKCPM	DMKCPP	DMKCPS	DMKCPT	DMKCPY	DMKCQC	DMKCQG	DMKCQH	DMKCQI	DMKCQQ
		DMKCQT	DMKCQY	DMKCSB	DMKCSF	DMKCSO	DMKCSQ	DMKCST	DMKCSU	DMKCSV	DMKCSW
		DMKCSX	DMKDAD	DMKDAS	DMKDEF	DMKDEG	DMKDEI	DMKDEX	DMKDIA	DMKDIB	DMKDIF
		DMKDRD	DMKDRE	DMKEPS	DMKERM	DMKHPU	DMKIDR	DMKIOE	DMKIOQ	DMKIUE	DMKIUJ
		DMKIUL	DMKIUN	DMKLNK	DMKLNM	DMKLOG	DMKLOH	DMKMCD	DMKMCI	DMKMOO	DMKMSG
		DMKNEA	DMKNLD	DMKNLE	DMKPEQ	DMKPET	DMKPGM	DMKPGS	DMKPTR	DMKRSF	DMKSNC
		DMKSSS	DMKSST	DMKSSU	DMKSTP	DMKSWA	DMKSWM	DMKTCS	DMKTHI	DMKTRC	DMKTRD
		DMKTRK	DMKUDR	DMKUNT	DMKURS	DMKUSP	DMKUSQ	DMKVCA	DMKVCB	DMKVCH	DMKVCP
		DMKVCR	DMKVCS	DMKVCT	DMKVCU	DMKVCW	DMKVDA	DMKVDC	DMKVDD	DMKVDR	DMKVDS
		DMKVDT	DMKVER	DMKVFC	DMKVFE	DMKVMC	DMKVMD	DMKVME	DMKVSD	DMKVSG	DMKVSP
		DMKWRM	DMKXAB	DMKXST	DMKZTD						
SAVEWRK7	000445	DMKACO	DMKATS	DMKCAC	DMKCCH	DMKCCS	DMKCFD	DMKCFF	DMKCFG	DMKCFH	DMKCFO
		DMKCFQ	DMKCFS	DMKCFU	DMKCFV	DMKCFY	DMKCKR	DMKCKS	DMKCKT	DMKCKV	DMKCPN
		DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPY	DMKCQG	DMKCQP	DMKCQQ	DMKCQT	DMKCQY
		DMKCSB	DMKCSF	DMKCSO	DMKCSP	DMKCSQ	DMKCST	DMKCSW	DMKDAD	DMKDAS	DMKDEF
		DMKDEG	DMKDEI	DMKDIA	DMKDIB	DMKDIF	DMKDRD	DMKDRE	DMKERM	DMKHPU	DMKHVF
		DMKIDR	DMKIOE	DMKIOG	DMKIOQ	DMKIUC	DMKIUE	DMKIUJ	DMKIUL	DMKLNK	DMKLNM
		DMKLOH	DMKLOM	DMKMCI	DMKMID	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKPET
		DMKPGM	DMKPGS	DMKQCN	DMKSEP	DMKSND	DMKSPT	DMKSSS	DMKSST	DMKSSU	DMKSWM
		DMKTCS	DMKTRA	DMKTRC	DMKTRD	DMKTRK	DMKTRP	DMKUDR	DMKUSP	DMKUSQ	DMKVCP
		DMKVCR	DMKVCT	DMKVCU	DMKVCV	DMKVCW	DMKVDA	DMKVDC	DMKVDD	DMKVDS	DMKVDT
		DMKVER	DMKVFC	DMKVFD	DMKVMC	DMKVMD	DMKVME	DMKVSD	DMKVSE	DMKVSG	DMKXAB
		DMKXST									
SAVEWRK8	000696	DMKALG	DMKATS	DMKCAC	DMKCAO	DMKCCH	DMKCDB	DMKCDM	DMKCDS	DMKCFC	DMKCFF
		DMKCFG	DMKCFH	DMKCFO	DMKCFQ	DMKCFR	DMKCFS	DMKCFT	DMKCFU	DMKCFW	DMKCFY
		DMKCKR	DMKCKS	DMKCKV	DMKCPM	DMKCPN	DMKCPP	DMKCPS	DMKCPT	DMKCPU	DMKCPV

LABEL	COUNT	REFERENCES									
		DMKCPW	DMKCPY	DMKCPZ	DMKCQG	DMKCQH	DMKCQI	DMKCQP	DMKCQQ	DMKCQU	DMKCQY
		DMKCSB	DMKCSF	DMKCSO	DMKCSP	DMKCSQ	DMKCST	DMKCSU	DMKCSV	DMKCSW	DMKCSX
		DMKDAD	DMKDAS	DMKDEF	DMKDEG	DMKDEI	DMKDEX	DMKDIA	DMKDIB	DMKDIF	DMKDRD
		DMKERM	DMKHPS	DMKHPU	DMKHVF	DMKIDR	DMKIOQ	DMKIUC	DMKIUE	DMKIUJ	DMKIUL
		DMKIUN	DMKIUP	DMKLNK	DMKLNM	DMKLOG	DMKLOH	DMKLOJ	DMKLOM	DMKMHV	DMKMSG
		DMKNEA	DMKNES	DMKNLD	DMKNLE	DMKPEL	DMKPEQ	DMKPET	DMKPGM	DMKPMA	DMKPTR
		DMKSEL	DMKSEP	DMKSNC	DMKSND	DMKSPL	DMKSPT	DMKSSS	DMKSST	DMKSSU	DMKSTP
		DMKSWA	DMKSWM	DMKTCS	DMKTHI	DMKTRC	DMKTRD	DMKTRK	DMKTRP	DMKTTX	DMKUNT
		DMKURS	DMKUSQ	DMKVCH	DMKVCP	DMKVCR	DMKVCS	DMKVCT	DMKVCV	DMKVCW	DMKVCX
		DMKVDA	DMKVDC	DMKVDD	DMKVDS	DMKVDT	DMKVER	DMKVFE	DMKVFR	DMKVMD	DMKVME
		DMKVSD	DMKVSE	DMKVSG	DMKWRM	DMKXAB					
SAVEWRK9	000539	DMKATS	DMKBIO	DMKBLD	DMKCAC	DMKCAO	DMKCCH	DMKCCW	DMKCFF	DMKCFG	DMKCFH
		DMKCFQ	DMKCFR	DMKCFS	DMKCKS	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPU
		DMKCPW	DMKCPY	DMKCPZ	DMKCQP	DMKCQY	DMKCSP	DMKCSQ	DMKCST	DMKCSV	DMKDAD
		DMKDAS	DMKDAU	DMKDEF	DMKDEG	DMKDEX	DMKDGD	DMKDIA	DMKDIB	DMKDIF	DMKDRD
		DMKEIG	DMKERM	DMKHPS	DMKHPU	DMKHVF	DMKIDR	DMKIOQ	DMKIUC	DMKIUE	DMKIUJ
		DMKIUL	DMKIUN	DMKIUP	DMKIUS	DMKLNK	DMKLNM	DMKLOJ	DMKLOM	DMKNEA	DMKNES
		DMKNET	DMKNLD	DMKNLE	DMKPEQ	DMKPGM	DMKPGS	DMKPST	DMKPTR	DMKSEL	DMKSEP
		DMKSEV	DMKSIX	DMKSNC	DMKSPT	DMKSSS	DMKSST	DMKSSV	DMKSTP	DMKTCS	DMKTRC
		DMKTRD	DMKTTY	DMKUDR	DMKUNT	DMKVCB	DMKVCH	DMKVCP	DMKVCR	DMKVCS	DMKVCT
		DMKVCU	DMKVCV	DMKVCX	DMKVDA	DMKVDB	DMKVDC	DMKVDD	DMKVDE	DMKVDR	DMKVDS
		DMKVDT	DMKVER	DMKVMD	DMKVME	DMKVSG	DMKVSQ	DMKXAB	DMKXST	DMKZTD	
SAVFPRES	000003	DMKCFF	DMKCFH	DMKCKM							
SAVGREGS	000004	DMKCFF	DMKCFH	DMKCKM							
SAVKEYS	000004	DMKCFF	DMKCFH	DMKCKM							
SAVNAME	000004	DMKCFF	DMKCFH	DMKCKM	DMKCQY						
SAVPSW	000003	DMKCFF	DMKCFH	DMKCKM							
SAVREG14	000009	DMKCCW									
SAVTABLE	000009	DMKCFF	DMKCFH	DMKCFS	DMKCKM	DMKCQY					
SAVTIME	000008	DMKCFF	DMKCFH	DMKCFS	DMKCKM	DMKCQY					
SAVUSER	000004	DMKCFH	DMKCFS	DMKCKM							
SBA	000062	DMKGRD	DMKGRE	DMKGRG							
SCAN	000024	DMKDDR	DMKDID	DMKDIR	DMKFMT	DMKLOC	DMKSSP	DMKSSV	DMKVDC		
SCCBCTL	000002	DMKMHV									
SCCBIBLK	000003	DMKMHC	DMKPST								
SCCBIFE	000001	DMKMHC									
SCCBIMAP	000002	DMKPST	DMKXST								
SCCBINUM	000002	DMKPST	DMKXST								
SCCBISIZ	000001	DMKMHC									
SCCBKMAX	000002	DMKMHC	DMKPST								
SCCBLEN	000006	DMKMHV	DMKPST	DMKVFC	DMKXST						
SCCBLOK	000012	DMKMHC	DMKMHV	DMKPST	DMKVFC	DMKXST					
SCCBMDRT	000001	DMKMHV									
SCCBMODP	000001	DMKMHV									
SCCBMXID	000001	DMKMHC									
SCCBRESP	000015	DMKMHC	DMKMHV	DMKPST	DMKVFC	DMKXST					
SCCBRMAX	000002	DMKMHC	DMKMHV								
SCCB0010	000007	DMKMHC	DMKMHV	DMKPST	DMKXST						
SCCB0020	000003	DMKMHV	DMKVFC								
SCCB0100	000001	DMKMHV									
SCCB0120	000002	DMKVFC									
SCCB40F0	000001	DMKMHV									
SCHEDCL	000001	DMKMCC									
SCRCMDR	000011	DMKCFW	DMKGRD	DMKLOM	DMKQCN	DMKQCQ	DMKRGB				
SDRBLOK	000006	DMKIOE	DMKIOF	DMKIOJ							
SDRBSIZE	000001	DMKIOE									
SDRCPID	000002	DMKIOE	DMKIOF								
SDRCTRS	000016	DMKIOE	DMKIOF	DMKIOJ							

LABEL	COUNT	REFERENCES									
SDRCTR8	000003	DMKIOF									
SDRCTR9	000001	DMKIOF									
SDRCUA	000003	DMKIOE	DMKIOF								
SDRFLAGS	000009	DMKIOE	DMKIOJ								
SDRFLCT	000004	DMKIOF									
SDRLNGTH	000009	DMKIOE	DMKIOF	DMKIOJ							
SDRMAX	000003	DMKIOF									
SDROVFWK	000006	DMKIOF									
SDRPRMCT	000004	DMKIOF									
SDRRDEV	000002	DMKIOF									
SDRRECD	000005	DMKIOE	DMKIOF	DMKIOJ							
SDRSHRT	000009	DMKIOE	DMKIOF	DMKIOJ							
SDRSIZE	000001	DMKIOF									
SDRSIZE1	000001	DMKIOF									
SECUSER	000006	DMKGRD	DMKQCN	DMKQCO	DMKQCQ						
SEEKOFF	000002	DMKCCD									
SEGENQ	000006	DMKBLD	DMKSTR								
SEGFLAG	000013	DMKATS	DMKPGM	DMKPGS	DMKPSA	DMKSEL	DMKSTR	DMKSWM			
SEGINV	000064	DMKATS	DMKBLD	DMKCFF	DMKCFG	DMKCFH	DMKCKM	DMKCKW	DMKDAD	DMKDAS	DMKDAU
		DMKHVD	DMKPGS	DMKPMA	DMKPSA	DMKPTR	DMKPTS	DMKPTT	DMKSEL	DMKSTR	DMKSWA
		DMKVFR	DMKVMA								
SEGMIG	000007	DMKCKM	DMKPGM	DMKSTR	DMKSWM						
SEGMIGPG	000002	DMKPGM	DMKSTR								
SEGMIGPP	000002	DMKPGM	DMKSTR								
SEGPAGE	000146	DMKATS	DMKBLD	DMKCFF	DMKCFG	DMKCKM	DMKCKP	DMKDAD	DMKDAS	DMKDAU	DMKHVD
		DMKPGM	DMKPGS	DMKPMA	DMKPTR	DMKPTS	DMKPTT	DMKSEG	DMKSEL	DMKSTR	DMKSWA
		DMKSWM	DMKVMA								
SEGPLEN	000007	DMKBLD	DMKSTR	DMKVMA							
SEGPROT	000022	DMKATS	DMKCFF	DMKPGS	DMKPRW	DMKPSA	DMKVAT	DMKVAU			
SEGTABLE	000024	DMKATS	DMKBLD	DMKCFG	DMKCKM	DMKCKP	DMKPGM	DMKPGS	DMKPSA	DMKVMA	
SEG1M	000002	DMKPRV	DMKPTT								
SELDATA	000005	DMKCQC	DMKTRP	DMKTRR	DMKTRT						
SELDISP	000014	DMKCQC	DMKTRP	DMKTRR	DMKTRT						
SELECT	000030	DMKCQC	DMKGRD	DMKTRP	DMKTRR	DMKTRT					
SELENTRY	000012	DMKCQC	DMKTRP	DMKTRR	DMKTRT						
SELFORW	000010	DMKCQC	DMKTRP	DMKTRR	DMKTRT						
SELLTH	000005	DMKCQC	DMKTRP	DMKTRR	DMKTRT						
SELRB	000002	DMKGRD									
SELRBP	000002	DMKGRE									
SELRM	000001	DMKGRD									
SELRMP	000002	DMKGRE									
SELSIZE	000004	DMKTRP	DMKTRR								
SELWRT	000002	DMKGRD									
SENDQMSG	000006	DMKIUE	DMKIUG	DMKIUN	DMKIUS						
SENSE	000123	DMKCCD	DMKCCO	DMKCCT	DMKCCW	DMKCPP	DMKDDR	DMKDIR	DMKDMP	DMKDMQ	DMKEXT
		DMKFMT	DMKOPR	DMKOVR	DMKSAD	DMKSSP	DMKVMI	DMKVRR	DMKVRS		
SENSEA4	000001	DMKCCO									
SENSEE4	000005	DMKCCO	DMKCCS								
SENSEID	000004	DMKGRF									
SENSEIO	000001	DMKCCT									
SENSEPID	000004	DMKCCO	DMKCCS								
SENSE04	000002	DMKCCO	DMKCCS								
SENSE44	000001	DMKCCO									
SEQMSK	000007	DMKFRE									
SETID	000002	DMKCCO									
SETTDK	000002	DMKACO	DMKCKF								
SETUP	000075	DMKACO	DMKCKF								
SETUP1	000002	DMKACO	DMKCKF								

LABEL	COUNT	REFERENCES									
SETUP2	000004	DMKACO	DMKCKF								
SEVER	000055	DMKAPS	DMKAPT	DMKAPV	DMKBIO	DMKCRM	DMKIDR	DMKIOF	DMKIUJ	DMKIUP	DMKMSG
		DMKRPI	DMKVCW	DMKVCX	DMKVMG						
SF	000059	DMKBOX	DMKGRD	DMKQCQ							
SFBACNT	000002	DMKACO	DMKCQI								
SFBBCONV	000016	DMKAPS	DMKAPU	DMKAPV	DMKCQH	DMKCQI	DMKCSV	DMKCSW	DMKCSY	DMKSPS	DMKWRM
SFBCDMP	000003	DMKIDU	DMKWRN								
SFBCKPMP	000002	DMKCKS	DMKIDU								
SFBCLAS	000039	DMKACO	DMKAPU	DMKAPV	DMKCKF	DMKCQH	DMKCQI	DMKCSQ	DMKCSR	DMKCSV	DMKCSW
		DMKCSX	DMKCSY	DMKDRD	DMKHVF	DMKMIA	DMKNLE	DMKRSP	DMKRST	DMKSEP	DMKSPL
		DMKSPS	DMKSPT	DMKTRT	DMKTRU	DMKURS	DMKVME	DMKVSP	DMKVSQ	DMKVSW	
SFBCONTO	000003	DMKCKF	DMKWRN								
SFBCONV	000016	DMKAPU	DMKAPV	DMKCQI	DMKCSV	DMKCSW	DMKCSY	DMKRSP	DMKSPS	DMKWRM	DMKXAB
SFBCOPY	000040	DMKACO	DMKAPU	DMKAPV	DMKCKF	DMKCKR	DMKCQI	DMKCSF	DMKCSW	DMKDRD	DMKHVF
		DMKMIA	DMKNLE	DMKRSP	DMKSPL	DMKTCS	DMKTRT	DMKTRU	DMKURS	DMKVME	
SFBDATE	000023	DMKACO	DMKCKH	DMKCKV	DMKCQI	DMKDMP	DMKMIA	DMKNLE	DMKSAD	DMKSPL	DMKTRT
		DMKTRU	DMKVMD	DMKVME	DMKWRM						
SFBDEST	000031	DMKAPU	DMKAPV	DMKCKF	DMKCQH	DMKCQI	DMKCSQ	DMKCSV	DMKCSW	DMKCSY	DMKHVF
		DMKRSP	DMKSEP	DMKSPL	DMKSPS	DMKSPT					
SFBDIST	000045	DMKACO	DMKCKF	DMKCKR	DMKCQI	DMKCSQ	DMKCSW	DMKDMP	DMKHVF	DMKMIA	DMKNLE
		DMKSAD	DMKSEP	DMKSPK	DMKSPL	DMKTRT	DMKTRU	DMKURS	DMKVME		
SFBDUMP	000018	DMKCKV	DMKCQI	DMKCSY	DMKDMP	DMKDRD	DMKNLE	DMKSPK	DMKSPS	DMKTRR	DMKTRT
		DMKTRU	DMKVME	DMKVSW							
SFBEOF	000015	DMKCKR	DMKDRD	DMKVSW	DMKVSX	DMKWRM					
SFBFCB	000009	DMKRSP	DMKVSP	DMKVSQ							
SFBFCBNL	000004	DMKRSP	DMKVSQ								
SFBFCBXL	000004	DMKRSP	DMKVSQ								
SFBFILID	000103	DMKAPT	DMKAPU	DMKAPV	DMKAPW	DMKCFU	DMKCKS	DMKCKT	DMKCKV	DMKCQI	DMKCSQ
		DMKCSR	DMKCST	DMKCSV	DMKCSW	DMKCSY	DMKDRD	DMKDRE	DMKHVF	DMKMIA	DMKMNJ
		DMKNLE	DMKRSE	DMKRST	DMKSEP	DMKSFB	DMKSPK	DMKSPL	DMKSPS	DMKTCS	DMKTCT
		DMKTRT	DMKTRU	DMKURS	DMKVME	DMKVSD	DMKVSF	DMKVSG	DMKVSP	DMKVST	DMKVSW
		DMKWRN	DMKXAB								
SFBFIRST	000002	DMKCKF	DMKSPK								
SFBFLAG	000198	DMKAPS	DMKAPU	DMKAPV	DMKCKH	DMKCKR	DMKCKV	DMKCQH	DMKCQI	DMKCQR	DMKCSF
		DMKCSQ	DMKCSR	DMKCST	DMKCSV	DMKCSW	DMKCSX	DMKCSY	DMKDRD	DMKDRE	DMKHVF
		DMKMIA	DMKNLE	DMKRSE	DMKRSP	DMKRSQ	DMKSFB	DMKSPK	DMKSPL	DMKSPS	DMKSPT
		DMKTRR	DMKTRT	DMKTRU	DMKUSP	DMKVME	DMKVSP	DMKVSU	DMKVSW	DMKVSX	DMKWRM
		DMKXAB									
SFBFLAG2	000054	DMKCKF	DMKCKH	DMKCKR	DMKCKV	DMKCQI	DMKCSO	DMKCSQ	DMKCSY	DMKDRD	DMKMIA
		DMKRSP	DMKSEP	DMKSPK	DMKSPL	DMKVSP	DMKVSQ	DMKVSW	DMKWRM		
SFBFLAG3	000045	DMKACO	DMKAPU	DMKCQI	DMKCSF	DMKCSY	DMKDRD	DMKDRE	DMKHVF	DMKRSP	DMKVSP
		DMKVSQ	DMKVSX								
SFBFLAG4	000108	DMKAPS	DMKAPU	DMKAPV	DMKCQH	DMKCQI	DMKCST	DMKCSV	DMKCSW	DMKCSY	DMKDRD
		DMKDRE	DMKHVF	DMKMIA	DMKRSP	DMKSPK	DMKSPS	DMKSPT	DMKVMD	DMKVSD	DMKVSE
		DMKVSQ	DMKWRM	DMKWRN	DMKXAB	DMKXAD					
SFBFLAG5	000006	DMKCKF	DMKIDU	DMKWRN							
SFBFLASH	000011	DMKAPU	DMKCQI	DMKCSW	DMKHVF	DMKRSP	DMKSPK	DMKSPL			
SFBFLNMT	000003	DMKVSP	DMKVSQ								
SFBFNAME	000025	DMKACO	DMKCKH	DMKCQI	DMKCSQ	DMKCSW	DMKHVF	DMKMIA	DMKNLE	DMKRSP	DMKRST
		DMKSPL	DMKSPS	DMKTRT	DMKTRU	DMKVMD	DMKWRM				
SFBFTYPE	000016	DMKCKH	DMKCQI	DMKM!A	DMKNLE	DMKRST	DMKTRT	DMKTRU	DMKVMD	DMKVME	DMKWRM
SFBHOLD	000011	DMKCSQ	DMKSPL	DMKVSP	DMKVSW						
SFBINUSE	000073	DMKAPS	DMKAPU	DMKAPV	DMKCKR	DMKCQH	DMKCQI	DMKCSQ	DMKCSR	DMKCST	DMKCSV
		DMKCSW	DMKCSY	DMKDRD	DMKDRE	DMKMIA	DMKRSP	DMKSFB	DMKSPS	DMKSPT	DMKUSP
		DMKVSW	DMKWRM	DMKXAB							
SFBINVS	000014	DMKCSV	DMKCSY	DMKDRD	DMKDRE	DMKVSD	DMKVSE	DMKWRN			
SFBLAST	000096	DMKACO	DMKCFU	DMKCKF	DMKCKH	DMKCKV	DMKCSV	DMKCSW	DMKCSY	DMKDMP	DMKDRD
		DMKDRE	DMKMIA	DMKNLE	DMKRSP	DMKRST	DMKSPK	DMKSPL	DMKSPS	DMKTRT	DMKVME

LABEL	COUNT	REFERENCES									
		DMKVSP	DMKVSQ	DMKVSX							
SFBLDBEG	000018	DMKAPU	DMKCQI	DMKCSF	DMKHVF	DMKRSP	DMKVSP	DMKVSX			
SFBLDMID	000017	DMKAPU	DMKCQI	DMKCSF	DMKHVF	DMKRSP	DMKVSP	DMKVSX			
SFBLOK	000317	DMKACO	DMKAPS	DMKAPT	DMKAPU	DMKAPV	DMKAPW	DMKCFU	DMKCKF	DMKCKH	DMKCKR
		DMKCKS	DMKCKT	DMKCKV	DMKCQH	DMKCQI	DMKCQR	DMKCSF	DMKCSO	DMKCSQ	DMKCSR
		DMKCST	DMKCSU	DMKCSV	DMKCSW	DMKCSX	DMKCSY	DMKDMP	DMKDRD	DMKDRE	DMKHVF
		DMKIDU	DMKMIA	DMKMNJ	DMKNLE	DMKRSE	DMKRSP	DMKRSQ	DMKRST	DMKSAD	DMKSEP
		DMKSFB	DMKSPK	DMKSPL	DMKSPS	DMKSPT	DMKTCS	DMKTCT	DMKTRP	DMKTRR	DMKTRT
		DMKTRU	DMKURS	DMKUSP	DMKVMD	DMKVME	DMKVSD	DMKVSE	DMKVSF	DMKVSG	DMKVSP
		DMKVSQ	DMKVST	DMKVSU	DMKVSW	DMKVSX	DMKWRM	DMKWRN	DMKXAB	DMKXAD	
SFBMISC1	000014	DMKCKV	DMKDRD	DMKSPK	DMKSPS	DMKTRR	DMKTRT	DMKTRU	DMKVME	DMKVSQ	
SFBMON	000010	DMKCKR	DMKCKV	DMKCQI	DMKCSY	DMKDRD	DMKMIA	DMKSPK	DMKVSW		
SFBNOHLD	000014	DMKCQH	DMKCQI	DMKCSQ	DMKSPL	DMKVSW					
SFBNORET	000008	DMKCSY	DMKVMD								
SFBOFORM	000037	DMKACO	DMKAPU	DMKAPV	DMKCKF	DMKCKH	DMKCKR	DMKCQH	DMKCQI	DMKCSV	DMKCSW
		DMKCSY	DMKMIA	DMKNLE	DMKRSP	DMKSEP	DMKSPL	DMKSPS	DMKSPT	DMKTRT	DMKTRU
		DMKVMD	DMKWRM								
SFBOPEN	000007	DMKCKR	DMKDRD	DMKVSW	DMKVSX	DMKWRM					
SFBORIG	000029	DMKACO	DMKCFU	DMKCKF	DMKCKS	DMKCQI	DMKCSY	DMKDRD	DMKDRE	DMKIDU	DMKMIA
		DMKNLE	DMKRST	DMKSAD	DMKSEP	DMKSPL	DMKTRT	DMKTRU	DMKVME	DMKVSF	DMKWRN
SFBPNT	000199	DMKAPS	DMKAPT	DMKAPU	DMKAPV	DMKCKF	DMKCKH	DMKCKR	DMKCKV	DMKCQH	DMKCQI
		DMKCST	DMKCSV	DMKCSW	DMKCSY	DMKDMP	DMKDRD	DMKDRE	DMKMIA	DMKRSE	DMKRSP
		DMKRSQ	DMKRST	DMKSFB	DMKSPK	DMKSPL	DMKSPS	DMKURS	DMKVME	DMKVSD	DMKVSE
		DMKVSF	DMKVSG	DMKVSP	DMKVSQ	DMKWRM	DMKXAB				
SFBPURGD	000014	DMKAPU	DMKCQH	DMKCQI	DMKCSV	DMKCSW	DMKCSY	DMKWRM	DMKXAB		
SFBPURGE	000009	DMKCKF	DMKCKH	DMKSPK	DMKVSP	DMKVSQ	DMKWRM				
SFBRECER	000018	DMKCKH	DMKCKR	DMKCSF	DMKRSE	DMKRSP	DMKRSQ	DMKSPK	DMKVSP	DMKVSU	DMKVSW
		DMKVSX	DMKWRM								
SFBRECNO	000051	DMKACO	DMKCKF	DMKCKV	DMKCQI	DMKHVF	DMKMIA	DMKNLE	DMKRSP	DMKRST	DMKSAD
		DMKSEP	DMKSPK	DMKSPS	DMKTRP	DMKTRT	DMKURS	DMKVME	DMKVSP	DMKVSQ	DMKVST
		DMKVSW									
SFBRECOK	000004	DMKCSF	DMKRSP	DMKRSQ							
SFBRECS	000035	DMKCKF	DMKCKH	DMKCKR	DMKCKS	DMKRSP	DMKRSQ	DMKSPK	DMKSPS	DMKVSX	DMKWRM
SFBRECSZ	000015	DMKACO	DMKNLE	DMKSPL	DMKTRT	DMKTRU	DMKVME	DMKVSP	DMKVST	DMKVSX	
SFBREQUE	000008	DMKCSO	DMKRSP	DMKSPL							
SFBRSTRT	000009	DMKCKR	DMKRSP	DMKSEP	DMKSPL	DMKWRM					
SFBSEEN	000009	DMKCSY	DMKDRD	DMKDRE							
SFBSHOLD	000053	DMKAPU	DMKAPV	DMKCQH	DMKCQI	DMKCQR	DMKCSF	DMKCSQ	DMKCSW	DMKCSY	DMKHVF
		DMKRSE	DMKRSP	DMKSPL	DMKSPS	DMKSPT					
SFBSIZE	000111	DMKACO	DMKAPT	DMKCKF	DMKCKH	DMKCKR	DMKCKS	DMKCKT	DMKCPI	DMKCSQ	DMKCSV
		DMKCSW	DMKCSY	DMKDMP	DMKDRD	DMKDRE	DMKIDU	DMKMIA	DMKNLE	DMKRSP	DMKRST
		DMKSPK	DMKSPL	DMKSPS	DMKTRT	DMKTRU	DMKUSP	DMKVMD	DMKVME	DMKVSD	DMKVSP
		DMKVSW	DMKWRM	DMKWRN							
SFBSIZEB	000027	DMKCKR	DMKCKT	DMKCKV	DMKCSY	DMKSFB	DMKVSE	DMKVSF	DMKWRM	DMKWRN	
SFBSP3	000001	DMKWRM									
SFBSTART	000140	DMKACO	DMKAPT	DMKAPU	DMKCFU	DMKCKF	DMKCKH	DMKCKV	DMKCQI	DMKCST	DMKCSV
		DMKCSW	DMKCSY	DMKDMP	DMKDRD	DMKDRE	DMKHVF	DMKIDU	DMKMIA	DMKNLE	DMKRSP
		DMKRST	DMKSEP	DMKSPK	DMKSPL	DMKSPS	DMKTCS	DMKTRR	DMKTRT	DMKTRU	DMKVME
		DMKVSP	DMKVSQ	DMKVST	DMKVSW						
SFBSTCPY	000004	DMKSPL	DMKTCS								
SFBSYSID	000089	DMKACO	DMKCFU	DMKCKF	DMKCKH	DMKCKR	DMKCKS	DMKCKT	DMKCQH	DMKCQI	DMKCSR
		DMKCSV	DMKCSW	DMKCSY	DMKDRD	DMKDRE	DMKIDU	DMKMIA	DMKNLE	DMKRST	DMKSFB
		DMKSPK	DMKSPL	DMKSPS	DMKSPT	DMKTRU	DMKVME	DMKVSD	DMKVSE	DMKVSP	DMKVSQ
		DMKVSW	DMKWRM	DMKWRN							
SFBTICER	000004	DMKRSP									
SFBTIME	000033	DMKACO	DMKCKV	DMKCQI	DMKDMP	DMKMIA	DMKNLE	DMKSAD	DMKSPL	DMKTRT	DMKTRU
		DMKVMD	DMKVME	DMKVSP	DMKVSQ						
SFBTUSE	000041	DMKCKR	DMKCQI	DMKCST	DMKCSW	DMKCSY	DMKDRD	DMKDRE	DMKHVF	DMKMIA	DMKRSP

LABEL	COUNT	REFERENCES									
		DMKSPS	DMKSPT	DMKWRM	DMKXAB						
SFBTYPE	000044	DMKACO	DMKAPU	DMKAPV	DMKCKT	DMKCQI	DMKCSY	DMKDMP	DMKDRD	DMKHVF	DMKMIA
		DMKNLE	DMKRSP	DMKSPK	DMKSPL	DMKTRT	DMKTRU	DMKURS	DMKVME	DMKVSG	DMKVSP
		DMKVSQ	DMKVSX	DMKXAB							
SFBUFORM	000035	DMKACO	DMKAPV	DMKCKF	DMKCKR	DMKCQH	DMKCQI	DMKCSQ	DMKCSV	DMKCSW	DMKCSY
		DMKHVF	DMKMIA	DMKNLE	DMKSPL	DMKTRT	DMKTRU	DMKVMD			
SFBUHOLD	000052	DMKAPU	DMKAPV	DMKCKR	DMKCQH	DMKCQI	DMKCSR	DMKCSW	DMKCSX	DMKCSY	DMKDRD
		DMKHVF	DMKRSP	DMKSPL	DMKSPS	DMKSPT	DMKVSW	DMKXAB			
SFBUSER	000153	DMKACO	DMKAPT	DMKCFU	DMKCKF	DMKCKT	DMKCKV	DMKCQH	DMKCQI	DMKCSQ	DMKCST
		DMKCSV	DMKCSW	DMKCSY	DMKDRD	DMKDRE	DMKHVF	DMKIDU	DMKMIA	DMKNLE	DMKRSP
		DMKRST	DMKSAD	DMKSEP	DMKSFB	DMKSPK	DMKSPL	DMKSPS	DMKTCS	DMKTCT	DMKTRT
		DMKTRU	DMKURS	DMKVMD	DMKVME	DMKVSD	DMKVSE	DMKVSG	DMKVSW	DMKWRN	DMKXAB
SFBVLEN	000001	DMKVSQ									
SFBXAB	000021	DMKCSV	DMKCSW	DMKCSY	DMKSPK	DMKSPS	DMKWRM	DMKXAB	DMKXAD		
SFBXABER	000005	DMKSPK	DMKWRM	DMKXAD							
SFBXABL	000014	DMKCSV	DMKCSW	DMKCSY	DMKSPK	DMKXAB	DMKXAD				
SFBXFER	000001	DMKCSY									
SFLAG	000009	DMKXAB									
SFPBLOK	000002	DMKSEP	DMKVSQ								
SFPBOT	000002	DMKVSQ									
SFPCLASS	000002	DMKSEP	DMKVSQ								
SFPEND	000002	DMKSEP	DMKVSQ								
SFPLEN	000005	DMKSEP	DMKVSQ								
SFPOPTS	000003	DMKVSQ									
SFPTITLE	000002	DMKSEP	DMKVSQ								
SFPTOP	000002	DMKVSQ									
SHPSLIM	000003	DMKCCD	DMKCCO								
SHQBLOK	000014	DMKCKS	DMKCQR	DMKCSQ	DMKCSY	DMKSPL	DMKWRM	DMKWRN			
SHQBSIZE	000011	DMKCKF	DMKCKS	DMKCKV	DMKCSQ	DMKWRM					
SHQCKPMP	000001	DMKCSQ									
SHQCKPT	000004	DMKCKS	DMKCSQ	DMKWRN							
SHQFLAGS	000001	DMKCSQ									
SHQPNT	000003	DMKCSQ	DMKCSY	DMKSPL							
SHQSHOLD	000007	DMKCQR	DMKCSQ	DMKCSY	DMKSPL						
SHQUSER	000006	DMKCQR	DMKCSQ	DMKCSY	DMKSPL						
SHRBPNT	000007	DMKATS	DMKCFF	DMKCFH	DMKPGS						
SHRFLAG	000035	DMKATS	DMKCCW	DMKCFF	DMKCPP	DMKCPU	DMKPGS	DMKPRV	DMKPRW	DMKPSA	DMKPTR
		DMKPTS	DMKPTT	DMKSEL							
SHRFPNT	000022	DMKATS	DMKCFF	DMKCFH	DMKCPP	DMKCPU	DMKPGS				
SHRLKCNT	000036	DMKAPI	DMKCCW	DMKCPM	DMKCPO	DMKCPS	DMKCPU	DMKDGD			
SHRNAME	000022	DMKATS	DMKCFF	DMKCFG	DMKCFH	DMKCPP	DMKPGS	DMKVMA			
SHRNOPRT	000031	DMKATS	DMKCCW	DMKCFF	DMKCPP	DMKCPU	DMKPGS	DMKPRV	DMKPRW	DMKPSA	DMKPTR
		DMKPTS	DMKPTT	DMKSEL							
SHRPAGE	000025	DMKATS	DMKCFF	DMKCPP	DMKCPU						
SHRSEGCT	000019	DMKATS	DMKCFF	DMKCPP	DMKCPU	DMKPGS	DMKPTR	DMKPTS	DMKVMA		
SHRSEGNM	000019	DMKATS	DMKCFF	DMKCPP	DMKCPU	DMKPGS	DMKVMA				
SHRSGPRT	000026	DMKATS	DMKCFF	DMKCPP	DMKCPU	DMKPGS	DMKPTR	DMKPTS	DMKPTT	DMKSEL	
SHRTABLE	000052	DMKATS	DMKCCW	DMKCFF	DMKCFG	DMKCFH	DMKCPP	DMKCPU	DMKPGS	DMKPRV	DMKPRW
		DMKPSA	DMKPTR	DMKPTS	DMKPTT	DMKSEL	DMKVMA				
SHRTSIZE	000003	DMKATS	DMKCFF	DMKPGS							
SHRUSECT	000009	DMKATS	DMKCFF	DMKPGS							
SIGAPR	000001	DMKMCT									
SIGCLK	000001	DMKCLK									
SIGCR	000002	DMKCFP	DMKDMP								
SIGDISP	000004	DMKDSP	DMKEXT	DMKFRE	DMKSTK						
SIGEMS	000016	DMKACO	DMKCLK	DMKCPO	DMKCPS	DMKEXT	DMKFRE	DMKMPO	DMKPMA	DMKPTR	DMKPTT
		DMKSEL									
SIGEXT	000002	DMKEXT	DMKPTR								

LABEL	COUNT	REFERENCES									
SIGIPR	000006	DMKCKP	DMKCPI	DMKCPP	DMKCPU	DMKDMP					
SIGMASK	000001	DMKDSP									
SIGPR	000001	DMKACS									
SIGQUI	000009	DMKACO	DMKCLK	DMKCPO	DMKCPS	DMKEXT	DMKPTR	DMKPTT	DMKSEL		
SIGRES	000010	DMKACO	DMKCLK	DMKCPP	DMKEXT	DMKPTR	DMKPTT	DMKSEL			
SIGREST	000013	DMKAPI	DMKCCH	DMKCKD	DMKDMP	DMKDMQ	DMKMCT	DMKMPO	DMKVRS		
SIGSAVE	000010	DMKCKW	DMKEXT								
SIGSENSE	000021	DMKCFP	DMKCKD	DMKCKP	DMKCPI	DMKCPP	DMKCPS	DMKCPU	DMKDMP	DMKDMQ	DMKEXT
		DMKLOJ	DMKMHC	DMKMPO	DMKVRR						
SIGSHD	000001	DMKCPS									
SIGSSS	000011	DMKCCH	DMKCKD	DMKCPI	DMKDMP	DMKMCT	DMKMPO				
SIGSTART	000007	DMKCPP	DMKLOJ	DMKMCH	DMKMHC	DMKMPO	DMKVRR				
SIGSTOP	000010	DMKCCH	DMKCPO	DMKCPS	DMKDMP	DMKMCH	DMKMCT	DMKMHC	DMKMPO		
SIGSYNC	000001	DMKCLK									
SIGWAKE	000010	DMKCPI	DMKCPU	DMKEXT	DMKFRE	DMKIOS	DMKMOO	DMKSCH			
SIGXC	000026	DMKACO	DMKCLK	DMKCPI	DMKCPP	DMKCPU	DMKDSP	DMKEXT	DMKFRE	DMKIOS	DMKMCT
		DMKMOO	DMKMPO	DMKPMA	DMKPTR	DMKPTT	DMKSCH	DMKSEL	DMKSTK		
SIGXEX	000001	DMKFRE									
SILI	002279	DMKACO	DMKACS	DMKBSC	DMKCAC	DMKCCD	DMKCCT	DMKCCW	DMKCFQ	DMKCKD	DMKCKH
		DMKCKN	DMKCKP	DMKCNS	DMKCPB	DMKCPI	DMKCPM	DMKCPN	DMKCPS	DMKCPW	DMKCPZ
		DMKCQT	DMKCSB	DMKCSC	DMKDAD	DMKDAS	DMKDAU	DMKDDR	DMKDEX	DMKDGD	DMKDGF
		DMKDIB	DMKDIR	DMKDMP	DMKDMQ	DMKDSB	DMKFMT	DMKGRD	DMKGRE	DMKGRF	DMKGRG
		DMKGRH	DMKHPS	DMKHPU	DMKIOS	DMKIOT	DMKLDOOE	DMKMCC	DMKMNL	DMKMNT	DMKNLD
		DMKNLE	DMKOPE	DMKOPR	DMKOVR	DMKPAG	DMKPAH	DMKPIA	DMKPIB	DMKRGA	DMKRGB
		DMKRGC	DMKRGD	DMKRNH	DMKRSF	DMKRSP	DMKRSQ	DMKRST	DMKSAD	DMKSAV	DMKSEP
		DMKSPK	DMKSPS	DMKSPT	DMKSSP	DMKTAP	DMKTAQ	DMKTCS	DMKTCT	DMKTPE	DMKTRK
		DMKTTX	DMKTTY	DMKTTZ	DMKUCB	DMKUCC	DMKUCS	DMKUDR	DMKVCA	DMKVCB	DMKVCN
		DMKVDE	DMKVDR	DMKVDT	DMKVMI	DMKVRR	DMKVRS	DMKVSP	DMKVSQ	DMKVST	DMKVSX
		DMKZTD									
SIOCCH	000003	DMKACS	DMKCCH								
SIZE	000022	DMKFRE									
SKALTCYL	000003	DMKCCD									
SKCECYL	000007	DMKCCD									
SKIP	000349	DMKBSC	DMKCCD	DMKCCF	DMKCCO	DMKCCS	DMKCCT	DMKCCW	DMKCKH	DMKCKP	DMKCNS
		DMKCSB	DMKCSC	DMKCST	DMKDAD	DMKDAS	DMKDAU	DMKDDR	DMKDIB	DMKDIR	DMKDMP
		DMKDRD	DMKDRE	DMKFMT	DMKHPS	DMKHPU	DMKIOS	DMKIOT	DMKPAG	DMKPAH	DMKRGA
		DMKRGE	DMKRSF	DMKRSP	DMKRSQ	DMKRST	DMKSAV	DMKSEP	DMKSPL	DMKTAQ	DMKTCS
		DMKTCT	DMKTRK	DMKTTX	DMKTTY	DMKTTZ	DMKUNT	DMKVCA	DMKVCN	DMKVDE	DMKVMI
		DMKVSC	DMKVSP	DMKVSQ	DMKVST	DMKVSX					
SKIPADD	000003	DMKACO	DMKCKF								
SKIPSENS	000002	DMKCCS	DMKCCW								
SKIPSUP	000005	DMKDMQ									
SLNONPOS	000002	DMKSCH									
SM	000056	DMKACS	DMKCNS	DMKCPM	DMKDDR	DMKDMP	DMKDMQ	DMKDSP	DMKEXT	DMKFMT	DMKIOS
		DMKIOT	DMKMNT	DMKNLD	DMKNLE	DMKOPR	DMKRSE	DMKSAD	DMKSAV	DMKSSP	DMKVIO
		DMKVMI	DMKVSI	DMKVSJ							
SMCOM	000018	DMKCCD	DMKCCO	DMKCCT	DMKCCW						
SNACARD	000002	DMKACO	DMKCKF								
SNAENBLE	000003	DMKVCT	DMKVCW								
SNARACO	000003	DMKVCT	DMKVCW								
SNARBACH	000014	DMKDSP	DMKVCP	DMKVCR	DMKVCS	DMKVCT	DMKVCU				
SNARBLOK	000109	DMKACO	DMKBLD	DMKCFM	DMKCFT	DMKCKF	DMKCQG	DMKCQQ	DMKCQS	DMKCQT	DMKCQU
		DMKCQY	DMKDIA	DMKDIB	DMKDIF	DMKDSP	DMKJRL	DMKLOG	DMKLOH	DMKLOM	DMKQCO
		DMKQCP	DMKQCQ	DMKTRD	DMKUSQ	DMKVCN	DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCT
		DMKVCU	DMKVCV	DMKVCW	DMKVCX	DMKVDS					
SNARBTCT	000003	DMKVCS									
SNARCNTX	000005	DMKVCV	DMKVCX								
SNARCONQ	000015	DMKVCR	DMKVCS	DMKVCV							

LABEL	COUNT	REFERENCES									
SNARCPFD	000006	DMKVCP	DMKVCU								
SNARCPT	000009	DMKVCP	DMKVCT	DMKVCV	DMKVCW	DMKVCX					
SNARDIAL	000040	DMKCQS	DMKCQT	DMKDIF	DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCU	DMKVCV
		DMKVCW	DMKVCX								
SNARDIPG	000008	DMKDIA	DMKDIF	DMKVCS							
SNARDIS	000020	DMKDIB	DMKLOG	DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCV	DMKVCW	DMKVCX
SNAREXWT	000007	DMKVCQ	DMKVCS	DMKVCX							
SNARFG1	000047	DMKCFM	DMKDIB	DMKDSP	DMKLOG	DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCU
		DMKVCV	DMKVCW	DMKVCX							
SNARFG2	000090	DMKCQS	DMKCQT	DMKCQU	DMKDIF	DMKQCO	DMKQCQ	DMKVCN	DMKVCP	DMKVCQ	DMKVCR
		DMKVCS	DMKVCT	DMKVCU	DMKVCV	DMKVCW	DMKVCX				
SNARFG3	000021	DMKCFM	DMKDIA	DMKDIF	DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCW	DMKVCX	
SNARFORC	000005	DMKVCQ	DMKVCR	DMKVCU	DMKVCX						
SNARFSS	000006	DMKVCR	DMKVCS	DMKVCU							
SNARIF	000002	DMKVCR									
SNARILER	000003	DMKCFM	DMKVCR								
SNARINN	000028	DMKQCO	DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCU	DMKVCV	DMKVCW	DMKVCX
SNARLUN	000020	DMKACO	DMKBLD	DMKCKF	DMKCQG	DMKCQQ	DMKCQS	DMKCQT	DMKDIA	DMKDIB	DMKJRL
		DMKLOH	DMKLOM	DMKQCP	DMKTRD	DMKUSQ	DMKVCV	DMKVCW			
SNARMDE	000006	DMKVCR	DMKVCS	DMKVCU							
SNARNOPR	000001	DMKVCV									
SNARNXT	000018	DMKCQS	DMKCQT	DMKVCT	DMKVCV	DMKVCW	DMKVCX				
SNAROUT	000022	DMKLOH	DMKQCO	DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCU	DMKVCV	DMKVCW
		DMKVCX									
SNARPASS	000005	DMKVCR	DMKVCS								
SNARPCT	000016	DMKVCP	DMKVCQ	DMKVCR	DMKVCS						
SNARPFIM	000003	DMKVCR	DMKVCU								
SNARPFKI	000004	DMKVCU									
SNARPRMT	000003	DMKCFM	DMKVCS								
SNARPVL	000006	DMKVCP	DMKVCQ	DMKVCR	DMKVCS						
SNARRSE	000001	DMKVCW									
SNARSIZ	000005	DMKVCT	DMKVCW	DMKVCX							
SNARSPN	000002	DMKVCW									
SNARSPT	000004	DMKVCP	DMKVCV								
SNARTTY	000028	DMKCQU	DMKQCO	DMKQCQ	DMKVCN	DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCV	
SNARVDEV	000005	DMKDIA	DMKVCR	DMKVCU	DMKVCV						
SNARVMB	000017	DMKCFM	DMKCFT	DMKCQG	DMKDIA	DMKJRL	DMKLOH	DMKVCN	DMKVCR	DMKVCS	DMKVCT
		DMKVCV	DMKVCW	DMKVDS							
SNASTATS	000005	DMKCPP	DMKVCT	DMKVCW							
SNAUSER	000021	DMKACO	DMKCKF								
SNSSUBCT	000003	DMKCCO	DMKCCS								
SNSSUBST	000004	DMKCCO	DMKCCS								
SPBOUND	000006	DMKFRE	DMKFRT	DMKPXA	DMKPXB						
SPBTSTAC	000001	DMKSPL									
SPCHAR	000015	DMKCQI	DMKCSW	DMKDRD	DMKDRE	DMKHVF	DMKSPL	DMKTCS			
SPCHAR1	000006	DMKCQI	DMKCSW	DMKHVF	DMKSPL	DMKTCS					
SPCHAR2	000006	DMKCQI	DMKCSW	DMKHVF	DMKSPL	DMKTCS					
SPCHAR3	000006	DMKCQI	DMKCSW	DMKHVF	DMKSPL	DMKTCS					
SPCMCHR	000008	DMKCQI	DMKCSW	DMKHVF	DMKSPL	DMKTCS					
SPCMOD	000009	DMKCQI	DMKCSW	DMKHVF	DMKSPL	DMKTCS					
SPCOPYFG	000007	DMKCQI	DMKCSW	DMKSPL	DMKTCS						
SPCPTRAP	000002	DMKSPS									
SPECIALV	000058	DMKBLD	DMKCPY	DMKERM	DMKHPS	DMKPTR	DMKRPA	DMKSTR	DMKVSE	DMKVSG	DMKWRM
SPFCB	000009	DMKCQI	DMKCSW	DMKHVF	DMKSPL	DMKTCS					
SPFILID	000014	DMKACO	DMKDRD	DMKDRE	DMKMIA	DMKTRR	DMKTRT	DMKTRU	DMKVSP	DMKVSQ	
SPFLAG1	000006	DMKCQI	DMKCSW	DMKSPL	DMKTCS						
SPFLSHC	000006	DMKCQI	DMKCSW	DMKHVF	DMKSPL	DMKTCS					
SPLINK	000064	DMKACO	DMKAPT	DMKCKF	DMKCKV	DMKCQI	DMKCSV	DMKCSW	DMKCSY	DMKDRD	DMKDRE

LABEL	COUNT	REFERENCES									
		DMKHVF	DMKMIA	DMKRSP	DMKRSQ	DMKRST	DMKSPK	DMKSPL	DMKSPS	DMKTCS	DMKTRR
		DMKTRT	DMKTRU	DMKVDD	DMKVME	DMKVSP	DMKVSQ	DMKVST	DMKVSW	DMKVSX	DMKXAB
		DMKXAD									
SPM	000002	DMKCKM	DMKCPI								
SPMPFX	000074	DMKCCW	DMKCDB	DMKCDM	DMKCDS	DMKCFG	DMKCFP	DMKCPP	DMKCPS	DMKCPU	DMKDMP
		DMKFPS	DMKHVD	DMKMPO	DMKPGS	DMKPMA	DMKPSA	DMKQVM	DMKSAD	DMKSPM	DMKVAT
		DMKVIO	DMKVME	DMKVRR							
SPNTR	000009	DMKFRE									
SPNXTPAG	000084	DMKACO	DMKAPT	DMKCKF	DMKCKV	DMKCSV	DMKCSW	DMKCSY	DMKDRD	DMKMIA	DMKRSP
		DMKRSQ	DMKRST	DMKSPS	DMKTCS	DMKTRR	DMKTRT	DMKVDD	DMKVME	DMKVSP	DMKVSQ
		DMKVSX	DMKXAB	DMKXAD							
SPOOLED	000036	DMKCPS	DMKDMQ	DMKMCC	DMKMCD	DMKMIA	DMKMNI	DMKMON			
SPPGLEN	000003	DMKSPL									
SPPREPAG	000050	DMKACO	DMKCKF	DMKCKV	DMKDRD	DMKMIA	DMKRSP	DMKRSQ	DMKRST	DMKSPS	DMKTCS
		DMKTRR	DMKTRT	DMKVME	DMKVSP	DMKVSQ	DMKXAB	DMKXAD			
SPRECMAX	000001	DMKSPL									
SPRECNUM	000026	DMKACO	DMKCKV	DMKDRD	DMKMIA	DMKRSP	DMKRST	DMKTRU	DMKVME	DMKVSQ	DMKVSX
SPRMISC	000036	DMKRSP	DMKRSQ	DMKRST	DMKSPK	DMKSPS	DMKTRR	DMKTRT	DMKVSP	DMKVSQ	
SPROFCL	000003	DMKMCC	DMKMON								
SPSIZE	000052	DMKDRD	DMKDRE	DMKRSP	DMKRSQ	DMKRST	DMKSAD	DMKSEP	DMKTRR	DMKTRT	DMKTRU
		DMKVME	DMKVSQ	DMKVSX							
SPSPLNKC	000006	DMKCQI	DMKCSV	DMKCSW	DMKCSY	DMKHVF	DMKSPL				
SPTBLOK	000006	DMKIOT	DMKSPR	DMKSPS	DMKSPT						
SPTBUFR1	000012	DMKSPS	DMKSPT								
SPTBUFR2	000004	DMKSPS	DMKSPT								
SPTBUFV1	000007	DMKSPS	DMKSPT								
SPTBUFV2	000005	DMKSPS	DMKSPT								
SPTCAN	000008	DMKSPS	DMKSPT								
SPTCLAS	000003	DMKSPS	DMKSPT								
SPTCLASS	000004	DMKSPS	DMKSPT								
SPTCODE	000005	DMKSPS									
SPTDEST	000005	DMKSPS	DMKSPT								
SPTDESTS	000003	DMKSPS	DMKSPT								
SPTDONE	000003	DMKSPS									
SPTENTPT	000004	DMKSPS	DMKSPT								
SPTFILES	000005	DMKSPS	DMKSPT								
SPTFLAG	000016	DMKSPS	DMKSPT								
SPTFLAG1	000030	DMKSPS									
SPTFLAG2	000032	DMKSPS	DMKSPT								
SPTFLAG3	000006	DMKIOT	DMKSPS	DMKSPT							
SPTFORM	000004	DMKSPS	DMKSPT								
SPTFRMST	000003	DMKSPS	DMKSPT								
SPTIME	000013	DMKACO	DMKCKV	DMKFRE	DMKMIA	DMKTRU	DMKVSP	DMKVSQ			
SPTINTAD	000002	DMKIOT	DMKSPS								
SPTINTR	000011	DMKSPS	DMKSPT								
SPTIOBAD	000004	DMKSPS	DMKSPT								
SPTISSUR	000007	DMKSPR	DMKSPS	DMKSPT							
SPTLAST	000004	DMKSPS									
SPTLDPRT	000007	DMKSPS									
SPTLDRDR	000005	DMKSPS									
SPTLINK	000025	DMKSPS									
SPTLKLST	000003	DMKSPS									
SPTLOAD	000003	DMKSPS	DMKSPT								
SPTMODE	000002	DMKSPT									
SPTMSGAD	000005	DMKSPS	DMKSPT								
SPTNOH	000006	DMKSPS	DMKSPT								
SPTOFLOW	000006	DMKIOT	DMKSPS	DMKSPT							
SPTOSFID	000002	DMKSPR	DMKSPS								

LABEL	COUNT	REFERENCES							
SPTOUSER	000002	DMKSPR	DMKSPS						
SPTPRT	000006	DMKSPS	DMKSPT						
SPTPUR	000003	DMKSPS	DMKSPT						
SPTRADDR	000004	DMKSPR	DMKSPS	DMKSPT					
SPTRDEV	000004	DMKSPS	DMKSPT						
SPTRDR	000012	DMKSPS	DMKSPT						
SPTREAD	000003	DMKSPS							
SPTREW	000003	DMKSPS	DMKSPT						
SPTRUN	000004	DMKSPS	DMKSPT						
SPTSAD	000003	DMKSPS	DMKSPT						
SPTSFB	000014	DMKSPS							
SPTSHOLD	000006	DMKSPS	DMKSPT						
SPTSIZE	000004	DMKSPS	DMKSPT						
SPTSPID1	000005	DMKSPS	DMKSPT						
SPTSPID2	000007	DMKSPS	DMKSPT						
SPTSTOP	000005	DMKSPS	DMKSPT						
SPTSYS	000004	DMKSPT							
SPTTM	000002	DMKSPS							
SPTUHOLD	000006	DMKSPS	DMKSPT						
SPTUNLD	000005	DMKSPS							
SPTUSER	000007	DMKSPS	DMKSPT						
SPTXAB	000012	DMKSPS							
SPTXABLK	000002	DMKSPS							
SPUBLOK	000043	DMKCKR	DMKVSD	DMKVSE	DMKVSF	DMKVSG	DMKWRM		
SPUFIRST	000019	DMKVSD	DMKVSE	DMKVSF	DMKVSG				
SPUIND	000004	DMKCKR	DMKVSG	DMKWRM					
SPULASGN	000006	DMKVSG							
SPULAST	000019	DMKVSD	DMKVSE	DMKVSG					
SPUMAP	000011	DMKVSG							
SPUMAPSZ	000011	DMKVSG							
SPUNEXT	000015	DMKVSE	DMKVSG						
SPURCNT	000013	DMKVSD	DMKVSF	DMKVSG					
SPUSIZEB	000003	DMKVSE	DMKVSG						
SPUSRTID	000024	DMKVSG							
SPUSYSID	000008	DMKVSG	DMKWRM						
SPUUSER	000003	DMKVSD	DMKVSG						
SPXCR6	000005	DMKCPS	DMKDMP	DMKPMA	DMKQVM				
SRTBLOK	000021	DMKCRM	DMKIDR						
SRTFLAG	000032	DMKCRM	DMKIDR						
SRTGIND	000004	DMKCRM	DMKIDR						
SRTIPND	000013	DMKCRM	DMKIDR						
SRTNEXT	000008	DMKCRM	DMKIDR						
SRTPATH	000010	DMKIDR							
SRTPREV	000005	DMKIDR							
SRTRESID	000003	DMKIDR							
SRTRRPND	000005	DMKIDR							
SRTRVPND	000010	DMKCRM	DMKIDR						
SRTSIZE	000008	DMKIDR							
SRTSPND	000005	DMKIDR							
SRTVMADD	000003	DMKIDR							
SSAVE	000004	DMKIMG							
SSBALLOC	000008	DMKPGM	DMKPGS	DMKPTR	DMKSTR	DMKSWM			
SSBBPNT	000007	DMKPGM	DMKPTR	DMKSEL	DMKSWA				
SSBCODE	000007	DMKMON	DMKPAH	DMKPGM	DMKPTR	DMKSWA			
SSBCPG	000004	DMKSEL							
SSBCYL	000017	DMKCKM	DMKMON	DMKPAG	DMKPGM	DMKPGU	DMKPTR	DMKSWA	DMKSWM
SSBEFLG	000036	DMKPAG	DMKPGM	DMKPGU	DMKPTR	DMKSEL	DMKSWA	DMKSWM	
SSBEINVL	000028	DMKPAG	DMKPGM	DMKPGU	DMKPTR	DMKSEL	DMKSWA	DMKSWM	

LABEL	COUNT	REFERENCES									
SSBENTRL	000033	DMKBLD	DMKCKM	DMKLOG	DMKPAG	DMKPGM	DMKPGS	DMKPGU	DMKPTR	DMKSEL	DMKSTR
		DMKSWA	DMKSWM	DMKVBM							
SSBENTRY	000021	DMKCKM	DMKPAG	DMKPGM	DMKPGS	DMKPGU	DMKPTR	DMKSEL	DMKSTR	DMKSWA	DMKSWM
SSBENUM	000012	DMKCKM	DMKPAG	DMKPGM	DMKPGS	DMKPGU	DMKPTR	DMKSTR	DMKSWA		
SSBEPGS	000027	DMKMON	DMKPAG	DMKPGM	DMKPTR	DMKSWA	DMKSWM				
SSBEXTND	000008	DMKPTR	DMKSWA	DMKSWM							
SSBFLAG	000047	DMKMON	DMKPGM	DMKPGS	DMKPTR	DMKPTS	DMKSEL	DMKSTR	DMKSWA	DMKSWM	
SSBFLUSH	000002	DMKSEL	DMKSWA								
SSBFPNT	000019	DMKMON	DMKPGM	DMKPTR	DMKPTS	DMKSEL	DMKSWA				
SSBHEADL	000027	DMKBLD	DMKCKM	DMKLOG	DMKPAG	DMKPGM	DMKPGS	DMKPGU	DMKPTR	DMKSEL	DMKSTR
		DMKSWA	DMKSWM	DMKVBM							
SSBLOK	000044	DMKCKM	DMKMON	DMKPAG	DMKPAH	DMKPGM	DMKPGS	DMKPGU	DMKPTR	DMKPTS	DMKSEL
		DMKSTR	DMKSWA	DMKSWM							
SSBMGREF	000003	DMKPGM	DMKSWA								
SSBNDLCT	000006	DMKPGM	DMKPTR	DMKSWA	DMKSWM						
SSBNPGS	000023	DMKMON	DMKPAG	DMKPGM	DMKPGU	DMKPTR	DMKSEL	DMKSWA	DMKSWM		
SSBOLD	000006	DMKMON	DMKPTR	DMKPTS	DMKSWA						
SSBPGSP	000002	DMKMON	DMKSWA								
SSBPSTOR	000005	DMKMON	DMKPGM	DMKPTR	DMKSWA						
SSBSWAP	000001	DMKSWA									
SSBTRANS	000024	DMKPGM	DMKPGS	DMKPTR	DMKSEL	DMKSTR	DMKSWA	DMKSWM			
SSBUPGS	000027	DMKMON	DMKPAG	DMKPGM	DMKPTR	DMKSWA	DMKSWM				
SSBVMSCB	000006	DMKPTR	DMKSEL	DMKSWA							
SSBVPAGE	000031	DMKMON	DMKPAG	DMKPGM	DMKPGU	DMKPTR	DMKSEL	DMKSWA	DMKSWM		
SSCBADDR	000014	DMKCPJ	DMKCRM	DMKIDR							
SSCBLOK	000017	DMKCPJ	DMKCRM	DMKIDR							
SSCIXADD	000014	DMKCPJ	DMKCRM	DMKIDR							
SSCIXLST	000015	DMKCPJ	DMKCRM	DMKIDR							
SSCRTADD	000005	DMKCPJ	DMKCRM	DMKIDR							
SSCRTCNT	000006	DMKCRM	DMKIDR								
SSCRTLST	000002	DMKCPJ	DMKIDR								
SSCSIZE	000002	DMKCPJ									
SSCTSFPT	000006	DMKCPJ	DMKCRM	DMKIDR							
SSCTSFVM	000009	DMKCRM	DMKIDR								
SSEB	000002	DMKPRG	DMKVAT								
SSEBC	000001	DMKPRG									
SSEF	000001	DMKPRG									
SSM	000009	DMKCCD	DMKCCO								
STACKVM	000013	DMKCPP	DMKDSP	DMKLOH							
STAKSIZE	000001	DMKFRT									
STARTIME	000019	DMKBLD	DMKCKF	DMKSCH	DMKSTP	DMKTOD	DMKWRM				
STCODE	000004	DMKACO	DMKCKF								
STOACTV	000017	DMKFPS	DMKPRG	DMKVAT	DMKVAU						
STOBLKLN	000004	DMKVAT									
STOBLOK	000053	DMKFPS	DMKPRG	DMKVAT	DMKVAU						
STOFLAG	000038	DMKFPS	DMKPRG	DMKVAT	DMKVAU						
STOFSTUS	000005	DMKVAT	DMKVAU								
STOLAST	000016	DMKFPS	DMKVAT								
STONEXT	000033	DMKFPS	DMKVAT	DMKVAU							
STONXTUS	000005	DMKVAT	DMKVAU								
STOPAGVR	000003	DMKVAT									
STORSIZE	000015	DMKDMQ									
STOSEGCT	000006	DMKVAT	DMKVAU								
STOSEGVR	000018	DMKFPS	DMKVAT	DMKVAU							
STOSHCR1	000035	DMKFPS	DMKPRG	DMKVAT	DMKVAU						
STOSHLEN	000009	DMKVAT									
STOSHSEG	000001	DMKVAT									
STOSIZE	000001	DMKVAT									

LABEL	COUNT	REFERENCES									
STOUSPT	000009	DMKFPS	DMKVAT	DMKVAU							
STOUSPTL	000012	DMKFPS	DMKVAT								
STOVCR1	000016	DMKFPS	DMKVAT								
STOVLEN	000004	DMKVAT	DMKVAU								
STOVPPG	000003	DMKFPS	DMKVAT								
STOVPSG	000010	DMKFPS	DMKVAT								
STO6CPG	000002	DMKFPS	DMKVAT								
STO6CSG	000005	DMKFPS	DMKVAT								
STPDRP	000004	DMKMON	DMKSTP								
STRTADDR	000001	DMKIMG									
STRTNEW	000002	DMKCCW									
STX	000012	DMKRGD									
STYPRDPT	000002	DMKGRF	DMKGRG								
SUBADDR	000015	DMKTEE	DMKTEF								
SUBBDTE	000010	DMKTEE	DMKTEF								
SUBCNT	000002	DMKMOO									
SUBCSFLG	000009	DMKTEE	DMKTEF								
SUBCSWPT	000006	DMKTEE	DMKTEF								
SUBFBYTE	000010	DMKTEE	DMKTEF								
SUBHEAD	000008	DMKFRT	DMKMOO	DMKPXA	DMKPXB						
SUBLEVEL	000003	DMKTEE	DMKTEF								
SUBLIST	000031	DMKTEE	DMKTEF								
SUBREAL	000003	DMKTEE	DMKTEF								
SUBSIZES	000028	DMKFRE									
SUBSTAT	000005	DMKTEE	DMKTEF								
SUBSTLA	000006	DMKFRE	DMKMOO	DMKPXA	DMKPXB						
SUBSTLS	000002	DMKFRE									
SUBTABLE	000006	DMKFRT	DMKMOO	DMKPXA	DMKPXB						
SUBTABMX	000005	DMKFRE	DMKFRT								
SUBTOP	000017	DMKFRE	DMKFRT								
SUBVIRT	000003	DMKTEE	DMKTEF								
SUBVMBAD	000003	DMKTEE	DMKTEF								
SUBVMFLG	000004	DMKTEE	DMKTEF								
SUSPEND	000017	DMKMIA	DMKMNI	DMKMON							
SVCNPSW	000022	DMKAPI	DMKCPI	DMKMCT	DMKPRG	DMKSAD	DMKSVC	DMKSVD	DMKTRC		
SVCOPSW	000045	DMKPRG	DMKSAD	DMKSVC	DMKSVD	DMKTRC					
SVCREGS	000021	DMKSVC	DMKSVD								
SVMNOUPD	000005	DMKCFO	DMKLOK	DMKSPR	DMKSPS						
SVMSTAY	000001	DMKLOK									
SVMUNLOK	000011	DMKACR	DMKCNS	DMKCPY	DMKFRE	DMKLOK	DMKRGA	DMKRNH	DMKSPR	DMKSPS	
SVSEN	000004	DMKCCO	DMKCCW								
SWPALLOC	000026	DMKATS	DMKPGM	DMKPGS	DMKPTR	DMKSEL	DMKSTR				
SWPAPP	000005	DMKATS	DMKCFF	DMKPTR	DMKSEL						
SWPCHG1	000032	DMKATS	DMKCFF	DMKCPP	DMKHVD	DMKMCH	DMKPAG	DMKPGM	DMKPRW	DMKPTR	DMKRPA
		DMKSEG	DMKSEL	DMKSWA	DMKSWM	DMKVSE	DMKWRM				
SWPCHG2	000011	DMKHVD	DMKMCH	DMKRPA	DMKSEL	DMKSWA	DMKSWM				
SWPCODE	000013	DMKATS	DMKCKW	DMKPAH	DMKPGM	DMKPGS	DMKPTR	DMKSEL	DMKVSE	DMKVSG	
SWPCYL	000083	DMKATS	DMKCDS	DMKCFF	DMKCFG	DMKCFU	DMKCKM	DMKCKW	DMKCPP	DMKDRD	DMKDRE
		DMKHVD	DMKHVF	DMKPAG	DMKPGM	DMKPGS	DMKPGU	DMKPTR	DMKRPA	DMKSEG	DMKSEL
		DMKSTR	DMKSWA	DMKUDU							
SWPFLAG	000297	DMKATS	DMKBLD	DMKCCW	DMKCDS	DMKCFF	DMKCFG	DMKCFH	DMKCFU	DMKCKM	DMKCKW
		DMKCPP	DMKDGD	DMKDRD	DMKDRE	DMKHVD	DMKHVF	DMKMCH	DMKPAG	DMKPAH	DMKPGM
		DMKPGS	DMKPGU	DMKPRV	DMKPRW	DMKPSA	DMKPTR	DMKPTS	DMKPTT	DMKQVM	DMKRPA
		DMKSEG	DMKSEL	DMKSPM	DMKSTR	DMKSWA	DMKSWM	DMKUDU	DMKVAT	DMKVMA	DMKVME
		DMKVSE	DMKVSG	DMKWRM							
SWPFLAG2	000005	DMKATS	DMKCFF	DMKPTR	DMKSEL						
SWPKEY1	000046	DMKATS	DMKCFF	DMKCFH	DMKCKM	DMKDRD	DMKMCH	DMKPGS	DMKPRV	DMKPRW	DMKPTR
		DMKQVM	DMKSEL	DMKSPM	DMKVME						

LABEL	COUNT	REFERENCES									
SWPKEY2	000029	DMKDRD	DMKMCH	DMKPRW	DMKQVM	DMKSEL	DMKSPM	DMKVME			
SWPPAG	000013	DMKATS	DMKBLD	DMKCCW	DMKCFF	DMKPSA	DMKPTR	DMKPTT	DMKSEL	DMKSTR	
SWPPSTOR	000013	DMKATS	DMKCKW	DMKPAG	DMKPGM	DMKPGS	DMKPGT	DMKPGU	DMKPTR	DMKSEL	DMKVSE
		DMKVSG									
SWPRECMP	000066	DMKATS	DMKBLD	DMKCDS	DMKCFF	DMKCFG	DMKCFU	DMKCKM	DMKCPP	DMKDRD	DMKDRE
		DMKHVD	DMKHVF	DMKPAG	DMKPAH	DMKPGM	DMKPGS	DMKPGU	DMKPTR	DMKRPA	DMKSEL
		DMKSTR	DMKSWA	DMKSWM	DMKUDU	DMKVSE	DMKWRM				
SWPREF1	000005	DMKHVD	DMKPRW	DMKPTS	DMKSEL						
SWPREF2	000004	DMKHVD	DMKPTS	DMKSEL							
SWPSEGNO	000008	DMKBLD	DMKPGS	DMKPTT	DMKSEL	DMKSTR	DMKSWA				
SWPSHR	000017	DMKATS	DMKCFF	DMKDGD	DMKPGS	DMKPRV	DMKPRW	DMKPTR	DMKRPA	DMKSEL	DMKVAT
SWPTABLE	000055	DMKATS	DMKBLD	DMKCFF	DMKCPP	DMKPGM	DMKPGS	DMKPTR	DMKPTT	DMKSEL	DMKSTR
		DMKSWA	DMKVAT	DMKVMA	DMKWRM						
SWPTRANS	000049	DMKCDS	DMKPAG	DMKPGM	DMKPGS	DMKPTR	DMKRPA	DMKSEL	DMKSTR	DMKSWA	DMKSWM
		DMKVMA									
SWPVM	000019	DMKATS	DMKBLD	DMKCFF	DMKCPP	DMKPGS	DMKSEL	DMKSTR			
SWPVPAGE	000037	DMKATS	DMKCCW	DMKPGM	DMKPGS	DMKPSA	DMKPTR	DMKPTS	DMKPTT	DMKSEL	DMKSTR
		DMKSWA	DMKVMA								
SWTHSAVE	000010	DMKSTK	DMKVMA								
SYNCLOG	000002	DMKCPI	DMKCPJ								
SYNCMASK	000009	DMKCLK	DMKDSP	DMKEXT	DMKIOT	DMKPMA					
SYSCPRD	000001	DMKMHV									
SYSCPWT	000001	DMKMHV									
SYSDFLT	000001	DMKLOH									
SYSIPLDV	000005	DMKCPI	DMKMNT	DMKSEG							
SYSLOCS	000032	DMKACO	DMKBLD	DMKCFT	DMKCFU	DMKCFV	DMKCKF	DMKLOC	DMKLOJ	DMKLOM	DMKUDR
		DMKUDU	DMKUSQ	DMKVCT	DMKVCW						
SYSNAME	000025	DMKATS	DMKCFF	DMKCFG	DMKCFH	DMKCFS	DMKCKM	DMKCPP	DMKCQY		
SYSOPER	000002	DMKLOH									
SYSPALL	000004	DMKSRM	DMKSTP								
SYSPALOC	000030	DMKMOO	DMKPGT	DMKPST	DMKSTP	DMKUSP	DMKVDG	DMKVDH	DMKXST		
SYSPATYP	000002	DMKPST	DMKVDH								
SYSPBPNT	000002	DMKPGT	DMKVDG								
SYSPCNT	000004	DMKPST	DMKVDG	DMKVDH							
SYSPCYL1	000011	DMKPST	DMKVDG	DMKVDH	DMKXST						
SYSPCYL2	000015	DMKPST	DMKVDG	DMKVDH	DMKXST						
SYSPDASD	000001	DMKVDH									
SYSPEXT	000012	DMKSRM	DMKSTP								
SYSPEXTN	000005	DMKSRM	DMKSTP								
SYSPFLG	000035	DMKPGT	DMKPST	DMKSRM	DMKSTP	DMKSWM	DMKVDG	DMKXST			
SYSPFLG2	000006	DMKPGM	DMKPST	DMKVDH	DMKXST						
SYSPFPNT	000018	DMKMOO	DMKPGM	DMKPGT	DMKPST	DMKSRM	DMKSTP	DMKSWM	DMKUSP	DMKVDG	DMKVDH
SYSPFULL	000004	DMKSRM	DMKSTP								
SYSPLIST	000015	DMKMOO	DMKPGM	DMKPGT	DMKPST	DMKSRM	DMKSTP	DMKSWM	DMKUSP	DMKVDG	DMKVDH
		DMKXST									
SYSPLVNO	000001	DMKSRM									
SYSPMGOU	000002	DMKSTP									
SYSPOVFL	000002	DMKSTP									
SYSPPAG	000001	DMKPST									
SYSPPCNT	000002	DMKSTP									
SYSPPST	000002	DMKPGM	DMKPST								
SYSPPUSE	000002	DMKSTP									
SYSPRIV	000001	DMKCCT									
SYSPTOTL	000002	DMKSRM	DMKSTP								
SYSPUSER	000001	DMKVDH									
SYSPVLEN	000004	DMKPST	DMKVDG	DMKVDH							
SYSPVLST	000003	DMKPST	DMKVDH								
SYSPVOL	000010	DMKPST	DMKVDG	DMKVDH							

LABEL	COUNT	REFERENCES									
SYSPXST	000003	DMKPST	DMKXST								
SYSSERV	000005	DMKCCD	DMKCCF	DMKCCO	DMKCCS						
SYSSIZE	000004	DMKVMD									
SYSTEM	000622	DMKACO	DMKATS	DMKCFC	DMKCFG	DMKCFH	DMKCFM	DMKCFS	DMKCFU	DMKCKR	DMKCKS
		DMKCKV	DMKCNS	DMKCPB	DMKCPI	DMKCPJ	DMKCPM	DMKCPO	DMKCPP	DMKCPS	DMKCPU
		DMKCPW	DMKCPY	DMKCQI	DMKCQS	DMKCQY	DMKCSC	DMKCSO	DMKCST	DMKCSV	DMKCSW
		DMKCSY	DMKDRD	DMKDRE	DMKERM	DMKGRA	DMKGRC	DMKGRF	DMKGRT	DMKHVD	DMKHVE
		DMKHVF	DMKIDU	DMKIOF	DMKIOG	DMKIOH	DMKIOT	DMKLNK	DMKMCC	DMKMIA	DMKMNJ
		DMKNLD	DMKNLE	DMKOPE	DMKPMA	DMKPTR	DMKRGC	DMKRNH	DMKRPA	DMKRSP	DMKRSQ
		DMKRST	DMKSEG	DMKSEP	DMKSNC	DMKSPK	DMKSPL	DMKSPS	DMKSPT	DMKSRM	DMKSSP
		DMKSSS	DMKSST	DMKSVC	DMKTCS	DMKTCT	DMKTEE	DMKTOD	DMKTRP	DMKTRT	DMKTRU
		DMKTTX	DMKUDR	DMKUDU	DMKVBM	DMKVCH	DMKVCX	DMKVDD	DMKVDR	DMKVME	DMKVSP
		DMKVSQ	DMKVST	DMKVSW	DMKVSX	DMKWRM	DMKWRN	DMKXAB	DMKXAD		
SYSTEMID	000007	DMKVMC									
SYSVIRT	000047	DMKBIO	DMKCCO	DMKCCW	DMKCFQ	DMKCPW	DMKCQP	DMKDEF	DMKDEI	DMKDGD	DMKDID
		DMKDSB	DMKIOS	DMKIOT	DMKLNK	DMKLOJ	DMKMNT	DMKSSS	DMKSST	DMKVDA	DMKVSC
SYSXCDEL	000002	DMKSTP									
SYSXCKP	000002	DMKSTP									
SYSXLONG	000003	DMKSTP									
SYSXMGCT	000001	DMKSRM									
SYSXOVCT	000001	DMKSRM									
SYSXPDEL	000002	DMKSTP									
SYSXRWCT	000002	DMKSRM	DMKSTP								
SYSXSHRT	000006	DMKSRM	DMKSTP								
SYSXSIZE	000012	DMKSTP									
S12	000002	DMKFRE									
S15	000002	DMKFRE									
S18	000002	DMKFRE									
S21	000002	DMKFRE									
S24	000002	DMKFRE									
S27	000002	DMKFRE									
S3	000002	DMKFRE									
S30	000002	DMKFRE									
S33	000002	DMKFRE									
S6	000002	DMKFRE									
S9	000002	DMKFRE									
TAB	000008	DMKGRI	DMKRGB	DMKRGC	DMKVCU	DMKVMI					
TABWDTH	000003	DMKMOO									
TABWDTH$	000004	DMKFRE	DMKFRT								
TEMPR0	000126	DMKAPI	DMKATS	DMKCFM	DMKCFV	DMKCNT	DMKCPI	DMKCSX	DMKDSP	DMKGRC	DMKGRF
		DMKGRG	DMKGRI	DMKGRT	DMKPRV	DMKPTS	DMKQCQ	DMKRGC	DMKSCH	DMKSTP	DMKVCA
		DMKVFR	DMKVMA	DMKVSR	DMKVSX						
TEMPR1	000073	DMKATS	DMKCFM	DMKCKV	DMKCPI	DMKGRC	DMKGRT	DMKIDU	DMKMHC	DMKMNI	DMKPGS
		DMKPRG	DMKPTR	DMKQCQ	DMKRGA	DMKRSP	DMKRSQ	DMKSTP	DMKVSQ	DMKVSR	DMKVSX
TEMPR10	000020	DMKATS	DMKCCW	DMKDSP	DMKPRG	DMKSCH	DMKSSU	DMKSTP			
TEMPR11	000008	DMKEXT	DMKPMA	DMKSCH							
TEMPR12	000011	DMKCKP	DMKFPS	DMKPRG	DMKSCH	DMKSSU	DMKVSR				
TEMPR13	000004	DMKEXT	DMKPRG	DMKSCH							
TEMPR14	000049	DMKATS	DMKCCW	DMKCDS	DMKCPJ	DMKFPS	DMKGRC	DMKMNT	DMKPRG	DMKSCH	DMKSEL
		DMKSSU	DMKSVC	DMKVCR							
TEMPR15	000028	DMKATS	DMKCCW	DMKCDS	DMKGRC	DMKMNT	DMKPRG	DMKRGA	DMKSCH	DMKVSC	DMKVSQ
TEMPR2	000104	DMKAPI	DMKATS	DMKCCW	DMKCKP	DMKCPI	DMKCSU	DMKCSV	DMKCSX	DMKEXT	DMKGRC
		DMKIUS	DMKMHC	DMKPMA	DMKPTR	DMKRGA	DMKSAV	DMKSTP	DMKVCA	DMKVCX	DMKVMA
TEMPR3	000044	DMKAPI	DMKCCW	DMKCPI	DMKCSW	DMKGRC	DMKGRG	DMKMNI	DMKPTR	DMKRGC	DMKSTP
		DMKVAT	DMKVCA	DMKVCV	DMKVSP						
TEMPR4	000093	DMKAPI	DMKCKP	DMKCPI	DMKCSW	DMKGRC	DMKPMA	DMKPRV	DMKPTS	DMKSAV	DMKSCH
		DMKSTP	DMKVCA	DMKVFR	DMKVSG	DMKVSR					
TEMPR5	000048	DMKAPI	DMKCPI	DMKPRV	DMKRGA	DMKSTP	DMKVCA	DMKVFR	DMKVMA	DMKVSI	

LABEL	COUNT	REFERENCES									
TEMPR6	000035	DMKAPI	DMKCCO	DMKCKP	DMKCPI	DMKDSP	DMKFRE	DMKGRT	DMKHVC	DMKRGA	DMKSTP
		DMKVFR	DMKVMA								
TEMPR7	000014	DMKAPI	DMKSCH	DMKSTP	DMKSVD	DMKVCR					
TEMPR8	000007	DMKHVC	DMKPTT	DMKVFR							
TEMPR9	000011	DMKGRC	DMKIOT	DMKMNT	DMKSCH	DMKVFR					
TEMPSAVE	000239	DMKACO	DMKAPI	DMKCCT	DMKCFS	DMKCFY	DMKCKF	DMKCKH	DMKCKP	DMKCKW	DMKCNS
		DMKCNT	DMKCPJ	DMKCPM	DMKCPZ	DMKCVT	DMKCVU	DMKDAD	DMKDAS	DMKDSP	DMKEXT
		DMKFPS	DMKFRE	DMKGRG	DMKIDU	DMKIOS	DMKIUS	DMKLOJ	DMKMID	DMKMNT	DMKNET
		DMKNLD	DMKOPE	DMKPGM	DMKPGU	DMKPRG	DMKPRV	DMKQCQ	DMKRGA	DMKRNH	DMKSAV
		DMKSCH	DMKSSU	DMKSTP	DMKTOD	DMKTRQ	DMKTTY	DMKVDG	DMKVIO	DMKVRR	DMKVSC
		DMKVSD	DMKVSJ								
TERMCHAN	000004	DMKCCH									
TERMEXIT	000015	DMKACO	DMKCKF								
TERMRMID	000002	DMKACO	DMKCKF								
TERMSYS	000007	DMKCCH	DMKEIG	DMKSEV	DMKSIX						
TEXT	000070	DMKIMG									
THISRCW	000013	DMKCCD	DMKCCO	DMKCCS	DMKCCW						
TICBLK	000002	DMKCCW									
TIMEDISP	000297	DMKACO	DMKACR	DMKALG	DMKAPI	DMKBLD	DMKCAO	DMKCFO	DMKCFQ	DMKCFR	DMKCLK
		DMKCNS	DMKCPI	DMKCPJ	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPU	DMKCPV	DMKCPY
		DMKCSW	DMKCSX	DMKDIA	DMKDIB	DMKDIF	DMKDSP	DMKEXT	DMKFPS	DMKFRE	DMKGRI
		DMKHPS	DMKIOQ	DMKIOS	DMKIOT	DMKIUA	DMKIUE	DMKIUL	DMKJRL	DMKLOG	DMKLOH
		DMKLOJ	DMKLOK	DMKLOM	DMKMCH	DMKMCT	DMKMIA	DMKMID	DMKMPO	DMKMSG	DMKMSW
		DMKNEA	DMKNES	DMKNET	DMKNLD	DMKOPE	DMKPAG	DMKPGM	DMKPMA	DMKPRG	DMKQCN
		DMKQCO	DMKRGA	DMKRGB	DMKRNH	DMKSCH	DMKSEL	DMKSND	DMKSPK	DMKSPL	DMKSPM
		DMKSPR	DMKSPS	DMKSSS	DMKSSV	DMKSTP	DMKSTR	DMKSVD	DMKSWA	DMKTCS	DMKTCT
		DMKTRQ	DMKUDU	DMKUSQ	DMKVCB	DMKVCH	DMKVCP	DMKVCS	DMKVCT	DMKVCU	DMKVCW
		DMKVDA	DMKVDB	DMKVDD	DMKVDE	DMKVDF	DMKVDT	DMKVMC	DMKWAI		
TIMEPOP	000002	DMKFRE									
TIMER	000055	DMKAPI	DMKCFY	DMKCMD	DMKCPI	DMKCPJ	DMKDDR	DMKDEF	DMKDIR	DMKDSP	DMKEXT
		DMKFPS	DMKIOT	DMKMCH	DMKMPO	DMKPGS	DMKPMA	DMKPRG	DMKPTR	DMKQVM	DMKSCH
		DMKSEL	DMKSTR	DMKSVD	DMKSWA	DMKTRA	DMKTRQ				
TIOCCH	000012	DMKCCH	DMKEIG	DMKSEV	DMKSIX						
TMRDRP	000003	DMKMON	DMKTRQ								
TNSCPIDN	000001	DMKIOJ									
TNSDEVAD	000017	DMKIOJ									
TNSKEYN	000003	DMKIOJ									
TNSREC	000004	DMKIOJ									
TNSSNS1	000003	DMKIOJ									
TNSSWS3	000024	DMKIOJ									
TNSVOLID	000004	DMKIOJ									
TNS3480S	000001	DMKIOJ									
TNS3800	000003	DMKIOJ									
TNS8809S	000001	DMKIOJ									
TODATE	000036	DMKACO	DMKCKF	DMKCKM	DMKCVT	DMKDMP	DMKMCD	DMKMID	DMKMNI	DMKSAD	DMKTOD
		DMKVME									
TODCLOCK	000007	DMKDMQ									
TODSYNC	000001	DMKCLK									
TPOPFSLB	000002	DMKBSC									
TPOPRDXP	000003	DMKBSC									
TRACBEF	000007	DMKACS	DMKCNS	DMKCPM	DMKIOS	DMKIOT	DMKVSJ				
TRACCEPT	000002	DMKVCW									
TRACCURR	000114	DMKACS	DMKCNS	DMKCPM	DMKCPZ	DMKDSP	DMKEXT	DMKFRE	DMKFRT	DMKIOS	DMKIOT
		DMKIUA	DMKLOK	DMKMCC	DMKMCH	DMKMHC	DMKMNT	DMKPMA	DMKPRG	DMKPTS	DMKRNH
		DMKSCH	DMKSTA	DMKSVC	DMKSVD	DMKSWA	DMKTRT	DMKTRU	DMKVCV	DMKVCX	DMKVSJ
TRACEFLG	000008	DMKMCC	DMKMCD	DMKMIA	DMKSTA						
TRACEND	000040	DMKACS	DMKCNS	DMKCPM	DMKCPZ	DMKDSP	DMKEXT	DMKFRE	DMKFRT	DMKIOS	DMKIOT
		DMKIUA	DMKLOK	DMKMCH	DMKMHC	DMKMNT	DMKPMA	DMKPRG	DMKPTS	DMKRNH	DMKSCH

LABEL	COUNT	REFERENCES									
		DMKSTA	DMKSVC	DMKSVD	DMKSWA	DMKTRT	DMKVCV	DMKVCX	DMKVSJ		
TRACEVCS	000002	DMKVCV									
TRACFLG1	000019	DMKEXT	DMKFRE	DMKFRT	DMKIOT	DMKMCH	DMKPMA	DMKPRG	DMKPTS	DMKSCH	DMKSVC
		DMKSVD	DMKSWA								
TRACFLG2	000018	DMKACS	DMKCNS	DMKCPM	DMKDSP	DMKEXT	DMKIOS	DMKIOT	DMKLOK	DMKRNH	DMKVSJ
TRACFLG3	000008	DMKDSP	DMKIUA	DMKMHC	DMKVCS	DMKVCV	DMKVCW	DMKVCX			
TRACHAR	000001	DMKVCV									
TRACIUCV	000001	DMKIUA									
TRACK7	000001	DMKTAQ									
TRACPROC	000037	DMKACS	DMKAPI	DMKCNS	DMKCPM	DMKCPZ	DMKDSP	DMKEXT	DMKFRE	DMKFRT	DMKIOS
		DMKIOT	DMKIUA	DMKLOK	DMKMCH	DMKMHC	DMKMNT	DMKMON	DMKPMA	DMKPRG	DMKPTS
		DMKRNH	DMKSCH	DMKSVC	DMKSVD	DMKSWA	DMKVCV	DMKVCX	DMKVSJ		
TRACPSAF	000001	DMKVCV									
TRACSTRT	000051	DMKACS	DMKAPI	DMKCNS	DMKCPM	DMKCPP	DMKCPZ	DMKDSP	DMKEXT	DMKFRE	DMKFRT
		DMKIOS	DMKIOT	DMKIUA	DMKLOK	DMKMCC	DMKMCH	DMKMHC	DMKMNT	DMKMPO	DMKPMA
		DMKPRG	DMKPTS	DMKRNH	DMKSAD	DMKSCH	DMKSTA	DMKSVC	DMKSVD	DMKSWA	DMKTRT
		DMKVCV	DMKVCX	DMKVSJ							
TRACSVCR	000035	DMKACS	DMKCNS	DMKCPM	DMKCPZ	DMKDSP	DMKEXT	DMKFRE	DMKFRT	DMKIOS	DMKIOT
		DMKIUA	DMKLOK	DMKMCH	DMKMHC	DMKMNT	DMKPMA	DMKPRG	DMKPTS	DMKRNH	DMKSCH
		DMKSVC	DMKSVD	DMKSWA	DMKVCV	DMKVCX	DMKVSJ				
TRACSVST	000006	DMKSAD	DMKSTA								
TRAC0A	000001	DMKDSP									
TRAC0C	000001	DMKDSP									
TRAC0D	000001	DMKVSJ									
TRAC01	000002	DMKEXT									
TRAC02	000002	DMKSVC	DMKSVD								
TRAC03	000003	DMKPMA	DMKPRG								
TRAC04	000001	DMKMCH									
TRAC05	000002	DMKIOT									
TRAC08	000003	DMKPTS	DMKSCH	DMKSWA							
TRAC09	000001	DMKSCH									
TRAC10	000001	DMKDSP									
TRAC11	000001	DMKRNH									
TRAC12	000001	DMKLOK									
TRAC13	000005	DMKEXT									
TRAC17	000001	DMKMHC									
TRAC67	000004	DMKFRE	DMKFRT								
TRAEDCHR	000001	DMKVCV									
TRAFPFLG	000001	DMKDSP									
TRAFUNCT	000001	DMKVCV									
TRAINSTR	000002	DMKVCV									
TRAIPRCD	000001	DMKVCV									
TRAIXBLK	000001	DMKVCV									
TRALGAID	000001	DMKVCV									
TRALUCON	000001	DMKVCW									
TRAMODE	000002	DMKVCV									
TRAMSGLM	000001	DMKVCV									
TRANMODE	000054	DMKCDS	DMKCFF	DMKCFG	DMKCKW	DMKCPB	DMKDAD	DMKDAS	DMKDAU	DMKDSP	DMKFPS
		DMKMCH	DMKPEI	DMKPMA	DMKPRG	DMKPRV	DMKPRW	DMKPSA	DMKPTT	DMKSAD	DMKSVD
		DMKTMR	DMKTRC	DMKTRD	DMKVAT	DMKVER	DMKVFR				
TRAP	000004	DMKMIA	DMKMON								
TRAPCP	000014	DMKPSA	DMKTRP	DMKTRT	DMKTRU						
TRAPCR8	000015	DMKDSP	DMKPRG	DMKSVD	DMKTRT	DMKTRU					
TRAPDATA	000028	DMKCKF	DMKCPP	DMKCQC	DMKTRP	DMKTRT	DMKTRU	DMKTRX			
TRAPOK	000004	DMKTRT	DMKTRU								
TRAPSTRT	000009	DMKTRP	DMKTRU								
TRAPTT	000014	DMKCQC	DMKDSP	DMKTRP	DMKTRT	DMKTRU					
TRAPVT	000006	DMKPRG	DMKTRT	DMKTRU							

LABEL	COUNT	REFERENCES			
TRAREPLY	000001	DMKVCV			
TRASEND1	000004	DMKVCV			
TRASEND2	000003	DMKVCV			
TRASEVER	000005	DMKVCW			
TRATIMER	000001	DMKVCV			
TRATNTYP	000011	DMKVCV			
TRAUDATA	000004	DMKVCV			
TRAUDIT1	000001	DMKVCV			
TRAUDIT2	000001	DMKVCV			
TRAUSER1	000001	DMKVCV			
TRAVCSET	000005	DMKVCS	DMKVCV	DMKVCW	DMKVCX
TRAVCSPA	000001	DMKVCV			
TRAVMADR	000002	DMKVCV			
TRAVSAMC	000002	DMKVCV			
TRAVSAPA	000003	DMKVCV			
TRAVSAQS	000002	DMKVCV	DMKVCW		
TRAVSARM	000002	DMKVCV	DMKVCW		
TRAVSARP	000006	DMKVCR	DMKVCS	DMKVCV	
TRAVSASV	000002	DMKVCV	DMKVCW		
TRAVSMCN	000001	DMKVCW			
TRCCLCH	000002	DMKVSJ			
TRCCSW	000002	DMKVSJ			
TRCDROP	000002	DMKSCH			
TRCENTRY	000101	DMKTEE	DMKTEF		
TRCEXT	000002	DMKEXT			
TRCFPRUN	000001	DMKDSP			
TRCFREE	000002	DMKFRE			
TRCFREEP	000002	DMKFRE			
TRCFRET	000002	DMKFRT			
TRCFRETP	000002	DMKFRT			
TRCHALT	000002	DMKVSJ			
TRCIUCV	000002	DMKIUA			
TRCLOK	000002	DMKLOK			
TRCMCH	000002	DMKMCH			
TRCMSSF	000002	DMKMHC			
TRCNCP	000002	DMKRNH			
TRCOUTLN	000020	DMKTEE	DMKTEF		
TRCOUTRN	000010	DMKTEE	DMKTEF		
TRCPARM	000011	DMKTEE	DMKTEF		
TRCPGM	000006	DMKPMA	DMKPRG		
TRCPTSRS	000002	DMKPTS			
TRCRUN	000001	DMKDSP			
TRCSCH	000002	DMKSCH			
TRCSIGP	000002	DMKEXT			
TRCSIOF	000001	DMKIOS			
TRCSMINT	000002	DMKIOT			
TRCSVC	000004	DMKSVC	DMKSVD		
TRCSWAPI	000002	DMKSWA			
TRCTCH	000002	DMKIOS			
TRCUNBLK	000001	DMKDSP			
TRCUNSTK	000001	DMKDSP			
TRCVCS	000004	DMKVCV	DMKVCX		
TREXADD	000020	DMKPRG	DMKPRV	DMKVAT	DMKVAU
TREXANSI	000006	DMKPGS	DMKTRA	DMKTRC	DMKTRD
TREXBRAN	000015	DMKSPM	DMKTRA	DMKTRC	DMKTRD
TREXBUFF	000008	DMKTRC	DMKTRD		
TREXCCW	000006	DMKTRA	DMKTRD		
TREXCCWI	000007	DMKTRD	DMKVSI	DMKVSJ	

LABEL	COUNT	REFERENCES									
TREXCR9	000001	DMKMPO									
TREXCSW	000006	DMKTRA	DMKTRC	DMKVIO							
TREXCTL	000004	DMKTRA									
TREXCTL1	000003	DMKTRC	DMKTRD								
TREXCTL2	000017	DMKSPM	DMKTRC	DMKTRD	DMKVIO	DMKVSI	DMKVSJ				
TREXFLAG	000017	DMKDSP	DMKGRG	DMKPRV	DMKRGC	DMKTRC	DMKTRD	DMKVRS			
TREXINST	000015	DMKSPM	DMKTRA	DMKTRC	DMKTRD						
TREXIN1	000013	DMKDSP	DMKPGS	DMKPRV	DMKSVD	DMKTRA	DMKTRC	DMKTRD	DMKVRS		
TREXIN2	000010	DMKDSP	DMKTRC	DMKTRD	DMKVRS						
TREXLCNT	000006	DMKTRC	DMKTRD								
TREXLOCK	000001	DMKCFM									
TREXMOR	000004	DMKGRG	DMKRGC	DMKTRD							
TREXNDSP	000008	DMKDSP	DMKPRV	DMKTRC	DMKTRD						
TREXNSI	000014	DMKPGS	DMKPRV	DMKTRC							
TREXPERA	000001	DMKMPO									
TREXPRNT	000005	DMKTRA	DMKTRC	DMKTRD							
TREXRUNF	000006	DMKTRA	DMKTRC	DMKTRD							
TREXSIZE	000005	DMKTRA	DMKTRC	DMKTRD							
TREXSVC1	000004	DMKTRC	DMKTRD	DMKVRS							
TREXSVC2	000004	DMKTRC	DMKTRD	DMKVRS							
TREXT	000026	DMKCFM	DMKDSP	DMKGRG	DMKMPO	DMKPGS	DMKPRV	DMKRGC	DMKSPM	DMKSVD	DMKTRA
		DMKTRC	DMKTRD	DMKVIO	DMKVRS	DMKVSI	DMKVSJ				
TREXTERM	000008	DMKCFM	DMKTRA	DMKTRC	DMKTRD						
TREXVAT	000006	DMKTRC	DMKTRD	DMKVRS							
TRLANCHR	000009	DMKCPP	DMKSCH	DMKWAI							
TRLCT	000022	DMKCPP	DMKSCH	DMKWAI							
TRLINQ	000002	DMKWAI									
TRLLABEL	000011	DMKCPP	DMKDSP	DMKPXA	DMKPXB	DMKSCH	DMKWAI				
TRLLOCK	000024	DMKAPI	DMKDSP	DMKPXA	DMKPXB	DMKSCH	DMKWAI				
TRLSTL	000004	DMKPXA	DMKPXB	DMKWAI							
TRQBBPNT	000010	DMKEXT	DMKTRQ								
TRQBCHIO	000020	DMKCFQ	DMKGRD	DMKGRF	DMKGRG	DMKRGA	DMKRGB	DMKUSQ			
TRQBCLIN	000012	DMKGRD	DMKGRG	DMKRGC							
TRQBCPQ	000031	DMKCFM	DMKCFQ	DMKGRD	DMKGRE	DMKGRF	DMKGRI	DMKRGA	DMKRGB		
TRQBCRT	000061	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRI	DMKRGA	DMKRGB	DMKVDS		
TRQBDEV	000005	DMKGRG	DMKGRI	DMKRGA	DMKRGB						
TRQBFLAG	000054	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRH	DMKGRI	DMKQVM			
TRQBFLG2	000084	DMKCFM	DMKCFQ	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKRGA	DMKRGB	DMKRGC	DMKUSQ
TRQBFLG3	000002	DMKGRF									
TRQBFPNT	000030	DMKCFP	DMKCPP	DMKDIB	DMKDIF	DMKDSP	DMKEXT	DMKPMA	DMKQVM	DMKSTP	DMKTMR
		DMKTRQ	DMKUSQ								
TRQBIRA	000047	DMKBLD	DMKCFJ	DMKCFQ	DMKCFR	DMKCFU	DMKCFY	DMKCPJ	DMKCPO	DMKCQT	DMKDID
		DMKDRD	DMKDRE	DMKGRF	DMKIUJ	DMKIUS	DMKJRL	DMKLOH	DMKMCH	DMKMHC	DMKMNI
		DMKMNJ	DMKMNL	DMKPGM	DMKPGS	DMKQCP	DMKREI	DMKRGA	DMKRGB	DMKSSU	DMKSTP
		DMKTOD	DMKVCV	DMKVDA	DMKVDR	DMKVDS	DMKVMC				
TRQBLINA	000029	DMKCFM	DMKGRF	DMKGRI	DMKGRT	DMKHPT	DMKQVM	DMKRGA	DMKRGC		
TRQBLINE	000025	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRH	DMKRGA	DMKRGB	DMKRGC		
TRQBLOGP	000002	DMKGRF									
TRQBLOK	000224	DMKBLD	DMKCDS	DMKCFJ	DMKCFM	DMKCFP	DMKCFQ	DMKCFR	DMKCFU	DMKCFY	DMKCPJ
		DMKCPO	DMKCPP	DMKCQT	DMKDIB	DMKDID	DMKDIF	DMKDRD	DMKDRE	DMKDSP	DMKENT
		DMKEXT	DMKFPS	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRH	DMKGRI	DMKGRT	DMKHPT
		DMKIUJ	DMKIUS	DMKJRL	DMKLOH	DMKMCH	DMKMHC	DMKMID	DMKMNI	DMKMNJ	DMKMNL
		DMKMOO	DMKPGM	DMKPGS	DMKPMA	DMKQCP	DMKQVM	DMKREI	DMKRGA	DMKRGB	DMKRGC
		DMKSCH	DMKSSU	DMKSTP	DMKSVD	DMKTMR	DMKTOD	DMKTRQ	DMKUSQ	DMKVCV	DMKVDA
		DMKVDR	DMKVDS	DMKVMC	DMKVRR	DMKVRS					
TRQBPA1R	000007	DMKRGA	DMKRGB	DMKRGC							
TRQBPOLL	000002	DMKRGA									
TRQBQUE	000025	DMKCFP	DMKEXT	DMKFPS	DMKQVM	DMKSCH	DMKSVD	DMKTMR	DMKTRQ	DMKVRR	DMKVRS

LABEL	COUNT	REFERENCES									
TRQBSAVE	000016	DMKCFQ	DMKCFR	DMKDID	DMKMNI	DMKVDA	DMKVDR				
TRQBSIZE	000104	DMKBLD	DMKCFG	DMKCFJ	DMKCFM	DMKCFQ	DMKCFR	DMKCFS	DMKCFU	DMKCFY	DMKCPJ
		DMKCPO	DMKCQT	DMKDIA	DMKDIB	DMKDID	DMKDIF	DMKGRI	DMKGRT	DMKHPT	DMKIOQ
		DMKIUJ	DMKIUS	DMKJRL	DMKLOH	DMKLOJ	DMKMCC	DMKMCD	DMKMCH	DMKMHC	DMKMNI
		DMKMNJ	DMKQCP	DMKQVM	DMKREI	DMKRGA	DMKRGB	DMKSSU	DMKSTP	DMKTOD	DMKUSQ
		DMKVCT	DMKVCV	DMKVDA	DMKVDR	DMKVDS					
TRQBTOD	000092	DMKCFJ	DMKCFM	DMKCFQ	DMKCFR	DMKCFU	DMKCPJ	DMKCPO	DMKCQT	DMKDID	DMKENT
		DMKMCH	DMKMHC	DMKMNI	DMKMNJ	DMKMNL	DMKMOO	DMKPGM	DMKPGS	DMKPMA	DMKQVM
		DMKREI	DMKSCH	DMKSSU	DMKSTP	DMKTMR	DMKTOD	DMKTRQ	DMKVDA	DMKVDR	DMKVMC
		DMKVRR									
TRQBUSER	000054	DMKBLD	DMKCFJ	DMKCFQ	DMKCFR	DMKCFU	DMKCFY	DMKCPJ	DMKCPO	DMKCQT	DMKDID
		DMKDRD	DMKDRE	DMKDSP	DMKEXT	DMKGRF	DMKIUJ	DMKIUS	DMKJRL	DMKLOH	DMKMCH
		DMKMHC	DMKMNI	DMKMNJ	DMKMNL	DMKPGM	DMKPGS	DMKQCP	DMKREI	DMKRGA	DMKRGB
		DMKSSU	DMKSTP	DMKTOD	DMKTRQ	DMKVCV	DMKVDA	DMKVDR	DMKVDS	DMKVMC	
TRQBVAL	000103	DMKCDS	DMKCFJ	DMKCFM	DMKCFP	DMKCFQ	DMKCFR	DMKCFU	DMKCPJ	DMKCPO	DMKCPP
		DMKCQT	DMKDID	DMKDRD	DMKDRE	DMKDSP	DMKENT	DMKEXT	DMKGRF	DMKIUJ	DMKIUS
		DMKJRL	DMKMCH	DMKMHC	DMKMID	DMKMNI	DMKMNJ	DMKMNL	DMKMOO	DMKPGM	DMKPGS
		DMKPMA	DMKQCP	DMKQVM	DMKREI	DMKRGA	DMKRGB	DMKSCH	DMKSSU	DMKSTP	DMKTMR
		DMKTOD	DMKTRQ	DMKVCV	DMKVDA	DMKVDR	DMKVDS	DMKVMC	DMKVRR		
TRQCACHL	000003	DMKMNL									
TRQCACHP	000003	DMKMNL									
TRQCHN	000003	DMKTRQ									
TRQINTEG	000006	DMKGRD	DMKGRF	DMKRGB	DMKRGC						
TRQNAME	000002	DMKRGA									
TRQREGS	000010	DMKDRD	DMKDRE	DMKIUJ	DMKIUS	DMKPGM					
TRQREGSD	000010	DMKCQT	DMKDRD	DMKDRE	DMKPGM	DMKPGS	DMKVMC				
TRQREGSZ	000004	DMKDRD	DMKDRE	DMKIUJ	DMKIUS						
TRQREG0	000006	DMKCFR	DMKPGS	DMKVMC							
TRQREG2	000002	DMKCQT									
TRTABLE	000011	DMKTTX									
TRUN	000008	DMKDMQ	DMKMNI	DMKMON							
TSEND	000004	DMKMON	DMKSCH								
TSKBLOK	000003	DMKCKV									
TSKCCPD	000009	DMKCKV									
TSKFLAG1	000002	DMKCKV									
TSKFLAG3	000006	DMKCKV									
TSKFLAG4	000002	DMKCKV									
TSKLAST	000008	DMKCKV									
TSKMISC1	000002	DMKCKV									
TSKPBUFF	000007	DMKCKV									
TSKRECS	000003	DMKCKV									
TSKSFB	000006	DMKCKV									
TSKSIZE	000002	DMKCKV									
TSKSIZEB	000001	DMKCKV									
TSKSTART	000010	DMKCKV									
TSKSYSID	000002	DMKCKV									
TSKTEMP	000006	DMKCKV									
TSKTIME	000004	DMKCKV									
TTCODE	000002	DMKTRR	DMKTRT								
TTENTRY	000003	DMKTRR	DMKTRT								
TTLENGTH	000009	DMKTRR	DMKTRT	DMKTRU							
TTSEGCNT	000013	DMKATS	DMKBLD	DMKCFF	DMKCPP	DMKSTP					
TWOENTS	000004	DMKCFQ	DMKVCN	DMKVDR							
TYPBSC	000081	DMKACO	DMKBLD	DMKCFM	DMKCFT	DMKCFY	DMKCKD	DMKCKF	DMKCPM	DMKCPO	DMKCPW
		DMKCQG	DMKCQP	DMKCQQ	DMKCQS	DMKCQU	DMKCQY	DMKDIA	DMKDIB	DMKEPS	DMKERM
		DMKGRC	DMKGRT	DMKHVD	DMKHVE	DMKIOF	DMKIOS	DMKIOT	DMKJRL	DMKLOG	DMKLOH
		DMKLOM	DMKNES	DMKNLD	DMKOPE	DMKQCN	DMKQCO	DMKQCP	DMKQCQ	DMKQVM	DMKRGA
		DMKUSQ	DMKVCN	DMKVDE	DMKVDS	DMKWRM					

LABEL	COUNT	REFERENCES									
TYPCTCA	000047	DMKCFQ	DMKCFR	DMKCPB	DMKCQG	DMKDEF	DMKDIB	DMKDIR	DMKIOS	DMKSCN	DMKVCA
		DMKVDA	DMKVDR	DMKVDS	DMKVER	DMKVIO	DMKVMI	DMKVSI	DMKVSJ		
TYPFBA	000009	DMKCPW	DMKDDR	DMKDEF	DMKLNK	DMKMNT	DMKVDS				
TYPIBM1	000009	DMKCFR	DMKDEF	DMKDIA	DMKDIF	DMKDIR	DMKNLD	DMKSCN			
TYPNUMAX	000005	DMKCQC	DMKTRP	DMKTRR							
TYPNUMIN	000002	DMKTRP	DMKTRR								
TYPPRT	000091	DMKCFU	DMKCKF	DMKCKV	DMKCQI	DMKCQR	DMKCSB	DMKCSF	DMKCSO	DMKCSQ	DMKCSR
		DMKCST	DMKCSU	DMKCSV	DMKCSW	DMKCSX	DMKCSY	DMKDEF	DMKDMP	DMKDRD	DMKHVF
		DMKNLE	DMKRSP	DMKSAD	DMKSCN	DMKSPL	DMKSSP	DMKURS	DMKVDD	DMKVSP	DMKVSQ
		DMKVST	DMKXAB	DMKXAD							
TYPPUN	000079	DMKCKF	DMKCKH	DMKCQG	DMKCQR	DMKCSF	DMKCSO	DMKCSQ	DMKCSR	DMKCST	DMKCSU
		DMKCSV	DMKCSW	DMKCSX	DMKCSY	DMKDRD	DMKMNT	DMKRSE	DMKRSP	DMKSCN	DMKSEP
		DMKSPL	DMKSSP	DMKURS	DMKVDS	DMKVSP					
TYPRDR	000025	DMKACO	DMKCQI	DMKCSP	DMKCSQ	DMKCSR	DMKCST	DMKCSW	DMKCSX	DMKCSY	DMKDRD
		DMKHVF	DMKSCN	DMKSPK	DMKSPL	DMKURS	DMKVMI	DMKVSI			
TYPSDLC	000007	DMKCCW	DMKCPV	DMKIOF	DMKSCN	DMKVDA					
TYPSRF	000001	DMKIOH									
TYPSS	000001	DMKCFQ									
TYPTELE2	000005	DMKDEF	DMKDIA	DMKDIF	DMKDIR	DMKSCN					
TYPTIMER	000007	DMKCQG	DMKDEF	DMKDIR	DMKVSP						
TYPTTY	000034	DMKBLD	DMKCCT	DMKCFR	DMKCFT	DMKCFY	DMKCNS	DMKCPV	DMKCQU	DMKIOF	DMKIOS
		DMKNES	DMKQCQ	DMKVCN	DMKVCP	DMKVDA					
TYPUNDEF	000003	DMKCNS	DMKNES	DMKNLD							
TYPUNSUP	000006	DMKCCO	DMKCCS	DMKCCW	DMKVMI						
TYP1050	000008	DMKCNS	DMKIOF	DMKQCN							
TYP1052	000013	DMKDEF	DMKDIR	DMKLOJ	DMKSPL	DMKVDR	DMKVDS	DMKVST			
TYP1053	000002	DMKDID									
TYP1403	000017	DMKDEF	DMKDIR	DMKDMP	DMKIOF	DMKRSE	DMKTRT	DMKTRU	DMKVME		
TYP1442R	000002	DMKCCO									
TYP1443	000008	DMKDEF	DMKDIR	DMKIOF	DMKRSE						
TYP2305	000084	DMKACR	DMKALO	DMKCCD	DMKCCS	DMKCFH	DMKCFS	DMKCKN	DMKCPJ	DMKCPM	DMKCPT
		DMKCPW	DMKCQG	DMKCQP	DMKDAS	DMKDDR	DMKDEF	DMKDGD	DMKDIR	DMKDRD	DMKDRE
		DMKDSB	DMKHVE	DMKIOC	DMKIOE	DMKIOG	DMKIOJ	DMKIOS	DMKLOJ	DMKMNT	DMKPAG
		DMKPMA	DMKSAV	DMKSPK	DMKSPS	DMKUNT	DMKVCH	DMKVDA	DMKVDC	DMKVDD	DMKVDE
		DMKVDF	DMKVDG	DMKVDR	DMKVDS	DMKVER	DMKZTD				
TYP2311	000019	DMKCCD	DMKCQG	DMKDDR	DMKDGD	DMKDIR	DMKIOC	DMKLNK	DMKSCO	DMKVDS	
TYP2314	000066	DMKALO	DMKATS	DMKBIO	DMKCCD	DMKCFG	DMKCFH	DMKCFS	DMKCKH	DMKCKN	DMKCKT
		DMKCKV	DMKCPJ	DMKCPP	DMKCQG	DMKDAS	DMKDDR	DMKDEF	DMKDGD	DMKDIR	DMKDMP
		DMKDSB	DMKHVD	DMKLNK	DMKNLD	DMKNLE	DMKPAG	DMKPMA	DMKSAV	DMKSEG	DMKSNC
		DMKSPK	DMKSPS	DMKSSP	DMKTCS	DMKVDG	DMKVER	DMKVSC	DMKZTD		
TYP2319	000007	DMKDDR	DMKDEF	DMKDRD	DMKDRE	DMKHVE					
TYP2401	000009	DMKDDR	DMKIOS	DMKIOT	DMKTAP	DMKTAQ	DMKVMI				
TYP2415	000005	DMKDDR	DMKTAP	DMKTAQ	DMKVMI						
TYP2420	000005	DMKDDR	DMKTAP	DMKTAQ	DMKVMI						
TYP2501	000008	DMKDEF	DMKDIR	DMKIOF	DMKRSE	DMKVMI					
TYP2520P	000002	DMKRSE									
TYP2520R	000001	DMKIOF									
TYP2540P	000010	DMKACO	DMKDEF	DMKDIR	DMKMIA	DMKRSE					
TYP2540R	000013	DMKDEF	DMKDIR	DMKIOF	DMKRSE	DMKRST	DMKVMI				
TYP2700	000004	DMKCCT	DMKIOF	DMKNES	DMKSCN						
TYP2741	000014	DMKCFC	DMKCFM	DMKCNS	DMKIOF	DMKOPE					
TYP2955	000001	DMKCCT									
TYP3066	000020	DMKBLD	DMKCPV	DMKGRD	DMKGRF	DMKIOF	DMKOPE	DMKOPR	DMKQCQ	DMKSSP	DMKVCN
		DMKVDA									
TYP3088	000002	DMKDEF									
TYP3138	000002	DMKDIR									
TYP3148	000002	DMKDIR									
TYP3158	000002	DMKDIR									

LABEL	COUNT	REFERENCES									
TYP3203	000017	DMKCSB	DMKCSC	DMKDEF	DMKDIR	DMKIOJ	DMKIOS	DMKRSF	DMKRSP	DMKSEP	DMKVSP
		DMKVSR	DMKVSX								
TYP3210	000052	DMKCCW	DMKCFQ	DMKCKD	DMKCKF	DMKCNS	DMKCPB	DMKCQG	DMKCSP	DMKCSQ	DMKCSR
		DMKCST	DMKDIA	DMKDIR	DMKHVE	DMKIOF	DMKOPE	DMKOPR	DMKSCN	DMKSPL	DMKSSP
		DMKVDS	DMKVSI	DMKVSP	DMKVSQ	DMKVST	DMKVSU				
TYP3211	000048	DMKCSB	DMKCSC	DMKCSO	DMKDEF	DMKDIR	DMKIOF	DMKIOJ	DMKIOS	DMKRSE	DMKRSP
		DMKSEP	DMKSPL	DMKURS	DMKVDR	DMKVSP	DMKVSQ	DMKVSR	DMKVST	DMKVSX	
TYP3215	000007	DMKCFQ	DMKDIR								
TYP3262	000017	DMKCSB	DMKCSC	DMKDEF	DMKDIR	DMKIOJ	DMKIOS	DMKRSF	DMKRSP	DMKSEP	DMKVSP
		DMKVSX									
TYP3277	000099	DMKACO	DMKCCW	DMKCFM	DMKCFT	DMKCFY	DMKCKD	DMKCPV	DMKDEF	DMKDIA	DMKDIB
		DMKDIF	DMKDIR	DMKGRD	DMKGRF	DMKGRG	DMKGRI	DMKHPS	DMKHVD	DMKHVE	DMKIOT
		DMKNEA	DMKOPE	DMKOPR	DMKQCN	DMKQCQ	DMKSSP	DMKVCN	DMKVCP	DMKVCQ	DMKVCR
		DMKVCU	DMKVDA	DMKVDS							
TYP3278	000072	DMKCCW	DMKCFM	DMKCFT	DMKCFY	DMKCKD	DMKCPV	DMKDIA	DMKDIB	DMKGRD	DMKGRF
		DMKGRG	DMKGRI	DMKHPS	DMKHVD	DMKHVE	DMKIOT	DMKOPE	DMKOPR	DMKQCN	DMKQCQ
		DMKSSP	DMKVCN	DMKVCP	DMKVCQ	DMKVCR	DMKVCU	DMKVDA	DMKVDS		
TYP3279	000002	DMKCFM									
TYP3284	000022	DMKCKD	DMKCPV	DMKDID	DMKGRF	DMKGRG	DMKHVE	DMKIOT	DMKVCN	DMKVDA	
TYP3286	000005	DMKHPS	DMKHPT								
TYP3289E	000020	DMKCSB	DMKCSC	DMKDEF	DMKDID	DMKDIR	DMKIOJ	DMKIOS	DMKRSF	DMKVSP	DMKVSX
TYP3310	000013	DMKALO	DMKBIO	DMKCPJ	DMKCPW	DMKCQG	DMKDEF	DMKIOC	DMKIOJ	DMKMNT	DMKTDK
		DMKVER									
TYP3330	000094	DMKALO	DMKATS	DMKBIO	DMKCCD	DMKCCO	DMKCCS	DMKCCW	DMKCFG	DMKCFH	DMKCFQ
		DMKCFS	DMKCKH	DMKCKN	DMKCKT	DMKCKV	DMKCPJ	DMKCPP	DMKCQG	DMKCQP	DMKDAS
		DMKDDR	DMKDEF	DMKDGD	DMKDID	DMKDIR	DMKDRD	DMKDRE	DMKDSB	DMKHVD	DMKHVE
		DMKIOC	DMKIOE	DMKIOG	DMKIOJ	DMKIOS	DMKIOT	DMKLNK	DMKLOJ	DMKNLD	DMKNLE
		DMKPAG	DMKPMA	DMKSAV	DMKSEG	DMKSNC	DMKSPS	DMKSSS	DMKSST	DMKTCS	DMKTDK
		DMKUNT	DMKVDA	DMKVDG	DMKVER	DMKVRR	DMKVSC	DMKZTD			
TYP3340	000081	DMKALO	DMKBIO	DMKCCD	DMKCCO	DMKCCS	DMKCFH	DMKCFS	DMKCKH	DMKCKN	DMKCPJ
		DMKCPM	DMKCQG	DMKDAS	DMKDDR	DMKDEF	DMKDGD	DMKDIR	DMKDMP	DMKDRD	DMKDRE
		DMKDSB	DMKGIO	DMKHVE	DMKIOE	DMKIOG	DMKIOJ	DMKMNT	DMKMSW	DMKPAG	DMKPMA
		DMKSAV	DMKSPK	DMKSPS	DMKSSP	DMKTDK	DMKUNT	DMKVDE	DMKVDG	DMKVER	DMKVIO
		DMKVSC	DMKZTD								
TYP3350	000086	DMKALO	DMKATS	DMKBIO	DMKCCD	DMKCCO	DMKCCS	DMKCFG	DMKCFH	DMKCFS	DMKCKH
		DMKCKN	DMKCKT	DMKCKV	DMKCPJ	DMKCPP	DMKCQG	DMKDAS	DMKDDR	DMKDEF	DMKDGD
		DMKDIR	DMKDMP	DMKDRD	DMKDRE	DMKHVD	DMKHVE	DMKIOE	DMKIOG	DMKIOJ	DMKMSW
		DMKNLD	DMKNLE	DMKPAG	DMKPGT	DMKPMA	DMKSAV	DMKSEG	DMKSNC	DMKSPK	DMKSPS
		DMKSSP	DMKTCS	DMKTDK	DMKUNT	DMKVDG	DMKVDH	DMKVER	DMKVSC	DMKZTD	
TYP3370	000015	DMKALO	DMKBIO	DMKCPJ	DMKCPW	DMKCQG	DMKDAU	DMKDEF	DMKIOJ	DMKMNT	DMKUNT
		DMKVDG									
TYP3375	000100	DMKALO	DMKATS	DMKBIO	DMKCCD	DMKCCO	DMKCCS	DMKCFG	DMKCFH	DMKCFS	DMKCKH
		DMKCKN	DMKCKP	DMKCKT	DMKCKV	DMKCPJ	DMKCPP	DMKCQG	DMKDAD	DMKDDR	DMKDEF
		DMKDGD	DMKDIR	DMKDMP	DMKDRD	DMKDRE	DMKHVD	DMKHVE	DMKIOE	DMKIOG	DMKIOJ
		DMKIOS	DMKMNT	DMKNLD	DMKNLE	DMKPAG	DMKPGT	DMKPMA	DMKSAV	DMKSEG	DMKSNC
		DMKSPK	DMKSSP	DMKTCS	DMKTDK	DMKUNT	DMKVDG	DMKVDH	DMKVER	DMKVSC	DMKZTD
TYP3380	000116	DMKALO	DMKATS	DMKBIO	DMKCCD	DMKCCO	DMKCCS	DMKCFG	DMKCFH	DMKCKH	DMKCKN
		DMKCKP	DMKCKT	DMKCKV	DMKCPJ	DMKCPM	DMKCPP	DMKCPW	DMKCQG	DMKDAD	DMKDDR
		DMKDEF	DMKDGD	DMKDIR	DMKDMP	DMKDRD	DMKDRE	DMKHVD	DMKHVE	DMKIOE	DMKIOG
		DMKIOJ	DMKIOS	DMKMNT	DMKNLD	DMKNLE	DMKPAG	DMKPGT	DMKPMA	DMKSAV	DMKSEG
		DMKSNC	DMKSPK	DMKSPS	DMKSSP	DMKTCS	DMKTDK	DMKUNT	DMKVDE	DMKVDG	DMKVDH
		DMKVER	DMKVSC	DMKZTD							
TYP3410	000009	DMKCCO	DMKDDR	DMKIOE	DMKIOF	DMKIOJ	DMKTAQ				
TYP3411	000002	DMKDDR									
TYP3420	000012	DMKCCO	DMKDDR	DMKDMQ	DMKIOE	DMKIOF	DMKIOJ	DMKTAQ			
TYP3422	000017	DMKCCO	DMKDDR	DMKDMQ	DMKIOE	DMKIOF	DMKIOJ	DMKMCC	DMKSPT	DMKTAP	DMKTAQ
TYP3430	000019	DMKCCO	DMKDDR	DMKDMQ	DMKIOE	DMKIOF	DMKIOJ	DMKMCC	DMKSPT	DMKTAP	DMKTAQ
TYP3480	000049	DMKACS	DMKCCO	DMKCPM	DMKCPN	DMKCPS	DMKCPW	DMKDDR	DMKDMQ	DMKHVC	DMKIOE

LABEL	COUNT	REFERENCES									
		DMKIOJ	DMKIOS	DMKMCC	DMKMNT	DMKMSW	DMKSPT	DMKVDR	DMKVSI		
TYP3505	000012	DMKDEF	DMKDIR	DMKIOJ	DMKRSE	DMKVSX					
TYP3525	000004	DMKDEF	DMKDIR								
TYP3704	000003	DMKCCT	DMKVDS	DMKVIO							
TYP3705	000024	DMKACR	DMKBLD	DMKCCO	DMKCKD	DMKCPO	DMKCPW	DMKCQP	DMKNES	DMKNET	DMKNLD
		DMKNLE	DMKRNH	DMKSCN	DMKUSQ	DMKVCH	DMKVDC	DMKVDS	DMKVIO	DMKVSI	DMKWRM
TYP3800	000102	DMKCCS	DMKCKH	DMKCKS	DMKCKV	DMKCPO	DMKCPS	DMKCPW	DMKCQG	DMKCSO	DMKDEF
		DMKDID	DMKDIR	DMKIOE	DMKIOJ	DMKIOT	DMKMNT	DMKRSE	DMKRSP	DMKSCN	DMKSEP
		DMKSPL	DMKTCS	DMKTCT	DMKURS	DMKVDR	DMKVDS	DMKVSP	DMKVSQ	DMKVSR	DMKVST
		DMKVSX	DMKWRM								
TYP38003	000129	DMKCCS	DMKCKH	DMKCKS	DMKCKV	DMKCPO	DMKCPS	DMKCPW	DMKCQG	DMKCSO	DMKDEF
		DMKDEG	DMKDID	DMKDIR	DMKIOE	DMKIOJ	DMKIOT	DMKMNT	DMKRSE	DMKRSP	DMKSCN
		DMKSEP	DMKSPL	DMKTCS	DMKTCT	DMKURS	DMKVDR	DMKVDS	DMKVSP	DMKVSQ	DMKVSR
		DMKVST	DMKVSV	DMKVSX	DMKWRM						
TYP38008	000062	DMKCKH	DMKCKS	DMKCKV	DMKCPO	DMKCPS	DMKCPW	DMKCSO	DMKDID	DMKIOE	DMKIOJ
		DMKIOT	DMKMNT	DMKRSE	DMKRSP	DMKSCN	DMKSEP	DMKTCS	DMKTCT	DMKURS	DMKVDR
		DMKVDS	DMKWRM								
TYP3851	000007	DMKCCW	DMKCFQ	DMKCKD	DMKCPM	DMKDID	DMKMNT	DMKSCN			
TYP4245	000033	DMKCPB	DMKCSB	DMKCSC	DMKDEF	DMKDIR	DMKIOE	DMKIOJ	DMKIOS	DMKRSE	DMKRSF
		DMKRSP	DMKSEP	DMKVSP	DMKVSX						
TYP4248	000052	DMKCPB	DMKCSB	DMKCSC	DMKDEF	DMKDIR	DMKIOE	DMKIOF	DMKRSE	DMKRSP	DMKSEP
		DMKSPL	DMKVSP	DMKVSQ	DMKVSR	DMKVST	DMKVSX				
TYP5ACCW	000008	DMKAPU	DMKCQI	DMKRSP	DMKVSP	DMKVSX					
TYP8809	000034	DMKCFR	DMKCPS	DMKDDR	DMKDMQ	DMKDSB	DMKIOE	DMKIOJ	DMKMCC	DMKMNT	DMKSPT
		DMKTAP	DMKTAQ	DMKVDD							
UC	000170	DMKACS	DMKBSC	DMKCKD	DMKCKH	DMKCNS	DMKCPI	DMKCPM	DMKCPZ	DMKCQT	DMKDAD
		DMKDAS	DMKDAU	DMKDDR	DMKDIB	DMKDID	DMKDIR	DMKDMP	DMKDMQ	DMKDSB	DMKDSP
		DMKFMT	DMKGRF	DMKHPS	DMKHPT	DMKHVC	DMKIOE	DMKIOS	DMKIOT	DMKLDOOE	DMKMNT
		DMKMON	DMKNLD	DMKNLE	DMKOPE	DMKOPR	DMKRGA	DMKRGE	DMKRNH	DMKRSE	DMKRSP
		DMKSAD	DMKSAV	DMKSPS	DMKSSP	DMKTAP	DMKTPE	DMKTRK	DMKUNT	DMKVCA	DMKVCB
		DMKVCN	DMKVDC	DMKVDE	DMKVIO	DMKVMI	DMKVRR	DMKVRS	DMKVSI	DMKVSP	DMKVSX
UCASE	000037	DMKCFH	DMKCFM	DMKCFU	DMKCNS	DMKCNT	DMKCPJ	DMKEPS	DMKGRG	DMKGRT	DMKMSW
		DMKNLD	DMKNLE	DMKOPE	DMKRGC	DMKRNH	DMKTOD	DMKVCR	DMKVCU		
UCNTRL	000003	DMKUDU									
UCNTRLSZ	000003	DMKUDU									
UCSBBLOK	000001	DMKCSB									
UCSBFOLD	000001	DMKCSB									
UCSBNAME	000002	DMKCSB									
UCSBSIZE	000002	DMKCSB									
UCSBVER	000003	DMKCSB									
UCSCCWS	000011	DMKCSB	DMKCSC								
UCSCCW0B	000001	DMKCSB									
UCSCCW06	000001	DMKCSB									
UCSCCW07	000002	DMKCSB	DMKCSC								
UCSFCBAD	000004	DMKCSB									
UCSFCBLD	000003	DMKCSB									
UCSFCBL1	000005	DMKCSB									
UCSFFLD	000002	DMKCSB									
UCSLOAD	000001	DMKCSB									
UCSNAME	000003	DMKCSB									
UCSRDCCW	000001	DMKCSB									
UCSRDCC1	000001	DMKCSC									
UCSREGS	000002	DMKCSB									
UCURPASS	000002	DMKUDU									
UDASDDEV	000004	DMKUDU									
UDASDDIR	000003	DMKUDU									
UDASDIPL	000001	DMKUDU									
UDASDMAC	000002	DMKUDU									

LABEL	COUNT	REFERENCES									
UDEVAD	000001	DMKUDU									
UDEVCNT	000001	DMKUDU									
UDEVCODE	000001	DMKUDU									
UDEVF	000001	DMKUDU									
UDIRAD	000002	DMKUDU									
UDIRF	000005	DMKUDU									
UDISPDEV	000003	DMKUDU									
UDISPMAC	000002	DMKUDU									
UE	000079	DMKBSC	DMKCNS	DMKCPB	DMKCSB	DMKDDR	DMKDIR	DMKDMQ	DMKFMT	DMKGIO	DMKGRF
		DMKHPS	DMKHVC	DMKIOH	DMKIOT	DMKLD00E	DMKMON	DMKRGA	DMKRNH	DMKRSP	DMKRST
		DMKSAD	DMKSAV	DMKSEP	DMKSPS	DMKSSP	DMKTAP	DMKVCA	DMKVCN	DMKVIO	DMKVMI
		DMKVRS	DMKVSJ	DMKVSP	DMKVSX						
UFLAGS	000035	DMKUDU	DMKVMG								
UIPARMS	000001	DMKUDU									
UIPARMSZ	000001	DMKUDU									
UIPL	000001	DMKUDU									
UIPLOP	000001	DMKUDU									
UIPLPRMS	000001	DMKUDU									
UIUCBLOK	000009	DMKDIR	DMKIDR	DMKIUC	DMKUDR						
UIUCCHN	000002	DMKDIR	DMKUDR								
UIUCDASD	000002	DMKDIR	DMKUDR								
UIUCDISP	000004	DMKDIR	DMKUDR								
UIUCGLBL	000003	DMKDIR	DMKIDR								
UIUCLAST	000002	DMKDIR	DMKUDR								
UIUCMLIM	000003	DMKDIR	DMKIUC								
UIUCPRTY	000004	DMKDIR	DMKIUC								
UIUCRES	000002	DMKDIR	DMKUDR								
UIUCREVK	000002	DMKDIR	DMKIDR								
UIUCSIZE	000004	DMKDIR	DMKUDR								
UIUCSTAT	000010	DMKDIR	DMKIDR	DMKIUC	DMKUDR						
UIUCUSER	000002	DMKDIR	DMKUDR								
ULOCDVAD	000001	DMKUDU									
UMACAD	000011	DMKUDU									
UMACF	000001	DMKUDU									
UMDISKAD	000001	DMKUDU									
UMDISKMD	000001	DMKUDU									
UMDISKMP	000001	DMKUDU									
UMDISKRP	000002	DMKUDU									
UMDISKWP	000001	DMKUDU									
UNCHG	000004	DMKSEL									
UNEWPASS	000002	DMKUDU									
UNFIN	000005	DMKMIA									
UNLOAD	000006	DMKCCS									
UNOUPF	000003	DMKUDU									
UNREF	000006	DMKSEL									
UNSHRVM	000001	DMKCPP									
UNTICLST	000002	DMKCCW									
UOBJVMBK	000002	DMKUDU									
UOP	000005	DMKUDU									
UOPTIONS	000001	DMKUDU									
UPPINCR	000002	DMKFRE	DMKFRT								
UPRIOR	000002	DMKUDU									
UPRIVLGE	000002	DMKUDU									
URECMP	000003	DMKUDU									
URETCODE	000015	DMKUDU									
URPAGDEV	000003	DMKUDU									
URPAGDIR	000002	DMKUDU									
URPAGMAC	000002	DMKUDU									

LABEL	COUNT	REFERENCES							
URPAGXIP	000001	DMKUDU							
URSBACK	000002	DMKRSP							
URSDEV	000008	DMKCQP	DMKCSO	DMKRSP	DMKRST	DMKURS			
URSFILE	000020	DMKCQP	DMKRSP	DMKRST	DMKURS				
URSFLUSH	000002	DMKRSP							
URSHELD	000002	DMKRSP							
URSOPER	000021	DMKCSO	DMKRSP	DMKRST	DMKURS				
URSPATH	000002	DMKCQP	DMKURS						
URSREAD	000002	DMKRST							
URSREP	000002	DMKRSP							
URSSTACK	000004	DMKCQP	DMKURS						
URSSTART	000003	DMKCSO	DMKURS						
URSUSER	000004	DMKCSO	DMKURS						
USCRCPO	000001	DMKUDU							
USCRINA	000001	DMKUDU							
USCRINR	000001	DMKUDU							
USCRSTA	000001	DMKUDU							
USCRVMO	000001	DMKUDU							
USERCARD	000002	DMKACO	DMKCKF						
USERCL	000004	DMKMCC	DMKMNI	DMKMOO					
USPMSK	000003	DMKFRE							
USPSIZE	000005	DMKFRE	DMKFRT						
USPSIZE$	000002	DMKFRE	DMKFRT						
USTORAGE	000002	DMKUDU							
USVDASD	000003	DMKUDU							
UTESTMD	000020	DMKUDU							
UT3310	000002	DMKCPW	DMKMNT						
UT3370	000002	DMKCPW	DMKMNT						
UT3370M4	000002	DMKCPW	DMKMNT						
UUSERID	000002	DMKUDU							
UVMBLOK	000002	DMKUDU							
UVPAGBUF	000005	DMKUDU							
UVPAGDIR	000003	DMKUDU							
UWORK	000020	DMKUDU							
UWORK2	000003	DMKUDU							
UXIPLAD	000002	DMKUDU							
VACOK	000012	DMKACO	DMKCKF						
VACOVFR	000009	DMKACO	DMKCKF	DMKEXT	DMKMOO	DMKPRV	DMKTHI	DMKVFR	DMKVFS
VBFBLOK	000023	DMKVSP	DMKVSQ	DMKVSR	DMKVSV				
VBFBUF	000006	DMKVSP	DMKVSR						
VBFBUF1	000001	DMKVSR							
VBFBUF2	000002	DMKVSR							
VBFCCW1	000012	DMKVSP	DMKVSQ	DMKVSV					
VBFCOUNT	000004	DMKVSP	DMKVSR						
VBFDATA	000023	DMKVSR	DMKVSV						
VBFDATLF	000002	DMKVSQ							
VBFDCACT	000002	DMKVSQ							
VBFDCUSD	000003	DMKVSQ							
VBFFLAG1	000005	DMKVSQ							
VBFLGLFT	000003	DMKVSQ							
VBFRADD	000004	DMKVSP	DMKVSR						
VBFRADD1	000013	DMKVSQ	DMKVSR	DMKVSV					
VBFRADD2	000009	DMKVSQ	DMKVSR	DMKVSV					
VBFRADD8	000001	DMKVSR							
VBFTIC	000004	DMKVSQ	DMKVSV						
VBFVSQR0	000008	DMKVSQ	DMKVSR						
VBFVSQR1	000004	DMKVSR							
VBFVSQR2	000002	DMKVSR							

LABEL	COUNT	REFERENCES									
VBFVSQR3	000004	DMKVSQ									
VBFWORK	000001	DMKVSR									
VCONADDR	000006	DMKVCN									
VCONANF	000011	DMKCFM	DMKCFQ	DMKCFT	DMKGRD	DMKGRE	DMKGRG				
VCONANF2	000008	DMKCFQ	DMKGRD	DMKGRE	DMKRGB	DMKVCS					
VCONBFSZ	000015	DMKCFQ	DMKDIB	DMKDIF	DMKVCN	DMKVDR					
VCONBRK	000023	DMKCFT	DMKCFY	DMKCQU	DMKGRE	DMKGRF	DMKGRG	DMKRGA	DMKRGC	DMKVCQ	DMKVCR
		DMKVCU	DMKVDS								
VCONBUF	000031	DMKCFQ	DMKDIB	DMKDIF	DMKVCN	DMKVCV	DMKVDR				
VCONCAW	000008	DMKVCN									
VCONCBRK	000008	DMKCFM	DMKCFT	DMKCQU	DMKGRD	DMKGRE	DMKGRF				
VCONCCW	000025	DMKVCN									
VCONCCW2	000003	DMKQCO	DMKVCN								
VCONCNT	000010	DMKVCN									
VCONCNT2	000004	DMKQCO	DMKVCN								
VCONCOMD	000040	DMKVCN									
VCONCPRD	000006	DMKGRE	DMKGRF								
VCONCTL	000084	DMKACO	DMKALG	DMKCFM	DMKCFQ	DMKCFT	DMKCFU	DMKCFY	DMKCKF	DMKCQG	DMKCQU
		DMKDEF	DMKDIA	DMKDIB	DMKDIF	DMKEXT	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKHVE
		DMKMSG	DMKNEA	DMKQCN	DMKQCO	DMKREI	DMKRGA	DMKRGB	DMKRGC	DMKRGE	DMKSCN
		DMKSND	DMKTHI	DMKTRD	DMKVCN	DMKVCQ	DMKVCR	DMKVCS	DMKVCU	DMKVCV	DMKVDR
		DMKVDS									
VCONDIAG	000002	DMKCFM									
VCONDWC	000022	DMKVCN									
VCONEWA	000002	DMKVCN									
VCONEWRT	000002	DMKVCN									
VCONEXTN	000005	DMKDIA	DMKVCN	DMKVDR							
VCONFLAG	000035	DMKVCN									
VCONFLG2	000021	DMKCFU	DMKGRD	DMKMSG	DMKQCN	DMKVCN					
VCONFSOP	000001	DMKVCN									
VCONFSS	000034	DMKCFQ	DMKVCN								
VCONIDAP	000003	DMKVCN									
VCONLED	000003	DMKVCN									
VCONNCB	000017	DMKCFQ	DMKVCN	DMKVCV	DMKVDR						
VCONNICB	000021	DMKACO	DMKCFQ	DMKCKF	DMKCQG	DMKDEF	DMKDIA	DMKDIB	DMKHVE	DMKQCN	DMKQCO
		DMKSCN	DMKTRD	DMKVCN	DMKVDR	DMKVDS					
VCONNTRM	000004	DMKVCN									
VCONOBRK	000014	DMKCFT	DMKCFY	DMKCQU	DMKGRE	DMKGRF	DMKGRG	DMKRGA	DMKRGC	DMKVCQ	DMKVCR
		DMKVCU	DMKVDS								
VCONOMSG	000005	DMKCFU	DMKMSG	DMKQCN							
VCONOPT	000066	DMKCFM	DMKCFQ	DMKCFT	DMKCQU	DMKDIA	DMKEXT	DMKGRD	DMKGRE	DMKGRF	DMKGRG
		DMKRGA	DMKRGB	DMKRGC	DMKVCN	DMKVCS	DMKVCU	DMKVCV	DMKVDS		
VCONPLF	000016	DMKVCN	DMKVCV	DMKVDR							
VCONPLST	000008	DMKVCN									
VCONPLSZ	000003	DMKVCN									
VCONPPA1	000002	DMKVCN									
VCONRBUF	000022	DMKALG	DMKCFQ	DMKCFT	DMKGRG	DMKRGC	DMKSND	DMKVCN	DMKVCU	DMKVDR	
VCONRBYT	000003	DMKRGE	DMKVCN								
VCONRCNT	000009	DMKALG	DMKGRG	DMKRGC	DMKSND	DMKVCN	DMKVCU				
VCONRD	000012	DMKGRE	DMKVCN								
VCONRDEV	000022	DMKACO	DMKCFQ	DMKCKF	DMKCQG	DMKDEF	DMKDIA	DMKDIB	DMKHVE	DMKNEA	DMKQCN
		DMKQCO	DMKSCN	DMKTHI	DMKVCN	DMKVCU	DMKVDR	DMKVDS			
VCONRDSZ	000011	DMKALG	DMKCFQ	DMKCFT	DMKGRG	DMKRGC	DMKSND	DMKVCN	DMKVCU	DMKVDR	
VCONREMD	000016	DMKCFQ	DMKCQG	DMKDIA	DMKSCN	DMKVCN	DMKVDR	DMKVDS			
VCONREMF	000031	DMKCFQ	DMKCQG	DMKDIA	DMKSCN	DMKVCN	DMKVCV	DMKVDR	DMKVDS		
VCONREXW	000003	DMKGRD	DMKVCN								
VCONRFLD	000003	DMKRGE	DMKVCN								
VCONRIND	000010	DMKRGE	DMKVCN								

LABEL	COUNT	REFERENCES									
VCONRINX	000007	DMKRGA	DMKVCN	DMKVCR							
VCONRLN	000003	DMKVCN									
VCONRMAX	000001	DMKVCN									
VCONRMCT	000004	DMKVCN	DMKVCR								
VCONRMOD	000002	DMKVCN									
VCONRMSZ	000006	DMKDIA	DMKDIB	DMKDIF	DMKVDR						
VCONSCOL	000002	DMKGRE									
VCONSCRN	000015	DMKCFM	DMKCFT	DMKCQU	DMKEXT	DMKGRD	DMKGRE	DMKGRF	DMKVCN		
VCONSHI	000002	DMKGRE									
VCONSIZE	000003	DMKVDR	DMKVDS								
VCONSKIP	000010	DMKVCN	DMKVCV								
VCONSPSS	000002	DMKGRE									
VCONSRM	000010	DMKGRE									
VCONWA	000003	DMKGRE									
VCONWBFS	000004	DMKVCN									
VCONWBSZ	000005	DMKCFQ	DMKVCN	DMKVDR							
VCONWBUF	000013	DMKCFQ	DMKVCN	DMKVDR							
VCONWCNT	000003	DMKVCN									
VCONWRRD	000003	DMKVCN									
VCONWRRM	000003	DMKVCN									
VCONWRT	000003	DMKVCN									
VCONWSF	000003	DMKVCN									
VCON3270	000028	DMKCFM	DMKCFT	DMKCQU	DMKDIA	DMKEXT	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKRGA
		DMKRGB	DMKRGC	DMKVCN	DMKVCS	DMKVCU	DMKVCV	DMKVDS			
VECAVAIL	000028	DMKAPI	DMKCPO	DMKMCH	DMKMHV	DMKPRG	DMKPRV	DMKVFC	DMKVFD	DMKVFE	DMKVFR
VECF	000021	DMKAPI	DMKDSP	DMKEXT	DMKFPS	DMKMCH	DMKPRG	DMKPRV	DMKVFC	DMKVFR	
VECINST	000006	DMKMOO	DMKVFC								
VECOPVF	000016	DMKACO	DMKAPI	DMKCKF	DMKCPP	DMKMCH	DMKMOO	DMKPRG	DMKTHI	DMKVFC	DMKVFD
		DMKVFE	DMKVFR								
VECSAOK	000011	DMKCPO	DMKDMP	DMKDSP	DMKPRG	DMKVFC	DMKVFD	DMKVFE	DMKVFR		
VECSTAT	000059	DMKACO	DMKAPI	DMKCKF	DMKCPO	DMKCPP	DMKDMP	DMKDSP	DMKMCH	DMKMHV	DMKMOO
		DMKPRG	DMKPRV	DMKTHI	DMKVFC	DMKVFD	DMKVFE	DMKVFR			
VECUSER	000046	DMKACO	DMKAPI	DMKCFY	DMKCKF	DMKCPO	DMKDMP	DMKDSP	DMKFPS	DMKMCH	DMKPRG
		DMKPRV	DMKTHI	DMKVFC	DMKVFD	DMKVFE	DMKVFR	DMKVFS			
VECVARY	000003	DMKVFC									
VERLEN	000001	DMKFCB									
VERNULL	000001	DMKWRM									
VERSP3	000001	DMKWRM									
VFAULT	000020	DMKBIO	DMKDGD	DMKPRG	DMKPTR	DMKSWA	DMKVAT				
VGPTR	000007	DMKCPP	DMKTRP	DMKTRU	DMKVMG						
VHD	000003	DMKMPO	DMKQVM	DMKSPM							
VIRCOMND	000213	DMKCCD	DMKCCF	DMKCCO	DMKCCS	DMKCCT	DMKCCW				
VIRFLAG	000037	DMKCCD	DMKCCO	DMKCCS	DMKCCT	DMKCCW					
VIRTDISA	000002	DMKCCT	DMKCCW								
VIRTENAB	000001	DMKCCT									
VIRTUAL	000056	DMKBIO	DMKCCO	DMKCCW	DMKCFQ	DMKCPW	DMKCQP	DMKDEF	DMKDEI	DMKDGD	DMKDID
		DMKDIR	DMKDSB	DMKIOS	DMKIOT	DMKLNK	DMKLOJ	DMKMNT	DMKSSS	DMKVDA	DMKVSC
VMABLOK	000006	DMKATS	DMKCFF	DMKPGS	DMKVMA						
VMACCOUN	000002	DMKHVD									
VMACOUNT	000009	DMKHVD									
VMAFF	000071	DMKWAI									
VMAFPNT	000010	DMKATS	DMKCFF	DMKPGS	DMKVMA						
VMAIP	000013	DMKHVD									
VMAIP2	000007	DMKHVD									
VMANAME	000003	DMKATS	DMKCFF	DMKPGS							
VMASCCPD	000006	DMKHVD									
VMASHRBK	000003	DMKCFF	DMKVMA								
VMASIZE	000004	DMKATS	DMKCFF	DMKPGS							

LABEL	COUNT	REFERENCES									
VMBLOK	001904	DMKACO	DMKACR	DMKALG	DMKAPI	DMKAPT	DMKAPU	DMKAPV	DMKAPX	DMKAPY	DMKATS
		DMKBIO	DMKBLD	DMKCAC	DMKCAO	DMKCCD	DMKCCF	DMKCCH	DMKCCO	DMKCCS	DMKCCT
		DMKCCW	DMKCDB	DMKCDM	DMKCDS	DMKCFC	DMKCFD	DMKCFF	DMKCFG	DMKCFH	DMKCFJ
		DMKCFM	DMKCFO	DMKCFP	DMKCFQ	DMKCFR	DMKCFS	DMKCFT	DMKCFU	DMKCFV	DMKCFW
		DMKCFY	DMKCKD	DMKCKF	DMKCKH	DMKCKM	DMKCKN	DMKCKP	DMKCKR	DMKCKS	DMKCKT
		DMKCKV	DMKCKW	DMKCLK	DMKCNS	DMKCNT	DMKCPB	DMKCPI	DMKCPJ	DMKCPM	DMKCPN
		DMKCPO	DMKCPP	DMKCPS	DMKCPT	DMKCPU	DMKCPV	DMKCPW	DMKCPX	DMKCPY	DMKCPZ
		DMKCQC	DMKCQG	DMKCQH	DMKCQI	DMKCQP	DMKCQQ	DMKCQR	DMKCQS	DMKCQT	DMKCQU
		DMKCQY	DMKCRM	DMKCSB	DMKCSC	DMKCSF	DMKCSO	DMKCSP	DMKCSQ	DMKCSR	DMKCST
		DMKCSU	DMKCSV	DMKCSW	DMKCSX	DMKCSY	DMKDAD	DMKDAS	DMKDAU	DMKDEF	DMKDEG
		DMKDEI	DMKDEX	DMKDGD	DMKDGF	DMKDIA	DMKDIB	DMKDID	DMKDIF	DMKDMP	DMKDMQ
		DMKDRD	DMKDRE	DMKDSB	DMKDSP	DMKENT	DMKEPS	DMKERM	DMKEXT	DMKFPS	DMKFRE
		DMKFRT	DMKGIO	DMKGRA	DMKGRC	DMKGRD	DMKGRE	DMKGRF	DMKGRG	DMKGRI	DMKGRT
		DMKHPS	DMKHPT	DMKHPU	DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKIDR	DMKIDU	DMKIOE
		DMKIOF	DMKIOG	DMKIOH	DMKIOJ	DMKIOQ	DMKIOS	DMKIOT	DMKISM	DMKIUA	DMKIUB
		DMKIUC	DMKIUE	DMKIUG	DMKIUJ	DMKIUL	DMKIUN	DMKIUP	DMKIUS	DMKJRL	DMKLNK
		DMKLNM	DMKLOG	DMKLOH	DMKLOJ	DMKLOK	DMKLOM	DMKMCC	DMKMCD	DMKMCH	DMKMCI
		DMKMCT	DMKMHC	DMKMHV	DMKMIA	DMKMID	DMKMNI	DMKMNJ	DMKMON	DMKMOO	DMKMPO
		DMKMSG	DMKMSW	DMKNEA	DMKNES	DMKNET	DMKNLD	DMKNLE	DMKOPE	DMKPAG	DMKPAH
		DMKPEI	DMKPEL	DMKPEN	DMKPEQ	DMKPER	DMKPET	DMKPGM	DMKPGS	DMKPGT	DMKPGU
		DMKPMA	DMKPRG	DMKPRV	DMKPRW	DMKPSA	DMKPTR	DMKPTS	DMKPTT	DMKPXA	DMKPXB
		DMKQCN	DMKQCO	DMKQCP	DMKQCQ	DMKQVM	DMKREI	DMKRET	DMKRGA	DMKRGB	DMKRGC
		DMKRGD	DMKRGE	DMKRNH	DMKRPA	DMKRPD	DMKRSE	DMKRSF	DMKRSP	DMKRSQ	DMKRST
		DMKSAV	DMKSCH	DMKSCN	DMKSCO	DMKSEG	DMKSEL	DMKSEP	DMKSFB	DMKSNC	DMKSND
		DMKSPK	DMKSPL	DMKSPM	DMKSPR	DMKSPS	DMKSPT	DMKSRM	DMKSSS	DMKSST	DMKSSU
		DMKSSV	DMKSTA	DMKSTK	DMKSTP	DMKSTR	DMKSVC	DMKSVD	DMKSWA	DMKSWM	DMKSYM
		DMKTCS	DMKTCT	DMKTEE	DMKTEF	DMKTES	DMKTHI	DMKTMR	DMKTOD	DMKTPE	DMKTRA
		DMKTRC	DMKTRD	DMKTRK	DMKTRP	DMKTRQ	DMKTRT	DMKTRU	DMKTRX	DMKTTX	DMKTTY
		DMKTTZ	DMKUDR	DMKUDU	DMKUNT	DMKURS	DMKUSP	DMKUSQ	DMKVAT	DMKVAU	DMKVBM
		DMKVCA	DMKVCB	DMKVCH	DMKVCN	DMKVCP	DMKVCQ	DMKVCR	DMKVCS	DMKVCT	DMKVCU
		DMKVCV	DMKVCW	DMKVCX	DMKVDA	DMKVDB	DMKVDC	DMKVDD	DMKVDE	DMKVDF	DMKVDG
		DMKVDR	DMKVDS	DMKVDT	DMKVER	DMKVFC	DMKVFD	DMKVFE	DMKVFR	DMKVFS	DMKVIO
		DMKVMA	DMKVMC	DMKVMD	DMKVME	DMKVMG	DMKVMI	DMKVRR	DMKVRS	DMKVSC	DMKVSD
		DMKVSE	DMKVSG	DMKVSI	DMKVSJ	DMKVSP	DMKVSQ	DMKVSR	DMKVST	DMKVSU	DMKVSW
		DMKVSX	DMKWAI	DMKWRM	DMKWRN	DMKXAB	DMKXAD				
VMBTRL	000018	DMKWAI									
VMCF	000044	DMKHVC	DMKVCP	DMKVCU							
VMCFREAD	000014	DMKVCW									
VMCFWAIT	000070	DMKHVC	DMKVCP	DMKVCU							
VMCLANY	000034	DMKHVC									
VMCLASSA	000005	DMKHVC									
VMCLASSB	000007	DMKHVC									
VMCLASSC	000010	DMKHVC									
VMCLASSD	000003	DMKHVC									
VMCLASSE	000008	DMKHVC									
VMCLASSF	000005	DMKHVC									
VMCLNULL	000009	DMKHVC									
VMCMDLEV	000019	DMKHVC									
VMCOMND	000015	DMKHVC	DMKVCW								
VMCOMSRV	000004	DMKIUS									
VMCONBUF	000006	DMKHVC									
VMCONLEN	000004	DMKHVC									
VMCPSV76	000003	DMKHVC									
VMCPVIRT	000003	DMKHVC									
VMCPWAIT	000013	DMKWAI									
VMCXSTAT	000068	DMKHVC									
VMDEFSTK	000017	DMKWAI									
VMDISC	000048	DMKVCP									

LABEL	COUNT	REFERENCES									
VMDSTAT	000113	DMKWAI									
VMDVSTRT	000471	DMKHVC	DMKVCS	DMKVCU							
VMECEXT	000128	DMKHVC									
VMEPRIOR	000027	DMKWAI									
VMESTAT	000197	DMKHVC	DMKHVD	DMKIUS							
VMEXTCM	000101	DMKHVC	DMKHVD	DMKIUS							
VMEXWAIT	000096	DMKHVC	DMKHVD	DMKVCS							
VMFSTAT	000102	DMKHVC									
VMFTRL	000030	DMKWAI									
VMGENIO	000031	DMKCFQ	DMKGRC	DMKGRD	DMKGRG	DMKLOJ	DMKQCN	DMKRGA	DMKRGB	DMKRGC	DMKSND
		DMKVCN	DMKVCR	DMKVCS							
VMGPRS	000165	DMKHVC	DMKHVD								
VMGRFTAB	000021	DMKVCU									
VMIDLE	000032	DMKIUS									
VMIHIST	000012	DMKWAI									
VMINST	000151	DMKHVC									
VMIOWAIT	000041	DMKHVC									
VMIPLDEV	000010	DMKHVD									
VMISTAT	000035	DMKIUS									
VMIUCV	000020	DMKIUS									
VMIUCVWT	000006	DMKIUS									
VMIUSTAT	000049	DMKIUS									
VMJSTAT	000017	DMKHVC									
VMKILL	000054	DMKVCP									
VMLDCTRS	000009	DMKHPU									
VMLKHIST	000013	DMKWAI									
VMLOCK	000063	DMKWAI									
VMLOGOFF	000102	DMKVCP	DMKVCS	DMKVCW							
VMLOGON	000077	DMKVCS	DMKVCU	DMKVCW							
VMMCODE	000016	DMKHVC									
VMMCPENV	000018	DMKVCU									
VMMHLITE	000011	DMKVCU									
VMMLEVEL	000073	DMKHVC	DMKVCU								
VMMLINED	000013	DMKVCU									
VMMLVL2	000091	DMKHVC	DMKVCU								
VMMTEXT	000019	DMKHVC									
VMNOCPRD	000004	DMKHVC									
VMNOECPS	000022	DMKHVD									
VMNOFLU	000010	DMKIUS									
VMNPWOCL	000004	DMKHVC									
VMOSTAT	000244	DMKHVC	DMKURS	DMKVCP	DMKVCU	DMKWAI					
VMPATH	000035	DMKWAI									
VMPA2APL	000007	DMKHVD	DMKVCU								
VMPEND	000087	DMKWAI									
VMPFUNC	000013	DMKVCU									
VMPROT	000037	DMKVCP	DMKVCU								
VMPSTAT	000139	DMKHVC	DMKHVD								
VMPSW	000543	DMKHPU	DMKHVC	DMKHVD	DMKIUS						
VMPSWDCT	000014	DMKHVD									
VMPXINT	000046	DMKHPU	DMKVCU								
VMQSTAT	000049	DMKHVD	DMKVCU	DMKVCW							
VMRLPROC	000011	DMKWAI									
VMRSTAT	000427	DMKHVC	DMKHVD	DMKIUS	DMKVCP	DMKVCS	DMKVCU	DMKVCW	DMKWAI		
VMRUN	000029	DMKWAI									
VMSEG	000519	DMKHPU	DMKHVC	DMKHVD	DMKSPK						
VMSHR	000029	DMKWAI									
VMSHRPRC	000008	DMKWAI									
VMSIZE	000075	DMKBIO	DMKBLD	DMKCCW	DMKCDB	DMKCDM	DMKCDS	DMKCFD	DMKCFF	DMKCFG	DMKCFH

LABEL	COUNT	REFERENCES									
		DMKCFP	DMKCFS	DMKCKM	DMKCKT	DMKCPY	DMKDEG	DMKDSP	DMKFPS	DMKHVD	DMKIUE
		DMKIUL	DMKLOH	DMKMHC	DMKPGS	DMKPRV	DMKPRW	DMKQVM	DMKSEG	DMKTRX	DMKUSQ
		DMKVAT	DMKVBM	DMKVMD	DMKVME	DMKVSR	DMKWRM	DMKXAB			
VMSLEEP	000018	DMKHVC									
VMSPMFLG	000029	DMKHVC									
VMSTKCPU	000023	DMKWAI									
VMSTKLST	000028	DMKURS									
VMSTKO	000017	DMKURS									
VMSTOR	000050	DMKHVC									
VMSVSTAT	000047	DMKHVD									
VMSYSOP	000032	DMKURS									
VMTDIAG8	000004	DMKHVC									
VMTERM	000121	DMKHVD	DMKVCU								
VMTRCTL	000094	DMKHVC									
VMTRMID	000036	DMKHVD									
VMTRPRV	000010	DMKHVC									
VMTTIME	000027	DMKHVC									
VMUSER	000381	DMKHVD	DMKVCW								
VMVIRCF	000021	DMKHVC									
VMVPOREL	000015	DMKHVC									
VMVSPACE	000008	DMKHPU									
VMVTERM	000074	DMKVCS	DMKVCU								
VMV370R	000100	DMKHVC	DMKHVD								
VMXUNSTK	000013	DMKWAI									
VPMAAVAI	000002	DMKCFG	DMKPMA								
VRALOC	000003	DMKBLD	DMKDEG	DMKLOH							
VRSVUID	000006	DMKCPI	DMKCPY	DMKLOH							
VSMPTR	000019	DMKCPP	DMKCQS	DMKCQT	DMKVCT	DMKVCV	DMKVCW	DMKVCX			
VSPBIGBF	000011	DMKVSP	DMKVSQ	DMKVSR							
VSPBUFBK	000043	DMKVSP	DMKVSQ	DMKVSR	DMKVST	DMKVSV	DMKVSW	DMKVSX			
VSPBUFSZ	000009	DMKVSP	DMKVSQ	DMKVST	DMKVSW	DMKVSX					
VSPCAW	000012	DMKDRD	DMKVSP	DMKVSR	DMKVSW	DMKVSX					
VSPCCW	000213	DMKDRD	DMKVSP	DMKVSQ	DMKVSR	DMKVST	DMKVSV	DMKVSX			
VSPDCFOP	000003	DMKVSP									
VSPDPAGE	000031	DMKDRD	DMKVSP	DMKVSQ	DMKVSX						
VSPERR	000008	DMKVSP	DMKVSQ								
VSPFLAG1	000032	DMKVSP	DMKVSQ	DMKVSR							
VSPIDACT	000007	DMKVSP	DMKVSR	DMKVSX							
VSPIDAL	000001	DMKVSR									
VSPIDASW	000009	DMKVSP	DMKVSR	DMKVSX							
VSPIDAW2	000003	DMKVSR									
VSPLCTL	000020	DMKCKF	DMKCSQ	DMKCSR	DMKDRD	DMKHVF	DMKSPL	DMKVSP	DMKVSQ	DMKVSR	DMKVST
		DMKVSU	DMKVSV	DMKVSW	DMKVSX						
VSPMISC	000009	DMKVSP	DMKVSQ	DMKVST	DMKVSW						
VSPMISC2	000003	DMKVSQ									
VSPNEXT	000008	DMKVSQ	DMKVSX								
VSPRECNO	000002	DMKVSX									
VSPSFBLK	000031	DMKCKF	DMKCSQ	DMKCSR	DMKDRD	DMKHVF	DMKSPL	DMKVSP	DMKVSQ	DMKVST	DMKVSU
		DMKVSW	DMKVSX								
VSPSIZE	000009	DMKDRD	DMKSPL	DMKVSP	DMKVSQ	DMKVST	DMKVSW				
VSPVPAGE	000031	DMKSPL	DMKVSP	DMKVSQ	DMKVST	DMKVSW	DMKVSX				
VSPVPG2	000004	DMKVSP	DMKVSQ								
VSP5ACCW	000010	DMKVSP	DMKVSQ	DMKVSR							
WAIT	000057	DMKAPI	DMKBLD	DMKCDS	DMKCFM	DMKCFP	DMKCPI	DMKDDR	DMKDMP	DMKDMQ	DMKDSP
		DMKEXT	DMKIOG	DMKIUA	DMKIUN	DMKIUS	DMKLOH	DMKMPO	DMKOVR	DMKPMA	DMKQVM
		DMKSAD	DMKSVD	DMKTRA	DMKTRC	DMKTRR	DMKVBM				
WAITEND	000008	DMKDSP	DMKEXT	DMKIOT	DMKMCH	DMKMPO	DMKWAI				
WAITSTRT	000005	DMKCLK	DMKDSP								

LABEL	COUNT	REFERENCES								
WARNGRPT	000001	DMKCPJ								
WCCALRM	000005	DMKGRD								
WCCO	000001	DMKGRE								
WCC3	000005	DMKGRD	DMKGRF							
WCKDNXTK	000002	DMKCCD	DMKCCO							
WR	000010	DMKGRD	DMKRGD							
WRITBRK	000001	DMKRNH								
WRITE	000025	DMKBSC	DMKCQS	DMKCQY	DMKDAU	DMKDGD	DMKDIR	DMKLD00E	DMKLNK	DMKVCN	
WRITEOP	000010	DMKCCW								
WRITEOT	000004	DMKRNH								
WRITNRM	000007	DMKRNH								
WRTREAD	000001	DMKVCN								
WRTSFLD	000002	DMKVCN								
WRTUKD	000002	DMKCCD	DMKCCO							
WRTUPD	000003	DMKCCD	DMKCCO							
WSF	000005	DMKGRE	DMKGRF							
XCAPR	000004	DMKEXT	DMKSCH							
XCDISP	000013	DMKDSP	DMKEXT	DMKIOS	DMKLOK	DMKPAG	DMKSCH	DMKSTK	DMKSVC	DMKVSJ	
XCMASK	000012	DMKAPI	DMKDSP	DMKFRE	DMKIOT	DMKPMA					
XCPEND	000034	DMKCPP	DMKDSP	DMKEXT	DMKIOS	DMKLOK	DMKMCT	DMKPAG	DMKSCH	DMKSTK	DMKSVC
		DMKVSJ								
XCRES	000005	DMKEXT	DMKSCH							
XCWAK	000002	DMKEXT	DMKSCH							
XFERSENS	000002	DMKCCS	DMKCCW							
XFF	000002	DMKCFY								
XKEYASST	000011	DMKAPI	DMKCPI	DMKDSP	DMKPRW	DMKSPM					
XKEYMODE	000048	DMKCCW	DMKCDB	DMKCDM	DMKCFH	DMKCFY	DMKCKM	DMKCPI	DMKCPU	DMKCPY	DMKDGD
		DMKDSP	DMKFPS	DMKLOJ	DMKMCH	DMKPRV	DMKPRW	DMKPSA	DMKPTR	DMKPTS	DMKQVM
		DMKSEL	DMKSPM	DMKSTA	DMKSTR	DMKVAT	DMKVAU	DMKVME			
XMIGACT	000007	DMKPGM	DMKSTP							
XMIGSWT	000003	DMKSTP	DMKSTR							
XOBRCCW1	000002	DMKRSF								
XOBRCCW2	000001	DMKRSF								
XOBRCCW3	000001	DMKRSF								
XOBRCCW4	000001	DMKRSF								
XOBREXT	000003	DMKRSF								
XOBRFCB	000003	DMKIOJ	DMKRSF							
XOBRFLAG	000008	DMKIOE	DMKIOJ	DMKRSF							
XOBRMIS1	000002	DMKRSF								
XOBRMIS2	000002	DMKRSF								
XOBRRT1	000006	DMKRSF								
XOBRRT2	000006	DMKRSF								
XOBRRT3	000006	DMKRSF								
XOBRRT4	000003	DMKRSF								
XOBRRT5	000007	DMKRSF								
XOBRRT6	000007	DMKRSF								
XOBRSIZE	000002	DMKRSF								
XOBRSTAT	000012	DMKRSF								
XOBRT1	000006	DMKIOE	DMKIOJ	DMKRSF							
XOBRT2	000001	DMKRSF								
XOBRT3	000004	DMKIOE	DMKIOJ	DMKRSF							
XOBR010	000003	DMKIOJ	DMKRSF							
XOBR150	000004	DMKIOJ	DMKRSF							
XOBR512	000008	DMKIOJ	DMKRSF							
XPAGNUM	000162	DMKAPT	DMKATS	DMKBIO	DMKCCO	DMKCCW	DMKCDB	DMKCDM	DMKCDS	DMKCFG	DMKCFH
		DMKCFS	DMKCKV	DMKCPJ	DMKCPP	DMKCPS	DMKCQY	DMKCSC	DMKCSO	DMKDGD	DMKDRD
		DMKDRE	DMKFPS	DMKFRT	DMKHPS	DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKISM	DMKIUA
		DMKIUC	DMKIUP	DMKMCH	DMKMHV	DMKMNI	DMKMPO	DMKNLD	DMKPEI	DMKPER	DMKPMA

LABEL	COUNT	REFERENCES									
		DMKPRV	DMKPRW	DMKPTR	DMKPTS	DMKQCN	DMKQCO	DMKRPA	DMKRSP	DMKSFB	DMKSNC
		DMKSTA	DMKTCS	DMKTRX	DMKUNT	DMKVAT	DMKVAU	DMKVIO	DMKVMC	DMKVMD	DMKVME
		DMKVSR	DMKVSV	DMKWRM	DMKXAB						
XPGFULL	000006	DMKPGM	DMKPGT	DMKSTP							
XPPFULL	000006	DMKPGM	DMKPGT	DMKSTP							
XRIGHT16	000126	DMKCCD	DMKCCF	DMKCCO	DMKCCS	DMKCCW	DMKCDM	DMKCFF	DMKCFG	DMKCFS	DMKDAD
		DMKDAS	DMKDAU	DMKDSP	DMKERM	DMKFPS	DMKGRA	DMKGRC	DMKGRD	DMKGRE	DMKGRG
		DMKGRI	DMKIUE	DMKIUL	DMKIUS	DMKLOH	DMKMSG	DMKOPR	DMKPEI	DMKPER	DMKPGS
		DMKQCN	DMKQCO	DMKQCQ	DMKRGA	DMKRGB	DMKRGC	DMKRGD	DMKRGE	DMKRNH	DMKSCH
		DMKSEV	DMKSIX	DMKSSP	DMKSTP	DMKTAP	DMKTRA	DMKTRK	DMKVCN	DMKVCR	DMKVCX
		DMKVRS	DMKVSC								
XRIGHT24	000027	DMKCCW	DMKCFP	DMKDSP	DMKHVD	DMKPEI	DMKPEL	DMKPEQ	DMKPER	DMKRGA	DMKRGD
		DMKSTR	DMKSVD	DMKVCR							
XSWFULL	000006	DMKPGM	DMKPGT	DMKSTP							
XTNDCR2	000023	DMKDSP	DMKEXT	DMKFRE							
XTNDLOCK	000035	DMKDSP	DMKFRE	DMKFRT	DMKGRF	DMKIOQ	DMKIOS	DMKLOK	DMKPAG	DMKPTR	DMKPTT
		DMKRGC	DMKSCH	DMKSEL	DMKVIO	DMKWAI					
X2048BND	000024	DMKCCW	DMKCDB	DMKCDM	DMKCSW	DMKDAD	DMKDAS	DMKDRD	DMKDRE	DMKHVD	DMKHVF
		DMKPSA	DMKTRD	DMKUNT	DMKVDS	DMKXAB					
X40FFS	000024	DMKCDB	DMKCDM	DMKCFG	DMKCFT	DMKCPB	DMKCPS	DMKCPT	DMKDEF	DMKDSP	DMKMCD
		DMKPMA	DMKQVM	DMKSCH	DMKVCH	DMKVDE	DMKVFD	DMKVFE	DMKVMD	DMKVRR	DMKVSP
		DMKVST									
Y0	000017	DMKDMP	DMKDSP	DMKEXT	DMKIOT	DMKMCH	DMKMPO	DMKPMA	DMKPRG	DMKQVM	DMKSVD
Y2	000017	DMKDMP	DMKDSP	DMKEXT	DMKIOT	DMKMCH	DMKMPO	DMKPMA	DMKPRG	DMKQVM	DMKSVD
Y4	000017	DMKDMP	DMKDSP	DMKEXT	DMKIOT	DMKMCH	DMKMPO	DMKPMA	DMKPRG	DMKQVM	DMKSVD
Y6	000017	DMKDMP	DMKDSP	DMKEXT	DMKIOT	DMKMCH	DMKMPO	DMKPMA	DMKPRG	DMKQVM	DMKSVD
ZERO	000115	DMKCFC	DMKCKF	DMKCKH	DMKFMT	DMKIOE	DMKIOF	DMKIOG	DMKIOJ	DMKIOS	DMKMCC
		DMKMCD	DMKMCH	DMKSSS							
ZEROES	000650	DMKACO	DMKACR	DMKAPI	DMKAPU	DMKAPV	DMKAPX	DMKATS	DMKBIO	DMKBLD	DMKBSC
		DMKCAC	DMKCCD	DMKCCF	DMKCCH	DMKCCO	DMKCCS	DMKCCT	DMKCCW	DMKCDB	DMKCDM
		DMKCDS	DMKCFD	DMKCFG	DMKCFH	DMKCFM	DMKCFO	DMKCFP	DMKCFQ	DMKCFR	DMKCFS
		DMKCFT	DMKCFU	DMKCFV	DMKCFW	DMKCFY	DMKCKD	DMKCKF	DMKCKM	DMKCKR	DMKCKS
		DMKCKV	DMKCLK	DMKCNS	DMKCPB	DMKCPI	DMKCPN	DMKCPO	DMKCPP	DMKCPS	DMKCPU
		DMKCPV	DMKCPW	DMKCPY	DMKCQC	DMKCQG	DMKCQH	DMKCQI	DMKCQQ	DMKCQY	DMKCSC
		DMKCSO	DMKCSQ	DMKCSR	DMKCST	DMKCSV	DMKCSW	DMKCSX	DMKCSY	DMKDAD	DMKDAS
		DMKDAU	DMKDEF	DMKDGD	DMKDGF	DMKDIB	DMKDIF	DMKDMP	DMKDSB	DMKDSP	DMKEXT
		DMKFPS	DMKHVC	DMKHVD	DMKHVE	DMKHVF	DMKIDR	DMKIOE	DMKIOJ	DMKIOS	DMKIOT
		DMKIUA	DMKIUC	DMKIUE	DMKIUJ	DMKIUL	DMKIUP	DMKIUS	DMKJRL	DMKLNK	DMKLOH
		DMKLOJ	DMKMCC	DMKMCD	DMKMCH	DMKMCT	DMKMHC	DMKMIA	DMKMID	DMKMNI	DMKMNL
		DMKMNT	DMKMON	DMKMOO	DMKMSW	DMKNEA	DMKNET	DMKNLD	DMKOPE	DMKPEI	DMKPEL
		DMKPEN	DMKPER	DMKPET	DMKPGM	DMKPGS	DMKPGT	DMKPMA	DMKPRG	DMKPRV	DMKPRW
		DMKPSA	DMKPTR	DMKQVM	DMKREI	DMKRET	DMKRGA	DMKRGB	DMKRNH	DMKRPA	DMKRSE
		DMKRSF	DMKRSP	DMKSCH	DMKSCO	DMKSEP	DMKSFB	DMKSNC	DMKSPL	DMKSPM	DMKSPS
		DMKSPT	DMKSRM	DMKSSS	DMKSVC	DMKSVD	DMKTAP	DMKTAQ	DMKTCS	DMKTCT	DMKTMR
		DMKTOD	DMKTPE	DMKTRC	DMKTRD	DMKTRK	DMKTRP	DMKTRT	DMKTRU	DMKTRX	DMKUDR
		DMKUDU	DMKUNT	DMKURS	DMKUSQ	DMKVAT	DMKVCA	DMKVCB	DMKVCH	DMKVCN	DMKVCQ
		DMKVCR	DMKVCS	DMKVCT	DMKVCV	DMKVCW	DMKVDC	DMKVDE	DMKVDR	DMKVDS	DMKVFC
		DMKVFD	DMKVFE	DMKVFR	DMKVIO	DMKVMC	DMKVME	DMKVMG	DMKVRS	DMKVSD	DMKVSI
		DMKVSJ	DMKVSP	DMKVSR	DMKVST	DMKVSV	DMKWRM	DMKWRN	DMKXAD	DMKZTD	

CP Diagnostic Aids

This section contains the following information:

- Entry Points for CP Commands
- Function Codes for DIAGNOSE Instructions.

Entry Points for CP Commands

The following is a list of CP commands and the modules that gain control to perform their functions:

Command	Entry Point(s)
ACNT	DMKCPVAC
ADSTOP	DMKCFDAD
ATTACH	(See footnote 1.)
ATTN	DMKCFJRQ
AUTOLOG	DMKALGON
BACKSPAC	DMKCSFBS
BEGIN	DMKCFJBE
CHANGE	DMKCSUCS DMKCSUCG
CLOSE	DMKCSQCL
COMMANDS	DMKCQSQC
COUPLE	DMKDIBCP
CP	DMKCFM
CPTRAP	DMKTRPST
DCP	DMKCDBDC
DEFINE	DMKDEFDS DMKDEFDG
DETACH	(See footnote 1.)
DIAL	DMKDIAL
DISABLE	DMKCPVDS
DISCONN	DMKUSODS
DISPLAY	DMKCDBDI
DMCP	DMKCDMDM
DRAIN	DMKCSODR
DUMP	DMKCDMDU
ECHO	DMKMSGEC
ENABLE	DMKCPVEN
EXTERNAL	DMKCPBEX
FLUSH	DMKCSFFL
FORCE	DMKUSOFL
FREE	DMKCSQFR
HALT	DMKCPSH
HOLD	DMKCSQH
INDICATE	(See footnote 1.)
IPL	DMKCFGIP
LINK	DMKLNKIN

Command	Entry Point(s)
LOADBUF	DMKCSBLD
LOADVFCB	DMKCSBVL
LOCATE	DMKCFDLF
LOCK	DMKCPVLK
LOGOFF	DMKUSOLG
LOGON	DMKLOGON
MESSAGE	DMKMSGMS
MIGRATE	DMKPGMUS
MONITOR	DMKMCCCL
MSGNOH	DMKMSGNH
NETWORK	(See footnote 1.)
NOTREADY	DMKCPBNR
ORDER	DMKCSVOS DMKCSVOG
PER	DMKPEINT
PURGE	DMKCSVPS DMKCSVPG
QUERY	(See footnote 1.)
QVM	DMKQVMEP
READY	DMKCPBRY
REPEAT	DMKCSFRP
REQUEST	DMKCFJRQ
RESET	DMKCPBRS
REWIND	DMKCPBRW
SAVESYS	DMKCFHSV
SCREEN	DMKCFWEP
SEND	DMKSNDNH
SET	(See footnote 1.)
SHUTDOWN	DMKCPSSH
SLEEP	DMKCFJSL
SMSG	DMKMSGSM
SPACE	DMKCSFSP
SPMODE	DMKSPMEP
SPOOL	DMKCSPSP
SPTAPE	DMKSPTEP
START	DMKCSOST
STCP	DMKCDSCP
STORE	DMKCDSTO
SYSTEM	DMKCPBSR
TAG	DMKCSTAG

Command	Entry Point(s)
TERMINAL	DMKCFTRM
TRACE	DMKTRACE
TRANSFER	DMKCSVTS DMKCSVTG
UNLOCK	DMKCPVUL
VARY	DMKCPTNF
VMDUMP	DMKVMDEP
WARNING	DMKMSGWN
*	DMKCFM

[1]DMKCFC uses a subcommand table in DMKCMD to find the entry points for the subcommands ATTACH, DETACH, INDICATE, NETWORK, QUERY, and SET. See the following chart.

Subcommand		Function Type	Entry Label in DMKCMD	Final Entry Label
ATTACH		R	DMKCMDAT	DMKVDAAA
ATTACH	CHANNEL	R		DMKVDAAC
CACHE	ON	R	DMKCMDCA	DMKCACON
	OFF	R		DMKCACOF
	OWN	R		DMKCAOWN
	QUERY	R		DMKCACQY
DETACH		R	DMKCMDDE	DMKVDDDR
DETACH	CHANNEL	R		DMKVDDDS
DETACH		G	DMKCMDDG	DMKVDDDG
DETACH	CHANNEL	G		DMKVDDDC
INDICATE	DPA	A,O	DMKCMDIN	DMKTHIDP
	FAVORED	A,O		DMKTHIFA
	I/O	A		DMKTHIIO
	PAGING	A		DMKTHIPA
	POSITION	A		DMKTHIPO
	QUEUES	A		DMKTHIQQ
	USER	A		DMKTHIUS
	LOAD	A,G	DMKCMDIG	DMKTHILO
	USER	G		DMKTHIUG
NETWORK	SHUTDOWN	O	DMKCMDNO	DMKNESHD
	ATTACH	O,R	DMKCMDNR	DMKNEAAH
	DETACH	O,R		DMKNEADH
	DISABLE	O,R		DMKNETDB
	DISPLAY	O,R		DMKNESDS
	DUMP	O,R		DMKNLEMP
	ENABLE	O,R		DMKNETEN
	LOAD	O,R		DMKNLDR
	POLLDLAY	O,R		DMKNESPL
	QUERY	O,R		DMKNETQU
	VARY	O,R		DMKNETVA

Subcommand		Function Type	Entry Label in DMKCMD	Final Entry Label
QUERY	ALL	R	DMKCMDQR	DMKCQPQA
	DASD	R		DMKCQPQD
	DUMP	R		DMKCQRDP
	GRAF	R		DMKCQPQG
	LINES	R		DMKCQPQL
	MITIME	R		DMKCQSMI
	STATUS	R		DMKCQTST
	STORAGE	R		DMKCQPQS
	SYSTEM	R		DMKCQPQS
	TAPES	R		DMKCQPQT
	TDSK	R		DMKCQQQT
	UR	R		DMKCQPQU
	VECTOR	R	DMKCMDQS	DMKVFCQV
	FILES	S		DMKCQHFS
	HOLD	S		DMKCQRHD
	PRINTER, PRT	S		DMKCQHTS
	PUNCH, PCH	S		DMKCQHNS
	READER, RDR	S	DMKCMDQA	DMKCQHRS
	AFFINITY	A,O		DMKCQRAF
	CPASSIST	A,O		DMKCQYCA
	JOURNAL	A,O		DMKJRLQU
	MINWS	A,O		DMKCQSMW
	PAGING	A,O		DMKCQRPG
	PRIORITY	A,O		DMKCQRPR
	PSTOR	O,R		DMKCQSPS
	QDROP	A,O		DMKCQRQD
	RESERVE	A,O		DMKCQSRE
	SASSIST	A,O		DMKCQYSA
	SRM	A		DMKSRMQS
	CPLEVEL	G		DMKCQTCL
	CPUID	G		DMKCQYCL
	CPLANG	G		DMKCQYCP
	FILES	G	DMKCMDQP	DMKCQHFG
	LINKS	all		DMKCQQQL
	LOGMSG	all		DMKCQYLM
	NAMES	all		DMKCQSNA
	PER	G		DMKPEQRY
	PF	G		DMKCQYPF
	PRINTER, PRT	all		DMKCQHTG
	PROCESSR	G		DMKCQPQP
	PUNCH, PCH	G		DMKCQHNG
	READER, RDR	G		DMKCQHRG
	SCREEN	G		DMKCQSSC
	SECUSER	G	DMKCMDQQ	DMKCQUSE
	SET	G,O		DMKCQUST
	SPMODE	G		DMKCQYSP
	S370E	G		DMKCQYS3
	TERMINAL	G		DMKCQUTE
	TIME	all		DMKCQYTI
	CPTRAP	G		DMKCQCPT
	USERS	G		DMKCQYUS
	USERID	G		DMKCQYUI
	VIRTUAL	P		DMKCMDQV
	VMSAVE			DMKCQYVS

Subcommand		Function Type	Entry Label in DMKCMD	Final Entry Label
QV[2]	ALL	G	DMKCMDQV	DMKCQGQA
	CHANNELS	G		DMKCQGQH
	CONSOLE	G		DMKCQGQC
	DASD	G		DMKCQGQD
	GRAF	G		DMKCQGQG
	LINES	G		DMKCQGQL
	STORAGE	G		DMKCQGQS
	TAPES	G		DMKCQGQT
	UR	G		DMKCQGQU
SET	AFFINITY	O	DMKCMDSO	DMKCFYAS
	CPASSIST	O		DMKCFOSC
	FAVORED	O		DMKCSOSF
	JOURNAL	O		DMKJRLSE
	PRIORITY	O		DMKCFOSP
	QDROP	O		DMKCFOSQ
	RESERVE	O		DMKCFOSR
	SASSIST	O		DMKCFOSA
	S370E	O		DMKCFOS3
SET	DUMP	R	DMKCMDSR	DMKCFUDU
	LOGMSG	R		DMKCFULO
	MITIME	R		DMKCFUMI
SET	MINWS	A	DMKCMDSA	DMKCFUMW
	PAGING	A		DMKCFUPA
	SRM	A		DMKSRMSS
SET	MODE	C	DMKCMDSC	DMKCFUMO
	RECORD	C		DMKCFURE
SET	ACNT	G	DMKCMDSG	DMKCFSAC
	AFFINITY	G		DMKCFYAG
	ASSIST	G		DMKCFYSA
	AUTOPOLL	G		DMKCFSAP
	CONCEAL	G		DMKCFYSC
	CPCONIO	G		DMKCFSCC
	CPUID	G		DMKCFSCP
	ECMODE	G		DMKCFSEC
	EMSG	G		DMKCFSEM
	IMSG	G		DMKCFSIM
	ISAM	G		DMKCFSIS
	LINEDIT	G		DMKCFSLE
	MIH	G		DMKCFVMI
	MSG	G		DMKCFSMG
	NOTRANS	G		DMKCFSNT
	PAGEX	G		DMKCFSPX
	RUN	G		DMKCFSRN
	SMSG	G		DMKCFSSM
	STBYPASS	G		DMKCFVSB
	STMULTI	G		DMKCFVSM
	SVCACCL	G		DMKCFSSA
	TIMER	G		DMKCFYSM
	VMCONIO	G		DMKCFSVC
	VMSAVE	G		DMKCFSVS
	WNG	G		DMKCFSWG
	370E	G		DMKCFS37

[2]QV – QUERY VIRTUAL

Function Codes for DIAGNOSE Instructions

The following table indicates the DIAGNOSE codes used in VM/SP HPO and gives a brief explanation of their uses.

Function Code	Class	Function	DMKHVC Label	DMKHVD Label	DMKHVE Label	DMKHVF Label
000	any	Store extended identification code.		HVDSTIDX		
004	C,E	Examine data from real storage.		HVDCPC		
008	any	Execute VM/SP HPO CP commands.	HVCONFN			
00C	any	Pseudo-timer facility.	HVCHRON			
010	any	Release virtual storage pages.	HVCPGRL			
014	any	Manipulate input spool files.		HVDSPRD		
018	any	Standard DASD I/O.	HVCDISK			
01C	F	Clear error recording area.		HVDLRER		
020	any	General virtual I/O.	HVCFAKE			
024	any	Virtual device type inquiry.		HVDDTYP		
028	any	Dynamic CCW modification.	HVCDCPM			
02C	C,E,F	Get DASD address of error recording and number of cylinders allocated for error recording.			HVEEREP1	
030	C,E,F	Read a page of error recording data.			HVEEREP2	
034	C,E	Reads the system dump spool file.		HVDRSDF		
038	C,E	Reads the system symbol table.		HVDRDSYM		
03C	A,B,C	Dynamically updates the directory.		HVDDIRCT		
040	any	Clean up after virtual IPL by device.		HVDIPL		
044		Reserved for IBM use.	HVCEXIT			
048	any	Notify first level CP that this is a second level VM/370 or VM/SP system and this virtual machine has issued SVC 76.	HVCEXIT			
04C	any	Generate accounting records.		HVDACCT		
050	A,B,C	Saves 3704/3705 control program image.		HVD3705		
054	any	Enable or disable external interruptions.		HVDEXPA		
058	any	Virtual console interface for 3270.	HVCGRAF			
05C	any	Edit message according to EMSG settings.	HVCEMSG			
060	any	Provide virtual machine storage size.	HVCSTOR			
064	any	Load, find, or purge a named system.	HVCSYS			
068	any	Virtual Machine Communication Facility.	HVCVMCF			
06C	any	Low-address-protection interface for shadow table maintenance.	HVCSTBY			
070	any	Virtual machine accounting interface.		HNDVMAI		
074	A,B,C	Loads a 3800 printer named system into virtual storage.		HVD3800		
078	any	MSS communication.	HVCSSS			
07C	any	Virtual RDEVBLOK creation.		HVDVMPS		
080	any	Processes MSSFCALL instruction.	HVCMSSF			
084	B	Updates in-place a directory control statement in its online control block form.		HVDDUIP		
088		Reserved for IBM use.				
08C	any	Accesses certain device-dependent information.			HVEQRLY	
094	any	Dumps virtual storage.			HVEDUMP	

Function Code	Class	Function	DMKHVC Label	DMKHVD Label	DMKHVE Label	DMKHVF Label
098	any	Real I/O support for virtual machines.			HVEDRIO	
0A0	any	ACI security function.	HVCGRP			
0B0	any	Accesses diagnostic information saved for protected application facility users.			HVEPROT	
0B4	any	Virtual printer external attribute buffer manipulation.			HVEXABD	
0B8	any	Spool file external attribute manipulation.			HVEXABS	
0BC	any	Opens a spool file for a spooled reader device.				HVFOSPID
0C8	any	Sets a language for CP.				HVFNLSST
0CC	E	Saves a CP message repository for a language.				HVFNLSSA
0D0	any	3480 tape volume serial support.	HVCVOL			
0D4	B	Specifies an alternate user ID.				HVFALTID
0D8	D	Gets system spool information.				HVFSSI
0FC		Reserved for IBM use.				
100		Start of functions specified by a user.	HVCUSER			

Appendix A. Hardware Assist Commands

Hardware Assist Commands

Figure 37 illustrates the possible ways of running a virtual machine with various combinations of hardware assists and how the SET command affects their operation.

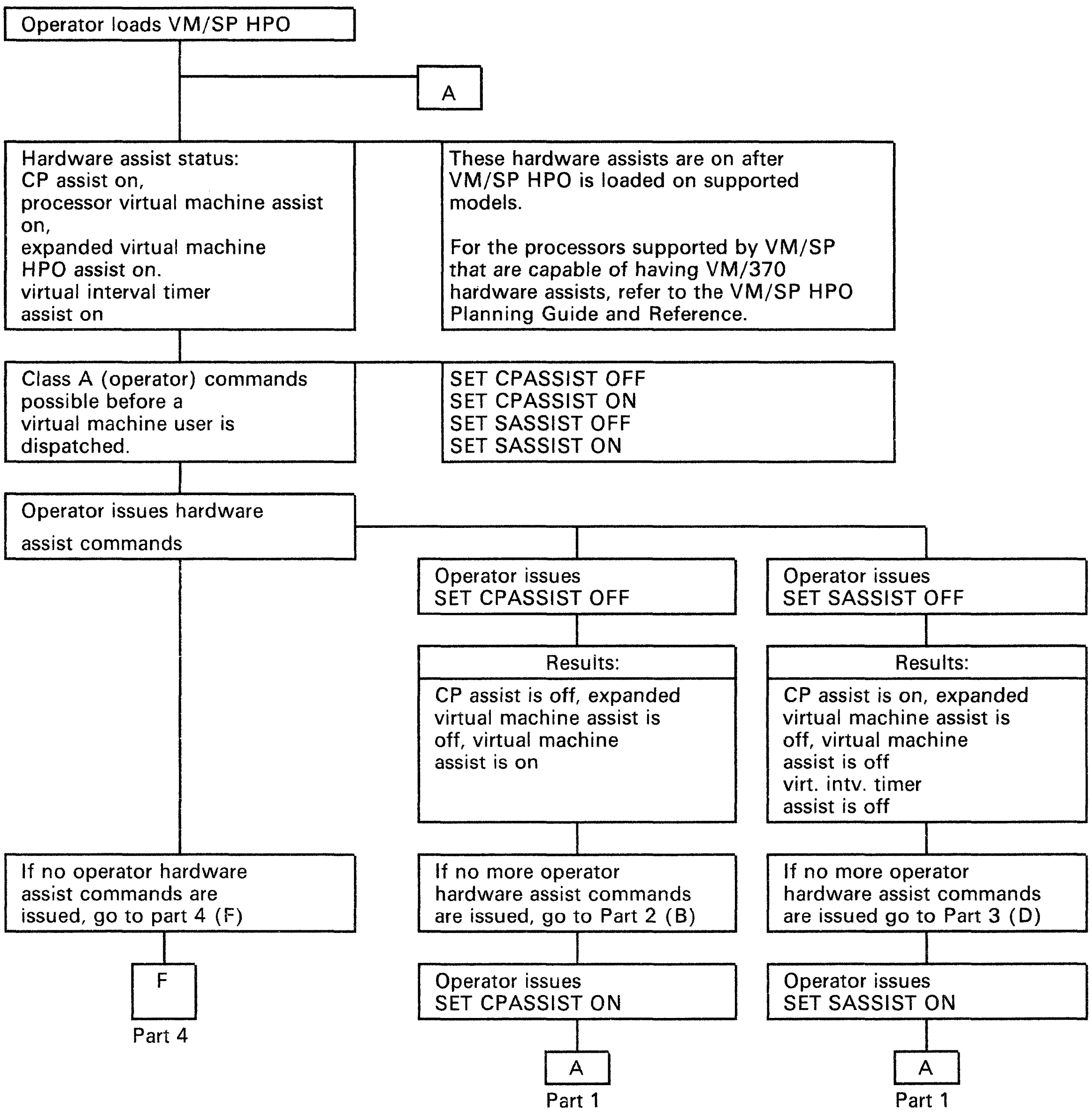


Figure 37 (Part 1 of 4). Hardware Assist Commands

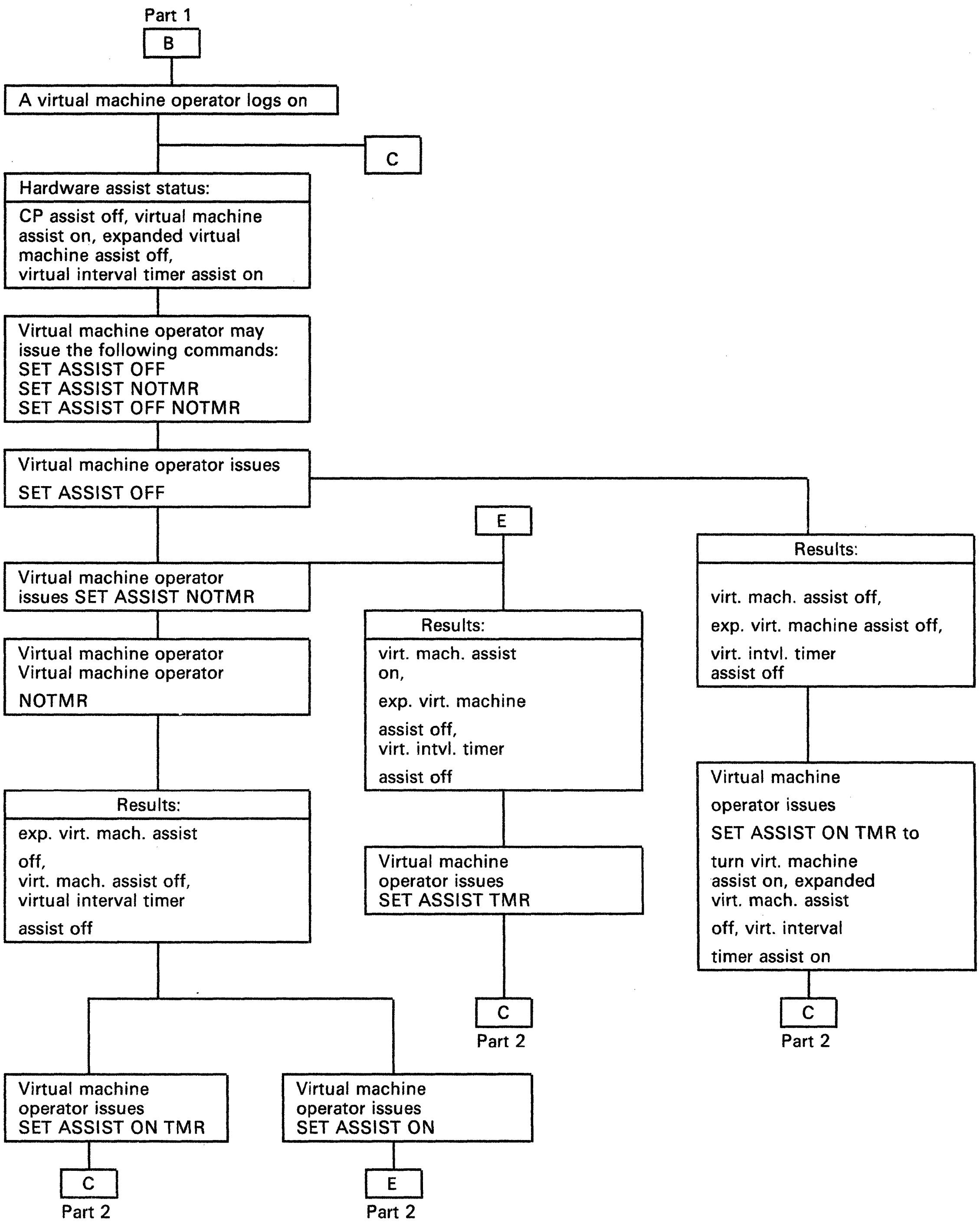


Figure 37 (Part 2 of 4). Hardware Assist Commands

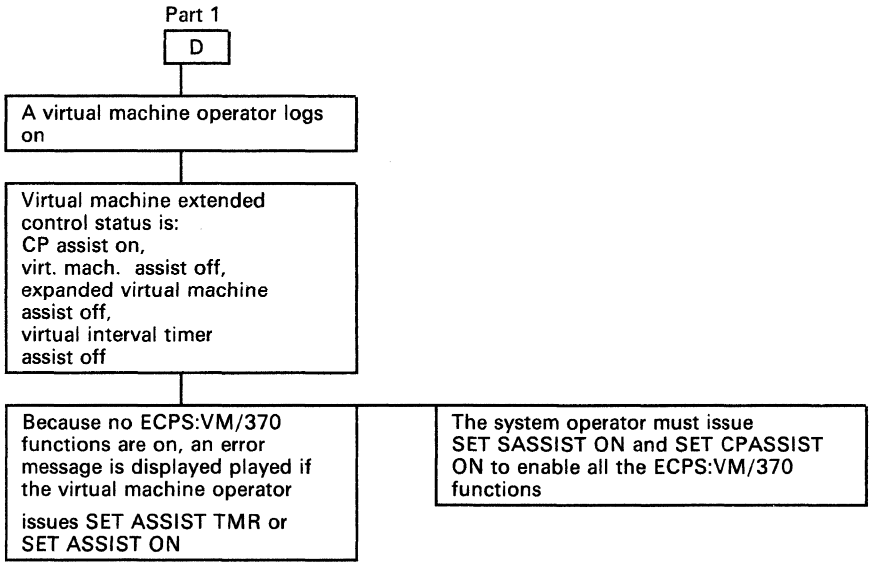


Figure 37 (Part 3 of 4). Hardware Assist Commands

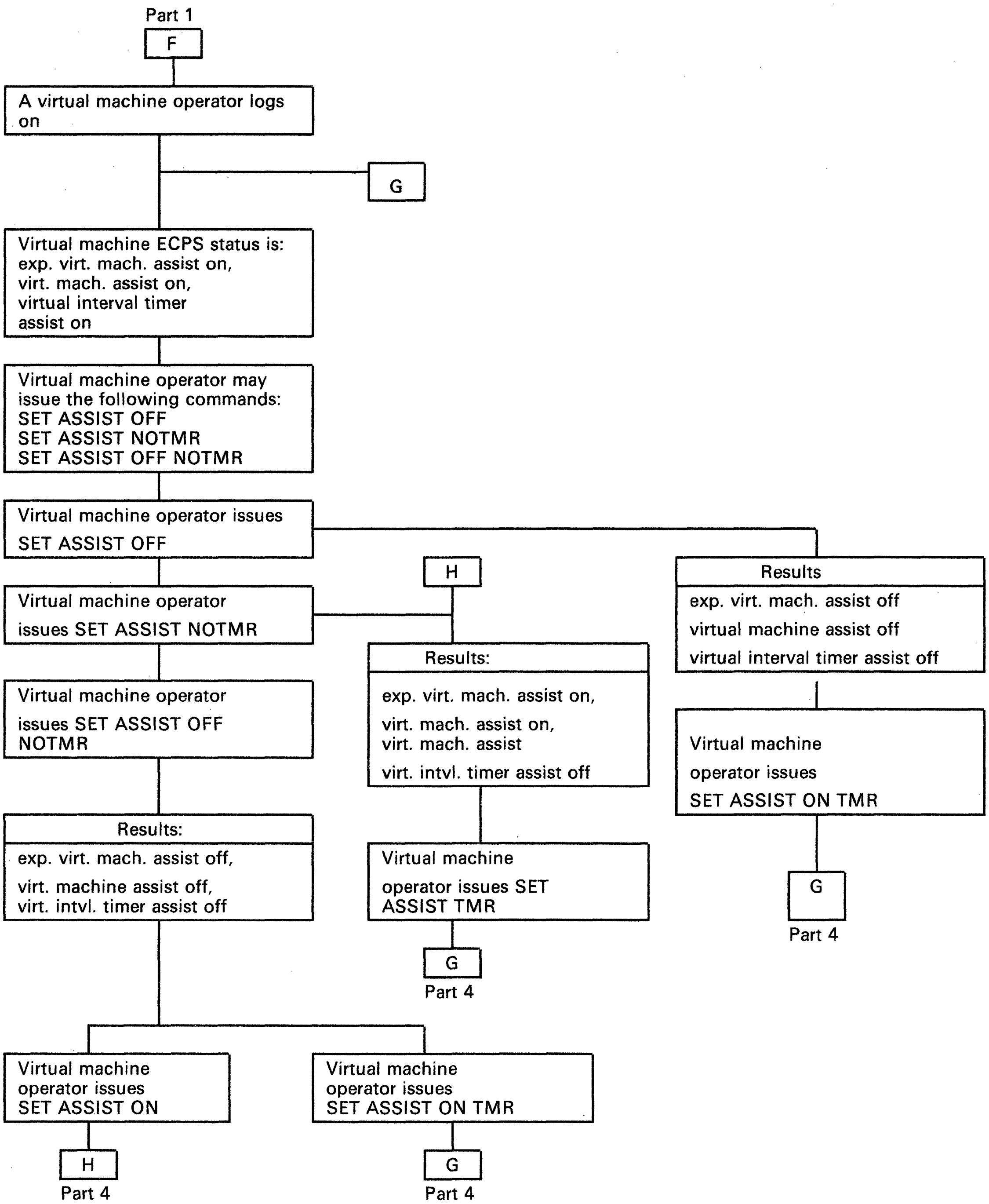


Figure 37 (Part 4 of 4). Hardware Assist Commands

Assist Status according to ECPS Level

Assists are disabled or selectively enabled in different combinations, according to the ECPS level installed on your system. Use the following information as a guide:

1. On **any** ECPS level, the PTLB assist is disabled.
2. If the ECPS level is 17 or less, all of the CP assists are disabled.
3. If the ECPS level is greater than 22, all of the CP assists are disabled.
4. For ECPS levels 18 - 22, the following assists are disabled or enabled as shown below:

CP Assist	ECPS Level 18, 19 UP	ECPS Level 18, 19 AP/MP	ECPS Level 20 UP	ECPS Level 20 AP/MP	ECPS Level 21 UP	ECPS Level 21 AP/MP	ECPS Level 22
Get prime free*	D	D	D	D	D	D	E
Get free storage*	E	D	E	D	E	D	E
Release free storage*	E	D	E	D	E	D	E
Free CCW storage*	E	D	E	D	E	D	E
Dispatcher 0	E	E	E	E	E	E	E
Dispatcher 1&2*	E	E	E	E	E	E	E
Dispatcher 3&4*	D	D	D	D	D	D	E
Dispatcher 5&6*	D	D	D	D	D	D	E
SVC 8 (LINK)	E	C	E	C	E	C	E
SVC 12 (RETURN)	E	C	E	C	E	C	E
Find virtual I/O control block	D	D	D	D	E	E	E
Find real I/O control block	E3D4	E3D4	E3D4	E3D4	E	E	E
Common CCW proc.	E	D	E	D	E	D	E
Diagnose 18 standard I/O*	D	D	D	D	D	D	E

Legend:

E – Assist enabled (assist instruction left or inserted in code or appropriate mask bit in Control Register 6 set = 1)

D – Assist disabled (assist instruction in code replaced with no-op instructions or appropriate mask bit in Control Register set = 0)

C – Assist is crippled (assist instruction left in place but parameter list modified so that assist is effectively disabled)

E3D4 – Assist enabled if 3-bit mask shift used for RDEVBLOK indices (bit RDIDX in CPSTAT5 = 0)
Assist disabled if 4-bit mask shift used for RDEVBLOK indices (bit RDIDX in CPSTAT5 = 1)

* – Assists marked with an asterisk are disabled if you generate your system with the FRET Trap logic.

Expanded VM Assist	**ECPS Level 18, 19 UP**	**ECPS Level 18, 19 AP/MP**	**ECPS Level 20 UP**	**ECPS Level 20 AP/MP**	**ECPS Level 21 UP**	**ECPS Level 21 AP/MP**	**ECPS Level 22**
SIO	D	D	D	D	E	E	E
TCH	D	D	E	E	E	E	E

Legend:

E – Assist enabled (assist instruction left or inserted in code or appropriate mask bit in Control Register 6 set = 1)

D – Assist disabled (assist instruction in code replaced with no-op instructions or appropriate mask bit in Control Register set = 0)

Appendix B. VM/SP HPO MSS Support

VM/SP HPO MSS Support

Following are annotated flow diagrams for the logic to support the IBM 3850 Mass Storage System.

Log On a User Having a Minidisk on an Unmounted System Volume

DMKLNK, CHK3330V
A required system volume is not mounted, try to get a 3330V mounted if the minidisk is a 3330.

DMKSSSLN
Entry to mount an MSS system volume.

DMKSSS, FINDRDEV
Allocate a SYSVIRT real 3330V device. This may involve demounting a volume which is mounted but not in use. If there are none such volumes available, issue message DMKSSS080E and return with return code 8.

DMKSSS, BLDCOMMT
Construct an MSSCOM, filling in the volume serial, device address selected, type of request (mount), and userid.

DMKSSS, SETMNTFG
Build a CPEXBLOK for the return to DMKLNK after the MSC has processed the request. Chain it from field MSSTASK2. Build a CPEXBLOK for the return to DMKLNK after the mount is complete (pack change interruption received on the 3330V). Chain it from field MSSTASK1.

DMKSSS, SCHMSSC
Put the MSSCOM in the queue, generate an attention interruption for the communication device if necessary, and exit to DMKDSP.

DMKSSS, HVC04ENT
Entry when DIAGNOSE code X'78', subcode 4 is received. OS/VS is ready to process an MSC request. Place the next MSSCOM in the virtual machine, and return to DMKHVC.

DMKSSS, HVC08ENT
Entry from DMKHVC when DIAGNOSE code X'78', subcode 8 is received. The MSC has processed the mount request.

DMKSSS, RESETMQR
If there was an MSS error, write message DMKSSS083E and return to DMKLNK with return code 8.

DMKSSS, MNTCOM
If there was not an MSS error, indicate that the MSSCOM is now waiting for the pack change interruption. Write message DMKSSS088I. Return to DMKLINK with return code 4.

DMKLNK, MNTSETUP
Return from DMKSSS. Save the current workarea and control information. Return to caller.

DMKDSB
Entry from DMKDAS on pack change interruption. If the device is a 3330V, look for an MSSCOM waiting for this volume serial. If one is found, stack a CPFXBLOK for entry to DMKSSSEN. Exit to DMKDSP.

DMKSSSEN
Pick up the CPEXBLOK for DMKLNKSS and stack it.

DMKLNKSS
Complete the LINK processing for the minidisk.

Log On a User Having a 3330V Dedicated As a 3330V

DMKLOG, CALLMSSA
Determine that a virtual 3330V is needed, save the UDEVBLOK, call DMKSSSL1.

DMKSSSL1
Go through device allocation, etc., to schedule a mount.

DMKLOG, MSSMOUNT
If an MSS mount is in process (return code 4 from DMKSSS), proceed to get the next directory statement. Otherwise, find the RDEVBLOK for the device that DMKSSS allocated and continue the dedicate process.

DMKLOGSS
Entry from DMKDSB and DMKSSSEN after mount.

DMKSCNRU
Get the RDEVBLOK.

DMKVDSAT
Attach the virtual device.

DMKLOG, TSTV333V
If the virtual device is a 3330V, set flag RDEV333V to indicate that there is no CP MSS CCW prefix.

DMKLOG, FREEUDEV
If there is virtual I/O waiting, as indicated by a CPEXBLOK address in field MSSTASK3 of the MSSCOM used for the mount, stack the IOBLOK. Return to DMKDSP.

Process DIAGNOSE Code X'78'

DMKSSSHV
Entry from DMKHVC when DIAGNOSE code is X'78'.

DMKSSS, HVC00ENT
The entry subcode was 0. Save the communication device address and the communicator VMBLOK address. Set PSAMSS indicating that the MSC is now available.

DMKSSS, HVC04ENT
The entry subcode was 4. If there is an MSSCOM in the queue to be processed, call DMKPTRAN to get the communicator's buffer address. Put the MSSCOM in the virtual machine buffer.

DMKSSS, HVC08ENT
The entry subcode was 8. The MSC has processed a request. If there was an error, write message DMKSSS088E, dequeue the MSSCOM, stack the return to the DMKSSS caller from MSSTASK2 with a return code 8, and return to DMKHVC. If there was no MSC error, stack the MSSTASK2 CPEXBLOK with a return code of 4, and return to DMKHVC.

Generate the Channel Program Prefix for a 3330V

DMKCCW
Entry to generate a real channel program from a virtual machine channel program.

DMKCCW, CCWINDSD
If the real device is a 3330V, set a flag indicating that the MSS channel program prefix is needed.

DMKCCW, CCW02
If the prefix-needed flag is on and the virtual device is not a virtual 3330V, put the prefix in the RCWTASK.

DMKCCW, DASDTBL AND DEDDTBL
These are tables of addresses of routines that are to get control to process specific CCW operation codes for DASD and dedicated devices. In each subroutine, a check is made to see if there is an unresolved MSS prefix. If so, it checks to see if the virtual channel

program contains a SEEK. If so, it checks to see if the argument is used to generate the SEEK argument for the prefix. If not, the prefix CCW is set to SEEK to cylinder 0.

Generate the Channel Program Prefix for CMS I/O to a 3330V

DMKDGD
Entry to process I/O requests to DASD as initiated by the special DIAGNOSE code '78' interface from CMS.

DMKDGD, NOPRE
If the real device is a 3330V, set up the prefix in the RCWTASK.

DMKDGD, CHKMOUNT
The VDEVBLOK for the virtual device could not be found. Check to see if there is an MSS mount in process for the required system volume. If so, build a CPEVBLOK for this request, put the address in the MSSTASK3 field of the MSSCOM, and exit to DMKDSP.

Process a Staging Adapter Cylinder Fault

DMKIOTIN
Entry when ending status is received from a device. Check to see if the CSW contains CE-DE with no error status.

DMKIOT, TESTCYL
If the device type is a 3330V, see if the CE-DE is in the MSS prefix NOP CCW. If not, or if the device is dedicated as a virtual 3330V, stack the IOBLOK.

DMKSSS12
Set the IOBFLT flag, indicating that a cylinder fault is being resolved. Chain the IOBLOK from the REDEVFIOB field in the RDEVBLOK. Build a TRQBLOK to recognize missing attention interruptions and put it on the timer queue. Exit to the dispatcher.

Process an Attention Interrupt from a 3330V

DMKIOS, IOSUNSOL
Entry to process unsolicited I/O interrupts.

DMKIOS, CALLMSSA
If the interrupt is an attention, the device is a 3330V, and it is not dedicated as a 3330V. Call DMKSSSI1 to restart I/O.

DMKSSSI1
Find each IOBLOK for this device that has the IOBFLT flag set. Find the associated timer queue element and remove it from the timer queue. Turn off IOBFLT so that the IOBLOK can be restarted when the device is available.

Appendix C. MVS Considerations

MVS/System Extensions and MVS System Product support, available as a part of VM/SP HPO, allows an MVS system running in a virtual machine to utilize the enhancements available in the MVS/System Extensions Program Product (Program No. 5740-XE1) or the MVS System Product. Included in MVS/System Extensions and MVS/System Product support is the use of the System/370 Extended Facility of the 3031, 3032, 3033, and 3081 processors, or the System/370 Extended Feature of the Model 158 and 168 processors. For additional information on the System/370 Extended Facility or Feature, see the publication *IBM System/370 Extended Facility,* Order No. GA22-7072.

Figure 38 illustrates the relationship among MVS/System Extensions, MVS/System Product, VM/SP HPO, and the System/370 Extended Facility.

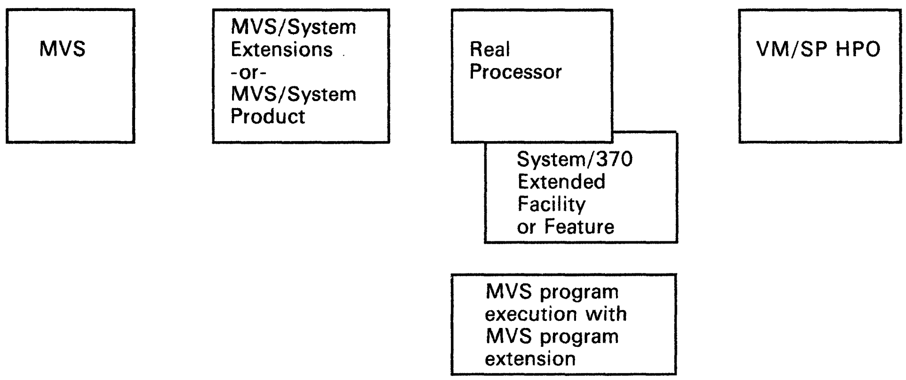


Figure 38. Relationship among MVS/System Extensions or MVS/System Product, VM/SP HPO, and the System/370 Extended Facility

The system operator enables and disables MVS/System Extensions or MVS/System Product support for all virtual machines using the SET S370E ON/OFF command. The user enables and disables it for his own virtual machine via the SET 370E ON/OFF command.

VM/SP HPO determines the status of MVS/System Extensions or MVS/System Product support through two bits in DMKPSA: CP370EAV and CP370EON. CP370EAV indicates whether the feature is available (1=yes, 0=no). CP370EON indicates whether the feature, if available, may be used by virtual machines (1=yes, 0=no). When DMKCPI executes an MVS/System Extensions or MVS/System Product instruction, and the instructions does not cause a program check, DMKCPI sets the value of CP370EAV to one. CP370EAV may not be turned on or off after it is set. CP370EON is initialized to the same value as CP370EAV; it may be turned on or off via the SET S370E command.

MVS/System Extensions and MVS/System Product support includes the following features:

- Low-address protection
- Common-segment facility
- Invalidate Page Table Entry (IPTE) instruction
- Test Protection (TPROT) instruction
- Virtual Machine extended-facility operations

Low-Address Protection

Low-address protection prevents improper storing by instructions using logical storage addresses between 0 and 511. This facility is designed to prevent inadvertent program destruction of those storage locations that are used by the processors to fetch new PSWs during interruption processing. Low-address protection does not apply to the following:

- The storing of status by the processors (that is, old PSWs, logout data)
- Channel stores (for example, the CSW, LCL, or data)
- Diagnose instructions issued by the virtual machine

Bit 3 of control register 0 is defined as the low-address-protection bit, and is used to control whether or not stores using logical addresses between 0 and 511 are permitted. When this bit is zero, stores are permitted; when it is one, stores are not permitted. When an instruction attempts a store using an address between 0 and 511 and low-address protection applies, the contents of the storage area addressed by the instruction are not modified. The execution of the current instruction is terminated or suppressed, and a program interruption for a protection exception occurs.

Low-address protection is in force when all of the following conditions are met:

1. Bit 3 of the virtual machine's virtual control register 0 is set to one.
2. The VMF370E bit in the VMBLOK is on, indicating that MVS/System Extensions or MVS/System Product functions are enabled for this virtual machine.
3. The CP370EON bit in the PSA is on.
4. The virtual PSW indicates that EC mode is on.

The storage-protection routine, DMKPSASP, in DMKPSA also checks for violations against low-address protection (this routine is used for simulating privileged instructions, such as Store Then AND System Mask).

Common Segments

The common-segment facility allows addressing segments to be classified as private or common. If bit 30 of the segment table entry for a given segment equals one, the segment is a common segment. Otherwise, it is private. A private segment table entry and the page table it designates may be used only in association with the segment table origin (STO) that designates the segment table in which the segment table entry resides. A common segment-table entry and the page table it designates may continue to be used for translating addresses even though a different STO is specified by changing control register 1.

DMKVAT ignores bit 30 of the virtual segment table entry when MVS/System Extensions or MVS/System Product support is enabled (that is, when the VMF370E bit in the VMBLOK and the CP370EON bit in the PSA are both on). If bit 30 equals one, and MVS/System Extensions or MVS/System Product support is not enabled, a translation-specification exception is reflected to the virtual machine.

In DMKVAT, tests are made at labels GIVE012 and LRAEXCP to determine whether the common-segment bit is the only "invalid" segment table entry bit on for a virtual machine with MVS/System Extensions or MVS/System Product support enabled. If it is, and the VMF370E bit in the VMBLOK equals one, and the CP370EON bit of the CPSTAT2 byte of the PSA equals one, then no translation-specification exception is generated. Execution returns to the in-line code after label DMKVATLA (for LRAEXCP) or label DMKVATRN for GIVE012).

Invalidate Page Table Entry (IPTE) Instruction

Execution of the IPTE instruction causes the page table entry designated by the second operand (Ry) to be invalidated in the page table designated by the first operand (Rx), and the associated translation-lookaside buffer (TLB) entries in the system to be purged.

IPTE is an RRE-format instruction (four bytes long) with operation code X'B221'. The Rx field is in bits 24-27 of the instruction and the Ry field is in bits 28-31. The contents of register Rx have the format of a segment table entry with only the page table address used. The contents of register Ry have the format of a virtual address with only the page index used. Other fields of registers Rx and Ry are ignored. The translation format is contained in bits 8-12 of control register 0. The bit positions of register Ry that are selected as a page index depend on the segment and page size specified in control register 0.

Bits 8-12 of Control Register 0	Ry Bit Positions for Page Index
01000 (PS=2K,SS=64K)	16-20
10000 (4K,64K)	16-19
01010 (2K,1M)	12-20
10010 (4K,1M)	12-19

The page table address and page index are used, subject to the rules of dynamic address translation for page table lookup, to form a real address that designates a halfword page table entry. If the page size is 2K, bit 13 of the entry is set to one. Otherwise, the page size is 4K, so bit 12 of the entry is set to one. In any event, key protection applies to the access.

VM/SP HPO simulates an IPTE instruction by setting the proper bit in the virtual machine's page tables and then proceeding as if the instruction were a PTLB (that is, resetting the virtual machine's shadow tables and executing a PTLB with the proper synchronization procedures).

The secondary-operation decode routine for X'B2xx' instructions at label BEETWOS causes a branch to the routine DMKPRWIP in module DMKPRW. This routine simulates the IPTE instruction. DMKPRWIP computes the second-level page table (PGT) address, then calls TRANS21 to have the page table entry address translated to a real address. This address is checked for fetch and storage protection via calls to DMKPFAFP and DMKPSASP, respectively. Failing either protection check causes a branch to label PROTEXCP to reflect a protection exception to the virtual machine. After a successful protection check, DMKPRWIF sets the validity bit in the PGTE (PGTPVM) to one, which indicates that the entry is invalid.

If an IPTE instruction is executed by a virtual machine that does not have MVS/System Extensions or MVS/System Product support enabled, an operation exception is returned to the virtual machine.

Test Protection (TPROT) Instruction

The TPROT instruction is an SSE-format instruction (six bytes long) with operation code X'E501'. The location specified by the first-operand address is tested for protection (including low-address protection) exceptions, using the key specified in bits 24-27 of the second-operand address. The first-operand address is subject to translation when dynamic address translation is on. Condition codes are set as follows:

cc=0 Both fetching and storing are permitted.
cc=1 Fetching is permitted, but storing is not.
cc=2 Neither fetching nor storing is permitted.
cc=3 Translation not available.

VM/SP HPO simulates the TPROT instruction in subroutine DMKPRWTP by comparing the key specified by bits 24-27 of the second-operand address to the key for the 2048-byte block specified by the swap table entry corresponding to the first-operand address. The virtual condition code is set to reflect the results of the comparison.

Virtual Machine Extended-Facility Assist

The IBM System/370 Extended Facility includes four lock instructions, six tracing instructions, and the Fix Page and SVC Assist instructions, all of which are MVS-dependent instructions. All are SSE-format instructions (six bytes long) with operation code X'E5xx', where xx represents the secondary-operation code, code, which can take values between X'02' and X'0D'. These instructions are described in more detail in *IBM System/370 Extended Facility,* Order No. GA22-7072.

MVS/System Extensions and MVS/System Product support includes the virtual machine extended-facility assist for the 12 MVS-dependent operations. Control of this facility is handled by bits 1 and 29 of control register 6. Bit 1 is the virtual-machine problem-state bit (one indicates the problem state, and zero indicates the supervisor state), and bit 29 activates the virtual machine extended-facility assist for the MVS-dependent instructions. If bit 29 is one and bit 1 is zero, the MVS-dependent operations are executed, even when bit 15 of the real PSW is zero.

If bit 1 is one or bit 29 is zero, then all of the MVS-dependent operations cause program interruptions (privileged operation exceptions).

VM/SP HPO does not support the simulation of the MVS-dependent instructions; these instructions are executed by the hardware only.

Virtual machine extended-facility assist is activated in the following way:

When the VMF370E bit in the VMBLOK is one, bit 1 of real control register 6 is set to the same value as bit 15 (the problem state bit) in the virtual machine's PSW (VMPSW). Then the following tests are made:

1. Is the global MVS/System Extensions or MVS/System Product status bit equal to one (bit CP370EON in byte CPSTAT2 of the PSA)?

2. Is the extended-control-mode bit equal to one (bit 12 of VMPSW)?

If these tests are met, then:

1. Bit 29 of real control register 6 is set to one (this activates the virtual machine extended-facility assist).

2. Bit 3 of real control register 0 is set to the value in virtual control register 0 (this indicates whether low-address protection is enabled).

If these tests are not met, then:

1. Bit 29 of real control register 6 is set to zero.

2. Bit 3 of real control register 0 is set to zero.

These settings deactivate virtual machine extended-facility assist and low-address protection, respectively. In either case, processing continues normally.

Appendix D. Access Verification Routines

The access verification routines provide the ability to install a security software package (such as RACF/VM) which will give you greater control over minidisk access, logon passwords, and movement of spool files.

Using the ACIGROUP directory control statement, you can identify userids as belonging to certain security groups. This allows you to restrict certain types of system activity to those who are present in the appropriate security groups. See the *VM/SP HPO Service Routines and Program Logic* for more information on the ACIGROUP control statement.

Three modules, DMKRPW, DMKRPI, and DMKRPD, contain the access verification routines. They are described below.

DMKRPW

DMKRPW is called when either the LOGON or AUTOLOG command is issued. The command processor initializes ACIPARMS to pass the following information to DMKRPW:

- Function request code (X'10')
- Request groupname (if one exists)
- Request userid
- Logon passwords.

When a terminal issues the AUTOLOG command through an EXEC, the command processor initializes ACIPARMS to pass the following information to DMKRPW:

- Function request code (X'14')
- Request groupname (if one exists)
- Request userid
- Logon passwords.

Once a security package has been installed, the LOGON/AUTOLOG command processor recognizes the following return codes:

Return Code	Status
0	Request authorized
4	Request not verified
8	Request failed
12	User logged off
16	Terminal error

DMKRPI

DMKRPI is called when the LOGOFF (FORCE), LINK, TRANSFER, SPOOL, or TAG command is issued. The appropriate command processor initializes ACIPARMS to pass information to DMKRPI. The information is different for each command.

ACIPARMS passes information to DMKRPI as listed below.

- For the LINK command:
  - Function request code (X'00')
  - Request groupname (if one exists)
  - Request userid
  - Target groupname (if one exists)
  - Target userid
  - Resource address
  - Access mode.
- For the SPOOL or TRANSFER command:
  - Function request code (X'04')
  - Request groupname (if one exists)
  - Request userid
  - Target groupname (if one exists)
  - Target userid
  - Resource address.
- For the TAG command:
  - Function request code (X'08')
  - Request groupname (if one exists)
  - Request userid
  - Target groupname (if one exists)
  - Target userid
  - Resource nodename.
- For the LOGOFF (FORCE) command:
  - Function request code (X'0C')
  - Request groupname (if one exists)
  - Request userid.

Once a security package has been installed, the LOGOFF (FORCE), LINK, TRANSFER, SPOOL, and TAG command processors recognize the following return codes:

Return Code	Status
0	Request authorized
4	Request not verified
8	Request failed, terminate command

Entry Points Defined by DMKRPI for IUCV

DMKRPI defines entry points to support an IUCV interface. These entry points are:

```
DMKRPICN
DMKRPISV
DMKRPIIL
DMKRPIQS
DMKRPIRM
```

DMKRPD

DMKRPD handles the DIAGNOSE code, X'A0'. Use DIAGNOSE X'A0' to retrieve a groupname for a given userid.

Entry Values

The register specified as Ry contains the subcode X'00'. This requests DMKRPD to retrieve a groupname. The Rx register contains the address of a field consisting of 2 doublewords. The first doubleword is a userid. The second doubleword is empty. DMKRPD will return the groupname for the given userid to this empty field.

Exit Values

If a groupname exists for the given userid, DMKRPD will return the groupname to the second half of the 2-doubleword field.

Condition Codes Set by DMKRPD

DMKRPD sets the following condition codes:

Condition Code	Status
0	Request completed
1	Request failed

Bibliography

Prerequisite Publications

VM/SP HPO Release 5 Guide, Order No. SC23-0189

VM/SP HPO Operator's Guide, Order No. SC19-6225

VM/SP HPO CP for System Programming, Order No. SC19-6224

VM System Facilities for Programming, Order No. ST24-5288

VM/SP HPO CP Command Reference for General Users, Order No. SC19-6227

Corequisite Publications

VM/SP Introduction, Order No. GC19-6222

VM/SP HPO Planning Guide and Reference, Order No. SC19-6223

VM/SP HPO Installation Guide, Order No. SC38-0107

VM/SP HPO System Messages Cross Reference, Order No. SC19-6226

VM/SP HPO OLTSEP and Error Recording Guide, Order No. SC19-6230

Virtual Machine Running Guest Operating Systems, Order No. GC19-6228

VM/SP HPO Data Areas and Control Block Logic- CP, Order No. LY20-0896

In addition, for EREP processing the following OS/VS Library manuals are required:

OS/VS, DOS/VSE, VM/370 Environmental Record Editing and Printing (EREP) Program Logic, Order No. GC28-0772

OS/VS, DOS/VSE, VM/370 Environmental Record Editing and Printing (EREP) Program Logic, Order No. SY28-0773

If the IBM 3850 Mass Storage System is attached, the following manuals are required:

OS/VS Message Library: Mass Storage System (MSS) Messages, Order No. GC38-1000

IBM 3850 Mass Storage System (MSS) Principles of Operation: Theory, Order No. GA32-0035

IBM 3850 Mass Storage System (MSS) Principles of Operation: Reference, Order No. GA32-0036

If the VTAM Communications Network Application (VM/VCNA) product is used, the following manual is a prerequisite:

IBM VM/VCNA Diagnostics, Order No. LY38-3033

For information on the Group Control System:

VM/SP Group Control System Guide, Order No. SC24-5249.

For information on VM/VTAM:

Network Program Products General Information, Order No. GC30-3350.

Network Program Products Samples: VM SNA, Order No. SC30-3309.

Supplementary Publications

IBM System/360 Principles of Operation, Order No. GA22-6821

IBM System/370 Principles of Operation, Order No. GA22-7000

IBM 3270 Information Display System Components Description, Order No. GA27-2749

General Information Binary Synchronous Communications, Order No. GA27-3004

Input/Output Configuration Program User's Guide and Reference, Order No. GC28-1027

Field Engineering Programming Systems General Information Reference Summary, Order No. G229-2228.

VM/SP High Performance Option Library

To understand the interrelationships between the publications comprising the VM/SP High Performance Option library, see Figure 39.

The VM/SP HPO Library

Evaluation

- VM/SP Introduction GT19-1977
- What's In VM/SP HPO Release 5 GC23-0384
- Announcing VM/SP HPO Release 5 GC19-6221
- VM/SP General Information GT00-1976

Index

- VM/SP HPO Library Guide, Glossary, and Master Index GC23-0187

Planning

- VM/SP HPO Planning Guide and Reference SC19-6223
- Virtual Machine Running Guest Operating Systems GC19-6212
- VM/SP Distributed Data Processing Guide SQ24-5241
- VM/SP HPO Release 5 Guide SC23-0189
- Input/Output Configuration Program User's Guide and Reference GC28-1027
- 3090 Processor Complex Input/Output Configuration Program User's Guide and Reference SC38-0038

Operation

- VM/SP HPO Operator's Guide SC19-6225

Installation

- VM/SP HPO Installation Guide SC38-0107

Administration

- VM/SP HPO CP for System Programming SC19-6224
- Virtual Machine System Facilities for Programming ST24-5288
- VM/SP CMS for System Programming ST24-5286
- VM/SP GCS Macro Reference SQ24-5250
- VM/SP TSAF Reference ST24-5287

End Use

- VM/SP Terminal Reference GT00-1979
- VM/SP CMS Primer ST00-1992
- VM/SP CMS Primer For Line-Oriented Terminals ST00-1993
- VM/SP CMS User's Guide ST00-1980
- VM/SP Macros and Functions Reference ST24-5284
- VM/SP CMS Command Reference ST00-1981
- VM/SP SP Editor User's Guide ST00-1985
- VM/SP SP Editor Command and Macro Reference ST00-1986
- VM/SP HPO CP Command Reference SC19-6227
- VM/SP SP Interpreter User's Guide ST00-1987
- VM/SP SP Interpreter Reference ST00-1988
- VM/SP EXEC-2 Reference ST00-1984

Reference Summaries

- VM/SP HPO Commands (General User) SX22-0003
- VM/SP HPO Commands (Other Than General User) SX22-0004
- Virtual Machine Problem Determination Reference Information LX23-0347
- VM/SP SP Editor Command Reference Summary ST00-1997
- VM/SP SP Interpreter Reference Summary ST00-1999
- VM/SP IPCS Reference Summary ST00-1601
- VM/SP EXEC-2 Reference Summary ST00-1372

Figure 39 (Part 1 of 2). Library – Interrelationship of Publications

Reference Summaries

- VM/SP HPO Quick Reference SX22-0005

Networking

- VM/SNA PSI Guide Methods and Components GG24-3059
- VM/SNA PSI Guide Use of Tools GG24-3060

Applications

- VM/SP Application Development Guide ST24-5247
- Programmer's Guide To The Server-Requester Programming Interface for VM/SP ST24-5291

Diagnosis

- VM/SP HPO System Messages and Codes SC19-6226
- Virtual Machine Diagnosis Guide LT00-2010
- VM/SP GCS Diagnosis Reference LT00-2012
- VM/SP Problem Reporting Guide SC24-5282
- VM/SP HPO Service Routines Program Logic LY20-0898
- VM/SP Data Areas and Control Blocks Volume 2 (CMS) LT00-2009
- VM/SP HPO System Logic and Problem Determination Guide-CP LY20-0897
- VM/SP System Logic and Problem Determination Guide Volume 2 (CMS) LT00-2007
- VM/SP HPO Data Areas and Control Blocks-CP LY20-0896

Auxiliary Service Support

- Device Support User's Guide and Reference GC35-0033

Device Support Facilities 5748XX9

- EREP User's Guide and Reference GC28-1378

Environmental Record Editing and Printing (EREP)

Auxiliary Communication Support

- RSCS Networking Version 2 General Information GH24-5055
- RSCS Networking Version 2 Planning and Installation SH24-5057
- RSCS Networking Version 2 Operation and Use SH24-5058
- RSCS Networking Version 2 Exit Customization LY24-5240
- RSCS Networking Version 2 Diagnosis Reference LY24-5228
- RSCS Networking Version 2 Reference Summary SX24-5135

RSCS Networking Version 2 5664-188

- VTAM Operation SGC23-0113
- VTAM Customization SC23-0112
- VTAM Messages and Codes SC23-0114
- VTAM Data Areas (VM) LY30-5583
- VTAM Diagnosis Guide LY23-0116
- VTAM Diagnosis Reference LY30-5582

Advanced Communications Function for VTAM (ACF/VTAM) 5664-280

- VTAM Programming SC23-0115
- VTAM Installation and Resource Definition SC23-0111
- VTAM Reference Summary SC23-0135
- VM/Pass-Through Facility General Information GC24-5206
- VM/Pass-Through Facility Guide and Reference SC24-5208
- VM/Pass-Through Facility Logic LY24-5208

VM/Pass-Through Facility 5748-RC1

Figure 39 (Part 2 of 2). Library – Interrelationship of Publications

Index

D

E

L

M

Q

R

S

W

Numerics

Virtual Machine/
System Product
High Performance Option

System Logic and Problem
Determination Guide – CP

LY20-0897-7
File No. S370-36

READER'S COMMENT FORM

Order No. LY20-0897-7

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

How did you use this publication?

[ ] As an introduction
[ ] As a reference manual
[ ] For another purpose (explain) ______________________________

____

[ ] As a text (student)
[ ] As a text (instructor)

Is there anything you especially like or dislike about the organization, presentation, or writing in this manual? Helpful comments include general usefulness of the book; possible additions, deletions, and clarifications; specific errors and omissions.

Page Number: Comment:

What is your occupation? ______________________________

Newsletter number of latest Technical Newsletter (if any) concerning this publication: __________

If you wish a reply, give your name and address: ______________________________

____

____

IBM branch office serving you ______________________________

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Note: Staples can cause problems with automatic mail-sorting equipment.
Please use pressure-sensitive or other gummed tape to seal this form.

LY20-0897-7

Reader's Comment Form

LY20-0897-7
File No. S370-36

Fold and Tape **Please Do Not Staple** **Fold and Tape**

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department 52Q MS 458
Neighborhood Road
Kingston, New York 12401

Fold and Tape **Please Do Not Staple** **Fold and Tape**

PRINTED IN U.S.A. LY20-0897-7

LY20-0897-7
File No. S370-36

VM/SP HPO System Logic and Problem Determination Guide—CP File No. S370-36 Printed in U.S.A. LY20-0897-7

LY20-0897-07