

# GCOS7

## System Installation and Configuration User's Guide

DPS7000/XTA  
NOVASCAL 7000

Operating System: Installation



REFERENCE  
47 A2 23US 05



# DPS7000/XTA NOVASCALE 7000 GCOS7

## System Installation and Configuration User's Guide

Operating System: Installation

**February 2005**

BULL CEDOC  
357 AVENUE PATTON  
B.P.20845  
49008 ANGERS CEDEX 01  
FRANCE

**REFERENCE**  
**47 A2 23US 05**

The following copyright notice protects this book under Copyright laws which prohibit such actions as, but not limited to, copying, distributing, modifying, and making derivative works.

Copyright © Bull SAS 1997, 2005

Printed in France

Suggestions and criticisms concerning the form, content, and presentation of this book are invited. A form is provided at the end of this book for this purpose.

To order additional copies of this book or other Bull Technical Publications, you are invited to use the Ordering Form also provided at the end of this book.

### **Trademarks and Acknowledgements**

We acknowledge the right of proprietors of trademarks mentioned in this book.

Intel® and Itanium® are registered trademarks of Intel Corporation.

Windows® and Microsoft® software are registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark in the United States of America and other countries licensed exclusively through the Open Group.

Linux® is a registered trademark of Linus Torvalds.

*The information in this document is subject to change without notice. Bull will not be liable for errors contained herein, or for incidental or consequential damages in connection with the use of this material.*



---

## Preface

### Scope and Objectives

This System Installation Configuration and Updating Guide is intended for data processing personnel responsible for installing and updating the latest release of GCOS 7-TA/V9 or GCOS 7-XTA/V10 on:

- DPS 7000/TAxxx/TAxxxC
- DPS 7000/MTxx
- DPS 7000/8xx/Dx0/Mx0
- DPS 7000/4x5/Cx0
- DPS 7000/XTAxxx

TA stands for GCOS 7 Twin Architecture and XTA for GCOS 7 eXtended Twin Architecture. Depending on the model and the customer order, some systems are delivered with the GCOS 7 Release already installed. Other DPS 7000s are delivered with an image of the GCOS 7 P-Set on tape or cartridge which is restored by Bull personnel.

All the utilities necessary for Installation and Updating are provided within a single product named the Installation and Updating Facility (IUF).

### Structure

- |           |   |
|-----------|---|
| Chapter 1 | <i>Introduction</i><br>Gives an overview of the IUF product. (Definitions of basic terms used in this manual can be found in the Glossary).                                 |
| Chapter 2 | <i>Installation Procedure</i><br>Gives examples of the procedures performed by Bull personnel for loading and updating the operating system.                                |
| Chapter 3 | <i>Multi-System Installation and Updating</i><br>Describes how to install and maintain several system disks for multiple systems with the same or different configurations. |



---

Chapter 4	<i>Installation and Updating Facility</i> Details the GIUF command which groups together the IUF functions and jobs used to install and maintain the operating system.
Chapter 5	<i>CONFIG (Configuration)</i> Describes the CONFIG job, for personalization of your system software configuration.
Chapter 6	<i>TAILOR</i> Describes the TAILOR utility, which enables the user to personalize his system files (produced as a result of the functions described in the main body of this manual), to suit the requirements of his own applications.
Chapter 7	<i>Updating Domains</i> Describes the procedures which enable you to perform preventative maintenance on the five system file Domains.
Chapter 8	<i>Patching Procedures</i> Describes the procedures which enable you to perform corrective maintenance on the two applicable system file Domains (GCOS and DSA).
Chapter 9	<i>Firmware Generation (FGF) and Test and Diagnostics (OLTD) Facilities</i> Describes the additional functions used by Bull personnel specific to installation and maintenance of the firmware Domain and the On-Line Test & Diagnostics Domain.
Chapter 10	<i>DSA Installation and Updating Facility</i> Details the DIUF facility specific to installation and maintenance of the frontend processor Domain.
Chapter 11	<i>GCOS 7 Service Facilities (GSF)</i> Details the additional functions specific to the installation and maintenance of the GCOS 7 Service Facilities Domain.
Chapter 12	<i>Maintenance and Recovery Procedures</i> Discusses the considerations of system installation and maintenance particular to the fixed-disk configuration.
Appendix A	<i>System File Descriptions</i> Describes the system files together with their Data Management attributes.



Appendix B	<i>SIP Functions</i> Describes loading and use of the backup operating system, SIP together with the associated utility UTIL.
Appendix C	Disk VOLPREP Procedure Describes how to prepare a disk, and explains the allocation of alternate tracks for bad tracks.
Appendix D	<i>Maintenance in a Protected Environment</i> Describes the manipulation of the files used in maintaining GCOS 7, when using access rights.

**Associated Documents**

The following manuals are essential for operating the IUF functions:

<i>Error Messages and Return Codes Message Directory</i> .....	47 A2 10UJ
<i>GCOS 7 System Administrator's Guide</i> .....	47 A2 54US
<i>GCOS 7 System Operator's Guide</i> .....	47 A2 53US
<i>GCOS 7 Console Messages Directory</i> .....	47 A2 61UU

The reader should also be familiar with the following manuals:

<i>GCOS 7 System Overview</i> .....	47 A2 22UG
<i>GCOS 7 Catalog Management User's Guide</i> .....	47 A2 35UF
<i>DPS 7000/4xx/Cx0 Operator's Guide</i> .....	77 A1 91UU
<i>DPS 7000/5x0/7xx Operator's Guide</i> .....	77 A1 81UU
<i>DPS 7000/8xx/Dx0/Mx0 Operator's Guide</i> .....	77 A1 94UU
<i>V7000 Operator's Guide</i> .....	47 A2 74US
<i>MainWay 2600LE User's Guide</i> .....	39A2 60AE
<i>MainWay 2600LE Installation and Maintenance</i> .....	39 A2 61AE

The following manuals describe complementary generation functions that are implemented after installation:

<i>DNS-V4 System Generation</i> .....	39 A2 22DN
<i>DNS-V4 NGL Reference Manual</i> .....	39 A2 23DN
<i>CNS 7A2-NGL Reference Manual</i> .....	39 A2 32DN
<i>CNS 7A2-NOI Operator's Guide</i> .....	39 A2 34DN
<i>DNS-E NGL Reference Manual</i> .....	39 A2 10EB
<i>DNS-E NOI Operator Manual</i> .....	39 A2 86RA
<i>OSF/SNA NGL Reference Manual</i> .....	39 A2 14RA
<i>MainWay DNS-E V1 System Generation Guide</i> .....	39 A2 74RA
<i>MainWay DNS-E V3 System Generation Guide</i> .....	39 A2 09EB
<i>GCOS 7 Networks: Overview</i> .....	47 A2 92UC
<i>GCOS 7 Network Generation</i> .....	47 A2 93UC
<i>GCOS 7 Network User's Guide</i> .....	47 A2 94UC



The following manuals are referred to in this document:

<i>Administering the Storage Manager</i> .....	47 A2 36UF
<i>File Migration Tool User's Guide</i> .....	47 A2 32UF
<i>File Recovery Facilities User's Guide</i> .....	47 A2 37UF
<i>Full Screen Editor User's Guide</i> .....	47 A2 06UP
<i>MFT User and Administrator's Guide</i> .....	47 A2 38UF
<i>Mirror Disks User's Guide</i> .....	47 A2 20UF
<i>Text Editor User's Guide</i> .....	47 A2 05UP

## Syntax Notation

The following conventions are used for presenting command syntax.

### ITEM

An item in upper case is a literal value, to be specified as shown. The upper case is merely a convention; in practice you can specify the item in upper or lower case.

### Item

An item in lower case is non-literal, indicating that a user-supplied value is expected.

In most cases it gives the type and maximum length of the value:

char105	a string of up to 105 alphanumeric characters
dec5	a decimal integer value of up to 5 digits
name31	a name of up to 31 characters
file78	a file name of up to 78 characters

In some cases, it gives the format of the value:

a	a single alphabetic character
nnnn	a 4-digit number
hh.mm	a time in hours and minutes

In other cases, it is simply descriptive of the value:

```
media:devclass
```

### ITEM

An underlined item is a default value.

### Bool

A boolean value which is either 1 or 0. A boolean parameter can be specified by its keyword alone, optionally prefixed by "N". Specifying the keyword alone always sets the value to 1. Prefixing the keyword with "N" always sets it to 0.

### { }

Braces indicate a choice of values. Only one may be selected.





- [ ] Square brackets indicate that the enclosed item is optional. An item not enclosed in square brackets is mandatory.
- ( ) Parentheses indicate that a single value or a list of values can be specified. A list of values must be enclosed by parentheses, with each value separated by a comma or a space.
- ... Ellipses indicate that the item concerned can be specified more than once.
- +=\$\*/ These are literal characters to be specified as shown.

**EXAMPLE 1:**

```
{
  { IMMED } }
{ WHEN={ [dd.mm.yy.]hh.mm } }
{      { +nnnn{W|D|H|M} } }
{
}
```

This means you can specify:

- Nothing at all (WHEN=IMMED applies).
- WHEN=IMMED (the same as nothing at all).
- WHEN=22.30 to specify a time (and today's date).
- WHEN=10.11.87.22.30 to specify a date and time.
- WHEN=+0002W to specify 2 weeks from now.
- WHEN=+0021D to specify 21 days from now.
- WHEN=+005H to specify 5 hours from now.
- WHEN=+0123M to specify 123 minutes from now.

□

**EXAMPLE 2:**

```
[ PAGES=(dec4 [-dec4] ...) ]
```

An optional parameter giving either a single value or a list of values enclosed in parentheses, with each value separated by a comma or space. Each value can consist of either a single number or a pair of numbers connected by a hyphen. For example:

```
PAGES=(2,4,10-25,33-36,78 83 90)
```

□

**EXAMPLE 3:**

```
[ REPLACE = { bool | 0 } ]
```

A boolean parameter whose default value is zero. You can specify:

- Nothing at all (REPLACE=0 applies)
- REPLACE=0 or simply NREPLACE
- REPLACE=1 or simply REPLACE

□

**Tape, Cartridge, and Cassette Devices**

Where a tape, cartridge, or cassette device has to be specified, this is shown in the syntax as `<tape-dvc>`.

This must be replaced by the appropriate full specification, as follows:

Standard tape drive	MT	(MT/D1600 or MT/6250 for VOLPREP only)
Streamer tape drive	MT/S	(MT/D1600 or MT/6250 for VOLPREP only)
18-track 1/2" cartridge	CT/M5	(with or without Sequential Cartridge Loader)
	CT/LIB/M5	(if in a library)
36-track 1/2" cartridge	CT/36T	(with or without Sequential Cartridge Loader)
	CT/LIB/36T	(if in a library)
8mm cassette	CT/M6	(CT/M6/S35 or CT/M6/S75 for VOLPREP)

**NOTES:**

1. Streamer mode for tape drives can be used only with VOLSAVE/VOLREST and FILSAVE/FILREST.
2. For cartridge drives, the attribute /C must be added for use of hardware compaction (if supported by the hardware).
3. When there is no possible ambiguity on a given site, that is where only one type of device is present, the device specification can be abbreviated, e.g. CT.



---

# Table of Contents

## 1. Introduction

1.1	Overview .....	1-1
1.1.1	System Disk Configurations .....	1-3
1.1.2	System Disks and Resident Disks .....	1-3
1.2	Operating System Maintenance - IUF .....	1-4
1.2.1	Features of IUF .....	1-4
1.2.2	IUF Functions .....	1-5
1.2.3	IUF Jobs .....	1-5
1.2.4	GCOS 7 Domains .....	1-6

## 2. Installation Procedure

2.1	Before Beginning .....	2-1
2.1.1	Hardware and Firmware Requirements .....	2-2
2.1.1.1	On DPS 7000/4x5/Cx0/MTx1 Systems .....	2-2
2.1.1.2	On DPS 7000/TAxXX/TAxXXC/8xx/Dx0/Mx0, and DPS 7000/MTx2 Systems and up .....	2-2
2.1.2	Installation Media Required .....	2-3
2.1.3	Initial Profiles .....	2-4
2.2	P-MEDIA Restoration Procedure .....	2-5
2.2.1	Systems not XTA .....	2-5
2.2.2	XTA Systems .....	2-5
2.3	Installation on a New DPS 7000 With Pre-loaded Disk .....	2-5
2.4	Installation for Update of GCOS 7 Release .....	2-10
2.4.1	Backup To Previous Release .....	2-10
2.4.2	Constraints .....	2-10
2.4.3	Alternative Installation Procedures .....	2-10
2.5	Creating a P2P Configuration .....	2-14
2.6	Creating an RP-Configuration .....	2-16



2.7	BackUp to a Previous Release in P0.....	2-18
2.8	BackUp to a Previous Release in P2P .....	2-18

### 3. Multi-System Installation and Updating

3.1	Introduction .....	3-1
3.1.1	Purpose of this Procedure .....	3-1
3.1.2	Master and Slave Systems.....	3-1
3.1.3	General Requirements .....	3-2
3.2	Installation Procedure .....	3-3
3.2.1	Saving a Set on the Master System .....	3-4
3.2.2	Installation on the Slave System .....	3-4
3.3	Updating .....	3-5
3.3.1	Purpose .....	3-5
3.3.2	Master System Updating Operations.....	3-7
3.3.2.1	Technical Status Application .....	3-7
3.3.2.2	Application of Software Add-Ons and Deletions .....	3-7
3.3.3	Slave System Updating Operations.....	3-9
3.3.3.1	Procedure.....	3-9
3.3.3.2	Selecting the Running Set or the Updated Set .....	3-9

### 4. Installation and Updating Facility

4.1	The GIUF Command.....	4-1
4.1.1	Environment Considerations .....	4-3
4.1.2	IUF Functions and Jobs.....	4-4
4.2	IUF Functions .....	4-5
4.2.1	The ADAPT Function.....	4-5
4.2.2	The ADD_MI Function .....	4-7
4.2.3	The BUILD_BKST Function.....	4-10
4.2.4	The BUILD_CMS Function (if applicable) .....	4-12
4.2.5	The BUILD_HALOCK Function (if applicable).....	4-14
4.2.6	The BUILD_JAS Function .....	4-16
4.2.7	The BUILD_JRNAL Function.....	4-17
4.2.8	The BUILD_LIB Function.....	4-19
4.2.9	The BUILD_MIRLOG Function (if applicable) .....	4-21
4.2.10	The BUILD_PVMF Function .....	4-23
4.2.11	The BUILD_QM Function .....	4-25
4.2.12	The BUILD_QM_BACKUP Function .....	4-27
4.2.13	The BUILD_SPDUMP Function.....	4-29
4.2.14	The BUILD_SPOOL Function.....	4-30



---

4.2.15	The BUILD_SYSDUMP Function .....	4-32
4.2.16	The BUILD_TVMF Function .....	4-34
4.2.17	The BUILDP Function.....	4-36
4.2.18	The BUILDP2 Function.....	4-38
4.2.19	The COPY_SITEFILES Function .....	4-41
4.2.20	The DELETE_MI Function.....	4-48
4.2.21	The DETECT_SITE Function .....	4-50
4.2.22	The DISPLAY_RUNSET Function.....	4-55
4.2.23	The DISPLAY_STATUS Function .....	4-57
4.2.24	The EDIT Function .....	4-59
4.2.25	The FSE Function.....	4-61
4.2.26	The LEVEL Function .....	4-63
4.2.27	The LEVEL_MI Function .....	4-65
4.2.28	The LIST_MI Function .....	4-66
4.2.29	The MODIFY_CONF Function .....	4-68
4.2.30	The PATCH_HLMLIB Function .....	4-70
4.2.31	The PATCH_SYSTEM Function.....	4-72
4.2.32	The PROTECT_GCOS Function.....	4-74
4.2.33	The RESTORE_SET Function .....	4-76
4.2.34	The RESTORE_SITE Function .....	4-78
4.2.35	The RESTORE_SITEFILES Function .....	4-80
4.2.36	The SAVE_SET Function .....	4-87
4.2.37	The SAVE_SITE Function .....	4-90
4.2.38	The SAVE_SITEFILES Function .....	4-92
4.2.39	The UPDATE_GCOS Function .....	4-95
4.2.40	The UPDATE_MI Function.....	4-101
4.2.41	The VALIDP Function.....	4-102
4.2.42	The VALIDP2 Function.....	4-105
4.2.43	The VALIDR Function.....	4-108
4.3	IUF Jobs .....	4-111
4.3.1	The INSTAL_ALL Job.....	4-111
4.3.2	The INSTAL_GCOS Job .....	4-113
4.3.3	The INSTAL_PSET Job.....	4-114
4.3.4	The INSTAL_RSET Job .....	4-115



## 5. CONFIG (Configuration)

5.1	Introduction .....	5-1
5.1.1	Input .....	5-2
5.1.2	Output .....	5-3
5.2	System Device Tables .....	5-4
5.3	Execution Procedure.....	5-5
5.3.1	Basic Procedure .....	5-5
5.3.2	Advanced Procedure .....	5-6
5.4	Configuration Statements .....	5-8
5.4.1	Syntax and Convention .....	5-8
5.4.2	Summary of Configuration Statements .....	5-9
5.4.3	The ACCOUNT Statement .....	5-12
5.4.4	The ACTSIZE Statement.....	5-14
5.4.5	The BJSIMU Statement.....	5-16
5.4.6	The DEFORG Statement.....	5-18
5.4.7	The DEVTRACE Statement .....	5-19
5.4.8	The DJPCTL Statement .....	5-21
5.4.9	The DUALSYS Statement .....	5-23
5.4.10	The DUMPACT Statement .....	5-25
5.4.11	The FILEOPT Statement.....	5-27
5.4.12	The FILSHARE Statement .....	5-30
5.4.13	The GAC Statement .....	5-32
5.4.14	The INPUTCTL Statement .....	5-34
5.4.15	The JOBCLASS and JBCLASS2 Statements .....	5-36
5.4.16	The JORSIZE Statement.....	5-45
5.4.17	The JPPCDIM Statement .....	5-46
5.4.18	The LSYS Statement.....	5-48
5.4.19	The MAXCAT Statement.....	5-49
5.4.20	The MAXDEF Statement .....	5-50
5.4.21	The MAXFILE Statement.....	5-51
5.4.22	The MAXIODEV Statement.....	5-52
5.4.23	The MAXJOB Statement .....	5-53
5.4.24	The MAXTAPE Statement.....	5-54
5.4.25	The MAXTASK Statement.....	5-55
5.4.26	The MULTLEV Statement .....	5-57
5.4.27	The OWCLASS Statement.....	5-58
5.4.28	The OWDEVICE Statement .....	5-60
5.4.29	The OWDFLT Statement.....	5-64
5.4.30	The PRLOG Statement .....	5-67



---

5.4.31	The PRLOGC Statement.....	5-69
5.4.32	The QUEUED Statement.....	5-71
5.4.33	The SADMOPT Statement.....	5-72
5.4.34	The SECOPT Statement.....	5-74
5.4.35	The VCAMSCN Statement.....	5-76
5.4.36	The VOLORG Statement.....	5-77
5.4.37	The XAS Statement.....	5-78
5.5	Default Values of CONFIG Statement.....	5-79
5.6	Output Listings.....	5-81
5.6.1	Explicit Configuration Commands.....	5-82
5.6.2	Memory Parameters.....	5-82
5.6.3	Scheduling Parameter Listing.....	5-83
5.6.4	Configuration Class Listing.....	5-83
5.6.5	Effective Configuration Commands.....	5-84
5.6.6	Successful Completion Message.....	5-88
5.6.7	Unsuccessful Completion Message(s).....	5-88
5.6.8	Hardware Configuration Listing.....	5-89
5.6.9	Error Messages.....	5-97
5.6.9.1	General Error Messages.....	5-97
5.6.9.2	Specific Error Messages.....	5-101

## 6. TAILOR

6.1	System Files.....	6-2
6.1.1	General Rules.....	6-2
6.1.2	System File Domains.....	6-2
6.2	Production Sets and Reference Set.....	6-3
6.3	TAILOR Constraints.....	6-4
6.3.1	Introduction to TAILOR.....	6-5
6.3.2	TAILOR and GCL.....	6-5
6.3.3	Structure of the TAILOR Command Language.....	6-6
6.4	Session Commands.....	6-8
6.5	Definition Commands.....	6-15
6.6	The GCOS Domain.....	6-23
6.6.1	Principal Parameters.....	6-24
6.6.2	File Definition Commands.....	6-25
6.7	The DSA Domain.....	6-56
6.7.1	Principal Parameters.....	6-56
6.7.2	File Definition Commands.....	6-57



6.8	The OLTD Domain .....	6-60
6.9	The FW Domain .....	6-61
6.10	The GSF Domain .....	6-62
6.11	The EOGEN Command .....	6-63
6.12	TLRGEN Session Considerations .....	6-64
6.12.1	Summary .....	6-64
6.12.2	Validation of a TLRGEN Session .....	6-64
6.12.3	Default TLRGEN Sessions .....	6-65
6.13	Context Rules .....	6-67
6.13.1	TAILOR and IUf .....	6-67
6.13.2	RESTART After Abort During File Creation .....	6-68
6.14	GCL Rules .....	6-69
6.15	Examples .....	6-70
6.15.1	TLRGEN Session .....	6-70
6.15.2	Default TLRGEN Session .....	6-75
6.15.3	TLREXEC Command Sequences .....	6-76
6.15.4	Dialogue at the MAIN Console .....	6-78
6.15.5	Examples of Errors .....	6-79
6.16	TAILOR Default Values and File Size Computation .....	6-81
6.16.1	GCOS File Domain .....	6-81
6.16.2	DSA File Domain .....	6-85
6.16.3	FW File Domain .....	6-86
6.16.4	GSF File Domain .....	6-86
6.17	Catalog Files: Size and Number of Objects .....	6-87

## 7. Updating Domains

7.1	Purpose .....	7-1
7.2	Set Technical Level .....	7-2
7.3	Updating Strategy in a PO-Configuration .....	7-2
7.4	Updating Strategy in a P2P-Configuration .....	7-3
7.5	Updating Strategy in an RP-Configuration .....	7-4
7.6	GCOS 7 Updating Procedure .....	7-5
7.7	GCOS 7 Updating Operations .....	7-7
7.8	After Journal Transfer .....	7-8





---

## 8. Patching Procedures

8.1	Purpose .....	8-1
8.2	GCOS 7 Patching Procedure.....	8-1
8.3	Operations .....	8-2

## 9. Firmware Generation (FGF) and Test and Diagnostics (OLTD) Facilities

9.1	Introduction .....	9-1
9.2	FGF (Firmware Generation Facility) .....	9-1
9.3	TDF (Tests and Diagnostics Facility).....	9-2
9.3.1	DPS 7000 (except DPS 7000/XTA).....	9-2
9.3.2	DPS 7000/XTA .....	9-2

## 10. DSA Installation and Updating Facility

10.1	Introduction .....	10-1
10.2	Environment .....	10-2
10.2.1	R-Set Considerations .....	10-3
10.2.2	DNS/CNS/FCP 7 Media .....	10-3
10.2.3	Technical Statuses and Releases .....	10-3
10.3	DIUF Tools .....	10-4
10.4	The ENTER_DIUF Command.....	10-5
10.5	Initial Installation .....	10-6
10.6	Add-on installation .....	10-9
10.6.1	PO-Configuration Add-On Procedure.....	10-9
10.6.2	P2P-Configuration Add-On Procedure .....	10-12
10.6.3	RP-Configuration Add-On Procedure.....	10-16
10.7	Updating .....	10-19
10.7.1	Updating Procedure Without Bringing Forward Patches.....	10-19
10.7.2	Updating Procedure With Patches Brought Forward .....	10-21
10.8	Patching .....	10-24
10.9	The DELETE_TS Function .....	10-26
10.10	Consulting .....	10-27
10.11	The LOAD_FACTORY Function.....	10-28
10.12	Patch Processor .....	10-30
10.13	The BUILD_TRCNCC Function .....	10-33



## 11. GCOS 7 Service Facilities (GSF)

11.1	Introduction .....	11-1
11.2	GSF Tools .....	11-2
11.3	GSF Release Media.....	11-2
11.4	The ENTER_GSF Command.....	11-3
11.5	Updating .....	11-4

## 12. Maintenance and Recovery Procedures

12.1	General Considerations .....	12-1
12.1.1	Disk Space Management .....	12-1
12.1.2	Recovery Procedures for Data Volumes .....	12-2
12.1.3	P2-Set/R-Set Disk and Updating Procedures .....	12-2
12.1.4	Tape, Cartridge, and Cassette Subsystems.....	12-2
12.1.5	MS/FSA Disk Subsystem .....	12-3
12.1.5.1	Large Configuration.....	12-3
12.1.5.2	Medium Configuration .....	12-3
12.1.5.3	Small Configuration.....	12-3
12.2	Updating .....	12-4
12.2.1	Large Configuration .....	12-4
12.2.2	Medium Configuration .....	12-4
12.2.3	Small Configuration .....	12-5
12.3	Slave Updating .....	12-6
12.3.1	Large Configuration .....	12-6
12.3.2	Small or Medium Configuration .....	12-6

## A. System File Descriptions

A.1	Introduction .....	A-1
A.2	The SITE.CATALOG File.....	A-5
A.3	The SITE.CMS_CSD File .....	A-6
A.4	The SITE.CMS_CDD_NAMEi File.....	A-7
A.5	The SITE.CMS_DS_NAME1_NAME2 File.....	A-8
A.6	The SITE.HALOCK File .....	A-9
A.7	The SITE.HELP File.....	A-10
A.8	The SITE.IN File .....	A-11
A.9	The SITE.MIRLOG File (if applicable) .....	A-12



---

A.10	The SITE.STARTUP File .....	A-13
A.11	THE SYS.BKST File .....	A-15
A.12	The SYS.BKST[i] File(s) .....	A-17
A.13	The SYS.BOOT File.....	A-20
A.14	The SYS.C.INCLUDE File .....	A-21
A.15	The SYS.CATALOG File.....	A-23
A.16	The SYS.CLC File .....	A-24
A.17	The SYS.DSABLIB File.....	A-25
A.18	The SYS.DSACMD File .....	A-26
A.19	The SYS.DSACONF File .....	A-28
A.20	The SYS.DSACORE File .....	A-29
A.21	The SYS.DSADUMP File.....	A-30
A.22	The SYS.DSALMLIB File .....	A-32
A.23	The SYS.DSALOG1 File.....	A-33
A.24	The SYS.DSALOG[i] Files .....	A-35
A.25	The SYS.DSASLIB File.....	A-37
A.26	The SYS.DSATRC_<LCT-NAME>_[i] Files (Specific to FCP 7) .....	A-38
A.27	The SYS.DSATRCIDT_NXYY File (Specific to an FCP 7 Technical Status) .....	A-40
A.28	The SYS.ERLOG File .....	A-42
A.29	The SYS.FW.BINLIB File.....	A-44
A.30	The SYS.FW.CCULIB File (not applicable for XTA systems).....	A-45
A.31	The SYS.FW.GENLIB File (not applicable for XTA systems).....	A-46
A.32	The SYS.FW.LMLIB File.....	A-47
A.33	The SYS.FW.MSP (not applicable for XTA systems) .....	A-48
A.34	The SYS.FW.NFTLIB File (not applicable for XTA Systems).....	A-49
A.35	The SYS.FW.PATCH File (not applicable for XTA Systems) .....	A-50
A.36	The SYS.FW.PRT File .....	A-51
A.37	The SYS.FW.SLLIB File .....	A-52
A.38	The SYS.FW.WSP (not applicable for XTA Systems).....	A-53
A.39	The SYS.GPL.MACLIB File .....	A-54
A.40	The SYS.GSF.BINLIB File .....	A-55
A.41	The SYS.GSF.LMLIB File .....	A-56
A.42	The SYS.GSF.SLLIB File.....	A-58



---

A.43	The SYS.GSF.SMLIB File.....	A-60
A.44	The SYS.GSF.UFAS File.....	A-62
A.45	The SYS.HBINLIB File.....	A-64
A.46	The SYS.HBINLIB2 File.....	A-66
A.47	The SYS.HCULIB File.....	A-67
A.48	The SYS.HELP File.....	A-69
A.49	The SYS.HLMLIB File.....	A-71
A.50	The SYS.HLMLIB2 File.....	A-73
A.51	The SYS.HRELLIB2 File.....	A-74
A.52	The SYS.HSLLIB File.....	A-75
A.53	The SYS.HSLLIB2 File.....	A-77
A.54	The SYS.HUB File (not applicable for XTA Systems).....	A-78
A.55	The SYS.HUB7 File (not applicable for XTA Systems).....	A-79
A.56	The SYS.HUBG File.....	A-80
A.57	The SYS.IN File.....	A-81
A.58	The SYS.IUF File.....	A-83
A.59	The SYS.JADIR File.....	A-85
A.60	The SYS.JASBLUE File.....	A-86
A.61	The SYS.JASGREEN File.....	A-87
A.62	The SYS.JRNAL File.....	A-88
A.63	The SYS.JTRAVAl[i] File.....	A-89
A.64	The SYS.KNODET File.....	A-90
A.65	The SYS.LIB[i] File(s).....	A-93
A.66	The SYS.LOGC File.....	A-94
A.67	The SYS.OUT File.....	A-95
A.68	The SYS.PVMF[i] File(s).....	A-98
A.69	The SYS.QM File.....	A-99
A.70	The SYS.QM_BACKUP File.....	A-100
A.71	The SYS.SITE.BIN File.....	A-101
A.72	The SYS.SITE.SL File.....	A-102
A.73	The SYS.SPDUMP File.....	A-103
A.74	The SYS.SPOOL[i] File(s).....	A-104
A.75	The SYS.SWLOG File.....	A-106



---

A.76	The SYS.SYSDUMP File .....	A-108
A.77	The SYS.SYSTEM File .....	A-110
A.78	The SYS.SYSTEM2 File .....	A-112
A.79	The SYS.TVMF[i] File(s) .....	A-113
A.80	The SYS.URCINIT File .....	A-114
A.81	System File Access Rights .....	A-115
A.82	Calculating Backing Store File Sizes .....	A-117
A.82.1	General Remarks.....	A-117
A.82.2	Paging Backing Store Files .....	A-118
A.82.3	Library Backing Store .....	A-118
A.82.4	Permanent Virtual Memory Files .....	A-119
A.82.5	Temporary Virtual Memory Files .....	A-119
A.82.6	Conversion of Megabytes to Blocks .....	A-119
A.82.7	Backing Store File Sizes.....	A-120

## **B. SIP Functions**

B.1	SIP (System Initialization Program) .....	B-1
B.2	SIP Loading .....	B-1
B.3	SIP Dialog .....	B-2
B.4	UTILity Functions Under SIP .....	B-3
B.5	Execution Example .....	B-3
B.5.1	The RESTORE Function .....	B-4
B.5.2	The SAVE Function .....	B-6
B.6	UTIL Error Messages.....	B-9

## **C. Disk VOLPREP Procedure**

C.1	Introduction .....	C-1
C.2	VOLPREP Activation .....	C-2



**D. Maintenance in a Protected Environment**

- D.1 Introduction .....D-1
- D.2 Cataloging System Files .....D-2
  - D.2.1 SYS.CATALOG Files (Group 1) .....D-2
  - D.2.2 SITE.CATALOG Files (Group 2) .....D-2
- D.3 Manipulating the SITE.CATALOG .....D-3
- D.4 The SYSADMIN Project.....D-4

**Glossary**

**Index**



---

## Table of Graphics

### Figures

Figure 2-1.	Installation Overview .....	2-6
Figure 2-2.	Installation: New Systems with Pre-loaded Disks .....	2-7
Figure 2-3.	Installation for Update Users .....	2-11
Figure 3-1.	Multi-System Installation .....	3-3
Figure 3-2.	Transport of the SET-MEDIA to the Slave System .....	3-4
Figure 3-3.	Multi-System Updating .....	3-6
Figure 3-4.	Master System Updating .....	3-8
Figure 3-5.	Slave System Update .....	3-10
Figure 4-1.	Set Evolution in a P2P configuration .....	4-99
Figure 4-2.	Set Evolution in an RP configuration .....	4-100
Figure 5-1.	CONFIG Overview .....	5-2
Figure 7-1.	Updating Strategy in a PO-Configuration .....	7-2
Figure 7-2.	Updating Strategy in a P2P-Configuration .....	7-3
Figure 7-3.	Updating Strategy in an RP-Configuration .....	7-4
Figure 8-1.	GCOS 7 Patching Procedures .....	8-3
Figure 10-1.	Initial Installation of DNS/CNS 7/FCP 7 via the DIUF Facility .....	10-8
Figure 10-2.	PO-Configuration Add-On Procedure .....	10-11
Figure 10-3.	P2P Configuration Add-On Procedure, using ADD_MI .....	10-14
Figure 10-4.	P2P Configuration Add-On Procedure, Without ADD_MI .....	10-15
Figure 10-5.	RP-Configuration Add-On Procedure .....	10-18
Figure 10-6.	Updating Procedure in the DIUF Facility .....	10-23
Figure 10-7.	Patching Procedure in the DIUF Facility .....	10-25
Figure 10-8.	Patching Procedure in the DIUF Facility .....	10-26
Figure 10-9.	Load Factory Procedure in the DIUF Facility .....	10-29
Figure 10-10.	Building Procedure for SYS.DSATRC_<lct-name>_x files .....	10-34
Figure 11-1.	Updating the GSF Domain .....	11-4



Tables

Table 1-1. IUF Functions for the GCOS Domain ..... 1-5

Table 2-1. Checklist for Installation Media ..... 2-3

Table 2-2. Information Contained in the SITE.CATALOG..... 2-4

Table 3-1. Slave Set Updating Decision Table ..... 3-10

Table 5-1. Default Values for Mono-Letter Job Classes ..... 5-44

Table 5-2. CONFIG Statement Default Values (1/2)..... 5-79

Table 5-3. Typical Explicit Configuration Command Listing..... 5-82

Table 5-4. Typical Scheduling Parameter Listing..... 5-83

Table 6-1. GCOS File Domain Default Values (1/4) ..... 6-81

Table 6-2. DSA File Domain Default Value ..... 6-85

Table 6-3. FW File Domain Default Values ..... 6-86

Table 6-4. GSF File Domain Default Values ..... 6-86

Table 6-5. File Size and Number of Objects on MS/FSA Volume ..... 6-87

Table 7-1. Files in PO-Configurations Affected by UPDATE\_GCOS..... 7-5

Table 7-2. Files in P2P & RP-Configurations Affected by UPDATE\_GCOS..... 7-5

Table 10-1. Request Actions According to the State of the KPN in the CLC..... 10-32

Table A-1. List of System Files ..... A-3

Table A-2. Backing Store Classification..... A-19

Table A-3. System File Access Rights (1/2) ..... A-115

Table A-4. Some Typical Backing Store File Sizes (in Blocks of 4096 Bytes) ..... A-120





---

# 1. Introduction

## 1.1 Overview

This manual describes the operations concerning GCOS 7 that may need to be carried out during the life cycle of DPS 7000/XTAxxx and the following DPS 7000 models:

- DPS 7000/TAxxx/TAxxxC
- DPS 7000/MTxx
- DPS 7000/8xx/Dx0/Mx0
- DPS 7000/4x5/Cx0

The installation procedure is specific to GCOS 7-TA/V9, to the systems listed above and to GCOS 7-XTA/V10. TA stands for the GCOS 7 Twin Architecture. XTA stands for the GCOS 7 eXtended Twin Architecture.

Users of DPS 7000/5x0/7xx and DPS 7000/4x0 should refer to the manual 47 A2 18US.

Users of DPS 7000/XTAxxx should refer to the manuals 47 A2 53US (GCOS 7 System Operator's Guide) and 47 A2 74US (V7000 Operator's Guide)

The operations described include:

- installation and migration from an earlier release (Installation and Updating Facility - IUF)
- creation of new system disk set configurations for the security of system files (IUF)
- customization of system disk sets (TAILOR)
- updating (UPDATE\_GCOS)
- patching (Emergency Corrections and PATCH)
- configuration (CONFIG)
- security (SAVE and RESTORE)



Depending on the model and the customer order, either systems are delivered with the GCOS 7 Release already installed or are delivered with an image of the GCOS 7 P-Set on tape, cartridge, or cassette, which is restored by the Bull Customer Service Engineer (CSE).

The delivered system disk configuration is referred to as a PO configuration. For security reasons, you are strongly advised to use one of the other configurations described below.

**NOTE:**

Information about DPS 7000/XTA models:

1. All references to VBO disks are irrelevant because these disks are no longer supported on these models.
2. For reasons of compatibility, the OLTD domain and its associated files still exist. However, their size has been reduced and there are no OLTD tests available on these models.
3. Annex B “SIP FUNCTIONS’ has no relevance for these models.
4. Seven firmware files have been deleted: SYS.FW.MSP, SYS.FW.WSP, SYS.FW.CCULIB, SYS.FW.NFTLIB, SYS.FW.PATCH, SYS.HUB and SYS.HUB7. The remaining ones are optimised in size.
5. All of the references to the CDP (Customer Dedicated Processor) are irrelevant.
6. The devclass “MT” no longer exists on these models.



**IMPORTANT:**

The use of mirror disks does not provide the required level of security for system disks.



### 1.1.1 System Disk Configurations

There are three configuration possibilities for system disks:

#### **PO Configuration**

In a PO configuration, the P-Set only is used. The P-Set may consist of one or several disks. This is the least expensive but also the **least secure** configuration.

#### **P2P Configuration**

This is the recommended configuration. A P2P configuration consists of two identical Production Sets in the form of a P-Set and a P2-Set. These Production Sets, which may each consist of several disks, work in alternate mode.

#### **RP Configuration**

An RP Configuration consists of a Production Set (P-Set) and a Reference Set (R-Set). The R-Set contains all the files necessary for IUF operation and can be used to create a P-Set by means of the BUILDP function (see the chapter *Installation and Updating Facility*).

### 1.1.2 System Disks and Resident Disks

A disk containing system files is referred to as a SYSTEM disk. Such disks must be FSA (Fixed Sector Architecture) in all cases.

Resident disks must generally be FBO (Fixed Block Organization).

However, on certain DPS 7000 models, as listed below, VBO (Variable Block Organization) is supported but FBO is to be preferred wherever possible:

DPS 7000/TAxxx/TAxxxC  
DPS 7000/8xx/Dx0/Mx0  
DPS 7000/MT42/52/62 and up

A disk which is declared RESIDENT must be permanently on line.



## 1.2 Operating System Maintenance - IUF

The Installation and Updating Facility (IUF) provides a set of functions and jobs to help install and maintain the GCOS 7 operating system. The facility is called by the GCL command GIUF. For full details, see the chapter *Installation and Updating Facility*.

### 1.2.1 Features of IUF

1. IUF supports multi-device classes in the following combinations:
  - between volumes of a given disk Set (except for DPS 7000/4x5/Cx0/MTx1)
  - between disk Sets of a given system (except for DPS 7000/4x5/Cx0/MTx1)
  - between disk Sets of several systems in a multi-system environment.GCOS 7 supports coexistence of FBO file organization and VBO file organization. **However, system files (that contain all the GCOS Domains) must be located on FSA volumes.**
2. IUF functions support multiple-MI software configurations in a multi-system environment.
3. IUF takes advantage of the GCL (GCOS 7 Command Language), the default language for the MAIN operator. The commands described in this document use GCL syntax.
4. IUF supports cartridges and tapes. The device classes are, respectively, CT and MT.

Throughout this document cartridges are included in all references to tapes. The term **tape (cartridge)** means: magnetic tape with the device class MT, or cartridge with the device class CT.

- DPS 7000/4x5/Cx0/MTx1 support 8mm cartridges (CT/M6).
- Other models support 8mm cassettes (CT/M6), half-inch cartridges (CT/M5 or CT/36T), or tapes (MT/T9), according to the order.

In certain cases, these systems can support emulated DLT cassette systems as 8mm cassettes (CT/M6).

See the Preface for a full list of tape, cartridge, and cassette specifications.



## 1.2.2 IUF Functions

Table 1-1 lists the functions available in IUF.

**Table 1-1. IUF Functions for the GCOS Domain**

Procedures	Installation	Updating	Patching
Major IUF Functions	VALIDP VALIDP2 BUILDP2	UPDATE_GCOS LEVEL	PATCH_HLMLIB PATCH_SYSTEM
Other IUF Functions	ADAPT ADD_MI BUILDP BUILD_BKST BUILD_CLC BUILD_CMS BUILD_HALOCK BUILD_JAS BUILD_JRNAL BUILD_LIB BUILD_MIRLOG BUILD_PVMF BUILD_QM BUILD_QM_BACKUP BUILD_SPDUMP BUILD_SPOOL BUILD_SYSDUMP BUILD_TVMF	COPY_SITEFILES DELETE_MI DETECT_SITE DISPLAY_STATUS EDIT FSE LEVEL_MI LIST_MI MODIFY_CONF PROTECT_GCOS RESTORE_SITE RESTORE_SITEFILES SAVE_SET RESTORE_SET SAVE_SITE SAVE_SITEFILES UPDATE_MI VALIDR	
IUF Function in H_NOCTX	DISPLAY_RUNSET	This function does not belong to the GIUF processor but is nevertheless documented with the other GCL commands in the chapter <i>Installation and Updating Facility</i>	

## 1.2.3 IUF Jobs

The following IUF jobs are fully described in the chapter *Installation and Updating Facility*:

```

INSTAL_ALL
INSTAL_GCOS
INSTAL_PSET
INSTAL_RSET

```



---

#### 1.2.4 GCOS 7 Domains

IUF acts on GCOS 7 in terms of its components, which are known as Domains. The Domains are:

<b>GCOS</b>	General Comprehensive Operating System
<b>FW</b>	Firmware, also called BSR (Basic System Release)
<b>OLTD</b>	On Line Test and Diagnostics
<b>DSA</b>	Distributed Systems Architecture, for the support of the GCOS 7 telecommunications facilities. This includes CNS7 software for CNP7 front-end processors, DNS for Datanets, and/or OCS (FCP7 software).
<b>GSF</b>	GCOS 7 Service Facility



---

## 2. Installation Procedure

Installation is carried out in all circumstances by the Bull Customer Service Engineer (CSE). This chapter gives an example of the installation procedure for a new system with pre-loaded disk followed by an example of the update of a system running under an earlier version of GCOS 7. **These descriptions are to be taken only as examples.** Each DPS 7000 model and each configuration has its own particularities which is why the task is now carried out by the CSE.

### 2.1 Before Beginning

Before beginning the installation procedure:

- Make sure that you have the minimum hardware as described in *Hardware and Firmware Requirements* below.
- Make sure that you have all the installation media summarized in Table 2-1 below.
- Decide on the number of FSA disks to be formatted as future system Set(s). See *Hardware and Firmware Requirements* below for the minimum requirements. A minimum P-Set built with one each of SYS.SPOOL(0), SYS.BKST(0), SYS.PVMF(0), SYS.TVMF(0), and SYS.LIB(0) requires about 0.8 to 1 Gbyte, depending on the number and type of the disks used.
- Make sure that enough disks are free for P-Set installation (see *Hardware and Firmware Requirements* below).
- Check that there is a minimum of 64 Mbytes of memory available.
- Check that there is a tape or cartridge drive connected that supports system functions. This is for restoring installation media, and for bootloading the system.
- Refer to the chapter Recovery Procedure.

New users requiring a P2P-Configuration must prepare the TAILOR parameters for use in the BUILD P2 function which creates the P2-Set.



---

## 2.1.1 Hardware and Firmware Requirements

### 2.1.1.1 On DPS 7000/4x5/Cx0/MTx1 Systems

- The installed Basic System Release (BSR) must be at least the appropriate N1 release (P2 for MTx1).
- System files are supported only on:
  - integrated MS/FSA disks located in the central cabinet. At least 1.4 Gbyte or 2.5 Gbyte disk is necessary to build the P-Set,
  - external RAID1 or RAIDS MS/FSA disks (other than those located in a Large Storage Subsystem (LSS)). Only one disk is necessary to build the P-Set.
- An 8mm cassette drive is required for reading the shipped media and for the bootload function (CT/M6).

### 2.1.1.2 On DPS 7000/TAxXX/TAxXXC/8xx/Dx0/Mx0, and DPS 7000/MTx2 Systems and up

- The installed Basic System Release (BSR) must be at least the appropriate P2 release.
- System files are supported only on:
  - a Large Storage Subsystem (LSS),
  - external RAID1 or RAIDS MS/FSA disks (other than those located in an LSS). Only one disk is necessary to build the P-Set.
- Any of the following: half-inch tape drive (MT/T9), half-inch cartridge drive (CT/M5, CT/LIB/M5, CT/36T, or CT/LIB/36T), or 8mm cassette (CT/M6), is capable of reading the shipped media and the bootload function.





### 2.1.2 Installation Media Required

Table 2-1 gives a checklist of the necessary installation media for the two cases discussed in this chapter. For other installation circumstances, check with your Bull CSE.

**Table 2-1. Checklist for Installation Media**

<b>Installation Media</b>	<b>New Users with P-Set pre-loaded disk(s)</b>	<b>Update Users with GCOS 7 minimum V7 TS 7356</b>
SHIP-MEDIA	YES	NO
FW-MEDIA	YES (backup)	NO
DOMAINS-MEDIA	YES (backup)	NO
K-MEDIA	YES	YES (mandatory)
P-MEDIA	YES (backup)	NO
P-SET	YES	YES (previous)
G-MEDIA	NO	YES

Migration from GCOS 7–V7 TS 7356 (and higher TSs) or from GCOS 7–V8 any Technical Statuses to GCOS 7–V9 needs to install a new complete validation key provided with the GCOS 7–V9 product kit. (G-MEDIA + K-MEDIA).

Migration from GCOS 7–V7 TS 7356 (and higher TSs) or from GCOS 7–V8 or from GCOS 7-V9 any Technical Statuses to GCOS 7–V10 needs to install a new complete validation key provided with the GCOS 7–V10 product kit. (G-MEDIA + K-MEDIA).



### 2.1.3 Initial Profiles

When you start a session at the system console by introducing a break, a message is displayed asking for the user name and password. The SITE.CATALOG that has been delivered is validated, and contains the information given in Table 2-2.

**Table 2-2. Information Contained in the SITE.CATALOG**

PROJECT	USER	PASSWORD	PARAMETERS	BILLING
SYSADMIN	SYSADMIN	SYS	NDFLT - IOF	-
SYSADMIN	GCOS7	G7	DFLT - IOF	-
OPERATOR	OPERATOR	OP	MAIN - DFLT - IOF	-
RMS	RMS	empty	MAIN - DFLT - IOF	INSTALL
RMSIOF	RMSIOF	empty	DFLT - IOF	INSTALL

The initial profile of each operator logging onto the system is:

```
GCLMODE=GCL
OPERATING MODE=CONS
GCLFORM=LINE
NOVICE=1
MENU=1
```

**NOTE:**

GCL mode is the default for all logons. To switch to JCL mode, enter:

```
S: MDP GCLMODE=JCL;
```

To return to GCL mode, enter:

```
S: GCL;
```



## 2.2 P-MEDIA Restoration Procedure

If the P-Set cannot be used for some reason, this procedure should be performed by the Bull CSE only as a backup. The delivered P-MEDIA is a VOLSAVE of your P-Set.

### 2.2.1 Systems not XTA

1. Initialize the system and load SIP from the hard disk. Refer to the appendix *SIP and UTIL Mode* for information on SIP.
2. Make a first work disk available. This disk must belong to the same disk family as the delivered P-MEDIA.
3. For each disk, use the RESTORE function of SIP in UTIL mode in order to restore the saved P-Set:

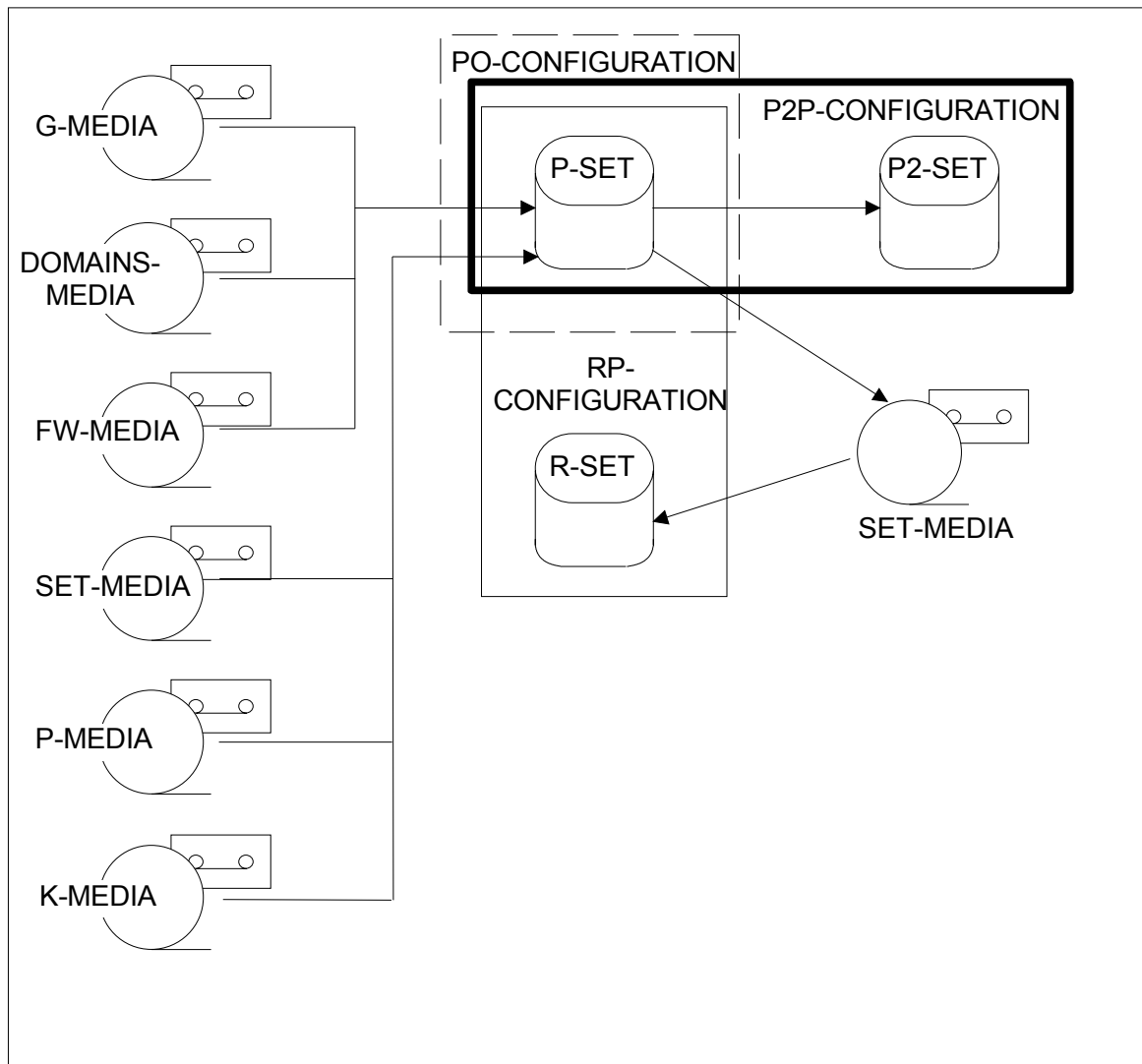
R MSxx

### 2.2.2 XTA Systems

1. Initialize the system and load GCOS system from the bootstrap disk (B-DISK).
2. For each disk use the RESTORE utility of GCOS in order to restore the saved P-Set..

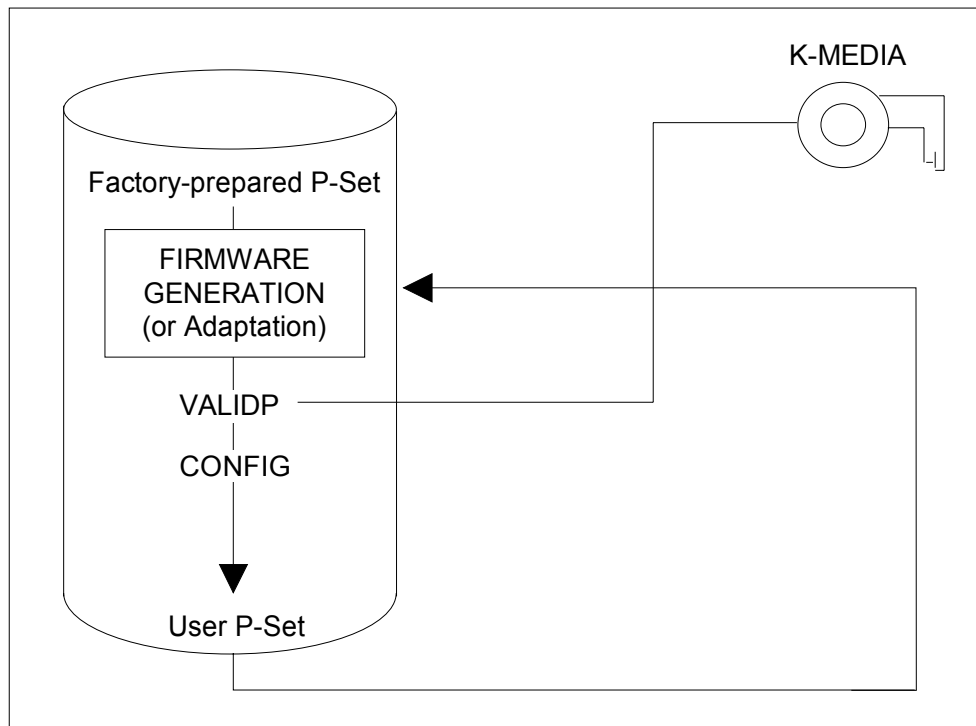
## 2.3 Installation on a New DPS 7000 With Pre-loaded Disk

The IUF installation procedure is used by the Bull CSE to build a GCOS 7 operating system from the Production Set (P-Set). The P-Set contains all the system files, including the Domains files. Figure 2-1 shows an overview of the installation.



**Figure 2-1. Installation Overview**

You are mandatorily supplied with a K-MEDIA for validation of software Marketing Identifiers (MIs).



**Figure 2-2. Installation: New Systems with Pre-loaded Disks**

### Systems With Securized Disks

On systems with securized disk subsystems, the sequence is:

1. UTIL (not available on XTA systems)  
RESTORE
2. INIT
3. RESTORE (/), RESTART (CLEAN), PVMF (/), TVMF (/), BKST (/),  
LIB (/), RESIDENT (/)
4. FGF BUILDP2
5. TSYS + TERMINATE SYSTEM
6. INIT (new disk)
7. RESTORE (/), RESTART (CLEAN), PVMF (/), TVMF (/), BKST (/),  
LIB (/), RESIDENT (/)

Follow this with the steps 3 through 10 of the case "Standard Disks" below.



### Systems With Standard Disks

On systems with standard disks, the sequence is:

1. INIT
2. RESTORE (/) , RESTART (CLEAN) , PVMF (/) , TVMF (/) , BKST (/) ,  
LIB (/) , RESIDENT (/)
3. FGF  
MNCONF  
FIRMGEN  
BLOAD
4. TSYS + TERMINATE SYSTEM
5. INIT
6. RESTORE
7. EJ CONFIG
8. TSYS
9. RL
10. RESTORE, RESTART (COLD)
11. GIUF MDC CURCONF=P2P NEW\_CONF=P0  
then perform a VOLPREP on the temporary P-Set so that the disk has the  
correct size and characteristics.  
See the chapter *Installation and Updating Facility* for a description of the  
MODIFY\_CONF function.
12. Reload the system:  
I : RESTORE , RESTART (COLD)
13. Run on-line tests and diagnostics (OLTD).
14. Run the relevant off-line tests (TOLs).
15. For a hardware upgrade when the CPU has been changed and the system disks  
have been reconnected, customize the site files SITE.CATALOG, SITE.HELP,  
and SYS.URCINIT. The SITE.\* files are copied from the old system disk.  
The IUF function COPY\_SITEFILES is described in the chapter *Installation  
and Updating Facility*.  
See also the appendix *Maintenance in a Protected Environment* for the use of  
GIUF with access rights.
16. Use the CONFIG utility described in the chapter *Configuration* to customize  
the configuration.



17. (Not applicable when the software MIs have been validated in the factory.)

Validate the P-Set as follows:

```
S: GIUF FUNC=VALIDP;
```

Parameters are requested on the screen, using GCL line mode.

Refer to the chapter *Installation and Updating Facility* for a description of the GIUF function VALIDP.

Details of the KEY location can be provided by the Local Distribution Center. (The KEY validates the products that you have ordered.)

18. You are recommended to save the whole P-Set to tape or cartridge, as appropriate. Use the GCL procedure SAVE\_DISK.

Now proceed to the description later in this chapter to create a P2P or RP-Configuration.

The basic P-set is now ready to use, and:

- This P-set can be customized (number and size of the BKST, TVMF, PVMF, LIB, Spool file), if a P0 is requested.
- Build a customized P2-set (in terms of the number and size of the BKST, TVMF, PVMF, LIB, Spool file), and use this P-set as backup.
- Build a customized P2-set (in terms of the number and size of the BKST, TVMF, PVMF, LIB, Spool file), clean this current P-set, and build a new identical customized P2-set, if a config with 2 identical system sets is chosen.

**NOTE:**

If a previous system disk is present on the site, the new set could be updated using the COPY\_SITEFILES function.



## 2.4 Installation for Update of GCOS 7 Release

Installation for update of a GCOS 7 Release can be done by the Bull CSE either as described below or, using an image of the P-Set, in exactly the same way as described above for a first delivery.

### 2.4.1 Backup To Previous Release

The time necessary for switching the processing from one Release to another is shorter when the old P-Set is kept on line. Therefore, you should decide if old P-Set disks are to be kept on line after installation of the new Release. If the old P-Set is not to be kept on line, it should be saved before starting the installation procedure.

The old P-Set is required on line during installation to copy the SITE files from it to the future P-Set. If the old P-Set cannot be kept on line, the SITE files may be saved to tape (or cartridge) before building the new P-Set.

Refer to the end of this chapter for a description of the backup procedure.

### 2.4.2 Constraints

See the paragraph "*Before Beginning*" above for details of the required memory size, number of disks for the P-Set, and minimum BSR level.

System files must not be installed on QUOTA-controlled disks.

Only disks defined in *Hardware and Software Requirements* above for each type of system are supported.

### 2.4.3 Alternative Installation Procedures

The Local Distribution Center can provide you with a SET-MEDIA, created by the IUF function SAVE\_SET, that contains the GCOS 7 Release and the associated Domains necessary to build your P-Set or R-Set. This SET-MEDIA is to be used with the INSTAL\_PSET or the INSTAL\_RSET jobs.

The factory can provide you with a P-MEDIA in the form of a VOLSAVE of one or several disks to be used with the GCL command RESTORE\_DISK (RSTD).

The Local Distribution Center can provide you with a G-MEDIA that you can use to update the GCOS 7 Release for the GCOS Domain only.



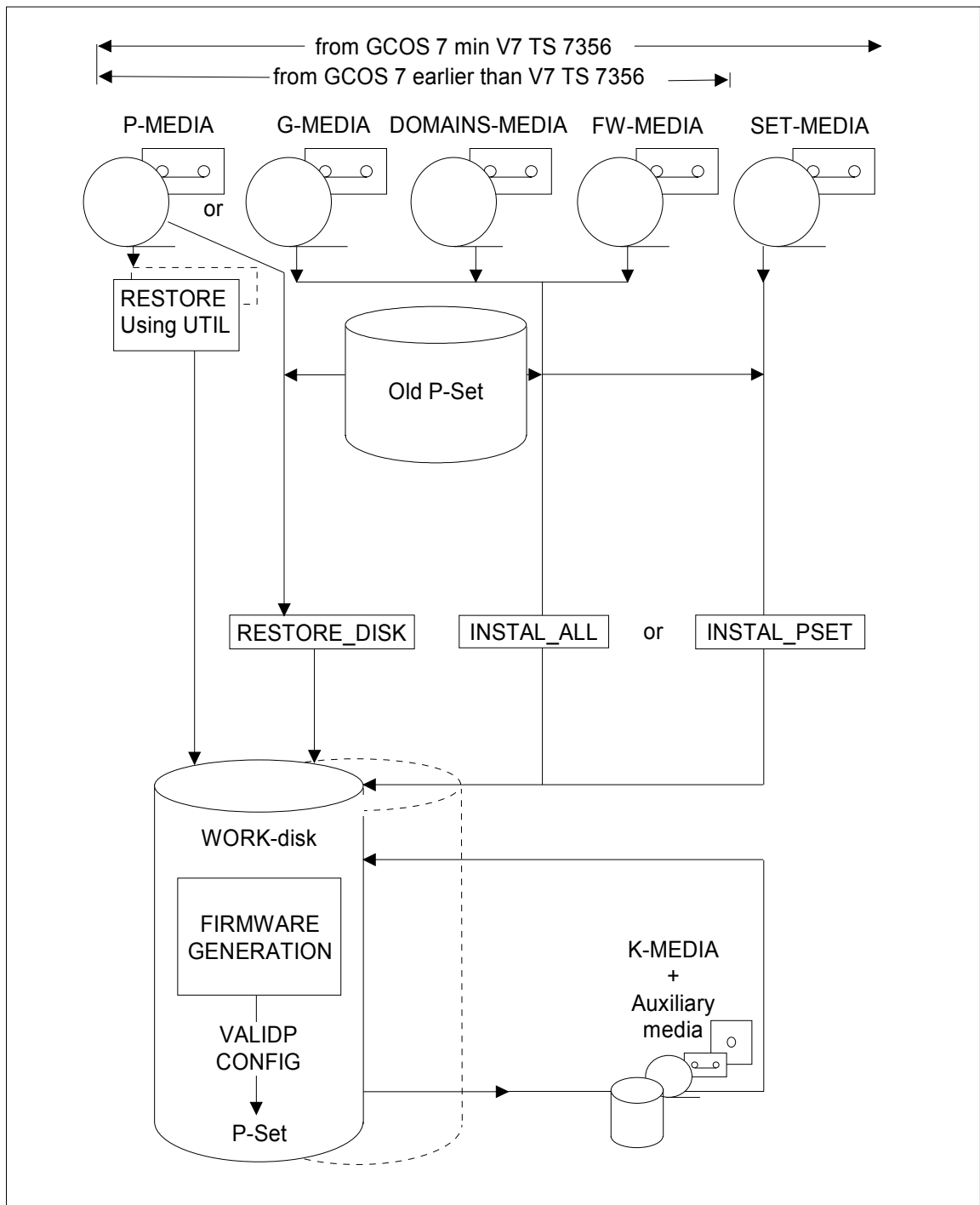


Figure 2-3. Installation for Update Users

**NOTE:**

The old P-Set must be loaded and running and the BSR supporting the GCOS 7 Release must already be installed.

1. Make available the required number of prepared work disks that are to become the P-Set.

It is assumed that the VOLORG parameter is set at ALL. If this is not the case, start the CONFIG job with VOLORG=ALL, terminate the system (TSYS 1 under GCOS 7 and Terminate System under the Service Processor) and start it again, as follows:

```
I: RESTORE, RESTART (COLD)
```

```
In response to the question: NG03 DO YOU WANT TO LOAD
PREVIOUSLY SAVED NETWORK AND TERMINAL
CONFIGURATIONS? (Y,N) , DEFAULT=Y, answer Yes.
```

2. Mount the G-MEDIA, FW-MEDIA, SET-MEDIA, DOMAINS-MEDIA or the P-MEDIA, as appropriate.
3. Start the INSTAL\_ALL job or INSTAL\_PSET job or RESTORE\_DISK command with, as appropriate:

```
S: EJ INFILE=INSTAL_ALL:G-md:<tape-dvc>$MFT;
```

or if the current Technical Status is >=TS 7356 and SET-MEDIA is available:

```
S: EJ INFILE=INSTAL_PSET:set-md:<tape-dvc>$MFT;
```

or

```
S: RSTD;
```

See the Preface for full syntax specification of tape/cartridge drives.

INSTAL\_ALL or INSTAL\_PSET then requests information on the screen.

INSTAL\_ALL or INSTAL\_PSET clears the disk(s) of the future P-Set, then allocates space for it and restores the Domains files to it. This job requires user SYSADMIN rights in the running SITE.CATALOG.

INSTAL\_ALL and INSTAL\_PSET execute the CONFIG job which updates the System Device Tables in the SYS.SYSTEM file on the newly created system volume.

If RSTD has been used, give values on the RESTORE\_DISK menu for the disk volume identification and the tape file image.

For more information on the GCL command RESTORE\_DISK, refer to the *IOF Terminal User's Reference Manual, Part 2*.



4. Perform the following:

```
LEVEL DOMAIN=FW
```

5. Terminate the system (TSYS 1 under GCOS 7 and Terminate System under the Service Processor).
6. Initialize the system using the boot load on the old P-Set and load GCOS 7 from the newly-created P-Set, as follows:

```
I:RESTORE (/) , RESTART (CLEAN) , PVMF (/) , TVMF (/) , LIB (/) , BKST (/)
```

For other ISL parameters, refer to the *GCOS 7 System Operator's Guide*.

Note that the complete RESTORE syntax need only be used once; subsequent RESTOREs may be run with:

```
I: RESTORE
```

as ISL commands are memorized by the system from one ISL session to the next. Note, however, that the disk supporting the SYS.SYSTEM file must be on line, even for a RESTART WARM.

The question: NG03 : DO YOU WANT TO LOAD PREVIOUSLY SAVED NETWORK AND TERMINAL CONFIGURATIONS? (Y,N) , DEFAULT=Y should be answered with "Y".

7. You may copy the SITE files (SITE.CATALOG, SITE.STARTUP, SITE.HELP, SYS.URCINIT) from the old P-Set.



**IMPORTANT:**

Do not copy help texts in French to the SYS.HELP file as they are incompatible. French help texts are stored only in the SITE.HELP.

Refer to the chapter *Installation and Updating Facility*, for a description of the IUF function COPY\_SITEFILES. Refer also to the appendix *Maintenance in a Protected Environment* for details of the use of GIUF with access rights.

The P-Set is now ready for use, once you have installed the software key (K-MEDIA), personalized the SITE files, configured the network and performed software optimization, using CONFIG job parameters. For details of CONFIG job parameters, see the chapter *CONFIG (Configuration)*.

The size of some of the system files resulting from the installation procedure of the PO-Configuration may not be convenient for the user's processing requirements. Additional files may be allocated using the IUF function BUILD\_filename, where filename = BKST, LIB, PVMF, SPOOL or TVMF.

8. You are recommended to save the whole P-Set to tape, cartridge, or cassette, as appropriate. Use the GCL procedure SAVE\_DISK.



## 2.5 Creating a P2P Configuration



### IMPORTANT:

In a P2P Configuration, the same rules apply for system files and resident disks as for the P-Set. See *System and Resident Disks* in the Introduction.

All files for any given Domain must be on the same volume.

1. Make available the required number of work disks. They must be FSA disks for the SYSVOL, and FBO (or possibly VBO) for RSDVOL, RSDVOL1, NRDVOL, NRDVOL1.

These work disks will become the P2-Set.

2. Make available the P2-Set TAILOR parameters.

TAILOR parameters may be stored in a command file from a library, or introduced during an interactive TAILOR session using the BUILD P2 function. Refer to Chapter 6 *Tailor* for full details.

3. Start the IUF function BUILD P2, as follows:

```
S: GIUF FUNC=BUILD P2;
```

or:

```
S: GIUF CMD='BUILD P2 COMFILE=file-name';
```

For example:

```
S: GIUF CMD='BUILD P2 COMFILE=MYLIB..TAILOR';
```

Refer to the IUF function BUILD P2 described in the chapter *Installation and Updating Facility*, as well as Chapter 6 *Tailor*.

4. Terminate the system (TSYS 1 under GCOS 7 and Terminate System under the Service Processor).



5. Initialize the system and load GCOS 7 from the newly created P2-Set as follows:

```
I:RESTORE(/),RESTART(CLEAN),PVMF(/),TVMF(/),LIB(/),BKST(/)
```

If BKST, PVMF, TVMF, and/or LIB files are not located on the system disk, the corresponding "/" must be replaced by the relevant media name. For example, PVMF(pvmf-md). This does not apply to BKST0 which is always on the system disk.

For other ISL parameters, refer to the *GCOS 7 System Operator's Guide*.

The question: NG03 DO YOU WANT TO LOAD PREVIOUSLY SAVED NETWORK AND TERMINAL CONFIGURATIONS? (Y,N),  
DEFAULT=Y should be answered with "Y".

6. You are recommended to save the whole P2-Set to tape or cartridge as appropriate. Use the GCL procedure SAVE\_DISK.



## 2.6 Creating an RP-Configuration



### IMPORTANT:

All system files must be on an FSA disk.

All files for any given Domain must be on the same volume.

1. Execute the IUF function SAVE\_SET to create the SET-MEDIA from the running P-Set and reply to the questions on the menu. All GCOS 7 Domains must be saved.

For further details, refer to the IUF function SAVE\_SET in the chapter *Installation and Updating Facility*.

2. Make a "prepared" work disk (FSA) available.

This work disk will become the R-Set.

Note that a complete VOLPREP (or PREPARE\_DISK) is necessary for a work disk that has previously been used as a system disk. For work disk preparation, refer to the appendix *Disk VOLPREP Procedure*.

3. Start the IUF job INSTAL\_RSET, as follows:

```
S: EJ INFILE=INSTAL_RSET:set-md:<tape-dvc>$MFT;
```

where set-md is the name of the first R-SET disk. For example:

```
S: EJ INFILE=INSTAL_RSET:ABCFB1:CT/M5$MFT;
```

See the Preface for full syntax specification of tape/cartridge drives.

INSTAL\_RSET then requests information on the screen.

INSTAL\_RSET clears the future R-Set before allocating space for it and then restoring to it the system files for all the Domains.

INSTAL\_RSET executes the CONFIG job which updates the System Device Tables in the SYS.SYSTEM file on the newly created volume.

4. Terminate the system (TSYS 1 under GCOS 7 and Terminate System under the Service Processor).
5. Initialize the system and load GCOS 7 from the newly created R-Set, as follows:

```
I:RESTORE,RESTART(CLEAN),TRC(OFF),BKST(/),PVMF(/),TVMF(/),LIB(/)
```

The question: NG03: DO YOU WANT TO LOAD PREVIOUSLY SAVED NETWORK AND TERMINAL CONFIGURATIONS? (Y,N), DEFAULT=Y should be answered with "Y".



During GCOS 7 loading, a message is displayed:

```
JL15 SYSTEM WITHOUT BEFORE JOURNAL. IF ABNORMAL RUN  
PRLOG RC=DFPRE 6, EFNUNKN
```

This message is sent because there is no SYS.JRNAL file on the R-Set. It may be safely ignored because a SYS.JRNAL is unnecessary on the R-Set.

6. TSYS + TERMINATE SYSTEM from the Service Processor.
7. Reinitialize the P-Set, as follows:

```
I:RESTORE,RESTART(CLEAN),TRC(OFF),BKST(/),PVMF(/),TVMF(/),LIB(/)
```

8. Execute the IUF function MODIFY\_CONF, to transform the PO-Configuration into an RP-Configuration, as follows:

```
S: GIUF FUNC=MODIFY_CONF;
```

Parameters are then requested on the screen in GCL line mode.

Refer to the IUF function MODIFY\_CONF described in the chapter *Installation and Updating Facility*.

9. You are recommended to save the whole R-Set to tape or cartridge, as appropriate. Use the GCL procedure SAVE\_DISK.



---

## 2.7 BackUp to a Previous Release in P0

1. Execute the RESTORE\_SET function from the multivolume SET\_MEDIA created by the SAVE\_SET function when the previous release GCOS 7 was running.
2. Initialize the system and load GCOS 7 from the restored P-set.

## 2.8 BackUp to a Previous Release in P2P

Reload the previous version. Set "non-modified".





---

## 3. Multi-System Installation and Updating

### 3.1 Introduction

#### 3.1.1 Purpose of this Procedure

This procedure enables you to install and maintain system disks belonging to more than one DPS 7000 system.

#### 3.1.2 Master and Slave Systems

A Master disk Set is created on the Master system and used to update disk Sets for all the Slave systems. These procedures are provided from magnetic tape or cartridge.

The Master and Slave systems have fixed system disks. The systems are not necessarily located at the same site. A tape or cartridge image of a Master Set is transported to the Slave system to update the Slave Set.



### 3.1.3 General Requirements

1. The Master and Slave systems may comprise any mix of DPS 7000 systems all running under the same GCOS 7 Release.
2. Ensure that the type of media used (tape and/or cartridge) is common to both Master and Slave systems.
3. Multi-system installation supports PO, P2P, and RP Configurations.



**IMPORTANT:**

If a pair of coupled systems uses the CONFIG statement DUALSYS with the parameter MYSELF, the two systems cannot share one common R-Set. This is because recovery from the R-Set is not possible on the Slave system. The solution must be one of the following:

(Preferably) two P2P-Configurations.

Two RP-Configurations.

Two PO-Configurations (since there is no R-Set in a PO Configuration, this is the least expensive configuration, but also the least secure of the three).



### 3.2 Installation Procedure

The installation procedure of a Multi-system Set contains two steps:

1. The "Master" step, during which the P-Set, P2-Set or R-Set (as applicable) of the Master system are built.
2. The "Slave" step, in which Slave Set(s) are built for the Slave system(s).

The installation on the Master is performed by Bull personnel in the same way as for an individual system (see the chapter *Installation Procedure*). Figure 3-1 shows an overview of the master and slave system installation.

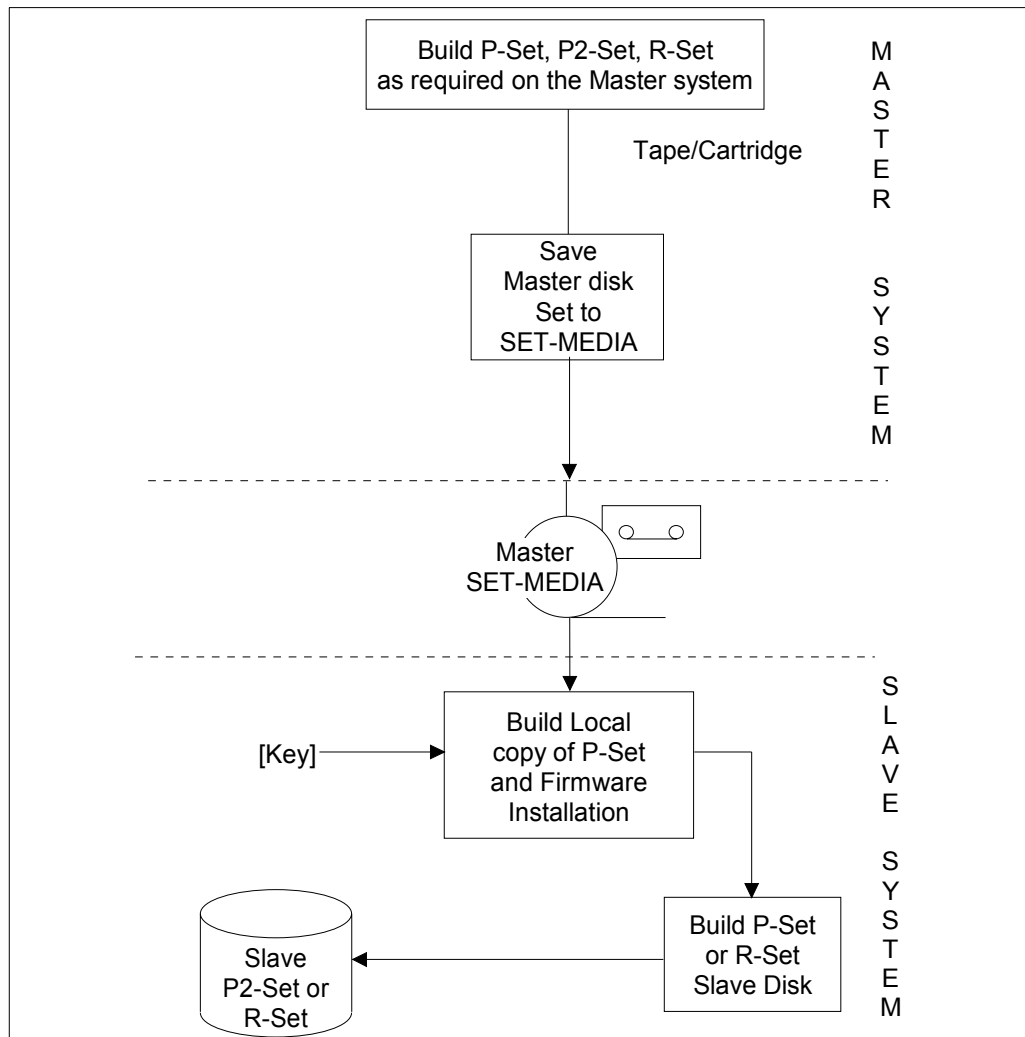
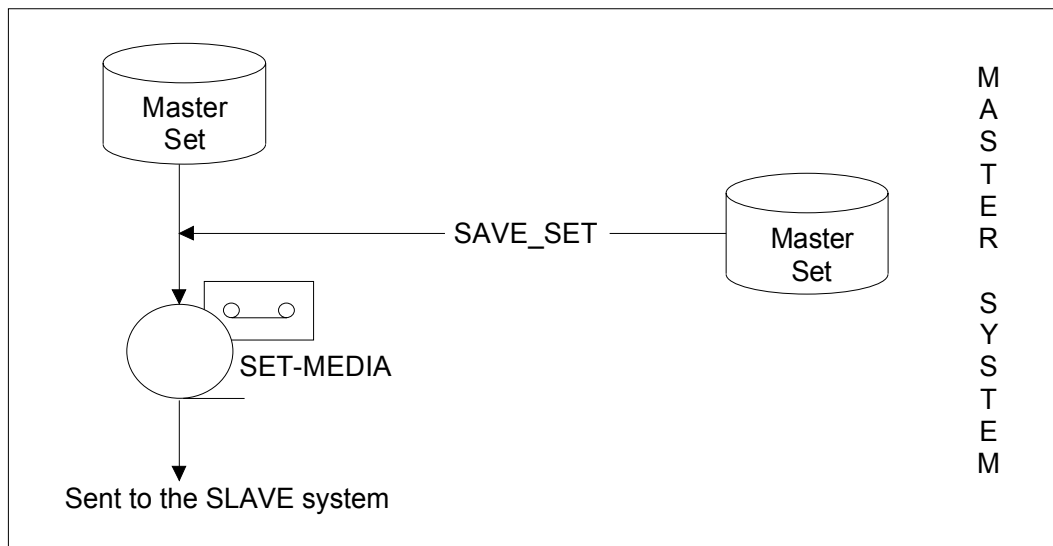


Figure 3-1. Multi-System Installation



### 3.2.1 Saving a Set on the Master System

1. These operations may be performed during normal processing. A non-running Master Set must be on line, for use as input to the IUF function `SAVE_SET`. The disk(s) may be a P-Set, a P2-Set, or an R-Set. All GCOS 7 Domains must be saved.
2. Execute the IUF function `SAVE_SET` to create the SET-MEDIA, using the MEDIA name convention `abcdex` where:
  - `abcde` is the user tape/cartridge prefix (in alphanumeric characters), as requested by the `SAVE_SET` function.
  - `x` is the tape/cartridge reel sequence number.
 For further details, refer to the IUF function `SAVE_SET` in the chapter *Installation and Updating Facility*.
3. Transport the SET-MEDIA to the Slave system. See Figure 3-2.



**Figure 3-2. Transport of the SET-MEDIA to the Slave System**

### 3.2.2 Installation on the Slave System

Use the saved Set for installation on each slave system.

See the chapter *Installation Procedure* for memory, hardware, and firmware requirements. The same chapter also gives an installation example but remember that each DPS 7000 model and each configuration may have specific requirements not dealt with in the example.



## 3.3 Updating

### 3.3.1 Purpose

The updating procedure consists essentially of applying GCOS 7, OLTD, FW, DSA, and GSF Technical Status, as well as software add-ons or deletions for the GCOS Domain.

In a multi-system environment, the updating strategy is as follows:

1. Apply the update to the Master System (P-Set or P2-Set, and R-Set, as applicable).
2. Save the correctable system files of all GCOS 7 Domains from a Master Set to a SET-MEDIA, using the IUF function SAVE\_SET.
3. Update the Slave P-Set, or P2-Set, with the IUF function RESTORE\_SET, from the SET-MEDIA.

For Slave P2P-Configurations, and for multiple-software MI configurations, the Updating procedure takes advantage of the opportunity of updating the non-running Set(s).

For multiple MI configurations, add-on or deletion must take place separately for each Slave system Set(s).

GCOS 7 Emergency Corrections installed on a Slave Set are dynamically managed if they have been installed on the non-updated Set.

If SITE=1 is specified in the IUF function RESTORE\_SET, the SITE patches will be re-applied.

In this case, the ECs are extracted from the Set concerned, prior to updating, and re-applied to the same Slave Set by the IUF function RESTORE\_SET after restore termination if SITE=1 has been specified.

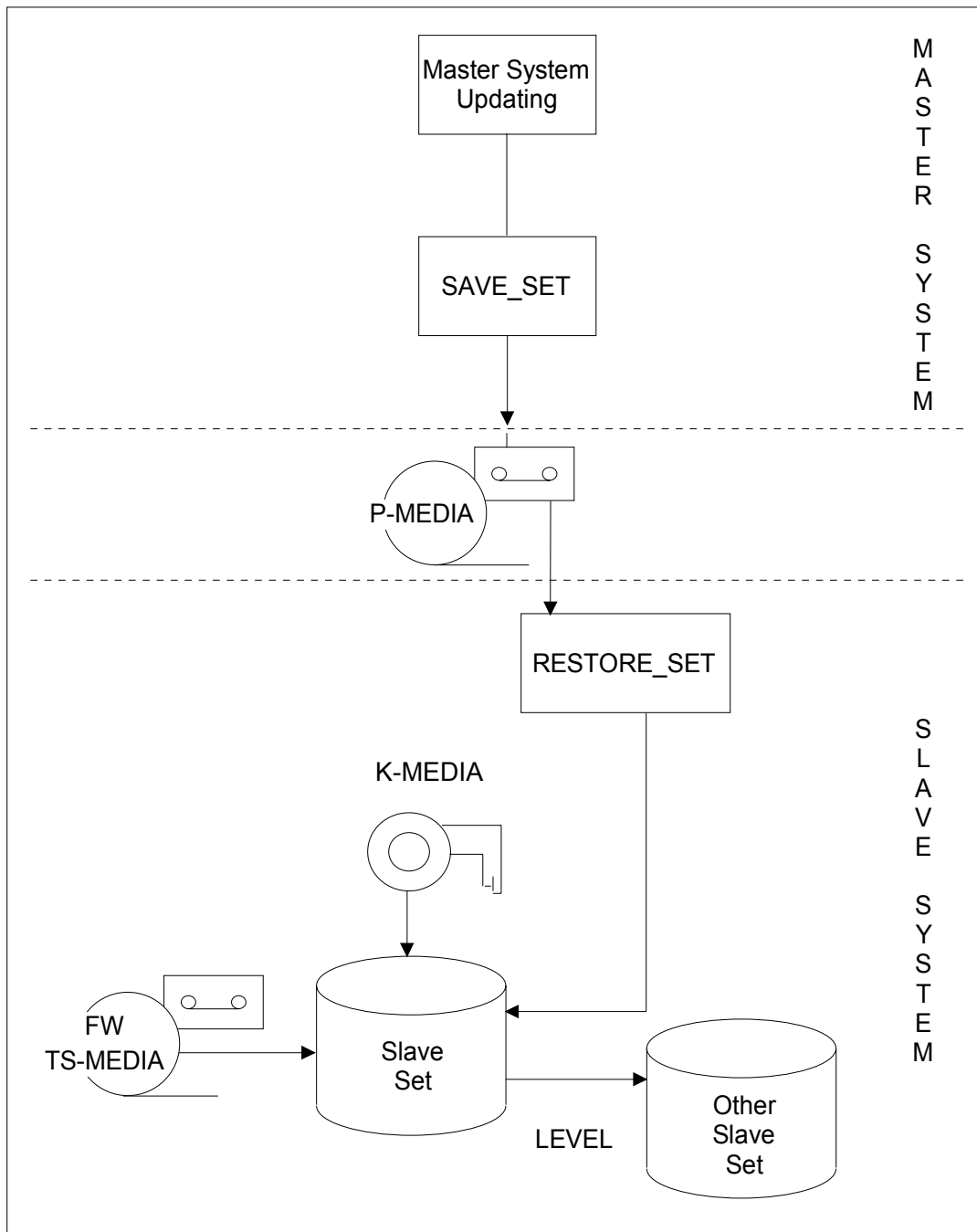


Figure 3-3. Multi-System Updating



### 3.3.2 Master System Updating Operations

#### 3.3.2.1 Technical Status Application

1. Execute the update function appropriate to the Domain to be modified.  
Refer to the chapter:  
*Updating Domains* for GCOS  
*Firmware Generation (FGF) and Test and Diagnostics (OLTD) Facilities* for FW OLTD  
*The DSA Installation and Updating Facility* for DSA  
GCOS 7 Service Facilities (GSF) for GSF.
2. Test the update after a trial period on the Master system.
3. Execute the IUF function LEVEL relevant to the Domain being considered, in order to update the other Master Set.  
LEVEL is described in the chapter *Installation and Updating Facility*.
4. For the Slave updating procedure, execute the IUF function SAVE\_SET to obtain a SET-MEDIA from a Master Set. Save all GCOS 7 Domains.  
SAVE\_SET is described in the chapter *Installation and Updating Facility*.

#### 3.3.2.2 Application of Software Add-Ons and Deletions

1. Execute the IUF function ADD\_MI or DELETE\_MI.  
ADD\_MI and DELETE\_MI are described in the chapter *Installation and Updating Facility*.
2. Test the new MI configuration for a trial period on the Master system.
3. Execute the IUF function LEVEL\_MI to update the other Master Set.  
LEVEL\_MI is described in the chapter *Installation and Updating Facility*.
4. If the Slave system has the same MI configuration as the Master system, perform Step 4 of Technical Status Application above.  
If not, the IUF functions ADD\_MI or DELETE\_MI (as appropriate) must be performed on each Slave system.

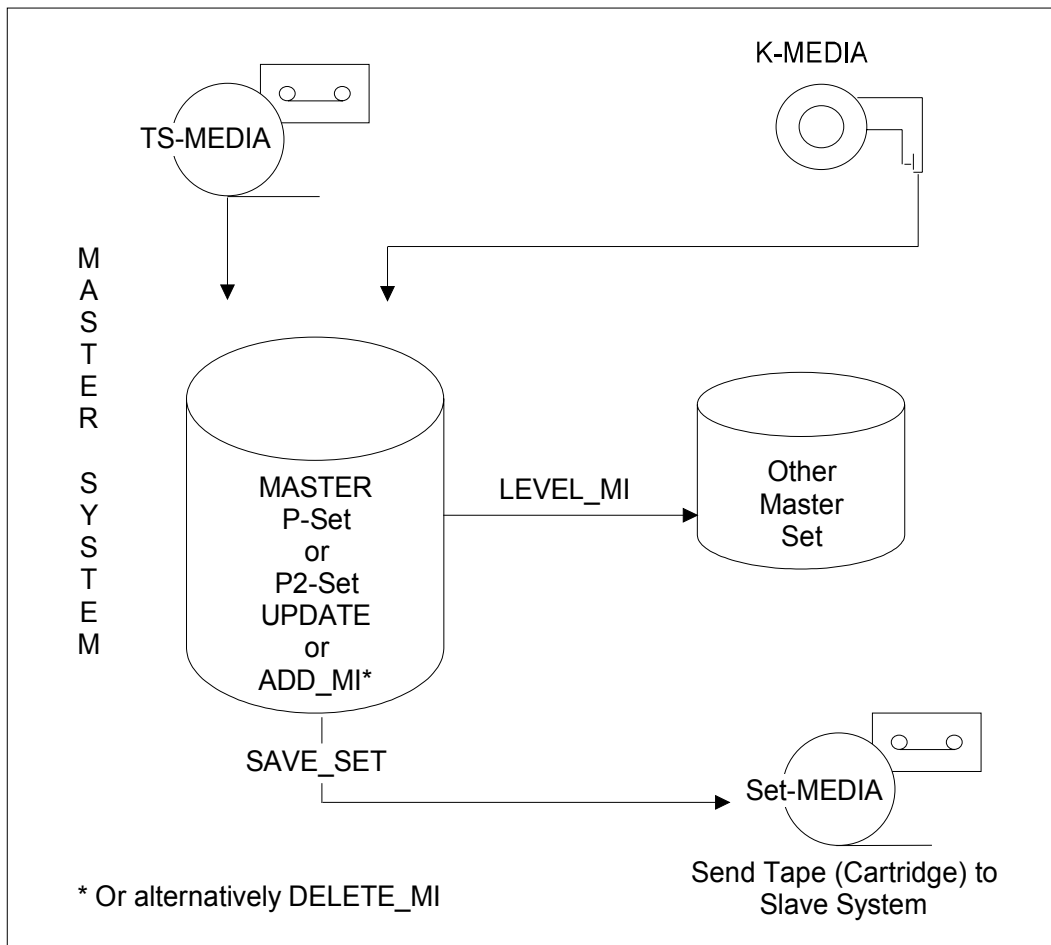


Figure 3-4. Master System Updating





### 3.3.3 Slave System Updating Operations

#### 3.3.3.1 Procedure

Either the Slave P-Set or P2-Set must be loaded and running.

**NOTE:**

There is a risk in updating a P-Set in a PO or an RP-Configuration, in a multi-programming environment, as several active libraries may be changed in the process.

1. For a Slave system that is being updated from a SET-MEDIA, execute the IUF function RESTORE\_SET relevant to the Domain updated from the SET-MEDIA to the Slave Set.

Refer to Table 3-1, for updating conditions, and to the chapter *Installation and Updating Facility* for the IUF functions in detail.

2. Terminate the system (TSYS 1 under GCOS 7 and Terminate System under the Service Processor).

3. Load GCOS 7 from the updated Set, as follows:

```
I: RESTORE (/) , RESTART (CLEAN)
```

For other ISL parameters, refer to the *GCOS 7 System Operator's Guide*.

4. The other Slave Set may be updated as appropriate, after a trial period, from the updated Set with the IUF function LEVEL.

LEVEL is described in the chapter *Installation and Updating Facility*.

#### 3.3.3.2 Selecting the Running Set or the Updated Set

It should be noted that, if the Slave's MI configuration is different from that of the Master, the IUF function RESTORE\_SET preserves the Slave MI configuration in the Slave Set(s), provided that the running Set is loaded with the Slave MI configuration.

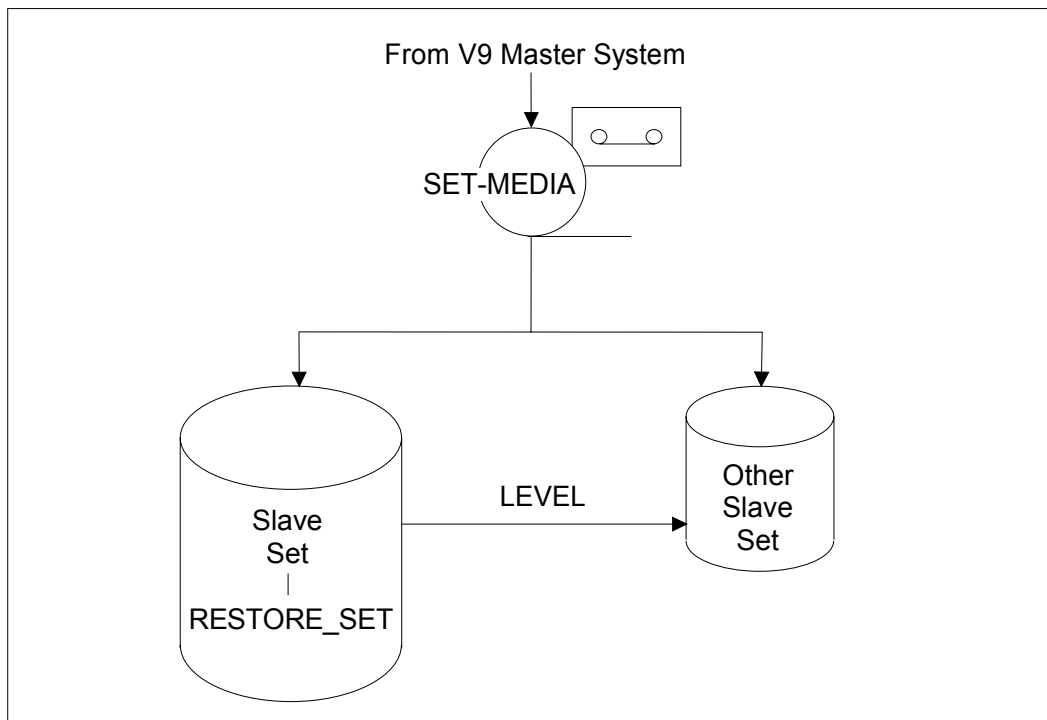
The choice of running Set and updated Set, as shown in Table 3-1, must depend on whether:

1. The Configuration concerned is PO, P2P or RP.
2. The respective MI Configuration Codes are the same or not.



**Table 3-1. Slave Set Updating Decision Table**

Slave Configuration	PO	P2P	RP
Updating Mode	Tape (Cartridge)	Tape (Cartridge)	Tape (Cartridge)
MI Conf. Code	-	-	-
Input	SET-MEDIA	SET-MEDIA	SET-MEDIA
Updated Disk	P-Set Slave	P2-Set/P-Set Slave	P-Set Slave
Running Disk	P-Set Slave on Slave System	P-Set/P2-Set Slave on Slave System	P-Set Slave on Slave System



**Figure 3-5. Slave System Update**



---

## 4. Installation and Updating Facility

The Installation and Updating Facility (IUF) provides the functions and jobs that are used to install and maintain the GCOS 7 operating system. Access to these functions and jobs is via the command GIUF.

### 4.1 The GIUF Command

You can start the GIUF processor if you are logged under either of the projects OPERATOR or SYSADMIN. These two projects are delivered in the standard SITE.CATALOG; the difference between them is that OPERATOR has MAIN as an attribute, whereas SYSADMIN does not.

In a "protected" GCOS 7 operating system, all IUF functions must be submitted under the SYSADMIN project. In an "unprotected" environment, on the other hand, a few of the IUF functions can be started from the OPERATOR project (refer to the detailed function descriptions later in this chapter).

To apply a new Technical Status in a P2P configuration, you must be logged on under the SYSADMIN project, with USER=GCOS7 or USER=SYSADMIN, otherwise the UPDATE\_GCOS function will abort when it attempts to reinstall the access rights on the updated Set.

Under the OPERATOR project, you must submit IUF functions in "EJ" mode. Under the SYSADMIN project, you can submit IUF functions in "EJ" or interactive mode.

In "EJ" mode, you have the possibility of deferred replies. In interactive mode, you must reply immediately.



#### **IMPORTANT:**

Interactive mode is recommended, on account of its enhanced operability as compared with "EJ" mode.

When the GIUF command is submitted with the parameter COMMAND= or CMD=, the MENU and HELP facilities are not available.



### Syntax in Interactive Mode

```
S: GIUF [{COMMAND='char100' |FUNCTION=name31}]
[PRTFILE=file78];
```

### Syntax in Batch Mode

```
S: EJ PROC=GIUF [VL=([COMMAND='char100' |FUNCTION=name31])
[PRTFILE=file78])] ;
```

### Parameters

COMMAND	introduces the function name and its parameters.
FUNCTION	introduces the function name if all output is to be displayed on the screen.
PRTFILE	introduces the name of the print file, and may be used with the COMMAND parameter to route printed output.

### Interactive Mode

Entering the command and its parameter:

```
S: GIUF CMD='LSMI R' ;
```

Entering only the function and requesting the LSMI menu, either:

```
S: GIUF FUNC=LSMI ;
```

or:

```
S: GIUF ;
G: LSMI ;
```

Requesting the GIUF menu to choose a GIUF function:

```
S: GIUF ;
G: ?
```



### Batch Mode

Entering the command and its parameter:

```
S: EJ PROC=GIUF VL=(CMD='LSMI R');
```

Entering only the function and requesting LSMI parameter:

```
S: EJ PROC=GIUF VL=(FUNC=LSMI);
```

Requesting the GIUF menu to choose an IUF function:

```
S: EJ PROC=GIUF;  
nn/G  
S: REP nn '?';
```

#### 4.1.1 Environment Considerations

Note the following points when using the GIUF command:

1. No catalogs must be attached.
2. You must not modify your profile. The user profile is saved at IUF initiation and restored at IUF termination; hence, any modifications made during IUF will be lost automatically.
3. You must not delete any GCL global variables while running IUF, because a set of such variables is used by IUF itself.
4. The `*$BRK` command is neither processed nor recognized during the running of IUF.
5. Any cataloged files assigned with either of the commands `MWINLIB` or `MWLIB` are saved at IUF initiation and restored at IUF termination.



## 4.1.2 IUF Functions and Jobs

### Functions

ADAPT	BUILD_SYSDUMP	PATCH_SYSTEM
ADD_MI	BUILD_TVMF	PROTECT_GCOS
BUILD_CLC	BUILDP	RESTORE_SET
BUILD_CMS	BUILDP2	RESTORE_SITE
BUILD_BKST	COPY_SITEFILES	RESTORE_SITEFILES
BUILD_HALOCK	DELETE_MI	SAVE_SET
BUILD_JAS	DETECT_SITE	SAVE_SITE
BUILD_JRNAL	DISPLAY_STATUS	SAVE_SITEFILES
BUILD_LIB	EDIT	UPDATE_GCOS
BUILD_MIRLOG	FSE	UPDATE_MI
BUILD_PVMF	LEVEL	VALIDP
BUILD_QM	LEVEL_MI	VALIDP2
BUILD_QM_BACKUP	LIST_MI	VALIDR
BUILD_SPDUMP	MODIFY_CONF	
BUILD_SPOOL	PATCH_HLMLIB	

### Jobs

INSTAL_ALL	INSTAL_PSET
INSTAL_GCOS	INSTAL_RSET



## 4.2 IUF Functions

### 4.2.1 The ADAPT Function

#### Purpose

- Linkage of a P-Set to its corresponding R-Set (in an RP configuration), or of a non-running Set to the running Set (in a P2P configuration).
- Change the R-Set's volume name, when the R-Set has been saved and then restored to another volume (only necessary when non-IUF functions, such as GCL procedures SAVE\_DISK and RESTORE\_DISK, have been used. The IUF function SAVE\_SET and the IUF job INSTAL\_RSET perform the necessary operations automatically).

#### Syntax

```
{ADAPT|ADP} {LINK_SET|LKSET}=1 [{MOD_RSET|MDR}=0]  
                {{PSET_NAME|PSET}|{P2SET_NAME|P2SET}}=media:devclass
```

or:

```
{ADAPT|ADP} {LINK_SET|LKSET}=0 {MOD_RSET|MDR}=1  
                {RSET_NAME|RSET}=media:devclass
```

#### Parameters

LINK_SET	if 1, link a new P-Set or P2-Set. Specification of LINK_SET=1 is mutually exclusive with that of MOD_RSET=1 below. There is no default value for this parameter; LINK_SET=0 must explicitly accompany MOD_RSET=1.
MOD_RSET	if 1, modify the name of the R-Set. Specification of this parameter is mutually exclusive with that of LINK_SET above.
PSET_NAME	defines a new P-Set media name and devclass, to be linked to the R-Set in an RP configuration, or to a running P2-Set in a P2P configuration.
P2SET_NAME	defines a new media name and devclass for the P2-Set, to be linked to a running P-Set in a P2P configuration.



RSET\_NAME defines a new media name and devclass for the R-Set, to be modified both in the R-Set and in the linked P-Set.

### Environment

The running Set may be the P-Set, the P2-Set or the R-Set. However, in a P2P configuration, the Set to be linked cannot be the running Set.

If several P-Sets or P2-Sets exist in a given configuration, ADAPT must be executed before any IUF functions which require both Sets (R and P, or P and P2).

ADAPT is executed implicitly by the BUILDP and BUILDP2 functions. ADAPT may additionally be used to link two Sets (MASTER and SLAVE) in a multi-system environment.

This function is not allowed in a PO configuration since it has no relevance.

### Detailed purpose

Checks the configuration type applicable.

Requests input parameters, if not given in the submitted command.

Displays the media name and devclass of the previous or current P-Set, P2-Set or R-Set, if not submitted with the command.

Updates the IUF\_CONFSET subfile in the SYS.IUF library of both Sets, as appropriate:

- RSETMD, RSETDVC
- PSETMD, PSETDVC
- P2SETMD, P2SETDVC.

### EXAMPLE:

```
G: ADAPT LINK_SET=1 PSET=P7-901:MS/FSA
    P2SET=P27901:MS/FSA
```

The P-Set, in which the SYS.CATALOG has been allocated on MS/FSA media P7-901, is to be linked to the P2-Set, in which the SYS.CATALOG file has been allocated on MS/FSA media P27901.

```
G: ADP LKSET=0 MOD_RSET=1 RSET_NAME=RSETA1:MS/FSA
```

The R-Set in the installation is to become that identified by "RSETA1:MS/FSA".

□





## 4.2.2 The ADD\_MI Function

### Purpose

Validates the addition of Marketing Identifiers to the P-Set in a PO- or RP configuration, or to the running Set in a P2P configuration, from a key delivered to the user.

### Syntax

```
{ADD_MI | ADMI}  {{KEY_FILE | KEYF}=file78  
                  | {KEY_CODE | KEYC}=char156};
```

### Parameters

**KEY\_FILE** identifies the name of the key file (i.e., of the K-MEDIA). This parameter is mutually exclusive with **KEY\_CODE**, below.

**KEY\_CODE** is the code of the key, between slashes. For example:

```
[identifier]/c25b67f2/
```

This parameter is mutually exclusive with **KEY\_FILE**, above. The optional identifier, of up to 31 alphanumeric characters in length, is defined by Bull to designate the site concerned.

### Environment

The running Set may be the P-Set, P2-Set, or R-Set.

The modified Set is the P-Set in an RP- or PO configuration, or the running Set in a P2P configuration.

The R-Set in an RP configuration, or the non-modified Set in a P2P configuration, must subsequently be updated with the **LEVEL\_MI** function.

When execution is completed, a warning message indicates the action to be taken for new MIs to be taken into account:

- an ISL RESTORE is required immediately, or
- the ISL RESTORE session can be put off until the next system restart.



If any new product requires an ISL session as part of its installation procedure, the ISL RESTORE can be done at the same time. See the installation guide for the product concerned.

### Detailed purpose

Requests the ADD\_MI parameters, if these are not supplied with the submitted command.

If the KEY\_CODE parameter is given, its contents are stored in the IUF\_KEY member of the SYS.IUF library.

Calls the IUF\_ADMI JCL to perform the following operations:

- Read and decode the key.
- Display the added MI's together with the target volume's name.
- Request agreement on the MI's displayed.
- Validate the added MI's in the SYS.SYSTEM file of the target Set.
- Display on the console and write in the JOR one of the following warning messages indicating what action is necessary for added MIs to be taken into account:
  - for MIs dynamically validated:
    - \* ADD\_MI DYNAMICALLY ACCOUNTED FOR THE CURRENT GCOS7 SESSION
    - \* ISL RESTORE WILL ONLY BE NECESSARY AT NEXT SYSTEM RESTART
  - for MIs not dynamically validated:
    - \* ADD\_MI PROCESSED: ISL RESTORE NECESSARY

### Constraints

Running GCOS 7-V9/TA, when MI's concerning "Customer Dedicated Processors" (CDP) (MI EXSD020 or TASD002) are added, a RESTORE, RESTART (CLEAN) is mandatory.

These MI's are not concerned with GCOS 7-V10/XTA



**EXAMPLES:**

```
S: GIUF CMD='ADD_MI KEY_FILE=MIKEY:MIKEY:CT';
```

The key is located in the file MIKEY.

```
S: GIUF CMD='ADD_MI KEY_FILE=SYS.IUF..KEY';
```

The key is located in the member KEY, of the running SYS.IUF library.

```
S: GIUF
```

```
G: ADMI KEY_CODE=BP31/D76N59A2/;
```

The key is supplied as a parameter entered from the keyboard.

□

**NOTE:**

The K-MEDIA is defined as:

```
MD=MIKEY   DVC=dvc   EFN=MIKEY
```



### 4.2.3 The BUILD\_BKST Function

#### Purpose

Allocation of a SYS.BKST[i] file (where  $0 \leq i \leq 15$ ), on a P-Set, a P2-Set, or an R-Set. This is a backing store file.

#### NOTES:

1. This function may be used for building the SYS.BKST file, which is considered as having the virtual suffix 0 for the purposes of the above algorithm.
2. This file may not be allocated on mirror disks.

#### Syntax

```
{BUILD_BKST|BKST} {NUMBER|NB}=n SET={P|P2|R}
                               {VOLUME|VOL}=media:devclass {SIZE|SZ}=nnnn;
```

#### Parameters

NUMBER	defines "i" in SYS.BKST[i] where $0 \leq i \leq 15$ .  <b>NOTE:</b> The first file in the Set is SYS.BKST (and <b>not</b> SYS.BKST0). The second file is SYS.BKST1, etc.
SET	identifies the Set in which the SYS.BKSTi file is to be built.
VOLUME	identifies the media and devclass names of the volume to contain the new SYS.BKSTi file.
SIZE	defines the size of the new SYS.BKSTi file. The maximum size is 65535 blocks.

#### Environment

The running Set may be the P-Set, the P2-Set, or the R-Set, as appropriate to the configuration concerned.



### Detailed purpose

The BUILD\_BKST parameters are requested interactively, if these have not already been supplied with the command.

The following operations are performed:

- Deletion of any existing SYS.BKSTi file from the Set specified
- Allocation of a new SYS.BKSTi file on the disk specified by VOLUME, and catalog the new file in the SYS.CATALOG belonging to the Set.
- Formatting the new SYS.BKST[i] file's space.

### EXAMPLE:

```
G: BBKST NUMBER=2 SET=P VOLUME=FSA3:MS/FSA SIZE=17000;
```

A SYS.BKST2 file is to be allocated on FSA3:MS/FSA and cataloged in the SYS.CATALOG on the P-Set.

□



#### 4.2.4 The BUILD\_CMS Function (if applicable)

##### Purpose

Allocation of the following files on a specified FBO volume:

```
SITE.CMS_CSD
SITE.CMS_CDD_name1
SITE.CMS_CDD_name2
SITE.CMS_DS_name1_name2
```

where name1 and name2 correspond to the names of the two coupled systems as they are known in the network. These files are used for the management of TDS-HA.

##### Syntax

```
{BUILD_CMS|BCMS} CATVOL=media:devclass
                        {VOLUME|VOL}=media:devclass
                        {NAME1|N1}=char4
                        {NAME2|N2}=char4;
```

##### Parameters

CATVOL	Volume containing the SITE.CATALOG
VOLUME	Defines the media and devclass names of the volume to contain the new SITE.CMS* files. The FBO disk is shared between two coupled systems.
NAME1	Network name of one of the coupled systems.
NAME2	Network name of the other coupled system.

##### Environment

The running Set may be the P-Set, the P2-Set or the R-Set, as appropriate to the configuration concerned.



### Detailed purpose

The BUILD\_CMS parameters are requested interactively, if they have not already been supplied.

The following operations are performed:

- Delete any existing SITE.CMS\* files from the SITE.CATALOG defined in the command.
- Allocate the new SITE.CMS\* files on the volume defined by the VOLUME parameter, and catalog them in the SITE.CATALOG specified.

### EXAMPLE:

```
G: BUILD_CMS CATVOL=FSA2:MS/FSA  
          VOLUME=FSA3:MS/FSA NAME1=C001 NAME2=C002;
```

The SITE.CMS\_CDD\_C001, SITE.CMS\_CDD\_C002, SITE.CMS\_CSD, SITE.CMS\_DS\_C002\_C001 files are to be allocated on FSA3:MS/FSA and cataloged in the SITE.CATALOG located on FSA2:MS/FSA

□



#### 4.2.5 The BUILD\_HALOCK Function (if applicable)

##### Purpose

Allocation of a new SITE.HALOCK file on a specified volume. It is used for TDS-HA recovery.

##### NOTES:

1. This file must be allocated on an FBO volume.
2. It may not be allocated on a MIRROR volume.

##### Syntax

```
{BUILD_HALOCK | BHALOCK} CATVOL=media:devclass  
                        {VOLUME | VOL}=media:devclass
```

##### Parameters

CATVOL	Volume containing the SITE.CATALOG
VOLUME	Defines the media and devclass names of the volume to contain the new SITE.HALOCK file. The disk is shared between two coupled systems.

##### Environment

The running Set may be the P-Set, the P2-Set or the R-Set, as appropriate to the configuration concerned.





### Detailed purpose

The BUILD\_HALOCK parameters are requested interactively, if they have not already been supplied. The following operations are performed:

- Delete any existing SITE.HALOCK file from the SITE.CATALOG defined in the command.
- Allocate a new SITE.HALOCK file on the volume defined by the VOLUME parameter, and catalog it in the SITE.CATALOG specified.
- Initialize the new SITE.HALOCK file.

### EXAMPLE:

```
G: BUILD_HALOCK CATVOL=FSA2:MS/FSA VOLUME=FSA3:MS/FSA;
```

The SITE.HALOCK file is to be allocated on FSA3:MS/FSA and cataloged in the SITE.CATALOG located on FSA2:MS/FSA.

□



## 4.2.6 The BUILD\_JAS Function

### Purpose

Allocation of new SYS.JASBLUE, SYS.JASGREEN files on the P-Set or P2-Set specified, used for TDS-HA or for the function called Multiple After Journals (the three JAS).

### Syntax

```
{BUILD_JAS|BJAS} SET={P|P2} {VOLUME|VOL}=media:devclass;
```

### Parameters

SET	Identifies the Set in which new SYS.JAS* files are to be built.
VOLUME	Defines the media and devclass names of the volume to contain the new SYS.JAS* files.

### Environment

The running Set may be the P-Set, the P2-Set or the R-Set, as appropriate to the configuration concerned.

### Detailed purpose

The BUILD\_JAS parameters are requested interactively, if they have not already been supplied.

The following operations are performed:

- Delete any existing SYS.JAS\* files from the Set defined in the command.
- Allocate new SYS.JAS\* files on the volume defined by the VOLUME parameter, and catalog them in the SYS.CATALOG file of the Set.

### EXAMPLE:

```
G: BUILD_JAS SET=P VOLUME=FSA3:MS/FSA;
```

The SYS.JASBLUE and SYS.JASGREEN files are to be allocated on FSA3:MS/FSA and cataloged in the SYS.CATALOG file on the P-Set.

□



## 4.2.7 The BUILD\_JRNAL Function

### Purpose

Allocation of a new SYS.JRNAL file on the P-Set, P2-Set, or R-Set specified, to permit use of the Before Journal.

### Syntax

```
{BUILD_JRNAL|BJRNAL}  
  
  SET={P | P2 | R}  
  {VOLUME | VOL}=media:devclass  
  [ {SIZE | SZ}=nnnn ]  
  [ {FILEBLOCK_SIZE | BLKSZ}={512 | 4096}];
```

### Parameters

SET	identifies the Set in which a new SYS.JRNAL file is to be built.
VOLUME	defines the media and devclass names of the volume to contain the new SYS.JRNAL file.
SIZE	defines the size of the new SYS.JRNAL file.
FILEBLOCK_SIZE	defines the block size in bytes for the SYS.JRNAL file. The allowed values are 512 and 4096.

The default value for the SIZE and FILEBLOCK\_SIZE parameters depends on the organization and the device class of the specified volume of allocation as shown in the following table:

Device class type	MS/FSA
File block size in BYTES	512
Size of SYS.JRNAL file	16368 (size in blocks)



### Environment

The running Set may be the P-Set, the P2-Set, or the R-Set, as appropriate to the configuration concerned.

### Detailed purpose

The BUILD\_JRNAL parameters are requested interactively, if these have not already been supplied.

The following operations are performed:

- Delete any existing SYS.JRNAL file from the Set defined in the command
- Allocate a new SYS.JRNAL file on the VOLUME defined by "media:devclass", and catalog it in the SYS.CATALOG file of the Set.

### EXAMPLE:

```
G: BUILD_JRNAL SET=P VOLUME=P7-901:MS/FSA;
```

A SYS.JRNAL file is to be allocated on P7-901:MS/FSA, and cataloged in the SYS.CATALOG file on the P-Set.

□



## 4.2.8 The BUILD\_LIB Function

### Purpose

Allocation of a SYS.LIB[i] file (where  $0 \leq i \leq 15$ ), on a P-Set, a P2-Set, or an R-Set. This is a backing store file.

### NOTES:

1. This function may be used for building the SYS.LIB file, which is considered as having the virtual suffix 0 for the purposes of the above algorithm.
2. This file may not be allocated on mirror disks.

### Syntax

```
{BUILD_LIB|BLIB} {NUMBER|NB}=n SET={P|P2|R}  
{VOLUME|VOL}=media:devclass {SIZE|SZ}=nnnn;
```

### Parameters

NUMBER	defines the SYS.LIB[i] number, where $0 \leq i \leq 15$ ; <b>NOTE:</b> The first file in the Set is SYS.LIB (and <b>not</b> SYS.LIB0). The second file is SYS.LIB1, etc.
SET	identifies the Set in which the SYS.LIBi file is to be built.
VOLUME	identifies the media and devclass names of the volume to contain the new SYS.LIBi file.
SIZE	defines the size of the new SYS.LIBi file. The maximum size is 65535 blocks.

### Environment

The running Set may be the P-Set, the P2-Set, or the R-Set, as appropriate to the configuration concerned.



### Detailed purpose

The BUILD\_LIB parameters are requested interactively, if these have not already been supplied with the command.

The following operations are performed:

- Deletion of any existing SYS.LIBi file from the Set specified.
- Allocation of a new SYS.LIBi file on the disk specified by VOLUME, and catalog the new file in the SYS.CATALOG belonging to the Set.
- Formatting the new SYS.LIB[i] file's space.

### EXAMPLE:

```
G: BLIB NUMBER=2 SET=P VOLUME=FSA3:MS/FSA SIZE=17000;
```

A SYS.LIB2 file is to be allocated on FSA3:MS/FSA and cataloged in the SYS.CATALOG on the P-Set.

□



#### 4.2.9 The BUILD\_MIRLOG Function (if applicable)

##### Purpose

Allocation of a new SITE.MIRLOG file on a specified volume, used for the recovery of mirror disks.

##### NOTES:

1. This file must be allocated on an FBO volume.
2. It may not be allocated on a MIRROR volume.

##### Syntax

```
{BUILD_MIRLOG|BMIRLOG} CATVOL=media:devclass  
                        {VOLUME|VOL}=media:devclass
```

##### Parameters

CATVOL	Volume containing the SITE.CATALOG
VOLUME	Defines the media and devclass names of the volume to contain the new SITE.MIRLOG file.

##### Environment

The following operations are performed:



### Detailed purpose

The BUILD\_MIRLOG parameters are requested interactively, if they have not already been supplied.

The JCL sequence IUF\_BUILD\_NONEFBO is called to perform the following operations:

- Delete any existing SITE.MIRLOG file from the SITE.CATALOG defined in the command.
- Allocate a new SITE.MIRLOG file on the volume defined by the VOLUME parameter, and catalog it in the SITE.CATALOG specified.
- Initialize the new SITE.MIRLOG file.

### EXAMPLE:

```
G: BUILD_MIRLOG  CATVOL=FSA2:MS/FSA  VOLUME=FSA3:MS/FSA;
```

The SITE.MIRLOG file is to be allocated on FSA3:MS/FSA and cataloged in the SITE.CATALOG on FSA2:MS/FSA.

□





#### 4.2.10 The BUILD\_PVMF Function

##### Purpose

Allocation of a SYS.PVMF[i] file (where  $0 \leq i \leq 15$ ), on a P-Set, a P2-Set, or an R-Set. This is a backing store file.

##### NOTES:

1. This function may be used for building the SYS.PVMF file, which is considered as having the virtual suffix 0 for the purposes of the above algorithm.
2. This file may not be allocated on mirror disks.

##### Syntax

```
{BUILD_PVMF | BPVMF} {NUMBER | NB} = n SET = {P | P2 | R}
                        {VOLUME | VOL} = media:devclass {SIZE | SZ} = nnnn;
```

##### Parameters

NUMBER	defines the SYS.PVMF[i] number, where $0 \leq i \leq 15$ .
	<b>NOTE:</b> The first file in the Set is SYS.PVMF (and <b>not</b> SYS.PVMF0). The second file is SYS.PVMF1, etc.
SET	identifies the Set in which the SYS.PVMFi file is to be built.
VOLUME	identifies the media and devclass names of the volume to contain the new SYS.PVMFi file.
SIZE	defines the size of the new SYS.PVMFi file. The maximum size is 65535 blocks. The size is actually allocated in units of 64 Kbytes.

##### Environment

The running Set may be the P-Set, the P2-Set, or the R-Set, as appropriate to the configuration concerned.



### Detailed purpose

The BUILD\_PVMF parameters are requested interactively, if these have not already been supplied with the command.

The following operations are performed:

- Deletion of any existing SYS.PVMFi file from the Set specified
- Allocation of a new SYS.PVMFi file on the disk specified by VOLUME, and catalog the new file in the SYS.CATALOG belonging to the Set.
- Formatting the file space on the new SYS.PVMF[i].

### EXAMPLE:

```
G: BPVMF NUMBER=2 SET=P VOLUME=FSA3:MS/FSA SIZE=17000;
```

A SYS.PVMF2 file is to be allocated on FSA3:MS/FSA and cataloged in the SYS.CATALOG on the P-Set.

□



#### 4.2.11 The BUILD\_QM Function

##### **Purpose**

Allocation of a new SYS.QM file on the P-Set, P2-Set, or R-Set, specified, used for sending operator messages.

##### **Syntax**

```
{BUILD_QM|BQM} SET={P|P2|R} {VOLUME|VOL}=media:devclass  
                {SIZE|SZ}=nnnn;
```

##### **Parameters**

SET	identifies the Set in which a new SYS.QM file is to be built.
VOLUME	defines the media and devclass names of the volume to contain the new SYS.QM file.
SIZE	defines the size of the new SYS.QM file.

##### **Environment**

The running Set may be the P-Set, the P2-Set, or the R-Set, as appropriate to the configuration concerned.



---

### Detailed purpose

The BUILD\_QM parameters are requested interactively, if these have not already been supplied.

The following operations are performed:

- Delete any existing SYS.QM file from the Set defined in the command
- Allocate a new SYS.QM file on the VOLUME defined by "media:devclass", and catalog it in the SYS.CATALOG file of the Set.

### EXAMPLE:

```
G: BUILD_QM SET=P VOLUME=P7-901:MS/FSA;
```

A SYS.QM file is to be allocated on P7-901:MS/FSA, and cataloged in the SYS.CATALOG file on the P-Set.

□



#### 4.2.12 The BUILD\_QM\_BACKUP Function

##### **Purpose**

Allocation of a new SYS.QM\_BACKUP file on the P-Set, P2-Set, or R-Set specified, used in case of failure of the SYS.QM file.

##### **Syntax**

{BUILD\_QM\_BACKUP | BQM\_BACKUP | BQMB}

SET={P|P2|R} {VOLUME|VOL}=media:devclass {SIZE|SZ}=nnnn;

##### **Parameters**

SET	identifies the Set in which a new SYS.QM_BACKUP file is to be built.
VOLUME	defines the media and devclass names of the volume to contain the new SYS.QM_BACKUP file.
SIZE	defines the size of the new SYS.QM_BACKUP file.

##### **Environment**

The running Set may be the P-Set, the P2-Set, or the R-Set, as appropriate to the configuration concerned.



### Detailed purpose

The BUILD\_QM\_BACKUP parameters are requested interactively, if these have not already been supplied.

The following operations are performed:

- Delete any existing SYS.QM\_BACKUP file from the Set defined in the command
- Allocate a new SYS.QM\_BACKUP file on the VOLUME defined by "media:devclass", and catalog it in the SYS.CATALOG file of the Set.

### EXAMPLE:

```
G: BUILD_QM_BACKUP SET=P VOLUME=P7-901:MS/FSA;
```

A SYS.QM\_BACKUP file is to be allocated on P7-901:MS/FSA, and cataloged in the SYS.CATALOG file on the P-Set.

□



### 4.2.13 The BUILD\_SPDUMP Function

#### Purpose

Allocation of new SYS.SPdump file on the P-Set or P2-Set specified, used for storing binary dumps of TDS and system processors.

#### Syntax

```
{BUILD_SPDUMP|BSPDUMP} SET={P|P2|R}  
                {VOLUME|VOL}=media:devclass
```

#### Parameters

SET	Identifies the Set in which a new SYS.SPdump file is to be built.
VOLUME	Defines the media and devclass name of the volume to contain the new SYS.SPdump file.

#### Environment

The running Set may be the P-Set, the P2-Set or the R-Set, as appropriate to the configuration concerned.

#### Detailed purpose

The BUILD\_SPDUMP parameters are requested interactively, if they have not already been supplied.

The following operations are performed:

- Delete any existing SYS.SPdump file from the Set defined in the command
- Allocate new SYS.SPdump file on the volume defined by "media:devclass", and catalog it in the SYS.CATALOG file of the Set.

#### EXAMPLE:

```
G: BUILD_SPDUMP SET=P VOLUME=FSA3:MS/FSA;
```

A SYS.DUMP file is to be allocated on FSA3:MS/FSA and cataloged in the SYS.CATALOG file on the P-Set.

□



#### 4.2.14 The BUILD\_SPOOL Function

##### Purpose

Allocation of a SYS.SPOOLi file (where  $0 \leq i \leq 9$ ), on a P-Set, a P2-Set, or an R-Set, as appropriate to the installation concerned, used for the management of IOF contexts.

##### Syntax

```
{BUILD_SPOOL|BSPOOL} {NUMBER|NB}=n SET={P|P2|R}
                               {VOLUME|VOL}=media:devclass
                               {SIZE|SZ}=nnnn;
```

##### Parameters

NUMBER	specifies the number "i" of the SYS.SPOOLi file where $0 \leq i \leq 9$ .
SET	identifies the Set in which a new SYS.SPOOLi file is to be built.
VOLUME	defines the media and devclass names of the resident volume on which a new SYS.SPOOLi file is to be built.
SIZE	defines the size of the new SYS.SPOOLi file.

##### Environment

The running Set may be the P-Set, the P2-Set, or the R-Set, as appropriate to the configuration concerned.

The SYS.SPOOL file must be allocated on a resident volume.





### Detailed purpose

The BUILD\_SPOOL parameters are requested interactively, if these have not already been supplied.

The following operations are performed:

- Delete any existing SYS.SPOOLi file from the Set specified
- Allocate a new SYS.SPOOLi file on the disk defined by media.devclass, and catalog it in the Set's SYS.CATALOG file.
- Modify the sharing parameter of the SYS.SPOOLi file, in the SYS.CATALOG of the Set, to SHARE = NORMAL.

### EXAMPLE:

```
G: BSPOOL NUMBER=6 SET=P2 VOLUME=FSA3:MS/FSA SIZE=30000;
```

A SYS.SPOOL6 file is to be allocated on FSA3:MS/FSA and cataloged in the SYS.CATALOG file on the P2-Set.

□



#### 4.2.15 The BUILD\_SYSDUMP Function

##### Purpose

Allocation of a new SYS.SYSDUMP file on the P-Set, or P2-Set specified, used for storing GCOS 7 dumps after a crash.

##### NOTE:

This file may not be used on mirror disks.

##### Syntax

```
{BUILD_SYSDUMP|BSYSDUMP} SET={P|P2}
                               {VOLUME|VOL}=media:devclass
                               {SIZE|SZ}=nnnn;
```

##### Parameters

SET	identifies the Set in which a new SYS.SYSDUMP file is to be built.
VOLUME	defines the media and devclass names of the volume to contain the new SYS.SYSDUMP file.
SIZE	defines the size of the new SYS.SYSDUMP file. The maximum size is 65535 blocks.

##### Environment

The running Set may be the P-Set, P2-Set, or R-Set, as appropriate to the configuration concerned.



### Detailed purpose

The SYS.SYSDUMP requires initializing. GIUF BUILD\_SYSDUMP is mandatory to rebuild this file.

The BUILD\_SYSDUMP parameters are requested interactively, if these have not already been supplied.

The following operations are performed:

- Delete any existing SYS.SYSDUMP file from the Set defined in the command
- Allocate a new SYS.SYSDUMP file on the VOLUME defined by "media:devclass", and catalog it in the SYS.CATALOG file of the Set.
- Format the file space of the new SYS.SYSDUMP.

### EXAMPLE:

```
G: BUILD_SYSDUMP SET=P SIZE=80000 VOLUME=FSA3:MS/FSA;
```

A SYS.SYSDUMP file of 80000 blocks is to be allocated on FSA3:MS/FSA, and cataloged in the SYS.CATALOG file on the P-Set.

□



## 4.2.16 The BUILD\_TVMF Function

### Purpose

Allocation of a SYS.TVMF[i] file (where  $0 \leq i \leq 15$ ), on an P-Set, a P2-Set, or an R-Set. This is a backing store file.

### NOTES:

1. This function may be used for building the SYS.TVMF file, which is considered as having the virtual suffix 0 for the purposes of the above algorithm.
2. This file may not be allocated on mirror disks.

### Syntax

```
{BUILD_TVMF | BTVMF} {NUMBER | NB} = n SET = {P | P2 | R}
                               {VOLUME | VOL} = media:devclass {SIZE | SZ} = nnnn;
```

### Parameters

NUMBER	defines the SYS.TVMF[i] number, where $0 \leq i \leq 15$ .  <b>NOTE:</b> The first file in the Set is SYS.TVMF (and <b>not</b> SYS.TVMF0), the second file is SYS.TVMF1, etc.
SET	identifies the Set in which the SYS.TVMFi file is to be built.
VOLUME	identifies the media and devclass names of the volume to contain the new SYS.TVMFi file.
SIZE	defines the size of the new SYS.TVMFi file. The maximum size is 65535 blocks. The size is actually allocated in units of 64 Kbytes.

### Environment

The running Set may be the P-Set, the P2-Set, or the R-Set, as appropriate to the configuration concerned.



### Detailed purpose

The BUILD\_TVMF parameters are requested interactively, if these have not already been supplied with the command.

The following operations are performed:

- Deletion of any existing SYS.TVMFi file from the Set specified
- Allocation of a new SYS.TVMFi file on the disk specified by VOLUME, and catalog the new file in the SYS.CATALOG belonging to the Set.
- Formatting the new SYS.TVMF[i] file's space.

### EXAMPLE:

```
G: BTVMF NUMBER=2 SET=P VOLUME=FSA3:MS/FSA SIZE=17000;
```

A SYS.TVMF2 file is to be allocated on FSA3:MS/FSA and cataloged in the SYS.CATALOG on the P-Set.

□



## 4.2.17 The BUILDP Function

### Purpose

Creation of a P-Set linked to the R-Set. BUILDP activates the TAILOR processor. This function may also be used to copy the set of DSA files from the R-Set to an already created P-Set, and to create any non-existent files with DIUF. However, it must not be used for any other purpose because there could be a danger of inconsistent results.

### Syntax

```
{BUILDP|BP} [{COMFILE|CMDP}=file78];
```

### Parameters

COMFILE identifies the file containing the TAILOR command input to enable a P-Set to be built.

### Environment

This function must be submitted under the SYSADMIN project, and is not allowed on a PO configuration.

### RP configurations:

The running Set may be either the R-Set to be linked with the new P-Set, or an already-created P-Set.

The input Set is the R-Set.

The P-Set system disk (as defined by the TAILOR command SYSVOL) must be a disk formatted in FBO.

### P2P configurations:

This is not the normal purpose of the BUILDP function and its use in a P2P configuration can lead to inconsistencies.



### Detailed purpose

When the COMFILE is supplied, the TAILOR processor is activated in batch mode with the COMFILE as input.

As an alternative to COMFILE, the member IUF\_COMF\_TAIL, which is located in the SYS.IUF library may be used, after modifications.

If COMFILE is not supplied, the TAILOR processor is activated in interactive mode, or aborted in batch mode, as appropriate.

This function checks the type of configuration concerned. It is not allowed in a PO configuration where, if it is attempted, the following message is sent:

```
"FUNCTION NOT ALLOWED IN A PO CONFIGURATION"
```

P-Set files are allocated, or copied from the R-Set (or P2-Set).

The PSETMD and PSETDVC are updated in the IUF\_CONFSET member of the SYS.IUF library in both Sets, that is the newly created P-Set plus the R-Set or P2-Set as appropriate.

The latest COMSET is copied from the R-Set (or P2-Set) to the new P-Set.

### NOTE:

The IUF\_CONFSET member of the SYS.IUF on the **old** P-Set is not updated and will therefore be inconsistent with the R-Set (or P2-Set) with which it remains linked.

### EXAMPLE:

```
G: BUILDP COMFILE=file78;
```

The command file designated by "file78" contains a series of TAILOR commands to be used during the BUILDP execution. For an example of COMFILE, refer to the BUILD P2 function.

□



## 4.2.18 The BUILDP2 Function

### Purpose

Creation of a P2-Set from the P-Set in an RP or PO configuration. The P2-Set contains all the P-Set files, although the number of disks may be different. BUILDP2 activates the TAILOR processor. This function may also be used to copy the set of DSA files from a P-Set to a P2-Set, and to create any non-existent files with DIUF. However, it must not be used for any other purpose because there could be a danger of inconsistent results.

### Syntax

```
{BUILDP2|BP2} [{COMFILE|CMDP}=file78];
```

### Parameters

COMFILE                      names the file containing the input TAILOR commands for use in building a P2-Set from a P-Set.

### Environment

This function must be submitted under the SYSADMIN project.

The running Set may be either the P-Set to be linked with the P2-Set, or an already-created P2-Set.

The input Set is the P-Set.

The P2-Set system disk (as defined by the TAILOR command SYSVOL) must be a disk formatted in FBO.

### P2P configurations:

This is not the normal purpose of the BUILDP2 function and its use in a P2P configuration can lead to inconsistencies.





### Detailed purpose

TAILOR may be activated with the last set of commands of the P-Set, and the new volume values.

P2-Set files are allocated, or copied from the P-Set.

Updating of the IUF\_CONFSET member in the SYS.IUF library in both the newly created P-Set and P2-Set:

P2SETMD, P2SETDVC, CONF = P2P.

### NOTE:

The IUF\_CONFSET member of the SYS.IUF on the **old** P-Set is not updated and will therefore be inconsistent with the R-Set (or P2-Set) with which it remains linked.

### EXAMPLES:

```
BP2 COMFILE=file78;
```

The command file designated by "file78" contains a series of TAILOR commands to be used during the BUILD P2 execution.

```
S: GIUF;  
G: BUILD P2 COMFILE=LIB.. SUBFILE:MEDIA:MS/FSA;
```

### Contents of SUBFILE: (\*)

```
TLRRESET WEAK;      (**)  
TLRGEN   FILESET=GCOS COMSET=LAST;  
SYSVOL   MEDIA=FSASYS DEVCLASS=MS/FSA;  
RSDVOL   MEDIA=FSARSD DEVCLASS=MS/FSA;  
NRDVOL   MEDIA=FSANRD DEVCLASS=MS/FSA;  
BKST1    TO=RSD SIZE=30000;  
KNODET   SIZE=5000;  
SPOOL    TO=RSD SIZE=510;  
SYO      PRES=1 TO=RSD SIZE=20000 INCRSIZE=85  
          MAXSIZE=99998;  
  
EOGEN;  
TLRGEN   FILESET=FW COMSET=LAST;  
FWGEN    TO=NRD;  
EOGEN;  
TLRGEN   FILESET=GSF COMSET=LAST;  
GSFGEN   TO=RSD;  
EOGEN;  
TLRGEN   FILESET=DSA COMSET=LAST;  
DSAFILES PRES=1 TO=RSD;  
EOGEN;  
TLREXEC  FILESET=GCOS;
```



```
TLREXEC  FILESET=FW;  
TLREXEC  FILESET=OLTD;  
TLREXEC  FILESET=DSA;  
TLREXEC  FILESET=GSF;
```

□

- (\*) The member IUF\_COMF\_TAIL, which is located in the SYS.IUF library may be used as a COMFILE after modifications.
- (\*\*) For the first launching of TAILOR, the TLRRESET keyword WEAK must **not** be used. TLRRESET STRONG is the default value.



#### 4.2.19 The COPY\_SITEFILES Function

##### Purpose

Copies the contents of a site file (SYS.URCINIT, SITE.CATALOG, SITE.HELP, SITE.STARTUP, SYS.SITE.SL, SYS.SITE.BIN, SYS.DSACONF, SYS.DSACORE) from one Set to another Set. Used primarily for copying the site files between the system Set of an old GCOS 7 Release to the Set being built for a new Release, this function may also be used in RP- and P2P configurations for copying a modified site file from one of the system's Sets to its other Set.

##### Syntax

```
{COPY_SITEFILES|CPSITE} {FILE|F}=char12 INVOL=media:devclass  
OUTVOL=media:devclass;
```

-----  
When FILE=SITE.CATALOG:

```
[ {CATBUILD|CATB}={0|1} ]  
[ {IN_ACCRIGHT|INACCR}={0|1} ]  
[ {RUN_ACCRIGHT|RUNACCR}={0|1} ]  
[ {RUN_OUTPUT|RUNOUT}={0|1} ];
```

When CATBUILD=1:

```
NBOBJ=dec5;
```

The RUN\_OUTPUT parameter is required.

-----



When FILE=SITE.STARTUP:

```
[ {PREALLOC | PALC} = {0 | 1} ]  
[ {STUP_OUTVOL | STUPOV} = media:devclass ]  
[ {IN_ACCRIGHT | INACCR} = {0 | 1} ]  
[ {RUN_ACCRIGHT | RUNACCR} = {0 | 1} ] ;
```

When PREALLOC=1:

```
[ {SIZE | SZ} = dec8 ] ;
```

-----

When FILE=SITE.HELP:

```
[ {PREALLOC | PALC} = {0 | 1} ]  
[ {HELP_OUTVOL | HELPOV} = media:devclass ]  
[ {IN_ACCRIGHT | INACCR} = {0 | 1} ]  
[ {RUN_ACCRIGHT | RUNACCR} = {0 | 1} ] ;
```

When PREALLOC=1:

```
[ {SIZE | SZ} = dec8 ] ;
```

-----

When FILE=SYS.URCINIT or SYS.SITE.SL or SYS.SITE.BIN:

```
[ {PREALLOC | PALC} = {0 | 1} ]
```

When PREALLOC=1:

```
[ {SIZE | SZ} = dec8 ] ;
```

-----

When FILE=SYS.DSACONF or SYS.DSACORE:

```
[ {PREALLOC | PALC} = {0 | 1} ]  
[ {DSA_OUTVOL | DSAOV} = media:devclass ]
```

When PREALLOC=1:

```
[ {SIZE | SZ} = dec8 ] ;
```



**Parameters**

FILE	names the file to be copied; this must be SYS.URCINIT, SITE.CATALOG, SITE.STARTUP, SITE.HELP, SYS.SITE.SL, SYS.SITE.BIN, SYS.DSACONF, or SYS.DSACORE
INVOL	identifies the media name and devclass of the volume containing: <ul style="list-style-type: none"><li>– the input SITE.CATALOG file, when FILE = {SITE.CATALOG SITE.STARTUP SITE.HELP}</li><li>– the input SYS.CATALOG file, when FILE = {SYS.URCINIT SYS.SITE.SL SYS.SITE.BIN SYS.DSACONF SYS.DSACORE}</li></ul>
OUTVOL	identifies the media name and devclass of the volume containing: <ul style="list-style-type: none"><li>– the output SITE.CATALOG file, when FILE = {SITE.CATALOG SITE.STARTUP SITE.HELP}</li><li>– the output SYS.CATALOG file, when FILE = {SYS.URCINIT SYS.SITE.SL SYS.SITE.BIN SYS.DSACONF SYS.DSACORE}</li></ul>
When FILE = SITE.CATALOG:	
CATBUILD	if 1, any existing SITE.CATALOG is deleted from the OUTVOL and a new one built. The output SITE.CATALOG must not be the running one.
IN_ACCRIGHT	= 1 when access rights are set on the input SITE.CATALOG
RUN_ACCRIGHT	= 1 when access rights are set on the running site.CATALOG
RUN_OUTPUT	= 1 if the output Set is the running Set. This parameter must be specified in agreement with the OUTVOL parameter. RUN_OUTPUT must be specified explicitly as either 1 or 0, if <b>both</b> IN_ACCRIGHT = 1 <b>and</b> RUN_ACCRIGHT = 0 are defined; as 0 if CATBUILD = 1 is defined. In all other cases, the value of 0 is applied.
NBOBJ	defines the approximate maximum number of objects to be accommodated in the SITE.CATALOG to be built.



When FILE = SITE.STARTUP:

PREALLOC	if 1, space is allocated for a SITE.STARTUP file on the volume named by STUP_OUTVOL
STUP_OUTVOL	identifies the media name and devclass of the volume to contain the new SITE.STARTUP file.
SIZE	defines the size of the new SITE.STARTUP file.
IN_ACCRIGHT	= 1 when access rights are set on the input SITE.CATALOG
RUN_ACCRIGHT	= 1 when access rights are set on the running SITE.CATALOG

When FILE = SITE.HELP:

PREALLOC	if 1, space is allocated for a SITE.HELP file on the volume named by HELP_OUTVOL
HELP_OUTVOL	identifies the media name and devclass of the volume to contain the new SITE.HELP file.
SIZE	defines the size of the new SITE.HELP file.
IN_ACCRIGHT	= 1 when access rights are set on the input SITE.CATALOG
RUN_ACCRIGHT	= 1 when access rights are set on the running SITE.CATALOG

When FILE = SYS.URCINIT or SYS.SITE.SL or SYS.SITE.BIN:

PREALLOC	if 1, space is allocated for a SYS.URCINIT or SYS.SITE.SL or SYS.SITE.BIN file on the volume named by OUTVOL
SIZE	defines the size of the new SYS.URCINIT or SYS.SITE.SL or SYS.SITE.BIN file.

When FILE = SYS.DSACONF or SYS.DSACORE:

PREALLOC	if 1, space is allocated for a SYS.DSACONF or SYS.DSACORE file on the volume named by DSA_OUTVOL
DSA_OUTVOL	identifies the media name and devclass of the volume to contain the new SYS.DSACONF or SYS.DSACORE file. This volume must be in FBO organization.



SIZE defines the size of the new SYS.DSACONF or SYS.DSACORE file.

### Environment

The running Set may be the P-Set, P2-Set, or R-Set.

In PO configurations, this function is used only for copying essential site files from the Set containing an old GCOS 7 Release to that being built for a new Release.

The input and output volumes may be both of the same type and organization or each of a different type and organization.

Files may be copied using the COPY\_SITEFILES function:

From an FBO site to another FBO site

From an VBO site to an FBO site.

However, they cannot be copied from an FBO site to a VBO site.

A **Migration Tool** is available for transferring files from an FBO site to a VBO site. File migration can take place on the same site, or from one site to another. The Migration Tool is reserved for users logged on under the SYSADMIN project, and is called by the GCL command MNMIG. For more information, refer to the *File Migration Tool User's Guide*.

The user must be logged on under the SYSADMIN project, in order to be able to copy the SITE.CATALOG.

The running SITE.CATALOG cannot be deleted and rebuilt (by specifying CATBUILD=1 when RUN\_OUTPUT=1).

When the input SITE.CATALOG has its access rights validated:

- The output SITE.CATALOG will also have its access rights validated; note that private catalogs' access rights (together with their master directories in the SITE.CATALOG) cannot be moved unless all the objects concerned have the appropriate access rights in the output SITE.CATALOG **as well**. The easiest way to move master directories together with the corresponding private catalog access rights, is to rebuild the new SITE.CATALOG (by specifying CATBUILD=1).
- If the disk supporting either the input or the output SITE.CATALOG is **not** the running disk, the running disk's access rights are temporarily validated for the duration of the job's execution.



When a new SITE.CATALOG file is built in output, the following files **must** be cataloged in the new SITE.CATALOG:

- SITE.STARTUP
- SITE.HELP
- SITE.IN
- and, when they exist, the SITE.CMS\_\*, SITE.HALOCK and SITE.MIRLOG files.

When the new site catalog is updated only (CATBUILD=0), the entries for the following files are not modified:

- SITE.STARTUP
- SITE.HELP
- SITE.IN
- and, when they exist, the SITE.CMS\_\* and SITE.HALOCK files.

The SITE.MIRLOG and other cataloged files keep the characteristics of the input SITE.CATALOG.

The SITE.STARTUP and SITE.HELP input files must both be cataloged.

#### **Fixed Disk Considerations:**

- When copying site files from a site which has only fixed disks to another site, magnetic tapes or cartridges are used as a vehicle.
- In this situation, the IUF functions SAVE\_SITEFILES and RESTORE\_SITEFILES must be used.

#### **Detailed purpose**

This function requests the COPY\_SITEFILES parameters, if these have not already been supplied with the command.

The following files may be copied:

SYS.URCINIT	SITE.CATALOG
SYS.SITE.BIN	SITE.HELP
SYS.SITE.SL	SITE.STARTUP
SYS.DSACONF	SYS.DSACORE

This function calls the JCL, as appropriate, for one of the following:

IUF\_CATCOPY, which uses the CATMOVE utility to copy the SITE.CATALOG file

IUF\_STPCOPY, which uses the LIBMAINT processor to move the SITE.STARTUP file





IUF\_HELPCOPY, which uses the LIBMAINT processor to move the SITE.HELP file

IUF\_URCCOPY, which uses the standard step H\_URFCP to copy the SYS.URCINIT file.

IUF\_DSACOPY, which uses the LIBMAINT processor to move the SYS.DSACONF and SYS.DSACORE files

IUF\_SITECOPY, which uses the LIBMAINT processor to move the SYS.SITE.SL and SYS.SITE.BIN files.

**EXAMPLES:**

```
G: CPSITE FILE=SITE.CATALOG INVOL=P1V9:MS/FSA
   OUTVOL=P2V9:MS/FSA;
```

The SITE.CATALOG file is to be copied from the Set designated by "P1V9:MS/FSA" to the Set designated by "P2V9:MS/FSA".

```
G: CPSITE FILE=SITE.STARTUP INVOL=P1V9:MS/FSA
   OUTVOL=P2V9:MS/FSA;
```

The SITE.STARTUP file is to be copied from the Set designated by "P1V9:MS/FSA" to the Set designated by "P2V9:MS/FSA".

```
G: COPY_SITEFILES FILE=SYS.URCINIT INVOL=P1V9:MS/FSA
   OUTVOL=P2V9:MS/FSA;
```

The SYS.URCINIT file is to be copied from the Set designated by "P1V9:MS/FSA" to the Set designated by "P2V9:MS/FSA".

```
G: COPY_SITEFILES FILE=SITE.HELP INVOL=P1V9:MS/FSA
   OUTVOL=P2V9:MS/FSA;
```

The SITE.HELP file is to be copied from the Set designated by "P1V9:MS/FSA" to the Set designated by "P2V9:MS/FSA".

```
G: CPSITE FILE=SYS.DSACONF INVOL=P1V9:MS/FSA
   OUTVOL=P2V9:MS/FSA;
```

The SYS.DSACONF file is to be copied from the Set designated by "P1V9:MS/FSA" to the Set designated by "P2V9:MS/FSA".

□



## 4.2.20 The DELETE\_MI Function

### Purpose

Invalidation of deleted Marketing Identifiers from the P-Set in an RP configuration or PO configuration, or from the running Set in a P2P configuration, using a key delivered to the user.

### Syntax

```
{DELETE_MI | DLMI} {{KEY_FILE | KEYF}=file78  
| {KEY_CODE | KEYC}=char156};
```

### Parameters

**KEY\_FILE** identifies the file name of the K-MEDIA. This parameter is mutually exclusive with **KEY\_CODE**, below.

**KEY\_CODE** supplies the code of the key, between slashes; for example:

```
[identifier]/c25b67f2/
```

This parameter is mutually exclusive with **KEY\_FILE**, above. The optional identifier, of up to 31 alphanumeric characters in length, is defined by Bull to designate the site concerned.

### Environment

The running Set may be the P-Set, P2-Set, or R-Set.

The modified Set is the P-Set in an RP- or PO configuration, or the running Set in a P2P configuration.

The R-Set in an RP configuration, or the non-modified Set in a P2P configuration, must subsequently be updated with the **LEVEL\_MI** function.

Deletion of the MIs is taken into account dynamically for the current GCOS 7 session.

An ISL RESTORE session is not necessary until the next system restart.



### Detailed purpose

This function requests the DELETE\_MI parameters, if these are not submitted with the command.

If the KEY\_CODE parameter has been given, its contents are stored in the SYS.IUF subfile IUF\_KEY.

The IUF\_DLMI JCL is called, to perform the following operations:

- Read and decode the key
- Display the deleted MI's, together with the target volume's name
- Request agreement on the displayed MIs
- Invalidate deleted MI's from the SYS.SYSTEM file in the target Set.
- Display on the console and write in the JOR the following warning message:

```
* DELETE_MI DYNAMICALLY ACCOUNTED FOR THE CURRENT SESSION
* ISL RESTORE WILL BE ONLY NECESSARY AT NEXT SYSTEM RESTART
```

### Constraints

Running GCOS 7-V9/TA, when MI's concerning "Customer Dedicated Processors" (CDP) (MI EXSD020 or TASD002) are deleted, a RESTORE, RESTART (CLEAN) is mandatory. These MI's are not applicable with GCOS 7-V10/XTA.

### EXAMPLE:

```
S: GIUF
G: DLMI KEY_FILE=file78;
```

MI(s) are to be invalidated, according to a key contained in a file designated by "file78".

```
S: GIUF
G: DLMI KEY_CODE=/C25B67F2/;
```

The key is supplied as a parameter entered from the keyboard.

□

### NOTE:

The K-MEDIA is defined as:

```
MD=MIKEY   DVC=dvc   EFN=MIKEY
```



## 4.2.21 The DETECT\_SITE Function

### Purpose

DETECT\_SITE must be run before applying a new Technical Status (with UPDATE\_GCOS). When a new Technical Status is applied, the software patches that have already been added (since the previous Technical Status) are not re-applied when UPDATE\_GCOS is run. DETECT\_SITE checks that the absence of these patches does not create a software regression.

The software patches concerned are those of the modules corrected in source code in the new Technical Status.

The DETECT\_SITE function requires a G-MEDIA (which contains the new Technical Status) and, optionally, a PATCH-MEDIA (which contains the equivalent patches, applicable to the new Technical Status). The PATCH-MEDIA may be added after the G-MEDIA.

For XTA system, since GCOS 7 TS 9920, the G-MEDIA and PATCH-MEDIA can now be a CD-ROM. In this case, the content of these CD-ROM must be transferred on a GCOS DISK using a script located on the CD-ROM, before using the DETECT\_SITE command (see "Using CD-ROM on XTA Systems").

### Syntax

```
{DETECT_SITE|DTST} INVOL1=media:devclass [INVOL2=media:devclass]
[ {DISPLAY|D}={1|0} ] ;
```

### Parameters

INVOL1	Identifies the media name(s) and device class of the volumes containing the Technical Status. The syntax is:  media-name1[/media-name2[/media-name3]]: <tape-dvc>  or  Gxxxx:MS (xxxx is the ts name , for example 9920)
INVOL2	Identifies the media-name(s) and device class of the volume containing the PATCH-MEDIA. The syntax is:



media-name1[/media-name2]:<tape-dvc>

or

DTyww:MS (DTyww is the name of PATCH-MEDIA)

If this parameter is not given a value, the DETECT\_SITE function produces the list of site patches, indicating:

- those which are corrected in the source code delivered in the G-MEDIA
- those which are not corrected in the source code delivered in the G-MEDIA

If this parameter is given a value, the DETECT\_SITE function is more complete and produces the list of patches, indicating:

- those which are corrected in the source code delivered in the G-MEDIA
- those which have an equivalent Patch delivered in the PATCH-MEDIA
- those which have no equivalent Patch delivered in the PATCH-MEDIA

DISPLAY

If = 1, the messages of the DETECT\_SITE function are displayed at the console. The default value is 0.

### Using CD-ROM on XTA Systems

For XTA Systems, if the G-MEDIA is a CD-ROM, the image of GCOS 7 Technical Status must be transferred on a GCOS 7 Disk (in the file SYS.Gxxxx) by using a script located on the CD-ROM.

Please read the installation instructions located on the CD-ROM in the file ReadMe\_Install\_updg\_components.txt. Details of this installation are documented in the paragraph "Using CD-ROM on XTA Systems" chapter 4.2.39.

The file SYS.Gxxxx, containing an image of the GCOS 7 Technical Status, is used as input data, by the function DETECT\_SITE. In this case INVOL1 must be given in the form Gxxxx:MS



For XTA Systems, if the PATCH-MEDIA is a CD-ROM, its contents must be transferred on a GCOS 7 Disk, in the file SYS.DTyww by using a script located on the CD-ROM.

Please read the installation instructions located on the CD-ROM in the file ReadMe\_Install\_detect\_site\_tsu\_components.txt.

Execute the following installation instructions, to transfer the contents of the PATCH-MEDIA in the file SYS.DTyww:

- Interop7 FTP servers must be started on Windows and GCOS 7
- Launch a Windows “ Command Prompt ” session
- Change your current directory for the CD-ROM directory
- Issue the command: Install\_Detect\_SITE\_TSU

The following information are requested:

- System Name: Windows name of the Diane System
- Gcos Disk: Media name of a work disk for a SYS.DTyww file (media name without device type on which 70.000 DBLKs free are necessary). The file SYS.DTyww will contain patch files.
- Gcos user: Name of an user in project SYSADMIN, used to perform FTP transfer, from CD-ROM to GCOS 7 file
- Password: GCOS 7 password of this user
- FTP Port: FTP port number used to communicate with GCOS 7, usually 9037

After this transfer on a GCOS 7 disk in the file SYS.DTyww, INVOL2 must be specified in the form: DTyww:MS.

Because the files SYS.Gxxxx and SYS.DTyww are catalogued in the SYS.CATALOG of the running GCOS 7 System, the functions DETECT\_SITE and UPDATE\_GCOS must be performed from this current running Set.

The file SYS.DTyww is deleted at the end of the DETECT\_SITE function, so to execute again the DETECT\_SITE function, it is necessary to transfer again the PATCH-MEDIA content in the SYS.DTyww file using the Install\_Detect\_SITE\_TSU script. The file SYS.Gxxxx is deleted only at the end of the UPDATE\_GCOS function.



**EXAMPLES:**

G: DTST INVOL1=G9920:MS INVOL2=DT451:MS;

**Environment**

The DETECT\_SITE function requires a G-MEDIA and, optionally, a PATCH-MEDIA.

The Set that is checked by DETECT\_SITE is the running Set in a P2P configuration, and the P-Set in an RP configuration.

The DETECT\_SITE and UPDATE\_GCOS function must be launched from the same running Set.

If a P2P configuration is to be updated, it is assumed that the level of site patches is the same on both the P-Set and the P2-Set.

**Detailed purpose**

The JCL job IUF\_DETECT\_SITE is called from the G-MEDIA, if Technical Status is delivered on a set of tapes or cartridges, and from SYS.IUF of running system if Technical Status is delivered on CD-ROM (for XTA System only).

The DETECT\_SITE function compares all site patches applied on customer sites with the software which is re-delivered with the new Technical Status, located on the G-MEDIA. These site patches may, or may not, be corrected in the new delivered Technical Status, or may have equivalent patches that are located in the PATCH-MEDIA (which may then be applied to the corresponding re-delivered software).



If any of these site patches, that are not corrected in the new Technical Status, have no applicable equivalent patches, a warning message is sent. The customer can therefore compare the software, analyze the differences, and decide for himself whether or not to apply the Technical Status.

The applicable equivalent patches will be automatically applied during the application of the Technical Status by the UPDATE\_GCOS function.

**EXAMPLES:**

```
G: DTST INVOL1=G99201/G99202:CT;
```

Before the GCOS Domain is updated using the G-MEDIA "G99201/G99202:CT", this command will detect, on the running Set, which of the customer's site patches, that are applied on re-delivered software, are corrected or not in the new Technical Status.

The customer's site patches and their status (corrected or not) are then listed.

```
G: DTST INVOL1=G99201/G99202:CT INVOL2=PATCH:CT;
```

Before the GCOS Domain is updated using the G-MEDIA "G99201/G99202:CT", this command will detect on the running Set and list, which of the customer's site patches, that are applied on re-delivered software:

- are corrected or not in the G-MEDIA
- have an equivalent site patch in the PATCH-MEDIA
- have no equivalent site patch in the PATCH-MEDIA

The equivalent site patches found in the PATCH-MEDIA "PATCH:CT" will be applied during the UPDATE\_GCOS function.

□





## 4.2.22 The DISPLAY\_RUNSET Function

### Purpose

Returns a value (R, P, or P2) to indicate which is the running set.

### Syntax

```
{DISPLAY_RUNSET|DRUNSET} [OUTAREA=char5]
```

### Parameter

**OUTAREA**                      Optionally specifies the name of a global variable that you have previously declared to receive the value (R, P, or P2) indicating the running set. If this parameter is not present, the result is sent to the console screen.

### Environment

This is a GCL procedure belonging to the domain H\_NOCTX. It can either be activated in the system startup or be called by any operator or IOF user.

### Detailed purpose

The GCL command DISPLAY\_RUNSET enables the system administrator to start the system with automated operational conditions adapted to the running set.

### EXAMPLE:

```
GB MYVAR LENGTH 5;  
  
DRUNSET OUTAREA=MYVAR;
```

After successful execution, the global variable MYVAR contains the value R, P, or P2 corresponding to the running set. If execution is unsuccessful, it contains the value ERROR.

□

**EXAMPLE OF USE IN AN IOF USER STARTUP:**

```
GB MYVAR CHAR 5;
DRUNSET OUTAREA=MYVAR;
AI TO=>RSET IF=#EQ(%MYVAR,R);
AI TO=>PSET IF=#EQ(%MYVAR,P);
AI TO=>P2SET IF=#EQ(%MYVAR,P2);
SEND '>>>ERROR <<<<<';
AI TO=>END;
$RSET:
    SEND '>>> RUNNING SET=R >>>';
AI TO=>END;
$PSET:
    SEND '>>> RUNNING SET=P >>>';
AI TO=>END;
$P2SET:
    SEND '>>> RUNNING SET=P2 >>>';
$END:
    SEND '>>> END USER STARTUP >>>';
```

□



### 4.2.23 The DISPLAY\_STATUS Function

#### Purpose

Displays the Technical Status of all the Domains installed on the target system Set, the location of all the system files, and the organization of the SET's volumes.

#### Syntax

```
{DISPLAY_STATUS|DSTAT} [SET={P|P2|R}]  
  
[ {LIST_TS|LSTS}={1|0} ]  
  
[ {LIST_SYSFILES|LSF}={1|0} ] ;
```

#### Parameters

SET	identifies which Set is to be the subject of the DISPLAY_STATUS function. The default value is the running Set.
LIST_TS	if this is equal to 1, the Technical Status of all Domains is listed together with the SET's organization.
LIST_SYSFILES	if this is equal to 1, all the system files are listed with their MEDIA and DEVCLASS.

#### Environment

The running set may be the P-Set, P2-Set, or R-Set, as appropriate to the configuration concerned.

The set whose status is to be displayed may be either the running set or the set currently linked to the running set.

#### Detailed purpose

The DISPLAY\_STATUS parameters are requested, if these have not already been supplied with the command.

One or two lists are displayed, as selected by the function's parameters.



**EXAMPLES:**

G: DSTAT SET=P;

The current Technical Status is to be displayed, for all the file Domains present on the system, based on the P-Set, together with the P-Set's organization.

G: DSTAT;

The current Technical Status is to be displayed, for all the Domains present on the system, based on the running SET, together with the organization concerned.

G: DSTAT LIST\_SYSFILES=1 SET=P2;

System file media and devclass are to be displayed for all the system file Domains on the P2-Set together with the current Technical Status.

□



## 4.2.24 The EDIT Function

### Purpose

Calls the Text Editor.

### Syntax

```
{EDIT|ED} [LIB=lib78]  
  
[ {COMFILE|CMDP}=file78];
```

### Parameters

LIB	identifies the SL library to be used as the output library.
COMFILE	names the file containing the EDIT commands to be executed. (The COMFILE parameter is mandatory in batch mode.)

### Environment

The running set may be the P-Set, P2-Set, or R-Set.

The accessible SL libraries are those cataloged in the running SYS.CATALOG and in the running SITE.CATALOG.

### Detailed purpose

EDIT allows you to create and modify the contents of SL library members. The Text Editor is described in the *Text Editor User's Guide*.

EDIT requests the Text Editor parameters, if not already supplied with the command.

If a LIB is:

- supplied, the EDIT processor assigns the specified library.
- not supplied, the EDIT processor assigns the SL library whose description is contained in the system variable #SLIB. (#SLIB may be set and modified by the MWLIB command before the GIUF processor is called.) If #SLIB has not been set, a temporary SL library (TEMP.SLLIB\$TEMPRY) is used.



If a COMFILE is:

- supplied, the EDIT processor will process the commands it contains.
- not supplied, the EDIT processor is started in interactive mode.

**EXAMPLES:**

G: EDIT;

Call the Text Editor. Use a default library.

G: EDIT LIB=P1.SL01;

Call the Text Editor. Use the cataloged library named P1.SL01.

G: EDIT LIB=MINE:P2V9:MS/FSA COMFILE=OURS..SF1\$RES;

Call the Text Editor. Use as output library the uncataloged SL library named MINE which resides on the MS/FSA volume named P2V9. Process the commands contained in subfile SF1 located in the uncataloged library named OURS which resides on a resident volume.

□



#### 4.2.25 The FSE Function

##### **Purpose**

Calls the Full Screen Editor.

##### **Syntax**

```
{FSE} [LIB=lib78]
```

##### **Parameters**

LIB identifies the SL library to be used as the output library.

##### **Environment**

The running set may be the P-Set, P2-Set, or R-Set.

This command cannot be used in batch mode.

If the used terminal does not have full screen facilities, use Text Editor (which is called by the EDIT command).

The accessible SL libraries are those cataloged in the running SYS.CATALOG and in the running SITE.CATALOG.

##### **Detailed purpose**

FSE allows you to create, modify, and delete SL library members. The Full Screen Editor is described in the *Full Screen Editor User's Guide*.

FSE requests the Full Screen Editor parameters, if not already supplied with the command.



If a LIB is:

- supplied, the FSE processor assigns the specified library.
- not supplied, the FSE processor assigns the SL library whose description is contained in the system variable #SLIB. (#SLIB may be set and modified by the MWLIB command before the GIUF processor is called.) If #SLIB has not been set, a temporary SL library (TEMP.SLLIB\$TEMPRY) is used.

**EXAMPLES:**

G: FSE;

Call the Full Screen Editor. Use a default library.

G: FSE LIB=P1.SL01;

Call the Full Screen Editor. Use the cataloged library named P1.SL01.

□





#### 4.2.26 The LEVEL Function

##### Purpose

Correctable system files in the P-Set, or in the R-Set, are updated from either the P-Set, or the P2-Set, or vice versa.

##### Syntax

```
{LEVEL|LV} INSET={P|P2|R} OUTSET={P|P2|R}
[ {DOMAIN|DOM}={ ALL |FW|DSA|OLTD|GCOS|GSF} ] ;
```

##### Parameters

INSET	identifies the input set.
OUTSET	identifies the Set to be updated.
DOMAIN	identifies the Domain whose files are to be updated: ALL (the default value) - all Domains will be updated. GCOS - the GCOS Domain is to be updated. OLTD - the OLTD Domain is to be updated. DSA - the DSA Domain will be updated. FW - the Firmware Domain is to be updated. GSF - the GCOS 7 Service Facility Domain is to be updated.

##### Environment

This function may only be submitted under the SYSADMIN project, and not in a PO configuration where, if it is attempted, the following message is sent:

```
FUNCTION NOT ALLOWED IN A PO CONFIGURATION
```

The running Set may be the P-Set, P2-Set, or R-Set. However, if only one Set has its access rights validated, that Set must be the running Set.

When MI configurations differ between the input and output SYS.SYSTEM files, the running disk **must** be the output system disk.



After performing LEVEL between Sets belonging to two **different** systems, a CONFIG is required on the output Set.

If the output Set is already, or is to become, the running Set, an ISL RESTORE session **must** be run.

Refer to the "*Important Note*" in the paragraph *GCOS 7 Updating Procedure* in Chapter 7 "*Updating Domains*".

### Detailed purpose

Requests the LEVEL parameters, if these are not supplied. Displays the consequences for the updated Set. These consequences consist of an upgrading or a downgrading of the specified Domains according to the Technical Status applied to them. This function checks the type of configuration involved and calls the IUF\_LEVEL JCL, which updates one or more Domains of a Set, as follows:

- When MI configuration codes are equal:
  - Duplication of the specified Domain files. (The Firmware Domain is copied when FW Releases are the same.)
  - The MI configuration is copied from the input to the output Sets.
- When the MI configuration codes are different:
  - Duplication of the specified Domain files. (The Firmware Domain is copied when FW Releases are the same.)
  - Restores the MI configuration from memory to the output SYS.SYSTEM.

### EXAMPLE:

```
G: LV INSET=P OUTSET=P2;
```

All system files in the P2-Set are to be updated, based on the P-Set.

□



#### 4.2.27 The LEVEL\_MI Function

##### Purpose

When the MI configuration is updated on a Set of system disks (P-Set, P2-Set, or R-Set), for example after ADD\_MI or DELETE\_MI, the non-updated Set may be updated with the new MI configuration using LEVEL\_MI.

LEVEL\_MI is mandatory after UPDATE\_MI.

##### Syntax

```
{LEVEL_MI | LEVMI} INSET={P | P2 | R} OUTSET={P | P2 | R};
```

##### Parameters

INSET identifies the input Set.  
OUTSET identifies the output Set.

##### Environment

The running Set may be the P-Set, P2-Set, or R-Set.

LEVEL\_MI is applicable only in a P2P- or an RP configuration. If attempted in a PO configuration, the following message is sent:

```
"FUNCTION NOT ALLOWED IN A PO CONFIGURATION"
```

If the output Set is already, or is to become, the running Set, an ISL RESTORE session **must** be run.

When execution is completed, a warning message is displayed on the console and written in the JOR:

```
* AN ISL RESTORE SESSION IS NECESSARY ON THE OUTPUT SET  
* IF THIS SET IS ALREADY OR IS TO BECOME THE RUNNING SET.
```

##### EXAMPLE:

```
G: LEVMI INSET=P2 OUTSET=P;
```

This example concerns a P2P configuration. The MI configuration in the P-Set is to be updated from that in P2-Set.

□



## 4.2.28 The LIST\_MI Function

### Purpose

Lists from memory, or from the SYS.SYSTEM file, the MI configuration code, the hardware model number, CPU number and the MIs. For each MI, this function lists the relevant:

- Product name
- Installation date
- Expiry date, if any.

### Syntax

```
{LIST_MI | LSMI} [SET={P|P2|R}] [ {INPUT|IN}={MEM |SYS} ]  
[ {DISPLAY|D}={1|0} ] ;
```

### Parameters

SET	designates the Set from which the list is to be displayed. The default value is the running Set.
INPUT	designates the origin of the list; i.e., either memory or SYS.SYSTEM. The parameter INPUT=MEM is valid only for a running Set.
DISPLAY	if 0 (the default value), output is directed to SYS.OUT; if 1, the output is displayed at the terminal as well.



### **Environment**

The running Set may be a P-Set, a P2-Set, or an R-Set.

The Set to be listed must be the running Set, or a Set linked to it.

#### **EXAMPLES:**

G: LSMI SET=P2;

Information on the MI's of the running P2-Set is to be listed from memory, for delivery only to SYS.OUT.

G: LSMI;

Based on the running Set, MI information is to be delivered from memory to SYS.OUT for display.

G: LSMI SET=P IN=SYS D=1;

Based on the P-Set, MI information is to be displayed from the SYS.SYSTEM file, both at the requesting terminal and via SYS.OUT.

□



## 4.2.29 The MODIFY\_CONF Function

### Purpose

Allows the user to move from a P2P- or an RP configuration, to a PO configuration, and vice versa.

### Syntax

```
{MODIFY_CONF | MDC} {CURCONF | CCONF}={PO | P2P | RP}
                               {NEW_CONF | NCONF}={PO | P2P | RP};
```

-----

When CURCONF=PO and NEWCONF=RP:

```
{RSET_NAME | RSET}=media:devclass;
```

-----

When CURCONF=PO and NEWCONF=P2P:

```
{P2SET_NAME | P2SET}=media:devclass;
```

### Parameters

CURCONF	defines the current configuration.
NEWCONF	defines the new configuration.
RSET_NAME	specifies the new R-Set's MEDIA name and DEVCLASS, for linking to the running P-Set in order to pass from a PO configuration to an RP configuration.
P2SET_NAME	specifies the new P2-Set's MEDIA name and DEVCLASS, for linking to the running P-Set in order to pass from a PO configuration to a P2P configuration.



### Environment

The running Set **must** be a P-Set (PO or P2P).

If the current configuration is PO, the (P2-, or R-) Set to be linked to the P-Set (in order to change configuration) must have already been built by:

- The IUF function BUILD P2, or the IUF function SAVE\_SET followed by INSTAL\_PSET, if a P2-Set.
- The IUF function SAVE\_SET followed by INSTAL\_RSET, if an R-Set

### Detailed purpose

The MODIFY\_CONF parameters are requested interactively, if these have not already been supplied.

This function updates the IUF\_CONFSET member of the SYS.IUF library in both Sets in the system with the following parameters, as appropriate:

- CONF
- RSETMD, RSETDVC
- PSETMD, PSETDVC
- P2SETMD, P2SETDVC.

### EXAMPLE:

```
G: MODIFY_CONF CURCONF=RP NEWCONF=PO;
```

The system concerned is to be modified from an RP configuration to a PO configuration.

```
G: MODIFY_CONF CURCONF=PO NEWCONF=RP RSET_NAME=
RSETA:MS/FSA;
```

An existing PO configuration is to become an RP configuration. The R-Set, located on MS/FSA disk RSETA, is to be linked to the P-Set.

```
G: MODIFY_CONF CURCONF=PO NEWCONF=P2P P2SET_NAME=
P2SETA:MS/FSA;
```

An existing PO configuration is to become a P2P configuration. The P2-Set, located on MS/FSA disk P2SETA, is to be linked to the P-Set.

□



### 4.2.30 The PATCH\_HLMLIB Function

#### Purpose

Patching of the SYS.HLMLIB file.

#### Syntax

```
{PATCH_HLMLIB|PATLM} [SET={P|P2|R}]  
  
[ {COMFILE|CMDF}=file78  
  [ {PRTFILE|PRTF}=file78] ] ;
```

#### Parameters

SET	identifies the Set to be patched. The default value is the running Set.
COMFILE	names the file containing the LIBMAINT LM commands to be executed.
PRTFILE	names a private print file to contain the LIBMAINT LM report. The default value is: <ul style="list-style-type: none"><li>– The console, in IOF mode</li><li>– SYS.OUT, in EJ mode.</li></ul>

#### Environment

The running Set may be the P-Set, the P2-Set, or the R-Set.

In IOF mode, this function must be submitted by a user whose profile has been set with the parameter NOVICE = 0.

If the Load Modules to be patched have been preinitialized, they must first be cleared from Backing Store (see the chapter *Patching Procedures*).





### Detailed purpose

Requests the PATCH\_HLMLIB parameters, if not already supplied with the command.

If a COMFILE is:

- Supplied, the LIBMAINT LM processor will process the commands it contains.
- Not supplied, the LIBMAINT LM processor is launched in interactive mode. Any LIBMAINT LM command may be used, except MODIFY with the TS parameter.

### EXAMPLES:

```
G: PATLM COMFILE=LIB..SF SET=P;
```

Corrections made with the LIBMAINT LM commands, contained in the library LIB subfile SF, are to be applied as patches to the SYS.HLMLIB file in the P-Set.

```
G: PATLM COMFILE=LIB..SF1:VOL1:MS/FSA SET=P2;
```

Corrections made with the LIBMAINT LM commands, contained in subfile SF1 of library LIB, are to be applied as patches to the SYS.HLMLIB in the P2-Set.

□



### 4.2.31 The PATCH\_SYSTEM Function

#### Purpose

Patching of the SYS.SYSTEM file.

#### Syntax

```
{PATCH_SYSTEM|PATSYS} [SET={P|P2|R}]  
  
[ {COMFILE|CMDF}=file78  
[ {PRTFILE|PRTF}=file78] ] ;
```

#### Parameters

SET	identifies the Set to be patched - by default the running Set.
COMFILE	names the file containing the MAINTAIN_SYSTEM commands to be executed.
PRTFILE	names a private print file to contain the MAINTAIN_SYSTEM report. The default value is: <ul style="list-style-type: none"><li>– The console, in IOF mode</li><li>– SYS.OUT, in EJ mode.</li></ul>

#### Environment

The running Set may be the P-Set, the P2-Set, or the R-Set.

In IOF mode, this function must be submitted by a user whose profile has been set with the parameter NOVICE = 0.

If the output Set is already, or is to become, the running Set, an ISL RESTORE session must be run in order to validate the patch(es) just applied (see the chapter *Patching Procedures*).



### Detailed purpose

Requests the PATCH\_SYSTEM parameters, if these have not been supplied with the command. If the COMFILE is:

- Supplied, the MAINTAIN\_SYSTEM processor will process the commands it contains.
- Not supplied, the MAINTAIN\_SYSTEM processor is launched in interactive mode; any MAINTAIN\_SYSTEM commands may be used except MODIFY with the TS parameter.

### EXAMPLES:

```
G: PATSYS SET=P COMFILE=LIB..SF;
```

MAINTAIN\_SYSTEM commands and patches contained in subfile SF of the resident library LIB, are applied to the SYS.SYSTEM file in the P-Set.

□



## 4.2.32 The PROTECT\_GCOS Function

### Purpose

This function may be used for updating the system files' access right list in the SYS.CATALOG on the running system disk.

For a list of the access rights set by PROTECT\_GCOS, refer to the table A-8 *System File Access Rights* at the end of the Appendix *System File Descriptions*. Refer also to the Appendix *Maintenance in a Protected Environment*.

### Syntax

```
{PROTECT_GCOS|PROG} {ACCESS_RIGHT|ACCR}={SET|RESET};
```

### Parameters

ACCESS\_RIGHT determines whether the access rights list is to be SET or RESET, for all objects in the running SYS.CATALOG.

### Environment

Access rights (if any) must have been initialized in the running SITE.CATALOG by the system administrator, with **at least** the following three GCL commands:

```
MDACL * PROJECT=SYSADMIN/OWNER
MDACL SITE TYPE=DIR PROJECT=SYSADMIN/OWNER
MDACL SITE.CATALOG PROJECT=*/WRITE
```

Once the access rights have been completely installed and before starting normal production, the system administrator should, for example, use MDACL on the SITE.STARTUP and SITE.HELP files with READ=\*, as follows:

```
MDACL SITE.STARTUP PROJECT=*/READ
MDACL SITE.HELP PROJECT=*/READ
```

in order for all users to be able to make use of these files.

For a detailed description of catalog object protection, refer to the *Data Security Manual* and to the *Catalog Management User's Guide*.

The projects OPERATOR and RMSIOF must be present in the SITE.CATALOG.



This function must be submitted under the SYSADMIN project, and **never** while TDS is active. This is because TDS requires **exclusive** access to the SYS.HBINLIB file, which means that PROTECT\_GCOS would be placed indefinite in the state "WAITS FOR FILE" with respect to SYS.HBINLIB.

### Detailed purpose

This function requests the PROTECT\_GCOS parameter, if this has not already been submitted with the command.

It checks that SYSADMIN is owner of the SITE.CATALOG, and then creates the IUF\_OPT\_FILE member in the SYS.IUF library for use by the JCL:

- Calls the IUF\_PROTGCOS JCL in the SYS.IUF library
- Sets (or resets, as requested) the access rights in the running SYS.CATALOG.
- Deletes the IUF\_OPT\_FILE member in the SYS.IUF library.

### EXAMPLES:

```
G: PROG ACCESS_RIGHT=SET;
```

Access rights are to be set in the SYS.CATALOG on the running system disk.

```
G: PROG ACCESS_RIGHT=RESET;
```

Access rights are deleted in the SYS.CATALOG of the running set.

□



### IMPORTANT:

To delete access rights in the SITE.CATALOG, the system administrator must execute the following GCL command:

```
MDACL *, TYPE=DIR, DELETE=SYSADMIN, CATNAME=SITE
```



### 4.2.33 The RESTORE\_SET Function

#### Purpose

Correctable system files are restored from the multivolume SET-MEDIA created by the SAVE\_SET function. The functions SAVE\_SET plus RESTORE\_SET are equivalent to the LEVEL function.

This facility is mainly used in a multi-system environment, to update a slave P-Set, or P2-Set. It may also be used instead of the LEVEL function to restore a previous TS level from the SET-MEDIA.

#### Syntax

```
{RESTORE_SET|RSTSET} INVOL=media:devclass OUTSET={P|P2|R}
                                     [ {DOMAIN|DOM}={ALL|GCOS|OLTD|FW|DSA|GSF} ]
                                     [SITE={1|0}];
```

#### Parameters

INVOL	identifies the media name(s) and devclass of the SET-MEDIA.
OUTSET	names the Set to be updated.
SITE	= 1 triggers the extraction of any site patches from the Set to be modified, followed by their reapplication after completion of the restore.
DOMAIN	identifies the Domain to be updated: <ul style="list-style-type: none"> <li>ALL (the default value) - all the Domains will be updated.</li> <li>GCOS - the GCOS Domain will be updated.</li> <li>OLTD - the OLTD Domain will be updated.</li> <li>DSA - the DSA Domain will be updated.</li> <li>FW - the FIRMWARE Domain will be updated.</li> <li>GSF - the GCOS 7 Service Facility Domain will be updated.</li> </ul>



### Environment

This function may be submitted only under the SYSADMIN project.

The Set to be updated may be multivolume, and may be the P-Set, P2-Set, or R-Set, as appropriate.

The running Set may be the P-Set, P2-Set, or R-Set.

### Detailed purpose

Requests the RESTORE\_SET parameters, if these have not already been supplied with the command.

Reads the SET.STATUS file from the SET-MEDIA; refer to the description of this file for the SAVE\_SET function later in this chapter.

Comparison is made between input and output TSNs, for the Domain(s) to be updated.

The following are displayed:

- For the original saved Set: Set type, MEDIA, DEVCLASS.
- For the updated Set: Set type, MEDIA, DEVCLASS.
- The TSNs for both Sets relevant to the Domains updated.
- The consequences for the updated Set: these represent either an upgrading or a downgrading of the specified Domains, according to the Technical Status applied to them.

The user's agreement is requested on the above displayed data, and then:

- The contents of the updated Domain files are restored with the FILREST utility. (The Firmware Domain is copied when FW Releases are the same).
- The MI configuration is restored to the SYS.SYSTEM file from Memory.

If SITE = 1, site patches are retrieved from the updated Set before the restore is performed, and reapplied to the updated Set after the restore has been completed.

The CONFIG job executes with the last user COMFILE saved in the LAST\_CONFIG\_INPUT member in the SYS.IUF library.

### EXAMPLE:

```
G: RSTSET INVOL=abcD51/abcD52/abcD53:CT OUTSET=P DOM=GCOS;
```

All the correctable GCOS 7 files are to be restored to the P-Set from the SET-MEDIA, located on the tapes abcD51, abcD52 and abcD53.

□



#### 4.2.34 The RESTORE\_SITE Function

##### Purpose

The system files SYS.SITE.SL and/or SYS.SITE.BIN is/are restored from tape/cartridge to the Set designated.

##### Syntax

```
{RESTORE_SITE|RSTS} {FILE|F}={SYS.SITE.SL|SYS.SITE.BIN|BOTH}  
  
INVOL=media:devclass OUTSET={P|P2|R};
```

##### Parameters

FILE	identifies the files to be restored. When FILE = BOTH, the files SYS.SITE.SL and SYS.SITE.BIN are both restored.
INVOL	identifies the media name(s) and devclass of the Set tape/cartridge volumes.
OUTSET	names the Set to be updated.

##### Environment

The Set to be updated may be multivolume, and may be the P-Set, P2-Set, or R-Set, as appropriate.

The running Set may be the P-Set, P2-Set, or R-Set.





**Detailed purpose**

Requests the RESTORE\_SITE parameters, if these have not already been supplied with the command.

The JCL IUFRST\_SITE is called.

According to the value of the parameter FILE, the SYS.SITE.SL file and/or the SYS.SITE.BIN file is/are restored from tape/cartridge via the FILREST utility to the Set designated.

**EXAMPLES:**

```
G: RSTS FILE=BOTH INVOL=abc:CT OUTSET=P;
```

The SYS.SITE.SL and SYS.SITE.BIN files are both to be restored from tape media name "abc" to the P-Set.

□



#### 4.2.35 The RESTORE\_SITEFILES Function

##### Purpose

Restores the contents of a site file (SITE.CATALOG, SITE.HELP, SITE.STARTUP, SYS.URCINIT, SYS.SITE.SL, SYS.SITE.BIN, SYS.DSACONF, SYS.DSACORE) from tape/cartridge to the Set designated.

##### Syntax

```
{RESTORE_SITEFILES|RSTSITE} {FILE|F}=char12
                                INVOL=media:devclass
                                OUTVOL=media:devclass;
```

-----

When FILE=SITE.CATALOG:

```
[ {CATBUILD|CATB}={0|1} ]
[ {OUT_ACCRIGHT|OUTACCR}={0|1} ]
[ {RUN_ACCRIGHT|RUNACCR}={0|1} ]
[ {RUN_OUTPUT|RUNOUT}={0|1} ];
```

When CATBUILD=1:

```
[ {NBOBJ=dec5} ];
```

The RUN\_OUTPUT parameter is required.

-----

When FILE=SITE.STARTUP:

```
[ {PREALLOC|PALC}={0|1} ]
[ {STUP_OUTVOL|STUPOV}=media:devclass ]
[ {OUT_ACCRIGHT|OUTACCR}={0|1} ]
[ {RUN_ACCRIGHT|RUNACCR}={0|1} ];
```

When PREALLOC=1:

```
[ {SIZE|SZ}=dec8 ];
```

-----



When FILE=SITE.HELP:

```
[ {PREALLOC | PALC} = {0 | 1} ]  
[ {HELP_OUTVOL | HELPOV} = media:devclass ]  
[ {OUT_ACCRIGHT | OUTACCR} = {0 | 1} ]  
[ {RUN_ACCRIGHT | RUNACCR} = {0 | 1} ] ;
```

When PREALLOC=1:

```
[ {SIZE | SZ} = dec8 ] ;
```

-----

When FILE=SYS.URCINIT or SYS.SITE.SL or SYS.SITE.BIN:

```
[ {PREALLOC | PALC} = {0 | 1} ]
```

When PREALLOC=1:

```
[ {SIZE | SZ} = dec8 ] ;
```

-----

When FILE=SYS.DSACONF or SYS.DSACORE:

```
[ {PREALLOC | PALC} = {0 | 1} ]  
[ {DSA_OUTVOL | DSAOV} = media:devclass ]
```

When PREALLOC=1:

```
[ {SIZE | SZ} = dec8 ] ;
```

-----

### Parameters

FILE	names the file to be restored. This must be SYS.URCINIT, SITE.CATALOG, SITE.STARTUP, SITE.HELP, SYS.SITE.SL, SYS.SITE.BIN, SYS.DSACONF, or SYS.DSACORE
INVOL	identifies the media name(s) and devclass of the Set tape/cartridge volumes.



OUTVOL	identifies the media name and devclass of the output volume containing: <ul style="list-style-type: none"> <li>– SITE.CATALOG, when FILE = {SITE.CATALOG SITE.STARTUP SITE.HELP}</li> <li>– SYS.CATALOG, when FILE = {SYS.URCINIT SYS.SITE.SL SYS.SITE.BIN SYS.DSACONF SYS.DSACORE}</li> </ul>
When FILE = SITE.CATALOG:	
CATBUILD	if 1, any existing SITE.CATALOG is deleted from the OUTVOL and a new one built. The output SITE.CATALOG must not be the running one.
OUT_ACCRIGHT	= 1 when access rights are set on the output SITE.CATALOG
RUN_ACCRIGHT	= 1 when access rights are set on the running SITE.CATALOG
RUN_OUTPUT	= 1 if the output Set is the running Set. This parameter must be specified in agreement with the OUTVOL parameter. RUN_OUTPUT must be specified explicitly as either 1 or 0, if <b>both</b> OUT_ACCRIGHT = 1 <b>and</b> RUN_ACCRIGHT = 0 are defined; as 0 if CATBUILD = 1 is defined. In all other cases, the value of 0 is applied.
NBOBJ	defines the approximate maximum number of objects to be accommodated in the SITE.CATALOG to be built.
When FILE = SITE.STARTUP:	
PREALLOC	if 1, space is allocated for a SITE.STARTUP file on the volume named by STUP_OUTVOL.
STUP_OUTVOL	identifies the media name and devclass of the volume to contain the new SITE.STARTUP file.
SIZE	defines the size of the new SITE.STARTUP file.
OUT_ACCRIGHT	= 1 when access rights are set on the output SITE.CATALOG
RUN_ACCRIGHT	= 1 when access rights are set on the running SITE.CATALOG.



When FILE = SITE.HELP:

PREALLOC	if 1, space is allocated for a SITE.HELP file on the volume named by HELP_OUTVOL
HELP_OUTVOL	identifies the media name and devclass of the volume to contain the new SITE.HELP file.
SIZE	defines the size of the new SITE.HELP file.
OUT_ACCRIGHT	= 1 when access rights are set on the output SITE.CATALOG
RUN_ACCRIGHT	= 1 when access rights are set on the running SITE.CATALOG

When FILE = SYS.URCINIT or SYS.SITE.SL or SYS.SITE.BIN:

PREALLOC	if 1, space is allocated for a SYS.URCINIT or SYS.SITE.SL or SYS.SITE.BIN file on the volume named by OUTVOL
SIZE	defines the size of the new SYS.URCINIT or SYS.SITE.SL or SYS.SITE.BIN file.

When FILE = SYS.DSACONF or SYS.DSACORE:

PREALLOC	if 1, space is allocated for a SYS.DSACONF or SYS.DSACORE file on the volume named by DSA_OUTVOL
DSA_OUTVOL	identifies the media name and devclass of the volume to contain the new SYS.DSACONF or SYS.DSACORE file. This volume must be in FBO organization.
SIZE	defines the size of the new SYS.DSACONF or SYS.DSACORE file.



## Environment

The running Set may be the P-Set, P2-Set, or R-Set.

The input volume on which the file is saved (see SAVE\_SITEFILES) and the output volume to which the file is restored may be both of the same type and organization or each of a different type and organization.

Files may be restored using the RESTORE\_SITEFILES function:

From an FBO site to another FBO site

From an VBO site to an FBO site, except for the SYS.URCINIT file for which user options must be generated again by a \$URINIT JCL step.

However, they cannot be copied from an FBO site to a VBO site.

A **Migration Tool** is available for transferring files from an FBO site to a VBO site. File migration can take place on the same site, or from one site to another. The Migration Tool is reserved for users logged on under the SYSADMIN project, and is called by the GCL command MNMIG. For more information, refer to the *File Migration Tool User's Guide*.

If SYS.URCINIT is restored with preallocation, an ISL RESTART (WARM) session must be run.

The user must be logged on under the SYSADMIN project, in order to be able to restore the SITE.CATALOG.

The running SITE.CATALOG cannot be deleted and rebuilt (by specifying CATBUILD = 1 when RUN\_OUTPUT = 1).

When the input SITE.CATALOG has its access rights validated:

- The output SITE.CATALOG will also have its access rights validated; note that private catalogs' access rights (together with their master directories in the SITE.CATALOG) cannot be moved unless all the objects concerned have the appropriate access rights in the output SITE.CATALOG **as well**. The easiest way to move master directories together with the corresponding private catalog access rights, is to rebuild the new SITE.CATALOG (by specifying CATBUILD = 1).
- If the disk supporting the output SITE.CATALOG is **not** the running disk, the running disk's access rights are temporarily validated for the duration of the job's execution.



When a new SITE.CATALOG file is built in output, the following files **must** be cataloged in the new SITE.CATALOG:

- SITE.STARTUP
- SITE.HELP
- SITE.IN
- and, when they exist, the SITE.CMS\_\*, SITE.HALOCK and SITE.MIRLOG files.

The SITE.STARTUP and SITE.HELP input files must both be cataloged.

### Detailed purpose

This function requests the RESTORE\_SITEFILES parameters, if these have not already been supplied with the command.

The following files may be copied:

SYS.URCINIT	SITE.CATALOG
SYS.SITE.BIN	SITE.HELP
SYS.SITE.SL	SITE.STARTUP
SYS.DSACONF	SYS.DSACORE

The RESTORE\_SITEFILES function calls the JCL, as appropriate, for one of the following:

IUF\_CATREST, which uses:

- the FILREST utility to restore the SITE.CATALOG file when the INVOL and OUTVOL organizations are the same
- the CREATE processor to restore the SITE.CATALOG file when INVOL is VBO and OUTVOL is FBO.

IUF\_STPRST, which uses the LIBMAINT processor to move the SITE.STARTUP file

IUF\_HELPRST, which uses the LIBMAINT processor to move the SITE.HELP file

IUF\_URCRST, which uses the FILREST processor to restore the SYS.URCINIT file

IUF\_DSARST, which uses the LIBMAINT processor to move the SYS.DSACONF and SYS.DSACORE files

IUF\_SITERST, which uses the LIBMAINT processor to move the SYS.SITE.SL and SYS.SITE.BIN files.



**EXAMPLES:**

```
G: RSTSITE FILE=SITE.CATALOG INVOL=SAV1V9:CT
   OUTVOL=P2V9:MS/FSA;
```

The SITE.CATALOG file is to be restored from the tape designated by "SAV1V9:CT" to the Set designated by "P2V9:MS/FSA".

```
G: RSTSITE FILE=SITE.STARTUP INVOL=SAV1V9:CT
   OUTVOL=P2V9:MS/FSA;
```

The SITE.STARTUP file is to be restored from the tape designated by "SAV1V9:CT" to the Set designated by "P2V9:MS/FSA".

```
G: RST_SITEFILES FILE=SYS.URCINIT INVOL=SAV1V9:CT
   OUTVOL=P2V9:MS/FSA;
```

The SYS.URCINIT file is to be restored from the tape designated by "SAV1V9:CT" to the Set designated by "P2V9:MS/FSA".

```
G: RSTSITE FILE=SYS.DSACONF INVOL=SAV1V9:CT
   OUTVOL=P2V9:MS/FSA;
```

The SYS.DSACONF file is to be restored from the tape designated by "SAV1V9:CT" to the Set designated by "P2V9:MS/FSA".

□





## 4.2.36 The SAVE\_SET Function

### Purpose

Correctable system files relevant to all Domains (GCOS, FW, OLTD, GSF and DSA), or to a specified Domain, are saved from a P-Set, a P2-Set, or an R-Set, to a multivolume tape/cartridge called SET-MEDIA.

If all the GCOS7 Domains are saved, the `INSTAL_RSET` and `INSTAL_PSET` jobs are created on the SET-MEDIA.

When all the Domains are saved, the SET-MEDIA may be used for installing a Slave R-Set with the `INSTAL_RSET` job, a Slave P-Set with the `INSTAL_PSET` job, or for updating a Slave Set with the IUF function `RESTORE_SET`. The functions `SAVE_SET` plus `RESTORE_SET` are equivalent to the `LEVEL` function.

### Syntax

```
{SAVE_SET|SVSET} INSET={P|P2|R} {OUTDVCLASS|OUTDVC}=devclass  
{PREFIX|TPPFIX}=char5 [ {DOMAIN|DOM}={ALL|GCOS|OLTD|FW|DSA|GSF} ] ;
```

### Parameters

<code>INSET</code>	identifies the Set to be saved to the SET-MEDIA.
<code>PREFIX</code>	defines the prefix to be used for generating SET-MEDIA names, as follows:  abcdex, where:  abcde = SET-MEDIA prefix (in alphanumeric characters) x = SET-MEDIA sequence number x is automatically generated by the <code>SAVE_SET</code> .  For example:  ABCDE1, ABCDE2, ... or 123451, 123452, ...
<code>OUTDVCLASS</code>	identifies the device class of the SET-MEDIA.



DOMAIN identifies the Domain to be saved.

- ALL (the default value)- all the Domains will be saved.
- GCOS - the GCOS Domain will be saved.
- FW - the FW Domain will be saved.
- OLTD - the OLTD Domain will be saved.
- GSF - the GSF Domain will be saved.
- DSA - the DSA Domain will be saved.

### Environment

The Set to be saved may be multivolume, and may be the running Set. The SET-MEDIA must have been prepared in advance.

### Detailed purpose

This function requests the SAVE\_SET parameters, if these have not already been supplied with the command. It creates the SYS.IUF file which contains information about the saved Set, and also calls the JCL contained in the IUF\_SET\_STATUS subfile.

The SET.STATUS sequential file is created on the SET-MEDIA.

Correctable files are saved from all Domains or from a specified Domain, mostly via the FILSAVE utility (SYS.GSF.UFAS is in fact saved via CREATE, and SYS.SYSTEM is saved via MAINTAIN\_SYSTEM), including:

- GCOS 7 files.
- OLTD files.
- FW files.
- GSF files.
- DSA files.



Description of the SET.STATUS file:

The file content is a concatenation of the subfiles:

```
SYS.IUF..IUF_SET_STATUS  
SYS.IUF..IUF_MICONF
```

**EXAMPLES:**

```
G: SVSET INSET=P PREFIX=abcde OUTDVCLASS=CT;
```

The contents of the P-Set are to be saved to a SET-MEDIA multi-volume tape file. SET-MEDIA is to be located on magnetic tapes whose names are each to be prefixed with "abcde".

□



## 4.2.37 The SAVE\_SITE Function

### Purpose

The system files SYS.SITE.SL and/or SYS.SITE.BIN is/are saved to tape/cartridge from the Set designated.

### Syntax

```
{SAVE_SITE|SVS} {FILE|F}={SYS.SITE.SL|SYS.SITE.BIN|BOTH}
      INSET={P|P2|R}
      {OUTDVCLASS|OUTDVC}=<tape-dvc>
      PREFIX=char3;
```

### Parameters

FILE	identifies the files to be saved. When FILE = BOTH, the files SYS.SITE.SL and SYS.SITE.BIN are both saved.
INSET	names the Set to be saved.
OUTDVCLASS	specifies the devclass of the Set tape/cartridge volumes. See the Preface for the full syntax of device specification.
PREFIX	defines the 3 character prefix to be used in the MEDIA name of the output tape/cartridge, as follows:  abc  where abc represents the tape/cartridge prefix defined with this parameter.

### Environment

The Set to be saved may be multivolume, but **must** be the running Set.

The tape/cartridge must have been prepared in advance, using the GCL procedure PREPARE\_TAPE or the VOLPREP utility.



**Detailed purpose**

Requests the SAVE\_SITE parameters, if these have not already been supplied with the command.

The JCL IUFSAVE\_SITE is called.

According to the value of the parameter FILE, the SYS.SITE.SL file and/or the SYS.SITE.BIN file is/are saved to tape/cartridge via the FILSAVE utility from the Set designated.

**EXAMPLE:**

```
G: SVS FILE=BOTH INSET=P OUTDVC=CT PREFIX=abc
```

The SYS.SITE.SL and SYS.SITE.BIN files are both to be saved to tape/cartridge media "abc" from the P-Set.

□



## 4.2.38 The SAVE\_SITEFILES Function

### Purpose

Saves the contents of a site file (SITE.CATALOG, SITE.HELP, SITE.STARTUP, SYS.URCINIT, SYS.SITE.SL, SYS.SITE.BIN, SYS.DSACONF, SYS.DSACORE) from the Set designated to tape or cartridge.

### Syntax

```
{SAVE_SITEFILES|SVSITE} {FILE|F}=char12 INVOL=media:devclass
                                OUTVOL=media:devclass;
```

When FILE = SYS.URCINIT, SYS.SITE.SL, SYS.SITE.BIN, SYS.DSACONF or SYS.DSACORE:

the above parameters are the only ones applicable.

When FILE = SITE.CATALOG, SITE.HELP or SITE.STARTUP:

```
[{IN_ACCRIGHT|INACCR}={0|1}]
[ {RUN_ACCRIGHT|RUNACCR}={0|1}];
```

### Parameters

FILE	names the file to be saved. This must be SYS.URCINIT, SITE.CATALOG, SITE.STARTUP, SITE.HELP, SYS.SITE.SL, SYS.SITE.BIN, SYS.DSACONF, or SYS.DSACORE.
INVOL	identifies the media name and devclass of the input volume containing: <ul style="list-style-type: none"> <li>– SITE.CATALOG, when FILE = {SITE.CATALOG SITE.STARTUP SITE.HELP}</li> <li>– SYS.CATALOG, when FILE = {SYS.URCINIT SYS.SITE.SL SYS.SITE.BIN SYS.DSACONF SYS.DSACORE}</li> </ul>



OUTVOL identifies the media name and devclass of the Set tape/cartridge volumes.

### Environment

The running Set may be the P-Set, P2-Set, or R-Set.

The tape/cartridge must have been prepared in advance, using the GCL procedure PREPARE\_TAPE or the VOLPREP utility.

The user must be logged on under the SYSADMIN project, in order to be able to save the SITE.CATALOG.

When the input SITE.CATALOG has its access rights validated:

- The output SITE.CATALOG will also have its access rights validated.
- If the disk supporting the input SITE.CATALOG is **not** the running disk, the running disk's access rights are temporarily validated for the duration of the job's execution.

The SITE.STARTUP and SITE.HELP input files must both be cataloged.

### Detailed purpose

This function requests the SAVE\_SITEFILES parameters, if these have not already been supplied with the command.

The following files may be saved:

SYS.URCINIT	SITE.CATALOG
SYS.SITE.BIN	SITE.HELP
SYS.SITE.SL	SITE.STARTUP
SYS.DSACONF	SYS.DSACORE



The SAVE\_SITEFILES function calls the JCL, as appropriate, for one of the following:

IUF\_CATSAVE, which uses the FILSAVE utility followed by the CREATE processor to save the SITE.CATALOG file.

IUF\_STPSAVE, which uses the LIBMAINT processor to move the SITE.STARTUP file.

IUF\_HELPSAVE, which uses the LIBMAINT processor to move the SITE.HELP file.

IUF\_URCSAVE, which uses the FILSAVE processor to save the SYS.URCINIT file.

IUF\_DSASAVE, which uses the LIBMAINT processor to move the SYS.DSACONF and SYS.DSACORE files.

IUF\_SITESAVE, which uses the LIBMAINT processor to move the SYS.SITE.SL and SYS.SITE.BIN files.

**EXAMPLES:**

```
G: SVSITE FILE=SITE.CATALOG INVOL=P2V9:MS/FSA
   OUTVOL=SAV1V9:CT;
```

The SITE.CATALOG file is to be saved from the Set designated by "P2V9:MS/FSA" to the tape designated by "SAV1V9:CT".

```
G: SVSITE FILE=SITE.STARTUP INVOL=P2V9:MS/FSA
   OUTVOL=SAV1V9:CT;
```

The SITE.STARTUP file is to be saved from the Set designated by "P2V9:MS/FSA" to the tape designated by "SAV1V9:CT".

```
G: SAVE_SITEFILES FILE=SYS.URCINIT INVOL=P2V9:MS/FSA
   OUTVOL=SAV1V9:CT;
```

The SYS.URCINIT file is to be saved from the Set designated by "P2V9:MS/FSA" to the tape designated by "SAV1V9:CT".

```
G: SVSITE FILE=SYS.DSACONF INVOL=P2V9:MS/FSA
   OUTVOL=SAV1V9:CT;
```

The SYS.DSACONF file is to be saved from the Set designated by "P2V9:MS/FSA" to the tape designated by "SAV1V9:CT".

□





### 4.2.39 The UPDATE\_GCOS Function

#### Purpose

Updates the GCOS Domain of a P-Set, in RP- and PO configurations by applying a new Technical Status from a G-MEDIA.

Also updates either the running Set or the non-running Set in a P2P configuration.

If a P2P configuration is to be updated, it is assumed that the level of site patches is the same on both the P-Set and the P2-Set.

For XTA system, starting from GCOS 7 TS 9920, the G-MEDIA can now be a CD-ROM. In this case, the content of this CD-ROM must be first transferred on a GCOS DISK using a script located on the CD-ROM, before using the UPDATE\_GCOS function.(see "Using CD-ROM on XTA Systems" below).

#### Syntax

```
{UPDATE_GCOS|UPDG} INVOL=media:devclass [OUTSET=set];
```

#### Parameters

INVOL identifies the media names and devclass of the volumes containing the Technical Status, which is recorded on a G-MEDIA.

The syntax is:

```
media-name1[/media-name2[/media-name3]]:  
<tape-dvc>
```

or

```
Gxxxx:MS (xxxx is the ts name , for example 9920)
```

OUTSET identifies the Set to be updated. In a P2P configuration, this can be either the running Set or the non-running Set. The default value is the non-running Set.

#### Using CD-ROM on XTA Systems

For XTA Systems, if the G-MEDIA is a CD-ROM, its contents must be first transferred on a GCOS 7 Disk (in the file SYS.Gxxxx) by using a script located on the CD-ROM.



Please read the installation instructions located on the CD-ROM in the file ReadMe\_Install\_updg\_components.txt.

Execute the following installation instructions, to transfer the contents of the G-MEDIA in the file SYS.Gxxxx:

- Interop7 FTP servers must be started on Windows and GCOS 7
- Launch a Windows “ Command Prompt ” session
- Change your current directory for the CD-ROM directory
- Issue the command: Install\_GCOS\_TECHNICAL\_STATUS

The following information are requested:

- System Name: Windows name of the Diane System
- Gcos Disk: Media name of a work disk for the SYS.Gxxxx file (media name without device type on which 700.000 DBLKs free are necessary). The file SYS. Gxxxx will contain an image of the TECHNICAL STATUS.
- Gcos user: Name of an user in project SYSADMIN, used to perform FTP transfer, from CD-ROM to GCOS 7 file
- Password: GCOS 7 password of this user
- FTP Port: FTP port number used to communicate with GCOS 7, usually 9037

In case of first installation of a Technical Status located on a CD-ROM, GIUF tools have to be updated, in order to support installation using the SYS.Gxxxx file. You have to launch a JCL with command: EJ BFGIUF\_MS LIB=SYS.HSLLIB.

The file SYS.Gxxxx, containing an image of the GCOS 7 Technical Status, is used as input data, by the function UPDATE\_GCOS. In this case INVOL must be given in the form: Gxxxx:MS

Because the file SYS.Gxxxx is catalogued in the SYS.CATALOG of the running GCOS 7 System, the functions UPDATE\_GCOS must be performed from this current running Set.

The file SYS.Gxxxx is deleted at the end of the UPDATE\_GCOS function, so to execute again this function, it is necessary to transfer again the G-MEDIA content's in the SYS.Gxxxx file using the Install\_GCOS\_TECHNICAL\_STATUS script.

**EXAMPLE:**

```
G: UPDG INVOL=G9920:MS;
```



### Environment

Both Sets must be of the same MI configuration, and mounted. The updated Set is:

- The P-Set in RP- and PO configurations
- the chosen Set (non-running or running) in a P2P configuration.

If the updated Set has its access rights validated, in a P2P configuration, the running Set must also have its access rights validated.

The UPDATE\_GCOS function must be run after the DETECT\_SITE function, and on the same running disk.

This function must be submitted under the SYSADMIN project.

Refer to the "*Important Note*" in the paragraph *GCOS 7 Updating Procedure* in Chapter 7 "*Updating Domains*".

### Detailed purpose

The UPDATE\_GCOS parameters are requested, if these have not already been submitted with the command.

A display is obtained of the:

- Current GCOS 7 Technical Status of the updated Set.
- The number of the Technical Status to be applied.
- Technical Status of the Set that has not been updated.

The JCL IUF\_UPDG is called, and the LEVEL function is executed so that the two Sets are both brought to the latest Technical Status; this is done in an RP configuration from P-Set to R-Set. The following files are copied from the G-MEDIA onto the Set to be updated:

- SYS.IUF
- SYS.HSLLIB
- SYS.HLMLIB
- SYS.HCULIB
- SYS.GPL.MACLIB
- SYS.HBINLIB
- SYS.C.INCLUDE
- SYS.HELP
- SYS.SYSTEM

**IMPORTANT:**

If the user has placed any private subfiles in these libraries, they will be lost during the updating process.

The SITE patches are extracted from the running Set, and reapplied in the new SYS.SYSTEM file as well as in the new SYS.HLMLIB library after the update.

If the DETECT\_SITE function has previously been run, the new patches that are the equivalent of the old site patches will be applied in the new SYS.SYSTEM file as well as in the new SYS.HLMLIB library after the update.

After execution of the UPDATE\_GCOS function, the members COMAP and COMNAP located in the SYS.IUF library contain, respectively, the list of the site patches re-applied and the list of those that are not re-applied. These members are printed in the output report.

The MIs are validated in the new SYS.SYSTEM file, from:

- The non-updated SYS.SYSTEM file in RP and P2P configurations
- Main Memory, in PO configurations.

**NOTE:**

In the case of the first update with GCOS 7-V9 or GCOS 7-V10, you must execute VALIDP with K-media supplied.

The CONFIG job is executed, using the latest user COMFILE (containing the user's explicit CONFIG commands) saved to the LAST\_CONFIG\_INPUT member in the SYS.IUF library.

An execution summary lists all the modules delivered, as well as all the reference and interim corrections (as appropriate) applied during the current UPDATE\_GCOS execution.

**Set Evolution in a P2P configuration**

The P-Set alternates with the P2-Set as the running Set. The UPDATE\_GCOS function is performed on the non-running Set just before it becomes the running Set. This allows the new Technical Status to be tried out on the running Set:

- While conserving the non-running Set in non-updated form for backup purposes in the event of unforeseen problems with the new Technical Status.
- So that the new Technical Status may be thoroughly tried out in production before performing the LEVEL function to copy it to the non-running Set.

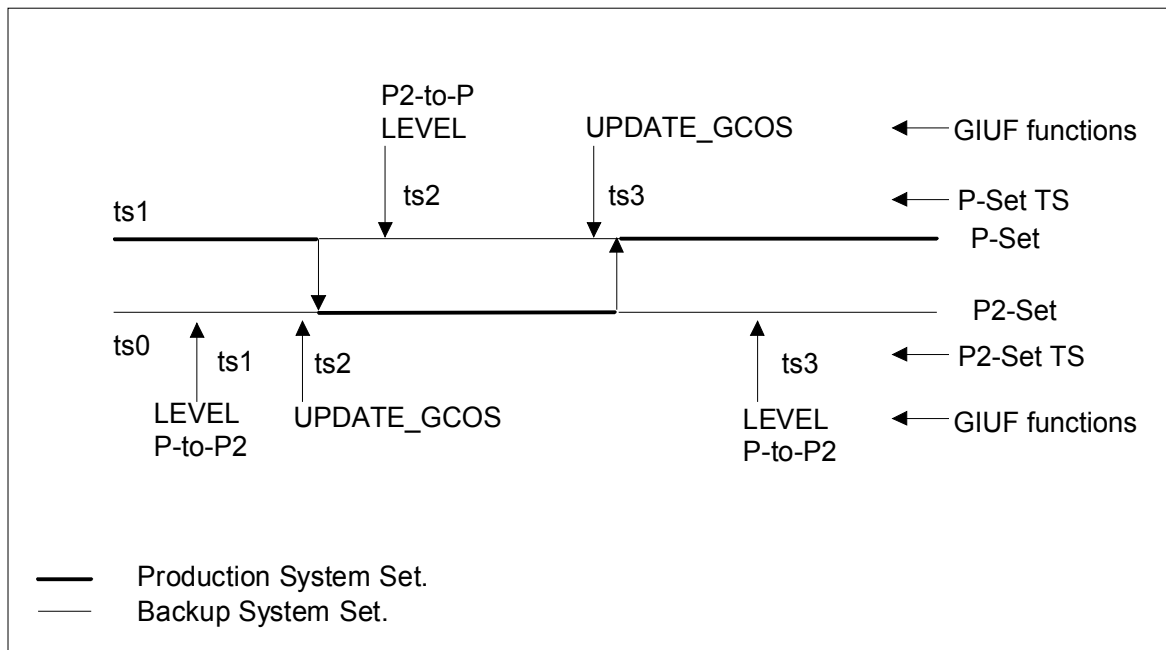


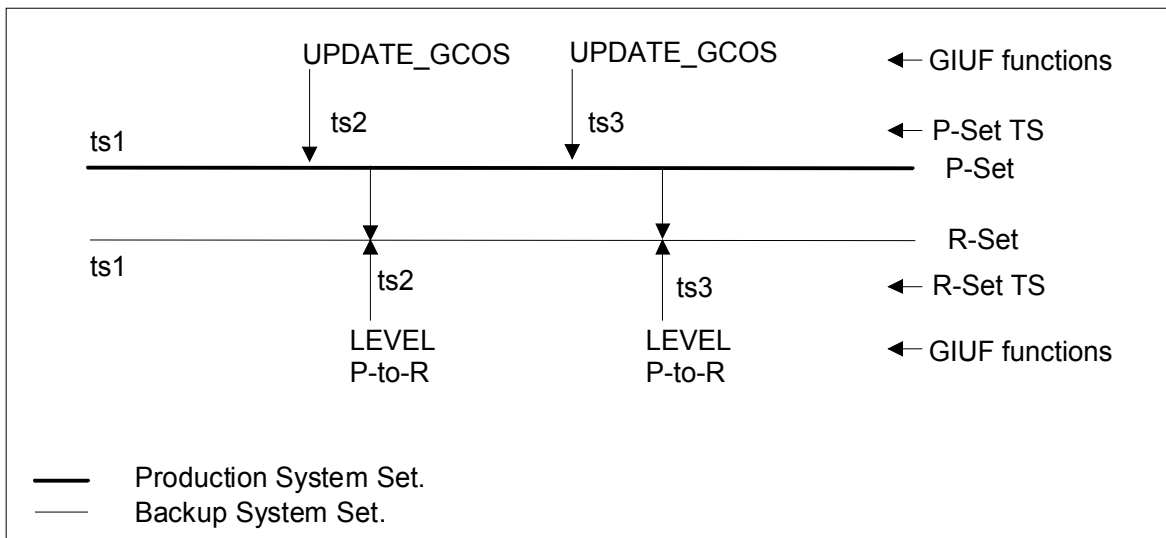
Figure 4-1. Set Evolution in a P2P configuration



**Set Evolution in an RP configuration**

The P-Set is always the running Set; the R-Set is always the backup Set. The UPDATE\_GCOS function is always performed on the running Set. This allows the new Technical Status to be tried out:

- While conserving the R-Set in non-updated form for backup purposes in the event of unforeseen problems with the new Technical Status.
- So that the new Technical Status may be thoroughly tried out in production before performing the LEVEL function to copy it to the R-Set.



**Figure 4-2. Set Evolution in an RP configuration**

**EXAMPLE:**

G: UPDG INVOL=G99201/G99202:CT;

The GCOS Domain of the P-Set, in RP- and PO configurations (or the non-running Set, in a P2P configuration), is to be updated with a Technical Status from the G-MEDIAs with the identifiers G99201:CT and G99202:CT.

□



#### 4.2.40 The UPDATE\_MI Function

##### Purpose

This function is used to modify the MI configuration in the P-Set, P2-Set, or R-Set, as appropriate to the site, for instance, after a technical model upgrade. UPDATE\_MI functions cannot be used to validate MIs at installation time.

##### Syntax

```
{UPDATE_MI | UPMI} UPD_SET={P | P2 | R} KEY_FILE=file78;
```

##### Parameters

UPD_SET	identifies the Set in which the MI configuration is to be modified.
KEY_FILE	describes, in GCL format, the file containing the key necessary for modification of the user's MI configuration.

##### Environment

The running Set may be the P-Set, P2-Set, or R-Set.

If the output Set is already, or is to become, the running Set, an ISL RESTORE session must be run.

When execution is completed, a warning message is displayed on the console and written in the JOR:

```
* AN ISL RESTORE SESSION IS NECESSARY ON THE OUTPUT SET  
* IF THIS SET IS ALREADY OR IS TO BECOME THE RUNNING SET.
```

##### EXAMPLES:

```
G: UPDATE_MI UPD_SET=R KEY_FILE=MIKEY:MIKEY:CT$MFT;
```

The key to be used is located in the file MIKEY on multifile-tape media MIKEY.

```
G: UPMI UPD_SET=P2 KEY_FILE=SYS.IUF..MIKEY;
```

The key to be used is in the subfile MIKEY located in the SYS.IUF file on the running Set.

□



#### 4.2.41 The VALIDP Function

##### Purpose

This function performs GCOS 7 MI validation on the P-Set.

##### Syntax

```
{VALIDP|VALP} KEY={YES|NO};
```

-----

When KEY=YES:

```
{KEY_FILE|KEYF}=file78;
```

-----

When KEY=NO:

```
{OLDVOL|OLDV}=media:devclass;
```

##### Parameters

###### KEY

Must be defined as YES when the GCOS 7-V9 or GCOS 7-V10 Release as appropriate is installed for the first time on the system from a Key-file provided by the your Bull CSE. (This is the installation procedure for a 'new user'.) There is no default value for this parameter which must be stated explicitly.

If NO, a previous GCOS 7-V9 or GCOS 7-V10 Release as appropriate is already available on the system, and no Key-file is provided. (This is the installation procedure for an 'update user'.)

**NOTE:** This key cannot be entered directly via the console.

-----

When KEY = YES:

###### KEY\_FILE

defines the file description of the key used to validate the MI configuration.

-----





When KEY = NO:

OLDVOL identifies the media name and devclass of the old GCOS 7-V9 or GCOS 7-V10 disk as appropriate containing the SYS.CATALOG.

### Environment

The running Set may be the P-Set, a P2-Set, or an R-Set.

If the output P-Set is already, or is to become, the running Set, an ISL RESTORE session must be run.

### Detailed purpose

The job requests the VALIDP parameters, if these have not been submitted with the command.

The JCL IUF\_VALMI is called, to perform the following operations:

- Read and decode the key, or get the MI configuration from the old SYS.SYSTEM file, depending on the KEY parameter.

When KEY = YES:

- Display the MIs to be validated.
- Request agreement on the displayed MI list.
- Validate the MI's concerned, and create the MICONF code in the SYS.SYSTEM file of the P-Set.

When KEY = NO:

- Retrieve the MI configuration from the old SYS.SYSTEM file, and validate it on the SYS.SYSTEM file of the P-Set.

The following IUF\_CONFSET member parameter is updated in the SYS.IUF library of the validated Set:

- REFDISK = NO



The following members of the SYS.IUF library:

```
IUF_SYSOPT2      (for GCOS 7-V9)
IUF_SYSOPT2_XTA (for GCOS 7-V10)
IUF_PRIVATE_MI
```

are input data files used by the H\_SYSOPT load module.

IUF\_SYSOPT2 or IUF\_SYSOPT2\_XTA as appropriate to the GCOS 7 Release contains standard MI data. The IUF\_PRIVATE\_MI reference member may be replaced, at the request of your Bull CSE, by a new IUF\_PRIVATE\_MI member containing MI data specific to the site, and missing from the current IUF\_SYSOPT2 or IUF\_SYSOPT2\_XTA member.

At each update of the GCOS Domain, an empty IUF\_PRIVATE\_MI reference member replaces the current one, because the old site-specific MI data becomes standard MI data, that is normally contained in an updated IUF\_SYSOPT2 or IUF\_SYSOPT2\_XTA member. If this is not the case, a new IUF\_PRIVATE\_MI site-specific member must be delivered by your Bull CSE.

When execution is completed, a warning message is displayed on the console and written in the JOR:

```
* AN ISL RESTORE SESSION IS NECESSARY ON THE OUTPUT SET
* IF THIS SET IS ALREADY OR IS TO BECOME THE RUNNING SET.
```

#### EXAMPLES:

```
G: VALP KEY=YES KEY_FILE=MIKEY:MIKEY:CT$MFT;
```

GCOS 7 MI validation is performed on a P-Set. The key to be used is contained in the file MIKEY on the multifile tape MIKEY.

```
G: VALP KEY=NO OLDVOL=PSETV9:MS/FSA;
```

GCOS 7 MI validation is performed on a P-Set. The MI configuration to be used is retrieved from the SYS.SYSTEM file cataloged in the SYS.CATALOG file, which is situated on the disk PSETV9.

□

#### NOTE:

The K-MEDIA is defined as:

```
MD=MIKEY   DVC=dvc   EFN=MIKEY
```



#### 4.2.42 The VALIDP2 Function

##### Purpose

This function performs GCOS 7 MI validation on the P2-Set.

##### Syntax

```
{VALIDP|VALP} KEY={YES|NO};
```

-----

When KEY=YES:

```
{KEY_FILE|KEYF}=file78;
```

-----

When KEY=NO:

```
{OLDVOL|OLDV}=media:devclass;
```

##### Parameters

KEY

Must be defined as YES when the GCOS 7-V9 or GCOS 7-V10 Release as appropriate is installed for the first time on the system from a Key-file provided by your Bull CSE. (This is the installation procedure for a 'new user'.) There is no default value for this parameter which must be stated explicitly.

If NO, a previous GCOS 7-V9 or GCOS 7-V10 Release as appropriate is already available on the system, and no Key-file is provided. (This is the installation procedure for an 'update user'.)

**NOTE:** This key cannot be entered directly via the console.

-----

When KEY=YES:

KEY\_FILE

defines the file description of the key used to validate the MI configuration.

-----



When KEY = NO:

OLDVOL identifies the media name and devclass of the old GCOS 7-V9 or GCOS 7-V10 disk as appropriate containing the SYS.CATALOG.

### Environment

The running Set may be the P-Set, a P2-Set, or an R-Set.

If the output P2-Set is already, or is to become, the running Set, an ISL RESTORE session must be run.

### Detailed purpose

The job requests the VALIDP parameters, if these have not been submitted with the command.

The JCL IUF\_VALMI is called, to perform the following operations:

- Read and decode the key, or get the MI configuration from the old SYS.SYSTEM file, depending on the KEY parameter.

When KEY = YES:

- Display the MIs to be validated.
- Request agreement on the displayed MI list.
- Validate the MI's concerned, and create the MICONF code in the SYS.SYSTEM file of the P2-Set.

When KEY = NO:

- Retrieve the MI configuration from the old SYS.SYSTEM file, and validate it on the SYS.SYSTEM file of the P2-Set.

The following IUF\_CONFSET member parameter is updated in the SYS.IUF library of the validated Set:

- REFDISK = NO



The following members of the SYS.IUF library:

IUF\_SYSOPT2 (for GCOS 7-V9)  
IUF\_SYSOPT2\_XTA (for GCOS 7-V10)  
IUF\_PRIVATE\_MI

are input data files used by the H\_SYSOPT load module.

IUF\_SYSOPT2 or IUF\_SYSOPT2\_XTA as appropriate to the GCOS 7 Release contains standard MI data. The IUF\_PRIVATE\_MI reference member may be replaced, at the request of your Bull CSE, by a new IUF\_PRIVATE\_MI member containing MI data specific to the site, and missing from the current IUF\_SYSOPT2 or IUF\_SYSOPT2\_XTA member.

At each update of the GCOS Domain, an empty IUF\_PRIVATE\_MI reference member replaces the current one, because the old site-specific MI data becomes standard MI data, that is normally contained in an updated IUF\_SYSOPT2 or IUF\_SYSOPT2\_XTA member. If this is not the case, a new IUF\_PRIVATE\_MI site-specific member must be delivered by your Bull CSE.

When execution is completed, a warning message is displayed on the console and written in the JOR:

```
* AN ISL RESTORE SESSION IS NECESSARY ON THE OUTPUT SET  
* IF THIS SET IS ALREADY OR IS TO BECOME THE RUNNING SET.
```

**EXAMPLES:**

```
G: VALP2 KEY=YES KEY_FILE=MIKEY:MIKEY:CT$MFT;
```

GCOS 7 MI validation is performed on a Slave P-Set. The key to be used is contained in the file MIKEY on the multifile tape MIKEY.

```
G: VALP2 KEY=NO OLDVOL=PSETV9:MS/FSA;
```

GCOS 7 MI validation is performed on a P-Set. The MI configuration to be used is retrieved from the SYS.SYSTEM file cataloged in the SYS.CATALOG file, which is situated on the disk PSETV9.

□

**NOTE:**

The K-MEDIA is defined as:

```
MD=MIKEY DVC=dvc EFN=MIKEY
```



#### 4.2.43 The VALIDR Function

##### Purpose

This function performs GCOS 7 MI validation on the R-Set.

##### Syntax

```
{VALIDR|VALR} KEY={YES|NO};
```

-----

When KEY = YES:

```
{KEY_FILE|KEYF}=file78
```

-----

When KEY = NO:

```
{OLDVOL|OLDV}=media:devclass
```

##### Parameters

KEY

Must be defined as YES if the GCOS 7-V9 or GCOS 7-V10 Release as appropriate is being installed for the first time on the system, from a Key-file provided by your Bull CSE. (This is the installation procedure for a 'new user'.) **There is no default value for this parameter which must be stated explicitly.**

If NO, a previous GCOS 7-V9 or GCOS 7-V10 Release as appropriate is already available on the system, and no Key-file is provided. (This is the installation procedure for an 'update user'.)

**NOTE:** This key cannot be entered directly via the console.

-----

When KEY = YES:

KEY\_FILE

defines the file description of the key used to validate the MI configuration.

-----



When KEY = NO:

OLDVOL identifies the media name and devclass of the old system disk containing the SYS.CATALOG file.

### Environment

The running Set must be the R-Set.

This function is not allowed in PO configurations.

### Detailed purpose

The job requests the VALIDR parameters, if these have not been submitted with the command.

The JCL IUF\_VALMI is called, to perform the following operations:

- Read and decode the key, or get the MI configuration from the old SYS.SYSTEM file, depending on the KEY parameter.

When KEY=YES:

- Display the MIs to be validated.
- Request agreement on the displayed MI list.
- Validate the MI's concerned, and create the MICONF code in the SYS.SYSTEM file.

When KEY = NO:

- Retrieve the MI configuration from the old SYS.SYSTEM file, and validate it on the running SYS.SYSTEM.

The following IUF\_CONFSET member parameters are updated in the SYS.IUF library:

- CONF = RP
- REFDISK = NO

The following members of the SYS.IUF library:

IUF\_SYSOPT2 (for GCOS 7-V9)  
IUF\_SYSOPT2\_XTA (for GCOS 7-V10)  
IUF\_PRIVATE\_MI

are input data files used by the H\_SYSOPT load module.



IUF\_SYSOPT2 or IUF\_SYSOPT2\_XTA as appropriate to the GCOS 7 Release contains standard MI data. The IUF\_PRIVATE\_MI reference member may be replaced, at the request of your Bull CSE, by a new IUF\_PRIVATE\_MI member containing MI data specific to the site, and missing from the current IUF\_SYSOPT2 or IUF\_SYSOPT2\_XTA member.

At each update of the GCOS Domain, an empty IUF\_PRIVATE\_MI reference member replaces the current one, because the old site-specific MI data becomes standard MI data, that is normally contained in an updated IUF\_SYSOPT2 or IUF\_SYSOPT2\_XTA member. If this is not the case, a new IUF\_PRIVATE\_MI site-specific member must be delivered by your Technical Support.

When execution is completed, a warning message is displayed on the console and written in the JOR:

```
* AN ISL RESTORE SESSION IS NECESSARY ON THE OUTPUT SET
* IF THIS SET IS ALREADY OR IS TO BECOME THE RUNNING SET.
```

**EXAMPLES:**

```
G: VALR KEY=YES KEY_FILE=MIKEY:MIKEY:CT$MFT;
```

GCOS 7 MI validation is performed on the R-Set and the file containing the key to be used is called MIKEY and located on the multi-file tape MIKEY.

```
G: VALR KEY=NO OLDVOL=PSETV9:MS/FSA;
```

GCOS 7 MI validation is performed on the R-Set. The MI configuration to be used is retrieved from the disk indicated in the SYS.SYSTEM file cataloged in the SYS.CATALOG file, which is situated on the disk PSETV9.

□





## 4.3 IUF Jobs

### 4.3.1 The INSTAL\_ALL Job

#### **Purpose**

Allocates empty GCOS 7 files and restores loaded system files (belonging to the five file Domains) from the G-MEDIA, DOMAINS-MEDIA and FW-MEDIA to a work disk which will become the P-Set or the R-Set.

The G-MEDIA, DOMAINS-MEDIA and FW-MEDIA may be either tapes or cartridges.

This job creates either a P-Set or an R-Set, as appropriate. However, it is mainly used to create a P-Set when installing a new GCOS 7 Release.

#### **Syntax**

```
EJ INFILE=INSTAL_ALL:G-MEDIA-md:<tape-dvc>$MFT;
```

#### **Environment**

The INSTAL\_ALL job is executed under the User and Project SYSADMIN.

A complete VOLPREP in FBO format is necessary for a work disk that has been previously used as a system disk.

One, two or three disks are required, according to the capacity of the used disks and the memory size specified at the installation of the Set.



### Detailed purpose

This JCL job is loaded from the `INSTAL_ALL` file, which is stored on the first G-MEDIA.

`INSTAL_ALL` performs the following:

- It requests, by means of SWI
  - The device classes of G-MEDIA, DOMAINS-MEDIA, FW-MEDIA
  - The type of Set to be built (P or R)
  - The type of DSA product to be installed (DNS and/or CNS and/or FCP7)
  - The media, device-class and memory requirements of the future P-Set or R-Set.
  - The memory size of the system
- It clears the future P-Set or R-Set
- It installs GCOS 7 files by
  - allocating empty GCOS 7 files. (Their size depends on the memory space reserved when SET=P)
  - restoring loaded GCOS 7 files from the G-MEDIA
- It installs OLTD, DSA and GSF Domain files from the DOMAINS-MEDIA
- It installs FW Domain files from the FW-MEDIA
- It loads the IUF\_CONFSET member from the IUF\_CONFSETREF member, which is stored in the SYS.IUF library. At the same time, the PSETMD, PSETDVC, (or RSETMD, RSETDVC), CONF and DSA parameters are updated.
- It executes the CONFIG job which updates the System Device Tables in the SYS.SYSTEM file on the newly created system volume.



### 4.3.2 The INSTAL\_GCOS Job

#### Purpose

Allocates empty GCOS 7 files, and restores loaded GCOS 7 files from the G-MEDIA to a work disk which will become the R-Set. The G-MEDIA may consist of either tapes or cartridges.

This job is used to create an R-Set when installing a new GCOS 7 Release. It is provided as a backup solution. When this solution is adopted, the other Domains are delivered by your Local Distribution Center on separate DOMAINS\_MEDIA (i.e. one media per Domain, as for previous GCOS 7 Releases).

#### Syntax

```
EJ INFILE=INSTAL_GCOS:G-MEDIA-md:<tape-dvc>$MFT;
```

#### Environment

A complete VOLPREP necessary for a work disk that has previously been used as a system disk.

The INSTAL\_GCOS job is executed under the User and Project SYSADMIN.

#### Detailed purpose

This job consists of JCL loaded from the file INSTAL\_GCOS, stored on the first G-MEDIA. The INSTAL\_GCOS file is loaded at G-MEDIA creation time from the IUF\_INSTAL\_GCOS JCL member.

The job runs the INSTAL\_GCOS job which:

- Requests, by SWI, the MEDIA and DEVCLASS of the future R-Set disk, as well as the input DEVCLASS of the G-MEDIA.
- Clears the future R-Set without track formatting.
- Allocates empty GCOS 7 files, with an organization of SEQ or NONE as appropriate, on the future R-Set disk.
- Restores other GCOS 7 files from the G-MEDIA, to the future R-Set disk, with the options OUTALC and CATNOW.
- Loads the IUF\_CONFSET subfile from the IUF\_CONFSETREF subfile in the SYS.IUF file, with updating of the RSETMD and RSETDVC parameters.
- Updates the R-Set SITE.CATALOG accordingly.



### 4.3.3 The INSTAL\_PSET Job

#### Purpose

Allocates empty GCOS 7 files and restores loaded system files (belonging to the five file Domains) from the SET-MEDIA to a work disk which will become the P-Set.

The SET-MEDIA may be either tapes or cartridges.

This job is used to create a P-Set disk when installing a new GCOS 7 Release.

#### Syntax

```
EJ INFILE=INSTAL_PSET:SET-MEDIA-md:<tape-dvc>$MFT;
```

#### Environment

The INSTAL\_PSET job is executed under the User and Project SYSADMIN.

A complete VOLPREP is necessary for a work disk that has been previously used as a system disk.

One or two disks are required, according to the capacity of the used disks and the memory size specified at the installation of the Set.

#### Detailed purpose

This JCL job is loaded from the INSTAL\_PSET file, which is stored on the first SET-MEDIA (FSN=2).

INSTAL\_PSET is created by the SAVE\_SET job from the IUF\_INSTAL\_PSET JCL member.

INSTAL\_PSET performs the following:

- It requests, by means of SWI, the media, device-class and memory requirements of the future P-Set.
- It clears the future P-Set.
- It allocates empty GCOS 7 files on the future P-Set. (Their size depends on the memory size.)
- It restores other GCOS 7, OLTD, DSA, GSF and FW files from the SET-MEDIA to the future P-Set, with the option OUTALC and CATNOW.

It loads the IUF\_CONFSET member from the IUF\_CONFSETREF member, which is stored in the SYS.IUF library. At the same time, the PSETMD, PSETDVC, CONF and DSA parameters are updated.



#### 4.3.4 The INSTAL\_RSET Job

##### **Purpose**

Allocates empty GCOS 7 files, and restores loaded system files (belonging to the five file Domains) from the SET-MEDIA to a work disk which will become the R-Set disk. The SET-MEDIA may be either tapes or cartridges.

This job is used mainly to create the Slave R-Sets (Refer to the chapter *Multi-System Installation and Updating*). It may also be used to rebuild an R-Disk (should the existing R-Disk be destroyed) preceded by the SAVE\_SET function (to make a save from the P-Set).

##### **Syntax**

```
EJ INFILE=INSTAL_RSET:SET-MEDIA-md:<tape-dvc>$MFT;
```

##### **Environment**

A complete VOLPREP is necessary for a work disk that has previously been used as a system disk.

The INSTAL\_RSET job is executed under the User and Project SYSADMIN.

##### **Detailed purpose**

This job consists of a JCL loaded from the file INSTAL\_RSET, stored on the first SET-MEDIA. INSTAL\_RSET is created by the SAVE\_SET job from the IUF\_INSTAL\_RSET JCL member.

Runs the INSTAL\_RSET job, which:

- Requests, with the command SWI, the MEDIA and DEVCLASS of the future slave R-Set, as well as the Release relevant to the running disk.
- Clears the future slave R-Set without track formatting.
- Allocates empty GCOS 7 files, on the future slave R-Set system disk.
- Restores other GCOS 7, FW, OLTD, GSF and DSA files from the SET-MEDIA, to the future slave R-Set, with the options OUTALC and CATNOW.
- Loads the IUF\_CONFSET member from the IUF\_CONFSETREF member in the SYS.IUF library, with updating of the RSETMD, RSETDVC and DSA parameters.





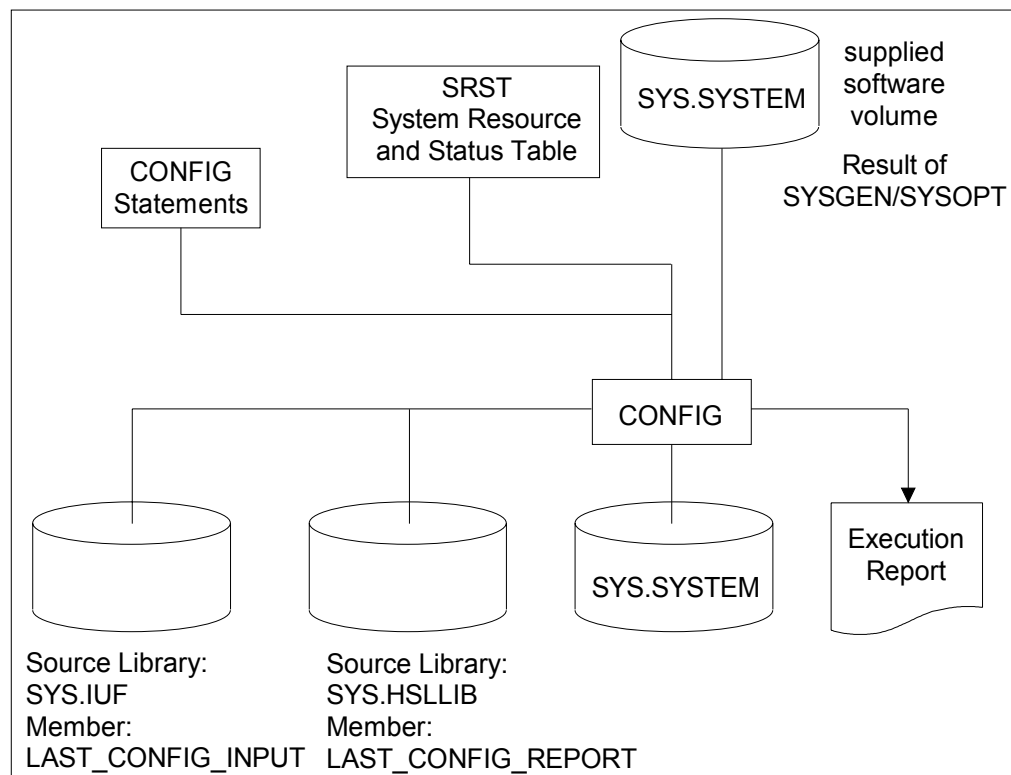
---

## 5. CONFIG (Configuration)

### 5.1 Introduction

This chapter describes the configuration of GCOS 7 systems.

The GCOS 7 software is supplied on a volume to be configured by the user for his site. CONFIG adapts the delivered software volume to the specific environment of the user's applications, and is carried out at his site. Figure 5-1 shows an overview of the CONFIG utility.



**Figure 5-1. CONFIG Overview**

### 5.1.1 Input

The principal input to CONFIG includes:

- The SRST (System Resource and Status Table) from the system disk, defining the status of the various devices available.
- The SYS.SYSTEM file, on the system disk, containing the operating system and the system device tables used at Initial System Load (ISL).
- A (sequential) configuration statement file containing statements (summarized and described later in this chapter), which specify the characteristics of the configuration.





### 5.1.2 Output

The results produced by CONFIG include:

- An execution report of the configuration of the site, in terms of:
  - hardware device configuration
  - explicit configuration commands
  - memory parameters
  - scheduling parameters
  - configured job classes
  - effective configuration commands
  - successful completion message, if appropriate
  - unsuccessful completion message(s), if appropriate.

This output is also written to the SYS.HSLLIB library; for this reason, when running CONFIG you must have WRITE access to SYS.HSLLIB.

- An updated system volume.
- The commands used for the CONFIG run are recorded in the SYS.IUF library, so when running CONFIG you must also have WRITE access to SYS.IUF.

**NOTE:**

If CONFIG is run several times on the same system disk, the changes specified by the configuration statements are made only with respect to the original volume as delivered. Thus a second run of CONFIG, with one specified change for example, will cancel all previous configuration changes to the system volume, only the one specified change being performed. This feature provides a constant reference point, the status of the original volume.



## 5.2 System Device Tables

GCOS 7 needs device tables (called System Device Tables), in order to access the hardware available at the site. Initial System Load (ISL) and CONFIG build and update the system device tables stored in the SYS.SYSTEM file on the delivered system volume. These tables initially describe a standard hardware configuration that is then adapted by CONFIG to meet the individual requirements of the site.

### CONFIG and ISL

Configuration statements define the specific software requirements of the site. When CONFIG has updated the system volume, start the newly configured system using the following ISL options:

1. RESTORE

Restores the backing store from the configured master system volume.

2. RESTART

If the restore option is not issued, the backing store remains unchanged and the system status, after ISL, is the same as that of the original, plus any updating due to the RESTART option.

If a correct RESTART option is not issued, the following message is displayed at the operator's console:

```
CONFIG NEEDS RESTORE + RESTART CLEAN
```

Followed by: IL01 OPTION?

Normally, an incident such as a system crash (or the job being stopped by the operator) during a CONFIG run does not affect the system volume. However, a CONFIG abort during the switch from "Old" to "New" state of the system volume may leave the SYS.SYSTEM file in an "undefined" state. If CONFIG is run again, the file will reach its correct state; but, if there is no re-run and an ISL RESTORE option exists, the operator will receive the message:

```
"CONFIGURATION PARAMETERS LOST DO YOU WANT TO RESTORE ?"
```

If the reply is NO, RESTORE is not carried out; if the reply is YES, a special restore returns the backing store to its originally defined state. Run CONFIG again to configure the system device tables.



## 5.3 Execution Procedure

### 5.3.1 Basic Procedure

#### Format

S: IV CONFIG;

If there are no configuration statements, CONFIG acts on the resident system volume. Note that the IV (INVOKE) is equivalent to the EJ CONFIG operator command.

#### NOTE:

In an RP or P2P configuration, or with multi-system disks, after any reconfiguration evolution (FIRMGEN or BLOAD) is followed by a CONFIG on the running system disk, the first initialization of a non-running system disk must be performed with an INIT (never with a RELOAD).



### 5.3.2 Advanced Procedure

CONFIG may be invoked as follows:

The GCL request:

```
S: EJ MYCONFIG;
```

is submitted, to enter the job MYCONFIG which has been stored using the following JCL format:

```
IV CONFIG VALUES=
( [MEDIA=media, DEVCLASS=devclass] [, DUMP={NO|DATA}
  [, COM_EFN={ DUMMY |com_efn} [ ,COM_ASP={ NIL |com_asp}]]]
  [, USER_CAT={ SITE.CATALOG |catalog-name}] )
```

#### Parameters

MEDIA	names the media containing the SYS.CATALOG to which the SYS.SYSTEM file to be configured is attached.
DEVCLASS	names the device-class for the media containing the SYS.CATALOG to which the SYS.SYSTEM file to be configured is attached. This parameter is mandatory if MEDIA is specified. If MEDIA and DEVCLASS are omitted, the SYS.SYSTEM attached to the resident SYS.CATALOG is configured.
DUMP	specifies the action to be taken in the event of a CONFIG abort, as follows:  NO defines that no dump is to be performed. DATA requests a dump of the CONFIG job.
COM_EFN	names the file containing the configuration statements. The default value is DUMMY.
COM_ASP	specified with COM_EFN, assigns parameters for the file containing the configuration statements. The default value is NIL.
USER_CAT	names the catalog to which the file containing the CONFIG statements is attached; by default, this is SITE.CATALOG.



### EXAMPLE:

```
S: EJ MYCONFIG;
```

which runs the standard job stored under the name MYCONFIG, containing the following:

```
$JOB MYCONFIG USER=OPERATOR;
```

```
IV CONFIG VALUES=  
(DUMP=DATA, COM_EFN=DISK-FILE,  
  COM_ASP='DVC=device, MD=media, SHARE=DIR, MB=IN_CONFIG');
```

```
ENDJOB;
```

### NOTES:

1. In an RP or P2P configuration, or with multi-system disks, after any reconfiguration evolution (FIRMGEN or BLOAD) is followed by a CONFIG on the running system disk, the first initialization of a non-running system disk must be performed with an INIT (never with a RELOAD).
2. The parameters DVC, MD, and SHARE present in the argument of COM\_ASP in the example above are used only when <DISK-FILE> is uncataloged.





## 5.4 Configuration Statements

### 5.4.1 Syntax and Convention

Configuration statement syntax is similar to that of JCL, except that:

1. There is no continuation character.
2. The end of record acts as a delimiter.

All normal JCL syntax facilities remain:

1. Free format.
2. Keyword parameters may be supplied in any sequence.
3. There may be any number of blanks before or after a separator. Thus, if a comma or semi-colon used as a separator is preceded by or followed by blanks, such blanks are ignored.
4. Use of positional and/or keyword parameters.
5. A comma separator is not mandatory between the arguments of a statement.
6. It is possible to insert comments in a set of configuration statements:

```
{COMMENT|COMM|C} comment text;
```

7. "\$" signs may optionally prefix statement names.

The same conventions, including notation variables, are used as in JCL description; i.e.:

digitN	indicates a decimal string of digits in the range 0 through 9, of maximum length N. For example, for digit3:  123
alphaN	indicates a string of alphabetic characters in the range A through Z, of maximum length N. For example, for alpha6:  AZERTY
identifierN	indicates a string of length N containing alphanumeric characters (A through Z, 0 through 9), beginning with an alpha. For example, for identifier4:  A38Z



stringN indicates any string of maximum length N characters, enclosed between single quotes. For example, for string10:

```
'AZERTY12+1'
```

A single quote ( ' ) may be used in a protected string, by entering it twice ( ' ' ). For example:

```
PRLOG COMMAND='A AB' 'AB' ' A'
```

is interpreted as: A AB'AB' A because a double quote ( ' ' ) within a protected string represents ( ' ).

#### 5.4.2 Summary of Configuration Statements

ACCOUNT	Accounting: specifies the accounting function.
ACTSIZE	Accounting file size: specifies the size of the accounting file.
BJSIMU	BEFORE JOURNAL simultaneities: specifies the maximum number of jobs simultaneously able to use the Before Journal.
DEFORG	Defines the default volume organization for sites that support both VBO and FBO volumes.
DEVTRACE	Device for trace: specifies the trace default device.
DJPCTL	Distributed Job Processing Control: specifies the maximum number of transfer processors allowed at any one time.
DUALSYS	Dual systems: specifies the serial number of the second CPU in a pair of coupled systems (if applicable).
DUMPACT	DUMPACT utility: specifies the OCL statement to be executed automatically whenever an accounting file becomes full.
FILEOPT	Enables the activation of the site-specific file management system options.
FILSHARE	File sharing: specifies the maximum number of jobs that may share any single file.



---

GAC	General Access Control: specifies the GAC parameters.
INPUTCTL	Specifies the input control options and parameters.
JOBCLASS	Mono-letter Job Class: specifies the system default values for a mono-letter job class (A to Z).
JBCLASS2	Double-letter Job Class: specifies the system default values for a double-letter job class (AA to ZZ). Up to 220 double-letter job classes may be specified in a single CONFIG session.
JORSIZE	Job Occurrence Report size: specifies a maximum limit of lines that may be printed for any single JOR.
JPPCDIM	JPPC dimension. Determines which of the five variants of the system service job, JPPC, will allocate segments for TDS applications using XCP2 or IQS.
LSYS	Local system: specifies the local system's name, for use in a DSA network environment.
MAXCAT	Maximum number of catalogs: specifies the maximum number of catalogs that may be known in the system at any one time.
MAXDEF	Maximum number of DEFINES: specifies the maximum number of DEFINES that may be active in the system at any one time.
MAXFILE	Maximum number of files: specifies the maximum number of files that may be active at any one time.
MAXIODEV	Maximum Input Output devices: specifies the maximum allowed number of I/O devices per station.
MAXJOB	Maximum number of jobs: specifies the maximum number of jobs that may be known to the system at any one time.
MAXTAPE	Maximum number of tape cartridges: specifies the maximum number of tape cartridges that may be known to the system.
MAXTASK	Specifies the maximum number of tasks that are to be allowed to be executing simultaneously.





MULTLEV	Multi-programming level: specifies the maximum number of jobs that may be "started" (i.e., in a post-scheduling state) at one time.
OWCLASS	Output Writer class: specifies the PRIORITY and MEDIA system defaults applicable to the output writer devices connected to the system.
OWDEVICE	Output Writer Device: specifies the system defaults to be applied for the CLASS parameter when used in SOW commands, for the output writer devices applicable.
OWDFLT	Output Writer default: specifies the system defaults for the CLASS, DEVCLASS, TAPE and Banner parameters in JCL/GCL statements.
PRLOG	Print SYS.ERLOG file: specifies the threshold value at which printing of the SYS.ERLOG file is to be triggered.
PRLOGC	Print the SYS.LOGC file: specifies the threshold value at which printing of the SYS.LOGC file is to be triggered.
QUEUED	Cache mechanism on directory and BAM (Block Allocation Map) blocks for queued files.
SADMOPT	System administrator options: selects options specific to the System Administrator.
SECOPT	Security options: selects options specific to security.
SITEOPT	Site Options: activation of specific options on site.
VCAMSCN	VCAM Session Connection
VOLORG	Defines the authorized volume organization on sites that support both VBO and FBO volumes.
XAS	Extended Address Space



### 5.4.3 The ACCOUNT Statement

#### Purpose

Specifies all or part of the accounting function. This parameter may be modified only by a subsequent run of CONFIG.

#### Format

```
ACCOUNT { ALL |NO| [, STEP] [, JOB] [, END] [, USER] [, JOBOUT]
          [, JOBEX] | [, MPROC] [, SVIOL] [, FVIOL] [, TDS]
          [, FILEUPD] [, TW] } ;
```

#### Parameters

If specifying neither ALL nor NO, at least one of the other parameters should be used; otherwise the default of ACCOUNT ALL; will apply. If specifying more than one parameter, the use of commas as separators is required.

ALL	every type of accounting record is to appear in the accounting file together with any user-specified accounting records; ALL is the default value.
NO	no accounting is required.
STEP	step records are to be written to the accounting file.
JOB	job records are to be written to the accounting file. This parameter is retained for compatibility with earlier releases and is the equivalent of JOBOUT plus JOBEX.
END	a record is to be written to the accounting file at each shutdown (GCL command TSY51) or RESTART.
USER	enables the user to write information to the accounting file.
JOBOUT	job output records are to be written to the accounting file.
JOBEX	job execution records are to be written to the accounting file.
MPROC	multi-process step records are to be written to the accounting file; this parameter may not be specified without STEP.



SVIOL	system violation records are written to the accounting file.
FVIOL	file violation records are written to the accounting file.
TDS	TDS accounting records (TDS session and TDS user activity records) are to be written to the accounting file.
FILEUPD	file updating records are to be written to the accounting file.
TW	GTWriter accounting records are to be written to the accounting file.

**EXAMPLES:**

ACCOUNT ALL;

This statement specifies that all records are applicable.

ACCOUNT STEP, JOB, END, USER;

This statement specifies that only the STEP, JOBOUT, JOBEX, END and USER records are applicable.

ACCOUNT NO;

This statement specifies that no accounting is required.

□

**Remarks:**

If ALL is specified, the Working Set cost of accounting at step level is approximately:

- 5K at step termination
- 5K at job termination.

Likewise, events corresponding to other keywords each cost approximately the same amount of Working Set.

Job management, the various job accounting facilities available are described together with the construction of the accounting file, in the System Administrator's Manual.



#### 5.4.4 The ACTSIZE Statement

##### Purpose

Specifies the size of the virtual memory management (VMM) accounting file. Once set, this parameter may be modified only by re-running CONFIG.

##### Format

```
ACTSIZE { 200 |nnnn}
```

##### Parameters

The size specified by this statement is in units of backing store allocation, each of which is equal to 8K bytes.

nnnn                      a decimal value of up to 4 digits, ranging from 1 through 2000, to define the size of an accounting file in units of 8K. The default value is 200 units (1600K).

##### EXAMPLE:

```
ACTSIZE 100;
```

This statement specifies that provision be made for 100 units (800K) for accounting information.

□



**Remarks:**

The accounting mechanism is based on two accounting files working in flip-flop mode; thus, if a value n is used (ACTSIZE n;), the total cost is 2 x n units of backing store.

When one accounting file is full, one of the messages:

AC01 GCOS: ACCOUNTING FULL. RUN DUMPACT

AC02 GCOS: ACCOUNTING FULL. DUMPACT COMMAND SUBMITTED

is displayed at the main operator console, depending on the DUMPACT statement in CONFIG; accounting information is written to the second file.

The size of the various accounting records is specified in the accounting documentation, for instance:

JOBOUT                    is specified as 181 bytes

JOBEX                    is specified as 115 bytes

STEP                    is specified as up to 1383 bytes.



### 5.4.5 The BJSIMU Statement

#### Purpose

Specifies the maximum number of jobs which, at any given time, may use the Before Journal file. These jobs may be of any type (IOF, TDS, Batch).

#### Format

```
BJSIMU nnn ;
```

#### Parameters

nnn specifies a decimal value of up to 3 digits, ranging from 1 through 254, subject to the following conditions:

- if the value is out of range, 8 is assumed.
- if no value is given, then:
  - if JMAX is greater than 8, the default value is (JMAX-4).
  - if JMAX is less than or equal to 8, the default value is (JMAX-2).

JMAX specifies the maximum number of jobs that may run concurrently, as specified in the MULTLEV statement.

#### EXAMPLES:

```
BJSIMU 40 ;
```

The specified value of 40 is used.

```
BJSIMU ;
```

If the JMAX value in the MULTLEV statement is greater than 8, a default value of (JMAX 4) is used; otherwise, if JMAX is less than or equal to 8, a default of (JMAX - 2) is used.

□



**Remarks:**

Use of this statement gives:

- a saving of backing store space (of approximately 0.4K per job using the Journal), as compared with the default value.
- a possible saving of memory space when the Before Journal is activated (of a maximum of approximately 1.5K).
- a saving of 1 track per job using the Before Journal, in the SYS.JRNAL file.

When a value of BJSIMU is specified through CONFIG, it must be consistent with the size given for the SYS.JRNAL file:

n: value of BJSIMU

m: Size of SYS.JRNAL in blocks

$m = (n + 2) * 132$  blocks of 512 bytes



---

## 5.4.6 The DEFORG Statement

### Purpose

On a site that supports both VBO and FBO disks, DEFORG defines the default disk type used by the VOLPREP utility in the case of a volume preparation of a CKD disk (for data purposes only).

### Format

```
DEFORG {VBO|FBO};
```

### Parameters

VBO                               The disk is prepared in VBO format.

FBO                               The disk is prepared in FBO format.

### Remark:

This option is taken into account only if the site supports a configuration of mixed disk types (and VOLORG default value is ALL).





## 5.4.7 The DEVTRACE Statement

### Purpose

Provides a trace of events occurring on any device, for display at the operator console.

The type of event may be selected; for example, Abnormal, Attention, Warning, Alarm. If the specified type of event occurs on any device, it is reported by appropriate messages at the operator console.

### Format

```
DEVTRACE [ABN] [ATN] [WARN] [ALARM] ;
```

### Parameters

If there is no DEVTRACE statement, no device trace is provided at the operator console. If DEVTRACE is specified, at least one parameter must be defined.

ABN specifies that every abnormal event (in the SYS.ERLOG sense) is to be displayed at the operator console.

ATN specifies that every attention message (in the SYS.ERLOG sense) is to be displayed at the operator console.

WARN specifies that every warning message ("normal" errors, in the SYS.ERLOG sense) is to be displayed at the operator console.



ALARM

specifies that an alarm message be given at the operator console, at each nth occurrence of an abnormal event warning. The threshold value depends on device type.

Device	n
URC	0
MTC	100
MSC	10

**EXAMPLE:**

```
DEVTRACE WARN;
```

Only warning messages are displayed at the operator console.





### 5.4.8 The DJPCTL Statement

#### Purpose

The Distributed Job Processing ConTroL command specifies the maximum number of transfer processors that may exist at any one time on the site.

#### Format

```
DJPCTL MAXTFSIT={ 2 | nnn} MINEVERY={ 15 | nn };
```

#### Parameters

MAXTFSIT	introduces a decimal value of up to 3 digits, ranging from 0 to 255, to assign an upper boundary to the number of GTPs (Generalized Transfer Processors) that may be "started" (executing or idle) at any one time on the system being configured. These GTPs are used to transfer jobs, files or output files to other sites known to the system's network description. The default value is 2.
MINEVERY	introduces a decimal value of up to 2 digits, ranging from 1 to 15, to assign a minimum delay in minutes between two consecutive executions of a repeated EFTR/EJR request (parameter EVERY=). The default is 15 minutes.



**Remarks:**

The MAXTFSIT parameter's effect may be overridden dynamically, using the GCL request MDDJP.

This value also limits the number of Readers + "local" GTPs that may start. A "local GTP" is a GTP doing local file transfers; INFILE and OUTFILE are local files.

If local file transfers are performed before job submission, GTPs are created which, when finished, do not disappear but become IDLE jobs, thus preventing potential Readers from starting.

On the other hand, Readers may be made permanent (remain in EX state) using either the INPUTCTL statement of CONFIG or the GCL command "SIR PERM". In this case, if the number of Readers reaches the value of MAXTFSIT, no local GTP can start.

Should such a situation arise, the operator should issue:

```
MDDJP m SITE=local-site
```

where m is greater than the value defined for MAXTFSIT.



### 5.4.9 The DUALSYS Statement

#### Purpose

For use with coupled systems, specifies the serial number of the second CPU concerned in order to allow the resolution of deadlocks over shared resources.

#### Format

```
DUALSYS cpu-sn [, MYSELF=cpu-sn] ;
```

#### Parameters

cpu-sn specifies an 8-digit hexadecimal value identifying the CPU serial number concerned, and conforming to the following conditions:

- Used after DUALSYS to indicate the CPU serial number of the second CPU in a coupled system configuration. This parameter is mandatory **whenever** the system on which the CONFIG is being run ("system A") has been coupled with another system ("system B"), and its value must accurately reflect the real CPU number of "system B".
- When introduced by the MYSELF keyword, is only required when both:
  - The "system B" is linked to yet another system ("system C"; i.e., other than "system A", on which the CONFIG is being run) also in a second coupled system configuration
  - The CPU serial number of "system A" is greater than that of "system B", **and** the CPU serial number of "system B" is greater than that of "system C" - a situation that would cause resource deadlocks.

If the above 2 conditions are both satisfied, this MYSELF keyword must specify an arbitrary CPU number for "system A" which is **less than** the "real" CPU number specified for "system B" directly after DUALSYS. In this case, no resource deadlocks will be allowed to occur between any of the 3 systems concerned.

With reference to the above "arbitrary" CPU number, check with your Bull representative that the CONFIG statement SITEOPT has not been set. If SITEOPT has been defined, verify that the bit, which triggers checking of the real CPU number, has not been set.



**EXAMPLES:**

```
DUALSYS 87501234;
```

The serial number of the second CPU in a coupled system configuration is specified as 87501234.

```
DUALSYS 85401234, MYSELF=84101234;
```

The serial number of the second CPU of a coupled system is specified as 85401234, and the serial number of the first CPU is specified as 84101234.

□

**Remarks:**

The DUALSYS statement is applicable only when the Coupled System Support software product has been ordered and validated.



### 5.4.10 The DUMPACT Statement

#### Purpose

This command specifies the OCL statement to be executed automatically whenever an accounting file becomes full. It defines the OCL statement that will be used by the system when executing the DUMPACT utility.

#### Format

```
DUMPACT COMMAND={'ocl-request' | NO } ;
```

#### Parameters

ocl-request defines a string containing a maximum of 105 characters, enclosed within single quotes. This must specify a valid operator command to be obeyed, as soon as either of the two-system accounting Virtual Memory files becomes full.

NO is the default value.

#### EXAMPLE:

```
DUMPACT COMMAND='EJR DUMPJOB,,USER.JCLLIB' ;
```

where the JCL contents of the USER.JCLLIB library member DUMPJOB are:

```
$JOB DUMPJOB USER=SYSADMIN PRTY=0;  
STEP H_DUMPACT SYS.HLMLIB OPTION='AP DB' XPRTY=1;  
ASSIGN USERACTF TESTFILE DVC=MS/FSA MD=FSD68;  
ENDSTEP;  
JUMP CONTINUE;  
STEP H_DUMPACT SYS.HLMLIB OPTION='AP 1E' XPRTY=1;  
ASSIGN USERACTF ACCFILE DVC=MS/FSA MD=FSD68;  
ENDSTEP;  
$ENDJOB;
```

□



**Remarks:**

The system default option (when no DUMPACT statement is provided at CONFIG time) is to warn the operator with the repetitive message:

```
*AC01.GCOS: ACCOUNTING FULL.RUN DUMPACT
```

whereupon the operator should start a DUMPACT utility job.

If, on the other hand, a DUMPACT statement is provided with CONFIG, the system uses this statement and the following message is issued:

```
*AC02.GCOS: ACCOUNTING FULL.DUMPACT COMMAND SUBMITTED
```

However, CONFIG does not test the validity of any operator command coded with the COMMAND keyword. In the event of an accounting file becoming full, coding the DUMPACT statement (particularly in the way shown in the example above) will avoid operator intervention, thus reducing delays to production programs.





---

### 5.4.11 The FILEOPT Statement

#### Purpose

This command enables the activation of site-specific file management options.

#### Format

```
FILEOPT  [COMPTS={PREVIOUS|CURRENT} ]  
         [MNTWORK={YES|NO} ]  
         [CTFIRST={YES|NO} ]  
         [LASTDATE={YES|NO} ]  
         [USESITE={YES|NO} ]  
         [CTUNLOAD={YES|NO} ] ;
```

#### Parameters

COMPTS provides compatibility with the specified Technical Status.

This parameter enables the system administrator to keep the file system substantially compatible with an earlier Technical Status and those that follow it. The earlier Technical Status is that used, for instance, on a backup system or the other system in a coupled system configuration. The argument of COMPTS defines the oldest Technical Status that can use the files or catalogs of the current system. GCOS 7 uses this value to decide whether new file system features controlled by COMPTS that are not backward compatible are to be used or not. The system administrator can subsequently update the value when the earlier Technical Status is finally abandoned.

**NOTE:** Only Technical Statuses still supported and significant from a file system point of view may be specified.

COMPTS can be set to PREVIOUS or CURRENT:

**PREVIOUS** (the default value) specifies compatibility of the file system with the Technical Status immediately preceding the current one. Files can allocated no more than 16 extents, per volume.



---

	<p><b>CURRENT</b> specifies compatibility of the file system with the current technical status only (no backward compatibility is required). Files can be allocated dynamically up to 255 extents per volume.</p>
MNTWORK	<p>If set to NO (the default value), when a cataloged file is assigned with an empty media list, an error occurs.</p> <p>If set to YES, when a a cataloged file is assigned with an empty media list:</p> <ul style="list-style-type: none"><li>– if there is at least one tape unit on the site a request is issued for a work tape to be mounted,</li><li>– if there are only cartridge units on the site or if device type CT is recorded in the cataloged file description, a request is issued for a cartridge to be mounted.</li></ul> <p>The retrieval order for devices can be changed by the CTFIRST parameter (see below).</p>
CTFIRST	<p>If set to NO (the default value), the type of media is as described for the MNTWORK parameter above.</p> <p>If set to YES (allowed only if MNTWORK=YES), the order for device retrieval is changed to:</p> <p>CT/LIB, CT/M5/C, CT/M5, MT, CT.</p> <p>The class retained is the first available on the site.</p>
LASTDATE	<p>If set to YES (the default value), the dates of last reference and last modification are updated when the file is closed even if there has been no I/O.</p> <p>If set to NO, the dates of last reference and last modification are updated when the file is closed only if there has been any I/O.</p>
USESITE	<p>If set to NO (the default value), the default volset for the project is used and the function aborts if there is insufficient space.</p> <p>If set to YES, the site volset is used for permanent file space if the default project volset is full.</p>



CTUNLOAD

For a cartridge library only. If YES, at the end of an IOF command, cartridges are automatically dismounted unless the LEAVE option was specified in this command or if the PASS option was specified in this or a previous command.

If NO, at the end of an IOF command, cartridges remain premounted if the step does not contain, explicitly or implicitly, the UNLOAD option. Note, however, that cartridges are ALWAYS dismounted at the end of the job.

**EXAMPLE:**

```
FILEOPT COMPTS=PREVIOUS;
```

The file system must be kept compatible with the previous Technical Status.





## 5.4.12 The FILSHARE Statement

### Purpose

Specifies the maximum number of jobs that may share ANY SINGLE FILE (such as a specific catalog, for instance) concurrently. This number may be modified only by a subsequent run of CONFIG.

### Format

```
FILSHARE { Min (MULTLEV+1,250) | nnn } ;
```

### Parameters

nnn specifies a decimal value of up to 3 digits, ranging from 2 through 250, to define the maximum number of jobs that may ever share the same file concurrently. Each time the system assigns a file, this counts for one job. Note that, if a file is assigned more than once in a step, this still counts for only one job. The default value is Min (MULTLEV+1,250).

### EXAMPLE:

```
FILSHARE 100;
```

No file may ever be shared by more than 100 jobs concurrently.

□



**Remarks:**

FILSHARE is used together with MAXFILE, to determine parameters important to system performance such as:

- the SYS.KNODET block size
- the maximum number of subfiles assigned to a given file.

The following formulas show the precise relationship:

BKSIZE (sys.knodet\_block-size) = the smallest factor of 512  
that is  $\geq 424 + (52 * \text{fileshare})$

DEF\_PER\_BK (number-of-defines-in-a-sys.knodet-block) = integer\_part (bksize/150)

NDF (max. number of DEFINES) = MAXDEF (see MAXDEF statement)

NBBK\_DEF (number-of-blocks-used-to-contain-defines) is then computed using the ratio:

$\text{ndf}/\text{def\_per\_bk}$

$\text{KS (total-size-of-sys.knodet)} = \text{bksize} * (\text{maxfile} + \text{nbbk\_def})$

Of course, this computation may sometimes result in a theoretical SYS.KNODET size greater than that actually built by TAILOR. Should this arise, the system will react in the following way:

1. It computes the ratio:  
 $R = \text{maxfile}/\text{maxdef}$
2. If the result R is in the range 1 through 4:  
MAXFILE and MAXDEF will both be reduced to a value compatible with the size of SYS.KNODET that is consistent with the ratio R.
3. If R is less than 1 or greater than 4:  
An adjustment will be made to set R to 1 (if less) or to 4 (if greater), with the same effect on MAXFILE and MAXDEF as in 2) above.

Depending on the action actually taken, messages will be sent at GCOS 7 initialization:

DF04 MAXIMUM NUMBER OF ASSIGNED FILES HAS BEEN REDUCED TO xxx

DF05 MAXIMUM NUMBER OF DEFINES HAS BEEN REDUCED TO xxx

DF05 MAXIMUM NUMBER OF DEFINES HAS BEEN INCREASED TO xxx



### 5.4.13 The GAC Statement

#### Purpose

Specifies GAC (i.e., General Access Control) parameters.

#### Format

```
GAC [LONGWAIT={ 60 |nnn}] [, LOCKSIZE={ 20 |nnn}]
    [, NBLOCKID={ 100 |nnnn}] ;
```

#### Parameters

**LONGWAIT** supplies a decimal value of up to 3 digits, ranging from 0 through 300, with a default of 60.

If a tenant retains a resource beyond the maximum time specified in LONGWAIT, on initiation of a commitment unit, other requestors of that resource receive the LONGWAIT notification together with the identity of the current tenant.

**LOCKSIZE** supplies a decimal value of up to 3 digits, ranging from 5 through 320, to specify the maximum size of the "lock list" in K bytes, to be used by GAC for holding details of the CIs locked by users of the system. The default value is 20.

**NBLOCKID** supplies a decimal value of up to 4 digits, to specify the maximum number of LOCKIDs that may be active in GAC simultaneously.

LOCKID is the "lock-owner-identifier" allocated:

- at step initiation for batch processing
- at transaction initiation for TDS.

The H\_DEFLOCK primitive returns this value in order to be able to identify requestors on LONGWAIT notifications. The default value is 100.

The maximum possible value for NBLOCKID is 4095. The value of NBLOCKID should satisfy the condition:

$$\text{NBLOCKID} * 20 < \text{LOCKSIZE} * 1024.$$



**EXAMPLE:**

```
GAC LOCKSIZE=50, LONGWAIT=100, NBLOCKID=120;
```

This means that TDS or IOF, as appropriate, will be notified if a waiting time greater than 100 seconds has occurred.

□

**Remarks:**

This GAC statement is applicable only when the GAC and/or TDS product has been ordered and validated; refer to the *GAC-EXTENDED User's Guide*.



#### 5.4.14 The INPUTCTL Statement

##### Purpose

Specifies the input control options and parameters required for the system.

##### Format

```
INPUTCTL [ SJREADER={TERM| NTERM } ]
          [, HSLLIB={ SYS.HSLLIB |xxx} ]
          [, JTRA={TERM| NTERM } ]
          [, HOABORT={YES|NO} ] ;
```

##### Parameters

SJREADER	<p>specifies SJREADER service job behaviour, as follows:</p> <p><b>TERM:</b> specifies that the service job is to terminate as soon as there are no further EJ requests pending.</p> <p><b>NTERM:</b> after a COLD RESTART, when the first EJ request has been processed, the SJREADER service job is to remain in execution (EX) waiting for the next EJ request. This is the default value.</p>
JTRA	<p>specifies JTRA service job handling, as follows:</p> <p><b>TERM:</b> JTRA is to go into the IDLE state, as soon as there are no more jobs to be translated.</p> <p><b>NTERM:</b> JTRA is to wait for the next job to be translated (that is, remains in EX). This is the default value.</p>
HSLLIB	<p>a maximum of 44 characters, identifying the default system HSLLIB; the default value is SYS.HSLLIB.</p>
HOABORT	<p><b>YES:</b> all jobs, aborting at JCL translation time, will have their output (JOR) held automatically. This allows the output to be more easily scanned.</p> <p><b>NO:</b> (default) job outputs are not held unless explicitly requested by the HOLDOUT parameter of EJ or \$JOB.</p>





**EXAMPLE:**

```
INPUTCTL SJREADER=NTERM, JTRA=NTERM;
```

The READER and JTRA service jobs are to wait if necessary for the next EJ request while remaining in EX state.

□

**Remarks:**

The NTERM option for the SJREADER service job, costs 4K of memory while waiting for EJ requests. However, EJ requests will be processed more quickly.

The NTERM option, for the JTRA service job, costs 48K of memory while waiting for jobs to be translated. However, jobs will be translated as soon as they are introduced, regardless of the system load.



### 5.4.15 The JOBCLASS and JBCLASS2 Statements

#### Purpose

These statements specify the job management system default values for the applicable mono-letter (A to Z, JOBCLASS statement) or double-letter (AA to ZZ, JBCLASS2 statement) job classes. System default values include:

- execution and scheduling priorities, and the limits of these priorities
- the multiprogramming level
- the state at system start-up
- connection of a class to a job class group
- default limits of steps for CPU times, elapsed times and SYSOUT number of lines.

It is possible to override some default values attached to a job class by using:

- the PRIORITY parameter in the JCL statement JOB,
- the XPRTY,ELAPTIME, CPTIME or LINES parameters in the JCL statement STEP or in the GCL command EXEC\_PG.

The default values attached to each JOBCLASS are usually changed using the MODIFY\_LOAD (MDLD), TERMINATE\_LOAD (TLD) and START\_LOAD (SLD) main operator commands. However, the JOBCLASS and JBCLASS2 configuration statements may be used to prevent a main operator from modifying the default values (parameters: NSC, NMXPRTY, NMPRIO and NMLOAD).

The connection of a class to a job class group may be changed dynamically using the CONNECT\_LOAD (CNLD) and DISCONNECT\_LOAD (DISLD) main operator commands.

#### Format

```
JOBCLASS job-class [, XPRTY=n] [, XPRLIM={ 0 |n}]
[, PRIORITY=n] [, PRLIM={ 0 |n}]
[, MAXLOAD=nnn] [, { STARTED |NSTARTED}]
[, NSC] [, NMXPRTY] [, NMPRIO] [, NMLOAD]
[, JCG=job-class-group] [, ELAPTIME=nnnn]
[, CPTIME=nnnnnnn] [, LINES=nnnnnnn];
```

```
JBCLASS2 position, JOBCLASS=job-class [, XPRTY=n]
[, XPRLIM={ 0 |n}] [, PRIORITY=n] [, PRLIM={ 0 |n}]
[, MAXLOAD=nnn] [, { STARTED |NSTARTED}]
[, NSC] [, NMXPRTY] [, NMPRIO] [, NMLOAD]
[, JCG=job-class-group] [, ELAPTIME=nnnn]
[, CPTIME=nnnnnnn] [, LINES=nnnnnnn];
```



There may be several JOBCLASS and JBCLASS2 statements, but only one can be attributed to each job class. The JOBCLASS statement can be used for mono-letter job classes only (A to Z). The JBCLASS2 statement can be used for double-letter job classes only (AA to ZZ).

The job class concerned (as relevant to the CLASS parameter of the JCL statement JOB) is defined by the first positional parameter in the JOBCLASS statement and by the JOBCLASS parameter in the JBCLASS2 statement.

The first positional parameter ('position') in JBCLASS2 is a number ranging from 1 to 220, indicating the position of the statement occurrence. The scheduler can manage up to 256 classes, including 26 mono-letter job classes. 220 double-letter job classes may be defined in the configuration statement file. 10 more job classes may be defined by the system at configuration for service jobs.

If there is no JOBCLASS statement for any given mono-letter job class, the default value applied by the system is as shown in Table 5-2, *CONFIG Statement Default Values*, at the end of this chapter. Except for some service job classes, if there is no JBCLASS2 statement for a double-letter job class, the class is not defined by the system. It must be created dynamically by a main operator using the job management command START\_LOAD (SLD).

### Parameters

position	(parameter specific to the JBCLASS2 statement) - number ranging from 1 to 220 indicating the position of the statement occurrence.
job-class	(parameter specific to the JOBCLASS statement) - an alphabetic character ranging from A to Z, specifying the mono-letter class to which the statement applies.
JOBCLASS	(parameter specific to the JBCLASS2 statement) - two alphabetic characters ranging from AA to ZZ, specifying the double-letter class to which the statement applies. Note: Job classes RA and RB are reserved for service jobs. They cannot be re-configured.



---

XPRTY	a decimal digit ranging from 0 to 9, defining the default execution/dispatching priority for the job class. The parameter may be overridden by the XPRTY keyword of the JCL statement STEP or the GCL command EXEC_PG; 0 is high priority and 9 is low priority. The default value is that given in Table 5-1 for mono-letter job classes (JOBCLASS statement) and is 9 for double-letter job classes (JBCLASS2 statement).
XPRLIM	<p>a decimal digit ranging from 0 through 9, defining the highest dispatching priority allowed for the class; the default value is 0.</p> <p>Neither the XPRTY value of the JCL statement STEP, nor the MDJ command from an IOF user, may exceed this limit.</p> <p>The value supplied for XPRTY may under no circumstances exceed that defined for XPRLIM.</p> <p>Note that the main operator may override the XPRLIM limits with the MDJ and MDLD commands.</p>
PRIORITY	a decimal digit ranging from 0 through 7, defining the default scheduling priority for the job class; this parameter may be overridden by the PRIORITY keyword in the JCL statement JOB. 0 is high priority and 7 is low priority. The default value is that given in Table 5-1 for mono-letter job classes (JOBCLASS statement) and is 7 for double-letter job classes (JBCLASS2 statement).
PRLIM	<p>a decimal digit ranging from 0 to through 7, defining the limit of scheduling priority to be allowed for the class; the default value is 0. Neither the PRIORITY value of the JCL statement JOB, nor the MDJ command of an IOF user, may exceed this limit. This limit may under no circumstances be higher than that defined for PRIORITY.</p> <p>Note that the main operator may override the PRLIM limits with the MDJ and MDLD commands.</p>



MAXLOAD	a 3-digit decimal ranging from 0 through 150, defining the multiprogramming level within the job class; i.e., the maximum number of jobs in the class which may be in execution simultaneously. See the MULTLEV statement. The default value is that given in Table 5-1 for mono-letter job classes (JOBCLASS statement) and is 0 for double-letter job classes (JBCLASS2 statement).
STARTED / NSTARTED	indicates whether the class is to be STARTED, or not started (NSTARTED), respectively. If both NSTARTED and NSC (below) are specified at the same time, the class becomes unavailable. STARTED is the default value.
NSC	when specified, the main operator is prevented from using the SLD and TLD commands for the corresponding class. When not specified, the operator can use these commands for the corresponding job class, so that any of the GCL facilities may be used to manage the class and dynamically change the "started" or "not started" status of that class. If both NSTARTED and NSC are specified at the same time, the class is made unavailable.
NMXPRTY	when specified, the main operator is prevented from using the MDLD command to modify the dispatching priority of the class. When not specified, the operator may use the MDLD command on the job class, to modify its default dispatching priority.
NMPRIO	when specified, the main operator is prevented from using the MDLD command to modify the scheduling priority of the class. When not specified, the operator may use the MDLD command on the job class, to modify its default scheduling priority.
NMLOAD	when specified, the main operator is prevented from using the MDLD command to modify the multiprogramming level for the class. When not specified, the operator may use the MDLD command on the job class to modify its multiprogramming level.



---

JCG	<p>one or two alphabetic characters ranging from A to ZZ, specifying the job class group to which the defined class belongs. The class to which the JBCLASS2 statement applies is then a 'dependent' job class in a job class group in which the 'parent' job class is that given in the JCG parameter. The 'parent' job class must be defined by a JOBCLASS or JBCLASS2 statement.</p> <p>A job class cannot be both 'parent' and 'dependent'. The connection is not made if JCG specifies a class that is already 'dependent' in another job class group. The default value is none.</p> <p><b>IMPORTANT:</b> Some service job classes cannot be attached to a job class group.</p>
ELAPTIME	<p>a 4-digit decimal number that sets the default maximum time during which job steps, in the defined job class, can execute. The unit is a minute. It is possible to override this default value, at step execution time, by using the ELAPTIME parameter in the JCL statement STEP, or in the GCL command EXEC_PG. The default value (9999) means that there is no default limit on the elapsed time for step execution. No limit can be set for service job classes (R-Z AND RA-ZZ).</p> <p><b>IMPORTANT:</b> The ELAPTIME value specified at step execution overrides the initially set elapsed time, but cannot specify 'no limit'. If a job class has an elapsed time limit, no step in this class can execute without a limit.</p>
CPTIME	<p>a 7-digit decimal number, indicating the default maximum permitted CPU time for job steps running in the defined job class. The unit is a thousandth of a minute. It is possible to override this default value, at step execution time, by using the CPTIME parameter in the JCL statement STEP, or in the GCL command EXEC_PG. The default value (9999999) means that there is no default limit on CPU use. No limit can be set for service job classes (R-Z AND RA-ZZ).</p> <p><b>IMPORTANT:</b> The CPTIME value specified at step execution overrides the initially set CPU time, but cannot specify 'no limit'. If a job class has a CPU time limit, no step in this class can execute without a limit.</p>

---



**LINES** a 8-digit decimal number, indicating the default maximum number of print records that may be written in SYSOUT file for job steps running in the defined job class. It is possible to override this default value, at step execution time, by using the LINES parameter in the JCL statement STEP, or in the GCL command EXEC\_PG. The default value (99999999) means that there is no default limit on number of print records. No limit can be set for service job classes (R-Z and RA-ZZ).

**IMPORTANT:** The LINES value specified at step execution overrides the initially set LINES, but cannot specify 'no limit'. If a job class has a LINES limit, no step in this class can execute without a limit.

**EXAMPLES (User Job Classes):**

```
JOBCLASS P, XPRTY=9, PRIORITY=7, MAXLOAD=3;
```

This means that job class P has a dispatching priority of 9, a scheduling priority of 7, a multiprogramming level of 3 and is STARTED, by default. The operator may use TLD, SLD, and MDLD commands on this class.

```
JOBCLASS H, XPRTY=9, PRIORITY=7, MAXLOAD=0;
```

This means that, although job class H is started by default, since the multiprogramming level is 0, no class H jobs will be started unless the operator has issued either FJ or MDLD commands.

```
JOBCLASS2 JA, XPRTY=8, XPRLIM=7, NMXPRTY, JCG=A, ELAPTIME=5;
```

This means that job class JA has a default dispatching priority of 8 that cannot be changed through the MDLD command. Steps belonging to jobs launched in this class cannot have a dispatching priority greater than 7 and, unless otherwise specified, cannot run for more than 5 minutes. JOBCLASS JA is connected to job class group A. The other characteristics of job class JA are the default characteristics of double-letter job classes, as follows:

```
PRIORITY=7  
PRLIM=0  
MAXLOAD=0  
STARTED  
no limit on CPU use for step execution.
```

□



### Remarks:

Classes A to PZ are traditionally dedicated to user batch jobs. For these classes, the default values are selected to reflect the requirements of the site, but may require changing in the following typical situations:

- Several teams on the site require the same priorities for work to run in a multiprogramming level other than 1 (2 for instance) within a class. The following statements may be used:
 

```
JOBCLASS K, XPRTY=9, PRIORITY=7, MAXLOAD=2, STARTED;
JOBCLASS L, XPRTY=9, PRIORITY=7, MAXLOAD=2, STARTED;
```
- There are two types of job, night jobs and day jobs. Day jobs are started as soon as possible, but night jobs are started on operator decision, once all-day jobs have been completed. The statement for night jobs could be:

```
JOBCLASS N, XPRTY=9, PRIORITY=7, MAXLOAD=5, NSTARTED;
```

### Service Job Classes

Classes Q to QZ are generally reserved for IOF jobs.

Classes R to Z and RA to ZZ are not directly available for user jobs. These classes are used by the system to spawn service jobs as follows:

R	Input readers
RA	Hardware TD_MVH job
RB	Mirrored disk REFILL jobs
S	TNS or FEPS jobs (telecommunications monitoring, QMON for MCS), RAEH, IOSER, ARS, SECCL
T	File transfer jobs (UFT, FTF, and SERVER2)
U	DJP operators' IOF jobs and RBF
V	Trace TRCCL job
W	Output writers
X	JCL translator JTRA and High Availability jobs (CMSR, RECOV, JRU)
Y	GTP, VCAM, JPPC
Z	Main operator's IOF jobs, LAEH.





For example:

```
JOBCLASS Q, MAXLOAD=200, STARTED;  
JBCLASS2 175, QA, PRIORITY=0, XPRTY=1, MAXLOAD=20, NSTARTED,  
JCG=Q;
```

In the above example: a maximum of 20 IOF users is allowed in job class QA, provided that no more than 200 users are connected to the whole Q job class group. Priorities must be specified if you want them to be similar to those of class Q. The main operator must issue an 'SLD QA' command before users who work in class QA may log on. If the main operator issues a 'TLD QA' command, further QA IOF users are prevented from logging on.

**Remarks:**

Q Job Class	The NSTARTED, NSC, and NMLOAD parameters enable the system administrator to monitor the operator's management of IOF users. The MAXLOAD parameter specifies the maximum possible number of IOF users. You should insure that the MAXLOAD parameter value is correctly related to the corresponding value of the positional parameter in the MULTLEV statement.
R Job Class	It is always started.
RB Job Class	It is always defined. Its characteristics cannot be changed.
S Job Class	It is always started. The XPRTY and XPRILIM parameters are always set to 0.
V Job Class	It is always started. The NSC parameter is always set.
W Job Class	It is always started.
X Job Class	It is always started. The NSC parameter is always set. MAXLOAD must be set to a value greater than 3.
Y Job Class	It is always started. The NSC parameter is always set. The XPRTY parameter must be set to a value lower than 3; the default value is 2. XPRILIM is always set to 0. MAXLOAD must be set to a value greater than 1.
Z Job Class	It is always started. MAXLOAD must be set to a value greater than 1; the default value is 2. This parameter value may exceed 2 only if the relevant option has been purchased. XPRTY and XPRILIM are always set to 0.



The MAXLOAD parameter for Class Q plus the MAXLOAD parameter for Class Z may only exceed 40 when a large IOF configuration is installed.

**Table 5-1. Default Values for Mono-Letter Job Classes**

DEFAULT VALUES FOR CLASSES										
MAXLOAD	PRIORITY	XPRTY	STARTED	SC	MXPRTY	MPRIO	MLOAD	PRLIM	XPR LIM	CLASS
1	7	9	Y	Y	Y	Y	Y	0	0	A
1	7	9	Y	Y	Y	Y	Y	0	0	B
1	7	9	Y	Y	Y	Y	Y	0	0	C
1	1	5	Y	Y	Y	Y	Y	0	0	D
1	2	4	Y	Y	Y	Y	Y	0	0	E
1	3	7	Y	Y	Y	Y	Y	0	0	F
1	4	9	Y	Y	Y	Y	Y	0	0	G
1	6	0	Y	Y	Y	Y	Y	0	0	H
1	7	9	Y	Y	Y	Y	Y	0	0	I
1	6	0	Y	Y	Y	Y	Y	0	0	J
1	7	9	Y	Y	Y	Y	Y	0	0	K
1	7	9	Y	Y	Y	Y	Y	0	0	L
1	7	9	Y	Y	Y	Y	Y	0	0	M
1	7	9	Y	Y	Y	Y	Y	0	0	N
1	7	9	Y	Y	Y	Y	Y	0	0	O
40	7	9	Y	Y	Y	Y	Y	0	0	P
100	7	9	Y	Y	Y	Y	Y	0	0	Q
6	0	1	Y	N	Y	N	Y	0	0	R
12	0	0	Y	N	N	N	N	0	0	S
6	7	2	Y	Y	N	N	Y	0	0	T
6	0	1	Y	Y	N	N	Y	0	0	U
1	7	9	Y	Y	N	N	Y	0	0	V
8	0	1	Y	N	Y	N	Y	0	0	W
4	0	0	Y	Y	N	N	Y	0	0	X
6	7	2	Y	Y	N	N	Y	0	0	Y
2	0	0	Y	N	N	N	Y	0	0	Z



## 5.4.16 The JORSIZE Statement

### Purpose

Specifies the maximum number of lines that will be printed for the JOR, by defining a limit for the number of lines that may ever be written to the JOR for any single user step. This number can be modified only by another run of CONFIG.

### Format

```
JORSIZE { 500 |nnnn} ;
```

### Parameters

nnnnn is a decimal value of up to 5 digits, ranging from 1 through 65535, specifying a maximum count for lines written to the JOR by any single user step using SEND and REPORT JCL statements, the ACCEPT/DISPLAY COBOL feature, or H\_PUTJOR GPL primitives. No single user may exceed this maximum line count in his JOR. The default value is 500 lines.

### EXAMPLE:

```
JORSIZE 50;
```

The number of such user-written lines per job step is to be limited to 50.

□

### Remarks:

Note that information written by the system to the JOR is not subject to this statement; the system is never subjected to a limit on the amount of information it may write.

If user-written information exceeds the limit given in the JORSIZE statement, excess lines (from the corresponding step) are not written to the JOR.



### 5.4.17 The JPPCDIM Statement

#### Purpose

To select the dimension of the JPPC (of the five existing variants).

JPPC is a system service job that allocates memory segments to TDS applications that use XCP2 and/or IQS.

There are five variants of JPPC, each of which contains a certain number of processes (therefore, of segments). If the adequate variant is selected, JPPC does not use more control memory than is strictly necessary.

JPPCDIM determines which of the five variants of JPPC is selected.

#### Format

```
JPPCDIM {0|n};
```

#### Parameters

0	(Default value.) Specifies that the JPPC service job is not required.
n	A decimal digit ranging from 1 to 5, that identifies the variant of JPPC whose capacity meets the requirements of all the TDS applications using XCP2 and/or IQS.



**Remarks:**

1. The five variants of JPPC and their occupation of the control memory are:

LM	Number of Processes	Number of Segments	Number of XCP2 Conversations*	Memory Occupation
H_JPPC1	5	1150	350	136K
H_JPPC2	10	2300	700	
H_JPPC3	26	5980	2400	464K
H_JPPC4	35	8050	3500	
H_JPPC5	50	11500	5000	860K

\* These are approximate numbers, given for information only. The exact number of conversations supported by JPPC depends on the XCP2 programmatic interface.

2. For XCP2, each TDS session reserves a number of segments. This number depends on the value of the MAXTX parameter of the XCP2WKS statement used by NETGEN. TDS aborts if it does not obtain this number of segments.  
The number of segments reserved is at least equal to  $3*MAXTX+4$ . More segments are reserved when necessary (for example, if a transaction has more than one conversation, or is using the PPCPI interface).
3. For each IQS transaction, the user should reserve between 8 and 16 segments. If the initial reservation is insufficient, a warning is issued and IQS goes into degraded mode, using GMEM to stock data in a UFAS file.
4. At the end of each TDS session, a PP03 message is sent both to the submitter and to the TDS JOR. This message contains, for XCP2 and/or IQS the numbers of JPPC segments used to the maximum. The sum total of these maximum values gives the limit of JPPC segments used.
5. The GCL commands DS or DJ enable you to visualize which of the JPPCs are being processed, and the command DDIM displays the amount of memory occupied by the JPPC.



### 5.4.18 The LSYS Statement

#### Purpose

Specifies a local system name. It should be noted that the parameter LSYS in the NG Network Generation must use the same value.

#### Format

```
LSYS NAME=xxxx ;
```

#### Parameters

NAME introduces a 4-character alphanumeric name to be assigned to the local system on which CONFIG is being performed. If this identification parameter is missing (there is no LSYS statement), the default name DPS7 is assigned.



### 5.4.19 The MAXCAT Statement

#### Purpose

Sets a limit on the maximum number of catalogs that may be known to the system at any one time.

#### Format

```
MAXCAT { 100 |nnn} ;
```

#### Parameters

nnn is a decimal value of up to 3 digits, not greater than 128, specifying the maximum number of catalogs that may be known concurrently in the system. The default value is 100.

#### Remarks:

The system forces a minimum of 64 catalogs to avoid the risk of return code SYSOV when a catalog is attached. If you specify a value lower than 64, the message:

```
MAXCAT VALUE IS LESS THAN 64. 64 IS ASSUMED
```

is returned in the configuration report.



## 5.4.20 The MAXDEF Statement

### Purpose

Limits the maximum number of DEFINES that may be active concurrently on the system.

### Format

```
MAXDEF { 512 |nnnn};
```

### Parameters

nnnn is a decimal value of up to 4 digits, ranging from 100 through 2048, to specify the maximum number of DEFINES that may be active concurrently on the system. Certain MIs allow a value up to 6300. The default value is 512.

### Remarks:

The SYS.KNODET file size (as specified through TAILOR) sets a maximum number of DEFINES (NDF), according to the values specified for the MAXFILE and FILSHARE statements. For further details, refer to the description of the FILSHARE statement in this chapter.





### 5.4.21 The MAXFILE Statement

#### Purpose

Specifies the maximum number of files that may be active on the system at any one time. An active file satisfies one of the following conditions:

- it is opened
- it is assigned but not opened
- it is not assigned but passed
- it is an attached user catalog.

System files such as SITE.CATALOG, SYS.IN, SYS.OUT and SYS.URCINIT are not counted in this number of active files. This limit may be modified only by a subsequent run of CONFIG.

#### Format

```
MAXFILE { 512 |nnnn} ;
```

#### Parameters

nnnn is a decimal value of up to 4 digits, ranging from 100 through 2048, specifying the maximum number of active files. Certain MIs allow a value up to 6300. The default value is 512.

#### EXAMPLE:

```
MAXFILE 1500;
```

There may be no more than 1500 files active on the system at any one time.



#### Remarks:

This statement should be used when the default value is unsuitable for the limitations of the site, and the value should be set as low as possible compatible with site needs.

If a new value is defined for MAXFILE greater than 3200 when the old value was less than 3200, or vice-versa, a RESTART(CLEAN) is mandatory.



## 5.4.22 The MAXIODEV Statement

### Purpose

Specifies the maximum number of input/output devices to be allowed for any one station.

### Format

```
MAXIODEV {42 | 9 } ;
```

### Parameters

MAXIODEV may be assigned with only one of two possible values, 9 or 42. Any other value results in an error message; the default value is 9.

### Remarks:

The GCL command MDST writes the input/output devices in the Input Output Device Table (IODT).

The IODT tables are kept in a VMM file, with a block length of 4K if the value 9 is used for MAXIODEV and 8K if the value 42 is used.

SYSADMIN determines whether more than 9 devices are required at the station. In most cases, the value 9 is sufficient.



### 5.4.23 The MAXJOB Statement

#### Purpose

Specifies the maximum number of jobs that may be known (introduced, scheduled, executing, or waiting for output) to the system at any one time. Both user and service jobs are included in this number, which cannot be modified except by a subsequent run of CONFIG.

#### Format

```
MAXJOB {500|nnnn};
```

#### Parameters

nnnn is a decimal value of up to 4 digits, ranging from 10 through 9999, defining the maximum number of jobs that may be known to the system. The default value is 500.

#### EXAMPLE:

```
MAXJOB 600;
```

Not more than 600 jobs are to be allowed in the system at any one time. If the limit of 600 jobs is reached, a new job may be introduced only when an existing job is completed.



#### Remarks:

CONFIG automatically adds the maximum number of started jobs (the value of MULTLEV) to the value supplied.

The combined values defined for MAXJOB and MULTLEV may under no circumstances exceed 9999; MULTLEV is explained later in this chapter.

The active limit on the value defined for MAXJOB depends on the size of the SYS.PVMF[i] files.

The following formula allows SYS.PVMF[i] file size requirements (in Megabytes) to be evaluated, based on the values specified for the three CONFIG statements MAXJOB, MULTLEV and ACTSIZE.

$$\text{PVMFSIZE(MB)} = 5 + 14/1024 * \text{multlev} + 16/1024 * \text{maxjob}$$



#### 5.4.24 The MAXTAPE Statement

##### Purpose

This statement is used to set the maximum number of tape cartridges that may be known (either assigned or passed) to the system at any one time.

Each volume serial number, that is each cartridge, counts as one "tape". This number cannot be modified except by a subsequent run of CONFIG.

##### Format

```
MAXTAPE { 64 |nn} ;
```

##### Parameters

nn is a decimal value of up to 2 digits, ranging from 1 through 64, to define the maximum number of tape cartridge VSNs that may be known to the system at any one time. The default value is 64.

##### EXAMPLE:

```
MAXTAPE 12;
```

Limits to 12 the number of tape cartridges that may be known at any one time to the system.

□

##### Remarks:

The number defined by this statement allows the size of the relevant segment to be tailored. Each tape cartridge carries an overhead of about 40 bytes of memory. When using this statement, therefore, make sure that the default value is not too high.



### 5.4.25 The MAXTASK Statement

#### Purpose

Specifies the maximum number of tasks (for user and service job processes) which may be executed simultaneously. This limit may be modified only by a subsequent run of CONFIG.

#### Format

```
MAXTASK { 150 |nnnn} ;
```

#### Parameters

nnnn is a decimal value of up to 4 digits, ranging from 1 through 3900, to define the maximum possible number of tasks (for user and service job processes) which may be executed simultaneously. The default value is 150.

#### EXAMPLE:

```
MAXTASK 800;
```

Up to 800 tasks may run simultaneously.

□

#### Remarks:

The value given with this statement should be related to that of the MULTLEV statement; if MULTLEV defines that no more than n jobs may be "started", a value of n should therefore be sufficient for the MAXTASK statement in a monoprocessing environment. The default value of 150 is consistent with the corresponding MULTLEV default value of 90 and allows for multiprocessing.

As the MAXTASK value decreases, so does the size of resident segments.



---

Each TDS application accounts for several tasks, this number being set at TDS generation time with the parameter SIMULTANEITY. Each user step normally counts for 1 task.

COBOL programs using the MCS (Message Control System) facility may have more than one process. The number of processes is specified in the TASK statements of the Static Linker utility.

**NOTE:**

There is no direct relation between the number of Compile Units and the number of processes in the resulting Load Module, since the TASK statement of the Static Linker utility must be used to create multi-task steps.



## 5.4.26 The MULTLEV Statement

### Purpose

Controls the multiprogramming level by specifying the maximum number of jobs that may be "started" concurrently. To achieve this, MULTLEV defines the maximum number of started jobs (including service jobs); this value may only be changed by a subsequent run of CONFIG.

### Format

```
MULTLEV nnn ;
```

### Parameters

nnn is a decimal value of up to 3 digits, specifying the number of jobs that may be started concurrently, in the range 2 through 254, to define the maximum number of concurrently "started" jobs (service jobs + user jobs). The default value is 90.

### EXAMPLE:

```
MULTLEV 100;
```

A total of 100 service and/or user jobs may be started.



### Remarks:

One user job may be considered as being equivalent to either:

- a TDS application
- an MCS application.

On the other hand, an IOF job is one service job in Class Q.

Each increment of the positional parameter costs an extra 47 bytes in resident memory.

The multiprogramming level of Class Q limits the number of interactive users (IOF users in job class Q), that is, by the MAXLOAD parameter in the JOBCLASS statement for job class Q.

The value specified for the MAXLOAD keyword in the JOBCLASS statement must be less than or equal to the value supplied for MULTLEV.



### 5.4.27 The OWCLASS Statement

#### Purpose

Specifies the default priority and media for each of the output writer classes. The default priority and media may be modified for a given output class, either:

- by SYSADMIN using MNCAT commands such as CRS and MDS, which may be applied to any station other than MAIN (refer to the *GCOS 7 System Administrator's Manual*)
- by the operator using the GCL command MDC with the OC option (refer to the *GCOS 7 System Operator's Guide*).

These default priority and media values may be overridden for individual deliveries, either:

- by the user specifying the relevant parameters in the batch JCL statements OUTVAL, SYSOUT and WRITER
- or by the operator issuing the GCL command MDO or FO.

#### Format

```
OWCLASS output-class [,PRIORITY=n] [,MEDIA=media] statement
JOBCLASS et JBCLASS2;
```

#### Parameters

output-class	specifies the output class for which the default selection priority and media are to be modified.
PRIORITY	introduces a 1-digit value ranging from 1 through 6, specifying the default selection priority to be attached to the relevant output class. System default priorities are as shown in the remarks below, priority 6 being the lowest priority.
MEDIA	introduces a string of up to 6 alphanumeric characters, specifying the default output media for the class.





**LINES**

introduces a 7-digit value ranging from 1 to 9999999, specifying the system default value for the maximum number of lines contained in the output for a given class.

If this value is omitted, or if equal to 9999999, there is no limit to the number of lines.

If the value is less than 1, the value 9999999 is assumed.

Any job that creates output in the class concerned, and whose size is greater than the quota of lines indicated by this parameter, will be aborted with the return code ERLMOV.

Dumps are not limited to this quota of lines.

The quota of lines can be modified dynamically, during a GCOS session, using the command `MODIFY_CONFIGURATION`.

**EXAMPLES:**

```
OWCLASS A, PRIORITY=2;  
OWCLASS C, PRIORITY=2, MEDIA=I10001;  
OWCLASS A, LINES=100000; (Maximum size of class A output is  
100000 lines)
```

□

**Remarks:**

There may be several OWCLASS statements, each specifying an output class to be given a new default priority and/or media

If there is no OWCLASS statement for a given output class, the standard system defaults are assumed:

Output Class	Default Priority
A	1
B	2
C	3
D	4
E	5
F to Z	6



## 5.4.28 The OWDEVICE Statement

### Purpose

Specifies the output classes to be started as soon as the output writer is activated for a specific device (from the MAIN station only) without any indication of class in the SOW command. These default classes may be changed only by a subsequent run of CONFIG.

### Format

```
OWDEVICE position [, DVID=dvid] [, CLASS=output-class-list] ;
```

### Rules

1. More than one OWDEVICE statement may be used. The first positional parameter value of the statement determines the position of the output device concerned. If position  $n$  is used, positions 1, 2 ... ( $n - 1$ ) must also be used, in any order and before or after the reference to  $n$ .  
  
If two statements have the same position, they must refer to the same device, the value for the DVID parameter being the same. This ensures that the latest statement is taken into account.
2. Device external names must exist in the SRST; otherwise, an error is detected and all the following statements (in the position sense) become ineffective.
3. If there is no OWDEVICE statement for a particular device in the SRST, by default all the output classes are started when the output writer is started on that device. The devices taken into account are those in the SRST suitable for the output writer (PR devices).
4. A device may not be named in more than one OWDEVICE statement with a difference of position.



### Parameters

- position is a decimal value of up to 2 digits, ranging from 1 through 15, which enables the position of up to 15 output devices to be set. Leading zeros are not allowed.
- This enables a position to be specified when there is more than one OWDEVICE statement (see 1. above).
- DVID introduces the device identification of a particular device, such as PR01. When an SOW operator command is issued, the output classes specified by the CLASS parameter are automatically started on this device.
- This name must represent a device known in the SRST and must represent a device suitable for the output writer: PRxx.
- CLASS introduces a string of up to 26 alphabetic characters, from A through Z, representing the classes to be started when the output writer is started on the device declared.

### EXAMPLES:

Suppose that the site knows the following devices: PR01, PR02.

1. No OWDEVICE statement is issued:

When the operator issues the command:	The following output classes are started on that device:
---------------------------------------	--

SOW PR01	A to Z
SOW PR02	A to Z

2. The following OWDEVICE statements have been issued:

```
OWDEVICE 1, DVID=PR01, CLASS=ABCDEF;
```

```
OWDEVICE 2, DVID=PR02, CLASS=GHIJKLMNOPQRSTUVWXYZ;
```

When the operator issues the command:	The following output classes are started on that device:
---------------------------------------	--

SOW PR01	A to F
SOW PR02	G to Z



3. The following OWDEVICE statements have been issued:

```
OWDEVICE 1, DVID=PR01, CLASS=ABCDEFGHIJKL;  
OWDEVICE 2, DVID=PR02, CLASS=IJKLMNOP;
```

When the operator issues the command:      The following output classes  
are started on that device:

```
SOW PR01                    A to L  
SOW PR02                    I to P
```

4. The following OWDEVICE statements have been issued:

```
OWDEVICE 1, DVID=PR01, CLASS=ABCD;  
OWDEVICE 1, DVID=PR01, CLASS=EFGH;
```

The second statement overrides the first; when the operator issues an SOW PR01 command, the output classes started on that device are E, F, G, and H.

5. An error would occur if the following OWDEVICE statements were issued:

```
OWDEVICE 1, DVID=PR01, CLASS=ABCD;  
OWDEVICE 1, DVID=PR02, CLASS=GFGH;
```

because two statements of same position attempt to refer to different devices.

6. An error would occur if the following OWDEVICE statements were issued:

```
OWDEVICE 1, DVID=PR01, CLASS=ABCD;  
OWDEVICE 2, DVID=PR01, CLASS=EFGH;
```

because a **single** device is referred to in two statements with **different** positions.

□



**Remarks:**

Suppose that the installation has only one printer, PR01, and that output files fall into one of two possible categories:

1. Output files belonging to classes A, B, C, D, E, F
2. Output files belonging to classes G to Z.

Output in category (i) is urgent and generally not bulky; listings are wanted as soon as possible. On the other hand, output files in category (ii) are lengthier and not urgent. It may be useful to issue the following statement:

```
OWDEVICE 1, DVID=PR01, CLASS=ABCDEF;
```

So that, when the operator issues the command SOW PR01, only urgent output will be printed, not output in classes G through Z.

When the time comes to print the remaining output, the operator issues the following command:

```
SOW PR01 G-Z
```

Similarly, if the site has two printers using different paper, and if there is a convention at the site to use output classes according to paper type:

Classes A to F use paper n1  
Classes G to Z use paper n2,

it may be convenient to use:

```
OWDEVICE 1, DVID=PR01, CLASS=ABCDEF;  
WDEVICE 2, DVID=PR02, CLASS=GHIJKLMNOPQRSTUVWXYZ;
```

with the result that, when the operator issues the commands SOW PR01 and SOW PR02, it will not be necessary to change the paper.



## 5.4.29 The OWDFLT Statement

### Purpose

Specifies the system defaults for the output writer parameters CLASS, DEVCLASS, TAPE and xBANxxx in JCL statements.

- These defaults are taken into account when processing the CLASS and DEVCLASS parameters of the OUTVAL/SYSOUT/WRITER batch JCL statements.
- The TAPE parameter is provided for the purposes of compatibility with much earlier releases, and modifies the internal format of SYSOUT tapes. This default may be changed only by a subsequent run of CONFIG.
- The banner parameters include FBANNER, EBANNER and BANCHAR.

### Format

```
OWDFLT CLASS={class| C }
      [, DEVCLASS={devclass| PR/H132}]
      [, TAPE={SYSOUT|NSYSOUT}] [, FBANNER={0|1|2}]
      [, EBANNER={0|1|2}] [, BANCHAR={'xx'|'OZ'}]
      [, PAPERBRK=[Y|N]] [, CHARBRK=['xy'|'OZ']]
      [, BIGCHAR=[Y|N]] ;
```

### Parameters

CLASS	introduces the default value for the CLASS parameter of the JCL statements OUTVAL, SYSOUT, and WRITER. If the CLASS parameter is not supplied, the system default value is C.
DEVCLASS	introduces the default device class with the attributes of its associated printer. If the DEVCLASS parameter: <ul style="list-style-type: none"> <li>– is not specified, the system default value depends on the type of printer installed; for example, if the printer is a PR54, the default value of DEVCLASS is PR/H132.</li> <li>– is wrong, the statement is rejected.</li> </ul>

It is always advisable to specify the DEVCLASS parameter explicitly, especially if there are different printer types on the same site.



TAPE	<p>establishes whether or not the output writer is to employ user-defined parameters. If the user has written a tape without preallocating it specifically to use the SYSOUT mechanism, parameters for the Output Writer must be taken from the user program.</p> <p>The value SYSOUT allows the Output Writer to override user instructions (as to block size, for instance). NSYSOUT prevents the Output Writer from doing this. This parameter is only provided for reasons of compatibility with a much earlier release.</p> <p>User programs reading SYSOUT tapes require use of the NSYSOUT value.</p>
FBANNER	<p>specifies the number of front banner pages between deliveries. The default value is 2; the other values possible are 0 or 1.</p>
EBANNER	<p>specifies the number of end banner pages between deliveries. The default value is 1; the other values possible are 0 or 2.</p>
BANCHAR	<p>defines the two graphic symbols, enclosed within single quotes, to be used for printing banners. The default value is 'OZ'.</p>
PAPERBRK	<p>specifies whether or not a special line, separating two consecutive outputs, is printed. The default value is N (no special line). If set to Y, a special line is printed immediately after the output banner.</p>
CHARBRK	<p>defines the two graphic symbols, enclosed in single quotes, of which the special separating line will be composed. The default value is 'OZ'. This parameter is meaningful only if PAPERBRK=Y.</p>
BIGCHAR	<p>specifies whether the big characters of the banner are composed of the corresponding small characters, or not.</p> <p>The default value is N; in this case the big characters of the banner are composed of the first character of the BANCHAR parameter.</p>

**EXAMPLES:**

```
OWDFLT;
```

The default class for deliveries is to be CLASS=C; the system default output device is a PR/Hxxx printer, with the belt/character set/paper combination: I10000. If the output writer has to create SYSOUT tapes, it may override user-supplied RECFORM and BLKSIZE parameters.

```
OWDFLT CLASS=D, DEVCLASS=PR/H160;
```

The default class for deliveries is CLASS=D, while the system default output device is a PR/H160.

```
OWDFLT FBANNER=0, EBANNER=0;
```

No banner will be printed between deliveries, unless the user specifies a BANNER parameter in OUTVAL, SYSOUT or WRITER in which case default values will be assumed for FBANNER and EBANNER.

```
OWDFLT PAPERBRK=Y CHARBRK='KX' BIGCHAR=Y;
```

A special line, composed of characters 'KX', will be printed in order to separate the two consecutive outputs. The big characters in the banner will be composed of the corresponding small characters.

□

**Remarks:**

This statement should be used when:

1. the installation has no PR/Hxxx printer.
2. the installation has more than one printer and the type PR/Hxxx is not to be the default output device.
3. the default standard belt is not suitable.
4. a complete application is working on SYSOUT tapes.





### 5.4.30 The PRLOG Statement

#### Purpose

Specifies the threshold value at which the operator is warned that the SYS.ERLOG file should be printed. PRLOG also specifies the associated printing options. This statement may be changed only by a subsequent run of CONFIG.

#### Format

```
PRLOG [THRESHLD={50 |m}] [, COMMAND='ocl-request'] ;
```

#### Parameters

**THRESHLD** introduces a decimal value of up to 2 digits, ranging from 1 through 99, to specify when the SYS.ERLOG file should be printed. The operator receives the message:

```
"EL01 SYS.ERLOG xx% FULL."
```

The default value is 50.

**COMMAND** introduces a string of a maximum of 105 characters in length, enclosed within single quotes, specifying a valid operator command to be activated whenever the threshold value is reached. If this parameter is not specified, by default the following message is sent to the operator:

```
"EL03 RUN PRLOG."
```



**EXAMPLE:**

```
PRLOG THRESHLD=90;
```



**Remarks:**

The threshold value may be modified depending on the frequency at which the file contents are printed, on the file size and on the rate at which it is filled.

The statement provides a convenient way to run a PRLOG job with the specific options:

```
PRLOG THRESHLD=65, COMMAND='EJR PRLOG';
```

If HSSLIB of the INPUTCTL statement is not specified, the way to run PRLOG job is:

```
PRLOG THRESHLD=65, COMMAND='EJR PRLOG,,SYS.LIB';
```

Private PRLOG job can be in a USER library and run as:

```
PRLOG THRESHLD=65, COMMAND='EJR PRLOG,,USER.LIB';
```



### 5.4.31 The PRLOGC Statement

#### Purpose

Specifies the threshold value at which the operator is warned that the SYS.LOGC file should be printed. PRLOGC also specifies the associated printing options. This statement may be changed only by a subsequent run of CONFIG.

#### Format

```
PRLOGC [THRESHLD={ 50 |m}] [, COMMAND='ocl-request'] ;
```

#### Parameters

THRESHLD	introduces a decimal value of up to 2 digits, ranging from 2 through 99, to specify when the SYS.LOGC file should be printed. The operator receives the message:  "SYS.LOGC xx% FULL."  The default value is 50.
COMMAND	introduces a string of maximum length 105 characters enclosed within single quotes, to specify a valid operator command for activation whenever the threshold value is reached. If this parameter is not specified, by default the following message is sent to the operator:  "RUN PRLOGC"



**EXAMPLE:**

```
PRLOGC THRESHLD=90;
```



**Remarks:**

The threshold value may be modified depending on the frequency at which the file contents are printed, on the file size and on the rate at which it is filled.

The statement provides a convenient way to run a PRLOGC job with the specific options:

```
PRLOGC THRESHLD=65, COMMAND='EJR PRLOGC';
```

If HSSLIB of the INPUTCTL statement is not specified, the way to run PRLOGC job is:

```
PRLOGC THRESHLD=65, COMMAND='EJR PRLOGC,,SYS.LIB';
```

Private PRLOG job can be in a USER library and run as:

```
PRLOGC THRESHLD=65, COMMAND='EJR PRLOGC,,USER.LIB';
```



### 5.4.32 The QUEUED Statement

#### Purpose

Governs the use of the Cache Mechanism for shared (but not coupled) queued linked files by activating or inhibiting this mechanism on Directory and BAM (Block Allocation Map) blocks of shared queued linked files (library organization).

#### Format

```
QUEUED { NCACHE | CACHE} ;
```

#### Parameters

CACHE	specifies that the Cache Mechanism be activated.
NCACHE	ensures that the Cache Mechanism is not activated, the default value.

Activation of the Cache Mechanism consumes memory but saves input-output activity.



### 5.4.33 The SADM OPT Statement

#### Purpose

Enables the selection of options specific to System Administration (for users working under the SYSADMIN project).

#### Format

```
SADMOPT      CHKJSADM={YES | NO}
              GCLKPROJ={YES | NO}
              GCLKSADM={YES | NO}
              GCSADMOP={YES | NO}
              GCL15BIN={YES | NO}
              MIRROR={YES | NO}
              SKBSUJOB={YES | NO};
```

#### Parameters

CHKJSADM	<p>Checks if a job is for system administration.</p> <p>YES: GCOS 7 will check that users running jobs for SYSADMIN are working under the SYSADMIN project.</p> <p>NO: default value.</p> <p>Note: If the project of the \$JOB statement is SYSADMIN, the submitter's project must also be SYSADMIN.</p>
GCLKPROJ	<p>YES: GCL procedures that contain the LOCK keyword in the PROC statement can be modified in maintain commands by any user working under the project.</p> <p>NO (default value): modification of these procedures is limited to the user compiling them.</p>



GCLKSADM	<p>YES: users of the SYSADMIN project will have the same rights in maintain commands on locked GCL procedures as the user who has compiled them.</p> <p>NO (default value): users working under the SYSADMIN project are not allowed to modify locked GCL procedures (LOCK option in PROC).</p>
GCSADMOP	<p>Checks SYSADMIN access to Main Operator commands through EXDIR.</p> <p>YES: users under the SYSADMIN project have the same rights as the Main Operator.</p> <p>NO (default value): users under the SYSADMIN project are normal IOF users.</p>
GCL15BIN	<p>YES: authorize to skip from 3 to 15 binaries libraries in the SEARCH path.</p> <p>NO: default value. (MWINLIB BIN clears up to 3 libraries).</p>
MIRROR	<p>Defines the Mirror Disks activity.</p> <p>YES: makes the Mirror Disks facility available on the system. Disk mirroring, and activities on mirrored disks are possible after a RESTORE, RESTART (COLD), provided that the SITE.MIRLOG file exists.</p> <p>NO (default value): the Mirror Disks facility will not be available on the system.</p> <p>For further details, refer to the <i>Mirror Disks User's Guide</i>.</p>
SKBSUJOB	<p>Specifies whether right blanks in values are to be skipped:</p> <p><b>YES:</b> right blanks in values are skipped when the field JOB_SKIP_BLANK is not specified in the JOB_DESCRIPTION parameter of CALL "SUBJOB".</p> <p><b>NO:</b> right blanks in values are kept when the field JOB_SKIP_BLANK is not specified in the JOB_DESCRIPTION parameter of CALL "SUBJOB".</p>



### 5.4.34 The SECOPT Statement

#### Purpose

Used in a network environment to select options specific to security. The version of SECU'ACCESS referred to is V2.

#### Format

```
SECOPT          SA7LOGON= {YES | NO}  
                SA7ADMIN= {YES | NO}  
                SA7NOCSL= {YES | NO}  
                NETSEC= {YES | NO}  
                CHKPW= {YES | NO}  
                ERASE= {YES | NO};
```

#### Parameters

SA7LOGON	YES: users' rights to access TDS/IOF applications are checked at connection time by SECU'ACCESS.  NO (default value): users' access rights are checked at connection time by GCOS 7.
SA7ADMIN	YES: user administration is performed by SECU'ACCESS.  NO (default value): user administration is performed by SYSADMIN through catalog management (MNCAT).
SA7NOCSL	YES: the rights of users to access the local system console are checked by GCOS 7.  NO (default value): the rights of users to log on to the local system console are checked by SECU'ACCESS. <b>Meaningful only when SA7LOGON=YES.</b>





NETSEC	<p>Network security.</p> <p>YES: connections always go through the front-end processor, which performs security checks using LACS (Logiciel d'Accès et de Contrôle de Sécurité) software, via the front-end processor. A special site (called BCLS) must be created for this purpose and, in the NETGEN directive, the path to this site must be declared as going through the front-end processor and having access to LACS.</p> <p>The front-end processor can be either the Datanet or the CNP 7.</p> <p>NO (default value): local connections are checked by GCOS 7.</p> <p>For more details, refer to Networks: Overview and Generation and the GCOS 7 System Administrator's Manual.</p>
CHKPW	<p>YES: in order to improve system security GCOS 7 verifies, on the acceptor site, the requestor's password and the application name for <b>all</b> connection requests to GCOS 7 applications (IOF, TDS, UFT, DJP, RPMOS...).</p> <p>GCOS 7 checks that the requestor's identity (USER, PROJECT, PASSWORD and BILLING - if a billing check is validated at site catalog level) is registered in the SITE.CATALOG file.</p> <p>GCOS 7 also checks that the requested application name is in the list of authorized applications in the SITE.CATALOG file, for the current USER and PROJECT.</p> <p>NO (default value) : on the acceptor site, GCOS 7 checks the requestor's password and the application name for connection requests to <b>certain</b> applications (such as IOF and TDS), but not for others (such as UFT and DJP).</p>
ERASE	<p>YES: after a file is deallocated, the entire data space that was allocated to the file is overwritten with zeros. Thus, data that was in the file prior to deallocation cannot be read. This applies to all files.</p> <p>NO (default value): the files are not physically erased after deallocation.</p>



### 5.4.35 The VCAMSCN Statement

#### Purpose

Tailors VCAM structures according to the maximum number of DSA, ISO, and GXTI session connections required.

GXTI (GCOS Extended Transport Interface) allows applications to access the transport layer directly, without using the session layer.

#### Format

```
VCAMSCN [MAXISO={ 0 |nnnn} ]  
        [MAXDSA={2500|nnnn} ]  
        [MAXTE={ 100|nnnn} ] ;
```

#### Parameters

MAXISO	maximum number of ISO connections.
MAXDSA	maximum number of DSA connections.
MAXTE	maximum number of GXTI connections.
nnnn	the sum of the 3 connection sessions (ISO, DSA, GXTI) must not be greater than 14000 <b>and</b> there must be at least one DSA session.



### 5.4.36 The VOLORG Statement

#### Purpose

On a site that supports both VBO and FBO disks, VOLORG defines the disk types that are authorized on that site.

#### Format

```
VOLORG {VBO|FBO|ALL};
```

#### Parameters

VBO	Only VBO disks are allowed on the site.
FBO	Only FBO disks are allowed on the site.
ALL	Both VBO and FBO disks are allowed on the site.

#### Remark:

This option is to be used:

- by the VOLPREP utility, to prohibit the creation of an unauthorized disk type
- to adjust GCL menus to the site configuration.



---

### 5.4.37 The XAS Statement

#### Purpose

For UFAS-EXTENDED files, XAS specifies the number of extended buffers.

#### Format

```
XAS [XBUF={8192|nnnnn}];
```

#### Parameters

nnnnn	5-digit decimal value, ranging from 1 to 32640, that specifies the maximum number of extended buffers. The value specified is rounded up to the nearest multiple of 256. The default value is 8192.
-------	---



## 5.5 Default Values of CONFIG Statement

Table 5-2. CONFIG Statement Default Values (1/2)

Command	Parameters	TA
ACCOUNT	POSIT 1	ALL
ACTSIZE	POSIT 1	200
BJSIMU	POSIT 1	Depends on MULTLEV
DEFORG	POSIT 1	VBO
DEVTRACE	POSIT 1	No Trace
DJPCTL	MAXTFSIT MINEVERY	2 15
DUALSYS	POSIT 1 MYSELF	FFFFFFFF FFFFFFFF
DUMPACT	COMMAND	NO
FILEOPT	COMPTS MNTWORK CTFIRST LASTDATE USESITE CTUNLOAD	PREVIOUS NO NO YES NO NO
FILSHARE	POSIT 1	50
GAC	LONGWAIT LOCKSIZE NBLOCKID	60 20 100
INPUTCL	KMEDIA JTRA SJREADER HSLLIB HOABORT	JCL NTERM NTERM SYS.HSLLIB NO
JOBCLASS JBCLASS2	XPRLIM PRLIM STARTED ELAPTIME CPTIME	0 0 YES 9999 9999999
JORSIZE	POSIT 1	500
JPPCDIM	POSIT 1	0
LSYS	NAME	XXXX
MAXCAT	POSIT 1	100
MAXCONV	POSIT 1	350
MAXDEV	POSIT 1	512
MAXFILE	POSIT 1	512
MAXIODEV	POSIT 1	9
MAXJOB	POSIT 1	500
MAXTAPE	POSIT 1	64
MAXTASK	POSIT 1	150
MULTLEV	POSIT 1	90



Table 5-2. CONFIG Statement Default Values (2/2)

Command	Parameters	TA
OWCLASS	POSIT 1=A POSIT 1=B POSIT 1=C POSIT 1=D POSIT 1=E POSIT 1=F-Z MEDIA	PRIORITY=1 PRIORITY=2 PRIORITY=3 PRIORITY=4 PRIORITY=5 PRIORITY=6 <b>I10000</b>
OWDEVICE	POSIT 1 DVID CLASS	(1,15) XXXX A-Z
OWDFLT	CLASS DEVCLASS TAPE BIGCHAR PAPERBRK CHARBRK FBANNER EBANNER BANCHAR	C PR/H132 SYSOUT N N 0Z 2 1 0Z
PRLOG	THRESHLD COMMAND	50 NO
PRLOGC	THRESHLD COMMAND	50 NO
QUEUED	POSIT 1	NCACHE
SADMOPT	CHKJSADM GCLBIN GCLKPROJ GCLKSADM GCSADMOP MIRROR SKSUBJOB	NO NO NO NO NO NO NO
SECOPT	SA7LOGON SA7ADMIN SA7NOCSL NETSEC CHKPW ERASE	NO NO NO NO NO NO
SITEOPT	POSIT1	00000000
VCAMSCN	MAXISO MAXDSA MAXTE	0 2500 100
VOLOG	POSIT 1	ALL
XAS	XBUF	8192



## 5.6 Output Listings

At the end of a CONFIG run, the following listings and messages are produced:

- Hardware device configuration listing
- Explicit configuration commands
- Memory parameters
- Scheduling Parameter listing
- Configuration Class listing
- Effective configuration commands
- Successful Completion message
- Unsuccessful Completion message(s).



### 5.6.1 Explicit Configuration Commands

Configuration commands supplied by the user are printed out under the heading:

```
*** EXPLICIT CONFIGURATION COMMANDS (PROVIDED BY USER) ***
```

Each statement is numbered; the numbers are referred to in any subsequent error message. Table 5-3 shows a typical example of the listing.

**Table 5-3. Typical Explicit Configuration Command Listing**

```
**** EXPLICIT CONFIGURATION COMMANDS (PROVIDED BY USER) ****  
0001 DUMPACT COMMAND = 'EJ DUMPACT:USER.JCLLIB';  
0002 JORSIZE 200;  
0003 DEVTRACE ABN, ATN, WARN, ALRM;  
0004 MAXTASK 400;  
0005 BJSIMU 100;  
0006 MULTLEV 140;  
0007 LSYS NAME = SST1;  
0008 DUALSYS AABCCDD;
```

### 5.6.2 Memory Parameters

Information pertinent to these parameters is printed under the heading "MEMORY PARAMETERS":

```
MEMORY PARAMETERS
```

```
IMPORTANT IF YOU HAVE CHANGED THE PARAMETER VALUE OF BKST PLM, SM,  
CKPT, UNITSZ AS COMPARED WITH THE SYSTEM CURRENT VALUES, THE NEW  
VALUE OF THESE PARAMETERS IS TAKEN INTO ACCOUNT ONLY WITH RESTORE  
+ RESTART (CLEAN).
```

```
ESTIMATED PERMANENT BKST1 NEEDS: 10 MEGA-BYTES.
```

```
ESTIMATED PERMANENT BKST2 NEEDS: 38 MEGA-BYTES.
```

```
ESTIMATED TEMPORARY BKST* NEEDS: 83 MEGA-BYTES.
```





### 5.6.3 Scheduling Parameter Listing

A summary of scheduling parameters is printed out under the heading SCHEDULING PARAMETERS. A typical printout is shown in Table 5-4. Information for this listing is extracted from the MAXJOB and MULTLEV statements.

**Table 5-4. Typical Scheduling Parameter Listing**

<b>SCHEDULING PARAMETERS</b>	
MAXIMUM NUMBER OF KNOWN JOBS	: 640
MAXIMUM NUMBER OF STARTED JOBS	: 140

### 5.6.4 Configuration Class Listing

A job class status listing is printed out, giving the following information:

CLASS	gives the name of the job class.
MAXLOAD	gives the number of multiprogramming levels within the job class.
PRIORITY	gives the scheduling priority attached to the job class.
XPRTY	gives the dispatching priority attached to the job class.
STARTED	"Y" means the job class is started; blank means the job class is not started.
SC	"Y" means that the SLD and TLD GCL commands are allowed in this job class. Blank means that these commands are not allowed; i.e., the NSC parameter has been used in the corresponding JOBCLASS statement.
MXPRTY	"Y" means that the operator may change the job class's dispatching priority with the MDLD GCL command. Blank means that the operator cannot change the class's priority; i.e., the NMXPRTY parameter has been used in the corresponding JOBCLASS statement.



MPRIO	"Y" means that the operator may change the job class's scheduling priority with the MDLD GCL command. Blank means that the operator cannot change the class's priority; i.e., the NMPRIO parameter has been used in the corresponding JOBCLASS statement.
MLOAD	"Y" means that the operator may change the job class's multiprogramming level with the MDLD GCL command. Blank means that the operator cannot modify the class's multiprogramming level; i.e., the NMLOAD parameter has been used in the corresponding JOBCLASS statement.
PRLIM	shows the limit of the scheduling priority allowed for the class; "0" is the default value.
XPRLIM	indicates the limit of the dispatching priority allowed for the class; "0" is the default value.

### 5.6.5 Effective Configuration Commands

Under the heading:

```
*****EFFECTIVE COMMANDS OF CONFIG-STEP IDENT : xx
```

is shown the entire set of commands effective for the configuration being a merge of explicit configuration commands as supplied by the user with all the default values (commands with default values assigned to their parameters). In practice, certain statements will be found in the listing which have not been included in the CONFIG statements detailed in this chapter; this is because these statements contain no user-'selectable' features and thus are of no concern. Below is an example listing of commands effective for a configuration (but excluding any 'non-selectable' features, for the sake of brevity and clarity).

```
***** EFFECTIVE COMMANDS OF C O N F I G - S T E P I D E N T : 22
```

```
ACCOUNT ALL;
```

```
JOBCLASS A,  XPRTY=9,PRIORITY= 7, XPRLIM=0, PRLIM=0,  MAXLOAD=1, STARTED;
JOBCLASS B,  XPRTY=9, PRIORITY=7, XPRLIM=0, PRLIM=0,  MAXLOAD=1, STARTED;
JOBCLASS C,  XPRTY=9, PRIORITY=7, XPRLIM=0, PRLIM=0,  MAXLOAD=1, STARTED;
JOBCLASS D,  XPRTY=5, PRIORITY=1, XPRLIM=0, PRLIM=0,  MAXLOAD=1, STARTED;
JOBCLASS E,  XPRTY=4, PRIORITY=2, XPRLIM=0, PRLIM=0,  MAXLOAD=1, STARTED;
JOBCLASS F,  XPRTY=7, PRIORITY=3, XPRLIM=0, PRLIM=0,  MAXLOAD=1, STARTED;
JOBCLASS G,  XPRTY=9, PRIORITY=4, XPRLIM=0, PRLIM=0,  MAXLOAD=1, STARTED;
JOBCLASS H,  XPRTY=0, PRIORITY=6, XPRLIM=0, PRLIM=0,  MAXLOAD=1, STARTED;
```



```
JOBCLASS I,  XPRTY=9, PRIORITY=7, XPRLIM=0, PRLIM=0,  MAXLOAD=1,  STARTED;
JOBCLASS J,  XPRTY=0, PRIORITY=6, XPRLIM=0, PRLIM=0,  MAXLOAD=1,  STARTED;
JOBCLASS K,  XPRTY=9, PRIORITY=7, XPRLIM=0, PRLIM=0,  MAXLOAD=1,  STARTED;
JOBCLASS L,  XPRTY=9, PRIORITY=7, XPRLIM=0, PRLIM=0,  MAXLOAD=1,  STARTED;
JOBCLASS M,  XPRTY=9, PRIORITY=7, XPRLIM=0, PRLIM=0,  MAXLOAD=1,  STARTED;
JOBCLASS N,  XPRTY=9, PRIORITY=7, XPRLIM=0, PRLIM=0,  MAXLOAD=1,  STARTED;
JOBCLASS O,  XPRTY=9, PRIORITY=7, XPRLIM=0, PRLIM=0,  MAXLOAD=1,  STARTED;
JOBCLASS P,  XPRTY=9, PRIORITY=7, XPRLIM=0, PRLIM=0,  MAXLOAD=40, STARTED;
JOBCLASS Q,  XPRTY=1, PRIORITY=0, XPRLIM=0, PRLIM=0,  MAXLOAD=100, STARTED, NMPRIO;
JOBCLASS R,  XPRTY=1, PRIORITY=0, XPRLIM=0, PRLIM=0,  MAXLOAD=6,  STARTED, NSC,
             NMXPRTY;
JOBCLASS S,  XPRTY=0, PRIORITY=0, XPRLIM=0, PRLIM=0,  MAXLOAD=12, STARTED, NSC,
             NMXPRTY, NMPRIO, NMLoad;
JOBCLASS T,  XPRTY=2, PRIORITY=7, XPRLIM=0, PRLIM=0,  MAXLOAD=6,  STARTED, NMXPRTY;
JOBCLASS U,  XPRTY=1, PRIORITY=0, XPRLIM=0, PRLIM=0,  MAXLOAD=6,  STARTED, NMXPRTY,
             NMPRIO;
JOBCLASS V,  XPRTY=9, PRIORITY=7, XPRLIM=0, PRLIM=0,  MAXLOAD=1,  STARTED;
JOBCLASS W,  XPRTY=1, PRIORITY=0, XPRLIM=0, PRLIM=0,  MAXLOAD=8,  STARTED, NSC,
             NMXPRTY;
JOBCLASS X,  XPRTY=0, PRIORITY=0, XPRLIM=0, PRLIM=0,  MAXLOAD=1,  STARTED, NSC,
             NMXPRTY, NMPRIO, NMLoad;
JOBCLASS Y,  XPRTY=2, PRIORITY=7, XPRLIM=0, PRLIM=0,  MAXLOAD=6,  STARTED, NMXPRTY;
JOBCLASS Z,  XPRTY=0, PRIORITY=0, XPRLIM=0, PRLIM=0,  MAXLOAD=4,  STARTED, NSC,
             NMXPRTY, NMPRIO;
```

```
MAXJOB 1000;
```

```
+MAXTASK 800;
```

```
,MULTLEV 140;
```

```
MAXIODEV 42;
```

```
OWCLASS A, PRIORITY=1, MEDIA=I10000;
OWCLASS B, PRIORITY=2, MEDIA=I10000;
OWCLASS C, PRIORITY=3, MEDIA=I10000;
OWCLASS D, PRIORITY=4, MEDIA=I10000;
OWCLASS E, PRIORITY=5, MEDIA=I10000;
OWCLASS F, PRIORITY=6, MEDIA=I10000;
OWCLASS G, PRIORITY=6, MEDIA=I10000;
OWCLASS H, PRIORITY=6, MEDIA=I10000;
OWCLASS I, PRIORITY=6, MEDIA=I10000;
OWCLASS J, PRIORITY=6, MEDIA=I10000;
OWCLASS K, PRIORITY=6, MEDIA=I10000;
OWCLASS L, PRIORITY=6, MEDIA=I10000;
OWCLASS M, PRIORITY=6, MEDIA=I10000;
OWCLASS N, PRIORITY=6, MEDIA=I10000;
OWCLASS O, PRIORITY=6, MEDIA=I10000;
OWCLASS P, PRIORITY=6, MEDIA=I10000;
OWCLASS Q, PRIORITY=6, MEDIA=I10000;
OWCLASS R, PRIORITY=6, MEDIA=I10000;
```



```
OWCLASS S, PRIORITY=6, MEDIA=I10000;
OWCLASS T, PRIORITY=6, MEDIA=I10000;
OWCLASS U, PRIORITY=6, MEDIA=I10000;
OWCLASS V, PRIORITY=6, MEDIA=I10000;
OWCLASS W, PRIORITY=6, MEDIA=I10000;
OWCLASS X, PRIORITY=6, MEDIA=I10000;
OWCLASS Y, PRIORITY=6, MEDIA=I10000;
OWCLASS Z, PRIORITY=6, MEDIA=I10000;

OWDFLT CLASS=C,      DEVCLASS=PR, TAPE=SYSOUT, FBANNER=2,  EBANNER=1,  BANCHAR=0Z;

OWDEVICE 1, DVID=xxxx, CLASS=ABCDEFGHIJKLMNOPQRSTUVWXYZ;
OWDEVICE 2, DVID=xxxx, CLASS=ABCDEFGHIJKLMNOPQRSTUVWXYZ;
OWDEVICE 3, DVID=xxxx, CLASS=ABCDEFGHIJKLMNOPQRSTUVWXYZ;
OWDEVICE 4, DVID=xxxx, CLASS=ABCDEFGHIJKLMNOPQRSTUVWXYZ;
OWDEVICE 5, DVID=xxxx, CLASS=ABCDEFGHIJKLMNOPQRSTUVWXYZ;
OWDEVICE 6, DVID=xxxx, CLASS=ABCDEFGHIJKLMNOPQRSTUVWXYZ;
OWDEVICE 7, DVID=xxxx, CLASS=ABCDEFGHIJKLMNOPQRSTUVWXYZ;
OWDEVICE 8, DVID=xxxx, CLASS=ABCDEFGHIJKLMNOPQRSTUVWXYZ;
OWDEVICE 9, DVID=xxxx, CLASS=ABCDEFGHIJKLMNOPQRSTUVWXYZ;
OWDEVICE 10, DVID=xxxx, CLASS=ABCDEFGHIJKLMNOPQRSTUVWXYZ;
OWDEVICE 11, DVID=xxxx, CLASS=ABCDEFGHIJKLMNOPQRSTUVWXYZ;
OWDEVICE 12, DVID=xxxx, CLASS=ABCDEFGHIJKLMNOPQRSTUVWXYZ;
OWDEVICE 13, DVID=xxxx, CLASS=ABCDEFGHIJKLMNOPQRSTUVWXYZ;
OWDEVICE 14, DVID=xxxx, CLASS=ABCDEFGHIJKLMNOPQRSTUVWXYZ;
OWDEVICE 15, DVID=xxxx, CLASS=ABCDEFGHIJKLMNOPQRSTUVWXYZ;

MAXFILE 2048;

MAXTAPE 64;

FILSHARE 141;

ACTSIZE 200;

.JORSIZE 200;

.DEVTRACE ABN, WARN, ATN, ALARM;

PRLOG THRESHLD=50, COMMAND=NO;

MAXDEF 512;

.DUALSYS AABCCDD, MYSELF=FFFFFFF;

.BJSIMU 100;

.DUMPACT COMMAND=EJR DUMPJOB :USER.JCLLIB;

GAC LONGWAIT=40, LOCKSIZE=128, NBLOCKID=1024;
```



```
INPUTCTL DKMEDIA=CHB, JTRA=TERM, SJREADER=TERM, SLLIB=SYS.HSLLIB;

MAXCAT 128;

DJPCTL MAXTFSIT=4;

PRLOGC THRESHLD=50, COMMAND=NO;

.LSYS NAME=SST1;

QUEUED NCACHE;
-----

( ) COMMAND BY DEFAULT : NO EXPLICIT PARAMETER, NO OVERRIDDEN PARAMETER.
(.) SOME EXPLICIT PARAMETER PROVIDED TO STEP, NO OVERRIDDEN PARAMETER.
(+) SOME EXPLICIT PARAMETER VALUE OF THIS COMMAND HAS BEEN OVERRIDDEN.

UNDEFINED PARAMETERS/COMMANDS ARE NOT PRINTED
-----

SUCCESSFUL CONFIGURATION : FILE SYS.SYSTEM HAS BEEN UPDATED
TO RUN CONFIGURED SYSTEM, USE ISL-OPTIONS : RESTORE,RESTART(COLD)
```

---



**IMPORTANT:**

Network configuration could be mandatory



### 5.6.6 Successful Completion Message

If no errors or warnings have been detected, the following message is output on the listing and at the operator's console:

```
SUCCESSFUL CONFIGURATION: FILE SYS.SYSTEM HAS BEEN UPDATED.  
TO RUN CONFIGURED SYSTEM, USE ISL-OPTIONS:  
RESTORE, RESTART (COLD)
```



#### **IMPORTANT:**

Network configuration could be mandatory

#### **NOTES:**

1. After a restore session following a CONFIG run, all devices except those listed implicitly or explicitly in ILO1 are in HOLD state. To use these devices, an RDV (release device) command is necessary.
2. If CONFIG is being executed on the running Set and the CONFIG statement LSYS is being used to specify a new Local SYStem name, the telecommunications network must be modified, by rerunning the network generation utility NG.

### 5.6.7 Unsuccessful Completion Message(s)

Each error message is preceded by one or three stars, depending on the severity of the error.

- \* One star indicates a warning; execution continues.
- \*\* Three stars indicate a fatal error; execution is aborted.



### 5.6.8 Hardware Configuration Listing

An example of a configuration extracted from the hardware configuration table is shown in the following pages.

Every hardware component configured in the installation is identified by its user-specified SRST "name" which the system associates with:

- its LDN (logical device number) for use by the software
- its "path" for the use of hardware.

The "path" is in the hexadecimal format XYZZ, where:

- X is the IOC and set to 0
- Y is the "physical channel"
- ZZ is the "logical channel".

The hardware configuration is arbitrary, the SRST "name" and the "path" varying from installation to installation.

In the example, LN12 of LDN 08 is followed by LN.. of LDN 09, in the same way as LN11 0A is followed by LN.. of LDN 0B.

This is possible when LN11 and LN12 are to be used for VIP and TRANSPAC communications.

#### Typical Hardware Device Configuration Listing

```

*****
****GCOS7                                     ****
****          H _ C O N F I G                 ****
****          VERSION: 92.00 DATED: APR 14, 1986 ****
*****SYSTEM_CONFIGURATION_REPORT*****
*****

```

ADDITIONAL INFO: 6 8 16 21 22 24

001	UC30	8400	002	UC80	AC00
003	UC90	9800	004	UC40	8C00
005	UC20	CB00	006	UC01	FE00
007	TC02	CA00	008	TC01	2200
009	TC56	B900	00A	TC52	A900
00B	TC42	8900	00C	TC51	A400
00D	TC04	C500	00E	TC03	B400
00F	MC06	C100	010	MC02	C000
011	MC04	C400	012	MCC9	C900




---

013	MC01	C800	014	MC13	2100
015	MC11	2000	016	MC24	2400
017	MC15	2900	018	MC12	2800
019	MCM1	BC00	01A	MCL3	A000
01B	MCK4	A200	01C	MCL2	9E00
01D	MCK1	9C00	01E	MCM4	8200
01F	CC01	C300	020	CC02	C200
021	CC11	9000	022	CC12	2A00
023	MF11	FE09	024	MF01	FE08
025	MF23	CB03	026	MF22	CB02
027	MF21	CB01	028	MF82	AC02
029	MF81	AC01	02A	MF92	9802
02B	MF91	9801	02C	MF42	8C02
02D	MF41	8C01	02E	MF33	8408
02F	MF32	8409	030	MF31	8401
031	DI01	CB0E	032	CT58	B903
033	CT57	B902	034	CT56	B901
035	CT04*	B40A	035	CT04*	C50A
036	CT03*	B409	036	CT03*	C509
037	CT02*	B402	037	CT02*	C502
038	CT01*	B401	038	CT01*	C501
039	CT51	A402	03A	CT50	A401
03B	CT43	8902	03C	CT42	8901
03D	CT53	A902	03E	CT52	A901
03F	LT28.	CB1F	040	LT28.	CB20
041	LT27.	CB1D	042	LT27.	CB1E
043	LT26.	CB1B	044	LT26.	CB1C
045	LT25.	CB19	046	LT25.	CB1A
047	LT24.	CB17	048	LT24.	CB18
049	LT23.	CB15	04A	LT23.	CB16
04B	LT22.	CB13	04C	LT22.	CB14
04D	LT21.	CB11	04E	LT21.	CB12
04F	LT86.	AC0D	050	LT86.	AC0E
051	LT85.	AC0B	052	LT85.	AC0C
053	LT84.	AC09	054	LT84.	AC0A
055	LT83.	AC07	056	LT83.	AC08
057	LT82.	AC05	058	LT82.	AC06
059	LT81.	AC03	05A	LT81.	AC04
05B	LT96.	980D	05C	LT96.	980E
05D	LT95.	980B	05E	LT95.	980C
05F	LT94.	9809	060	LT94.	980A
061	LT93.	9807	062	LT93.	9808
063	LT92.	9805	064	LT92.	9806
065	LT91.	9803	066	LT91.	9804
067	LT48.	8C11	068	LT48.	8C12
069	LT47.	8C0F	06A	LT47.	8C10
06B	LT46.	8C0D	06C	LT46.	8C0E
06D	LT45.	8C0B	06E	LT45.	8C0C
06F	LT44.	8C09	070	LT44.	8C0A
071	LT43.	8C07	072	LT43.	8C08
073	LT42.	8C05	074	LT42.	8C06

---





---

075	LT41.	8C03	076	LT41.	8C04
077	LN08	FE07	078	LN07	FE06
079	LN06	FE05	07A	LN05	FE04
07B	LN04	FE03	07C	LN03	FE02
07D	LN02	FE01	07E	CL3F	C30F
07F	CL3E	C30E	080	CL3D	C30D
081	CL3C	C30C	082	CL3B	C30B
083	CL3A	C30A	084	CL39	C309
085	CL38	C308	086	CL37	C307
087	CL36	C306	088	CL35	C305
089	CL34	C304	08A	CL33	C303
08B	CL32	C302	08C	CL31	C301
08D	CL2F	C20F	08E	CL2E	C20E
08F	CL2D	C20D	090	CL2C	C20C
091	CL2B	C20B	092	CL2A	C20A
093	CL29	C209	094	CL28	C208
095	CL27	C207	096	CL26	C206
097	CL25	C205	098	CL24	C204
099	CL23	C203	09A	CL22	C202
09B	CL21	C201	09C	CL1F	900F
09D	CL1E	900E	09E	CL1D	900D
09F	CL1C	900C	0A0	CL1B	900B
0A1	CL1A	900A	0A2	CL19	9009
0A3	CL18	9008	0A4	CL17	9007
0A5	CL16	9006	0A6	CL15	9005
0A7	CL14	9004	0A8	CL13	9003
0A9	CL12	9002	0AA	CL11	9001
0AB	CL0F	2A0F	0AC	CL0E	2A0E
0AD	CL0D	2A0D	0AE	CL0C	2A0C
0AF	CL0B	2A0B	0B0	CL0A	2A0A
0B1	CL09	2A09	0B2	CL08	2A08
0B3	CL07	2A07	0B4	CL06	2A06
0B5	CL05	2A05	0B6	CL04	2A04
0B7	CL03	2A03	0B8	CL02	2A02
0B9	CL01	2A01	0BA	PR31	8402
0BB	PR21	CB07	0BC	MT04*	2204
0BC	MT04*	CA04	0BD	MT03*	2203
0BD	MT03*	CA03	0BE	MT02*	2202
0BE	MT02*	CA02	0BF	MT01*	2201
0BF	MT01*	CA01	0C0	MS16*	2108
0C0	MS16*	C408	0C1	MS15*	2107
0C1	MS15*	C407	0C2	MS14*	2106
0C2	MS14*	C406	0C3	MS13*	2105
0C3	MS13*	C405	0C4	MS12*	2104
0C4	MS12*	C404	0C5	MS11*	2103
0C5	MS11*	C403	0C6	MS10*	2102
0C6	MS10*	C402	0C7	MS09*	2101
0C7	MS09*	C401	0C8	MS24*	2908
0C8	MS24*	C108	0C9	MS23*	2907
0C9	MS23*	C107	0CA	MS22*	2906
0CA	MS22*	C106	0CB	MS21*	2905



0CB	MS21*	C105	0CC	MS20*	2904
0CC	MS20*	C104	0CD	MS19*	2903
0CD	MS19*	C103	0CE	MS18*	2902
0CE	MS18*	C102	0CF	MS17*	2901
0CF	MS17*	C101	0D0	MS07*	2007
0D0	MS07*	2807	0D0	MS07*	C007
0D0	MS07*	C807	0D1	MS06*	2006
0D1	MS06*	2806	0D1	MS06*	C006
0D1	MS06*	C806	0D2	MS05*	2005
0D2	MS05*	2805	0D2	MS05*	C005
0D2	MS05*	C805	0D3	MS04*	2004
0D3	MS04*	2804	0D3	MS04*	C004
0D3	MS04*	C804	0D4	MS03*	2003
0D4	MS03*	2803	0D4	MS03*	C003
0D4	MS03*	C803	0D5	MS02*	2002
0D5	MS02*	2802	0D5	MS02*	C002
0D5	MS02*	C802	0D6	MS01*	2001
0D6	MS01*	2801	0D6	MS01*	C001
0D6	MS01*	C801	0D7	MSM4*	BC04
0D7	MSM4*	8204	0D8	MSM3*	BC03
0D8	MSM3*	8203	0D9	MSM2*	BC02
0D9	MSM2*	8202	0DA	MSM1*	BC01
0DA	MSM1*	8201	0DB	MSL4*	9E04
0DB	MSL4*	A004	0DC	MSL3*	9E03
0DC	MSL3*	A003	0DD	MSL2*	9E02
0DD	MSL2*	A002	0DE	MSL1*	9E01
0DE	MSL1*	A001	0DF	MSK4*	9C04
0DF	MSK4*	A204	0E0	MSK3*	9C03
0E0	MSK3*	A203	0E1	MSK2*	9C02
0E1	MSK2*	A202	0E2	MSK1*	9C01
0E2	MSK1*	A201	0E3	MS32+	C904
0E4	MS31+	C903	0E5	MS30+	C902
0E6	MS29+	C901	0E7	MS28+	2404
0E8	MS27+	2403	0E9	MS26+	2402
0EA	MS25+	2401			

-----  
 (.) MULTI LC DEVICE  
 (\*) MULTI PC DEVICE  
 (+) SHARABLE DEVICE  
 -----

```
0001 ACCOUNT ALL;
0002 ACTSIZE 800;
0003 BJSIMU 64;
0004 DJPCTL MAXTFSIT=6;
0005 DUMPACT COMMAND='EJ DUMPACT, , SITE.CONFIG;';
0006 FILSHARE 150;
0007 GAC LONGWAIT=150, LOCKSIZE=320, NBLOCKID=4095;
0008 INPUTCTL SJREADER=NTERM, JTRA=NTERM, HSLLIB=SYS.HSLLIB;
0009 JOBCLASS A, MAXLOAD=6, XPRTY=2, PRIORITY=2;
0010 JOBCLASS B, MAXLOAD=6, XPRTY=3, PRIORITY=3;
```



```

0011 JOBCLASS M,MAXLOAD=1,XPRTY=3,PRIORITY=3,NMLOAD;
0012 JOBCLASS Q,MAXLOAD=50;
0013 JOBCLASS R,MAXLOAD=4;
0014 JOBCLASS Z,MAXLOAD=20;
0015 JORSIZE 5000;
0016 JPPCDIM 2;
0017 LSYS NAME=BCC9;
0018 MAXCAT 128;
0019 MAXDEF 512;
0020 MAXFILE 5000;
0021 MAXJOB 2356;
0022 MAXTAPE 64;
0023 MAXTASK 1000;
0024 MULTLEV 254;
0025 OWDFLT DEVCLASS=PR/H136;
0026 PRLOG COMMAND='EJ PRLOG,,SYS.HSLLIB';
0027 PRLOGC COMMAND='EJ PRLOGC,,SYS.HSLLIB';
0028 QUEUED NCACHE;
0029 SADMPT MIRROR=YES GCLKSADM=YES;
0030 SITEOPT 00001040;
0031 VOLORG ALL;
0032 XAS XBUF=8192;
0033 SECOPT SA7LOGON=NO SA7ADMIN=NO SA7NOC SL=YES CHKPW=YES;
0034 DUALSYS 89500028;

```

```

MAXIMUM NUMBER OF KNOWN JOBS           :    2610
MAXIMUM NUMBER OF STARTED JOBS         :    254
MAXIMUM NUMBER OF CLASSES              :    256
CLASSES CONFIGURATION :

```

```

CLASS MAXLOAD PRIORITY XPRTY STARTED SLD MXPRTY MPRIO MLOAD PRLIM XPR LIM
ELAPTIME CPTIME JCG

```

A	6	2	2	Y	Y	Y	Y	Y	0	0	9999	9999999
B	6	3	3	Y	Y	Y	Y	Y	0	0	9999	9999999
C	1	7	9	Y	Y	Y	Y	Y	0	0	9999	9999999
D	1	1	5	Y	Y	Y	Y	Y	0	0	9999	9999999
E	1	2	4	Y	Y	Y	Y	Y	0	0	9999	9999999
F	1	3	7	Y	Y	Y	Y	Y	0	0	9999	9999999
G	1	4	9	Y	Y	Y	Y	Y	0	0	9999	9999999
H	1	6	0	Y	Y	Y	Y	Y	0	0	9999	9999999
I	1	7	9	Y	Y	Y	Y	Y	0	0	9999	9999999
J	1	6	0	Y	Y	Y	Y	Y	0	0	9999	9999999
K	1	7	9	Y	Y	Y	Y	Y	0	0	9999	9999999
L	1	7	9	Y	Y	Y	Y	Y	0	0	9999	9999999
M	1	3	3	Y	Y	Y	Y	Y	0	0	9999	9999999
N	1	7	9	Y	Y	Y	Y	Y	0	0	9999	9999999
O	1	7	9	Y	Y	Y	Y	Y	0	0	9999	9999999
P	40	7	9	Y	Y	Y	Y	Y	0	0	9999	9999999
Q	50	0	1	Y	Y	Y	Y	Y	0	0	9999	9999999



R	4	0	1	Y		Y	Y	0	0	9999	9999999
S	12	0	0	Y				0	0	9999	9999999
T	6	7	2	Y	Y	Y	Y	0	0	9999	9999999
U	6	0	1	Y	Y		Y	0	0	9999	9999999
V	1	7	9	Y	Y	Y	Y	0	0	9999	9999999
W	8	0	1	Y		Y	Y	0	0	9999	9999999
X	4	0	0	Y				0	0	9999	9999999
Y	6	7	2	Y		Y	Y	0	0	9999	9999999
Z	20	0	0	Y			Y	0	0	9999	9999999
RB	175	0	1	Y				0	1	9999	9999999

```
.ACCOUNT ALL;
.JOBCLASS A,XPRTY=2,PRIORITY=2,XPRLIM=0,PRLIM=0,MAXLOAD=6,JCG=,ELAPTIME=9999,
CPTIME=9999999,STARTED;
.JOBCLASS B,XPRTY=3,PRIORITY=3,XPRLIM=0,PRLIM=0,MAXLOAD=6,JCG=,ELAPTIME=9999,
CPTIME=9999999,STARTED;
.JOBCLASS C,XPRTY=9,PRIORITY=7,XPRLIM=0,PRLIM=0,MAXLOAD=1,JCG=,ELAPTIME=9999,
CPTIME=9999999,STARTED,NSC;
.JOBCLASS D,XPRTY=5,PRIORITY=1,XPRLIM=0,PRLIM=0,MAXLOAD=1,JCG=,ELAPTIME=9999,
CPTIME=9999999,STARTED;
.JOBCLASS E,XPRTY=4,PRIORITY=2,XPRLIM=0,PRLIM=0,MAXLOAD=1,JCG=,ELAPTIME=9999,
CPTIME=9999999,STARTED;
.JOBCLASS F,XPRTY=7,PRIORITY=3,XPRLIM=0,PRLIM=0,MAXLOAD=1,JCG=,ELAPTIME=9999,
CPTIME=9999999,STARTED;
.JOBCLASS G,XPRTY=9,PRIORITY=4,XPRLIM=0,PRLIM=0,MAXLOAD=1,JCG=,ELAPTIME=9999,
CPTIME=9999999,STARTED;
.JOBCLASS H,XPRTY=0,PRIORITY=6,XPRLIM=0,PRLIM=0,MAXLOAD=1,JCG=,ELAPTIME=9999,
CPTIME=9999999,STARTED;
.JOBCLASS I,XPRTY=9,PRIORITY=7,XPRLIM=0,PRLIM=0,MAXLOAD=1,JCG=,ELAPTIME=9999,
CPTIME=9999999,STARTED;
.JOBCLASS J,XPRTY=0,PRIORITY=6,XPRLIM=0,PRLIM=0,MAXLOAD=1,JCG=,ELAPTIME=9999,
CPTIME=9999999,STARTED;
.JOBCLASS K,XPRTY=9,PRIORITY=7,XPRLIM=0,PRLIM=0,MAXLOAD=1,JCG=,ELAPTIME=9999,
CPTIME=9999999,STARTED,NSC;
.JOBCLASS L,XPRTY=9,PRIORITY=7,XPRLIM=0,PRLIM=0,MAXLOAD=1,JCG=,ELAPTIME=9999,
CPTIME=9999999,STARTED,NSC;
.JOBCLASS M,XPRTY=3,PRIORITY=3,XPRLIM=0,PRLIM=0,MAXLOAD=1,JCG=,ELAPTIME=9999,
CPTIME=9999999,STARTED,NMLOAD;
.JOBCLASS N,XPRTY=9,PRIORITY=7,XPRLIM=0,PRLIM=0,MAXLOAD=1,JCG=,ELAPTIME=9999,
CPTIME=9999999,STARTED,NSC;
.JOBCLASS O,XPRTY=9,PRIORITY=7,XPRLIM=0,PRLIM=0,MAXLOAD=1,JCG=,ELAPTIME=9999,
CPTIME=9999999,STARTED,NSC;
.JOBCLASS P,XPRTY=9,PRIORITY=7,XPRLIM=0,PRLIM=0,MAXLOAD=40,JCG=,ELAPTIME=9999,
CPTIME=9999999,STARTED;
.JOBCLASS Q,XPRTY=1,PRIORITY=0,XPRLIM=0,PRLIM=0,MAXLOAD=50,JCG=,ELAPTIME=9999,
CPTIME=9999999,STARTED,NMPRIO;
.JOBCLASS R,XPRTY=1,PRIORITY=0,XPRLIM=0,PRLIM=0,MAXLOAD=4,JCG=,ELAPTIME=9999,
CPTIME=9999999,STARTED,NSC,NMXPRTY;
.JOBCLASS S,XPRTY=0,PRIORITY=0,XPRLIM=0,PRLIM=0,MAXLOAD=12,JCG=,ELAPTIME=9999,
CPTIME=9999999,STARTED,NSC,NMXPRTY,NMPRIO,NMLOAD;
```



```
JOBCLASS T,XPRTY=2,PRIORITY=7,XPRLIM=0,PRLIM=0,MAXLOAD=6,JCG=,ELAPTIME=9999,
CPTIME=9999999,STARTED,NMXPRTY;
JOBCLASS U,XPRTY=1,PRIORITY=0,XPRLIM=0,PRLIM=0,MAXLOAD=6,JCG=,ELAPTIME=9999,
CPTIME=9999999,STARTED,NMXPRTY,NMPRIO;
JOBCLASS V,XPRTY=9,PRIORITY=7,XPRLIM=0,PRLIM=0,MAXLOAD=1,JCG=,ELAPTIME=9999,
CPTIME=9999999,STARTED,NSC;
JOBCLASS W,XPRTY=1,PRIORITY=0,XPRLIM=0,PRLIM=0,MAXLOAD=8,JCG=,ELAPTIME=9999,
CPTIME=9999999,STARTED,NSC,NMXPRTY;
+JOBCLASS X,XPRTY=0,PRIORITY=0,XPRLIM=0,PRLIM=0,MAXLOAD=4,JCG=,ELAPTIME=9999,
CPTIME=9999999,STARTED,NSC,NMXPRTY,NMPRIO,NMLOAD;
JOBCLASS Y,XPRTY=2,PRIORITY=7,XPRLIM=0,PRLIM=0,MAXLOAD=6,JCG=,ELAPTIME=9999,
CPTIME=9999999,STARTED,NSC,NMXPRTY;
.JOBCLASS Z,XPRTY=0,PRIORITY=0,XPRLIM=0,PRLIM=0,MAXLOAD=20,JCG=,ELAPTIME=9999,
CPTIME=9999999,STARTED,NSC,NMXPRTY,NMPRIO;
.MAXJOB 2356;
.MAXTASK 1000;
.MULTLEV 254;
.MAXIODEV 42;
OWCLASS A,PRIORITY=1,MEDIA=I10000,LINES=9999999;
OWCLASS B,PRIORITY=2,MEDIA=I10000,LINES=9999999;
OWCLASS C,PRIORITY=3,MEDIA=I10000,LINES=9999999;
OWCLASS D,PRIORITY=4,MEDIA=I10000,LINES=9999999;
OWCLASS E,PRIORITY=5,MEDIA=I10000,LINES=9999999;
OWCLASS F,PRIORITY=6,MEDIA=I10000,LINES=9999999;
OWCLASS G,PRIORITY=6,MEDIA=I10000,LINES=9999999;
OWCLASS H,PRIORITY=6,MEDIA=I10000,LINES=9999999;
OWCLASS I,PRIORITY=6,MEDIA=I10000,LINES=9999999;
OWCLASS J,PRIORITY=6,MEDIA=I10000,LINES=9999999;
OWCLASS K,PRIORITY=6,MEDIA=I10000,LINES=9999999;
OWCLASS L,PRIORITY=6,MEDIA=I10000,LINES=9999999;
OWCLASS M,PRIORITY=6,MEDIA=I10000,LINES=9999999;
OWCLASS N,PRIORITY=6,MEDIA=I10000,LINES=9999999;
OWCLASS O,PRIORITY=6,MEDIA=I10000,LINES=9999999;
OWCLASS P,PRIORITY=6,MEDIA=I10000,LINES=9999999;
OWCLASS Q,PRIORITY=6,MEDIA=I10000,LINES=9999999;
OWCLASS R,PRIORITY=6,MEDIA=I10000,LINES=9999999;
OWCLASS S,PRIORITY=6,MEDIA=I10000,LINES=9999999;
OWCLASS T,PRIORITY=6,MEDIA=I10000,LINES=9999999;
OWCLASS U,PRIORITY=6,MEDIA=I10000,LINES=9999999;
OWCLASS V,PRIORITY=6,MEDIA=I10000,LINES=9999999;
OWCLASS W,PRIORITY=6,MEDIA=I10000,LINES=9999999;
OWCLASS X,PRIORITY=6,MEDIA=I10000,LINES=9999999;
OWCLASS Y,PRIORITY=6,MEDIA=I10000,LINES=9999999;
OWCLASS Z,PRIORITY=6,MEDIA=I10000,LINES=9999999;
.OWDFLT CLASS=C,DEVCLASS=PR/H136,TAPE=SYSOUT,FBANNER=2,EBANNER=1,BANCHAR=0Z,
PAPERBRK=N,CHARBRK=0Z,BIGCHAR=N;
OWDEVICE 1,DVID=XXXX,CLASS=ABCDEFGHIJKLMNPOQRSTUVWXYZ;
OWDEVICE 2,DVID=XXXX,CLASS=ABCDEFGHIJKLMNPOQRSTUVWXYZ;
OWDEVICE 3,DVID=XXXX,CLASS=ABCDEFGHIJKLMNPOQRSTUVWXYZ;
OWDEVICE 4,DVID=XXXX,CLASS=ABCDEFGHIJKLMNPOQRSTUVWXYZ;
OWDEVICE 5,DVID=XXXX,CLASS=ABCDEFGHIJKLMNPOQRSTUVWXYZ;
```



```

OWDEVICE 6 ,DVID=XXXX,CLASS=ABCDEFGHIJKLMNPOQRSTUVWXYZ ;
OWDEVICE 7 ,DVID=XXXX,CLASS=ABCDEFGHIJKLMNPOQRSTUVWXYZ ;
OWDEVICE 8 ,DVID=XXXX,CLASS=ABCDEFGHIJKLMNPOQRSTUVWXYZ ;
OWDEVICE 9 ,DVID=XXXX,CLASS=ABCDEFGHIJKLMNPOQRSTUVWXYZ ;
OWDEVICE 10 ,DVID=XXXX,CLASS=ABCDEFGHIJKLMNPOQRSTUVWXYZ ;
OWDEVICE 11 ,DVID=XXXX,CLASS=ABCDEFGHIJKLMNPOQRSTUVWXYZ ;
OWDEVICE 12 ,DVID=XXXX,CLASS=ABCDEFGHIJKLMNPOQRSTUVWXYZ ;
OWDEVICE 13 ,DVID=XXXX,CLASS=ABCDEFGHIJKLMNPOQRSTUVWXYZ ;
OWDEVICE 14 ,DVID=XXXX,CLASS=ABCDEFGHIJKLMNPOQRSTUVWXYZ ;
OWDEVICE 15 ,DVID=XXXX,CLASS=ABCDEFGHIJKLMNPOQRSTUVWXYZ ;
.MAXFILE 5000;
.MAXTAPE 64;
.FILSHARE 150;
.ACTSIZE 800;
.JORSIZE 5000;
.PRLOG THRESHLD=50,COMMAND=EJ PRLOG, ,SYS.HSLLIB;
.MAXDEF 512;
+DUALSYS 89500028,MYSELF=89500029;
.BJSIMU 64;
.DUMPACT COMMAND=EJ DUMPACT, ,SITE.CONFIG;;
.GAC LONGWAIT=150,LOCKSIZE=320,NBLOCKID=4095;
.INPUTCTL DKMEDIA=JCL,JTRA=NTERM,SJREADER=NTERM,HSLLIB=SYS.HSLLIB,HOABORT=NO;
.MAXCAT 128;
.SITEOPT 00001040;
.DJPCTL MAXTFSIT=6,MINEVERY=15;
.PRLOGC THRESHLD=50,COMMAND=EJ PRLOGC, ,SYS.HSLLIB;
.LSYS NAME=BCC9;
.XAS XBUF=8192;
.QUEUED NCACHE;
  DEFORG VBO;
.VOLORG ALL;
  VCAMSCN MAXISO=0,MAXDSA=2500,MAXTE=100;
.SECOPT SA7LOGON=NO,SA7ADMIN=NO,CHKPW=YES,NETSEC=NO,SA7NOCSL=YES,ERASE=NO;
.JPPCDIM 2;
.SADMOPT CHKJSADM=NO,GCLKPROJ=NO,GCLKSADM=YES,GCSADMOP=NO,MIRROR=YES,
GCL15BIN=NO,SKBSUJOB=NO;
  FILEOPT COMPTS=PREVIOUS,MNTWORK=NO,CTFIRST=NO,LASTDATE=YES,USESITE=NO,
  CTUNLOAD=NO;
-----
( ) COMMAND BY DEFAULT : NO EXPLICIT PARAMETER, NO OVERRIDDEN PARAMETER.
(.) SOME EXPLICIT PARAMETER PROVIDED TO STEP, NO OVERRIDDEN PARAMETER.
(+) SOME EXPLICIT PARAMETER VALUE OF THIS COMMAND HAS BEEN OVERRIDDEN.
UNDEFINED PARAMETERS/COMMANDS ARE NOT PRINTED
-----

SUCCESSFUL CONFIGURATION : FILE SYS.SYSTEM HAS BEEN UPDATED
TO RUN CONFIGURED SYSTEM, USE ISL-OPTIONS : RESTORE,RESTART(COLD)
IMPORTANT : NETWORK CONFIGURATION COULD BE MANDATORY

```



## 5.6.9 Error Messages

There are two kinds of error message:

1. General Error Messages. These are numbered messages which do not always signal errors as such (e.g., 0012). The messages are printed in numerical sequence.
2. Specific Error Messages. These messages refer to a statement name. Messages are printed in alphabetical sequence of statement names.

### 5.6.9.1 General Error Messages

Error 0000: UNSUCCESSFUL CONFIGURATION PROCESS

**Meaning**

Due to severe errors, the CONFIG run was not completed.

Error 0001: UNABLE TO OPEN SYS.SYSTEM

**Meaning**

The SYS.SYSTEM file could not be opened.

**Action**

Check that a SYS.SYSTEM file exists on the master system volume.

Error 0002: UNABLE TO READ INPUT FILE

**Meaning**

I/O error while reading the configuration statement file.

Error 0003: PREMATURE END OF STREAM

**Meaning**

'End of File' on the configuration statement file has been reached.

**Action**

Correct the last statement - a semicolon has been omitted.

Error 0004: INVALID SEPARATOR AFTER PARAMETER VALUE

**Meaning**

Self-explanatory.

Error 0005: ILLEGAL SEPARATOR SEQUENCE

**Meaning**

An illegal set of separators has been encountered; e.g., keyword=, , .

Error 0006: UNABLE TO CLOSE INPUT FILE

**Meaning**

The configuration statement file could not be closed.

Error 0007: UNABLE TO READ SYS.SYSTEM

**Meaning**

I/O error while reading SYS.SYSTEM file.



Error 0008: UNABLE TO CREATE A WORK SEGMENT

**Meaning**

Error internal to CONFIG.

Error 0009: ON STATEMENT statement-name NO VALID SEPARATOR  
BEFORE PROTECTED STRING.

**Meaning**

The statement identified contains ...=ZZ'string'.

**Action**

This should be written, ...='ZZ string'.

Error 0010: ON STATEMENT statement-name EMPTY PROTECTED STRING

**Meaning**

The statement identified contains an empty string.

**Action**

Omit COMMAND parameter or complete argument, if statements are DUMPACT  
or PRLOG.

Omit BANCHAR parameter or complete argument, if statement is OWDFLT.

Error 0011: ON STATEMENT statement-name NO VALID SEPARATOR  
AFTER PROTECTED STRING.

**Meaning**

The statement identified contains ...='string'ZZ.

**Action**

This should be written ...='string ZZ'.

Error 0012: NO USER SUPPLIED CONFIGURATION STATEMENT

**Meaning**

No user statements were taken into account during the CONFIG run.

Error 0013: ON STATEMENT statement-name UNKNOWN STATEMENT NAME

**Meaning**

The statement identified contains an unknown statement name.

**Action**

Correct the statement concerned.

Error 0014: ON STATEMENT statement-name ILLEGAL SEPARATOR  
AFTER FIRST SUBVERB'S NAME

**Meaning**

The statement identified contains an illegal separator after the first positional  
parameter.

Error 0015: ON LINE line-number UNKNOWN SUBVERB'S NAME

**Meaning**

The first positional parameter value of the identified statement is unknown.

Error 0016: ON STATEMENT statement-name KEYWORD/SIV NOT FOUND

**Meaning**

In the statement identified, a mandatory keyword parameter or self-identifying  
value is missing, or an unknown keyword has been used.





Error 0018: ON STATEMENT statement-name PRESENCE OF EXCLUSIVE SIVs

**Meaning**

In the statement identified, two or more mutually-exclusive self-identifying values have been used; e.g., STARTED and NSTARTED

Error 0019: ON STATEMENT statement-name ILLEGAL SEPARATOR AFTER  
AN SIV

**Meaning**

In the statement identified, an illegal separator has been used after a self-identifying value.

Error 0020: ON STATEMENT statement-name PARAMETER DUPLICATION

**Meaning**

In the statement identified, a parameter has been used more than once; e.g., STARTED and STARTED.

Error 0021: ON STATEMENT statement-name ILLEGAL SEPARATOR AFTER A  
KEYWORD

**Meaning**

In the statement identified, an illegal separator has been used after a keyword.

Error 0022: ON STATEMENT statement-name ILLEGAL STATEMENT /  
KEYWORD NAME (>8 CHARACTERS)

**Meaning**

A statement or keyword name is longer than 8 characters.

Error 0023: ON STATEMENT statement-name ILLEGAL PARAMETER TYPE

**Meaning**

In the statement identified, a parameter has been specified which is not applicable.

Error 0024: ON STATEMENT statement-name TOO LARGE PARAMETER VALUE

**Meaning**

In the statement identified, the value of a parameter exceeds the maximum permitted size.

Error 0025: ON STATEMENT statement-name DUPLICATE STATEMENT

**Meaning**

The statement identified, has already been used in the configuration statement set.

Error 0026: ON STATEMENT statement-name MANDATORY ARGUMENT MISSING

**Meaning**

In the statement identified, an argument introduced by the keyword specified has been omitted.

Error 0028: ON STATEMENT statement-name SEPARATOR FOUND  
INSTEAD OF STATEMENT NAME

**Meaning**

Self-explanatory.



Error 0029: ON STATEMENT statement-name STATEMENT WITHOUT  
PARAMETER

**Meaning**

Self-explanatory.

Error 0059: UNABLE TO CREATE LARGE WORK SEGMENT

**Meaning**

A request for creation of a large segment has not been performed.

Error 0060: UNABLE TO MODIFY LARGE SEGMENT SIZE

**Meaning**

A request for modification of a large segment has not been performed.

Error 0071: TOO MANY PARAMETERS

**Meaning**

Check the explicit configuration commands; some might be duplicated.

Error 0090: IMPORTANT : SYS.IUF FILE NOT FOUND

**Meaning**

The SYS.IUF standard file has not been found.

Error 0091: IMPORTANT : USER LAST CONFIGURATION COMMANDS HAVE  
NOT BEEN SAVED

**Meaning**

An error has occurred during saving of the user's explicit configuration commands.

Error 0099: CONFIG COMMAND SAVE : UNABLE TO OPEN OUTPUT SUBFILE

**Meaning**

The subfile to contain the user's explicit configuration commands cannot be created.

Error 0100: CONFIG COMMANDS SAVE, UNABLE TO READ INPUT FILE  
COMMANDS

**Meaning**

The explicit configuration commands submitted by the user cannot be read.

Error 0101: CONFIG COMMANDS SAVE : SYS.IUF FILE OVERFLOW

**Meaning**

The SYS.IUF file is full.

Error 0102: CONFIG COMMANDS SAVE : UNABLE TO WRITE IN OUTPUT  
SUBFILE

**Meaning**

The user's explicit configuration commands cannot be recorded in a SYS.IUF subfile.



### 5.6.9.2 Specific Error Messages

ERROR IN STATEMENT ACTSIZE: LIMIT FOR SIZE=2000 BLOCKS

**Meaning**

The value of ACTSIZE exceeds the legal limit of 2000.

ERROR IN STATEMENT GAC; WRONG PARAMETER VALUE IN GAC

**Meaning**

Incorrect values for one or more of the GAC parameters, have been specified; i.e., LOCKSIZE, LONGWAIT and/or NBLOCKID.

ERROR IN STATEMENT "JOBCLASS job-class, MAXLOAD;" EXCEEDS NUMBER OF JOBS

**Meaning**

The value of MAXLOAD in JOBCLASS exceeds the limit on the maximum number of jobs.

ERROR IN STATEMENT "JOBCLASS n, XPRTY;" LIMIT FOR DISPATCHING PRIORITY=9

**Meaning**

The value of XPRTY in JOBCLASS exceeds the range of dispatching priority values.

ERROR IN STATEMENT "JOBCLASS n, PRIORITY;" LIMIT FOR SCHEDULING PRIORITY: 7

**Meaning**

The value of PRIORITY in JOBCLASS exceeds the range of scheduling priority values.

ERROR IN STATEMENT "JOBCLASS n, NSC;" CLASS BECOMES UNAVAILABLE SINCE NOT STARTED AND SC NOT ALLOWED

**Meaning**

Both NSC and NSTARTED have been specified in JOBCLASS.

ERROR IN STATEMENT "JOBCLASS;" PRIORITY LIMIT DOES NOT FIT WITH PRIORITY VALUE OF THE CLASS

**Meaning**

The values of PRIORITY and PRLIM in JOBCLASS are not correctly related.

ERROR IN STATEMENT "JBCLASS2 position, JOBCLASS;" JOB CLASS NOT IN THE AA-ZZ RANGE

**Meaning**

The value of JBCLASS2 is out of the AA-ZZ range.

ERROR IN STATEMENT "JBCLASS2 position, JOBCLASS;" JOB CLASS ALREADY EXISTS

**Meaning**

The class given for job-class (JOBCLASS parameter) has already been defined.



ERROR IN STATEMENT "JBCLASS2 position, JOBCLASS;" CONFIGURATION NOT ALLOWED FOR THE SPECIFIED JOB CLASS

**Meaning**

The class given for job-class (JOBCLASS parameter) is a service job-class whose characteristics cannot be changed.

ERROR IN STATEMENT "JBCLASS2 position, JCG;" JOB CLASS GROUP NOT IN THE A-ZZ RANGE

ERROR IN STATEMENT "JOBCLASS job-class, JCG;" JOB CLASS GROUP NOT IN THE A-ZZ RANGE

**Meaning**

The value of JCG, in the JOBCLASS/JBCLASS2 statements, is out of the A-ZZ range.

ERROR IN STATEMENT "JBCLASS2 position, JCG;" JOB CLASS GROUP UNKNOWN

ERROR IN STATEMENT "JOBCLASS job-class, JCG;" JOB CLASS GROUP UNKNOWN

**Meaning**

The job class given for JCG is neither defined by a JOBCLASS statement nor by a JBCLASS2 statement.

ERROR IN STATEMENT "JBCLASS2 position, JCG;" JOB CLASS GROUP ALREADY CONNECTED

ERROR IN STATEMENT "JOBCLASS job-class, JCG;" JOB CLASS GROUP ALREADY CONNECTED

**Meaning**

In the JOBCLASS/JBCLASS2 statement, the job class given for JCG is already connected to a job class group. It cannot, itself, become a job class group.

ERROR IN STATEMENT "JBCLASS2 position, JCG;" CONNECTION NOT ALLOWED FOR THIS JOB CLASS

ERROR IN STATEMENT "JOBCLASS job-class, JCG;" CONNECTION NOT ALLOWED FOR THIS JOB CLASS

**Meaning**

In the JOBCLASS/JBCLASS2 statement, the job class defined by the current statement is a service job class that cannot be connected to a job class group.

ERROR IN STATEMENT "JBCLASS2 position, ELAPTIME;" LIMIT FOR ELAPTIME=9999

ERROR IN STATEMENT "JOBCLASS job-class, ELAPTIME;" LIMIT FOR ELAPTIME=9999

**Meaning**

In the JOBCLASS/JBCLASS2 statement, the value of ELAPTIME is negative, or exceeds 9999 (9999 meaning that there is no limit for step execution time).



ERROR IN STATEMENT "JBCLASS2 position, ELAPTIME;" PARAMETER NOT ALLOWED FOR THIS JOB CLASS  
ERROR IN STATEMENT "JOBCLASS job-class, ELAPTIME;" PARAMETER NOT ALLOWED FOR THIS JOB CLASS

**Meaning**

In the JOBCLASS/JBCLASS2 statement, the job class defined by the current statement is a service job class for which no limit to step execution time can be set.

ERROR IN STATEMENT "JBCLASS2 position, CPTIME;" LIMIT FOR CPTIME=9999999  
ERROR IN STATEMENT "JOBCLASS job-class, CPTIME;" LIMIT FOR CPTIME=9999999

**Meaning**

In the JOBCLASS/JBCLASS2 statement, the value of CPTIME is negative, or exceeds 9999999 (9999999 meaning that there is no limit for CPU use in a step).

ERROR IN STATEMENT "JBCLASS2 position, CPTIME;" PARAMETER NOT ALLOWED FOR THIS JOB CLASS  
ERROR IN STATEMENT "JOBCLASS job-class, CPTIME;" PARAMETER NOT ALLOWED FOR THIS JOB CLASS

**Meaning**

In the JOBCLASS/JBCLASS2 statement, the job class defined by the current statement is a service job class for which no limit to step execution time can be set.

ERROR IN STATEMENT "JORSIZE;" LIMIT FOR SIZE: 10000 RECORDS

**Meaning**

The value of JORSIZE exceeds the limit of 10000 records.

ERROR IN STATEMENT "MAXIODEV;" INCORRECT MAXIODEV PARAMETER 9 ASSUMED

**Meaning**

An incorrect value has been specified for MAXIODEV. The default value of 9 has been assumed.

ERROR IN STATEMENT "MAXJOB POSITIONAL1;" LIMIT FOR MAXIMUM NUMBER OF JOBS: 9999

**Meaning**

The value of MAXJOB exceeds the limit of 9999.

ERROR IN STATEMENT "MAXTASK;" LIMIT FOR NUMBER OF TASKS: 3900

**Meaning**

The value of MAXTASK exceeds the limit of 3900.

ERROR IN STATEMENT "OWCLASS;" INVALID OUTPUT CLASS PRIORITY, NORMAL DEFAULT VALUE ASSUMED

**Meaning**

The value of PRIORITY in OWCLASS, is not within the permitted range; the default value has therefore been assumed. Refer to the description of the OWCLASS statement, and to the summary of CONFIG Statement Default Values.



ERROR IN STATEMENT "OWCLASS;" UNACCEPTABLE DEFAULT MEDIA

**Meaning**

The argument of MEDIA in the OWCLASS statement is incorrect. The default value is assumed.

ERROR IN STATEMENT "OWDEVICE;" XXXX NOT PRESENT AT INSTALLATION

**Meaning**

The argument of DVID (XXXX) in OWDEVICE was not found when the SRST was searched; media XXXX is not available.

ERROR IN STATEMENT "OWDFLT;" DEVICE CLASS ERROR

**Meaning**

The argument of DEVCLASS in OWDFLT is incorrect. The default value, taken from the SRST, has therefore been used.

ERROR IN STATEMENT "OWDFLT;" UNACCEPTABLE "OWDFLT;"

**Meaning**

The default values do not match the devices. The argument of DEVCLASS in OWDFLT is incorrect according to the device in the SRST.

ERROR IN STATEMENT "OWDFLT;" ERRONEOUS BANNER n ASSUMED

**Meaning**

The values of either FBANNER and/or EBANNER in OWDFLT are incorrect. Value 1 is assumed for FBANNER and value 2 for EBANNER.

ERROR IN STATEMENT "OWDFLT;" INCORRECT TAPE PARAMETER. TAPE=SYSOUT ASSUMED

**Meaning**

The argument of TAPE in OWDFLT must be either SYSOUT or NSYSOUT. The default option, SYSOUT, has been used.

ERROR IN STATEMENT "PRLOG THRESHLD;" THRESHLD VALUE MUST BE LESS THAN 100 AND GREATER THAN 0

**Meaning**

An incorrect value has been specified for THRESHLD in PRLOG.

ERROR IN STATEMENT "JOBCLASS Q, MAXLOAD" EXCEEDS MAX NUMBER OF IOF JOBS

**Meaning**

The appropriate option has not been purchased; the number of IOFs cannot exceed 40.

ERROR IN STATEMENT "MAXJOB;" UNABLE SET SYSLOAD TABLE TO ITS APPROPRIATE SIZE

**Meaning**

System problem, or limit for MAXJOB is too high.



ERROR IN STATEMENT "MAXJOB;" UNABLE TO SET MAXSIZE OF SYSLOAD FROM  
MAXJOB

**Meaning**

System problem, or limit for MAXJOB is too high.

ERROR IN STATEMENT "MULTLEV"; LARGE IOF NOT PURCHASED, NB OF IOF  
CANNOT EXCEED 40

**Meaning**

Despite the general level of multiprogramming having been fixed at greater than  
40, the number of active IOFs may never exceed 40 if the appropriate option is not  
present on the system.

ERROR IN STATEMENT "MULTLEV"; UNABLE TO SET JET TABLE TO ITS  
APPROPRIATE SIZE

**Meaning**

System problem, or limit for MULTLEV is too high.







---

## 6. TAILOR

The TAILOR processor forms one part of the procedure for the installation of a new GCOS 7 Release. The purpose of the TAILOR processor is to build the Set of disks necessary for an operational system by using, as appropriate:

- A standard Production Set (P-Set) as input, or
- A standard Reference Set (R-Set) as input.

In addition to the TAILOR processor, the GIUF function COPY\_SITEFILES is available for copying the site-specific files that are to be kept with the new GCOS 7 Release.



## 6.1 System Files

An operational system includes a set of system files located on one or several FSA disks.

### 6.1.1 General Rules

Some system files must be present at each installation, and are said to be mandatory. Other system files may be absent because they relate to an optional feature which is not considered useful and/or because the feature has not been purchased. All such files are said to be optional. Moreover, various residency rules are enforced by GCOS 7, which affects the possible locations for the files on the disks.

TAILOR does not treat certain files, since they have their own processors:

- SITE.MFT (file used for management of VOLSETs). See the *MFT User and Administrator's Guide*.
- SITE.QUOTA (file used for management of disk occupation). See *Administering the Storage Manager*.

### 6.1.2 System File Domains

System files can be sorted into five Domains:

- The GCOS files Domain
- The DSA files Domain
- The OLTD files Domain
- The FIRMWARE files Domain
- The GSF files Domain.

The GCOS, OLTD, GSF, and FIRMWARE Domains must be present on all working sites. The DSA Domain is not obligatory. If telecommunications via a front-end processor are required, the DSA file Domain may be delivered and installed asynchronously.

The TAILOR commands detailed in this chapter are grouped according to the file Domain to which each belongs.



## 6.2 Production Sets and Reference Set

In a P2P configuration, a second production Set is built from the P-Set. The created Set is called the P2-Set. Advantages of the P2P-Configuration include:

- The P-Set and P2-Set work in alternate mode. At any given moment, either one of them may be the running system Set.
- Either Set may be mono- or multi-volume, independently of the other.
- The non-running Set may be updated without affecting the other Set.

The Reference-Set (R-Set) is a Set of disks created by the IUF job `INSTAL_RSET` at Installation time. This Set consists predominantly of loaded system files. Although it is possible to use the R-Set as the running Set, this is not what it is intended for. It is primarily intended to be used as the TAILOR input Set for building Sets of running disks. These Sets of running disks are in turn named "Production-Sets" (P-Sets).



### 6.3 TAILOR Constraints

Volumes defined as RSDVOL and RSDVOL1 must be FSA.

SYSVOL, RSDVOL, and RSDVOL1 are not allowed on mirrored disks.

The following TAILOR commands prevent the allocation of unacceptable system files on mirrored disks:

BKST1, ISL, LIB, PVMF, TVMF, MIRLOG, HALOCK

The following files may be allocated on FSA disks other than SYSVOL:

SYS.DSABLIB  
SYS.DSALMLIB  
SYS.DSASLIB  
SYS.DSACMD  
SYS.DSACONF  
SYS.DSACORE  
SYS.DSADUMP  
SYS.DSATRC\* (for FCP7 only)  
SYS.FW.BINLIB  
SYS.FW.CCULIB (not available for XTA)  
SYS.FW.GENLIB  
SYS.FW.LMLIB  
SYS.FW.NFTLIB (not available for XTA)  
SYS.FW.SLLIB  
SYS.FW.PATCH (not available for XTA)  
SYS.FW.PRT  
SYS.GSF.BINLIB  
SYS.GSF.LMLIB  
SYS.GSF.SLLIB  
SYS.GSF.SMLIB  
SYS.GSF.UFAS  
SITE.HALOCK (if applicable)  
SITE.MIRLOG (if applicable)  
SITE.CMS\_CDD\_namei (if applicable)  
SITE.CMS\_CSD (if applicable)  
SITE.CMS\_DS\_name1\_name2 (if applicable)



### 6.3.1 Introduction to TAILOR

TAILOR uses a processor with a command language which allows the five system file Domains (GCOS, DSA, OLTD, FW, and GSF) to be treated independently of each other.

For each system file Domain, a set of commands is available for the definition of the characteristics for those files, which have selectable features, and for the disks to support them.

TAILOR builds the Set of disks in two steps:

1. Command analysis
2. File creation.

The second step does not start until the first is finished, which means that requests are stored and not immediately executed.

There are three ways of giving commands, each corresponding to a specific environment. Commands may be submitted from:

- A file
- The MAIN console
- An IOF console.

### 6.3.2 TAILOR and GCL

TAILOR uses the GCL interpreter to manage its commands, which makes available HELP, prompting facilities, and menus, according to the calling environment.



### 6.3.3 Structure of the TAILOR Command Language

#### SESSION AND DEFINITION COMMANDS

TAILOR commands may be sorted into two categories:

- "Session" commands, which determine the actions to be taken during a TAILOR session.
- "Definition" commands, which are used to define the characteristics of the disks and files which make up the "Production-Set".

As we saw earlier, each system file Domain is associated with a command set. A given Domain-related "definition" command set is accessed via the TLRGEN session command.

Each definition command set includes two types of command:

- Commands that define characteristics for the disks that are to contain the files of the Domain concerned.
- Commands which define characteristics for the files belonging to the specified Domain.

Note that there are not as many file definition commands as files in any given Domain, because some mandatory files have no selectable features.

A sequence of TAILOR commands related to any one of the Domains - GCOS, DSA, OLTD, GSF, or FW - is described as a "TLRGEN Session".

#### DEFAULT VALUES

To create a file, TAILOR needs a set of values, or allocation parameters. The file definition command parameters supply these.

All have default values. If no definition command is given for a mandatory file, its default values will be used. For optional files, the same rule applies: the activation of a file definition command is needed only if any of its default parameters does not match user requirements.

A TLRGEN session results in a set of values obtained by merging default and user values. This set of values is stored in a permanent file and may be used for the default values for the next TLRGEN session on the same Domain.

This set of values may be updated at each TLRGEN session and constitutes, for a given Domain, a dynamic user-specific set of defaults, called "last COMSET".



In the following pages, "initial COMSET" specifies a standard delivered set of default values, and "last COMSET" (or "latest" COMSET) stands for the user-specific set. See the paragraph *Default TLRGEN Sessions* below for more information on default values.

**IMPORTANT:**

1. If COMSET = LAST when the SYSVOL concerned does not belong to the same DEVCLASS as that declared for the last (i.e., previous latest) TLRGEN session, certain default file definition parameters (especially SIZE) must be modified, as they would otherwise be insufficient.
2. User subfiles should never be placed in system files/libraries, as they would be lost when updating or returning to a previous Technical Status. The only exceptions to this rule are the SITE.HELP file and the SYS.DSACONF file.

It is recommended, in these two cases only, that suitable user objects be stored in the files concerned where they will be preserved in the event of a system update or a return to an earlier Technical Status.



---

## 6.4 Session Commands

This command level is designated by the specific prefix "T". The seven session commands are:

- TLRGEN
- TLREXEC
- TLRPRINT
- TLRRESET
- TLRCANCEL
- TLRSTAT
- QUIT

The session command level may be left by entering either:

- QUIT;
- / (alone on a line).





## The TLRGEN command

### Function

To start a TLRGEN session.

### Format

```
TLRGEN FILESET={GCOS|DSA|OLTD|FW|GSF}  
        [COMSET={LAST|INITIAL}] ;
```

### Parameters

FILESET	is mandatory and names the file Domain for which the accompanying definition commands are to be used.
COMSET	is used to specify the set of default values to be applied to the definition commands.
When COMSET =	
LAST	the values resulting from the last TLRGEN session for the specified Domain, are taken as the defaults; this is the default value for COMSET.
INITIAL	the set of standard initial default values is used.

### Rules

COMSET = LAST may be used only if a "latest" COMSET exists for the specified Domain.

For a given Domain, this command cannot be issued after a TLREXEC command.

### EXAMPLES:

```
TLRGEN FILESET=GCOS COMSET=INITIAL;
```

```
TLRGEN DSA LAST;
```

```
TLRGEN FW;
```

```
TLRGEN GSF;
```

□



## The TLREXEC Command

### Function

To specify the file Domain for which files are to be created.

### Format

```
TLREXEC FILESET={GCOS|DSA|OLTD|FW|GSF}
        [FROM={BEG|alphanum8}] ;
```

### Parameters

FILESET	is mandatory and defines the Domain for which files are to be created.
FROM	is used after an ABORT during file creation in the Domain concerned, in order to define a restart label name (as displayed in the creation phase report for the time at which the ABORT occurred). By default, FROM = BEG is assumed; this is appropriate for the normal case where no files have yet been created for the Domain.

### Rules

- The TLREXEC command may be given only if a complete "latest" COMSET exists for the specified Domain.
- Only the latest TLREXEC command activation may be used for a given Domain.

### EXAMPLES:

```
TLREXEC FILESET=GCOS FROM=BEG;
```

```
TLREXEC DSA;
```

```
TLREXEC GSF;
```

```
TLREXEC FILESET=FW FROM=F5;
```

```
TLREXEC DSA D3;
```

□



## The TLRPRINT Command

### Function

To display a summary of the last COMSET for a given file Domain.

### Format

```
TLRPRINT FILESET={GCOS|DSA|OLTD|FW|GSF} ;
```

### Parameters

FILESET is mandatory and names the file Domain whose last COMSET is to be summarized.

### Rules

The TLRPRINT command may be given only if a "latest" COMSET exists for the file Domain concerned.

### EXAMPLES:

```
TLRPRINT FILESET=GCOS;
```

```
TLRPRINT DSA;
```

```
TLRPRINT GSF;
```

□



## The TLRRESET Command

### Function

To clear the "latest" COMSETs, either completely or partially, for all the file Domains.

### Format

```
TLRRESET [RESET={STRONG|WEAK}] ;
```

### Parameters

STRONG	means that all the "latest" COMSETs are reset. This is the default option.
WEAK	means that only those values relevant to disk characteristics are reset.

### Rules

- A TLRRESET command cannot follow TLREXEC commands.
- After TLRRESET RESET = STRONG, the only available default values for TLRGEN sessions are those of the initial COMSET.
- After TLRRESET RESET = WEAK, "latest" COMSETs may be used for TLRGEN sessions, but cannot be used for TLREXEC or TLRSTAT.

### EXAMPLES:

```
TLRRESET RESET=STRONG;
```

```
TLRRESET WEAK;
```

```
TLRRESET;
```

□



## The TLRCANCEL Command

### Function

To cancel the TLREXEC command for a given file Domain.

### Format

```
TLRCANCEL FILESET={GCOS|DSA|OLTD|FW|GSF} ;
```

### Parameters

FILESET represents the name of the file Domain whose creation request is to be cancelled.

### Rules

The TLRCANCEL command has no effect if no TLREXEC command has been activated for the Domain concerned.

### NOTE:

The TLRCANCEL command is the only way of modifying definition command parameter values for a given Domain after a TLREXEC command has been issued.

### EXAMPLES:

```
TLRCANCEL FILESET=GCOS;
```

```
TLRCANCEL DSA;
```

```
TLRCANCEL GSF;
```

□



---

## The TLRSTAT Command

### Function

To display a summary of TAILOR processing.

### Format

TLRSTAT;

### NOTES:

In the resulting printout:

1. "Executable" means that a complete last COMSET exists
2. "To be built" means that a TLREXEC command has been activated for the specified Domain.



## 6.5 Definition Commands

The default values applied to definition commands may differ, depending on the value supplied for the COMSET parameter in the TLRGEN command.

For this reason, no default values are specified in the definition command descriptions. The standard initial default values are given in Table 6-1 *GCOS File Domain Default Values*.

### Disk Definition Commands

These commands relate to the five Domains, GCOS, DSA, OLTD, GSF and FW, and are used to define disk characteristics. The disk definition commands are:

- SYSVOL
- RSDVOL
- RSDVOL1
- NRDVOL
- NRDVOL1
- CATVOL (for the GCOS Domain only).

All of these commands have the same syntax, requiring specification of the media name and device class for a disk.



---

## The SYSVOL Command

### Function

To define system disk characteristics.

### Format

```
SYSVOL MD=alphanum6 DVC=MS/FSA;
```

### Parameters

MD	is mandatory and introduces the MEDIA name of the disk.
DVC	is mandatory and specifies the DEVICE CLASS of the disk that must be FSA for the system disk.

### Rules

- The system disk must be FSA.
- The system disk is the same for the five file Domains, and may be defined in any Domain.
- Once defined, system disk characteristics cannot be changed until a TLRRESET command is issued.
- Any reference to the system disk in file definition commands must use the strings "SYS" or "SYSVOL".

### EXAMPLE:

```
SYSVOL MD=P27901 DVC=MS/FSA;
```

□





## The RSDVOL Command

### Function

To define the characteristics of a resident disk other than the system disk, for a given file Domain.

### Format

```
RSDVOL MD=alphanum6 DVC=MS/FSA;
```

### Parameters

MD	is mandatory and introduces the MEDIA name of the disk.
DVC	is mandatory and specifies the DEVICE CLASS of the disk.

### Rules

- A resident disk must be FSA.
- A resident disk defined using the RSDVOL command is private to the Domain that defines it.
- Any reference to this disk in file definition commands must be made with the strings "RSD" or "RSDVOL" (in the Domain for which RSDVOL command has been activated).

### EXAMPLE:

```
RSDVOL MD=P27902 DVC=MS/FSA;
```





## The RSDVOL1 Command

### Function

To define the characteristics of a second resident disk other than the system disk, for a given file Domain.

### Format

```
RSDVOL1 MD=alphanum6 DVC=MS/FSA;
```

### Parameters

MD	is mandatory and introduces the MEDIA name of the disk.
DVC	is mandatory and specifies the DEVICE CLASS of the disk.

### Rules

- A resident disk must be FSA.
- A resident disk defined using the RSDVOL1 command is private to the file Domain that defines it.
- Any reference to this disk in file definition commands must use the strings "RSD1" or "RSDVOL1" (in the Domain for which RSDVOL1 command has been activated).

### EXAMPLES:

```
RSDVOL1 MD=FSA3 DVC=MS/FSA;
```

```
RSDVOL1 FSA3 MS/FSA;
```





## The NRDVOL Command

### Function

To define, for a given file Domain, the characteristics of a non-resident disk.

### Format

```
NRDVOL MD=alphanum6 DVC=MS/FSA;
```

### Parameters

MD	is mandatory and introduces the MEDIA name of the disk.
DVC	is mandatory and specifies the DEVICE CLASS of the disk.

### Rules

- A non-resident disk defined using the NRDVOL command must be FSA.
- A non-resident disk defined using the NRDVOL command is private to the Domain that defines it.
- Any reference to this disk in file definition commands must use the strings "NRD" or "NRDVOL" (in the Domain for which NRDVOL command has been activated).

### EXAMPLES:

```
NRDVOL MD=FSA4 DVC=MS/FSA;
```

```
NRDVOL FSA4 MS/FSA;
```





## The NRDVOL1 Command

### Function

To define the characteristics of a second non-resident disk for a given file Domain.

### Format

```
NRDVOL1 MD=alphanum6 DVC=MS/FSA;
```

### Parameters

MD	is mandatory and introduces the MEDIA name of the disk.
DVC	is mandatory and specifies the DEVICE CLASS of the disk.

### Rules

- A non-resident disk defined using the NRDVOL1 command must be FSA.
- A non-resident disk defined using the NRDVOL1 command is private to the Domain that defines it.
- Any reference to this disk in file definition commands must use the strings "NRD1", or "NRDVOL1" (in the Domain for which the NRDVOL1 command has been activated).

### EXAMPLE:

```
NRDVOL1 MD=FSA5 DVC=MS/FSA;
```





## The CATVOL Command

### Function

To define the characteristics of the disk containing a valid SITE.CATALOG file for use in cataloging the site files created.

### Format

```
CATVOL MD=alphanumeric6 DVC=MS/FSA;
```

### Parameters

MD	is mandatory and introduces the MEDIA name of the disk.
DVC	is mandatory and specifies the DEVICE CLASS of the disk.

### Rules

- When SITE.STARTUP, SITE.HELP, SITE.IN, SITE.HALOCK or SITE.MIRLOG file creation is required while SITE.CATALOG file creation is not, both the CATVOL command and CATFL PRES = 0 must be specified explicitly.
- When SITE.CATALOG file creation is required (CATFL PRES = 1), the CATVOL command is ineffective.

### EXAMPLES:

```
CATVOL MD=P27901 DVC=MS/FSA;
```

```
CATVOL P27901 MS/FSA;
```

□



## Remarks on Disk Definition Commands

### DEFAULT VALUES

When starting a TLRGEN session, you can choose the set of default values to be applied as parameters of file definition commands.

When COMSET = INITIAL, no default values are assumed for disk characteristics, except for those of the system disk if this was defined in an earlier TLRGEN session.

When COMSET = LAST, the same rule applies as for file definition command parameters (see earlier in this chapter), in that the last-entered values are taken as the default values.



### IMPORTANT:

Remember that if COMSET = LAST when the SYSVOL concerned does not belong to the same DEVCLASS as that declared for the last (i.e. previous latest) TLRGEN session, certain default file definition parameters (especially SIZE) must be modified, as they would otherwise be insufficient.

### ABOUT THE DISKS

Each of these disks is private to a given file Domain, with the exception of the system disk. Therefore, if a disk has to support files belonging to different Domains, it must be defined separately for each of those Domains.

The system disk must be FSA.



## 6.6 The GCOS Domain

This Domain has the specific prefix "G" and contains the definition commands relevant to GCOS 7 system files. These commands may be used by activating the TLRGEN command with FILESET = GCOS.

In addition to the six disk definition commands already dealt with, the GCOS Domain contains a file-definition command for each of the files that have selectable features. These commands are:

BKST  
BKST1  
CATFL  
CLC  
CMS  
EOGEN  
ERLOG  
HALOCK  
ISL  
JAFTER  
JAS  
JBEFORE  
KNODET  
LIB  
LOGC  
MIRLOG  
PVMF  
QM  
QMBACK  
SITHELP  
SITIN  
SPOOL  
SPDUMP  
STUP  
SWLOG  
SYIN  
SYO  
SYSDUMP  
TVMF



### 6.6.1 Principal Parameters

File definition commands relevant to the GCOS Domain all have the same structure, featuring three main parameters as follows:

**PRES** is used to require or prevent the creation of an optional file. It does not usually exist for mandatory files.

**TO** is used to specify the location of a file. The values it may take are the symbolic disk names as described in the disk definition commands in the paragraph *Definition Commands* above. This parameter is not used for files that must be created on the system disk.

Any disk referenced by this parameter must have previously been defined using the appropriate disk-definition command.

**SIZE** is a decimal value used to specify the size of a file. Not applicable to files with a fixed size.

The other parameters used are described with each command.







## The BKST1 Command

### Function

Specifies the location and size of the SYS.BKST1 file and checks that it is not allocated on a mirrored disk.

### Format

```
BKST1 [PRES={0|1}] [TO={SYS|RSD|RSD1|NRD|NRD1}]  
      [SIZE=digit5];
```

### Parameters

PRES, TO	As described in the paragraph <i>Principal Parameters</i> . See the standard default value in Table 6-1 <i>GCOS File Domain Default Values</i> .
SIZE	As described in the paragraph <i>Principal Parameters</i> . See the standard default value in Table 6-1 <i>GCOS File Domain Default Values</i> . See also Calculating Backing-Store File Sizes in the appendix <i>System Files</i> . The maximum size cannot exceed 65535 blocks.

### EXAMPLES:

```
BKST1 PRES=1 TO=RSD SIZE=10000;
```

```
BKST1 1 SYS;
```

```
BKST1 1 SIZE=20000;
```

□



## The CATFL Command

### Function

To require or prevent creation of the SITE.CATALOG file as well as specifying its location and number of objects.

### Format

FSA:

```
CATFL [PRES={0|1}] [TO={SYS|RSD|RSD1}] [NBOBJ=digit6] ;
```

### Parameters

PRES, TO	As described in the paragraph <i>Principal Parameters</i> .
NBOBJ	Represents the maximum number of objects (100000 on an FSA disk) in the SITE.CATALOG file. See the standard default values in Table 6-1 <i>GCOS File Domain Default Values</i> .



**Rules**

- The SITE.CATALOG file can be on either the system disk or any other resident disk.
- See the tables *File Size and Number of Objects* at the end of this chapter for the correspondence between objects and blocks.

**EXAMPLES:**

```
CATFL PRES=1 TO=SYS NBOBJ=600;
```

```
CATFL 1 RSD;
```

```
CATFL NBOBJ=600;
```

```
CATFL PRES=0;
```

□



## The CLC Command

### Function

To require or prevent creation of the SYS.CLC file and to specify its location.

### Format

FSA:

```
CLC [PRES={0|1}] [TO={SYS|RSD|RSD1|NRD|NRD1}];
```

### Parameters

PRES, TO

As described in the paragraph *Principal Parameters*; see their standard default values in Table 6-1 *GCOS File Domain Default Values*.

### Rules

The SYS.CLC file may reside on any disk.

### EXAMPLES:

```
CLC PRES=1 TO=RSD1;
```

```
CLC 1 RSD;
```

```
CLC 1;
```

□



## The CMS Command

### Function

To require or prevent creation of the following files and to specify their location:

```
SITE.CMS_CSD  
SITE.CMS_CDD_name1  
SITE.CMS_CDD_name2  
SITE.CMS_DS_name1_name2
```

name1 and name2 are the network names of the two coupled systems.

### Format

```
CMS [PRES={0|1}] [TO={SYS|RSD|RSD1|NRD|NRD1}]  
NAME1=char4 NAME2=char4;
```

### Parameters

PRES, TO	As described in the paragraph <i>Principal Parameters</i> . See their standard default values in Table 6-1 <i>GCOS File Domain Default Values</i> .
NAME1	The network name of one coupled system.
NAME2	The network name of the other coupled system.

### Rules

The SITE.CMS\_\* files may be allocated on any FBO shared disk.

### EXAMPLES:

```
CMS PRES=1 TO=NRD NAME1=M001 NAME2=M002;
```

```
CMS 1 NRD1 M001 M002;
```

□



## The ERLOG Command

### Function

To require or prevent creation of the SYS.ERLOG file as well as to specify its location.

### Format

```
ERLOG [PRES={0|1}] [TO={SYS|RSD|RSD1}];
```

### Parameters

PRES, TO                      As described in the paragraph *Principal Parameters*.  
See their standard default values in Table 6-1 *GCOS File Domain Default Values*.

### Rules

The SYS.ERLOG file may be on any disk.

### NOTE:

The SYS.ERLOG file is mandatory so you must create it if TAILOR has not done this.

### EXAMPLES:

```
ERLOG PRES=1 TO=RSD;
```

```
ERLOG 1;
```





---

## The HALOCK Command

### Function

To require or prevent creation of the SITE.HALOCK file and to specify its location.

### Format

```
HALOCK  [PRES={0 | 1}] [TO={SYS | RSD | RSD1 | NRD | NRD1}]
```

### Parameters

PRES, TO                      As described in the paragraph *Principal Parameters*.  
See their standard default values in Table 6-1 *GCOS File Domain Default Values*.

### Rules

The SITE.HALOCK file may be allocated on any FBO disk.  
It may not be allocated on mirrored disks.

### EXAMPLES:

```
HALOCK  PRES=1  TO=RSD;
```

```
HALOCK  1;
```

□







## The JAFTER Command

### Function

To require or prevent creation of the SYS.JADIR file and of primary and secondary journal files and to specify their location and sizes.

### Format

```
JAFTER [PRES={0|1}] [TO={SYS|RSD|RSD1}] [SIZE=digit8]
[JOURNAL_SIZE=digit4] [BLOCK_SIZE=digit5]
[PRIMARY_MEDIA=name6] [PRIMARY_DVC=char11]
[SECONDARY_MEDIA=name6] [SECONDARY_DVC=char11]
[AUTOCYCLE=bool] [SEDCOPY=bool];
```

### Parameters

PRES, TO	As described in the paragraph <i>Principal Parameters</i> .
SIZE	Defines the size of the SYS.JADIR file.
JOURNAL_SIZE	Defines the size of the Primary Journal files in cylinders.
BLOCK_SIZE	Defines the block size of the Primary Journal files in bytes.
PRIMARY_MEDIA	Defines the media name of the Primary Journal files.
PRIMARY_DVC	Defines the device class of the Primary Journal files.
SECONDARY_MEDIA	Defines the media name of the Secondary Journal files.
SECONDARY_DVC	Defines the device class of the Secondary Journal files.
AUTOCYCLE	Sets the automatic cycling mode for JAS generation. (JAS = Journalization Advanced Service).
SEDCOPY	Sets the dual copy mode for JAS generation.

Standard default values for the above parameters are given in Table 6-1 *GCOS File Domain Default Values*.

For more information, refer to the *File Recovery Facilities User's Guide*.

**Rules**

1. The SYS.JADIR file may be allocated on any disk.
2. The allocation of After Journal primary and secondary files by TAILOR is standard. That is to say, with the JAFTER command primary files can be allocated only on disk, secondary files can be allocated only on tape or cartridge.
3. If PRES = 1, the P\_MD and P\_DVC parameters must be given values.  
In order to define a list of media names for primary and secondary After Journal files, the MAINTAIN\_JAS utility may be used. For information on MAINTAIN\_JAS, refer to the *File Recovery Facilities User's Guide*.

**EXAMPLES:**

```
JAFTER 0;
```

```
JAFTER PRES=1 TO=SYS SIZE=250 JOURNAL_SIZE=80  
BLOCK_SIZE=8192 PRIMARY_MEDIA=MED1  
PRIMARY_DVC=MS/FSA SECONDARY_MEDIA=MED2  
SECONDARY_DVC=MT SECDCOPY=1 AUTOCYCLE=1;
```

```
JAFTER PRES=1 TO=RSD PRIMARY_MEDIA=MED1  
PRIMARY_DVC=MS/FSA AUTOCYCLE=0;
```

□



## The JAS Command

### Function

To require or prevent creation of the SYS.JASBLUE and SYS.JASGREEN files and to specify their location.

### Format

```
JAS  [PRES={0 | 1}] [TO={SYS | RSD | RSD1 | NRD | NRD1}]
```

### Parameters

PRES, TO                      As described in the paragraph *Principal Parameters*. See their standard default values in Table 6-1 *GCOS File Domain Default Values*.

### Rules

The SYS.JASBLUE and SYS.JASGREEN files may be allocated on any disk.

### EXAMPLES:

```
JAS  1;
```

```
JAS  PRES=1  TO=RSD;
```

□



## The JBEFORE Command

### Function

To require or prevent creation of the SYS.JRNAL file and to specify its location, size and file block size.

### Format

FSA:

```
JBEFORE [PRES={0|1}] [TO={SYS|RSD|RSD1|NRD|NRD1}]  
        [SIZE=digit8] [{FILEBLOCK_SIZE|BLKSZ}={512|4096}];
```

### Parameters

PRES, TO, SIZE            As described in the paragraph *Principal Parameters*.  
See their standard default values in Table 6-1 *GCOS File Domain Default Values*.

FILEBLOCK\_SIZE        Defines the block size for the file in bytes.

### Rules

The SYS.JRNAL file may be on any disk.

### EXAMPLES:

```
JBEFORE PRES=1 TO=RSD1 SIZE=680 FILEBLOCK_SIZE=4096;
```

```
JBEFORE 1 RSD;
```

```
JBEFORE 1 SIZE=3;
```

```
JBEFORE 1 NRD 16368 512;
```

□



## The KNODET Command

### Function

To specify the location and size of the SYS.KNODET file.

### Format

FSA:

```
KNODET [TO={SYS|RSD|RSD1}] [SIZE=digit8];
```

### Parameters

TO, SIZE                      As described in the paragraph *Principal Parameters*.  
See the standard default values in Table 6-1 *GCOS File Domain Default Values*.

### Rules

The SYS.KNODET file may be on any resident disk.

### EXAMPLES:

```
KNODET TO=SYS SIZE=3000;
```

```
KNODET SIZE=2000;
```

```
KNODET RSD;
```

□



## The LIB Command

### Function

Specifies the location and size of the SYS.LIB file and checks that it is not allocated on a mirrored disk.

### Format

FSA:

```
LIB [TO={SYS|RSD|RSD1|NRD|NRD1}] [SIZE=digit8];
```

### Parameters

TO As described in the paragraph *Principal Parameters*.  
See the standard default value in Table 6-1 *GCOS File Domain Default Values*.

SIZE As described in the paragraph *Principal Parameters*.  
See the standard default value in Table 6-1 *GCOS File Domain Default Values*.

See also Calculating Backing-Store File Sizes, in the appendix *System Files*.

The maximum size cannot exceed 65535 blocks.

### EXAMPLES:

```
LIB TO=RSD SIZE=10000;
```

```
LIB SIZE=6800;
```

```
LIB NRD;
```

□



---

## The LOGC Command

### Function

To require or prevent creation of the SYS.LOGC file and to specify its location.

### Format

```
LOGC [PRES={0|1}] [TO={SYS|RSD|RSD1}];
```

### Parameters

PRES, TO                      As described in the paragraph *Principal Parameters*.  
See the standard default values in Table 6-1 *GCOS File Domain Default Values*.

### Rules

The SYS.LOGC file may be on any disk.

### EXAMPLES:

```
LOGC PRES=1 TO=RSD;
```

```
LOGC 1;
```

□





## The MIRLOG Command (if applicable)

### Function

Specifies the location of the SITE.MIRLOG file and checks that it is not allocated on a MIRROR volume.

### Format

```
MIRLOG [PRES=0 | 1] [TO={SYS | RSD | RSD1 | NRD | NRD1}];
```

### Parameters

PRES, TO                      As described in the paragraph *Principal Parameters*.  
See the standard default value in Table 6-1 *GCOS File Domain Default Values*.

### Rules

The SITE.MIRLOG file may be allocated on any FBO volume.  
It may not be allocated on a MIRROR volume.

### EXAMPLES:

```
MIRLOG PRES=1 TO=RSD1;
```

```
MIRLOG 1;
```

□



## The PVMF Command

### Function

Specifies the location and size of the SYS.PVMF file and checks that it is not allocated on a mirrored disk.

### Format

FSA:

```
PVMF [TO={SYS|RSD|RSD1|NRD|NRD1}] [SIZE=digit8];
```

### Parameters

TO	As described in the paragraph <i>Principal Parameters</i> . See the standard default value in Table 6-1 <i>GCOS File Domain Default Values</i> .
SIZE	As described in the paragraph <i>Principal Parameters</i> . See the standard default value in Table 6-1 <i>GCOS File Domain Default Values</i> . See also Calculating Backing-Store File Sizes, in the appendix <i>System Files</i> . The maximum size cannot exceed 65535 blocks.

### EXAMPLES:

```
PVMF TO=SYS SIZE=10000;
```

```
PVMF SIZE=3400;
```

```
PVMF NRD1;
```

□





---

## The QMBACK Command

### Function

To specify the location of the SYS.QM\_BACKUP file.

### Format

```
QMBACK [TO={SYS|RSD|RSD1|NRD|NRD1}] [SIZE=digit 5];
```

### Parameters

TO As described in the paragraph *Principal Parameters*.  
See the standard default value in Table 6-1 *GCOS File Domain Default Values*.

### EXAMPLES:

```
QMBACK TO=RSD, SIZE=400;
```

```
QMBACK SYS;
```

□



## The SITHELP Command

### Function

To specify the location and size of the SITE.HELP file.

### Format

FSA:

```
SITHELP [PRES={0|1}] [TO={SYS|RSD|RSD1}] [SIZE=digit8];
```

### Parameters

PRES, TO, SIZE                      As described in the paragraph *Principal Parameters*.  
See the standard default values in Table 6-1 *GCOS File Domain Default Values*.

### Rules

The SITE.HELP file may be located on any disk.

If PRES = 1 and CATFL PRES = 0, the CATVOL command must be used to define where the SITE.HELP file is to be cataloged.

### EXAMPLES:

```
SITHELP TO=SYS SIZE=300;
```

```
SITHELP SIZE=510;
```

```
SITHELP RSD1;
```

□



## The SITIN Command

### Function

To require or prevent creation of the SITE.IN file and to specify its location and size.

### Format

FSA:

```
SITIN [PRES={0|1}] [TO={SYS|RSD|RSD1}] [SIZE=digit8] ;
```

### Parameters

PRES, TO, SIZE                      As described in the paragraph *Principal Parameters*.  
See the standard default values in Table 6-1 *GCOS File Domain Default Values*.

### Rules

The SITE.IN file may be on any disk. If PRES = 1 and CATFL PRES = 0, the CATVOL command must be used to define where SITE.IN is be cataloged.

### EXAMPLES:

```
SITIN PRES=1 TO=RSD SIZE=500;
```

```
SITIN 1 SYS;
```

```
SITIN 1 SIZE=850;
```

□



## The SPDUMP Command

### Function

To specify the location of the SYS.SPDUMP file.

### Format

```
SPDUMP [TO={SYS|RSD|RSD1|NRD|NRD1}];
```

### Parameters

TO As described in the paragraph *Principal Parameters*.  
See the standard default values in Table 6-1 *GCOS File Domain Default Values*.

### Rules

The SYS.SPDUMP file may be allocated on any disk.

### EXAMPLES:

```
SPDUMP TO=RSD;
```

```
SPDUMP NRD1;
```

□



## The SPOOL Command

### Function

To specify the location and size of the SYS.SPOOL0 file.

### Format

FSA:

```
SPOOL [TO={SYS|RSD|RSD1}] [SIZE=digit8];
```

### Parameters

TO, SIZE                      As described in the paragraph *Principal Parameters*.  
See the standard default values in Table 6-1 *GCOS File Domain Default Values*.

### Rules

The SYS.SPOOL0 file may be on any resident disk.

### EXAMPLES:

```
SPOOL TO=RSD SIZE=2000;
```

```
SPOOL RSD SIZE=1275;
```

□





## The STUP Command

### Function

To require or prevent creation of the SITE.STARTUP file and to specify its location and size.

### Format

FSA:

```
STUP [PRES={0|1}] [TO={SYS|RSD|RSD1}] [SIZE=digit8];
```

### Parameters

PRES, TO, SIZE                      As described in the paragraph *Principal Parameters*.  
See the standard default values in Table 6-1 *GCOS File Domain Default Values*.

### Rules

The SITE.STARTUP file must be on a resident disk. If PRES = 1 and CATFL PRES = 0, the CATVOL command must be used to define where SITE.STARTUP is to be cataloged.

### EXAMPLES:

```
STUP PRES=1 TO=SYS SIZE=200;
```

```
STUP 1 RSD;
```

```
STUP PRES=1 SIZE=680;
```

□





## The SYIN Command

### Function

To specify certain allocation parameters for the SYS.IN file.

### Format

```
SYIN [SIZE=digit8] [INCRSIZE=digit5] [MAXSIZE=digit8];
```

### Parameters

SIZE	described in the paragraph <i>Principal Parameters</i> .
INCRSIZE	defines the increment size, in blocks, of the SYS.IN file.
MAXSIZE	defines the maximum size, in blocks, of the SYS.IN file.

See the standard default values in Table 6-1 *GCOS File Domain Default Values*.

### EXAMPLES:

```
SYIN SIZE=1000 INCRSIZE=100 MAXSIZE=2000;
```

```
SYIN 680,170,1700;
```

```
SYIN MAXSIZE=5000;
```

□



## The SYO Command

### Function

To require or prevent creation of the SYS.OUT file and to specify its location, size and some other allocation parameters.

### Format

FSA:

```
SYO [PRES={0|1}] [TO={SYS|RSD|RSD1}] [SIZE=digit8]
    [SPLIT=(name:size [,name:size])] [INCRSIZE=digit5]
    [MAXSIZE=digit8] [BLKSZ=digit5] [RECSIZE=digit5]
    [LTKSZ=digit5];
```

### Parameters

PRES, TO, SIZE	As described in the paragraph <i>Principal Parameters</i> .  See the standard default values in Table 6-1 <i>GCOS File Domain Default Values</i> .
SPLIT	may be used to define a maximum of two extra locations for the SYS.OUT file, provided that all the disks concerned have the same device class, and where:  name = {SYS RSD RSD1}  size = the size of the extent to be allocated to SYS.OUT on the volume concerned.
INCRSIZE	defines the increment size in cylinders or blocks as appropriate.
MAXSIZE	defines the maximum file size in cylinders or blocks as appropriate.
BLKSZ	defines the block size for the file in bytes.
RECSIZE	defines the record size for the file in bytes.
LTKSZ	defines the logical track size for the file as a number of blocks.

See the standard default values in Table 6-1 *GCOS File Domain Default Values*.

**Rules**

The SYS.OUT file must be on a resident disk.

**NOTE:**

SYS.OUT is a mandatory file. You must create it if TAILOR does not do it.

**EXAMPLES:**

```
SYO PRES=1 TO=RSD SIZE=30000 INCRSIZE=2 MAXSIZE=50000;
```

```
SYO 1;
```

```
SYO PRES=1 MAXSIZE=60000;
```

```
SYO PRES=1 TO=SYS SPLIT=(RSD:30000, RSD1:30000);
```

```
SYO PRES=1;
```

□



## The SYSDUMP Command

### Function

To specify the location and size of the SYS.SYSDUMP file.

### Format

FSA:

```
SYSDUMP [TO={SYS|RSD|RSD1}] [SIZE=digit8];
```

### Parameters

TO, SIZE

As described in the chapter *Tailor*.

See the standard default values in the table *GCOS File Domain Default Values* in the chapter *Tailor*.

The maximum size cannot exceed 65535 blocks.

### Rules

The SYS.SYSDUMP file may be on any resident disk.

### EXAMPLES:

```
SYSDUMP TO=RSD SIZE=20000;
```

```
SYSDUMP RSD;
```

```
SYSDUMP SIZE=10200;
```

□



## The TVMF Command

### Function

Specifies the location and size of the SYS.TVMF file and checks that it is not allocated on a mirrored disk.

### Format

FSA:

```
TVMF [TO={SYS|RSD|RSD1|NRD|NRD1}] [SIZE=digit8];
```

### Parameters

TO	As described in the paragraph <i>Principal Parameters</i> . See the standard default value in Table 6-1 <i>GCOS File Domain Default Values</i> .
SIZE	As described in the paragraph <i>Principal Parameters</i> . See the standard default value in Table 6-1 <i>GCOS File Domain Default Values</i> . See also Calculating Backing-Store File Sizes, in the appendix System Files. The maximum size cannot exceed 65535 blocks.

### EXAMPLES:

```
TVMF TO=NRD SIZE=5000;
```

```
TVMF SIZE=10000;
```

```
TVMF RSD1;
```





## 6.7 The DSA Domain

This Domain, the specific prefix of which is "D", contains the definition commands associated with the DSA set of files.

The DSA file Domain may be accessed by issuing the TLRGEN command, with FILESET = DSA.

In addition to the five disk definition commands covered earlier, this Domain contains two commands for defining the characteristics of the frontend processor files: generation and operation files. For full details of the files associated with the DSA Domain, refer to the chapter *DSA Installation and Updating Facility*. These two file definition commands are:

- DSALOG
- DSAFILES

The EOPEN command, detailed in this chapter, is also applicable.

### 6.7.1 Principal Parameters

The DSA file definition commands both adhere to the same structure. This structure uses two main parameters, the descriptions of which are as follows:

PRES	is used to require or prevent the creation of the group of files concerned by the command.
TO	is used to specify the location of the group of files concerned. Any disk referenced through this parameter must have been previously defined through the appropriate disk definition command.

The other parameters used are described with each command.





## 6.7.2 File Definition Commands

### The DSALOG Command

#### Function

To require or prevent creation of the SYS.DSALOG1 file and to specify its location and size, when access to the DSA network is to be made via a CNP7 or Datanet frontend processor.

#### Format

FSA:

```
DSALOG [PRES={0|1}] [TO={SYS|RSD|RSD1}] [SIZE=digit8];
```

#### Parameters

PRES, TO	As described in Paragraph 6.7.1.  See the standard default values in Table 6-2 <i>DSA File Domain Default Values</i> .
SIZE	See the standard default values in Table 6-2 <i>DSA File Domain Default Values</i> .

#### Rules

The SYS.DSALOG1 file may be located on any disk.

#### EXAMPLES:

```
DSALOG PRES=1 TO=RSD SIZE=500;
```

```
DSALOG 1 SIZE=800;
```

□



## The DSAFILES Command

### Function

To require or prevent creation of the frontend processor files and specify their location; these are:

```
SYS.DSACONF
SYS.DSACORE
SYS.DSADUMP
SYS.DSABLIB
SYS.DSALMLIB
SYS.DSASLIB
SYS.DSACMD
SYS.DSATRCIDT_* (for FCP7 only)
```

In addition, this command specifies file size for:

```
SYS.DSACORE
SYS.DSADUMP
```

For creation of the FCP7-specific files, SYS.DSATRC\_<LCT-name>\_x, see the paragraph *The BUILD\_TRCNCC Function* in the chapter *DSA Installation and Updating Facility*.

### Format

```
DSAFILES [PRES={0|1}] [TO={SYS|RSD|RSD1}] [CORESZ=digit8]
          [DUMPSZ=digit8];
```

### Parameters

PRES, TO	As described in Paragraph 6.7.1.
CORESZ	is used to specify the size of the SYS.DSACORE file.
DUMPSZ	is used to specify the size of the SYS.DSADUMP file.

See the standard default values in Table 6-2 *DSA File Domain Default Values*.



### Rules

SYS.DSA\* files are allocated on resident FBO disks.

### EXAMPLES:

```
DSAFILES PRES=1 TO=RSD;
```

```
DSAFILES PRES=1 CORESZ=400 DUMPSZ=1000;
```

```
DSAFILES 1 CORESZ=500 DUMPSZ=5000;
```

□



---

## 6.8 The OLTD Domain

This Domain, the specific prefix of which is "O", contains only the commands SYSVOL (see Paragraph 6.5) and EOGEN (see Paragraph 6.11), because OLTD files have no selectable features.

The OLTD Domain is accessed using the TLRGEN command, with FILESET = OLTD.



## 6.9 The FW Domain

This Domain, the specific prefix of which is "F", contains the definition commands associated with the linked queued files SYS.FW.\* which generate the Firmware. The Domain is accessed using the TLRGEN command with FILESET = FW.

In addition to the five disk definition commands already dealt with, this Domain contains the command FWGEN, for defining the location of the group of Firmware generation files.

EOGEN is also applicable. (See Paragraph 6.11.)

### The FWGEN Command

#### Function

To specify the location of the Firmware generation files.

#### Format

```
FWGEN TO=SYS;
```

#### Parameters

TO Defines the location of the Firmware generation files. The values it may take are the symbolic disk names (see disk definition commands). Any disk referenced through this parameter must have been previously defined through the appropriate disk definition command.

See the standard default value in Table 6-3 *FW Domain Default Values*.

#### Rules

Firmware generation files are allocated on the system disk.

#### EXAMPLES:

```
FWGEN TO=SYS;
```

```
FWGEN SYS;
```

□



## 6.10 The GSF Domain

This Domain, the specific prefix of which is "M", contains the definition commands associated with the GSF files. This Domain may be accessed using the TLRGEN command, with FILESET = GSF.

In addition to the first three disk definition commands (SYSVOL, RSDVOL and RSDVOL1) dealt with in paragraph 6.5, this Domain contains the command GSFGEN for defining the location of the group of GSF files.

EOGEN is also applicable. (See Paragraph 6.11.)

### The GSFGEN Command

#### Function

To specify the location of the GCOS 7 Service Facility files.

#### Format

```
GSFGEN TO=SYS;
```

#### Parameters

TO defines the location of the GCOS 7 Service Facility files. The values it may take are the symbolic disk names (see disk definition commands). Any disk referenced through this parameter must have been previously defined through the appropriate disk definition command.

See the standard default value in Table 6-4 *GSF Domain Default Values*.

#### Rules

The GSF files are allocated on the system disk.

#### EXAMPLES:

```
GSFGEN TO=SYS;
```

```
GSFGEN SYS;
```

□



## 6.11 The EOGEN Command

This command is available in the five GCL Domains GCOS 7, DSA, OLTD, FW, GSF, and is used to exit a Domain, saving the result of the associated TLRGEN session.

### Rules

- The EOGEN command is mandatory in a COMFILE, to end a TLRGEN session.
- When commands are issued from the user console, the "/" character may be used instead of the EOGEN command to exit from a GCL Domain. In this case you will be prompted to save the result of the current TLRGEN session.



## 6.12 TLRGEN Session Considerations

### 6.12.1 Summary

As we have seen in this chapter, a TLRGEN session consists of a sequence of commands in any one of the five GCL Domains, GCOS, DSA, OLTD, FW or GSF. The purpose of a TLRGEN session is to define file characteristics for a given file Domain.

A TLRGEN session starts with the TLRGEN command, and ends with an EOPEN command.

The result of a TLRGEN session is a set of values constituting the "latest" COMSET. For a given Domain, the "latest" COMSET is a merge between the previously-active set of default values (which may be itself a "latest" COMSET) and those which you have just defined.

### 6.12.2 Validation of a TLRGEN Session

#### Rules

- To validate a TLRGEN session, all the disks referenced (through the TO parameter) in file definition commands must be defined when activating the EOPEN command.
- The system disk must be defined when ending a TLRGEN session for the GCOS or OLTD Domain, even if it has not been referenced in the file definition commands.
- In batch mode, TAILOR aborts if a TLRGEN session is not validated.
- In interactive mode (from an IOF terminal or the main console), you will be prompted until you validate the TLRGEN session. However, you can leave the TLRGEN session without saving by using the "/" character.

#### NOTE:

When a TLRGEN session is validated, a TLRPRINT command is activated, by default for the current file Domain.





### 6.12.3 Default TLRGEN Sessions

If your DPS 7000 system is delivered with a P-Set, you can use a default value set, "Last COMSET", to build a Production-Set identical to the delivered P-Set, by defining only the characteristics of the system disk.

This enables a "default" TLRGEN session to be run. This procedure should be used whenever possible.

If the system files for all the Domains (GCOS, FW, OLTD, DSA and GSF) are created with their default SIZES, the space required will exceed the 500 Mbytes capacity of the single system disk on which TAILOR tries to locate them by default.

Thus, the user must define another resident disk using the TAILOR command RSDVOL and allocate certain system files on this disk.

For example, files belonging to the DSA Domain and the GCOS Domain files SYS.SYSTEM, SYS.LIB, SYS.PVMF and SYS.TVMF could suitably be created on this resident disk as follows:

```
TLRGEN  GCOS INITIAL;  
SYSVOL  sysmd sysdvc;  
RSDVOL  rsdmd rsddvc;  
ISL     TO=RSD;  
LIB     TO=RSD;  
PVMF    TO=RSD;  
TVMF    TO=RSD;  
EOGEN;  
TLRGEN  DSA INITIAL;  
RSDVOL  rsdmd rsddvc;  
DSAFILES TO=RSD;  
DSALOG  TO=RSD;  
EOGEN;
```

---



Once the system disk has been defined through the SYSVOL command, during a TLRGEN session for a given files Domain, "latest" COMSETs may be built for the other file Domains (for those that do not already have one), by merging the system disk definition with the standard default value sets.

This process should be used whenever possible.

**NOTE:**

After TLRRESET WEAK, if the only disk referenced in the latest COMSETs is the system disk, it is also possible to run a default TLRGEN session once the system disk has been defined. This is done by merging the incomplete latest COMSETs, produced by TLRRESET, with the system disk definition.

See 6.15.2 for an example of a default TLRGEN session.



## 6.13 Context Rules

### 6.13.1 TAILOR and IUF

#### **TAILOR CALLED BY GIUF FUNCTIONS**

In an IUF context, the TAILOR processor may be called by several functions:

- BUILDP, to create or complete your P-Sets. According to the IUF configuration, the input Set will be:
  - The P2-Set, in a P2P configuration.
  - The R-Set, in an RP configuration
- BUILDP2, to create or complete a P2-Set from a P-Set. The First BUILDP2 activation modifies the IUF configuration from RP to P2P.

#### **CREATING OR UPDATING PRODUCTION SETS**

When TAILOR is called by IUF functions, there are two possible situations:

- You are creating a Production Set:

From TAILOR's point of view, this is equivalent to a TLREXEC command for the GCOS file Domain. The resulting production Set must reflect the content of the input Set. For this reason, a check is made, before starting file creation, that TLREXEC commands have been issued for all the file Domains available on the input Set.

If such commands are missing, an abort will occur when TAILOR commands are read from a COMFILE. If commands are given either at the main console or at an IOF terminal, it will be impossible to interrupt the dialogue as long as TLREXEC commands are missing.
- You are completing an existing Production Set:

This might be as a consequence of an add-on. From TAILOR's point of view, no TLREXEC command for the GCOS Domain has been activated, so no check is made.



### 6.13.2 RESTART After Abort During File Creation

#### ORDER OF CREATION

Note that the file Domains are always created by TAILOR in the same order:

1. GCOS Domain
2. OLTD Domain
3. FW Domain
4. DSA Domain
5. GSF Domain.

The TLREXEC commands for the five file Domains may be given in any order, but in the event of restart after an abort, it is necessary to reissue the TLREXEC commands for the file Domains that have not yet been created. For example, to create a Production Set containing the GCOS, OLTD, DSA, GSF and FW file Domains, an appropriate TAILOR command sequence might be:

```
TLREXEC GCOS;  
TLREXEC DSA;  
TLREXEC OLTD;  
TLREXEC FW;  
TLREXEC GSF;
```

If an abort occurred during OLTD file creation at label O5, the sequence of TLREXEC commands in TAILOR necessary to restart file creation would be:

```
TLREXEC OLTD FROM=O5;      (to restart OLTD file creation)  
  
TLREXEC DSA;              (the DSA, GSF and FW files  
TLREXEC FW;               had not yet been created  
TLREXEC GSF;              at the time of the abort)
```

#### CONTROL OF FILE LOCATION

Before restarting file creation after an abort, a check is made to verify that no file location has been changed as compared with the initial launch; changing location means changing the value of the PRES or TO parameters in the command associated with a given file.

If such changes are detected, RESTART is not performed until the old locations have been restored.



## 6.14 GCL Rules

You will find it helpful when using TAILOR and you need to issue commands from a console (MAIN or IOF), to keep in mind some specific GCL rules:

- The "/" character is used to notify the end of the current action. It may be used to leave prompting mode for a given command, or to leave a GCL Domain.
- The "?" character is used to require help, and may be used at Domain level, at command level, or at keyword level.
- Prompting mode is entered if you make syntax errors, or if you give incomplete commands (for example, command name followed only by a semi-colon).



## 6.15 Examples

### 6.15.1 TLRGEN Session

The following example represents, for each Domain:

1. the TAILOR commands, for batch processing mode
2. the output that appears on the terminal screen.

#### GCOS Domain

1. TAILOR Commands

```
-----  
T: TLRRESET STRONG;  
>>> SUCCESSFUL RESET  
T: TLRGEN FILESET=GCOS;  
G: SYSVOL MD=FSD02 DVC=MS/FSA;  
G: RSDVOL MD=FSD03 DVC=MS/FSA;  
G: NRDVOL MD=FSD04 DVC=MS/FSA;  
G: SYO SPLIT=(RSD:3);  
G: JBEFORE PRES=1;  
G: BKST1 SYS SIZE=17000 TO=RSD;  
G: MIRLOG 1 NRD;  
G: HALOCK 1 SYS;  
G: EOPEN;  
-----
```

#### NOTE:

The commands actually submitted are as shown above (G: is the GCL prompt). These are followed by:

```
PRINT OF TL_GCOS DOMAIN STATUS
```

The TLRPRINT command does not have to be submitted. It is implicitly activated at the end of a TLRGEN session. The result is as follows:



## 2. Output

## VOLUME STATUS

COMNAME	MEDIA	DEVCLAS	ORG	MIRROR
RSDVOL	FSD03	MS/FSA	FSA	N
RSDVOL1	UNDEF	UNDEF		
NRDVOL	FSD04	MS/FSA	FSA	N
NRDVOL1	UNDEF	UNDEF		
SYSVOL	FSD02	MS/FSA	FSA	N
CATVOL	UNDEF	UNDEF		

## COPIED FILES STATUS

COMNAME	FILENAME	LOADED	ON
ISL	SYS.SYSTEM	YES	SYS

## PREALLOCATED FILES STATUS

COMNAME	FILENAME	CREATED	ON	SIZE	INCSZ	MAXSZ	BLKSZ	RECSZ	LT SZ
CATFL	SITE.CATALOG	YES	SYS	1940 BLK					
SITIN	SITE.IN	NO							
STUP	SITE.STARTUP	YES	SYS	170 BLK					
BKST	SYS.BKST	YES	SYS						
BKST1	SYS.BKST1	YES	RSD1	17000 BLK					
ERLOG	SYS.ERLOG	YES	SYS	85 BLK					
SYIN	SYS.IN	YES	SYS	170 BLK	170	680			
JBEFORE	SYS.JRNAL	YES	SYS	16368 BLK					
KNODET	SYS.KNODET	YES	SYS	85 BLK					
LOGC	SYS.LOGC	YES	SYS	240 BLK					
SYO	SYS.OUT	YES	SYS	340 BLK					
			RSD	8 cyl	85	32000	8166	4024	5
SPOOL	SYS.SPOOL0	YES	SYS	510 BLK	2	600	8116	4024	5
SYSDUMP	SYS.SYSDUMP	YES	SYS	11730 BLK					
SWLOG	SYS.SWLOG	YES	SYS						
LIB	SYS.LIB	YES	SYS	11900 BLK					
PVMF	SYS.PVMF	YES	SYS	5100 BLK					
TVMF	SYS.TVMF	YES	SYS	2550 BLK					
SITHELP	SITE.HELP	YES	SYS	170 BLK					
QM	SYS.QM	YES	SYS	500 CI					
QMBACK	SYS.QM_BACKUP	YES	SYS	500 CI					
MIRLOG	SITE.MIRLOG	YES	NRD						
JAS	SYS.JASBLUE	NO							
JAS	SYS.JASGREEN	NO							
CLC	SYS.CLC	NO							
CMS	SITE.CMS-CSD	NO							
CMS	SITE.CMS-CDD-name1	NO							
CMS	SITE.CMS-CDD-name2	NO							
CMS	SITE.CMS-DS-name1-name2	NO							
HALOCK	SITE.HALOCK	YES	SYS						
SPDUMP	SYS.SPDUMP	YES	SYS						

UNITS OF SIZE, INCSZ, MAXSZ ARE THE SAME.  
 >>>COMPLETED TLRGEN SESSION.



**GSF Domain**

1. TAILOR Commands

```
T: TLRGEN GSF;
M: SYSVOL FSD02 MS/FSA;
M: EOPEN;
```

2. Output

**VOLUMES STATUS**

COMNAME	MEDIA	DEVCLASS	ORG	MIRROR
SYSVOL	FSD02	MS/FSA	FSA	N

**COPIED FILES STATUS**

COMNAME	FILENAME	LOADED	ON
GSFGEN	GSFGEN_FILES	YES	SYS

**OLTD Domain**

1. TAILOR Commands

```
T: TLRGEN OLTD;
O: EOPEN;
```





## 2. Output

**VOLUMES STATUS**

<b>COMNAME</b>	<b>MEDIA</b>	<b>DEVCLASS</b>	<b>ORG</b>	<b>MIRROR</b>
SYSVOL	FSD02	MS/FSA	FSA	N

**FW Domain**

## 1. TAILOR Commands

---

T: TLRGEN FW;  
F: SYSVOL FSD02 MS/FSA;  
F: EOGEN;

---

## 2. Output

**VOLUME STATUS**

<b>COMNAME</b>	<b>MEDIA</b>	<b>DEVCLASS</b>	<b>ORG</b>	<b>MIRROR</b>
SYSVOL	FSD02	MS/FSA	FSA	N

**COPIED FILES STATUS**

<b>COMNAME</b>	<b>FILENAME</b>	<b>LOADED</b>	<b>ON</b>
FWGEN	FWGEN_FILES	YES	SYS



**DSA Domain**

1. TAILOR Commands

```
T: TLRGEN DSA;
D: SYSVOL FSD02 MS/FSA;
D: EOPEN;
```

2. Output

**VOLUME STATUS**

COMNAME	MEDIA	DEVCLASS	ORG	MIRROR
SYSVOL	FSD02	MS/FSA	FSA	N

**COPIED FILES STATUS**

COMNAME	FILENAME	LOADED	ON
DSAFILES	SYS.DSA* FILES	YES	SYS

**PREALLOCATED FILES STATUS**

COMNAME	FILENAME	CREATED	ON	SIZE	INCS	MAX SZ	BLK SZ	REC SZ	LT SZ
DSAFILE	SYS.DSACONF	YES	SYS	340 BLK					
DSAFILE	SYS.DSACORE	YES	SYS	340 BLK					
DSAFILE	SYS.DSADUMP	YES	SYS	1360 BLK					
DSALOG	SYS.DSALOG1	YES	SYS	680 BLK					

UNITS OF SIZE, INCSZ, MAXSZ ARE THE SAME.



### 6.15.2 Default TLRGEN Session

This example shows the status of the "latest" COMSETs after the TLRGEN session described in the first example. At the end of the TLRGEN session performed for the GCOS files Domain, "latest" COMSETs have been constituted by merging the SYSVOL definition with the initial COMSETs for the DSA, OLTD, GSF and FW file Domains.

```
T: TLRSTAT;
```

```
TAILOR PROCESSING STATUS :
```

```
FILESET    EXECUTABLE    TO BE BUILT
```

GCOS	YES	NO	Default TLRGEN sessions for FW, OLTD, GSF and DSA Domains have been performed which allow TLREXEC activation for these Domains.
FW	YES	NO	
OLTD	YES	NO	
DSA	YES	NO	
GSF	YES	NO	



### 6.15.3 TLREXEC Command Sequences

This example describes two sequences of TLREXEC commands: the first launching the creation of several Domains, the second restarting creation after an abort. It is assumed that a TLRGEN session has been performed for the Domains to be created (see the above examples).

---

```
T: TLREXEC GCOS;
>>> TAILOR READY TO BUILD GCOS DOMAIN FILES
T: TLREXEC DSA;
>>> TAILOR READY TO BUILD DSA DOMAIN FILES
T: TLREXEC FW;
>>> TAILOR READY TO BUILD FW DOMAIN FILES
T: TLREXEC OLTD;
>>> TAILOR READY TO BUILD OLTD DOMAIN FILES
T: TLREXEC GSF;
>>> TAILOR READY TO BUILD GSF DOMAIN FILES
T: TLRSTAT;
```

TAILOR PROCESSING STATUS :

FILESET	EXECUTABLE	TO BE BUILT
GCOS	YES	YES
FW	YES	YES
OLTD	YES	YES
DSA	YES	YES
GSF	YES	YES

---



Suppose that an abort now occurs during GCOS 7 file creation at label G15. After correcting any errors, by performing a TLRGEN session with FILESET = GCOS and COMSET = LAST, an appropriate sequence for restarting creation would be as follows:

```
-----  
T: TLREXEC GCOS FROM=G15;  
>>> TAILOR READY TO BUILD GCOS      DOMAIN FILES  
T: TLREXEC DSA;  
>>> TAILOR READY TO BUILD DSA      DOMAIN FILES  
T: TLREXEC OLTD;  
>>> TAILOR READY TO BUILD OLTD     DOMAIN FILES  
T: TLREXEC FW;  
>>> TAILOR READY TO BUILD FW      DOMAIN FILES  
T: TLREXEC GSF;  
>>> TAILOR READY TO BUILD GSF     DOMAIN FILES  
-----
```

**NOTE:**

TLREXEC commands are mandatory for the FW, DSA, GSF and OLTD file Domains, because these have not yet been created.



#### 6.15.4 Dialogue at the MAIN Console

The following gives an example of the dialogue when commands are issued from the main console. In this case, the answers to prompts are given in deferred mode, preceded by the answer number.

```
-----  
>>>12.37 TAILOR 11.01  
01/T:  
S:01 TLRRESET STRONG;  
--->>> SUCCESSFUL RESET  
02/T:  
S:02 TLRGEN GCOS INITIAL;  
03/G:  
S:03 SYSVOL K111 MS/FSA;  
---> ?  
---> SYSVOL MUST BE IN FSA ORGANIZATION  
04/MEDIA  
S:04 K333  
05/DVC  
S:05 MS/FSA  
06/G:  
S:06 RSDVOL K222;  
07/G:  
S:07 STUP PRES=1 TO=NRD;  
---> ?  
---> LOCATION MUST BE A RESIDENT DISK  
08/TO LOCATION (NRD) :  
S:08 RSD  
09/G:  
S:09 JBEFORE;  
--->JBEFORE  
10/PRES REQUIRED CREATION (BOOLEAN VALUE) (0) :  
S:10 1  
11/TO LOCATION (SYS) :  
S:11  
12/SIZE SIZE IN CYLINDERS (10) :  
S:12 12  
13/FILEBLOCK_SIZE FILE BLOCK SIZE IN BYTES (512 or 4096) (512):  
S:13  
14/G:  
S:14 EOPEN;  
-----
```

Then, as in interactive mode, an implicit TLRPRINT command is activated.



### 6.15.5 Examples of Errors

The following examples describe some error sequences with the corresponding error messages.

#### **VOLUME REFERENCED BUT UNDEFINED**

This example describes the case where an undefined disk is referenced in a file definition command. In this case, the command is aborted.

---

```
T: TLRRESET STRONG;
>>> SUCCESSFUL RESET
T: TLRGEN GCOS INITIAL;
G: SYSVOL FSA111 MS/FSA;
G: ERLOG PRES=1 TO=RSD;
*** RSDVOL REFERENCED AND NOT DEFINED
>>> ABORTED TLRGEN SESSION (only in batch mode)
>>> ABORTED TAILOR (only in batch mode)
```

---

#### **NO LATEST COMSET EXISTS**

After a TLRRESET STRONG command, all "latest" COMSETs are cleared and thus unavailable for use in a TLRGEN session.

---

```
T: TLRRESET STRONG;
>>> SUCCESSFUL RESET
T: TLRGEN GCOS COMSET=LAST;
***NO EXISTING LAST FOR GCOS FILESET
>>> ABORTED TLRGEN SESSION (only in batch mode)
>>> ABORTED TAILOR (only in batch mode)
```

---



### ERRONEOUS VALUE FOR SPLIT

This example describes the case where the SYS.OUT file is located on several disks and the same volume name is specified for each location:

```
-----  
T: TLRRESET STRONG;  
>>> SUCCESSFUL RESET  
T: TLRGEN GCOS INITIAL;  
G: SYSVOL FSA1 MS/FSA;  
G: SYO TO=SYS SPLIT=(SYS:510);  
*** INVALID VALUE FOR SPLIT: SYSVOL CANNOT BE  
    REFERENCED SEVERAL TIMES  
>>> ABORTED TLRGEN SESSION  
>>> ABORTED TAILOR  
-----
```





## 6.16 TAILOR Default Values and File Size Computation

### 6.16.1 GCOS File Domain

Table 6-1. GCOS File Domain Default Values (1/4)

Command	Parameter	Default Value MS/FSA
BKST	SIZE	15980
BKST1	PRES	0
	TO	SYS
	SIZE	4250
CATFL	PRES	1
	TO	SYS
	NBOBJ	1940
CLC	PRES	0
	TO	SYS
CMS	PRES	0
	TO	RSD
	NAME1	XXXX
	NAME2	YYYY
ERLOG	PRES	1
	TO	SYS
HALOCK	PRES	0
	TO	RSD
ISL	TO	SYS



**Table 6-1. GCOS File Domain Default Values (2/4)**

<b>Command</b>	<b>Parameter</b>	<b>Default Value MS/FSA</b>
JAFTER	PRES	0
	TO	SYS
	SIZE	10
	JOURNAL_SIZE	80
	BLOCK_SIZE	8192
	PRIMARY_MEDIA	NIL
	PRIMARY_DVC	NIL
	SECONDARY_MEDIA	NIL
	SECONDARY_DVC	NIL
	AUTOCYCLE	1
	SEDCOPY	0
JAS	PRES	0
	TO	SYS
JBEFORE	PRES	1
	TO	SYS
	SIZE	16368
	FILEBLOCK_SIZE	512
KNODET	TO	SYS
	SIZE	85
LIB	TO	SYS
	SIZE	11900
LOGC	PRES	1
	TO	SYS

**Table 6-1. GCOS File Domain Default Values (3/4)**

<b>Command</b>	<b>Parameter</b>	<b>Default Value MS/FSA</b>
MIRLOG	PRES	0
	TO	SYS
PVMF	TO	SYS
	SIZE	5100
QM	TO	SYS
	SIZE	500
QMBACK	TO	SYS
	SIZE	500
SITHELP	PRES	1
	TO	SYS
	SIZE	170
SITIN	PRES	0
	TO	SYS
	SIZE	170
SPDUMP	TO	SYS
SPOOL	TO	SYS
	SIZE	1360
STUP	PRES	1
	TO	SYS
	SIZE	170
SWLOG	TO	SYS



**Table 6-1. GCOS File Domain Default Values (4/4)**

<b>Command</b>	<b>Parameter</b>	<b>Default Value MS/FSA</b>
SYSIN	SIZE	170
	INCRSIZE	170
	MAXSIZE	680
SYO	PRES	1
	TO	SYS
	SIZE	340
	SPLIT	UNDEF
	INCRSIZE	85
	MAXSIZE	32000
	BLKSZ	8116
	RECSIZE	4024
	LTKSZ	5
SYSDUMP	TO	SYS
	SIZE	11730
TVMF	TO	SYS
	SIZE	2550



## 6.16.2 DSA File Domain

### STANDARD DEFAULT VALUES

**Table 6-2. DSA File Domain Default Value**

Command	Parameter	Default Value MS/FSA
DSALOG	PRES	1
	TO	SYS
	SIZE	680 blocks
DSAFILES	PRES	1
	TO	SYS
	CORESZ	340 blocks
	DUMPSZ	7200 blocks



**6.16.3 FW File Domain**

STANDARD DEFAULT VALUES

**Table 6-3. FW File Domain Default Values**

<b>Command</b>	<b>Parameter</b>	<b>Default Value MS/FSA</b>
FWGEN	TO	SYS

**6.16.4 GSF File Domain**

STANDARD DEFAULT VALUES

**Table 6-4. GSF File Domain Default Values**

<b>Command</b>	<b>Parameter</b>	<b>Default Value MS/FSA</b>
GSFGEN	TO	SYS



## 6.17 Catalog Files: Size and Number of Objects

Table 6-5. File Size and Number of Objects on MS/FSA Volume

Number of Logical Blocks in the catalog	Number of Objects
6	45 (min.)
11	100
88	1000
433	5000
866	10000
1728	20000
2590	30000
3454	40000
4316	50000
5179	60000
6042	70000
6905	80000
7766	90000
8629	100000 (max.)

### NOTES:

1. The number of blocks in a subcatalog may be obtained through the LSCAT and CATLIST requests.
2. The size of a catalog block is 4096 bytes on MS/FSA.







---

## 7. Updating Domains

### 7.1 Purpose

Updating procedures enable you to perform preventative maintenance on the following Domains:

- GCOS
- OLTD
- FW
- DSA
- GSF.

This is done by applying a new Technical Status to the Domain concerned.

For the GCOS Domain, your Bull CSE provides a **G-MEDIA**, which can be either tapes or cartridges containing the GCOS 7 system files used by both the Installation and Updating procedures. As a general rule, the Set to be updated is:

- the P-Set, in a PO-Configuration or an RP-Configuration
- the chosen Set (non-running or running) in a P2P-Configuration.

In P2P and RP-Configurations, the other Set may be updated using the IUF function LEVEL, after a probationary period of time.

The operations to be performed are described in the updating procedure relevant to each of the Domains: GCOS, OLTD, FW, DSA and GSF.

For slave system updating, refer to the chapter *Multi-System Installation and Updating*.



## 7.2 Set Technical Level

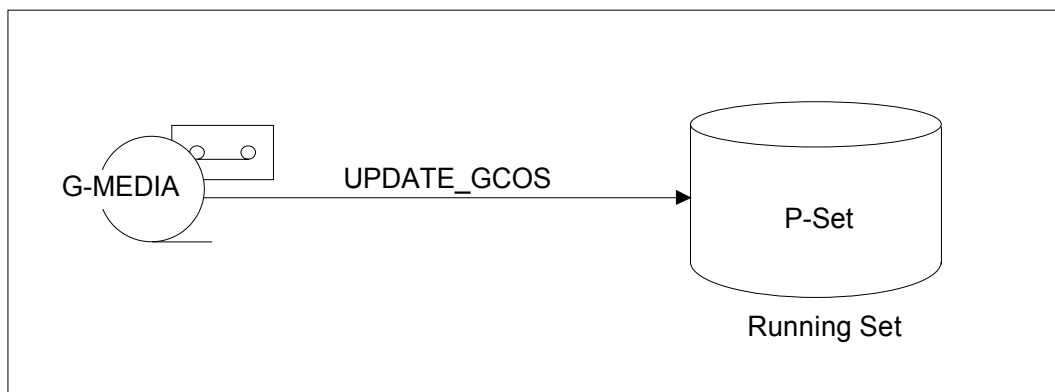
The number of its Technical Status indicates the Technical Level of a P-Set, P2-Set, or R-Set, for each Domain.

The application of a new Technical Status to a Set modifies the updated Domain's Technical Status Number (TSN).

## 7.3 Updating Strategy in a PO-Configuration

After Release installation has been completed, the only system Set is the P-Set.

The UPDATE\_GCOS function applies TS corrections to the running P-Set.



**Figure 7-1. Updating Strategy in a PO-Configuration**



## 7.4 Updating Strategy in a P2P-Configuration

After Release installation, the P-Set and P2-Set are at the same Technical Level (as a result of running the BUILD P2 function).

In a P2P-Configuration, the default value of the updated Set is the non-running Set. However, the OUTSET parameter of the IUF function UPDATE\_GCOS provides the possibility of updating the running Set.

The update function applies TS corrections to either the running or the non-running Set (P-Set or P2-Set), allowing the user to execute the updating procedure without any risk for the running Set.

After the updated Set has been loaded, the After Journal environment and the SITE.CATALOG must, as appropriate, be transferred from the old Set to the new Set.

During the probationary period of the new update, returning to the previous update is very fast, because it is only necessary to perform a warm restart on the other (reserve) Set.

When the probationary period is over, or just before applying a new update, the LEVEL function must be executed, to bring both Sets up to the same Technical Level.

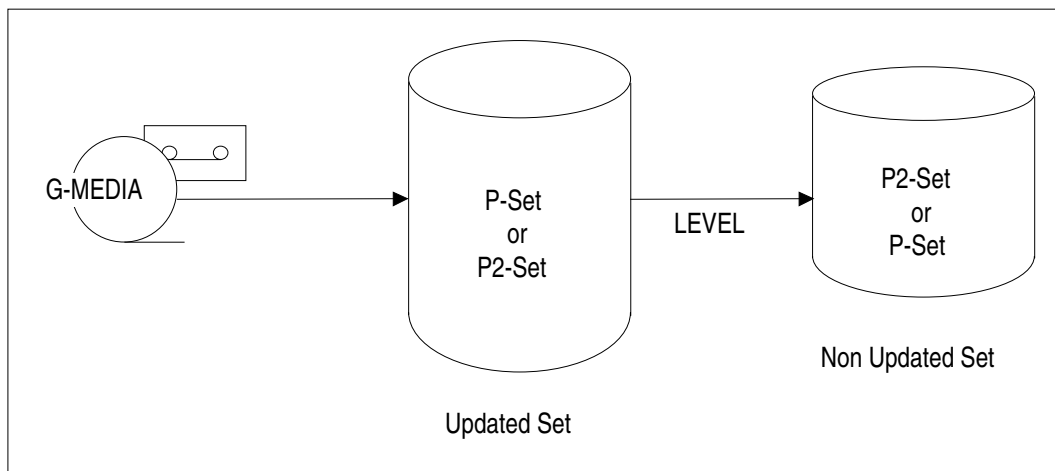


Figure 7-2. Updating Strategy in a P2P-Configuration



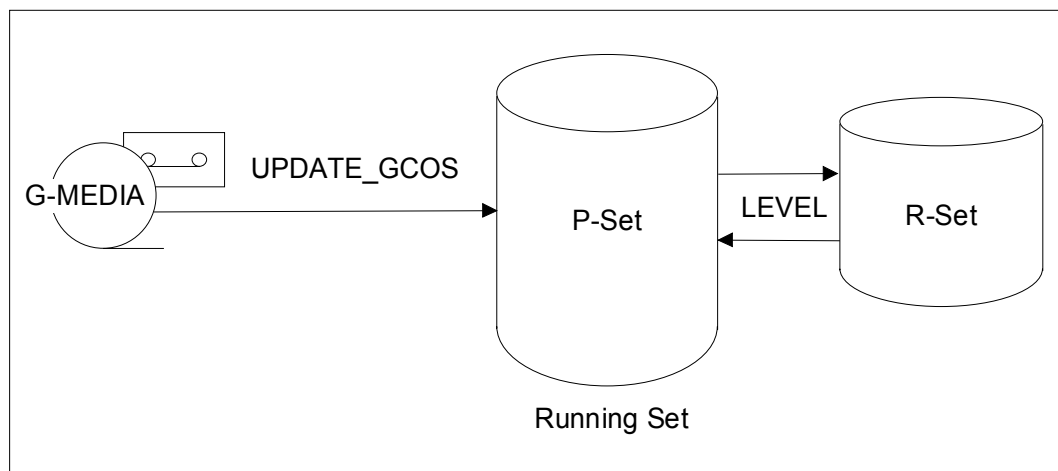
## 7.5 Updating Strategy in an RP-Configuration

After installation of a new Release, the R-Set and P-Set are at the same Technical Level (as a result of the BUILDP function).

The update function applies TS corrections to the P-Set only. After the application of a new Technical Status, the R-Set remains unchanged until the next execution of the UPDATE\_GCOS function. Refer to the chapter *Installation and Updating Facility*.

Note that it is not possible to create a new P-Set while the R-Set is at a lower Technical Level than the P-Set. The R-Set is brought up to the same Technical Level as that of the P-Set by running the LEVEL function.

The LEVEL function may also be used for back-up purposes, to bring a P-Set back to the same Technical Level as that of the R-Set, in case of a failure on the P-Set files affected by the UPDATE\_GCOS function.



**Figure 7-3. Updating Strategy in an RP-Configuration**

For Domains other than GCOS, refer respectively to:

- for FW, the chapter FGF (*Firmware Generation Facility*)
- for OLTD, the chapter *The TDF Facility (OLTD)*
- for DSA, the chapter *The DIUF Facility*
- for GSF, the chapter *The GSF Facility*.



## 7.6 GCOS 7 Updating Procedure

**Table 7-1. Files in PO-Configurations Affected by UPDATE\_GCOS**

	<b>File Replace</b>	<b>Module Replace</b>
SYS.BOOT		(1)
SYS.C.INCLUDE	*	
SYS.GPL.MACLIB	*	
SYS.HBINLIB		*
SYS.HCULIB	*	
SYS.HELP	*	
SYS.HLMLIB		*
SYS.HSLLIB		*
SYS.IUF		*
SYS.SYSTEM	*	

(1) This file is modified in a specific way.

**Table 7-2. Files in P2P & RP-Configurations Affected by UPDATE\_GCOS**

	<b>File Replace</b>	<b>Module Replace</b>
SYS.BOOT		(1)
SYS.C.INCLUDE	*	
SYS.GPL.MACLIB	*	
SYS.HBINLIB		*
SYS.HCULIB	*	
SYS.HELP	*	
SYS.HLMLIB	*	
SYS.HSLLIB	*	
SYS.IUF		*
SYS.SYSTEM	*	

(1) This file is modified in a specific way.

In P2P and RP-Configurations, both Sets must be present during execution of the UPDATE\_GCOS function.

In RP-Configurations, the UPDATE\_GCOS function performs an automatic LEVEL from the P-Set to the R-Set.

**IMPORTANT:**

UPDATE\_GCOS must be submitted under the SYSADMIN project.

**In a PO-Configuration**, in the event of the SYS.SYSTEM file becoming corrupted, there is no possibility of salvaging this file from the SYS.SYSTEM on another Set. The only solution is to take regular saves of the P-Set, and use the backup procedure outlined in the chapter *Installation Procedure*, in order to restore the SYS.SYSTEM file.

The UPDATE\_GCOS function must be performed in a monoprogramming environment when the Technical Status is applied on the running Set. There is a risk involved in performing this IUF function in a multiprogramming environment as several system files may be dynamically changed in the process.

The following functions must never be performed while TDS is active:

- UPDATE\_GCOS
- LEVEL GCOS
- LEVEL ALL
- RESTORE\_SET GCOS
- RESTORE\_SET ALL

This is because TDS assigns itself exclusive access to the file SYS.HBINLIB, which would result in aborting the UPDATE\_GCOS session (without a restart point) at the moment when it tries to update the SYS.HBINLIB library.

After application of a new Technical Status (using UPDATE\_GCOS), the RESTART(CLEAN) command should be executed before regenerating TDS.



## 7.7 GCOS 7 Updating Operations

The P-Set or the P2-Set must be loaded and running.

1. In the case of a PO Set (or, for a P2P-Configuration, the Set to be updated, either the running or the non-running Set) on line. The Set to be updated must be on line so that the system can return its current Technical Level before you apply the TS corrections.
2. In the case of an RP-Configuration, put the R-Set on line. The R-Set must be on line so that the system can return the R-Set's current Technical Level as well as to permit the UPDATE\_GCOS function to perform an automatic LEVEL from P-Set to R-Set.
3. Mount the G-MEDIA.

The G-MEDIA is either a set of tapes or cartridges containing the GCOS 7 files, or a CD-ROM (only available for XTA Systems). In this later case, the image of the GCOS 7 Technical Status, will be transferred to a work disk using the `Install_GCOS_TECHNICAL_STATUS` script located on the CD-ROM.

4. Start the IUF function `UPDATE_GCOS`.

For details of the `UPDATE_GCOS` function, refer to the chapter *The Installation and Updating Facility* (paragraph 4.2.39).

5. End the system.
6. Completely reinitialize the system, then load GCOS 7 from the updated disk Set. If ISL is redelivered, all the parameters have to be re-specified. Otherwise it is sufficient to proceed as follows:

```
I: RESTORE,RESTART(CLEAN)
```

For the ISL parameters, refer to the *GCOS 7 System Operator's Guide*.

7. In the case of a P2P-Configuration:

The After Journal environment must be transferred from the old Set to the new (updated) Set.

The After Journal environment is transferred using the procedure described in below.

The `SITE.CATALOG` may be updated, as necessary, from the old Set using the `COPY_SITEFILES` function.

8. Save the P-Set with the `SAVE_DISK` command.

You will then be able to recover your P-Set using `RESTORE` under the SIP facility in UTIL mode (see the appendix *SIP Functions*).



## 7.8 After Journal Transfer

This operation can be performed only when there is no activity on the After Journal. Its purpose is to copy the SYS.JADIR file from the old Set to the new Set which is now to be used, and to move the name path of the SYS.JADIR file from the SYS.CATALOG in the old Set to the SYS.CATALOG in the Set now to be used.

The following GCL and JCL must be executed under the project SYSADMIN:

```
S: ATTACH (CAT=SYS.CATALOG:new-set-md:new-set-dvc,
          SYS.CATALOG:old-set-md:old-set-dvc);

S: COPY_FILE      INFILE=SYS.JADIR$CAT2
                  OUTFILE=SYS.JADIR   DYNALC=CAT1;
```

In batch mode only:

```
$JOB jobname;

ATTACH (CAT1=SYS.CATALOG: new-set-md: new-set-dvc,
        CAT2=SYS.CATALOG: old-set-md: old-set-dvc);

CATMOVE      FROM=SYS.JA
              INFILE=(SYS.CATALOG CAT=2)
              OUTFILE=(SYS.CATALOG CAT=1)
              REPLACE;

$ENDJOB;
```

### NOTE:

The JCL commands CATMOVE, CATPRINT and CATCHCHECK cannot be used in interactive mode.



### IMPORTANT:

The first SYS.CATALOG file to be attached must be the SYS.CATALOG of the running system.





---

## 8. Patching Procedures

### 8.1 Purpose

These Patching procedures enable you to perform corrective maintenance on the GCOS and DSA Domains.

This is done with an EC-MEDIA containing one or several word corrections provided by the Local Distribution Center or by your Bull CSE.

These corrections, called **EC (Emergency Corrections)**, may be applied to any running or non-running Set: P-Set, P2-Set, or R-Set.

The Patching procedures described in this chapter concern the GCOS Domain only. For the Patching procedures of the DSA Domain, refer to the chapter *DSA Installation and Updating Facility*.

There are no Patching procedures for the FW, OLTD and GSF Domains.

### 8.2 GCOS 7 Patching Procedure

The GCOS 7 files that may be modified using Emergency Corrections, are SYS.HLMLIB and SYS.SYSTEM.

This function should be used only at the request of your Bull CSE.



## 8.3 Operations

1. Make available the Set to be updated.

The Set to be updated may be running or not, depending on the IUF configuration.

2. Mount the EC-MEDIA, if any.
3. Start the IUF function PATCH\_HLMLIB or PATCH\_SYSTEM, depending on whether the EC is to be applied to the SYS.HLMLIB or SYS.SYSTEM file.

For details of the IUF functions PATCH\_HLMLIB and PATCH\_SYSTEM, refer to chapter *Installation and Updating Facility*.

4. If PATCH\_SYSTEM function has been used, the updated SYS.SYSTEM must be restored, as follows:

```
I: RESTORE
```

For other ISL parameters, refer to the *GCOS 7 System Operator's Guide*.

5. If the PATCH\_HLMLIB function has been used, the updated Load Modules must be cleared from backing store if they are preinitialized.

This is done in one of two ways:

- a. The command MDC CLM followed by MDC PLM
- b. The command RESTART (CLEAN) followed by the command MDC PLM.

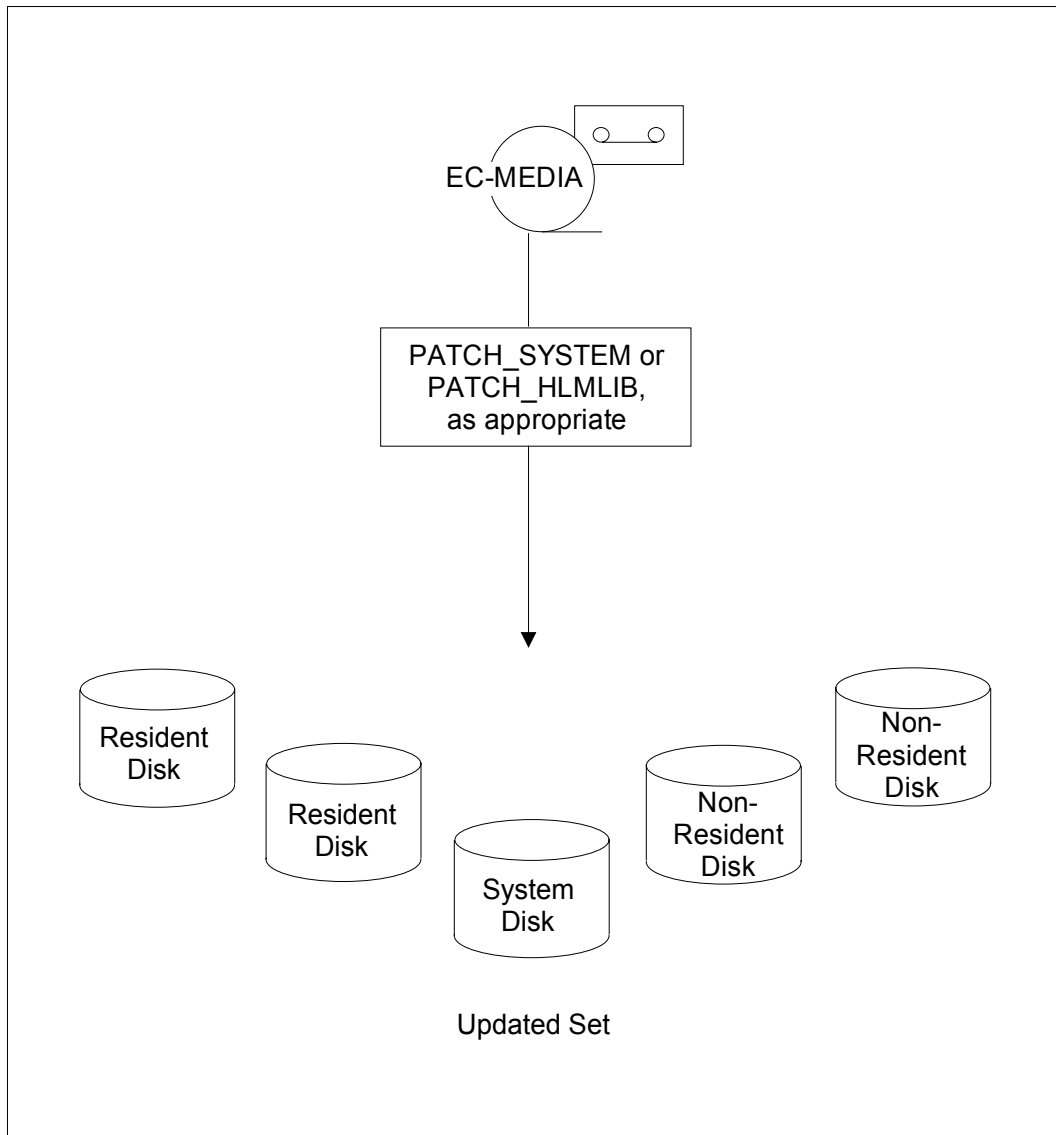


Figure 8-1. GCOS 7 Patching Procedures





---

## 9. Firmware Generation (FGF) and Test and Diagnostics (OLTD) Facilities

### 9.1 Introduction

Your Bull CSE uses these facilities to install the FW (Firmware) and OLTD (On-Line Tests & Diagnostics) Domains.

### 9.2 FGF (Firmware Generation Facility)

The Firmware Generation Facility is used to install and maintain all the components of the FW (Firmware) Domain on the P-Set, P2-Set, or R-Set, as appropriate.

There are two separate cases of Firmware installation.

- Installation on a new system:

Initial firmware installation is performed on the P-Set. The FW Domain can then be automatically transferred to the P2-Set or the R-Set as appropriate:

  - for a P2P-Configuration, to the P2-Set, using the GIUF function BUILD P2,
  - for an RP-Configuration, to the R-Set, using the GIUF job INSTAL\_RSET.
- Installation of a new Firmware release on an existing system:

Installation is performed on the running Set (P-, P2-, or R-Set, as appropriate to the Configuration type). The non-updated Sets may be subsequently brought up to the level of the new version by using the GIUF function LEVEL.



## 9.3 TDF (Tests and Diagnostics Facility)

### 9.3.1 DPS 7000 (except DPS 7000/XTA)

The TDF (Tests and Diagnostics Facility) is used to install and maintain all the components of the OLTD (On-Line Tests & Diagnostics) Domain on the P-Set, P2-Set, or R-Set, as appropriate. There are two separate cases of Firmware installation:

- installation on a new system.

Initial installation is performed on the P-Set. The OLTDs can then be automatically transferred to the P2-Set or the R-Set as appropriate:

- for a P2P-Configuration, to the P2-Set, using the GIUF function BUILDP2,
- for an RP-Configuration, to the R-Set, using the GIUF job INSTAL\_RSET.

- installation of a new release on an existing system.

Installation is performed on the running Set (P-, P2-, or R-Set, as appropriate to the Configuration type). The non-updated Sets may be subsequently brought up to the level of the new version by using the GIUF function LEVEL.

### 9.3.2 DPS 7000/XTA

The TDF (Tests and Diagnostics Facility) is now obsolete.

The necessary files are included and managed by the BSR tools (FGF) for compatibility reasons with GIUF.



---

## 10. DSA Installation and Updating Facility

### 10.1 Introduction

DIUF (DSA Installation and Updating Facility) is for the installation and maintenance of the telecommunications-related software and files on your system. The associated system files, for storing:

- the Datanet DNS software and/or
- CNS 7 software for CNP 7 and/or
- the FCP 7 software

belong to the DSA system file Domain.

This facility enables you to install and maintain all the components of the DSA system file Domain on the P-Set, P2-Set, R-Set as appropriate.

DIUF supports:

- The management of several versions of DNS and/or CNS 7 and/or FCP 7 on a single site at the same time.
- The application of source corrections to a Technical Status delivery.

**NOTE:**

DSA Domain files are to be installed on FBO volumes only, except:  
SYS.DSALOGi  
SYS.DSATRC\_<lct-name>\_x  
SYS.DSATRCIDX\_<lct-name>  
which may be allocated on FBO or VBO volumes.







### 10.2.1 R-Set Considerations

You are strongly recommended to run under a P2P-Configuration. One of the reasons for this is that the R-Set cannot be used to run telecommunications. This means that the following files are not created on the R-Set:

SYS.DSACONF, SYS.DSACORE, SYS.DSADUMP, SYS.DSACMD,  
SYS.DSALOG1, SYS.DSATRC\_<lct-name>\_x and SYS.DSATRCIDT\_<fcp-ts>  
files

### 10.2.2 DNS/CNS/FCP 7 Media

New Technical Statuses of CNS, DNS, or FCP 7 are either included in the DOMAINS-MEDIA, which is supplied by the factory, or are on separate media supplied by your Bull CSE.

### 10.2.3 Technical Statuses and Releases

The first two characters of the Technical Status identifier correspond to the Release number. A new Technical Status to be installed, for which no Technical Status of the same Release is already installed, is called a new Release.



### 10.3 DIUF Tools

The IOF user may access the DIUF Facility by using the GCL command ENTER\_DIUF, which provides the following facilities:

INSTALL	is used to install a new DNS/CNS 7/FCP 7 Release, and to install a Technical Status but without bringing forward any site-specific corrections. This procedure is required when installing a new DNS, CNS 7, or FCP 7 Release for the first time in the DSA Domain.
UPDATE	is used to install a new Technical Status on an already-installed DNS, CNS 7, or FCP 7 Release, with optional bringing forward of site-specific corrections, provided that all the files for the DSA Domain have already been created. This procedure may also be used for installing a new Release, or a new Technical Status without bringing forward any site-specific corrections. An already-installed Technical Status cannot be re-installed with the UPDATE function.
PATCH	is used to apply emergency corrections to an already-installed DNS or CNS 7 Technical Status, and to modify the contents of Patch files. This function is not available for FCP 7.
DELETE_TS	is used to delete a DNS and/or CNS 7 and/or FCP 7 Technical Status.
LIST	is used to list the DNS and/or CNS Technical Statuses already installed on a given Set, and the version number of the DIUF Factory installed on the system.
LOADFACT	is used for loading a new version of the DIUF Factory, <b>only at the request of your Bull CSE.</b>
BUILD_TRCNCC	is used to create the SYS.DSATRC_<let-name>_x files. The FCP 7 trace files must be created if FCP 7 is being installed for the first time, or if a new LCT (Local Communication Controller) is being added to the configuration.



## 10.4 The ENTER\_DIUF Command

### Purpose

Provides access to all the DIUF functions.

Any functions which modify the contents of the files concerned must be submitted under the project SYSADMIN with USER=SYSADMIN.

### Syntax

```
{ENTER_DIUF | DIUF}  
  
{FUNCTION=function-name | COMMAND=immediate-command}  
  
[SET={P | P2 | R}] [JOBIN=MS];
```

### Parameters

FUNCTION	names a DIUF function to be performed, with subsequent parameters being taken at the user's terminal; forbidden in batch mode, it is mutually exclusive with COMMAND.
COMMAND	specifies a string containing no more than one DIUF function followed by its parameters.
SET	designates the Set to be updated; by default, this is: <ul style="list-style-type: none"><li>– At first installation (DSA Domain not yet initialized):<ul style="list-style-type: none"><li>The P-Set, in a P2P- or PO-Configuration.</li><li>The R-Set, in an RP-Configuration.</li></ul></li><li>– The running Set, in all other cases.</li></ul>
JOBIN	may occasionally be required for debugging purposes. When set to MS, it launches the DIUF functions below from library SYS.DSASLIB, not from the installation media: <ul style="list-style-type: none"><li>– INSTAL, for the installation option</li><li>– LOADFACT, for the load factory option.</li></ul>



## 10.5 Initial Installation

Your Bull CSE performs this procedure during the installation described in the chapter *Installation Procedure*. It allows the installation of one or several DNS and/or CNS 7 and/or FCP 7 Technical Statuses on the P-Set during its building.

### Procedure

#### NOTE:

The first five steps will often have been done already (when the P-Set was built in the factory from the P-MEDIA).

1. Mount the media containing the DNS/CNS 7/FCP 7 Release to be installed.
2. Proceed with installation on the P-Set, as follows, either:

```
S: DIUF [SET=P] COMMAND='INSTAL TSVOL=tsmd:tsdvc,
    TSTYP={CNS|DNS|NCC}, DSAVOL=dsamd:dsadvc;';
```

or:

```
S: DIUF [SET=P] FUNCTION=INSTAL;
```

followed by introduction of the TSVOL, TSTYP and DSAVOL parameters on the next screen.

tsmd and tsdvc are, respectively, the media name and device class of the media containing the DNS/CNS 7/FCP 7 Release to be installed.

TSTYP is the DSA product to be installed: DNS, CNS or FCP 7.

dsamd and dsadvc are, respectively, the media name and device class of the volume on which the SYS.DSA\* files are to be allocated. The DSAVOL parameter is given only at first installation of a DNS or CNS 7 Release. It must be given on every installation of an FCP 7 Release. The volume must be of FBO organization.

INSTAL allocates the SYS.DSABLIB, SYS.DSASLIB, SYS.DSALMLIB, SYS.DSACONF, SYS.DSACORE, SYS.DSADUMP, SYS.DSACMD, and SYS.DSALOG1 files on the P-Set at first installation. If an FCP 7 Release is installed, INSTAL also allocates the SYS.DSATRCIDT\_<fcp-ts> file on the P-Set.

3. If an FCP 7 Release is to be installed, run the DIUF function BUILD\_TRCNCC to allocate the SYS.DSATRC\_<lct-name>\_x files on the P-Set.

The BUILD\_TRCNCC function is described later in this chapter.



4. If further DNS/CNS 7/FCP 7 Technical Statuses are to be installed, repeat Steps 1, 2 and 3, the requisite number of times.

This is the same as for Steps 1, 2 and 3 above, except that DSAVOL must not be given when installing a DNS or CNS 7 Technical Status, since all the required SYS.DSA\* files have already been allocated. However DSAVOL must always be specified when installing an FCP 7 Technical Status.

5. Validate the P-Set and create the P2P or R-Configuration as described in the chapter *Installation Procedure*, using the GIUF functions VALIDP, BUILDP2, or INSTAL\_RSET functions as appropriate.
6. Reload the system from the P-Set.
7. Perform Datanet and/or CNP7 SYSGEN, as appropriate, on the frontend processors affected by the Technical Status(es) installed in Steps 1, 2 and 4 above.

Refer to the appropriate *System Generation Guide*.

Note that FCP 7 does not need system generation.

**NOTE:**

DNS or CNS 7 Patch files may be modified, if necessary, at the end of Step 3 and Step 4, using the DIUF function PATCH:

```
S: DIUF FUNCTION=PATCH SET=...;
```

or:

```
S: DIUF COMMAND='PATCH ...' SET=...;
```

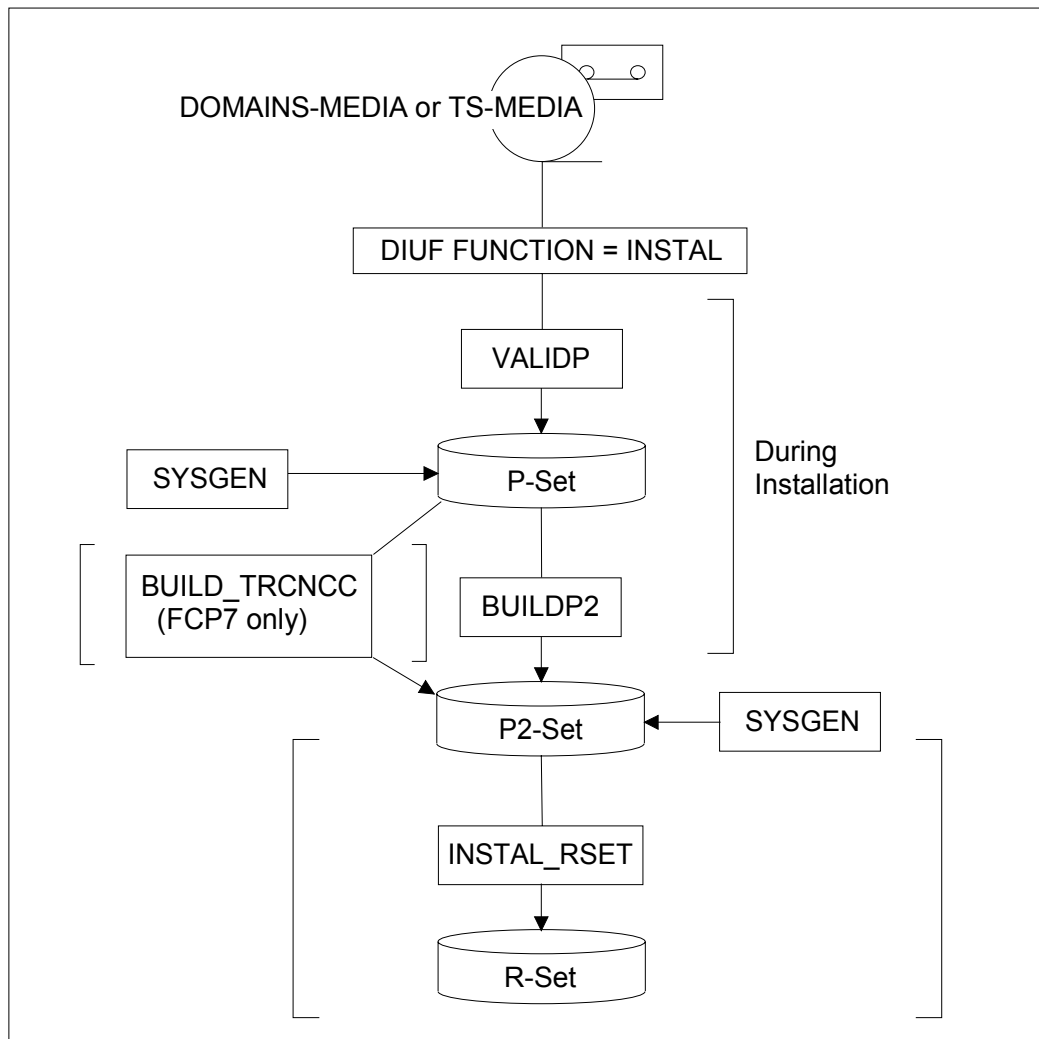


Figure 10-1. Initial Installation of DNS/CNS 7/FCP 7 via the DIUF Facility



## 10.6 Add-on installation

DNS and/or CNS 7 and/or FCP 7 is added to the user's existing configuration by adding to the system, as appropriate:

- The corresponding MIs (Marketing Identifiers)
- The required file Domain.

If at least one of the DNS, CNS 7 or FCP 7 Releases already exists, then one or both of the other two can be installed simply by adding their MI(s). The file Domain does not need to be recreated.

Adding an MI must be done before applying the "Updating Procedure for a New DNS or CNS 7 or FCP 7 Release", as explained below. The procedure for adding an MI is as follows:

- In a PO Configuration, by running the GIUF function ADD\_MI on the P-Set.
- In a P2P Configuration, by running the GIUF function ADD\_MI on the running Set and subsequently transfer the update to the non-modified Set with the GIUF function LEVEL\_MI.
- In an RP Configuration, by running the GIUF function ADD\_MI on the P-Set and subsequently transfer the update to the R-Set with the GIUF function LEVEL\_MI.

If neither DNS nor CNS 7 nor FCP 7 exist on the system, the procedure differs according to the type of configuration involved, PO, P2P or RP.

### 10.6.1 PO-Configuration Add-On Procedure

**NOTE:**

Installation is performed on the P-Set.

1. Mount the media containing the DNS and/or CNS 7 and/or FCP 7 Release to be installed.
2. Perform installation on the P-Set, as appropriate, with either:

```
S: DIUF [SET=P] COMMAND='INSTAL TSVOL=tsmd:tsdvc,  
TSTYP={CNS|DNS|NCC}, DSAVOL=dsamd:dsadvc';
```

or:

```
S: DIUF [SET=P] FUNCTION=INSTAL;
```

followed by the introduction of the TSVOL, TSTYP and DSAVOL parameters on the next screen.



tsmd and tsdvc, respectively, the media name and device class of the volume containing the DNS/CNS 7/FCP 7 Release to be installed.

TSTYP is the DSA product to be installed: DNS, CNS or FCP 7.

dsamd and dsadvc are, respectively, the media name and device class of the volume on which the SYS.DSA\* files are to be allocated. The DSAVOL parameter is given only at first installation of a DNS or CNS 7 Release. It must be given on every installation of an FCP 7 Release. The volume must be of FBO organization.

INSTAL allocates the SYS.DSABLIB, SYS.DSASLIB, SYS.DSALMLIB, SYS.DSACONF, SYS.DSACORE, SYS.DSADUMP, SYS.DSACMD, and SYS.DSALOG1 files on the P-Set at first installation. If an FCP 7 Release is installed, INSTAL also allocates the SYS.DSATRCIDT\_<fcp-ts> file on the P-Set.

3. If an FCP 7 Release is to be installed, run the DIUF function BUILD\_TRCNCC to allocate the SYS.DSATRC\_<lct-name>\_x files on the P-Set.

The BUILD\_TRCNCC function is described later in this chapter.

4. If further DNS/CNS 7/FCP 7 Technical Statuses are to be installed, repeat Steps 1, 2 and 3, the requisite number of times.

This is the same as for Steps 1, 2 and 3 above, except that DSAVOL must not be given when installing a DNS or CNS 7 Technical Status, since all the required SYS.DSA\* files have already been allocated. However DSAVOL must always be specified when installing an FCP 7 Technical Status.

5. If new MIs are to be validated, run the GIUF function ADD\_MI, in order to validate the DSA system file Domain on the P-Set.
6. Reload the system from the P-Set.

If new MIs have been validated, the RESTORE option is mandatory, in order to implement the MIs just validated.

If no MIs have been validated, the RESTORE option is not required.

7. Perform Datanet and/or CNP7 SYSGEN, as appropriate, on the front-end processors affected by the Technical Status(es) installed in Steps 1, 2 and 4 above.

Refer to the appropriate *System Generation Guide*.

Note that FCP 7 does not need system generation.





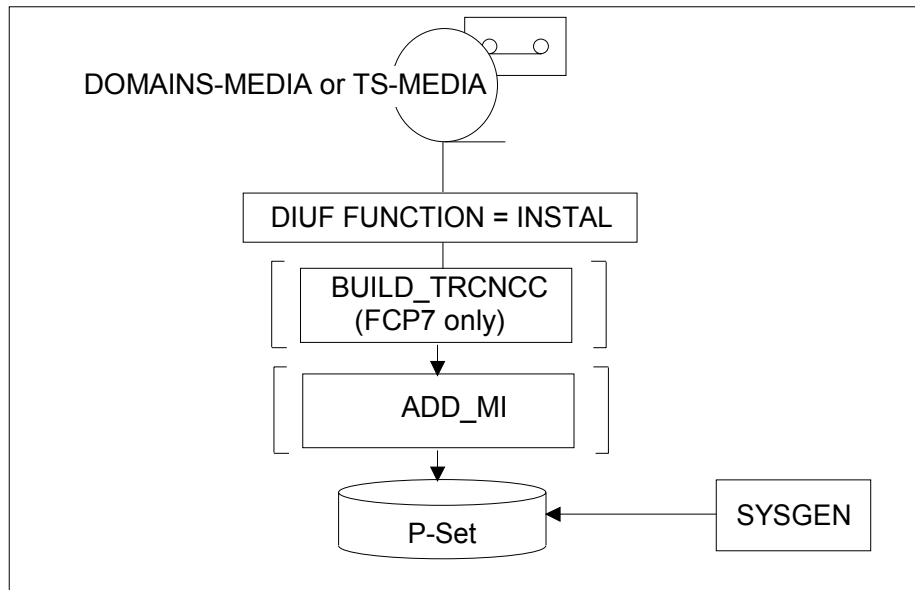
**NOTE:**

DNS or CNS 7 Patch files may be modified, if necessary, at the end of Step 3 and Step 4, using the DIUF function PATCH:

S: DIUF FUNCTION=PATCH SET=...;

or:

S: DIUF COMMAND='PATCH ...' SET=...;



**Figure 10-2. PO-Configuration Add-On Procedure**



## 10.6.2 P2P-Configuration Add-On Procedure

### NOTE:

Installation is started with the P-Set running.

1. Mount the media containing the DNS and/or CNS 7 and/or FCP 7 Release to be installed.
2. Perform the installation on the P-Set, as appropriate, with either:

```
S: DIUF [SET=P] COMMAND='INSTAL TSVOL=tsmd:tsdvc,
    TSTYP={CNS|DNS|NCC}, DSAVOL=dsamd:dsadvc;';
```

or:

```
S: DIUF [SET=P] FUNCTION=INSTAL;
```

followed by introduction of the TSVOL, TSTYP and DSAVOL parameters on the next screen.

tsmd and tsdvc are, respectively, the media name and device class of the volume containing the DNS/CNS 7/FCP 7 Release to be installed.

TSTYP is the DSA product to be installed: DNS, CNS or FCP 7.

dsamd and dsadvc are, respectively, the media name and device class of the volume on which the loaded SYS.DSA\* files are to be allocated. The DSAVOL parameter is given only at first installation of a DNS or CNS 7 Release. It must be given on every installation of an FCP 7 Release. The volume must be of FBO organization.

INSTAL allocates the SYS.DSABLIB, SYS.DSASLIB, SYS.DSALMLIB, SYS.DSACONF, SYS.DSACORE, SYS.DSADUMP, SYS.DSACMD, and SYS.DSALOG1 files on the P-Set at first installation. If an FCP 7 Release is installed, INSTAL also allocates the SYS.DSATRCIDT\_<fcp-ts> file on the P-Set.

3. If further DNS/CNS 7/FCP 7 Technical Statuses are to be installed, repeat Steps 1 and 2 above as appropriate.

This is the same as for Steps 1 and 2 above, except that DSAVOL must not be given when installing a DNS or CNS 7 Technical Status, since the loaded SYS.DSA\* files have already been allocated. However DSAVOL must always be specified when installing an FCP 7 Technical Status.



4. The GIUF function BUILD P2 can be used to copy the P-Set to the P2-Set concerned. The corresponding TLRGEN options are:

```
TLRGEN FILESET=DSA COMSET=INITIAL ;
```

BUILD P2 is used:

- to extend the DNS/CNS 7/FCP 7 Technical Status just installed (SYS.DSABLIB, SYS.DSASLIB, SYS.DSALMLIB, and possibly the SYS.DSATRCIDT\_<fcp-ts> files), from the P-Set to the P2-Set
- and to allocate the SYS.DSACONF, SYS.DSACORE, SYS.DSADUMP, SYS.DSACMD and SYS.DSALOG1 files on the P2-Set.

For the TLRGEN options, refer to Chapter 6.

5. If an FCP 7 Release is to be installed, run the DIUF function BUILD\_TRCNCC to allocate the SYS.DSATRC\_<lct-name>\_x files on the (running) P-Set and then on the (non-running) P2-Set.

The BUILD\_TRCNCC function is described later in this chapter.

If no new MIs need to be validated, jump to Step 7; otherwise proceed with Step 6.

6. If new MIs are to be validated, run the GIUF function ADD\_MI on the running P-Set.

Run the GIUF function LEVEL\_MI in order to validate the DSA file Domain on the (non-running) P2-Set.

7. Reload the system from the P-Set.

- If new MIs have been validated, the RESTORE option is mandatory, in order to implement the MIs just validated.
- If no MI has been validated, the RESTORE option is not required.

8. Perform Datanet and/or CNP7 SYSGEN, as appropriate, on the front-end processors affected by the Technical Status(es) installed in Steps 1 to 3 above.

Refer to the appropriate *System Generation Guide*.

Note that FCP 7 does not need system generation.

**NOTE:**

DNS or CNS 7 Patch files may be modified, if necessary, at the end of Step 2 and Step 3, using the DIUF function PATCH:

```
S: DIUF FUNCTION=PATCH SET=...;
```

or:

```
S: DIUF COMMAND='PATCH ...' SET=...;
```

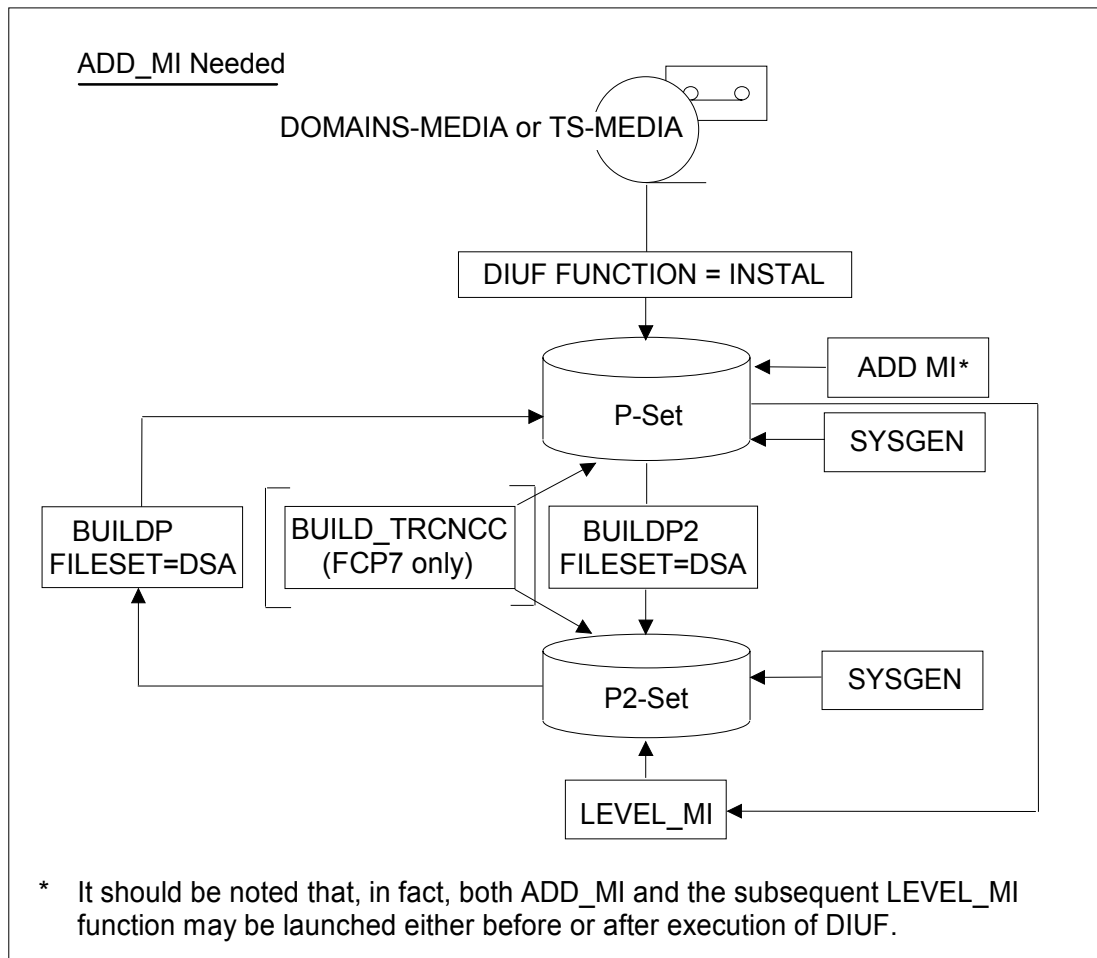


Figure 10-3. P2P Configuration Add-On Procedure, using ADD\_MI

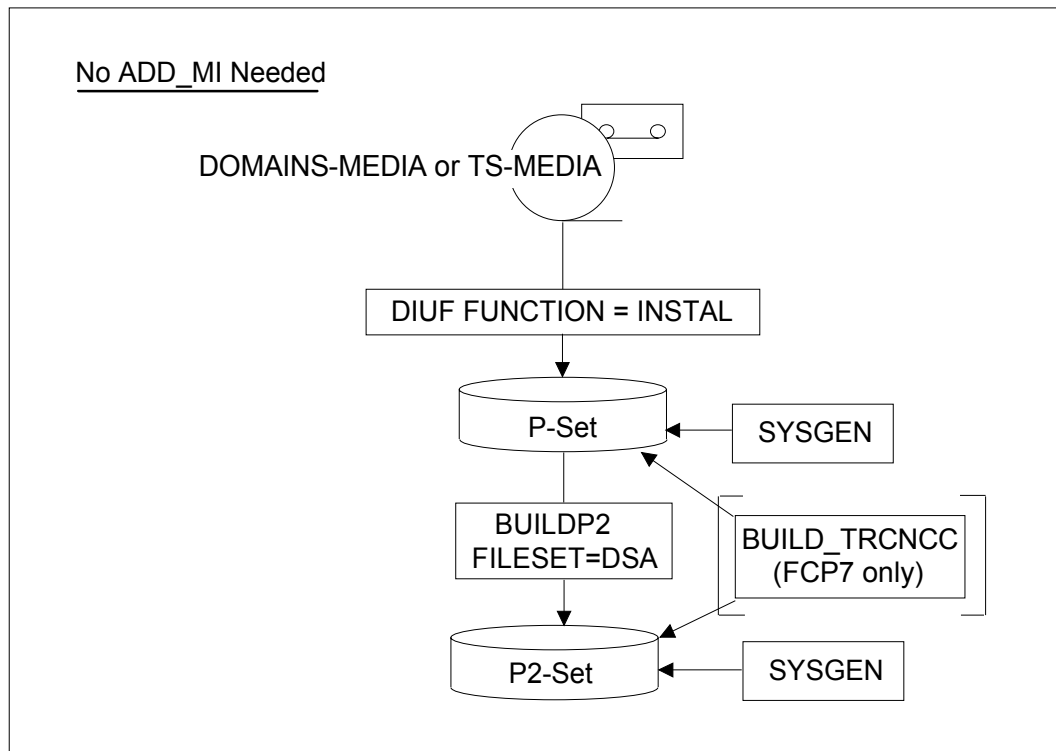


Figure 10-4. P2P Configuration Add-On Procedure, Without ADD\_MI



### 10.6.3 RP-Configuration Add-On Procedure

**NOTE:**

Installation is performed on the R-Set.

1. Mount the media containing the DNS and/or CNS 7 and/or FCP 7 Release to be installed.
2. Proceed with installation on the R-Set, as follows, either:

```
S: DIUF [SET=R] COMMAND='INSTAL TSVOL=tsmd:tsdvc,  
TSTYP={CNS|DNS|NCC}, DSAVOL=dsamd:dsadvc;';
```

or:

```
S: DIUF [SET=R] FUNCTION=INSTAL;
```

followed by introduction of the TSVOL, TSTYP and DSAVOL parameters on the next screen.

tsmd and tsdvc are, respectively, the media name and device class of the volume containing the DNS/CNS 7/FCP 7 Release to be installed.

TSTYP is the DSA product to be installed: DNS, CNS or FCP 7.

dsamd and dsadvc are, respectively, the media name and device class of the volume on which the loaded SYS.DSA\* files are to be allocated. The DSAVOL parameter is given only at first installation of a DNS or CNS 7 Release. It must be given on every installation of an FCP 7 Release. The volume must be of FBO organization.

INSTAL allocates the SYS.DSABLIB, SYS.DSALMLIB, and the SYS.DSASLIB files on the R-Set at first installation. If an FCP 7 Release is installed, INSTAL also allocates the SYS.DSATRCIDT\_<fcp-ts> file on the R-Set.

3. If further DNS/CNS 7/FCP 7 Technical Statuses are to be installed, repeat Steps 1 and 2 above as appropriate.

This is the same as for Steps 1 and 2 above, except that DSAVOL must not be given when installing a DNS or CNS 7 Technical Status, since the loaded SYS.DSA\* files have already been allocated. However DSAVOL must always be specified when installing an FCP 7 Technical Status.



4. The GIUF function BUILDP must be run from the R-Set to the P-Set concerned. The corresponding TLRGEN options are:

```
TLRGEN FILESET=DSA COMSET=INITIAL ;
```

BUILD P is used:

- to extend the DNS/CNS 7/FCP 7 Technical Status just installed (SYS.DSABLIB, SYS.DSASLIB, SYS.DSALMLIB, and possibly the SYS.DSATRCIDT\_<fcp-ts> files), from the R-Set to the P-Set.
- and to allocate the SYS.DSACONF, SYS.DSACORE, SYS.DSADUMP, SYS.DSACMD and SYS.DSALOG1 files on the P-Set.

For the TLRGEN options, refer to Chapter 6.

5. If an FCP 7 Release is to be installed, run the DIUF function BUILD\_TRCNCC to allocate the SYS.DSATRC\_<lct-name>\_x files on the (running) R-Set and then on the (running or non-running) P-Set.

The BUILD\_TRCNCC function is described later in this chapter.

If no new MIs need to be validated, jump to Step 7; otherwise proceed with Step 6.

6. If new MIs are to be validated, run the GIUF function ADD\_MI on the non-running P-Set.

Run the GIUF function LEVEL\_MI in order to validate the DSA file Domain on the R-Set.

7. Reload the system from the P-Set or R-Set.
  - If new MIs have been validated, the RESTORE option is mandatory, in order to implement the MIs just validated.
  - If no MI has been validated, the RESTORE option is not required.
8. Perform Datanet and/or CNP7 SYSGEN, as appropriate, on the front-end processors affected by the Technical Status(es) installed in Steps 1 to 3 above.

Refer to the appropriate *System Generation Guide*.

Note that FCP 7 does not need system generation.

**NOTE:**

DNS or CNS 7 Patch files may be modified, if necessary, at the end of Step 2 and Step 3, using the DIUF function PATCH. Either:

```
S: DIUF FUNCTION=PATCH SET=...;
```

or:

```
S: DIUF COMMAND='PATCH ...' SET=...;
```

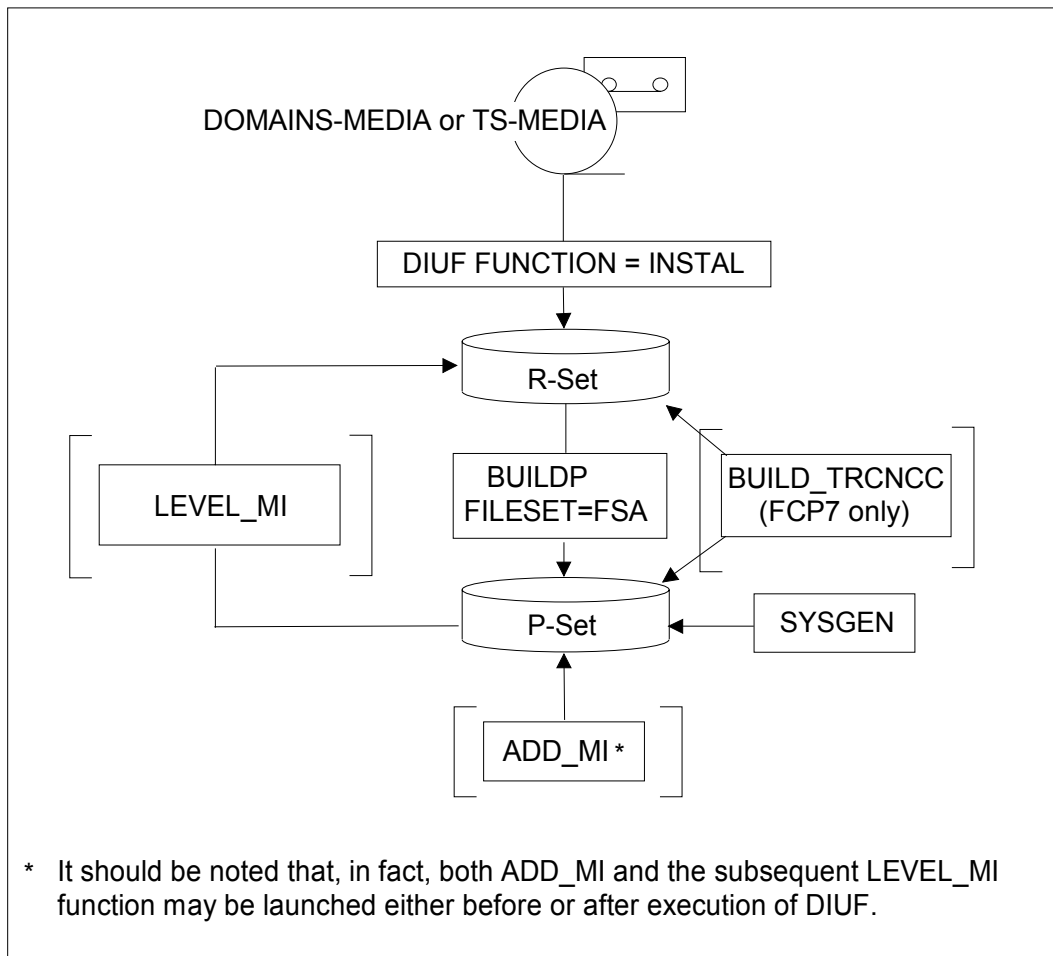


Figure 10-5. RP-Configuration Add-On Procedure





## 10.7 Updating

This operation may be performed on any one of the user's Sets (P, P2 or R), as appropriate to the requirements of his installation, and involves the installation of a *new* Technical Status which may be either:

1. A new DNS, CNS 7 or FCP 7 Release, or a new Technical Status without bringing forward any site Patch corrections. In this case, either the `INSTAL` function or the `UPDATE` function without the `TSREF` parameter may be used.
2. An update to an already-installed Release. In this case there are two possibilities:
  - without bringing forward any site Patch corrections, the `INSTAL` function or the `UPDATE` function without the `TSREF` parameter may be used.
  - with bringing forward site specific Patch corrections, the `UPDATE` function with `TSREF` may be used.

### NOTES:

1. An already-installed Technical Status cannot be re-installed with the `UPDATE` function. For this purpose use the `INSTAL` function.
2. The function `DELETE_TS` (described later in this chapter) must be used to delete the superseded version of DNS, CNS 7 or FCP 7, if the new version is to be processed by `DIUF` as the default option.

### 10.7.1 Updating Procedure Without Bringing Forward Patches

1. Mount the media containing the DNS and/or CNS 7 and/or FCP 7 Release to be installed.
2. Proceed with installation on the desired Set, with one of the following:

- `S: DIUF [SET={P|P2|R}] COMMAND='INSTAL TSVOL = tsmc:tsdvc, TSTYP={CNS|DNS|NCC}';`
- `S: DIUF [SET={P|P2|R}] FUNCTION=INSTAL;`

followed by introduction of the `TSVOL` parameter on the next screen.

Or with one of the following:

- `S: DIUF [SET={P|P2|R}] COMMAND='UPDATE TSVOL = tsmc:tsdvc, TSTYP={CNS|DNS|NCC}';`
- `S: DIUF [SET={P|P2|R}] FUNCTION=UPDATE;`

followed by the introduction of the `TSVOL`, or `TSTYP`, parameter (but not `TSREF`) on the following screens.



tsmd and tsdvc are, respectively, the media name and device class of the media containing the DNS/CNS 7/FCP 7 Technical Status to be installed.

TSTYP is the DSA product to be installed: DNS, CNS or FCP 7. The DSA product FCP 7 is related to FCP 7.

When using the INSTAL function for DNS or CNS 7 installation, DSAVOL must **not** be given since this is not the first installation. For FCP 7 installation, however, DSAVOL must always be given.

3. At first installation of an FCP 7 Release, run the DIUF function BUILD\_TRCNCC to allocate the SYS.DSATRC\_<lct-name>\_x files on the P-Set and P2-Set.

The BUILD\_TRCNCC function is described later in this chapter.

4. In P2P- and RP-Configurations, reload the system from the target Set, provided that this is neither the running Set nor the R-Set.

PO-Configuration should skip this step, and go directly to Step 6.

5. Perform SYSGEN in order to generate DNS or CNS 7, as appropriate, on the running target Set, unless this is the R-Set.

Refer to the appropriate *System Generation Guide*. This is the final Step to be performed on PO-Configuration.

Note that FCP 7 does not need system generation.

6. In order to extend the Technical Status just applied in P2P- and RP-Configurations, run the GIUF function LEVEL with the DSA parameter on the other Set of the configuration.
7. Reload the system from the other Set of the configuration, unless this is either the running Set or the R-Set.
8. Perform SYSGEN in order to generate DNS or CNS 7, as appropriate, on the other Set of the configuration (unless it is the R-Set), for the frontend processors affected by the Technical Status installed in Steps 1 and 2 above.

Refer to the appropriate *System Generation Guide*.

Note that FCP 7 does not need system generation.

**NOTE:**

DNS or CNS 7 Patch files may be modified, if necessary, at the end of Step 2, using the DIUF function PATCH:

```
S: DIUF FUNCTION=PATCH SET=...;
```

or:

```
S: DIUF COMMAND='PATCH ...' SET=...;
```



## 10.7.2 Updating Procedure With Patches Brought Forward

1. Mount the media containing the DNS and/or CNS 7 and/or FCP 7 Technical Status to be installed.
2. Proceed with installation on the desired Set, as follows, either:

```
S: DIUF [SET={P|P2|R}] COMMAND='UPDATE
      TSVOL=tsmd:tsdvc, TSREF=tsref,
      TSTYP={CNS|DNS|NCC}, [TS=xxxx],
      [REPLACE={0|1}], [DELPAT=seqfile],
      [PRTFILE=seqfile] ;';
```

or:

```
S: DIUF [SET={P|P2|R}] FUNCTION=UPDATE;
```

followed by introduction of the TSVOL, TSTYP, TS, TSREF, REPLACE, DELPAT and PRTFILE parameters on the screens that follow.

tsmd and tsdvc are, respectively, the media name and device class of the media containing the DNS/CNS 7/FCP 7 Technical Status to be installed.

TSTYP is the DSA product to be installed: DNS, CNS or FCP 7.

DELPAT is used if Updating Procedure has aborted because of corrections which could not be brought forward, to designate a sequential file (or subfile) containing DELETE requests for corrections not to be brought forward to the new Technical Status; for further details, see under REQUEST SYNTAX later in this chapter.

TS identifies the 4-digit number of the Technical Status to be installed.

PRTFILE introduces a sequential file (or subfile) to contain the Patch Processor report; SYS.OUT is the default value.

TSREF identifies the 4-digit number of the Technical Status from which site-specific Patches are to be conserved with the new Technical Status being installed.

REPLACE=0 will preserve the referenced Technical Status.

REPLACE=1 will result in the Technical Status designated by TSREF being deleted.

3. At first installation of an FCP 7 Release, run the DIUF function BUILD\_TRCNCC to allocate the SYS.DSATRC\_<lt-name>\_x files on the P-Set (RP, PO and P2P-Configurations) and then on the P2-Set (P2P-Configuration).

The BUILD\_TRCNCC function is described later in this chapter.



4. In P2P- and RP-Configurations, reload the system from the target Set, provided that this is neither the running Set nor the R-Set.  
PO-Configuration should not perform this step, but skip to Step 6.
5. Perform SYSGEN in order to generate DNS or CNS 7, as appropriate, on the running target Set, unless this is the R-Set.  
Refer to the appropriate *System Generation Guide*. This is the final Step to be performed on PO-Configurations.  
Note that FCP 7 does not need system generation.
6. In order to extend the Technical Status just applied in P2P- and RP-Configurations, run the GIUF function LEVEL with the DSA parameter on the other Set of the configuration.
7. Reload the system from the other Set of the configuration, unless this is either the running Set or the R-Set.
8. Perform SYSGEN in order to generate DNS or CNS 7, as appropriate, on the other Set of the configuration, unless this is the R-Set, in an RP-Configuration.  
Refer to the appropriate *System Generation Guide*.  
Note that FCP 7 does not need system generation.

**NOTE:**

DNS or CNS 7 Patch files may be modified, if necessary, at the end of Step 2, using the DIUF function PATCH. Either:

```
S: DIUF FUNCTION=PATCH SET=...;
```

or:

```
S: DIUF COMMAND='PATCH ...' SET=...;
```

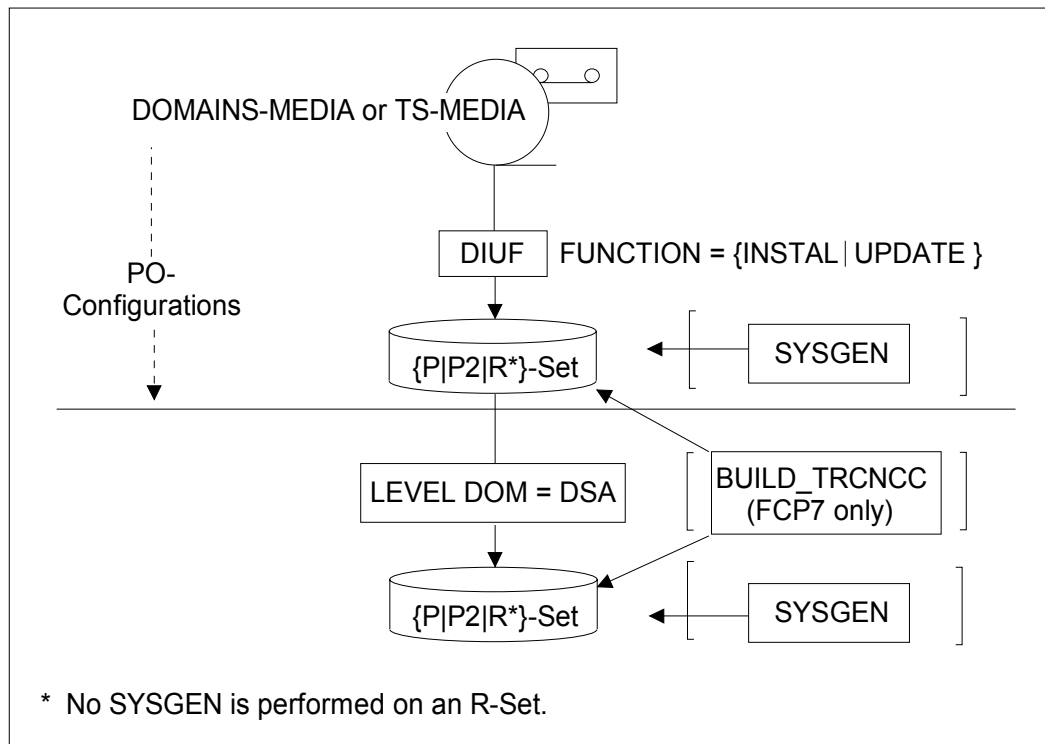


Figure 10-6. Updating Procedure in the DIUF Facility



## 10.8 Patching

This function allows the user to apply Emergency Corrections (EC) which may consist of any of:

- Patches
- Patch deletions
- The reactivation of Patches which have been cancelled

For a detailed description of the corrections supported, refer to "**Patch Processor**" later in this chapter.

### NOTE:

Patching is not applicable to FCP 7

### Patching Procedure

1. Mount the media containing the Patch(es) to be applied.
2. Launch the DIUF Patch session with either:

```
S: DIUF [SET={P|P2|R}] COMMAND='PATCH TS=ts PATFILE=
patfile [PRTFILE=prtfile];'
```

or:

```
S: DIUF [SET={P|P2|R}] FUNCTION=PATCH;
```

Followed by introduction of the TS, PATFILE and PRTFILE parameters on the next screen.

In all cases, the default value for SET is the running Set. In a P2P-Configuration, the updated Set may be either the P- or the P2-Set. In an RP-Configuration, the running Set may be either the R- or the P-Set (although usually the latter). The file containing the corrections may be:

lib..mb	for the member of a resident or cataloged library member
lib..mb:md:dvc	for an uncataloged non-resident library member
efn	for a RESIDENT or cataloged sequential file
efn:md:dvc	for an uncataloged non-resident sequential file
::TN	for user console in asynchronous mode (for the command language, refer to "PATCH PROCESSOR" later in this chapter).



3. In P2P- and RP-Configurations, reload the system from the target Set, provided that this is neither the running Set nor the R-Set.  
PO-Configurations should not execute this step, but skip directly to step 4, below.
4. Perform Datanet and/or CNP7 SYSGEN, as appropriate, on the running target Set, unless this is the R-Set.  
Refer to the appropriate *System Generation Guide*. This is the final Step to be performed on PO-Configurations.
5. In order to extend the Emergency Corrections just applied in P2P- and RP-Configurations, run the GIUF function LEVEL with the DSA parameter on the other Set of the configuration.
6. Reload the system from the other Set of the configuration, unless this is the R-Set, in an RP-Configuration.
7. Perform Datanet and/or CNP7 SYSGEN, as appropriate, on the other Set of the configuration, unless this is the R-Set, in an RP-Configuration.  
Refer to the appropriate *System Generation Guide*.

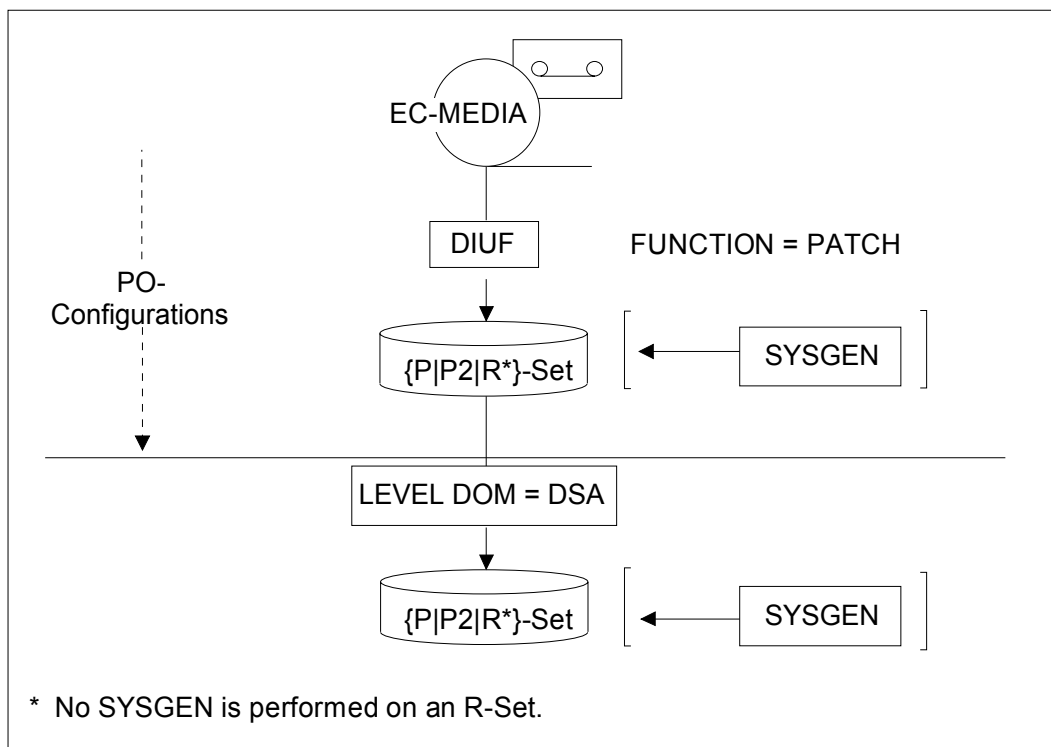


Figure 10-7. Patching Procedure in the DIUF Facility



## 10.9 The DELETE\_TS Function

This function enables the user to delete a DNS/CNS 7/FCP 7 Technical Status that is of no further use.

### Deletion Procedure

1. Launch the DELETE\_TS session, with either:

```
S: DIUF [SET={P|P2|R}] COMMAND='DELETE_TS TS=ts;';
```

or:

```
S: DIUF [SET={P|P2|R}] FUNCTION=DELETE_TS;
```

followed by introduction of the TS parameter on the next screen.

The default value for SET is the running Set. In a P2P-Configuration, SET may be P or P2; in an RP-Configuration, SET may be R or P. Note that, in a PO-Configuration, this is the only step to be executed.

2. In P2P- and RP-Configurations, delete the Technical Status from the installation's other Set, by repeating Step 1 above for that Set.

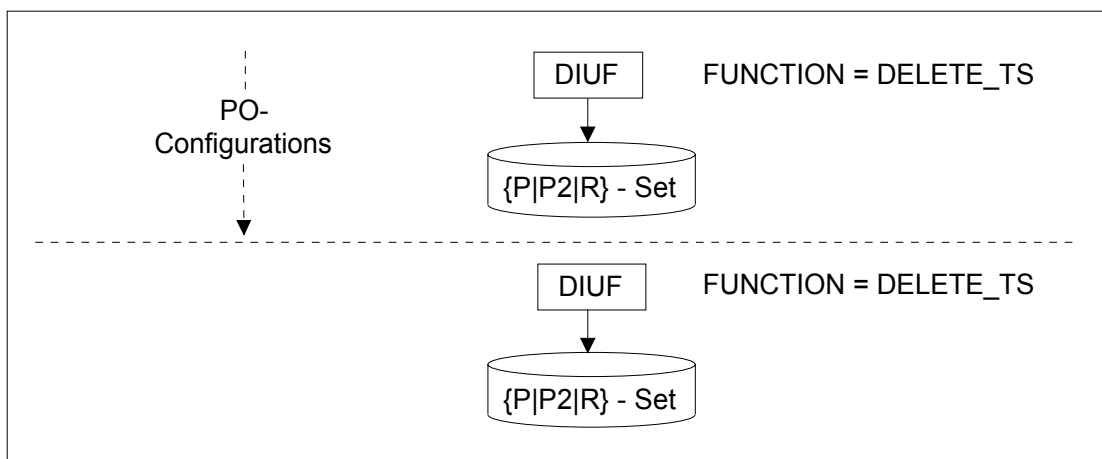


Figure 10-8. Patching Procedure in the DIUF Facility





## 10.10 Consulting

This allows the user to list all the DNS/CNS 7/FCP 7 Technical Statuses installed on a given Set, and the Version Number of the DIUF Factory installed on that Set.

### Consulting Procedure

Launch the LIST session with:

```
S: DIUF FUNCTION=LIST [SET={P|P2|R}];
```

SET is by default the running Set. In a P2P-Configuration, the Set may be P or P2; in an RP-Configuration, the Set may be R or P.



## 10.11 The LOAD\_FACTORY Function

This function enables the user to load a new version of the DIUF Factory from an installation media, but only to be undertaken at the request of your Bull CSE.

### Load Factory Procedure

1. Mount the media containing the new DIUF Factory.
2. Launch the Load Factory session with either:

```
S: DIUF [SET={P|P2|R}] COMMAND='LOADFACT TSVOL=
      tsmc:tsdvc';
```

or:

```
S: DIUF FUNCTION=LOADFACT;
```

followed by introduction of the TSVOL parameter on the next screen.

tsmd and tsdvc are, respectively, the media name and the device class of the media containing the DIUF Factory to be installed.

This is the final step, for PO-Configurations.

3. In P2P- and RP-Configurations, the new DIUF Factory may be extended to the installation's other Set, in either of the two following ways:
  - Running the GIUF function LEVEL with the DOM=DSA parameter, against the other Set.
  - Launching a LOADFACT session on the other Set, in the same way as in Step 2 above.

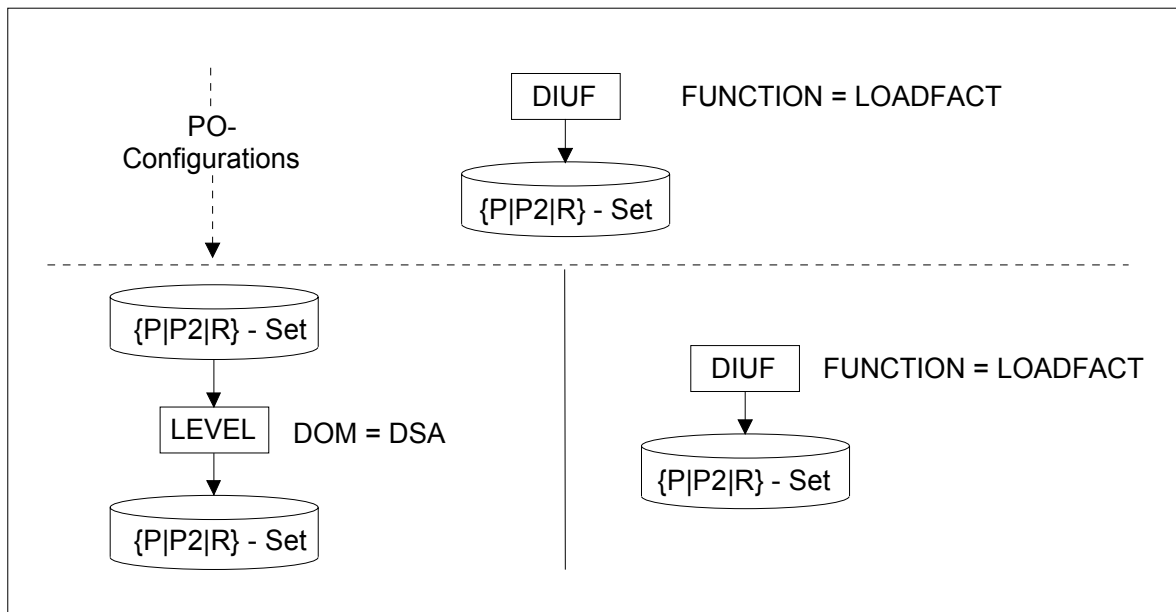


Figure 10-9. Load Factory Procedure in the DIUF Facility



---

## 10.12 Patch Processor

Patch file management is performed using the Patch Processor H\_DIUF\_MNPTF.

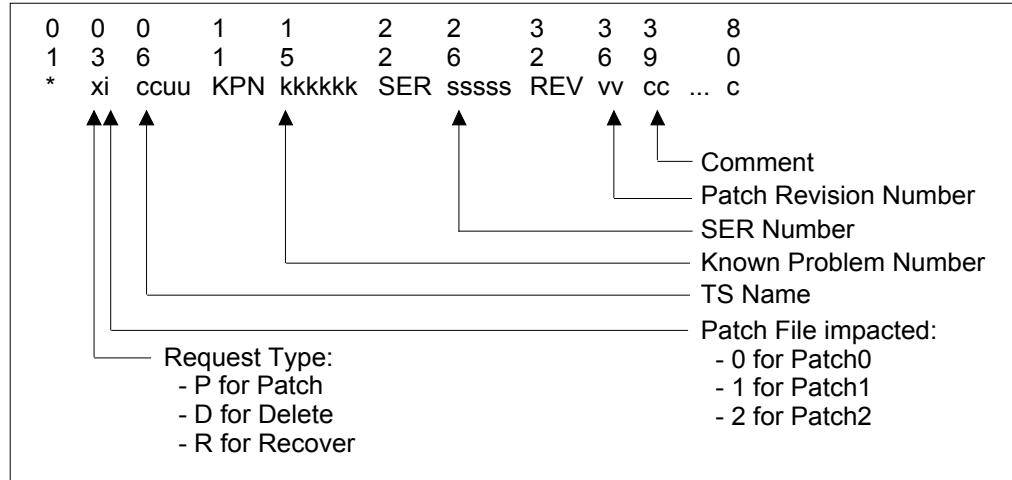
Consultation and modification of the Patch files must always be via the DIUF function PATCH. There are five kinds of request:

- Pi requests are used to enter a Patch into a Patch file.
- Di requests are used to invalidate Patches from a Patch file.
- Ri requests are used to revalidate formerly deleted Patches in a Patch file.
- LS requests are used to list information about a given Known Problem Number (KPN) stored in a Patch file.
- WR requests are used to print the contents of Patches stored in a Patch file.

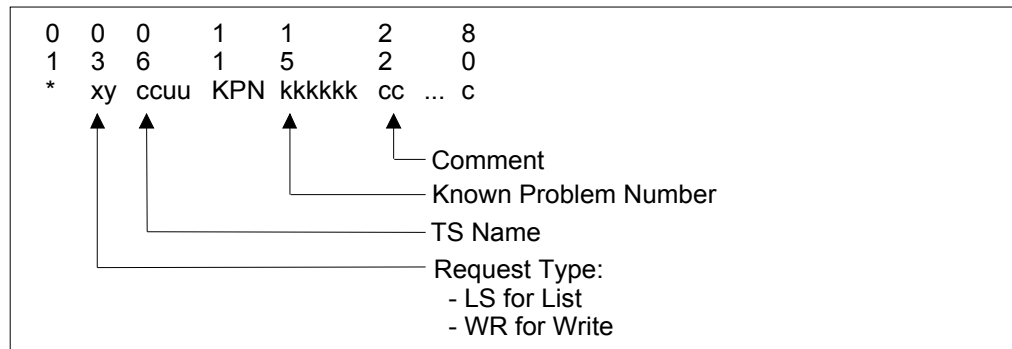


### Request Syntax

#### Pi, Di AND Ri REQUESTS



#### LS AND WR REQUESTS





**Table 10-1. Request Actions According to the State of the KPN in the CLC**

Request	State of KPN in Patch Files								
	Absent	Wrong Patch File	Source Corrected	Rev < Active	Rev < Deleted	Rev = Active	Rev = Deleted	Rev > Active	Rev > Deleted
P	OK	*** Wrong Patch File	*** Source Corr'n Patch Cannot Apply	OK Prev Vsn. Deleted	OK	* Already Active	* Patch Reinstated	*** Newer Version Found	*** Newer Version Found
D	*** Unknown KPN	*** Wrong Patch File	*** Source corr'n cannot be Deleted	OK	* Already Deleted	OK	* Already Deleted	*** Newer Version Found	*** Newer Version Found
R	*** Unknown KPN	*** Wrong Patch File	*** Source corr'n Lost?	* Already Active	OK	* Already Active	OK	*** Newer Version Found	* Newer Version Found +
LS	* Unknown KPN	=====	OK						
WR	* Unknown KPN	=====	OK						

This Table represents a Patch history of the installation kept by the system.

**KEY**

KPN=Known Problem Number.

CLC=Cumulative List of Corrections.

+ One or more newer versions have been found. A previous version (never deleted physically, only logically) is reactivated, with the result that all more recent versions are physically deleted (i.e., definitively).

< Indicates a revision older than that being applied.

> Indicates a revision more recent than that being applied.

? Indicates that the Source Correction cannot be recovered.



## 10.13 The BUILD\_TRCNCC Function

This function enables you to allocate a set of `n` `SYS.DSATRC_<lct-name>_x` on the P-Set, P2-Set or R-Set specified. The `lct-name` is the name of the Local Communication Controller related to FCP 7 (whose traces will be reported in these files), and `x=1` through `n`.

This function must be executed at the first installation of an FCP 7 Release, or if a new LCT is added to the configuration.

### Building Procedure for `SYS.DSATRC_<lct-name>_x` Files

1. Run the BUILD\_TRCNCC function with either:

```
S: DIUF [SET={P|P2|R}] COMMAND='BUILD_TRCNCC LCT=lct-name,  
      VOLUME=dsamd:dsadvc, NUMBER=n, SIZE=size ;' ;
```

or

```
S: DIUF [SET={P|P2|R}] FUNCTION=BUILD_TRCNCC ;
```

followed by introduction of the LCT, VOLUME, NUMBER, and SIZE parameters on the next screen.

SET is by default the running Set. In a P2P-Configuration, the Set may be P or P2; in an RP-Configuration, the Set may be R or P. Note that, in a PO-Configuration, this is the only step to be executed.

`lct-name` identifies the LCT whose traces will be reported in the associated `SYS.DSATRC_<lct-name>_x` files.

`dsamd` and `dsadvc` are, respectively, the media name and device class of the volume on which `SYS.DSATRC_<lct-name>_x` are to be allocated.

`n` is the number of `SYS.DSATRC_<lct-name>_x` files to be allocated, where  $2 \leq n \leq 9$ . For example, if `n=2`, then two files are allocated:

`SYS.DSATRC_<lct-name>_1` and `SYS.DSATRC_<lct-name>_2`.

SIZE is the combined size of all the `SYS.DSATRC_<lct-name>_x` files. SIZE is in blocks.

2. In RP and P2P-Configurations, run the BUILD\_TRCNCC function on the other Set, by repeating Step 1 for that Set.

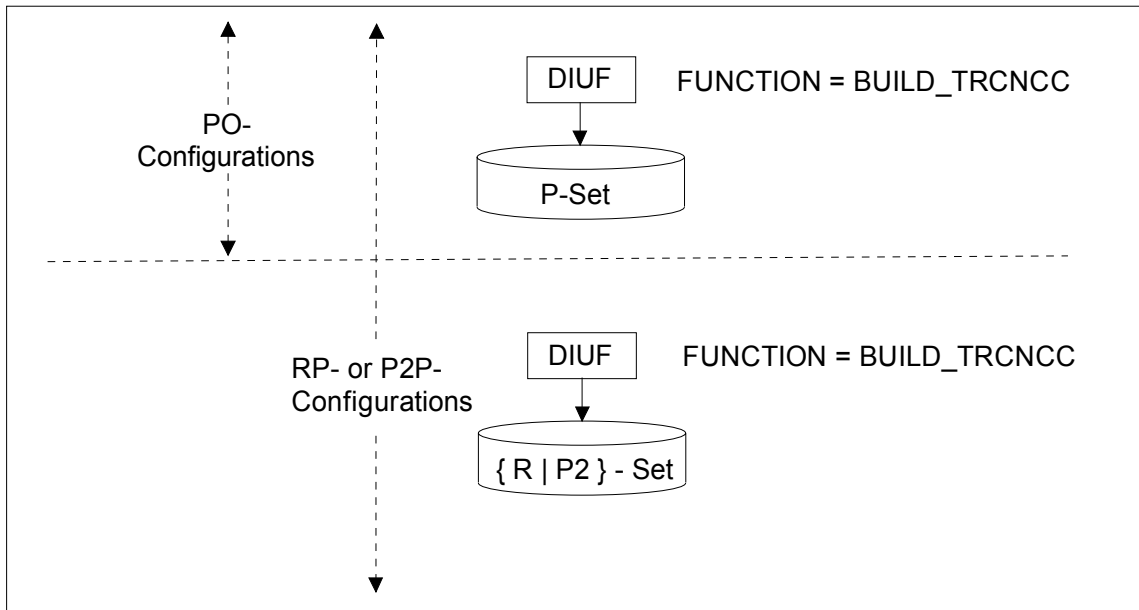


Figure 10-10. Building Procedure for SYS.DSATRC\_<let-name>\_x files





---

## 11. GCOS 7 Service Facilities (GSF)

### 11.1 Introduction

This facility is used to install and maintain all the components of the mandatory GSF (GCOS 7 Service Facilities) Domain on the P-Set, P2-Set, or R-Set, as appropriate.

The GSF tools (RIDE, ASR7, DYNDUMP, TELECONTROL7, etc.), are needed to maintain a system either on site or through Remote Maintenance Access. These tools are located in the system files which belong to the GSF Domain and are cataloged in the SYS.CATALOG file, namely:

- SYS.GSF.BINLIB
- SYS.GSF.LMLIB
- SYS.GSF.SLLIB
- SYS.GSF.SMLIB
- SYS.GSF.UFAS.

These GSF tools, **use of which is restricted to the Bull CSE**, may be accessed using the GSF function USAGE.

GSF installation may take place in two distinct sets of circumstances:

1. During initial GCOS 7 software Release installation as described in the chapter *Installation Procedure*.
2. On an existing system, during the life of a GCOS 7 software Release.

From a GSF Release Media supplied by the your Bull CSE, installation is performed on any of the user's Sets (P-, P2-, or R-Set, as appropriate to the Configuration type). The non-updated Sets may subsequently be modified with the new version by using the GIUF function LEVEL (DOMAIN=GSF).



## 11.2 GSF Tools

The IOF user may access the GSF Facility by using the GCL command ENTER\_GSF (alias GSF), which provides the following facilities:

INSTALL	is used to install the GSF Domain initially. This is done from the GSF Release Media, which can be magnetic tape or cartridge.
UPDATE	is used to install a new version of GSF on an already-installed GSF Domain, from the GSF Release Media which can be magnetic tape or cartridge.
USAGE	is used by Bull personnel to access the GSF tools, in order to maintain the system.



### CAUTION:

The Installation and Updating procedures consist of installing, or reinstalling the GSF system file Domain each time a new GSF Release becomes available. Consequently, a GSF Release Media may be considered as being equivalent to a GSF Technical Status Media.

## 11.3 GSF Release Media

A GSF Release Tape or Cartridge always has a MEDIA name in the form:

GSFxyy

where:

x stands for the GSF Release Number

yy stands for the GSF Technical Status

for example:

GSF122

identifies GSF Release 1 and Technical Status 22.

When updating the GSF Domain, the GSF components are sometimes located on a multi-Domain Release media. If this is the case, your Bull CSE will inform you accordingly.



## 11.4 The ENTER\_GSF Command

### Purpose

- Provides access to the GSF functions INSTAL, UPDATE and USAGE.
- All functions must be submitted under the SYSADMIN Project.

### Syntax

```
{ENTER_GSF | GSF} {FUNCTION={ INSTAL | UPDATE | USAGE}  
                  | COMMAND=immediate-command}  
  
[SET={P | P2 | R}] ;
```

### Parameters

FUNCTION	names a GSF function to be performed, with subsequent parameters being taken at the user's terminal; forbidden in batch mode, it is mutually exclusive with COMMAND.
COMMAND	specifies a string containing no more than one GSF function followed by its parameters.
SET	designates the Set to be updated; by default, this is:

At first installation (GSF Domain not yet initialized):

- The P-Set, in a PO or P2P Configuration.
- The R-Set, in an RP Configuration
- The running Set, in all other cases.

Note that the USAGE function does not require the COMMAND and SET parameters.



## 11.5 Updating

### Before Beginning

Before applying a new GSF Technical Status, ensure that the SYS.GSF.BINLIB is not in use otherwise the following message will appear:

```
"WAITING FOR FILE SYS.GSF.BINLIB"
```

This operation may be performed on any one of the user's Sets, as appropriate to the requirements of the installation, and involves the installation of a new GSF Release Media on **an already-installed GSF Domain**.

The site's other Sets will be updated using the GIUF function LEVEL, but for the GSF Domain only; i.e., LEVEL ... DOM=GSF. In updating operations, the following GSF function UPDATE **must** be used.

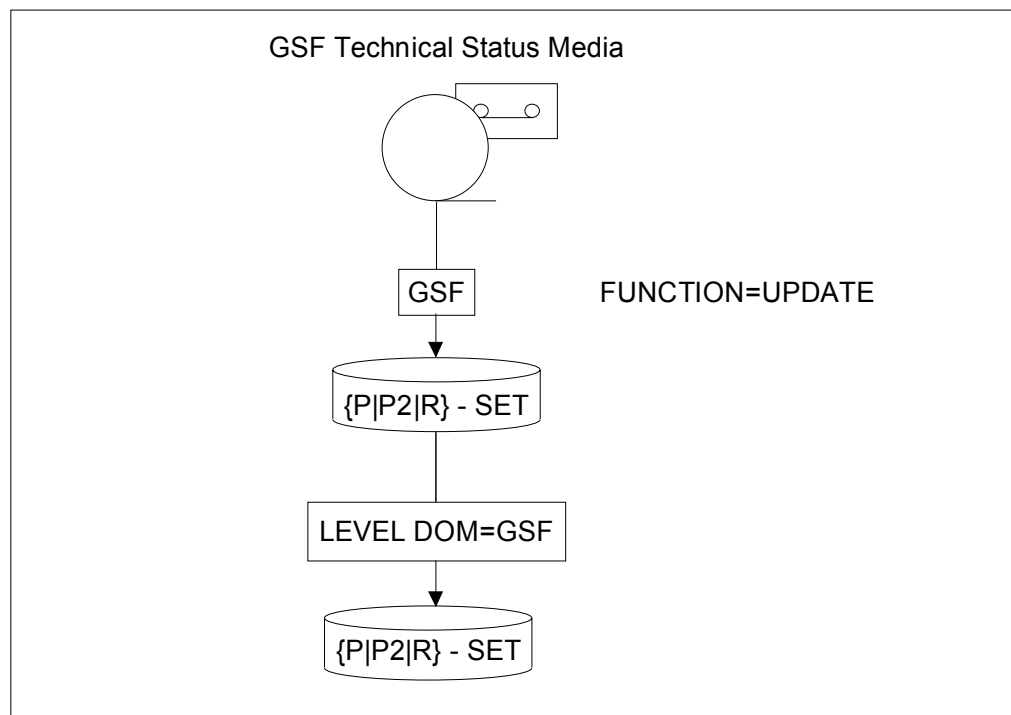


Figure 11-1. Updating the GSF Domain



### Updating Procedure

1. Mount the GSF Technical Status Media containing the GSF Release to be installed.

2. Proceed with installation on the desired Set, with either:

```
S: GSF [SET={P|P2|R}] COMMAND='UPDATE TSVOL=tsmd:tsdvc  
GSFVOL=gsfmd:gsfdvc;';
```

or:

```
S: GSF [SET={P|P2|R}] FUNCTION=UPDATE;
```

followed by introduction of the TSVOL and GSFVOL parameters on the following screen.

where:

TSVOL identifies the GSF Release Media's name and device class.

GSFVOL identifies the Set on which the GSF system files are to be allocated, as appropriate. Any resident disk in the Set may be used.





---

## 12. Maintenance and Recovery Procedures

### **NOTE:**

The RESTORE function, which is carried out under SIP in UTIL mode on non-XTA systems and is described in this chapter, has been replaced by the RESTORE facility which is run from the Bootstrap GCOS disk (B-DISK)

### 12.1 General Considerations

This chapter deals with the installation and maintenance requirements of fixed-disk systems, especially with regard to the placement of system files. Performance aspects are not treated, but certain general considerations should be kept in mind when planning disk allotment:

- Disk space management
- Recovery procedures
- P2-Set (or R-Set) disk and Updating procedures.

#### 12.1.1 Disk Space Management

All the data required for normal operation should be kept on line at all times, since immediate disk reconfiguration is not possible (i.e., no MOUNT processing). This implies careful disk-space management by both system managers and users; on the other hand, it permits unattended/remote operations and better disk-drive performance.



### 12.1.2 Recovery Procedures for Data Volumes

There are no such procedures specific to fixed-disk systems; but, to ensure reasonably rapid recovery from a possible disk failure, you should provide enough free space in the storage subsystem to allow user files to be stored elsewhere as well as to allow the application of "after images" before restarting operations.

To simplify disk space management, you should identify one disk drive as a potential spare; the disk may either be cleared or contain data that is already saved and not required during normal production operations. This drive may be used to restore (with the GCL procedure `RESTORE_FILE`, or `RESTORE_FILESET`) all the files previously located on the failed drive.

### 12.1.3 P2-Set/R-Set Disk and Updating Procedures

The P2P-Configuration allows the use of a second production Set as the reference for IUF updating operations. This second Set, referred to as the P2-Set, ensures both:

- Full support of parallel updating
- Easy restart in the event of P-Set failure.

In an RP-Configuration, the R-Set is used both for rollback and for ensuring consistency in Updating procedures. The R-Set is required during maintenance procedures (except for EC patching with the GIUF functions `PATCH_*`), which means that it must be restored from tape/cartridge if is not kept on line, with a consequent increase in maintenance operation time. This point is discussed in more detail below.

In multi-system configurations, Slave Systems may thus dispense completely with their own R-Set, when they are updated via the GIUF functions `SAVE_SET + RESTORE_SET` once Technical Status application has been performed on the Master System.

In this latter case, it is highly desirable that a `SAVE_DISK` of a P-Set be made, to enable restore operations using SIP, in the event of disk failure.

### 12.1.4 Tape, Cartridge, and Cassette Subsystems

Tape, cartridge, and cassette subsystems are used to make a backup of the system. Save and Restore times vary widely depending on the DPS 7000 platform and the type of backup medium.





## 12.1.5 MS/FSA Disk Subsystem



### **IMPORTANT:**

Even in a protected environment where securized RAID-1 and RAID-S disks are used, it is nonetheless important to have a P2P configuration or at least an RP configuration. A securized system cannot guarantee that files will not be damaged or even destroyed.

Three types of disk configuration may be defined: large, medium and small.

### 12.1.5.1 Large Configuration

In a "large" configuration, the P2-Set is kept on line, to reduce software updating time. Since this Set is saved after any change, it may also be used as a potential spare drive in the event of a user disk or system disk failure.

In a very large configuration, the P2-Set is permanently on line and at the same time there is a permanently identified spare disk drive, to reduce software updating time.

Since this Set is saved after any change, it may also be used as a potential spare drive in the event of a user disk or system disk failure.

### 12.1.5.2 Medium Configuration

This will usually be an RP-Configuration. A P2P-Configuration is feasible, although the full benefits of this would not be realized, partly because of the increased overhead of saving a P2-Set to and restoring a P2-Set from tape or cartridge.

Whatever the total number of drives in the system, a "medium" configuration means a disk subsystem containing only P-Set disks and user disks, without any spare drives and with a single R-Set disk image on tape/cartridge, to be restored to disk whenever needed.

### 12.1.5.3 Small Configuration

This will usually be a PO-Configuration, where there is neither P2-Set nor R-Set as a backup in case of P-Set failure.

As mentioned earlier, it is highly desirable that a SAVE\_DISK of the P-Set be made, to enable restore operations using SIP, in the event of disk failure.



## 12.2 Updating

In P2P- and RP-Configurations, most GIUF procedures need the P2-Set or the R-Set on line, to operate correctly.

### 12.2.1 Large Configuration

The P2-Set or the R-Set should normally be on line.

If the drive supporting the P2-Set or the R-Set has been used for some other purpose (e.g., recovery, or work), the System Administrator must restore the Set to the spare drive. This operation may be performed in parallel with normal operations, if a tape/cartridge drive is available for the purpose.

If the P2-Set or R-Set is updated during the procedure, it must of course be saved to tape/cartridge again.

If the P2-Set feature is used, the system may continue its production activity during the update.

### 12.2.2 Medium Configuration

The R-Set is available only on tape/cartridge, and the disk to receive it normally contains user files. The sequence of operations is as follows:

1. Save user files to tape/cartridge.
2. VOLPREP the disk.
3. Restore the R-Set to disk with the GCL procedure RESTORE\_DISK.
4. Perform the IUF functions.
5. If the R-Set has been modified by 4, resave it to tape/cartridge.
6. VOLPREP the disk.
7. Restore the user files from tape/cartridge to disk.
8. Restart the applications.

If the user files in 1) and 7) are accessed by the applications in 8), the above steps should be performed separately from normal operations.

In the event of a problem with the P-Set, the contents of a user disk will be overwritten by the restored P2-Set or R-Set; this is because SIP does not have disk test routines as powerful as those provided by GCOS 7.



### 12.2.3 Small Configuration

As mentioned elsewhere in this manual, a PO-Configuration is the least expensive, but also the least secure, type of configuration available.

When a site has installed a PO-Configuration, recovery from P-Set failure may be met by restore operations under SIP; this is explained at the end of the chapter Installation Procedure.

Any site that uses a P2P-Configuration (or RP-Configuration) should consider itself, for the purposes of this chapter, as a "medium" configuration and use the appropriate procedure above.



---

## 12.3 Slave Updating

When a site is updated using the GIUF functions `SAVE_SET + RESTORE_SET`, the P2-Set/R-Set need not be on line.

### 12.3.1 Large Configuration

The P2-Set should normally be on line. The `RESTORE_SET` function updates this disk during normal production operation, and the user may then switch to this new system.

### 12.3.2 Small or Medium Configuration

There is no P2-Set (on line). The `RESTORE_SET` function updates the running system, which should be idle in order to avoid conflicts on system files.

A single execution of the GCL procedure `SAVE_DISK` on the P-Set ensures sufficient backup when using the SIP function `RESTORE` in UTIL mode.



---

## A. System File Descriptions

### A.1 Introduction

In the following pages, certain files are described by a PREALLOC or LIBALLOC JCL statement, as appropriate, even if the TAILOR processor in fact calls a FILDUPLI-OUTALC JCL statement to create it. Constant or parametered characteristics are given in connection with the visibility of the file through the TAILOR language.

#### NOTES:

1. Throughout this Appendix, "any FSA resident disk" means either the system disk or any disk declared as resident.
2. The disk DEVCLASSES allowed are MS/FSA only for the system and resident disks.  
**System files for all Domains that are delivered already loaded are on FSA volumes. System files that are created on the user site may be on the system disk, or on FSA resident/non-resident volumes, as indicated for each file.** Wherever possible, all files making up the P-Set should be grouped together on the system disk, since this greatly facilitates management and recovery.
3. "CATALOG = n", or "CAT = n", specifies the rank "n" of the catalog to be used, as defined in the currently-active ATTACH statement.
4. Sizes of files in the Firmware Domain depend on the model of DPS 7000 and on the BSR release. The examples in Appendix A give average sizes.
5. Sizes of files in Domains other than Firmware depend on the evolution of GCOS 7 according to the Technical Status applied. The examples in Appendix A give average sizes.



6. The GCL equivalents of JCL statements referred to here are as follows:

JCL	GCL
PREALLOC	BUILD_FILE CREATE_MT_FILE MODIFY_FILE_SPACE
LIBALLOC	BUILD_LIBRARY
FILDUPLI	COPY_FILE COPY_FILESET



**IMPORTANT:**

1. User sub-files must never be placed in system libraries, as they would be lost when updating or returning to a previous Technical Status. The only exceptions to this rule are the SITE.HELP file, the SYS.DSACONF file. In these two cases only, it is recommended that suitable user objects be stored in the files concerned where they will be preserved in the event of a system update.
2. The PREALLOC/LIBALLOC JCL statements used to describe some of the files are given **for information purposes only**. They must not be used without the agreement of Bull, and then only in exceptional circumstances.
3. The files of the FW and OLTD domains contain parameters that are required by the system and they are referenced by a physical address. For this reason, they must NEVER be handled other than by using the tools provided for this purpose (FGF, TDF). Any attempt to move, delete, copy, or relocate these files, including moving them to another volume, can cause incidents that may even be catastrophic.
4. Volumes containing FW files must never be restored using REORG unless all ad hoc FGF commands are executed after the restore. You are recommended to confide these tasks to qualified Bull CSEs.



**Table A-1. List of System Files**

<b>File Name</b>	<b>Allocatable on Mirror Disk</b>
SITE.CATALOG	
SITE.CMS_*	YES
SITE.HALOCK	
SITE.HELP	
SITE.IN	
SITE.MIRLOG	
SITE.STARTUP	
SYS.BKST	
SYS.BKST[i]	
SYS.BOOT	
SYS.C.INCLUDE	
SYS.CLC	
SYS.CATALOG	
SYS.DSA*	
SYS.DSALOG[i]	
SYS.ERLOG	
SYS.FW.*	
SYS.FW.MSP (see Note)	
SYS.FW.WSP(see Note)	
SYS.GPL.MACLIB	
SYS.GSF.*	
SYS.HBINLIB	
SYS.HBINLIB2	
SYS.HCULIB	
SYS.HELP	
SYS.HLMLIB	
SYS.HLMLIB2	
SYS.HSLLIB	
SYS.HSLLIB2	
SYS.HUB (see Note)	
SYS.HUB7 (see Note)	
SYS.HUBG	
SYS.IN	
SYS.IUF	



File Name	Allocatable on Mirror Disk
SYS.JADIR	YES
SYS.JAS*	YES
SYS.JRNAL	YES
SYS.JTRVAL[i]	
SYS.KNODET	
SYS.LIB[i]	
SYS.LOGC	
SYS.OUT	
SYS.PVMF[i]	
SYS.QM*	
SYS.SITE.*	
SYS.SPDUMP	YES
SYS.SPOOL[i]	YES
SYS.SWLOG	YES
SYS.SYSDUMP	
SYS.SYSTEM	
SYS.SYSTEM2	
SYS.TVMF[i]	
SYS.URCINIT	

**NOTE:**

SYS.HUB7 is loaded by the BSR P2 and mandatory for the DPS 7000/TAXXX/TAXXXC, and DPS7000/MTx2 and up.

SYS.FW.MSP, SYS.FW.WSP, SYS.HUB and SYS.HUB7 do not exist on XTA systems





## A.2 The SITE.CATALOG File

### File Name

SITE.CATALOG

### Domain

GCOS

### Function

This file contains descriptions of the following items:

- User, project, billing, site
- Private catalogs
- Cataloged files and volumes.

### Organization

Undefined

### Size

SIZE (NBOBJ) 1940

### Location

This is a mandatory file that may be located on any FSA resident disk; it contains at least log-on data. If preallocated by TAILOR, it may be located on SYSVOL, RSDVOL, or RSDVOL1. SITE.CATALOG may be copied from any disk to any other disk by the IUF function COPY\_SITEFILES.

### EXAMPLE OF ALLOCATION:

```
CATBUILD SITE.CATALOG, DVC=MS/FSA, MD=md, SYSTEM, NBOBJ=1940;
```

□

### TAILOR Commands:

```
CATFL, SYSVOL, RSDVOL, RSDVOL1
```



---

## A.3 The SITE.CMS\_CSD File

### File Name

SITE.CMS\_CSD

### Domain

GCOS

### Function

Contains complex generation result tables that describe the configuration of GCOS 7.

### Organization

UFAS Indexed

### Size

SIZE (CI) 100

### Location

This file may be located on any FSA shared disk. If created by TAILOR, it may be located on RSDVOL, RSDVOL1, NRDVOL or NRDVOL1.

### TAILOR Commands:

CMS, SYSVOL, RSDVOL, RSDVOL1, NRDVOL, NRDVOL1

### GIUF Command:

BUILD\_CMS



## A.4 The SITE.CMS\_CDD\_NAMEi File

### File Name

SITE.CMS\_CDD\_namei

namei corresponds to the network name of a coupled system, where i=1 for one of the systems and i=2 for the other system.

### Domain

GCOS

### Function

This file is used for managing coupled systems. It contains the initialization of the dynamic state of the component corresponding to "namei".

### Organization

UFAS Indexed

### Size

SIZE (CI) 100

### Location

This file may be located on any FSA shared disk. If created by TAILOR, it may be located on SYSVOL, RSDVOL, RSDVOL1, NRDVOL or NRDVOL1.

### TAILOR Commands:

CMS, SYSVOL, RSDVOL, RSDVOL1, NRDVOL, NRDVOL1

### GIUF Command:

BUILD\_CMS



---

## A.5 The SITE.CMS\_DS\_NAME1\_NAME2 File

### File Name

SITE.CMS\_DS\_name1\_name2

### Domain

GCOS

### Function

This file is used for managing coupled systems. It contains the initialization of the structures for mutual surveillance between the members "name1" and "name2".

### Organization

NONE

### Size

SIZE (BLOCK) 5

### Location

This file may be located on any FSA shared disk. If created by TAILOR, it may be located on SYSVOL, RSDVOL, RSDVOL1, NRDVOL or NRDVOL1.

### TAILOR Commands:

CMS, SYSVOL, RSDVOL, RSDVOL1, NRDVOL, NRDVOL1

### GIUF Command:

BUILD\_CMS



## A.6 The SITE.HALOCK File

### File Name

SITE.HALOCK

### Domain

GCOS

### Function

Used with TDS-HA (TDS High Availability), this file provides the synchronization and serialization services required within a coupled systems configuration.

### Organization

NONE

### Size

SIZE (BLOCK) 4

### Location

This file may be located on any FSA disk. If created by TAILOR, it may be located on SYSVOL, RSDVOL, RSDVOL1, NRDVOL or NRDVOL1.

### TAILOR Commands:

HALOCK, SYSVOL, RSDVOL, RSDVOL1, NRDVOL, NRDVOL1

### GIUF Command:

BUILD\_HALOCK



## A.7 The SITE.HELP File

### File Name

SITE.HELP

### Domain

GCOS

### Function

This file contains site-specific help texts related to GCL commands, and standard help texts in the official local language.

### Organization

Library

### Size

SIZE (BLOCK) 170

### Location

This is a mandatory file that may be located on any FSA disk. If preallocated by TAILOR, it may be located on SYSVOL, RSDVOL, or RSDVOL1.

### EXAMPLE OF ALLOCATION:

```
LIBALLOC SL, (SITE.HELP, UNIT=BLOCK, DVC=MS/FSA,  
             MD=md, SIZE=(170, 170),  
             FILESTAT=CAT, CAT=n, CATNOW, EXPDATE=365),  
             DIRSIZE=14, MAXSIZE=1500;
```

□

### TAILOR Commands:

SITHELP, SYSVOL, RSDVOL, RSDVOL1



## A.8 The SITE.IN File

### File Name

SITE.IN

### Domain

GCOS

### Function

This file is used to store input data defined by the JCL statement \$DATA.

### Organization

Queued Linked

### Size

SIZE (BLOCK) 170

### Location

This is an optional file that may be located on any FSA disk, and must be cataloged if created. If preallocated by TAILOR, it may be located on SYSVOL, RSDVOL, or RSDVOL1 and must be cataloged accordingly.

### EXAMPLE OF ALLOCATION:

```
PREALLOC SITE.IN, UNIT=BLOCK,  
          DVC=MS/FSA, GLOBAL=(MD=md, SIZE=170),  
          LINKQD=(TYPE=NONE, BLKSIZE=4070, RECSIZE=264,  
                LTRKSZ=1, DIRSIZE=3, RECFORM=VB, NDLREC),  
          CATNOW, FILESTAT=CAT, CAT=n;
```

□

### TAILOR Commands:

SITIN, SYSVOL, RSDVOL, RSDVOL1



## A.9 The SITE.MIRLOG File (if applicable)

**File Name**

SITE.MIRLOG

**Domain**

GCOS

**Function**

This file keeps a trace of internal activities on mirrored disks.

**Organization**

NONE

**Size**

SIZE (BLOCK) 40

**Location**

This file may be located on any FSA non-MIRROR volume. If created by TAILOR, it may be located on SYSVOL, RSDVOL, RSDVOL1, NRDVOL or NRDVOL1.

**TAILOR Commands:**

MIRLOG, SYSVOL, RSDVOL, RSDVOL1, NRDVOL, NRDVOL1

**GIUF Command:**

BUILD\_MIRLOG





## A.10 The SITE.STARTUP File

### File Name

SITE.STARTUP

### Domain

GCOS

### Function

This file contains site-specific JCL and GCL sequences for automatic insertion at the beginning of batch and IOF jobs.

### Organization

Library

### Size

SIZE (BLOCK) 170

### Location

This is a mandatory file that may be located on any FSA disk, and must be cataloged. If preallocated by TAILOR, it may be located on SYSVOL, RSDVOL, or RSDVOL1. The file may be copied from any disk to any other disk, using the IUF function COPY\_SITEFILES.



**EXAMPLE OF ALLOCATION:**

```
CATALOG      SITE.STARTUP, SHARE=DIR, DUALSHR=ONEWRITE,  
              CATALOG=n;
```

```
LIBALLOC SL,  (SITE.STARTUP, UNIT=BLOCK, MD=md,  
              DVC=MS/FSA, SIZE=(170,170),  
              FILESTAT=CAT, CAT=n, CATNOW),  
              DIRSIZE=8, MAXSIZE=850;
```

□

**TAILOR Commands:**

```
STUP, SYSVOL, RSDVOL, RSDVOL1
```



## A.11 THE SYS.BKST File

### File Name

SYS.BKST

### Domain

GCOS

### Function

This file provides system backing store, to contain:

- The crash area
- The non-releasable area loaded at restore time
- A temporary paging backing store (which may also be located in SYS.BKSTi).

### Organization

Non-standard FBO.

### Size

SIZE (BLOCK) 15980

The size of SYS.BKST is independent of memory size. However, it does depend on the number of active jobs. The larger the memory size, the more active jobs there will be.

When built on-site, the default SIZE is 15980 blocks, irrespective of the amount of memory installed. The maximum possible size of a BKST is 65535 blocks.



---

### **Location**

This is a mandatory file, to be located on the system disk. The file may have up to 2 extents but must be mono-volume.

If this file is allocated with two extents, its extent #1 must be of a minimum size: 32 MB in order to be able to contain the crash area.

The SYS.BKST file cannot be allocated on mirror disks.

It may be allocated by either:

- TAILOR
- The IUF function BUILD\_BKST.

### **TAILOR Commands:**

BKST, SYSVOL

### **GIUF Command:**

BUILD\_BKST



## A.12 The SYS.BKST[i] File(s)

### File Name

SYS.BKST[i] (where i = from 1 to 15)

### Domain

GCOS

### Function

These files provide paging backing store for temporary use, intended to contain, in virtual memory format, the temporary segments of all types belonging to job steps.

The size of BKST(i) files is related to the number of jobs running and to the applications themselves. It must also take account of the available memory that determines the number of segment SWAPIN/SWAPOUTs.

### Organization

Non-standard FBO.

### Size

SIZE (BLOCK) 4250

### Location

These are optional files that may be located on any FSA resident or non-resident disk, and are preallocated by the user with the IUF function BUILD\_BKST.

The SYS.BKSTi files may be allocated on SYSVOL, RSDVOL, RSDVOL1, NRDVOL, or NRDVOL1. They may have multiple extents but must be mono-volume.



---

The SYS.BKSTi files cannot be allocated on mirror disks.

**NOTE:**

The sum of the extents of all the SYS.BKSTi files must be < 16. To improve processing performance the user is advised to allocate several SYS.BKSTi files, each with a few extents, and on different volumes.

**TAILOR Commands:**

BKST1, SYSVOL, RSDVOL, RSDVOL1, NRDVOL, NRDVOL1

**GIUF Command:**

BUILD\_BKST



**Backing Store Classification**

**Table A-2. Backing Store Classification**

		<b>SYS. BKST</b>	<b>SYS. BKSTi</b>	<b>SYS. TVFMi</b>	<b>SYS. PVMFi</b>	<b>SYS. LIBi</b>
SYSTEM J = 0 TYPE = 0	- Initial Segments - Segment Created Dynamically	X X	X			
Step with Non-preinitialized LM	- Initial Segment - Segment Created Dynamically - Temporary VMF	X	X X			
Step with Preinitialized LM	- Initial Segment - Segment Created Dynamically - Temporary VMF	X	X			X
Step with REPEAT Option	- Initial Segment - Segment Created Dynamically - Temporary VMF - Checkpoints	X	X			X X
Permanent Virtual Memory Files					X	
Sharable Modules						X



---

## A.13 The SYS.BOOT File

### **File Name**

SYS.BOOT

### **Domain**

GCOS

### **Function**

This file is used when starting a system session.

### **Organization**

Non-standard

### **Size**

SIZE (BLOCK) 650

### **Location**

This is a mandatory file that must be located on the system disk. If preallocated by TAILOR, it is located on SYSVOL, and must be cataloged accordingly.

### **TAILOR Commands:**

Not visible to TAILOR language.





## A.14 The SYS.C.INCLUDE File

### File Name

SYS.C.INCLUDE

### Domain

GCOS

### Function

This file contains the standard C files and C source header files to be included in C language user programs.

### Organization

Library

### Sizes

SIZE (BLOCK)	750
INCRSIZE (BLOCK)	170
MAXSIZE (BLOCK)	2000

### Location

This is a mandatory file that may be located on any FSA disk (preferably the system disk). If copied by TAILOR, it is located on SYSVOL.



### IMPORTANT:

This file must not contain any user sub-files.



**EXAMPLE OF ALLOCATION:**

```
PREALLOC SYS.C.INCLUDE, DVC=MS/FSA, UNIT=BLOCK,  
GLOBAL=(MD=md, SIZE=750),  
LINKQD=(NONE=(BLKSIZE=4070, RECSIZE=264,  
RECFORM=VB, TYPE=SL, NDLREC, COMPACT, DIRSIZE=8,  
LTRKSZ=2, MAXSIZE=2000)), INCRSIZE=170,  
FILSTAT=CAT, CATNOW, CAT=n;
```

□

**TAILOR Commands:**

Not visible to TAILOR language.



## A.15 The SYS.CATALOG File

### File Name

SYS.CATALOG

### Domain

GCOS

### Function

This file is the system catalog, used for cataloging system files.

### Organization

Non-standard.

### Size

SIZE (NBOBJ) 1940

### Location

This is a mandatory file that must be located on the system disk.

### EXAMPLE OF ALLOCATION:

```
CATBUILD SYS.CATALOG, DVC=MS/FSA, MD=md, NBOBJ=1940;
```

□

### TAILOR Commands:

Not visible to TAILOR language.



---

## A.16 The SYS.CLC File

### **File Name**

SYS.CLC

### **Domain**

GCOS

### **Function**

Used for cartridge library management.

### **Organization**

Linked Queued

### **Size**

SIZE (BLOCK) 510

### **Location**

This is an optional file that may be located on any FSA disk. If created by TAILOR, it may be located on SYSVOL, RSDVOL, RSDVOL1, NRDVOL, or NRDVOL1.

### **TAILOR Commands:**

CLC, SYSVOL, RSDVOL, RSDVOL1, NRDVOL, NRDVOL1

### **GIUF Command:**

BUILD\_CLC



## A.17 The SYS.DSABLIB File

### File Name

SYS.DSABLIB

### Domain

DSA

### Function

This file contains all the binary input necessary to CNS7 administration and the FCP 7 image.

### Organization

Library

### Size

SIZE (BLOCK) 4500

### Location

This is an optional file that may be located on any FSA disk. If required, it is preallocated and copied by TAILOR on SYSVOL, RSDVOL, or RSDVOL1.

### EXAMPLE OF ALLOCATION:

```
LIBALLOC BIN, (SYS.DSABLIB, UNIT=BLOCK, MD=md, DVC=MS/FSA,  
              SIZE=(4500, 1700),  
              CATNOW, FILESTAT=CAT, CAT=n),  
              MEMBERS=6000, MAXSIZE=15000;
```

□

### TAILOR Commands:

DSAFILES, SYSVOL, RSDVOL, RSDVOL1



---

## A.18 The SYS.DSACMD File

### File Name

SYS.DSACMD

### Domain

DSA

### Function

This file contains certain telecoms-specific commands sent to the system from other DSA systems in the network. A separate subfile is created to contain each such command together with the response(s) it receives.

### Organization

Library

### Size

SIZE (BLOCK) 510

### Location

This is an optional file that may be located on any FSA disk. If allocated by the TAILOR functions BUILD P and BUILD P2, it may be located on SYSVOL, RSDVOL, or RSDVOL1.



**EXAMPLE OF ALLOCATION:**

```
LIBALLOC BIN (SYS.DSACMD, UNIT=BLOCK, MD=md, DVC=MS/FSA,  
             SIZE=(510, 680), EXPDATE=365, CATNOW,  
             FILESTAT=CAT, CAT=n),  
             MEMBERS=2000, MAXSIZE=3300;
```



**TAILOR Commands:**

DSAFILES, SYSVOL, RSDVOL, RSDVOL1



## A.19 The SYS.DSACONF File

### File Name

SYS.DSACONF

### Domain

DSA

### Function

This file contains user input relative to Network Administration.

### Organization

Library

### Size

SIZE (BLOCK) 340

### Location

This is an optional file that may be located on any FSA disk. If preallocated and copied by TAILOR, it may be located on SYSVOL, RSDVOL, or RSDVOL1.

### EXAMPLE OF ALLOCATION:

```
LIBALLOC SL, (SYS.DSACONF, UNIT=BLOCK, MD=md, DVC=MS/FSA,  
             SIZE=(340, 340),  
             CATNOW, FILESTAT=CAT, CAT=n),  
             COMPACT, MEMBERS=2000, MAXSIZE=6000;
```

□

### TAILOR Commands:

DSAFILES, SYSVOL, RSDVOL, RSDVOL1





## A.20 The SYS.DSACORE File

### File Name

SYS.DSACORE

### Domain

DSA

### Function

This file contains core images of the CNS7 systems. Its contents are created by CNS7 generation to be used as input to CNS7 system loading.

### Organization

Library

### Size

SIZE (BLOCK) 340

### Location

This is an optional file that may be located on any FSA disk. If required, it is preallocated by TAILOR on SYSVOL, RSDVOL, or RSDVOL1.

### EXAMPLE OF ALLOCATION:

```
LIBALLOC BIN, (SYS.DSACORE, UNIT=BLOCK, MD=md, DVC=MS/FSA,  
              SIZE=(340, 680),  
              CATNOW, FILESTAT=CAT, CAT=n),  
              MAXSIZE=7200, DIRSIZE=7;
```

□

### TAILOR Commands:

DSAFILES, SYSVOL, RSDVOL, RSDVOL1



---

## A.21 The SYS.DSADUMP File

### File Name

SYS.DSADUMP

### Domain

DSA

### Function

This is a "post-mortem" dump file that is used for output when a DNS, CNS7 or FCP 7 system aborts.

### Organization

Library

### Size

SIZE (BLOCK) 7200

Note, for FCP 7 a minimum size of 7200 blocks is required.

### Location

This is an optional file that may be located on any FSA disk. If required, it is preallocated by TAILOR on SYSVOL, RSDVOL, or RSDVOL1.



**EXAMPLE OF ALLOCATION:**

```
LIBALLOC BIN, (SYS.DSADUMP, UNIT=BLOCK, MD=md, DVC=MS/FSA,  
              SIZE=(7200, 7200)  
              CATNOW, FILESTAT=CAT, CAT=n),  
              MAXSIZE=72000, DIRSIZE=7;
```



**TAILOR Commands:**

```
DSAFILES, SYSVOL, RSDVOL, RSDVOL1
```



## A.22 The SYS.DSALMLIB File

### File Name

SYS.DSALMLIB

### Domain

DSA

### Function

This file contains the Load Modules H\_DIUF\_MNPTF and H\_DIUF\_GCL used by the DIUF Factory.

### Organization

Library

### Size

SIZE (BLOCK) 170

### Location

This is an optional file that may be located on any FSA disk. If required, it is preallocated and copied by TAILOR on SYSVOL, RSDVOL, or RSDVOL1.

### EXAMPLE OF ALLOCATION:

```
LIBALLOC LM, (SYS.DSALMLIB, UNIT=BLOCK, MD=md, DVC=MS/FSA,  
             SIZE=(170, 85),  
             CATNOW, FILESTAT=CAT, CAT=n),  
             MEMBERS=100;
```

□

### TAILOR Commands:

DSAFILES, SYSVOL, RSDVOL, RSDVOL1



## A.23 The SYS.DSALOG1 File

### **File Name**

SYS.DSALOG1

### **Domain**

DSA

### **Function**

This file contains AEP (Administrative Exchange Protocol) records sent by the NADs (Network ADministrators) of other DSA systems in the network.

### **Organization**

UFAS Relative

### **Size**

SIZE (BLOCK) 680

### **Location**

This is an optional file that may be located on any FSA disk. If required, it is preallocated by TAILOR on SYSVOL, RSDVOL, or RSDVOL1.



**EXAMPLE OF ALLOCATION:**

As SYS.DSALOG1 must be cataloged with specific sharability attributes, its creation requires two steps:

```
CATALOG SYS.DSALOG1 SHARE=ONEWRITE CATALOG=n;

PREALLOC SYS.DSALOG1, UNIT=BLOCK, DVC=MS/FSA,
          GLOBAL=(MD=md, SIZE=680),
          UFAS=(RELATIVE=(CISIZE=4096, RECSIZE =200,
          RECFORM=VB, NBULBL=1)),
          FILESTAT=CAT, CAT=n;
```

□

**TAILOR Commands:**

DSALOG, SYSVOL, RSDVOL, RSDVOL1



## A.24 The SYS.DSALOG[i] Files

### **File Name**

SYS.DSALOGi (where i = from 2 to 10)

### **Domain**

DSA

### **Function**

These files contain AEP (Administrative Exchange Protocol) records sent by NADs (Network ADministrators) from other DSA systems in the network.

### **Organization**

UFAS Relative

### **Sizes**

Chosen by user.

### **Location**

These are optional files created by the user that may be located on any disk.



**EXAMPLE OF ALLOCATION:**

As SYS.DSALOGi must be cataloged with specific sharability attributes, file creation requires the two following steps:

```
CATALOG SYS.DSALOGi SHARE=ONEWRITE CATALOG=n;

PREALLOC SYS.DSALOGi, UNIT=BLOCK, DVC=MS/FSA,
GLOBAL=(MD=md, SIZE=size),
UFAS=(RELATIVE=(CISIZE=4096, RECSIZE =200,
RECFORM=VB, NBULBL=1)),
FILESTAT=CAT, CAT=n, CATNOW;
```

□

**TAILOR Commands:**

Not within the scope of TAILOR.





## A.25 The SYS.DSASLIB File

### File Name

SYS.DSASLIB

### Domain

DSA

### Function

This file contains all the source input necessary for DNS and/or CNS7 and/or FCP 7 administration, and the DIUF factory.

### Organization

Library

### Size

SIZE (BLOCK) 850

### Location

This is an optional file that may be located on any FSA resident disk. If required, it is preallocated and copied by TAILOR on SYSVOL, RSDVOL, or RSDVOL1.

### EXAMPLE OF ALLOCATION:

```
LIBALLOC SL, (SYS.DSASLIB, UNIT=BLOCK, MD=md, DVC=MS/FSA
             SIZE=(850, 340),
             CATNOW, FILESTAT=CAT, CAT=n),
             MAXSIZE=15000, MEMBERS=1000;
```

□

### TAILOR Commands:

DSAFILES, SYSVOL, RSDVOL, RSDVOL1



---

## A.26 The SYS.DSATRC\_<LCT-NAME>\_[i] Files (Specific to FCP 7)

### File Name

SYS.DSATRC\_XXXX\_i (where XXXX = the LCT name, and  
i = 1 to 9)

### Domain

DSA

### Function

These files contain FCP 7 traces of the associated LCT (Local Communication Controller).

### Organization

UFAS Sequential

### Sizes

Chosen by user.

### Location

These files are mandatory if the FCP 7 product is installed.

They are created by the user with the IUF function BUILD\_TRCNCC as described in the chapter *DSA Installation and Updating Facility*, and may be allocated on any disk.



**EXAMPLE OF ALLOCATION:**

As a SYS.DSATRC\_XXXX\_i file must be cataloged with specific sharability attributes, its creation requires two steps:

```
CATALOG SYS.DSATRC_EA91_3 SHARE=ONEWRITE CATALOG=n;

PREALLOC SYS.DSATRC_EA91_3, UNIT=BLOCK, DVC=MS/FSA,
GLOBAL=(MD=md, SIZE=size),
UFAS=(SEQ=(CISIZE=25088, RECSIZE =4104,
RECFORM=F, NBULBL=0)),
FILESTAT=CAT, CAT=n, CATNOW;
```

□

**TAILOR Commands:**

Not within the scope of TAILOR.

**DIUF Command:**

BUILD\_TRCNCC with the following parameters:

LCT=lct-name, VOLUME=md:dvc, NUMBER=n, SIZE=size.



## A.27 The SYS.DSATRCIDT\_NXYY File (Specific to an FCP 7 Technical Status)

### File Names

SYS.DSATRCIDT\_<fcp-ts> (<fcp-ts>=Nxxy for not XTA systems and  
<fcp-ts> = Dxxy for XTA systems)

### Domain

DSA

### Function

This file contains input data necessary for the management of FCP 7 traces of a particular FCP 7 Technical Status.

### Organization

UFAS Indexed

### Size

variable

### Location

This file is mandatory if the associated FCP 7 Technical Status is installed.

This file is automatically restored on the volume containing the SYS.DSASLIB file, during the installation of its associated FCP 7 Technical Status. The Technical Status is installed using the DIUF functions INSTAL and UPDATE, the IUF functions LEVEL and RESTORE\_SET, the IUF installation jobs INSTAL\_ALL, INSTAL\_PSET and INSTAL\_RSET, and by TAILOR.

This file must be allocated on any resident or non-resident FSA volume.



**EXAMPLE OF ALLOCATION:**

The SYS.DSATRCIDT\_<fcp-ts> file must be cataloged with the SHARE=ONEWRITE attribute and so its creation requires two steps:

```
CATALOG SYS.DSATRCIDT_<fcp-ts> SHARE=ONEWRITE CATALOG=n;  
  
PREALLOC SYS.DSATRCIDT_<fcp-ts>, UNIT=BLOCK, DVC=MS/FSA,  
GLOBAL=(MD=md, SIZE=340), INCRSIZE=170,  
UFAS=(INDEXED=(CISIZE=1356, RECSIZE=256,  
RECFORM=V, KEYLOC=1, KEYSIZE=4))  
FILESTAT=CAT, CAT=n, CATNOW;
```

□

**DIUF, GIUF, and TAILOR Commands:**

Not within the scope of TAILOR, DIUF, or GIUF.



---

## A.28 The SYS.ERLOG File

### **File Name**

SYS.ERLOG

### **Domain**

GCOS

### **Function**

This file contains a trace of hardware and system errors that occur during a system session.

### **Organization**

Non-standard.

### **Size**

SIZE (BLOCK) 85

### **Location**

This is a mandatory file that may be located on any disk. Its recommended location is on the system disk, if the other logging files are not located there. If preallocated by TAILOR, it may be located on SYSVOL, RSDVOL, or RSDVOL1.



**EXAMPLE OF ALLOCATION:**

As SYS.ERLOG must be cataloged with specific sharability attributes, its creation requires two steps:

```
CATALOG SYS.ERLOG SHARE=ONEWRITE CATALOG=n;
```

```
PREALLOC SYS.ERLOG, DVC=MS/FSA, INCRSIZE=170,  
GLOBAL=(MD=md, SIZE=85),  
UNIT=BLOCK, NONE=(BLKSIZE=8192),  
FILESTAT=CAT, CAT=n, CATNOW;
```



**TAILOR Commands:**

```
ERLOG, SYSVOL, RSDVOL, RSDVOL1
```



---

## A.29 The SYS.FW.BINLIB File

### File Name

SYS.FW.BINLIB

### Domain

FW

### Function

This file contains binary firmware objects.

### Organization

Library

### Sizes

SIZE (BLOCK)	14000
INCRSIZE (BLOCK)	10000

### Location

This file may be located on any FSA resident disk. All SYS.FW.\* queued linked (library) files must be located on the same volume.





## A.30 The SYS.FW.CCULIB File (not applicable for XTA systems)

### File Name

SYS.FW.CCULIB

### Domain

FW

### Function

This file contains binary firmware objects.

### Organization

Library

### Sizes

SIZE (BLOCK)	250
INCRSIZE (BLOCK)	5000

### Location

This file may be located on any FSA resident disk. All SYS.FW.\* queued linked (library) files must be located on the same volume.



---

## A.31 The SYS.FW.GENLIB File (not applicable for XTA systems)

### File Name

SYS.FW.GENLIB

### Domain

FW

### Function

This file contains configuration-dependent firmware files produced by the FIRMGEN function.

### Organization

Library

### Sizes

File sizes are model dependent.

SIZE (BLOCK)	550
INCRSIZE (BLOCK)	5000

### Location

This file may be located on any FSA resident disk. All SYS.FW.\* queued linked (library) files must be located on the same volume.



## A.32 The SYS.FW.LMLIB File

### File Name

SYS.FW.LMLIB

### Domain

FW

### Function

This file contains firmware generation and updating Load Modules.

### Organization

Library

### Sizes

SIZE (BLOCK)	2700
INCRSIZE (BLOCK)	4100

### Location

This file may be located on any FSA resident disk. All SYS.FW.\* queued linked (library) files must be located on the same volume.



---

### A.33 The SYS.FW.MSP (not applicable for XTA systems)

**File Name**

SYS.FW.MSP

**Domain**

FW

**Function**

This file contains the MSP firmware controller. It is created and managed by BLOAD only.

**Organization**

Non-standard.

**Sizes**

SIZE (BLOCK)	153
DBLK	1400

**Location**

On each system disk.



## A.34 The SYS.FW.NFTLIB File (not applicable for XTA Systems)

### File Name

SYS.FW.NFTLIB

### Domain

FW

### Function

This file is used to store Central Subsystem non-functional tests.

### Organization

Binary Library

### Sizes

SIZE (BLOCK)	4000
INCRSIZE (BLOCK)	5000

### Location

This file may be located on any FSA resident disk. All SYS.FW.\* queued linked (library) files must be located on the same volume.



---

## A.35 The SYS.FW.PATCH File (not applicable for XTA Systems)

### File Name

SYS.FW.PATCH

### Domain

FW

### Function

This file is used to store firmware EC and OPT patches.

### Organization

Library

### Sizes

SIZE (BLOCK)	125
INCRSIZE (BLOCK)	375

### Location

This file may be located on any FSA resident disk. All SYS.FW.\* queued linked (library) files must be located on the same volume.



## A.36 The SYS.FW.PRT File

### File Name

SYS.FW.PRT

### Domain

FW

### Function

This file contains output produced by FGF functions.

### Organization

Library

### Sizes

SIZE (BLOCK)	500
INCRSIZE (BLOCK)	9700

### Location

This file may be located on any FSA resident disk. All SYS.FW.\* queued linked (library) files must be located on the same volume.



---

## A.37 The SYS.FW.SLLIB File

### File Name

SYS.FW.SLLIB

### Domain

FW

### Function

This file contains JCL and SL data used for firmware generation.

### Organization

Library

### Sizes

SIZE (BLOCK)	1000
INCRSIZE (BLOCK)	5000

### Location

This file may be located on any FSA resident disk. All SYS.FW.\* queued linked (library) files must be located on the same volume.





## A.38 The SYS.FW.WSP (not applicable for XTA Systems)

### File Name

SYS.FW.WSP

### Domain

FW

### Function

This file contains the firmware controller (WSP-A, WSP-R or WSP depending on the type of disk subsystem). It is created and managed by BLOAD only.

### Organization

Non-standard.

### Sizes

SIZE (BLOCK)	153
DBLK	1400

### Location

On the system disk.



## A.39 The SYS.GPL.MACLIB File

### File Name

SYS.GPL.MACLIB

### Domain

GCOS

### Function

This file contains macro definitions used in connection with the GPL language.

### Organization

Library

### Size

SIZE (BLOCK) 510

### Location

This mandatory file may be located on any FSA disk. If copied by TAILOR, this file is located on SYSVOL.

### EXAMPLE OF ALLOCATION:

```
PREALLOC SYS.GPL.MACLIB, UNIT=BLOCK, DVC=MS/FSA,  
GLOBAL=(MD=md, SIZE=510),  
LINKQD=(NONE=(BLKSIZE=4070, RECSIZE =1024,  
TYPE=BIN, LTRKSZ=1, NDLREC,DIRSIZE=9,  
RECFORM=VB)), INCRSIZE=170,  
CATNOW, FILESTAT=CAT, CAT=n;
```

□

### TAILOR Commands:

Transparent to TAILOR language.



## A.40 The SYS.GSF.BINLIB File

### File Name

SYS.GSF.BINLIB

### Domain

GSF

### Function

This file contains binary objects for Service components.

### Organization

Library

### Sizes

SIZE (BLOCK)	170
INCRSIZE (BLOCK)	170

### Location

This is a mandatory file that may be located on any FSA resident disk. If copied by TAILOR, it is located on SYSVOL.



### IMPORTANT:

This file must under no circumstances contain any user sub-files.

### EXAMPLE OF ALLOCATION:

```
LIBALLOC BIN (SYS.GSF.BINLIB, UNIT=BLOCK, MD=md, DVC=MS/FSA,  
             SIZE=(170, 170), EXPDATE=365, FILESTAT=CAT, CATNOW,  
             CAT=n), MEMBERS=1000;
```



### TAILOR Commands:

```
GSFGEN, SYSVOL, RSDVOL, RSDVOL1.
```



## A.41 The SYS.GSF.LMLIB File

### File Name

SYS.GSF.LMLIB

### Domain

GSF

### Function

This file contains Load Modules for Service components.

### Organization

Library

### Sizes

SIZE (BLOCK)	170
INCRSIZE (BLOCK)	170

### Location

This is a mandatory file that may be located on any FSA resident disk. If copied by TAILOR, it is located on SYSVOL.



### IMPORTANT:

This file must under no circumstances contain any user sub-files.



**EXAMPLE OF ALLOCATION:**

```
PREALLOC SYS.GSF.LMLIB, DVC=MS/FSA, UNIT=BLOCK,  
GLOBAL=(MD=md, SIZE=170),  
LINKQD=(TYPE=LM, BLKSIZE=7168, RECSIZE=1024,  
RECFORM=FB, DIRSIZE=7, LTRKSIZE =1, MAXSIZE=10625,  
NDLREC), INCRSIZE=170, FILESTAT=CAT, CATNOW, CAT=n;
```

□

**TAILOR Commands:**

```
GSFGEN, SYSVOL, RSDVOL, RSDVOL1.
```



## A.42 The SYS.GSF.SLLIB File

### File Name

SYS.GSF.SLLIB

### Domain

GSF

### Function

This file contains source members for Service components as well as for GSF Facilities.

### Organization

Library

### Sizes

SIZE (BLOCK)	170
INCRSIZE (BLOCK)	170

### Location

This is a mandatory file that may be located on any FSA resident disk. If copied by TAILOR, it is located on SYSVOL.



### IMPORTANT:

This file must under no circumstances contain any user sub-files.



**EXAMPLE OF ALLOCATION:**

```
LIBALLOC SL (SYS.GSF.SLLIB, UNIT=BLOCK, MD=md, DVC=MS/FSA,  
            SIZE=(170, 170), EXPDATE=365,  
            FILESTAT=CAT, CATNOW, CAT=n), MEMBERS=3000;
```



**TAILOR Commands:**

```
GSFGEN, SYSVOL, RSDVOL, RSDVOL1.
```



## A.43 The SYS.GSF.SMLIB File

### File Name

SYS.GSF.SMLIB

### Domain

GSF

### Function

This file contains Sharable Modules and Linked Units used by Service components.

### Organization

Library

### Sizes

SIZE (BLOCK)	170
INCRSIZE (BLOCK)	170

### Location

This is a mandatory file that may be located on any FSA resident disk. If copied by TAILOR, it is located on SYSVOL.



### IMPORTANT:

This file must under no circumstances contain any user sub-files.





**EXAMPLE OF ALLOCATION:**

```
LIBALLOC SM (SYS.GSF.SMLIB, UNIT=BLOCK, MD=md, DVC=MS/FSA,  
            SIZE=(170, 170), EXPDATE=365,  
            FILESTAT=CAT, CATNOW, CAT=n), MEMBERS=100;
```



**TAILOR Commands:**

```
GSFGEN, SYSVOL, RSDVOL, RSDVOL1.
```



---

## A.44 The SYS.GSF.UFAS File

### File Name

SYS.GSF.UFAS

### Domain

GSF

### Function

This file contains items for use by Service components.

### Organization

UFAS Indexed

### Size

SIZE (BLOCK) 114

### Location

This is a mandatory file that may be located on any FSA resident disk. If copied by TAILOR, it is located on SYSVOL.



**EXAMPLE OF ALLOCATION:**

```
PREALLOC SYS.GSF.UFAS, DVC=MS/FSA, UNIT=BLOCK,  
GLOBAL=(MD=md, SIZE=114), INCRSIZE=57,  
UFAS=(INDEXED=(CISIZE=12288, RECSIZE=3700,  
RECFORM=V, KEYLOC=1, KEYSZ=6,  
CIFSP=20, CAFSP=20)), MAXEXT=15, EXPDATE=365,  
FILESTAT=CAT, CATNOW, CAT=n;
```



**TAILOR Commands:**

```
GSFGEN, SYSVOL, RSDVOL, RSDVOL1.
```



---

## A.45 The SYS.HBINLIB File

### File Name

SYS.HBINLIB

### Domain

GCOS

### Function

This file contains binary objects such as:

- GCL procedures
- Forms
- Schemas (data dictionary).

### Organization

Library

### Sizes

SIZE (BLOCK)	1870
INCRSIZE (BLOCK)	170
MAXSIZE (BLOCK)	4420

### Location

The SYS.HBINLIB file must be allocated on the system disk.



**EXAMPLE OF ALLOCATION:**

```
PREALLOC SYS.HBINLIB, UNIT=BLOCK,  
          DVC=MS/FSA, GLOBAL=(MD=md, SIZE=1870),  
          LINKQD=(NONE=(TYPE=BIN, BLKSIZE=8166,  
          RECSIZE=1024, LTRKSZ=1, MAXSIZE=4420,  
          DIRSIZE=10, NDLREC, RECFORM =VB)),  
          INCRSIZE=170, CATNOW, FILESTAT=CAT, CAT=n;
```

□

**TAILOR Commands:**

Not visible to TAILOR language.



## A.46 The SYS.HBINLIB2 File

### File Name

SYS.HBINLIB2

### Domain

OLTD

### Function

This file contains the On-Line Test and Diagnostic (OLTD) binary modules.

### Organization

Library

### Size

SIZE (BLOCK) 500

### Location

This is a mandatory file that must be located on the FSA disk that contains the SYS.CATALOG file.

### EXAMPLE OF ALLOCATION:

```
LIBALLOC BIN, (SYS.HBINLIB2, UNIT=BLOCK, MD=md, DVC=MS/FSA,  
              SIZE=(500, 85), CATNOW, FILESTAT=CAT,  
              CAT=n), MEMBERS=20;
```

□

### TAILOR Commands:

Not visible to TAILOR language.



## A.47 The SYS.HCULIB File

### **File Name**

SYS.HCULIB

### **Domain**

GCOS

### **Function**

This file contains system Compile Units.

### **Organization**

Library

### **Size**

SIZE (BLOCK) 170

### **Location**

This is a mandatory file that may be located on any FSA disk. If copied by TAILOR, it is located on SYSVOL.



**EXAMPLE OF ALLOCATION:**

```
PREALLOC SYS.HCULIB, DVC=MS/FSA, UNIT=BLOCK,  
GLOBAL=(MD=md, SIZE=170),  
LINKQD=(NONE=(BLKSIZE=4070, RECSIZE=1024,  
TYPE=CU, LTRKSZ=1, NDLREC,DIRSIZE=1,  
RECFORM=VB)),  
INCRSIZE=340, FILESTAT=CAT, CATNOW, CAT=n;
```

□

**TAILOR Commands:**

Not TAILOR-visible.





## A.48 The SYS.HELP File

### File Name

SYS.HELP

### Domain

GCOS

### Function

This file contains help texts for GCL commands in English (default value).

### Organization

Library

### Sizes

SIZE (BLOCK)	5440
INCRSIZE (BLOCK)	170
MAXSIZE (BLOCK)	10200

### Location

This is a mandatory file that may be located on any FSA disk. If copied by TAILOR, it is located on SYSVOL.



**EXAMPLE OF ALLOCATION:**

```
PREALLOC SYS.HELP, DVC=MS/FSA, UNIT=BLOCK,  
GLOBAL=(MD=md, SIZE=5440),  
LINKQD=(NONE=(BLKSIZE=1024, RECSIZE=264,  
TYPE=SL, LTRKSZ=1, NDLREC, MAXSIZE=10200,  
RECFORM=VB, DIRSIZE=252, COMPACT)),  
INCRSIZE=170, FILESTAT=CAT, CATNOW, CAT =n;
```



**TAILOR Commands:**

Not visible to TAILOR language.



## A.49 The SYS.HLMLIB File

### File Name

SYS.HLMLIB

### Domain

GCOS

### Function

This file contains the Load Modules belonging to the system utilities; e.g., the COBOL compiler, the file and volume utilities.

### Organization

Library

### Sizes

SIZE (BLOCK)	4930
INCRSIZE (BLOCK)	170
MAXSIZE (BLOCK)	10625

### Location

This is a mandatory file that may be located on any FSA disk (preferably the system disk). If TAILOR makes a copy, the file is located on SYSVOL.



### IMPORTANT:

This file must not contain any user members.



**EXAMPLE OF ALLOCATION:**

```
PREALLOC SYS.HLMLIB, EXPDATE=365, DVC=MS/FSA, UNIT=BLOCK,  
GLOBAL=(MD=md, SIZE=4930),  
LINKQD=(NONE=(TYPE=LM, BLKSIZE=4070,  
RECSIZE=1024, RECFORM=VB, NDIREC, DIRSIZE=7,  
LTRKSZ=1, MAXSIZE=10625)), INCRSIZE=170,  
FILESTAT=CAT, CATNOW, CAT=n;
```

□

**TAILOR Commands:**

Not TAILOR-visible.



## A.50 The SYS.HLMLIB2 File

### File Name

SYS.HLMLIB2

### Domain

OLTD

### Function

This file contains Load Modules used by OLTD, the test and diagnostic facilities.

### Organization

Library

### Sizes

SIZE (BLOCK)	1400
INCRSIZE (BLOCK)	80
MAXSIZE (BLOCK)	2120

### Location

This is a mandatory file that must be located on the FSA disk that contains the SYS.CATALOG file.

### EXAMPLE OF ALLOCATION:

```
LIBALLOC LM, (SYS.HLMLIB2, UNIT=BLOCK, MD=md, DVC=MS/FSA,  
             SIZE=(1400, 80), CATNOW, FILESTAT=CAT,  
             CAT=n), MAXSIZE=2120, MEMBERS=60;
```

□

### TAILOR Commands:

Not TAILOR-visible.



## A.51 The SYS.HRELLIB2 File

### File Name

SYS.HRELLIB2

### Domain

OLTD

### Function

This file contains a source member and space reserved for members created by the test and diagnostic facilities.

### Organization

Library

### Size

SIZE (BLOCK) 133

### Location

This is a mandatory file that must be located on the FSA disk that contains the SYS.CATALOG file.

### EXAMPLE OF ALLOCATION:

```
LIBALLOC SL, (SYS.HRELLIB2, UNIT=BLOCK, MD=md, DVC=MS/FSA,  
             SIZE=(133, 85), CATNOW, FILESTAT=CAT,  
             CAT=n), MEMBERS=50;
```

□

### TAILOR Commands:

Not TAILOR-visible.



## A.52 The SYS.HSLLIB File

### File Name

SYS.HSLLIB

### Domain

GCOS

### Function

This file contains source elements (mainly in JCL) for the system.

### Organization

Library

### Sizes

SIZE (BLOCK)	1280
INCRSIZE (BLOCK)	170
MAXSIZE (BLOCK)	2380

### Location

This is a mandatory file that may be located on any FSA disk (preferably the system disk). If copied by TAILOR, it is located on SYSVOL.



### IMPORTANT:

This file must not contain any user sub-files.



**EXAMPLE OF ALLOCATION:**

```
PREALLOC SYS.HSLLIB, EXPDATE=365, DVC=MS/FSA, UNIT=BLOCK,  
GLOBAL=(MD=md, SIZE=1200),  
LINKQD=(NONE=(BLKSIZE=4070, RECSIZE=264,  
RECFORM=VB, TYPE=SL, NDLREC, COMPACT, DIRSIZE=8,  
LTRKSZ=2, MAXSIZE=2380)), INCRSIZE=170,  
FILESTAT=CAT, CATNOW, CAT=n;
```

□

**TAILOR Commands:**

Not visible in TAILOR language.





## A.53 The SYS.HSLLIB2 File

### File Name

SYS.HSLLIB2

### Domain

OLTD

### Function

This file contains source members used by the test and diagnostic facilities.

### Organization

Library

### Size

SIZE (BLOCK) 1190

### Location

This is a mandatory file that may be located on any resident FSA disk, but should be on the same disk as the SYS.CATALOG file.

### EXAMPLE OF ALLOCATION:

```
LIBALLOC SL, (SYS.HSLLIB2, MD=md, DVC=MS/FSA, UNIT=BLOCK,  
             SIZE=(1190, 85), CATNOW, FILESTAT=CAT,  
             CAT=n), COMPACT, MEMBERS=250;
```

□

### TAILOR Commands:

Not TAILOR-visible.



---

## A.54 The SYS.HUB File (not applicable for XTA Systems)

**File Name**

SYS.HUB

**Domain**

FW

**Function**

This file contains system firmware for Central Processor Unit initialization. SYS.HUB is created and loaded using the FGF function BLOAD.

**Organization**

NONE

**Size**

SIZE (BLOCK) 1000

**Location**

This file must be on the system disk.



## A.55 The SYS.HUB7 File (not applicable for XTA Systems)

Applicable only to file loaded by BSRP2 and mandatory for  
DPS 7000/TAXxx/TAXxxC, and DPS 7000/MTx2 and up.

### File Name

SYS.HUB7

### Domain

FW

### Function

This file contains system firmware for Central Processor Unit initialization.  
SYS.HUB7 is created and loaded using the FGF function BLOAD.

### Organization

NONE

### Size

SIZE (BLOCK) 1000

### Location

This file must be on the system disk.



---

## A.56 The SYS.HUBG File

### File Name

SYS.HUBG

### Domain

FW

### Function

This file contains firmware used by the SIP (System Initialization Program) and PIAR (Peripheral Integrity Assurance Routine) functions. SYS.HUBG is created or updated using the FGF function BLOAD.

### Organization

Library

### Sizes

SIZE (BLOCK)	2500
INCRSIZE (BLOCK)	250

### Location

This file must be on the system disk.



## A.57 The SYS.IN File

### File Name

SYS.IN

### Domain

GCOS

### Function

The Stream Reader uses this file to store JCL statements and job input enclosures.

### Organization

Queued Linked

### Sizes

SIZE (BLOCK)	170
INCRSIZE (BLOCK)	170
MAXSIZE (BLOCK)	680

### Location

This is a mandatory file that may be located on any resident FSA disk. If preallocated by TAILOR, it is located on SYSVOL.

To reallocate a SYS.IN file (in case of a problem, or to change its size), you proceed as follows:

1. Rename the existing file:

```
MDF SYS1.SYS.IN SYS1.SYS.OLD
```

2. Allocate the new file:

```
PREALLOC SYS1.SYS.IN ...
```

3. Terminate the system normally.

```
IL01 RESTART(COLD)
```

or

```
IL01 RESTART(CLEAN)
```



4. Delete the old file:

```
DLF SYS1.SYS.OLD BYPASS
```

If you do not have a backup of your system and need to expand the SYS.IN file, or if there are problems with the file such as I/O errors, proceed as follows:

- use the allocation example below,
- delete the old SYS.IN file,
- provoke a system crash,
- perform a RESTART(COLD) at IL01 options.

**EXAMPLE OF ALLOCATION:**

```
PREALLOC SYS1.SYS.IN, UNIT=BLOCK, DVC=MS/FSA,  
GLOBAL=(MD=md, SIZE=170),  
LINKQD=(TYPE=NONE, BLKSIZE=4070, RECSIZE=264,  
LTRKSZ=1, MAXSIZE=680, RECFORM=VB, NDLREC,  
DIRSIZE=8), INCRSIZE=170,  
FILESTAT=UNCAT, EXPDATE=365;
```



**TAILOR Commands:**

```
SYIN, SYSVOL
```



## A.58 The SYS.IUF File

### File Name

SYS.IUF

### Domain

GCOS

### Function

This file is used by TAILOR as well as by the IUF functions and jobs.

### Organization

Library

### Sizes

SIZE (BLOCK)	1020
INCRSIZE (BLOCK)	170

### Location

This is a mandatory file that may be located on any FSA disk. If preallocated by TAILOR, it is located on SYSVOL.



**EXAMPLE OF ALLOCATION:**

```
PREALLOC SYS.IUF, UNIT=BLOCK, EXPDATE=365, DVC=MS/FSA,  
GLOBAL=(MD=md, SIZE=1020),  
LINKQD=(BLKSIZE=4070, RECSIZE=264,  
RECFORM=VB, TYPE=SL, NDLREC,  
DIRSIZE=8, LTRKSZ=1, COMPACT),  
INCRSIZE=170, FILESTAT=CAT, CATNOW, CAT=n;
```



**TAILOR Commands:**

Not visible to TAILOR language.





## A.59 The SYS.JADIR File

### File Name

SYS.JADIR

### Domain

GCOS

### Function

This file is used by the JAS facility.

### Organization

Queued Linked

### Sizes

SIZE (100KB)	120
INCRSIZE (100KB)	5
MAXSIZE (100KB)	4000

### Location

This is an optional file that may be located on any FSA resident disk, preferably the system disk. However, it must be cataloged in the SYS.CATALOG.

### TAILOR Commands:

Not visible to TAILOR language.

SYS.JADIR is created by the MAINTAIN\_JAS utility. For more information, refer to the *File Recovery User's Guide*.



---

## A.60 The SYS.JASBLUE File

### File Name

SYS.JASBLUE

### Domain

GCOS

### Function

Used with TDS-HA (TDS High Availability), this is an output file for a JAS server that guarantees integrity of files.

### Organization

Queued Linked

### Size

SIZE (BLOCK) 3

### Location

This is an optional file that may be located on any FSA resident disk, preferably the system disk. However, it must be cataloged in the SYS.CATALOG.

### TAILOR Commands:

JAS, SYSVOL, RSDVOL, RSDVOL1, NRDVOL, NRDVOL1

### GIUF Command:

BUILD\_JAS



## A.61 The SYS.JASGREEN File

### File Name

SYS.JASGREEN

### Domain

GCOS

### Function

Used with TDS HA (High Availability), this is an output file for a JAS server that guarantees integrity of files.

### Organization

Queued Linked

### Size

SIZE (BLOCK) 3

### Location

This is an optional file that may be located on any FSA resident disk.

### TAILOR Commands:

JAS, SYSVOL, RSDVOL, RSDVOL1, NRDVOL, NRDVOL1

### GIUF Command:

BUILD\_JAS



---

## A.62 The SYS.JRNAL File

**File Name**

SYS.JRNAL

**Domain**

GCOS

**Function**

This file is used by the Before Journal facility.

**Organization**

Non-standard

**Size**

SIZE (512 bytes)            16368

**Location**

This is an optional file that may be located on any FSA disk, but preferably the system disk. If preallocated by TAILOR, it may be located on SYSVOL, RSDVOL, RSDVOL1, NRDVOL, or NRDVOL1.

**TAILOR Commands:**

JBEFORE, SYSVOL, RSDVOL, RSDVOL1, NRDVOL, NRDVOL1

**GIUF Command:**

BJRNAL



## A.63 The SYS.JTRVAL[i] File

### File Name

SYS.JTRVAL[i] [where i = 0 or 1]

### Domain

GCOS

### Function

This file is used by JTRA to store temporary information about the expansion of the values in a JCL sequence.

### Organization

UFAS SEQ

### Size

SIZE (BLOCK) 5000

### Location

This is an optional file which is automatically allocated by JTRA on any FSA resident disk.

### TAILOR Commands:

Not TAILOR-visible.



## A.64 The SYS.KNODET File

### File Name

SYS.KNODET

### Domain

GCOS

### Function

This file is used by GCOS 7 to store control information about active jobs.

### Organization

Non-standard

### Size

SIZE (BLOCK) 85

### Location

This is a mandatory file that must be located on a FSA resident disk. If preallocated by TAILOR, it may be located on SYSVOL, RSDVOL, or RSDVOL1.

To reallocate a SYS.KNODET file (in case of a problem, or to change its size), you proceed as follows:

1. Rename the existing file:

```
MDF SYS1.SYS.KNODET SYS1.SYS.OLD
```

2. Allocate the new file:

```
PREALLOC SYS1.SYS.KNODET ...
```

3. Terminate the system normally.

```
IL01 RESTART(COLD)
```

or

```
IL01 RESTART(CLEAN)
```



4. Delete the old file:

```
DLF SYS1.SYS.OLD BYPASS
```

If you modify the size of this file, you must perform RESTART (CLEAN) before attempting to use the system.

**EXAMPLE OF ALLOCATION:**

```
PREALLOC SYS.KNODET, UNIT=BLOCK, MAXEXT=1,  
DVC=MS/FSA, GLOBAL=(MD=md, SIZE=85),  
NONE=(BLKSIZE=16384), INCRSIZE=170, EXPDATE=365,  
CATNOW, FILESTAT=CAT, CAT=n;
```



**TAILOR Commands:**

```
KNODET, SYSVOL, RSDVOL, RSDVOL1
```

**SYS.KNODET SIZE EVALUATION**

Three parameters are supplied by CONFIG:

MAXFILE	the maximum number of active files.
FILSHARE	the maximum number of process groups allowed to share a file.
MAXDEF	the maximum of active DEFINE statements.

These three parameters are used in the following evaluations:

BKSZ	the block size of KNODET (over-ridden at run-time).
DEFBK	the number of blocks used for DEFINES.
KNBK	the number of blocks required to be accommodated in SYS.KNODET.

**FORMULAS**

```
BKSZ=(52 * filshare) + 416
```

```
DEFBK=first integer greater than (maxdef * 150) / bksz + 1
```

```
KNBK=maxfile + defbk
```



If SYS.KNODET file size is too small, the system automatically reduces the MAXFILE and MAXDEF limits set by CONFIG, and issues the warnings:

DF04 MAXIMUM NUMBER OF ASSIGNED FILES HAS BEEN REDUCED TO xxx

DF05 MAXIMUM NUMBER OF DEFINES HAS BEEN REDUCED TO xxx





## A.65 The SYS.LIB[i] File(s)

### File Name

SYS.LIB, SYS.LIB[i] (where i = from 1 to 15)

### Domain

GCOS

### Function

These files contain:

- Preinitialized Load Modules
- Sharable Modules
- Checkpoint/Restart information.

### Organization

Non-standard

### Sizes

SIZE (BLOCK)	11900
MAXSIZE (BLOCK)	65535

### Location

Whereas the SYS.LIB file is mandatory, the SYS.LIBi files (where i is in the range 1 to 15) are optional. Each may be multi-extent but must be mono-volume. These files may be located on any FSA resident or non-resident disk by the IUF function BUILD\_LIB. SYS.LIB may alternatively be preallocated using TAILOR.

The SYS.LIBi files cannot be allocated on mirror disks.

### TAILOR Commands:

LIB, SYSVOL, RSDVOL, RSDVOL1, NRDVOL, NRDVOL1

### GIUF Command:

BUILD\_LIB



## A.66 The SYS.LOGC File

### File Name

SYS.LOGC

### Domain

GCOS

### Function

This file contains all the messages sent to and from the MAIN console.

### Organization

Non-standard

### Size

SIZE (BLOCK) 240

### Location

This is an optional file which may be located on any FSA disk. If required, it is preallocated by TAILOR on SYSVOL, RSDVOL, or RSDVOL1.

### EXAMPLE OF ALLOCATION:

As SYS.LOGC must be cataloged with specific sharability attributes, its creation requires two steps:

```
CATALOG SYS.LOGC SHARE=ONEWRITE CATALOG=n;
```

```
PREALLOC SYS.LOGC, UNIT=BLOCK, DVC=MS/FSA,  
GLOBAL=(MD=md, SIZE=240), EXPDATE=365,  
NONE=(BLKSIZE=4096), INCRSIZE=170,  
FILESTAT=CAT, CAT=n, CATNOW;
```

□

### TAILOR Commands:

LOGC, SYSVOL, RSDVOL, RSDVOL1



## A.67 The SYS.OUT File

### File Name

SYS.OUT

### Domain

GCOS

### Function

This is the standard output file intended to contain output from jobs, for subsequent printing.

### Organization

Queued Linked

### Sizes

SIZE (BLOCK)	340
INCRSIZE (BLOCK)	85
MAXSIZE (BLOCK)	32000
BLKSIZE (BLOCK)	8166
RECSIZE (BLOCK)	4024
LTRKSZ (BLOCK)	5
DIRSIZE (BLOCK)	8



### Location

This mandatory file may be distributed over up to 10 resident disks, but should if possible be located on the system disk. If required, it can be preallocated by TAILOR on SYSVOL, RSDVOL, and/or RSDVOL1 (i.e. a maximum of 3 different disks). File creation may be performed using PREALLOC in order to include resident volume(s) other than RSDVOL and RSDVOL1.

To reallocate a SYS.OUT file (in case of a problem, or to change its size), you proceed as follows:

1. Rename the existing file:

```
MDF SYS1.SYS.OUT SYS1.SYS.OLD
```

2. Allocate the new file:

```
PREALLOC SYS1.SYS.OUT ...
```

3. Terminate the system normally.

```
IL01 RESTART(COLD)
```

or

```
IL01 RESTART(CLEAN)
```

4. Delete the old file:

```
DLF SYS1.SYS.OLD BYPASS
```

If you do not have a backup of your system and need to expand the SYS.OUT file, or if there are problems with the file such as I/O errors, proceed as follows:

- delete the old SYS.OUT file

```
DLF SYS1.SYS.OUT
```

- use the "*Example of Allocation*" below, or execute the following JCL sequence:

```
EJ IUF_PREA_SYSOUT..SYS.IUF
   VL=(MDNAME, SIZE, MDNAME1, SIZE1, ...)
```

- provoke a system crash,
- perform a RESTART(COLD) at IL01 options.



**EXAMPLE OF ALLOCATION:**

```
PREALLOC SYS1.SYS.OUT, DVC=MS/FSA, UNIT=BLOCK,  
        SPLIT=(mcname, SIZE=340), (mcname1, SIZE=size1),  
        (mcname2, SIZE=size2)), LINKQD=(TYPE=NONE,  
        BLKSIZE=8166, RECSIZE=4024, MAXSIZE=32000,  
        LTRKSZ=5, RECFORM=VB, NDLREC, DIRSIZE=8),  
        INCRSIZE=85, FILESTAT=UNCAT, EXPDATE=365;
```



**TAILOR Commands:**

SYO, SYSVOL, RSDVOL, RSDVOL1



## A.68 The SYS.PVMF[i] File(s)

### File Name

SYS.PVMF, SYS.PVMF[i] (where i = from 1 to 15)

### Domain

GCOS

### Function

These files contain permanent virtual memory files.

### Organization

Non-standard

### Sizes

SIZE (BLOCK)	5100
MAXSIZE (BLOCK)	65535

### Location

While the SYS.PVMF file is mandatory, the SYS.PVMFi files (where i is in the range 1 to 15) are optional. Each may be multi-extent but must be mono-volume. These files may be located on any resident or non-resident disk by the IUF function BUILD\_PVMF. SYS.PVMF may alternatively be preallocated using TAILOR.

The SYS.PVMFi files cannot be allocated on mirror disks.

### TAILOR Commands:

PVMF, SYSVOL, RSDVOL, RSDVOL1, NRDVOL, NRDVOL1.

### GIUF Command:

BUILD\_PVMF



## A.69 The SYS.QM File

### File Name

SYS.QM

### Domain

GCOS

### Function

This file contains system directives together with the associated answers. The file is accessed by the operator station Queue Manager and replaces the Operator Message Handler mailboxes accessed by the operator station Queue Manager in previous releases.

### Organization

UFAS Relative

### Size

SIZE (CI) 500

### Location

This is a mandatory file that may be located on any FSA resident disk. If created by TAILOR, it may be located on SYSVOL, RSDVOL, RSDVOL1, NRDVOL, or NRDVOL1.

SYS.QM must be allocated on a disk that contains a backing store file.

### TAILOR Commands:

QM, SYSVOL, RSDVOL, RSDVOL1, NRDVOL, NRDVOL1.

### GIUF Command:

BUILD\_QM



---

## A.70 The SYS.QM\_BACKUP File

### File Name

SYS.QM\_BACKUP

### Domain

GCOS

### Function

This file contains system directives together with the associated answers. The file is accessed by the operator station Queue Manager and replaces the Operator Message Handler mailboxes accessed by the operator station Queue Manager in previous releases.

### Organization

UFAS Relative

### Size

SIZE (CI) 500

### Location

This is a mandatory file that may be located on any FSA resident disk. If created by TAILOR, it may be located on SYSVOL, RSDVOL, or RSDVOL1, NRDVOL, NRDVOL1.

SYS.QM\_BACKUP must be allocated on a disk that contains a backing store file.

### TAILOR Commands:

QMBACK, SYSVOL, RSDVOL, RSDVOL1, NRDVOL, NRDVOL1.

### GIUF Command:

BUILD\_QMBACKUP





## A.71 The SYS.SITE.BIN File

### File Name

SYS.SITE.BIN

### Domain

GCOS

### Function

This file is allocated by GCOS 7, in order to contain site-specific items in binary format used by the FW, OLTD and GSF Domains.

### Organization

Library

### Size

SIZE (BLOCK) 450

### Location

This is a mandatory file that may be located on any FSA disk. If copied by TAILOR, it is located on SYSVOL.

### EXAMPLE OF ALLOCATION:

```
LIBALLOC BIN, (SYS.SITE.BIN, UNIT=BLOCK,  
              DVC=MS/FSA, MD=md, SIZE=(450,170),  
              CATNOW, FILESTAT=CAT, CAT=n),  
              MEMBERS=1000;
```

□

### TAILOR Commands:

Not visible to TAILOR language.



## A.72 The SYS.SITE.SL File

### File Name

SYS.SITE.SL

### Domain

GCOS

### Function

This file is allocated by GCOS 7, in order to contain site-specific source-format items used by the FW, OLTD, and GSF Domains.

### Organization

Library

### Size

SIZE (BLOCK) 450

### Location

This is a mandatory file that may be located on any FSA disk. If copied by TAILOR, it is located on SYSVOL.

### EXAMPLE OF ALLOCATION:

```
LIBALLOC SL, (SYS.SITE.SL, UNIT=BLOCK, DVC=MS/FSA, MD=md,  
             SIZE=(450, 170), CATNOW, FILESTAT=CAT,  
             CAT=n), MEMBERS=1000;
```

□

### TAILOR Commands:

Not visible to TAILOR language.



## A.73 The SYS.SP\_DUMP File

### File Name

SYS.SP\_DUMP

### Domain

GCOS

### Function

This is the standard output file, intended to contain binary dumps from the system processor and TDS jobs. These dumps are to be saved for analysis by the Bull Service Center.

### Organization

Linked Queued

### Size

SIZE (BLOCK) 7500

### Location

This is a mandatory file that may be located on any FSA disk. If created by TAILOR, it may be located on SYSVOL, RSDVOL, RSDVOL1, NRDVOL, or NRDVOL1.

### TAILOR Commands:

SP\_DUMP, SYSVOL, RSDVOL, RSDVOL1, NRDVOL, NRDVOL1

### GIUF Command:

BUILD\_SP\_DUMP



## A.74 The SYS.SPOOL[i] File(s)

### File Name

SYS.SPOOLi (where i = from 0 to 9)

### Domain

GCOS

### Function

The SPOOL files contain the active GCL procedures and environment associated with the GCL user.

### Organization

Linked Queued

### Sizes

SIZE (BLOCK)	1360
INCRSIZE (BLOCK)	170

### Location

At least one SPOOL file is mandatory. SPOOL files may be located on any FSA resident disk. It is recommended that SPOOL files be distributed over several disks. The number of SPOOL files depends upon the number of users:

Ideally, there should be between 15 and 20 IOF users per spool file, but never more than 25 users.



Note that, if SYS.SPOOLi is allocated on a disk which is specified at ISL time as being either NON RESIDENT or NON SHARED, the JCL statements, VOLSAVE, VOLREST, VOLDUPLI or the GCL procedures SAVE\_DISK, RESTORE\_DISK will be prevented by the system from working on the disk concerned. Should such a situation occur, use of the JCL statements FILSAVE, FILREST, FILDUPLI or the GCL procedures SAVE\_FILE, RESTORE\_FILE, COPY\_FILE is mandatory.

SYS.SPOOL0 is preallocated by TAILOR on SYSVOL, RSDVOL, or RSDVOL1.

**TAILOR Commands:**

SPOOL, SYSVOL, RSDVOL, RSDVOL1

**GIUF Command:**

BUILD\_SPOOL



---

## A.75 The SYS.SWLOG File

### File Name

SYS.SWLOG

### Domain

GCOS

### Function

This file contains events recorded by the Trace System, for access by the Extractor service job TRCEXTR.

### Organization

UFAS Relative

### Size

SIZE (BLOCK) 800

### Location

This is a mandatory file that may be located on any FSA disk, resident or non-resident. It must be present in order to allow the Collector service job to start. If created by TAILOR, it is located on SYSVOL, RSDVOL, RSDVOL1, NRDVOL, or NRDVOL1.



**EXAMPLE OF ALLOCATION:**

```
CATALOG  SYS.SWLOG, SHARE=FREE, DUALSHR=NONE;

PREALLOC  SYS.SWLOG, UNIT=BLOCK, DVC=MS/FSA,
          GLOBAL=(MD=md, SIZE=800), MAXEXT=1,
          UFAS=(RELATIVE=(CISIZE=4096, RECSIZE=4082,
          RECFORM=FB)), EXPDATE=999, CATNOW,
          FILESTAT=CAT, CAT=n;
```

□

**TAILOR Commands:**

```
SYSVOL, RSDVOL, RSDVOL1, NRDVOL, NRDVOL1, SWLOG
```



---

## A.76 The SYS.SYSDUMP File

### File Name

SYS.SYSDUMP

### Domain

GCOS

### Function

This file is used to store the memory dump (i.e., all the data for all the processes present in memory) in the event of a system crash.

The size of SYS.SYSDUMP does not depend on memory size, but on the number of jobs that are active. The number of active jobs allowed increases as memory size increases.

### Organization

Non-standard

### Sizes

SIZE (BLOCK)	11730
MAXSIZE (BLOCK)	65535





**Location**

This is a mandatory file that may be located on any FSA disk. If preallocated by TAILOR, or alternatively using the IUF utility BUILD\_SYSDUMP, it may be located on SYSVOL, RSDVOL, or RSDVOL1.

The SYS.SYSDUMP file cannot be allocated on mirror disks.

The GIUF function BUILD\_SYSDUMP is mandatory to rebuild this file.

The default value shown under SIZE above (and in the table *GCOS 7 File Domain Default Values* in the chapter *TAILOR*) is appropriate for an average system load.

**TAILOR Commands:**

SYSDUMP, SYSVOL, RSDVOL, RSDVOL1

**GIUF Command:**

BUILD\_SYSDUMP



---

## A.77 The SYS.SYSTEM File

### File Name

SYS.SYSTEM

### Domain

GCOS

### Function

This file contains the backing-store image of the operating system, and is copied to backing store by ISL-RESTORE when starting a system session.

### Organization

UFAS Indexed

### Sizes

SIZE (BLOCK)	3397
CISIZE (BLOCK)	16384

### Location

This is a mandatory file that may be located on any resident or non-resident FSA disk. If created by TAILOR, it may be located on SYSVOL, RSDVOL, RSDVOL1, NRDVOL, or NRDVOL1.

The SYS.SYSTEM file cannot be allocated on mirror disks.



**EXAMPLE OF ALLOCATION:**

```
PREALLOC SYS.SYSTEM, UNIT=BLOCK, MAXEXT=16,  
          DVC=MS/FSA, GLOBAL=(MD=md, SIZE=3397),  
          UFAS=(INDEXED=(CISIZE=16384, RECSIZE=4096,  
          KEYLOC=1, KEYSIZE=9, RECFORM=V)), INCRSIZE=215,  
          VERSION=CURRENT, CATNOW, FILESTAT=CAT, CAT=n;
```

Note that the GCL command BSYS may be used for this file allocation.

□

**TAILOR Commands:**

```
ISL, SYSVOL, RSDVOL, RSDVOL1, NRDVOL, NRDVOL1
```



## A.78 The SYS.SYSTEM2 File

### File Name

SYS.SYSTEM2

### Domain

OLTD

### Function

This file contains Sharable Modules used by the test and diagnostic utility.

### Organization

UFAS Indexed

### Sizes

SIZE (CI)	43
CISIZE (BLOCK)	12800

### Location

This is a mandatory file that should be located on the same FSA resident disk that contains the SYS.CATALOG file. If created by TAILOR, it is located on SYSVOL.

### EXAMPLE OF ALLOCATION:

```
PREALLOC SYS.SYSTEM2, UNIT=BLOCK, MAXEXT=16,  
DVC=MS/FSA, GLOBAL=(MD=md, SIZE=43),  
UFAS=(INDEXED=(CISIZE=12800, RECSIZE=4096,  
KEYLOC=1, KEYSIZE=9, RECFORM=V)),  
INCRSIZE=43, CATNOW, FILESTAT=CAT, CAT=n,  
EXPDATE=365, VERSION=CURRENT;
```

□

### TAILOR Commands:

Not TAILOR-visible.



## A.79 The SYS.TVMF[i] File(s)

### File Name

SYS.TVMF, SYS.TVMF[i] (where i = from 1 to 15)

### Domain

GCOS

### Function

These files are used for storing temporary virtual memory files.

### Organization

Non-standard

### Sizes

SIZE (BLOCK)	2550
MAXSIZE (BLOCK)	65535

### Location

Although the SYS.TVMF file is mandatory, the SYS.TVMFi files (where i is in the range 1 to 15) are optional. Each SYS.TVMFi file may be multi-extent but must be mono-volume. These files may be located on any resident or non-resident disk by the IUF function BUILD\_TVFMF. SYS.TVMF may alternatively be preallocated using TAILOR.

The SYS.TVMFi files cannot be allocated on mirror disks.

### TAILOR Commands:

TVMF, SYSVOL, RSDVOL, RSDVOL1, NRDVOL, NRDVOL1

### GIUF Command:

BUILD\_TVFMF



## A.80 The SYS.URCINIT File

### File Name

SYS.URCINIT

### Domain

GCOS

### Function

This file contains the control tables used with unit record devices: printer character sets and VFU's.

### Organization

Queued Linked

### Size

SIZE (BLOCK) 170

### Location

This is a mandatory file, which may be located on any FSA resident disk. If created by TAILOR, it is located on SYSVOL.

### EXAMPLE OF ALLOCATION:

```
PREALLOC SYS.URCINIT, UNIT BLOCK,  
          DVC=MS/FSA, GLOBAL=(MD=md, SIZE=170),  
          LINKQD=(TYPE=NONE, BLKSIZE=4070, RECSIZE=580,  
          LTRKSZ=1, DIRSIZE=1, RECFORM=VB), EXPDATE=365,  
          CATNOW, FILESTAT=CAT, CAT=n;
```

□

### TAILOR Commands:

Not visible in TAILOR language.



## A.81 System File Access Rights

Table A-3. System File Access Rights (1/2)

efn	SHARE	RECOVERY	WRITE	READ
SITE.CATALOG	FREE		*	
SITE.CMS_CSD	NORMAL			
SITE.CMS_CDD_namei	NORMAL			
SITE.CMS_DS_name1_name2	UNSPEC			
SITE.HALOCK	FREE			
SITE.HELP	DIR		*	
SITE.IN	DIR			
SITE.MIRLOG	FREE		*	
SITE.STARTUP	DIR		*	
SYS.BKST[i]	NORMAL		OPERATOR	*
SYS.BOOT	NORMAL	RMSIOF		OPERATOR
SYS.CATALOG	FREE		*	
SYS.C.INCLUDE	DIR			*
SYS.CLC	DIR			*
SYS.DSA*	DIR		OPERATOR	*
SYS.DSALOGi	ONEWRITE		OPERATOR	*
SYS.DSARCC_<lct-name>_x	ONEWRITE		OPERATOR	*
SYS.DSATRCIDT_<fcp-ts>	ONEWRITE		OPERATOR	*
SYS.ERLOG	ONEWRITE		OPERATOR	*
SYS.FW.*	DIR	RMSIOF	OPERATOR	*
SYS.GPL.MACLIB	DIR			*
SYS.GSF.*	DIR	RMSIOF	OPERATOR TELECONTROL	*
SYS.HBINLIB	DIR	RMSIOF		
SYS.HBINLIB2	DIR	RMSIOF	OPERATOR	*
SYS.HCULIB	DIR	RMSIOF		*
SYS.HELP	DIR		OPERATOR	*
SYS.HLMLIB	DIR	RMSIOF		*
SYS.HLMLIB2	DIR	RMSIOF	OPERATOR	*
SYS.HRELLIB2	DIR	RMSIOF	OPERATOR	*
SYS.HSLLIB	DIR	RMSIOF		*
SYS.HSLLIB2	DIR	RMSIOF	OPERATOR	*
SYS.HUB / SYS.HUB7	NORMAL	RMSIOF	OPERATOR	*
SYS.HUBG	DIR	RMSIOF	OPERATOR	*
SYS.IN				



**Table A-3. System File Access Rights (2/2)**

<b>efn</b>	<b>SHARE</b>	<b>RECOVERY</b>	<b>WRITE</b>	<b>READ</b>
SYS.IUF	DIR	RMSIOF	OPERATOR	*
SYS.JADIR	DIR		OPERATOR [or *]	*
SYS.JASBLUE	DIR		OPERATOR	*
SYS.JASGREEN	DIR		OPERATOR	*
SYS.JRNAL	NORMAL		OPERATOR [or *]	*
SYS.KNODET	NORMAL	RMSIOF		
SYS.LIB[i]	NORMAL		OPERATOR	*
SYS.LOGC	ONEWRITE		OPERATOR	*
SYS.OUT			OPERATOR	*
SYS.PVMF[i]	NORMAL		OPERATOR	*
SYS.QM*	NORMAL		OPERATOR	
SYS.SITE*	DIR		*	
SYS.SPDUMP	DIR		*	*
SYS.SPOOLi	NORMAL		OPERATOR	*
SYS.SWLOG	FREE		OPERATOR	*
SYS.SYSDUMP	NORMAL		OPERATOR [or *]	*
SYS.SYSTEM	NORMAL	RMSIOF		*
SYS.SYSTEM2	NORMAL	RMSIOF	OPERATOR	*
SYS.TVMF[i]	NORMAL	RMSIOF	OPERATOR	*
SYS.URCINIT	DIR			*

**NOTES:**

1. All system files have the owner SYSADMIN.
2. The above default access rights are minima and so may never be reduced or cancelled by the user, although they may of course always be increased/added to by SYSADMIN.





## A.82 Calculating Backing Store File Sizes

### A.82.1 General Remarks

Backing-store space is split into 4 areas:

- The paging backing store: SYS.BKST through SYS.BKST15
- The library backing store: SYS.LIB through SYS.LIB15
- The backing store for permanent virtual memory files: SYS.PVMF through SYS.PVMF15
- The backing store for temporary virtual memory files: SYS.TVMF through SYS.TVMF15.

The system administrator must allocate sufficient space to avoid all but exceptional cases of backing-store overflow, without this space being so excessive as to cause unnecessary disk space wastage and disk drive arm displacement. Each of the 4 areas must be calculated separately, and no single backing store file extent may exceed 256MB.

The size of a backing-store file cannot exceed 65535 blocks.

While it is usually more efficient to distribute the files belonging to each area over several disks, files belonging to different backing-store areas may coexist on any single disk volume. For example, in a small configuration, the files SYS.BKST, SYS.PVMF, SYS.LIB and SYS.TVMF may be located on one disk while SYS.BKST1, SYS.PVMF1, SYS.LIB1 and SYS.TVMF1 are located on a second disk.

Backing-store occupancy may easily be checked during normal processing by using the DCBS-MAP. In a "normal" phase of processing, a comfortable amount of backing store should remain unused. The following figures seek to be realistic, erring if anything on the side of pessimism. They are however only average estimates that cannot by definition take account of any unusual site-specific conditions.

See the table *Backing Store File Sizes* below.



### A.82.2 Paging Backing Store Files

SYS.BKST file size depends on the size of the segments referenced by the JOPO table for its permanent part, and upon the number of dynamically created pages for its temporary part (which must have the same size for all of the files SYS.BKST and SYS.BKSTi).

SYS.BKST file size no longer depends on main memory size.

The contents of a SYS.BKST file may be analyzed into 3 parts:

1. The system itself plus segments which may be created dynamically by the system (10MB).
2. The crash area, MMSZ, 32MB. This is independent of main memory size.
3. The pages belonging to each executing job. Each job needs approximately 0.8MB of backing store, although certain applications (such as IQS and Oracle) need more.

$$10 + 32 + 0.8 * \text{multlev}$$

A bonus should be added to the resulting paging backing-store total, to allow for those applications that are large consumers of backing store.

### A.82.3 Library Backing Store

The contents of the SYS.LIB[i] files may be analyzed into 3 parts:

1. Sharable Modules (SM): system SMs account for about 25MB. User SMs are roughly proportional to the number of TDSs that may be loaded concurrently on the site (but could be much higher for a large TDS application):

$$2 * \text{NTDS}$$

2. Preinitialized Load Modules (LM): the amount of space required for frequently-used system LMs may vary from site to site, but often accounts for about 10MB. User Load Modules may vary widely, but are often related to the system's multiprogramming level:

$$2 + 0.3 * \text{multlev}$$

3. Repeatable steps and Checkpoints: based on (a) about 15% of executing jobs performing checkpoints, and (b) the size of each checkpoint being about 1MB:

The total Library Backing Store may thus be calculated as:

$$37 + 0.5 * \text{multlev} + 2 * \text{ntds}$$



#### A.82.4 Permanent Virtual Memory Files

The contents of SYS.PVMF[i] files may be analyzed into:

1. A constant part that includes system PVMF Files, usually about 5MB.
2. A variable part related to the number of (a) known jobs and (b) executing jobs:

$$24/1024 * multlev + 16/1024 * maxjob$$

The total requirements for Permanent Virtual Memory Files may thus be calculated as:

$$5 + 24/1024 * multlev + 16/1024 * maxjob$$

#### A.82.5 Temporary Virtual Memory Files

Only only a few GCOS 7 processors use these files. 300KB should suffice for executing jobs. The constant element of the system does not require more than 1MB:

$$1 * 0.3 * multlev$$

#### A.82.6 Conversion of Megabytes to Blocks

Conversion of MB into blocks:

MS/FSA: 1MB = 2000 blocks



### A.82.7 Backing Store File Sizes

The table below shows some typical backing store file sizes. Remember, however, that there are many influencing factors, in particular the number of jobs running at any one time.

**Table A-4. Some Typical Backing Store File Sizes (in Blocks of 4096 Bytes)**

Files	Main memory Size in MBytes		
	64	128	256
SYS.BKST[i]	7800	14000	25200
SYS.LIB[i]	21600	39000	63900
SYS.PVMF[i]	9200	16600	29800
SYS.TVMF[i]	4700	8500	15200



---

## B. SIP Functions

### **NOTE:**

This appendix is not concerned for XTA systems

### **B.1 SIP (System Initialization Program)**

SIP is a small Operating System, entirely resident in memory. It is designed to initialize a DPS 7000 system in INIT or RELOAD mode, or to monitor the execution of certain functions in UTIL mode when GCOS 7 is either not available, or not operational.

It contains a SAVE function used to save an FSA disk and a RESTORE function used to restore an FSA system disk from the contents of the backup P-MEDIA (generated using either SAVE\_DISK for each disk or through SIP).

### **B.2 SIP Loading**

SIP can be loaded in three different modes:

INIT mode:

SIP initializes the controllers as well as certain specific resources, such as the console and the printer, then loads and activates ISL (the GCOS 7 loader).

RELOAD mode:

SIP only loads and activates ISL.

UTIL mode:

SIP activates the UTILity functions. The aim of this mode is to save an FSA disk on tapes or cartridges without reinitializing the CSS (Central Sub-System).

SIP requests the parameters it needs, via a dialog with the operator.

Note that this UTIL mode is not compatible with the AUTOMATIC mode of the Service Processor. SIP loading is dependent on the DPS 7000 model. See the paragraphs below.



SIP is loaded by the SYC application running in the Service Processor with specific SYC commands entered in the SYC window:

- in INIT mode with the commands IN/INR/IND:  
IN/INR/IND <bootload device> LD <os> <ISL device>
- in RELOAD mode with the command RL:  
RL LD <ISL device>  
or with specific instructions after a System Reset or a GCOS 7 crash.
- in UTIL mode with the commands IN/INR/IND and the parameter UTIL.  
IN/INR/IND <bootload device> LD UTIL

### B.3 SIP Dialog

SIP messages are displayed on the System console.

Messages requiring an answer begin with an asterisk (\*) and end with the prompt "?Si?" or "?R?".

There is no deferred answer mechanism equivalent to that provided under GCOS 7.

The BRK (BREAK) key should not be used during SIP execution.

There are no SIP commands. UTILity functions are activated by the operator's answers to SIP messages requiring a response.



## B.4 UTILITY Functions Under SIP

In the event of a problem with your P-Set, use SIP in UTIL mode. If you are unable to initialize GCOS 7 and thus need to restore each disk belonging to your P-Set from the P-Tapes (or cartridges), this may be done by supplying an R MSxx answer to the INIT20 request message. On the systems with FSA disks you can save data disks if necessary before restoring, this may be done by supplying an S MSxx answer to the INIT20 request message.

When the RESTORE, or SAVE, function has completed, this INIT20 request is displayed again. You may leave UTIL mode and initialize the system from the newly restored Set, by giving the I MSxx answer.

To perform a RESTORE function under SIP, you need to know the P-Set file name. This file name should not be confused with the media names.

### NOTE:

Be careful to enter the correct file name because the system could take up to half an hour to search through the whole set of media before declaring that it cannot find a file.

These procedures under SIP restore or save a whole disk at a time. No manipulation of individual files is possible.

When an FSA disk is saved under SIP, CATALOG information is not saved in the SAVE file. The SAVE file can be restored under GCOS 7 by RESTORE\_DISK or RESTORE\_FILE GCL procedure, however the operator must update the CATALOG files if necessary.

## B.5 Execution Example

The first request allows three functions to be executed:

```
* INIT20 UTIL FUNCTION: INIT, RESTORE, SAVE.  
REPLY (I MSXX, R MSXX, S MSXX)
```

?R?

I MSXX	Leave UTIL mode and initialize the system from the corresponding MS.
R MSXX	Restore the corresponding FSA disk.
S MSXX	Save the corresponding FSA disk.



### B.5.1 The RESTORE Function

1. Before executing the RESTORE function SIP requests all the necessary parameters.

- a. SIP requires the device on which the media containing the SAVE file (or the first media if the image is stored on several media) is mounted:

```
* INIT44 GTDV DEFINE FIRST INPUT TAPE DEVICE.
      REPLY (UD <DEVICE>, CR)
?R?UD MT01
```

The device can be any tape, cassette, or cartridge that is currently supported. This device must be declared in the IRT and must be available.

- b. If there are several SAVE images stored on the same media, the operator can specify the image to be restored, as follows:

```
* INIT21 MT01 DEFINE SAVE FILE NAME. (DEFAULT VALUE=FSA41)
?R?
```

If the media contains one image only, the default value must correspond to the image to be restored.

- either press the "Enter" key to select the default value
- or give the name of another image present on the media. In this case the operator must give the right file name. If he types the wrong file name, it may take the system 30 minutes to search through the whole of the tape (or cartridge) before declaring that the file cannot be found.

SIP reads the beginning of this image and checks it.

```
INIT22 MT01 MOUNTED MEDIA: SAVT FILE: FSA43 SEQ: 0001
```

This message gives the characteristics of the image (or the first part of the image). The SEQ (sequence number) indicates that this media contains the beginning of the image.

- c. If the SAVE image is stored on several media, up to 6 cartridge handlers can be used at the same time to read the image. The operator mounts the other corresponding media and gives the corresponding drives to SIP.

```
* INIT24 UTIL DEFINE MORE INPUT DEVICE (MTXX ..)
?R?MT02 MT05
```

SIP checks these media.

```
INIT22 MT02 MOUNTED MEDIA: SAVT3 FILE: FSA43 SEQ: 0003
INIT22 MT05 MOUNTED MEDIA: SAVT2 FILE: FSA43 SEQ: 0002
```

Press "Enter" to restore from a single device. (Even if there are several backup media, it is not compulsory to specify several devices. When the system requires the other next media, a message is sent requesting the operator to mount it.)





- d. SIP checks the media to be restored, according to the information stored in the first SAVE media. This media can be standard, for example:

```
INIT23 MS42 MOUNTED MEDIA: FSA516
```

or not standard, for example:

```
INIT81 MS42 MOUNTED MEDIA NOT STANDARD
```

This message means that the disk does not have a volume label, but it is correctly formatted.

```
* INIT06 <extid> IS A MIRROR DISK
FORCE DISK RESTORE. REPLY (Y,N)
```

This means that the target disk selected for a restore operation is a mirrored disk.

The reply Y will force the restore to take place on one of the two mirrored disks. In this case, the output disk is no longer considered to be a mirrored disk.

Answer N if you wish to change your target disk.

- e. For FSA disks:

- if the formatting does not correspond to the SAVE image
- or if only the default partition is defined on the disk

another message is issued:

```
INIT36 MS42 FORMAT MANDATORY
```

- f. SIP is able to format the disk. This operation is unnecessary unless the message INIT36 has been displayed (disks should normally be formatted in the factory).

```
* INIT25 MS42 DISK FORMATTING. REPLY (Y,N)
```

```
?R?N
```

Disk formatting takes approximately 15 minutes.

- g. Final confirmation is requested.

```
* INIT26 UTIL FUNCTION : RESTORE : DO YOU AGREE? REPLY (Y,N)
```

```
?R?
```

Answer N if you want to change the value of any parameters.

When this message is displayed, no write operation has been performed on the disk.



2. If the operator accepts the RESTORE, all data on the disk will be destroyed.
  - a. SIP informs the operator of the restored files:
 

```
INIT27 MS42 FILE INST.FW.BINLIB RESTORED
INIT27 MS42 FILE SITE.APL RESTORED
```
  - b. When the disk is fully restored, SIP displays the new name of the restored disk:
 

```
INIT23 MS42 MOUNTED MEDIA: FSA43
INIT29 UTIL RESTORE FUNCTION COMPLETED
```
  - c. Otherwise, if the SAVE image is stored on several media, SIP informs the operator when the first SAVE media has been restored.
 

```
INIT MT01 TAPE: SAVT SEQ: 0001 RESTORED
```

SIP searches for the following SAVE media in the defined drives and continues. If the following media is not found, it requested by SIP, as follows:

```
* INIT44 GTDV DEFINE NEXT INPUT TAPE DEVICE.
  REPLY (UD <DEVICE>, CR)
?R?UD MT01
```
3. When the RESTORE is finished, the message INIT20 is issued again.

### B.5.2 The SAVE Function

1. Before executing the SAVE function, SIP requests all the necessary parameters.
  - a. SIP checks the media to be saved. Only FSA disks can be saved under SIP.
 

```
INIT23 MS42 MOUNTED MEDIA: FSA43
```
  - b. SIP requests the device on which the media to contain the SAVE file is mounted, (or the first part of the SAVE file, if it is too big for one media):
 

```
* INIT44 GTDV DEFINE FIRST OUTPUT TAPE DEVICE.
  REPLY (UD <DEVICE>, CR)
?R?UD MT01
```

The device can be any tape, cassette, or cartridge that is currently supported.

This device must be declared in the IRT and must be available. SIP checks the mounted media. A message specifies its media name.

```
INIT23 MT01 MOUNTED MEDIA: TEST
or
INIT81 MT01 MOUNTED MEDIA NOT STANDARD
```



At the beginning of the save, SIP will prepare the media with the new characteristics specified in the parameters.

- c. If the tape drive supports several densities, the operator can specify it.

```
* INIT44 MT01 DEFINE TAPE DENSITY.  
REPLY (1600,6250)  
?R? 6250
```

- d. A new volume name is requested for this mounted media:

```
* INIT41 MT01 DEFINE NEW VOLUME NAME (1-6 CHAR)  
?R? SAV1
```

This name is composed of 6 characters maximum. The first character must be alphanumeric.

- e. If the file to be saved is too big for one media, the operator can specify up to 6 other devices to mount corresponding media.

```
* INIT24 UTIL DEFINE MORE OUTPUT DEVICE (MTXX ...)  
?R?MT02 MT05
```

SIP checks these media:

```
INIT23 MT02 MOUNTED MEDIA: WORK
```

```
INIT81 MT05 MOUNTED MEDIA NOT STANDARD
```

For every media, a new volume name is requested (message INIT41).

Press "Enter" to save on a single device. (Even if there is not enough space on the first media to save the disk, it is not compulsory to specify several devices. When the system requires the next media, a message is sent, requesting the operator to mount it).

- f. The name of the Save file is requested.

```
* INIT21 MT01 DEFINE SAVE FILE NAME. (1 - 17 CHAR)  
?R?
```

This name is composed of 17 characters maximum. The first character must be alphanumeric.

- g. Final confirmation is requested.

```
* INIT26 UTIL FUNCTION : RESTORE : DO YOU AGREE? REPLY (Y,N)  
?R?
```

Answer N if you want to change the value of any of the parameters. When this message is displayed, no write operation has been performed on the tape (or cartridge).



2. If the operator accepts the SAVE, all data on the mounted tapes (or cartridges) will be destroyed.
  - a. SIP checks the first output media again and prepares it with the parameters specified in the first phase (volume name, file name, density if necessary).

The format is compatible with GCOS 7. The date stored in control blocks corresponds to the disk volume preparation date, the time to "00h 00m 00s ", and OS identification to SIP575:00.

- b. SIP scans the disk to be saved and displays all the disk file characteristics. This information may be helpful for a specialist if a problem occurs, or if the operator wants to restore disk files from the SAVE file under GCOS 7.

```
INIT11  MS42 FILE : TEST1.FILE.SL
        NB_EXTENT: 0001  FB_SIZE: 001000
INIT12  MS42 EXTENT: 0001  DESC:  01000000  0005DB80  00000000
        00000184  480001B3  2700
```

These messages describe the mapping of the found files stored on the disk. For example:

The ESSA11.FILE.SL file is composed of one EXTENT

Its file block size is "1000" hexa

Extent description: 01 extent type (data records)

```
000000  number from the file beginning of the first FILE BLOCK
        in this extent
0005DB  number from the file beginning of the last FILE BLOCK
        in this extent
80      extent flags
00000000 RFU
00      Partition number
00018448 data block number of the first data block which is
        included in the first file block in this extent
0001B327 data block number of the last data block that is included
        in the last file block in this extent.
00      RFU
```

- c. SIP saves all disk files.
 

SIP informs the operator of the saved files (except for the VTOC files):

```
INIT27  MS42  FILE INST.FW.BINLIB SAVED
```

```
INIT27  MS42  FILE SITE.APL SAVED
```

- d. when the tape (or cartridge) is full or when the disk is fully saved:

```
INIT48  MT01  TAPE SAVT1 FULL
```



- e. when the disk is fully saved:

```
INIT23  MS42  VOLUME FSA42 HAS BEEN SAVED ONTO FILE
        SAVFSA42
INIT29  UTIL  SAVE FUNCTION COMPLETED
```

- f. Otherwise SIP searches for another tape (or cartridge) in the defined drives, checks it (message INIT23), prepares it and continues saving the disk.

If no other tape (or cartridge) is found, it requests it:

```
* INIT44  GTDV DEFINE NEXT INPUT TAPE DEVICE.
        REPLY (UD <DEVICE>, CR)
        ?R?UD MT01
```

3. When the SAVE operation is over, the message INIT20 is issued again.

## B.6 UTIL Error Messages

### NOTE:

Where the meaning of each message is described:

S = Applies to the SAVE function.

R = Applies to the RESTORE function.

B = Applies to both SAVE and RESTORE.

```
INIT18  <extid>  FILE : <file name> : TOO MANY EXTENTS
```

meaning: (S) During the disk <extid> scan, if it is discovered that a file has too many extents or is badly described in the disk VTOC, then it is not possible to save the disk under SIP.

action: Try again with another disk in the installation.

```
INIT19  <extid>  FILE : <file name> : ILLEGAL DESCRIPTION IN VTOC
```

meaning: (S) During the disk <extid> scan, if it is discovered that a file is badly described in the disk VTOC, then it is not possible to save the disk under SIP.

action: Try again with another disk in the installation.



INIT30 UTIL <util function> FUNCTION ABORTED

meaning: (B) After messages INIT35, INIT18, INIT19 the <util function> is aborted by the system.

action: Try again with another SAVE image or another target media.

INIT31 UTIL <util function> FUNCTION CANCELLED BY OPERATOR

meaning: (B) After answering CR at the requests INIT44, INIT69, or N to the requests INIT26, INIT39 the <util function> is aborted.

action: Try again with another SAVE image or another device.

INIT32 <extid> FILE: <file name> SEQ <tape number> MISSING

meaning: (R) The media mounted on the device <extid> defined at the request INIT24 are not in sequence.

action: Mount the media in sequence.

INIT33 <extid> FILE: <file name> SEQ <tape number> EXPECTED

meaning: (R) An INPUT media containing the <file name> and the SEQ <tape number> is expected.

action: At the request INIT44 , mount the expected INPUT media.

INIT34 <extid> ALREADY ALLOCATED

meaning: (B) The INPUT device <extid> defined at the request INIT24 is already allocated as the first INPUT device.

action: Define another INPUT device at the request INIT24.



INIT36 <extid> FORMAT MANDATORY

meaning: (R) This message is issued:

- if the disk <extid> format does not correspond to the SAVE image.
- if only the default partition is defined on the target disk <extid>.

action: Answer Y after the INIT25 request that follows, or try again on another target disk.

INIT37 <extid> NOT ENOUGH SPACE FOR RESTORE FUNCTION

meaning: (R) The target disk <extid> is too small to restore the SAVE image.

action: Define another OUTPUT MASS STORAGE device at the INIT44 request that follows, or change the SAVE image.

\* INIT39 <extid> RETRY IO AT CHECK POINT? REPLY (Y,N)

meaning: (R) After the I/O error message INIT65 on the device <extid>, and if a retry is possible, this message is issued.

action: According to the error, the operator can clean the reading driver, or set the device <extid> to the Ready state, and answer Y for a retry. If he answers N, or if the retry is not effective, the RESTORE function will abort.

INIT40 <extid> FILE <file name> NOT COMPLETELY RESTORED

meaning: (R) If a BLOCK F5 is detected on the SAVE image <file name> (BLOCK indicating that an I/O error occurred during the disk save under GCOS 7), SIP stops restoring.

action: After a disk save under GCOS 7, check that no I/O error occurred during the save operation, if the SAVE image is to be restored under SIP. Regenerate a correct SAVE disk image.



INIT43 <extid> DEFINED DENSITY NOT SUPPORTED BY THIS DEVICE

meaning: (S) This message is issued after the INIT24 request, if the given device <extid> does not support the density chosen during the definition of the first output device.

action: Mount the media on a device which supports the expected density defined in the INIT24 request.

INIT45 <extid> MEDIA EXPECTED : <vsn>

meaning: (S) The media mounted on the device <extid> was changed after the definition of its parameters and before its use by SIP.

action: After the INIT44 request that follows, mount the expected media <vsn>, or cancel the request.

INIT50 <extid> ILLEGAL MASS STORAGE MODEL

meaning: (B) The mass storage device <extid> is not supported by the UTIL function. Only FSA disks are supported for the SAVE and RESTORE functions.

action: Define another device or check the IRT (Installation Resource Table).

INIT51 HDLR FATAL ERROR <alpha-8> : IC= <instruction counter>

meaning: (B) This message is issued when SIP aborts at the <instruction counter>. <alpha-6> explains the SIP exceptions:

ABSCCE	Absolutize the CCE error. It is probably an error in SIP itself.
BADCALL	Wrong procedure call. It is probably an error in SIP itself.
LSYSFAIL	The INIT54 request to define the ISL device has been cancelled.
MMOVFLW	Main memory overflow (no more space for SIP internal buffer allocation). It is probably an error in SIP itself.

action: For the errors ABSCCE, BADCALL, MMOVFLW call a Bull specialist.  
For LSYSFAIL error, re-initialize the system with the right ISL device.





INIT55 <extid> ILLEGAL DEVICE TYPE. ONLY <extyp>/<extyp> ALLOWED

meaning: (B) This message is displayed after the requests INIT20, INIT24 or INIT44, if the device type of <extid> does not correspond to the expected device type <extyp>.

action: At the next request, define a device with an expected device type.

INIT56 <extid> UNKNOWN RESOURCE

meaning: (B) This message is displayed after the requests INIT20, INIT24 or INIT44, if the device <extid> is not described in the IRT.

action: At the next request, define a device that is described in the IRT.

INIT63 <extid> MEDIA IN WRITE PROTECT

meaning: (B) The OUTPUT media mounted on the device <extid> is in write protect mode. This message is followed by the request INIT44.

action: Set the media in write permit mode and mount it again.

INIT65 <extid> IO ERROR : PCLC= <plc> CCE= <word1><word2>  
EV= <event type> PSB= <psb0 to psb3> [DSB= <dsb0  
to dsb3> <dsb4 to dsb7> ... ]

meaning: (B) This message is issued after an access to the device <extid>:

- a. when the operator cancels the request INIT89
- b. if an I/O ERROR occurs during a device access

For both a) and b) an I/O error message with device Detailed Status Byte (if available) is displayed and retried, or new devices are requested (INIT44, INIT54).

action: Call the Bull Customer Service.



INIT66 <extid> IO ERROR : PCLC= <pclc> CCE= <word1><word2>  
CP DISCONNECTED : ELAPSE TIME= <timer value>

meaning: (B) After a Channel Program connection by the <pclc> path, the device <extid> has not sent any termination message although the <timer value> has elapsed. The CP is disconnected by SIP.

action: Check the device state. Call the Bull Customer Service.

INIT67 <extid> CONNECT ERROR : PCLC= <pclc> CCE= <word1><word2>  
PHYSICAL CHANNEL NOT OPERATIONAL/LC ENQUEUED  
OR ACTIVE/TERMINATION NOTIFICATION PENDING

meaning: (B) SIP accesses a device <extid> that has one of the following characteristics:

- a. connected to a failed controller
- b. connected to a controller which belongs to a non-initialized IOC
- c. after the message INIT66, the CP disconnection has failed

The request INIT44, or INIT54, is issued in order to retry on another device if possible.

action: Check the IOC or controller state and perform INIT again if the retry on another device is not possible.

\* INIT69 <extid> RECOVER.REPLY (UD,CR)

meaning: (B) If the <extid> peripheral (tape, disk, printer) is not ready, SIP loops on this message.

action: Put the device in the READY state and reply UD. If this is not possible, the answer CR sends the device state with the INIT65 message.

INIT74 <extid> DEVICE RESERVED

meaning: (B) The device <extid> is seized by another system. An INIT24, INIT44, or INIT54 request follows this message.

action: Define another device at the next request, or release this device on the other system (with the command HDV under GCOS 7) and use the device again.



INIT80 <extid> IMPORTANT : VOLUME IN AN UNSTABLE STATE

- meaning: (B) This message is issued when SIP opens a file on the disk <extid> and it detects that the media is in an unstable state.
- Either an unstable bit is present in the volume label because the disk has not been completely restored after a RESTORE operation under SIP or GCOS 7
  - Or, for FSA disks, the disk characteristics do not correspond to the logical characteristics stored in the VTOC).
- SIP will continue to work with this disk but unpredictable results may occur when it is used.
- action: Under GCOS 7, check the disk and rebuild it by using the SAVE\_FILE and RESTORE\_FILE utilities.

INIT86 UTIL INVALID RESPONSE

- meaning: (B) This message is issued after any SIP request if the answer given is invalid.
- action: Call the Bull Customer Service.

INIT91 <extid> MEDIA <usn> : ILLEGAL FILE SAVE

- meaning: (R) The SAVE file accessed by SIP on the media <usn> has an illegal format.
- action: Define an input media containing a SAVE image standard to the VOLSAVE FILE FORMAT for a GCOS 7 FBO release (this image must be generated using SAVE\_DISK under GCOS 7 or SAVE under SIP).



INIT97 <extid> GCOS INVALID VERSION: <version> VERSION EXPECTED

meaning:

If <version> = V8.1: if system S1 is initialized for the Millennium environment Test, the GCOS 7 version must be at least V8 TS 8560.

If <version> = V9.1: for DPS 7000/TAxXX/TxxxC, the GCOS 7 version must be at least V9 TS 9662.

In MANU mode: this message is followed by the INIT54 request.

In AUTO mode: SIP aborts with the INIT51 message (LSYSFAIL).

action:

At INIT54 request:

If <version> = V8.1: define a new ISL device with GCOS 7 version, which supports Millennium environment Test.

If no V8.1 GCOS 7 version is available:

1. Cancel the INIT54 request and
2. Restore a new GCOS 7 version 8.1 from the system S0.

If <version> = V9.1: define a new ISL device with GCOS 7 version, which supports TAxXX/TAxXXC (model 86xx).

If no V9.1 GCOS 7 version is available:

1. Cancel the INIT54 request and
2. Restore a new GCOS 7 version 9.1 with SIP/UTIL or contact the Bull Customer Service.

INIT98 <extid> FILE : <file name> IS LOGICALLY UNSTABLE

meaning:

(S) During the disk <extid> scan, SIP warns the operator if a file was badly closed at the previous GCOS 7 session. The SAVE function is not aborted, but unpredictable results may occur when the SAVE file is used again.

action:

Keep this information, which may be helpful when you next use the SAVE file.



INIT99 <extid> FILE : <file name> IS MULTI VOLUME

meaning:

(S) During the disk <extid> scan, SIP warns the operator if a file was stored on several media. Only the data stored on the saved disk will be stored in the SAVE file. The SAVE function is not aborted, but unpredictable results may occur when the SAVE file is used again.

action:

Keep this information, which may be helpful when you next use the SAVE file.





---

## C. Disk VOLPREP Procedure

### C.1 Introduction

This Appendix is intended as an aid to installation for new users. It describes the procedure for preparing volumes to be used in the installation procedure, as well as providing an example of the correct technique to be followed. A complete description of the VOLPREP utility (the GCL procedure is PREPARE\_DISK) may be found in the *Data Management Utilities User Guide*.

Three types of VOLPREP may be identified:

1. A complete VOLPREP, with surface formatting. This type of VOLPREP automatically assigns defective blocks, and takes at least 10 minutes of elapsed time, possibly more, depending on disk capacity and machine load.
2. A quick VOLPREP that bypasses surface formatting, but which specifies the BADBLOCK parameter. This type is used to reassign additional bad blocks. The new reassigned blocks are only those that have been defined explicitly in the BADBLOCK parameter of the VOLPREP statement. All previously-set defective blocks are preserved as such. This type of VOLPREP typically executes in a few seconds.
3. A quick VOLPREP which bypasses surface checking and does not specify the BADBLOCK parameter. This is used to deallocate all the existing files on a disk, or to change the MEDIA name of the disk. All previously-set defective blocks are preserved as such. This type of VOLPREP is used typically at the beginning of a restartable disk-loading job, to eliminate any files created by a previous, and perhaps aborted, execution of the job.



---

## C.2 VOLPREP Activation

If the volume does not already have a GCOS 7 label, the following message is displayed when the disk is READY:

```
MSxx NON STANDARD VOLUME
```

In this case, the operator should reply:

```
NV MSxx md
```

where xx is the drive number, as given in the NON STANDARD VOLUME message, and md is the MEDIA name to be given to the disk (of up to 6 alphanumeric characters in length).

The following message is then sent:

```
MSxx md PREMOUNTED NSTD
```

If, on the other hand, the volume already has a GCOS 7 label, the message is:

```
MSxx md PREMOUNTED
```





---

## D. Maintenance in a Protected Environment

### D.1 Introduction

The GCOS 7 Operating System allows the protection of objects (i.e., files, directories and volumes) through the Access Rights Facility. Details of this facility are given in the *Data Security User's Guide*.

Use of this facility, for objects involved in GCOS 7 software maintenance, requires advance planning and an understanding of all the implications. This Appendix describes the necessary considerations and, where appropriate, offers recommendations.



## D.2 Cataloging System Files

All GCOS 7 system files are cataloged, with the exception of SYS.IN and SYS.OUT. Cataloged system files may be divided into two groups.

### D.2.1 SYS.CATALOG Files (Group 1)

The files cataloged in SYS.CATALOG are all GCOS 7 system files prefixed by SYS. The contents of SYS.CATALOG are handled by the maintenance functions.

All these files are cataloged at installation time, and must never be uncataloged by the user.

The user may set access rights on these files, using the GIUF function PROTECT\_GCOS. Execution of this function requires that the owner of the SITE.CATALOG be the project SYSADMIN.

The system and site files are supplied with certain minimum access rights, as shown in Table A-5 at the end of Appendix A. These minimum access rights may be added to, or increased, by SYSADMIN. They may not be reduced or cancelled by the user.

### D.2.2 SITE.CATALOG Files (Group 2)

The system site files cataloged in the SITE.CATALOG are:

- SITE.HELP
- SITE.IN
- SITE.STARTUP
- SITE.CATALOG itself
- and, if they exist, SITE.HALOCK, SITE.CMS\_\*, and SITE.MIRLOG.

Some other system files can be cataloged in SITE.CATALOG but are not managed under GIUF. Instead, they are managed directly by system commands (SITE.QUOTA for example).

SITE.CATALOG is handled by dedicated maintenance functions; however, the user has control over the cataloging of SITE.HELP, SITE.IN and SITE.STARTUP. It is also the user's responsibility to maintain this catalog information whenever necessary. The user may wish to locate the SITE.CATALOG on a volume not handled by TAILOR. In any case, it is the user's responsibility to set access rights on all these files.



### D.3 Manipulating the SITE.CATALOG

After creating a new P-Set/P2-Set, the user must transfer the contents of the SITE.CATALOG file (from the site of the old P-Set/P2-Set) using the COPY\_SITEFILES function (described in the chapter *Installation and Updating Facility*). If the original SITE.CATALOG file was protected, the transferred file will also be protected.

The other maintenance utilities, which manipulate already-built Sets, will abort with the return code "ARVIOL" (i.e., Access Rights VIOLation) if any attempt is made to move the SITE.CATALOG's contents from a protected Set to an unprotected Set. In this case, the user must either set access rights on the unprotected Set, or remove them from the protected one.



## D.4 The SYSADMIN Project

The user is recommended to protect the system with the GIUF function PROTECT\_GCOS. Adding the OWNER attribute to SYSADMIN may protect the volumes that contain GCOS 7 system files. If this is done, operations at file and volume level (for example, automatic allocation of space for a temporary file) can be performed by the SYSADMIN project only.

Most maintenance jobs include file or volume level operations and, if the files are protected, these jobs can be executed by SYSADMIN only. The maintenance jobs are divided into two groups:

- Group 1:                      Jobs including functions that are always restricted to SYSADMIN (BUILDP, UPDATE\_GCOS, LEVEL, ...).
- Group 2:                      Jobs that are run under projects other than SYSADMIN (certain management jobs).

The projects under which these jobs may execute are controlled by the project under which the user submits the GIUF function. The Group 1 functions always restricted to SYSADMIN are:

```
BUILDP
BUILDP2
PROTECT_GCOS
UPDATE_GCOS
LEVEL
```

These functions are rejected whenever submission is attempted under any project other than SYSADMIN.

At delivery time, two users are recorded under the project SYSADMIN:

- GCOS7
- SYSADMIN.

Whenever required, the user may modify the user names attached to the project SYSADMIN.

Project identity is not checked when submitting a function belonging to Group 2. For this reason, it is recommended that they be submitted under the project SYSADMIN nevertheless, whenever access rights are validated on the site.



---

## Glossary

### B

#### Basic System

The initialization firmware and software, generated under GCOS 7 and personalized according to the hardware configuration of the site.

The Basic System files are:

SYS.FW.MSP (not available for XTA systems)  
SYS.FW.WSP (not available for XTA systems)  
SYS.HUB (not available for XTA systems)  
SYS.HUB7 (not available for XTA systems)  
SYS.HUBG

**NOTE:**

SYS.HUB7 is loaded by the BSR P2 and mandatory for the  
DPS 7000/TAxxx/TAxxxC, and DPS 7000/MTx2 and up.

#### BSR

The Basic System Release, which contains the source components of Firmware. The Firmware system files, the names of all of which begin with SYS.FW.\* , are detailed in Appendix A.



## C

### CCE

Channel Command Entry. This is a basic element of a channel program, that describes either positioning or transfer commands for the controller.

### CDA 7

Cache Disk Array for DPS 7000 systems.

### CKD

Count Key Data type disk.

### CP

Channel Program. This is a set of headers and CCEs. The I/O complex (represented by a PIOBLOCK) handles the logical CP, the physical CP is connected to the hardware.

## D

### Disk Set

A set of disks required to run GCOS 7. The disks within any single set are interdependent and so do not require any other disk in order to run in an operational environment.

No system file may be duplicated within any single set.

See P-Set, P2-Set, R-Set.

### Distribution Center

The office from which the user receives the G-MEDIA, K-MEDIA, STARTER-MEDIA, DOMAINS-MEDIA, P-MEDIA, SET-MEDIA, FW-MEDIA, EC-MEDIA, and INSTAL-MEDIA (for stand-alone Domains media).

### Domain

A Domain contains the files required by a given part of the system. This concept is used by most IUF functions.

The user's system may have up to five Domains for system files: GCOS, DSA, OLTD, GSF and FIRMWARE.

### DOMAINS-MEDIA

A set of media containing the three Domains: OLTD, DSA, and GSF.

## E

### EC-Media

Contains emergency corrections and is used by the patching procedure relevant to the system file Domain concerned.

**F****FBO (Fixed Block Organization)**

GCOS 7 file architecture characterized by fixed-length file blocks occupying the whole file space. Applied to volumes, FBO indicates the GCOS 7 disk volume format where an entire volume's space is formatted with fixed-length data blocks. This organization is applicable to all DPS 7000 systems.

**FSA**

Fixed Sector Architecture, the only type of disk format supported for system files.

**FW-MEDIA**

Contains the Firmware files belonging to the FIRMWARE Domain and is provided for backup purposes for new users.

**G****G-Media**

The media containing the delivered GCOS 7 system, and is a set of tapes or cartridges for use by the Installation and Updating procedures. The G-MEDIA is not a disk-image and so cannot be restored using the SIP function RESTORE in UTIL mode (which is described in Appendix D).

**GCOS 7**

In the context of IUF, GCOS 7 is considered in a larger sense than usual; that is, as including products not strictly belonging to it. These products are:

- The Distributed Network Supervisor (DNS) running on the Datanet.
- The Distributed Network Supervisor-Extended (DNS-E) running on the MainWay.
- The FDDI Communications Processor 7 (FCP7), which allows a DPS 7000 to connect to a LAN Extender 7, or to a MainWay communications processor.
- The CNS7 supervisor running the CNP7(s).
- The On-Line Tests and Diagnostics (OLTD).
- The Firmware Generation files (FW).
- The GCOS 7 Service Facility (GSF).

**GSF**

GCOS 7 Service Facilities deal with the Installation and Updating procedures for the GSF system file Domain required to support the mandatory GSF service tools used for the maintenance of GCOS 7.



## H

### Hub Files

See Basic System.

## I

### IOC

Input Output Controller

### IRT

Installation Resource Table

### ISL (Initial system Load)

At ISL time, the logical and physical I/O paths are validated for each device defined in the hardware configuration, as described in the SRST (System Resource Table). The devices and controllers are initialized, and the software tables are updated accordingly.

ISL is executed at each system restart, prior to GCOS 7.

## K

### K-Media

The K-Media is the media containing the unbundling key, which causes GIUF processor to validate the GCOS 7 software products ordered by the client. If this K-Media is a piece of paper, the key is entered at the console (only for the IUF functions ADD\_MI and DELETE\_MI, when the KEY\_CODE parameter is used).

The unbundling key is used by the VALIDR, VALIDP, VALIDP2, ADD\_MI, DELETE\_MI and UPDATE\_MI functions, and is client- and site-dependent.

The K-MEDIA is defined as:

```
MD=MIKEY  DVC=dvc  EFN=MIKEY
```

## L

### LSS (Large Storage Sub-system)

Large capacity FSA disk of 1.4 or 2.5 giga bytes.





## M

### **Master Set**

Specifies a disk set (a P-Set, a P2-Set, or an R-Set) belonging to the master system; see Multi-system, below.

### **MI Configuration Code (MI CONF.CODE)**

This code identifies an MI (Marketing Identifier) software configuration, and is created when the MIs are validated on an x-Set disk at Installation time. It consists of the time and date of the MI validation.

### **Module Correction**

Module corrections are unique to the provision of a new Technical Status. They replace entire modules, and are applied via the Updating procedure.

### **Multi-System**

Environment in which a user may install and maintain disk Sets belonging to several DPS 7000 systems.

One of these systems is defined as being the MASTER system and all others are considered as being SLAVE systems.

## P

### **P-MEDIA**

This is a VOLSAVE of the factory-prepared P-Disk.

A set of tapes or cartridges supplied with a new DPS 7000 that does not have a pre-loaded disk and containing a factory-prepared image of the P-Set. This image is obtained by running the VOLSAVE utility or the GCL procedure SAVE\_DISK.

The P-MEDIA is used on the customer site to create the P-Set.

### **P-Set**

See Production Set.

### **P2-Set**

See Production Set and P2P-Configuration.

### **P2P-Configuration**

A configuration which consists of one P-Set and one P2-Set. Each Set is used alternately (in "flip-flop" mode) to run GCOS 7. The system alternates from one Set to the other after application of a Technical Status to the non-running Set.

**PCLC**

Physical Channel Logical Channel.

**PO-Configuration**

A configuration that consists of only one P-Set. There is no R-Set disk in such a configuration, which makes this the least expensive type of configuration, but also the least secure.

**Production Set**

A Production Set consists of one or more disks containing all the system files necessary to operate GCOS 7. A site may have more than one Production Set.

**R****R-Set**

See Reference Set.

**R-Set On Line**

The R-Set is said to be on line when its disk space remains allocated on a fixed disk of device class MS/FSA.

**R-Set Off Line**

The R-Set is said to be off line on a fixed disk configuration when its disk space is allocated to the user. Its image is saved on tape or cartridge and restored when necessary.

**Reference Set**

A disk Set containing the GCOS Domain files and other Domain files necessary for IUF operation. This Set is used to create a first or additional P-Set by using the BUILD P function. The R-Set is monodisk.

**RP Configuration**

A configuration which consists of one R-Set and one or several P-Set(s). Only the P-Set may be used to run GCOS 7 and Technical Statuses are applied to this Set. The R-Set is available as a backup in the event of the P-Set being damaged.

**S****SDA 7**

Small Disk Array for DPS 7000 systems.

**Set-MEDIA**

A save of the system files (belonging to all the Domains applicable) on tape or cartridge. The save is performed by the IUF function SAVE\_SET. The contents of the tape (or cartridge) may only be restored by the IUF function RESTORE\_SET (or by the job INSTAL\_RSET).

**Ship Media**

This is a component delivered with the system and used to perform the first initializations until the Firmware HUB (Basic System) files have been generated.



**Slave Set**

Specifies a P-Set, P2-Set, or R-Set belonging to a slave system. See Multi-system.

**SOP**

Service Operator Panel.

**SRST**

System Resource Software Tables.

**Starter-MEDIA**

Part of the material supplied to the user by the Local Distribution Center. The media contain a disk image belonging to the same family as the first work disk to which it is to be restored.

**System Disk**

An FSA disk containing loaded files which enable the user to start GCOS 7 and obtain the SYSTEM READY message. The set of system disks constitutes the Production Set.

**T**

**TA (Twin Architecture)**

Architecture of the DPS 7000/TAxxx/TAxxxC platform based on GCOS 7 and OPEN processors.

**Technical Level**

The Technical Level of a Set is determined by the Technical Status applied to it. (See definition of Technical Status.)

**Technical Status (TS)**

A TS consists of module corrections, and may be applied to any P-Set or P2-Set to be used as input to the Updating procedure.

The number of the Technical Status of the GCOS, OLTD, FW, and GSF domains, is displayed after the DISPLAY\_STATUS command of GIUF.

**TS-MEDIA**

The TS-MEDIA contains a Technical Status of a given Domain (GCOS, FW, OLTD, DSA or GSF) to be used by the Domain Updating procedures. For the GCOS Domain, the TS-MEDIA is the same as the G-MEDIA.



## V

### **VBO (Variable Block Organization)**

The GCOS 7 file architecture characterized by variable-length file blocks occupying the file space. Some file organizations also contain blocks holding physical keys. Applied to volumes, indicates the GCOS 7 disk volume format where a volume's space contains variable-length data blocks and/or keyed physical records.

**Only dedicated for data user.**

## W

### **WORK DISK**

IUF uses work disks that have been preformatted, with replacement tracks already allocated. The advantage is that IUF can clear the disk with the GCL procedure PREPARE\_DISK or the JCL utility VOLPREP. However, the user is unable to change the disk's name.

A work disk may be temporary or may become a permanent disk belonging to a Set.

## X

### **XTA (eXtended Twin Architecture)**

Architecture of the DPS7000/XTAxxx platform based on GCOS 7.



---

## Index

### A

#### Access rights

- For system files A-116
- Running Set 4-63
- SITE.CATALOG 4-45, 4-74, 4-84, 4-93
- SYS.CATALOG 4-74
- System files 5-3

#### ACCOUNT

- CONFIG statement 5-12

#### ACTSIZE

- CONFIG statement 5-14

#### ADD\_MI

- GIUF function 3-7, 4-7, 10-9

#### Add-on installation

- DIUF facility 10-9

#### Add-ons 3-5

#### AEP records A-33

#### After Journal

- Transfer 7-3

#### ATTACH

- GCL command 7-8

### B

#### Backing Store Classification A-19

#### Backing store files

- SYS.LIB[i] A-93
- SYS.PVMF[i] A-98
- SYS.TVMF[i] A-113

#### Backup to a previous release 2-18

#### Basic system files

- list g-1
- SYS.HUBG A-80

#### Before Journal

- maximum number of jobs 5-16
- System file A-88

#### BJSIMU

- CONFIG statement 5-16

#### BKST

- TAILOR command 6-25

#### BKST1

- TAILOR command 6-26

#### BLOAD

- FGF function A-80

#### BSR

- definition g-1

#### BUILD\_BKST 4-10, A-17

#### BUILD\_CMS

- GIUF function 4-12

#### BUILD\_HALOCK

- GIUF function 4-14

#### BUILD\_JAS

- GIUF function 4-16

#### BUILD\_JRNAL 4-17

#### BUILD\_LIB 4-19, A-93

#### BUILD\_MIRLOG

- GIUF function 4-21

#### BUILD\_PVMF 4-23, A-98

#### BUILD\_QM 4-25

#### BUILD\_QM\_BACKUP 4-27

#### BUILD\_SPDUMP 4-29

#### BUILD\_SPOOL 4-30

#### BUILD\_SYSDUMP 4-32, A-109

#### BUILD\_TVMF 4-34, A-113

#### BUILD $\bar{P}$ 1-3, 10-2, A-26

- GIUF function 4-36

#### BUILD $\bar{P}$ 2 2-1, 7-3, 10-2, 10-7, A-26

- GIUF function 4-38

**C**

Cache Mechanism 5-71  
 CATFL  
   TAILOR command 6-27  
 CATVOL  
   TAILOR command 6-21  
 CCE  
   definition g-2  
 CDA 7  
   definition g-2  
 CKD  
   definition g-2  
 CKD/FBO B-1  
 CLC  
   TAILOR command 6-29  
 CLC (Cumulative List of Corrections) 10-32  
 CMS  
   TAILOR command 6-30  
 CNP7  
   Installing files 10-1  
   SYSGEN job 10-7  
 CNS7  
   Release installation 10-4  
   Release media 10-16  
   TS media 10-19  
 Coexistence of VBO and FBO 1-4  
 CONFIG  
   Default values 5-79  
 CONFIG and ISL 5-4  
 COPY\_FILE  
   GCL command 7-8  
 COPY\_SITEFILES  
   GIUF function 4-41  
 CP  
   definition g-2

**D**

DEFORG  
   CONFIG statement 5-18  
 DELETE\_MI  
   GIUF function 3-7, 4-48  
 DELETE\_TS  
   DIUF function 10-4  
 Deletions 3-5

DETECT\_SITE  
   GIUF function 4-50, 4-97, 4-98  
 Device class  
   Cartridge 1-4  
   Tape 1-4  
 DEVTRACE  
   CONFIG statement 5-19  
 Disk set  
   definition g-2  
 DISPLAY\_STATUS  
   GIUF function 4-57  
 DIUF  
   P2P-Configuration 10-3  
   R-Set Considerations 10-3  
 DIUF (DSA Installation and Updating Facility)  
   10-1  
 DJPCTL  
   CONFIG statement 5-21  
 Domain  
   BSR (Basic System Release) 1-6  
   definition g-2  
   DSA (Distributed Systems Architecture)  
     1-6  
   FW (Firmware) 1-6  
   GCOS 1-6  
   GSF (GCOS 7 Service Facility) 1-6  
   OLTD (On Line Test and Diagnostics) 1-6  
   Technical Status 4-57  
 DUALSYS  
   CONFIG statement 5-23  
 DUMPACT  
   CONFIG statement 5-25

**E**

EC (Emergency Corrections) 8-1  
 EDIT  
   GIUF function 4-59  
 Emergency Corrections 3-5  
 ENTER\_DIUF  
   GCL command 10-4  
 Environment 1-4  
 EOPEN  
   TAILOR command 6-63  
 ERLOG  
   TAILOR command 6-31

**F**

FBO  
  definition g-3  
  Disks 2-1  
FBO/CKD B-1  
FBO/FSA B-1  
FGF (Firmware Generation Facility) 9-1  
FILDUPLI  
  Data management utility A-105  
FILEOPT  
  CONFIG statement 5-27  
FILREST  
  Data management utility A-105  
FILSAVE  
  Data management utility A-105  
FILSHARE  
  CONFIG statement 5-30  
Firmware Domain  
  Installation 9-1  
Fixed Disks 2-7  
FSA  
  definition g-3  
FSA/FBO B-1  
FSE  
  GIUF function 4-61  
Full Screen Editor 4-61  
FW  
  System file domain A-44  
FWGEN  
  Tailor command 6-61

**G**

GAC  
  CONFIG statement 5-32  
GIUF  
  Functions 4-4  
  Jobs 4-4  
GIUF Command  
  Environment 4-3  
  Examples 4-2  
  Syntax 4-2  
G-MEDIA 4-50, 7-1, 7-7  
GSF  
  definition g-3

GSFGEN  
  TAILOR command 6-62  
GTP (Generalized Transfer Processor) 5-21

**H**

HALOCK  
  TAILOR command 6-32

**I**

INPUTCTL  
  CONFIG statement 5-34  
INSTAL  
  DIUF function 10-4, 10-19  
  GSF function 11-2  
INSTAL\_ALL  
  before beginning 2-1  
  GIUF job 4-111  
INSTAL\_GCOS  
  GIUF job 4-113  
INSTAL\_PSET  
  Alternative installation procedure 2-10  
  before beginning 2-1  
  clearing disk 2-12  
  GIUF job 4-114  
  job 2-12  
INSTAL\_RSET  
  Alternative installation procedure 2-10  
  GIUF job 2-16, 4-115  
  updating strategy 7-4  
Installation Procedure  
  New Systems with Pre-Loaded Disk 2-7  
IRT  
  definition g-4  
ISL  
  definition g-4  
  Tailor command 6-33  
ISL RESTORE session 4-48, 4-64, 4-65, 4-72  
IUF  
  functions 1-5  
  jobs 1-5  
IV (INVOKE)  
  GCL function 5-5

**J**

JAFTER  
     TAILOR command 6-34  
 JAS  
     TAILOR command 6-36  
 JBEFORE  
     TAILOR command 6-37  
 JOBCLASS/JBCLASS2  
     CONFIG statements 5-36  
 JORSIZE  
     CONFIG statement 5-45  
 JTRA file A-89

**K**

K-MEDIA 4-7, 4-48, g-4  
 KNODET  
     TAILOR command 6-38  
 KPN (Known Problem Number) 10-30

**L**

LEVEL  
     GIUF function 3-7, 3-9, 4-63, 7-1  
 LEVEL\_MI  
     GIUF function 3-7, 4-65  
 LIB  
     TAILOR command 6-39  
 LIBALLOC statement A-1  
 LIST  
     DIUF function 10-4, 10-27  
 LIST\_MI  
     GIUF function 4-66  
 LOAD\_FACTORY 10-28  
 LOADFACT  
     DIUF function 10-4  
 LOGC  
     TAILOR command 6-40  
 LSYS  
     CONFIG statement 5-48

**M**

MAIN operator 1-4  
 MAINTAIN\_SYSTEM 4-72  
 Master Set 3-1, 3-4, 3-5, 4-6, g-5  
 Master System 3-1, 3-5  
 MAXCAT  
     CONFIG statement 5-49  
 MAXDEF  
     CONFIG statement 5-50  
 MAXFILE  
     CONFIG statement 5-51  
 MAXIODEV  
     CONFIG statement 5-52  
 MAXJOB  
     CONFIG statement 5-53  
 MAXTAPE  
     CONFIG statement 5-54  
 MAXTASK  
     CONFIG statement 5-55  
 MDACL 4-74  
 MI Configuration 4-63, 4-97, 4-101  
 Migration Tool 4-45, 4-84  
 MIRLOG  
     TAILOR command 6-41  
 MODIFY\_CONF  
     GIUF function 2-17, 4-68  
 MSC-4 Controller 2-7  
 Multi-device classes 1-4  
 Multiple MIs 1-4  
 Multi-programming  
     Environment 3-9  
 Multi-system  
     Environment 3-5  
     Installation 3-2  
     Set 3-3  
 MULTLEV  
     CONFIG statement 5-57  
 MWINLIB  
     GCL command 4-3  
 MWLIB  
     GCL command 4-3



**N**

NRDVOL  
TAILOR command 6-19  
NRDVOL1  
TAILOR command 6-20

**O**

OLTD  
Domain 9-2  
OWCLASS  
CONFIG statement 5-58  
OWDEVICE  
CONFIG statement 5-60  
OWDFLT  
CONFIG statement 5-64

**P**

P2P Configuration 1-3, 2-1  
P2-Set  
definition g-5  
Disk occupancy 2-1  
PATCH\_HLMLIB  
GIUF function 4-70, 8-2  
PATCH\_SYSTEM  
GIUF function 4-72, 8-2  
Patching  
DIUF procedure 10-24  
GCOS 7 files 8-1  
Patching procedures  
GCOS Domain 8-1  
PATCH-MEDIA 4-50  
P-MEDIA B-1  
Restoration Procedure 2-5  
PO-Configuration 1-3, 7-6  
PREALLOC statement A-1  
PRLOG  
CONFIG statement 5-67  
PRLOGC  
CONFIG statement 5-69  
PROTECT\_GCOS D-4  
GIUF function 4-74  
Protection D-4

P-Set 2-1, 2-5, 2-10  
definition g-5  
PVMF  
TAILOR command 6-42

**Q**

QM  
TAILOR command 6-43  
QMBACK  
TAILOR command 6-44  
QUEUED  
CONFIG statement 5-71

**R**

Recovery Procedures 12-1  
Remote Maintenance  
Access to GSF 11-1  
Resident disks 1-3  
RESTORE\_SET  
GIUF function 3-5, 3-9, 4-76  
RESTORE\_SITE  
GIUF function 4-78  
RESTORE\_SITEFILES  
GIUF function 4-80  
RP-Configuration 1-3, 4-69  
RSDVOL  
TAILOR command 6-17  
RSDVOL1  
TAILOR command 6-18  
R-Set  
definition g-6  
Running Set 3-9

**S**

SADMOPT  
CONFIG statement 5-72  
SAVE\_DISK  
GCL procedure 12-3, 12-6  
SAVE\_SET  
GIUF function 2-16, 3-4, 3-5, 3-7, 4-87  
SAVE\_SITE  
GIUF function 4-90



- SAVE\_SITEFILES
  - GIUF function 4-92
- SDA 7
  - definition g-6
- SECOPT
  - CONFIG statement 5-74
- SET-MEDIA 2-16
- SIP (System Initialization Program) B-1
- SITE files 2-10
- SITE.CATALOG
  - access rights 4-45, 4-74, 4-84, 4-93
  - copying 4-41
  - file A-5, D-2
  - PROTECT\_GCOS 4-75
  - require or prevent creation 6-27
  - restoring 4-80
  - saving 4-92
- SITE.CMS\_CDD\_name1 file 4-12
- SITE.CMS\_CDD\_name2 file 4-12
- SITE.CMS\_CDD\_namei file A-7
- SITE.CMS\_CSD file 4-12, A-6
- SITE.CMS\_DS\_name1\_name2 file 4-12, A-8
- SITE.HALOCK
  - building 4-14
  - file A-9
  - require or prevent creation 6-32
- SITE.HELP file 4-41, 4-80, 4-92, 6-45, D-2
- SITE.IN file 6-46, D-2
- SITE.MFT file 6-2
- SITE.MIRLOG file 4-21, 6-41, A-12
- SITE.QUOTA file 6-2
- SITE.STARTUP file 4-41, 4-80, 4-92, 6-49, D-2
- SITHELP
  - TAILOR command 6-45
- SITIN
  - TAILOR command 6-46
- Slave Set 3-9, 4-6
- Slave Set Updating 3-4, 3-10
- Slave System 3-1, 3-4
- Slave Updating
  - Fixed disk environment 12-6
- SPDUMP
  - TAILOR command 6-47
- SPOOL
  - TAILOR command 6-48
- SRST
  - definition g-7
- Streaming mode 2-12
- STUP
  - TAILOR command 6-49
- SWLOG
  - TAILOR command 6-50
- SYIN
  - TAILOR command 6-51
- SYO
  - TAILOR command 6-52
- SYS.BKST file 4-10, 6-25
- SYS.BKST1 file 6-26
- SYS.CATALOG
  - access rights 4-74
  - environment 10-2
  - file A-23
- SYS.CLC file 6-29, A-24
- SYS.DSACONF file 4-41, 4-80, 4-92
- SYS.DSACORE file 4-41, 4-80, 4-92
- SYS.ERLOG file 6-31
- SYS.IN file 6-51
- SYS.JADIR file 6-34, 7-8
- SYS.JASBLUE
  - building 4-16
  - file 6-36, A-86
- SYS.JASGREEN
  - building 4-16
  - file 6-36, A-87
- SYS.JRNAL
  - building 4-17
  - file A-88
  - require or prevent creation 6-37
- SYS.JTRAVAl file A-89
- SYS.KNODET file 6-38
- SYS.LIB file 6-39
- SYS.LIB[i] file 4-19
- SYS.LOGC file 6-40
- SYS.OUT file 6-52
- SYS.PVMF file 6-42
- SYS.PVMF[i] file 4-23
- SYS.QM file 4-25, 6-43
- SYS.QM\_BACKUP file 4-27, 6-44
- SYS.SITE.BIN file 4-41, 4-80, 4-92
- SYS.SITE.SL file 4-41, 4-80, 4-92
- SYS.SP\_DUMP file 4-29, 6-47, A-103



SYS.SPOOL0 file 6-48  
SYS.SPOOLi file 4-30  
SYS.SWLOG file 6-50  
SYS.SYSDUMP file 4-32, 6-54  
SYS.SYSTEM file 6-33  
SYS.TVMF[i] file 4-34  
SYS.URCINIT file 4-41, 4-80, 4-92  
SYSADMIN project D-4  
SYSDUMP  
    TAILOR command 6-54  
SYSGEN  
    CNP7 job 10-7  
SYSTEM disks 1-3  
System files  
    location 1-3  
    Placement 12-1  
System Initialization Program (SIP) B-1  
SYSVOL  
    TAILOR command 6-16

## T

TA  
    definition g-7  
TAILOR  
    COMSET: 6-7  
    Default values 6-6  
TDS 4-75  
Technical Status 4-98  
    definition g-7  
Text Editor 4-59  
TLRCANCEL  
    TAILOR command 6-13  
TLREXEC  
    TAILOR command 6-10  
TLRGEN  
    TAILOR command 6-9  
TLRPRINT  
    TAILOR command 6-11  
TLRRESET  
    TAILOR command 4-39, 6-12  
TLRSTAT  
    TAILOR command 6-14  
GSF 11-1  
Tools  
    DIUF 10-4

TS  
    definition g-7  
TVMF  
    TAILOR command 6-55

## U

UPDATE  
    DIUF function 10-4, 10-19  
    GSF function 11-2, 11-4  
UPDATE\_GCOS 7-6  
    GIUF function 4-50, 4-95, D-4  
UPDATE\_MI  
    GIUF function 4-65, 4-101  
Updated Set 3-9  
Updating 3-5, 3-9  
Updating procedures 7-1  
USAGE  
    GSF function 11-2

## V

VALIDP  
    GIUF function 4-102, 4-105, 10-7  
VALIDR  
    GIUF function 4-108  
VBO  
    definition g-8  
VBO/FBO Coexistence 1-4  
VCAMSCN  
    CONFIG statement 5-76  
VOLDUPLI  
    Data management utility A-105  
VOLORG  
    CONFIG statement 5-77  
VOLPREP  
    Data management utility 12-4, C-1  
VOLREST  
    Data management utility A-105  
VOLSAVE 2-5  
    Data management utility A-105



**X**

XAS  
CONFIG statement 5-78

## Technical publication remarks form

**Title :** DPS7000/XTA NOVASCALE 7000 GCOS7 System Installation and Configuration  
User's Guide Operating System: Installation

**Reference N° :** 47 A2 23US 05

**Date :** **February 2005**

### ERRORS IN PUBLICATION

### SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

Your comments will be promptly investigated by qualified technical personnel and action will be taken as required.  
If you require a written reply, please include your complete mailing address below.

NAME : \_\_\_\_\_ Date : \_\_\_\_\_

COMPANY : \_\_\_\_\_

ADDRESS : \_\_\_\_\_

Please give this technical publication remarks form to your BULL representative or mail to:

Bull - Documentation D<sup>é</sup>pt.  
1 Rue de Provence  
BP 208  
38432 ECHIROLLES CEDEX  
FRANCE  
info@frec.bull.fr

# Technical publications ordering form

To order additional publications, please fill in a copy of this form and send it via mail to:

**BULL CEDOC**  
**357 AVENUE PATTON**  
**B.P.20845**  
**49008 ANGERS CEDEX 01**  
**FRANCE**

**Phone:** +33 (0) 2 41 73 72 66  
**FAX:** +33 (0) 2 41 73 70 66  
**E-Mail:** [srv.Duplicopy@bull.net](mailto:srv.Duplicopy@bull.net)

CEDOC Reference #	Designation	Qty
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		

[ ] : The latest revision will be provided if no revision number is given.

NAME: \_\_\_\_\_ Date: \_\_\_\_\_

COMPANY: \_\_\_\_\_

ADDRESS: \_\_\_\_\_

PHONE: \_\_\_\_\_ FAX: \_\_\_\_\_

E-MAIL: \_\_\_\_\_

**For Bull Subsidiaries:**

Identification: \_\_\_\_\_

**For Bull Affiliated Customers:**

Customer Code: \_\_\_\_\_

**For Bull Internal Customers:**

Budgetary Section: \_\_\_\_\_

**For Others: Please ask your Bull representative.**



**BULL CEDOC**  
**357 AVENUE PATTON**  
**B.P.20845**  
**49008 ANGERS CEDEX 01**  
**FRANCE**

REFERENCE  
**47 A2 23US 05**