BURROUGHS CORPORATION
DETROIT, MICHIGAN 48232

B2000/B3000/B4000

SYSTEMS SOFTWARE RELEASE #6.8

B1000 TO B2000/3000/4000 SERIES SYSTEMS MIGRATION GUIDE

PRINTED IN U. S. AMERICA

12-84

Burroughs Corporation
Detroit, Michigan 48232

Preliminary Form 1090735 September 17, 1984 19:01

Burroughs cannot accept any financial or other responsibilites that may be the result of your use of this information or software material, including direct, indirect, special or consequential damages. There are no warranties extended or granted by this document or software material.

You should be very careful to ensure that the use of this software material and/or information complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used. The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions. additions.

Any comments or suggestions regarding this document should be forwarded to B1000 Migrations - Programming Activity, Systems Development Group, Burroughs Corporation, 460 Sierra Madre Villa, Pasadena, CA 91109.

INTRODUCTION

This manual has been prepared to help you migrate from the B1000 Series Systems (running native software, not CMS software) to the B2000/3000/4000 Series Systems. This manual contains:

1. Differences between the two systems
2. Descriptions of common migration problems and their solutions
3. General recommendations for handling the overall migration

This is version one of this manual. It covers many of the major areas of a B1000 to B2000/3000/4000 migration. Additional sections, along with updated information, will be added to future versions.

This manual is also accessible through your B2000/3000/4000 terminals with the aid of the Online Migration Guide program.

SOFTWARE VERSION

This document pertains to the 10.0 release of the B1000 series systems and the 6.8 release of the B2000/3000/4000 series systems.

ORGANIZATION OF THE MANUAL

The first section of this manual discusses general migration management techniques that apply to all migrations. The subsequent sections give detailed instructions for completing each part of the migration. Each section is independent of the others, except as noted.

The manual is divided into the following sections:

1. Section 1 presents an overview of the migration process.
2. Section 2 discusses the process of file transfer.
3. Section 3 discusses the migration of RPG programs.
4. Section 4 outlines the B2000/3000/4000 SORT Utility program.
5. Section 5 discusses the migration of B1000 COBOL74 programs to B2000/3000/4000 COBOL74.
6. Section 6 discusses the migration of B1000 COBOL68 programs to B2000/3000/4000 COBOL74.
7. Section 7 discusses the migration of DASDL programs.

8. Section 8 discusses the DMSII host language syntax and semantic differences.

9. Section 9 discusses the differences between B1000 and B2000/3000/4000 ISAM and RELATIVE files.

10. Section 10 compares B1000 queue files and B2000/3000/4000 port files.

11. Section 11 discusses the migration from B1000 Remote files.

12. Section 12 discusses the migration of Reader Sorter programs.

13. Section 13 discusses the differences between the B1000 and B2000/3000/4000 file attributes.

14. Section 14 discusses the changes you need to make to migrate the transaction programs and GEMCOS specifications.

15. Section 15 and 16 discuss REPORTER and ODESY.

16. To help with the migration, a list of references and a planning guide are included in the appendices.

MIGRATION ASSISTANCE

If you have a migration problem or solution that is not covered in this document, please send a detailed description to:

Burroughs Corporation
Programming Activity - B1000 Migrations
460 Sierra Madre Villa
Pasadena, CA 91109

If you have an immediate problem, please contact your local Burroughs Technical Representative.

SECTION 1
THE MIGRATION PROCESS

When you upgrade from a B1000 to a B2000/3000/4000 system, it will probably be necessary to make some changes to your software. In support of your B1000 migration to B2000/3000/4000 series systems, Burroughs provides this documentation and an array of time and cost-saving migration software aids.

This section provides you with some tips on migration and a review of the available migration software aids.

We urge you to do a straight migration. Do not be tempted to "fix up" or "enhance" the system while doing the migration. Make only the changes necessary to get your software running on your B2000/3000/4000 system. After the migration is complete and you feel comfortable with your new system, you may wish to make changes to your software to take advantage of the additional features available on the B2000/3000/4000 series systems.

PLANNING is another important element in a successful migration. Planning should start when you sign for your new B2000/3000/4000 system. You should have a plan for the entire migration process. Planning is discussed more fully below.

TRAINING is the third key to a successful migration. We urge you to take advantage of Burroughs education classes before you start your migration. A thorough understanding of B2000/3000/4000 series systems will help the migration effort considerably. Consult your Burroughs Technical Representative for further information on customer education.

In summary, for a good migration make sure you:

1. Make only minimum changes.
2. Plan and organize carefully.
3. Get additional training.

Appendix B provides a planning guide to help you with your planning.

MIGRATION RECORD KEEPING

The migration log was designed to help you with record keeping. A migration journal provides an easy reference source for all members of the migration team.

If the migration log and reports are not used, we recommend using a loose-leaf notebook to allow new information to be added as necessary. Include all information required for the migration in the journal.

The following is a suggested list of types of information that could be contained in your journal:

1. Migration sequence definition.
2. Migration package definition.
3. Translator run instructions.
4. Known translator problems or deficiencies.
5. File naming conventions.
6. System naming conventions.
7. Program naming conventions.
8. Work Flow Language (WFL) run procedures.
9. Data communications implementation information.
10. Data base implementation information.
11. Testing procedures.
12. Acceptance procedures.
13. Technical Information Papers (TIPs).
14. Problems encountered and their solutions.

MIGRATION TASKS

The following are suggested tasks for your migration. You may need to add or remove activities depending on your specific migration needs:

1. Freeze the program library.

   Compile the source code and make a copy of the source and object. No changes should be made to the sources or copy libraries after this point.

2. Prepare the migration package.

   Gather together:

   a. All of the source code.
   b. Copy libraries.
   c. Translation aids.
   d. Migration instructions such as:

      1. New naming conventions.
      2. Data base implementation information.
      3. Data communications information.

3. Deliver the migration package.

   Give the migration package over to the individual or team who has the responsibility for the migration.

4. Translate the source code to clean compile.

   Translate the source code. Generally this is done using one of the translators, but it may be done manually if no tool exists.

Even with the use of a translator, it may be necessary to make some manual changes to achieve a clean compile.

5. Transfer the test data files.

   Retain a copy of the test data for testing purposes. The test data files should be varied enough to ensure a thorough test of the entire program. The files should also be small enough to keep run time short.

6. Perform the unit tests and make the necessary corrections.

   Unit tests test individual programs after their migration. To complete a unit test successfully and efficiently, obtain access to:

   a. Source code.
   b. Documentation.
   c. All input files.
   d. All output files.
   e. All work files.

   The work files are an important part of the debugging process because then you can check intermediate results. Their importance should not be minimized.

   A source program is successfully migrated when the unit test reveals that the results of the migrated program on the B2000/3000/4000 system are identical to the results of the original program on the B1000 system.

7. Perform the system test and make corrections.

   A system test is conducted after the successful compilation of all unit tests of the individual programs which make up the system. A system test requires:

   a. Documentation.
   b. All input files.
   c. All output files associated with the system.

   The system test is successful after it confirms that the results produced by the migrated system on the B2000/3000/4000 system are identical to the results given by the original system on the B1000 system.

8. Update the documentation.

   Update all documentation to reflect the changes made to the applications software.

9. Implement changes to non-frozen programs.

At this time, make the same changes to the migrated programs that were made to the live production programs.

10. Deliver the system to operations.

    Deliver the translated programs and any changes in operating procedure to the operations personnel.

11. Run parallel tests.

    Run identical processing on both the B1000 and B2000/3000/4000 systems. File comparisons may be conducted to verify identical results.

12. Begin live operation.

    Processing continues on the B2000/3000/4000 system. The migration process is complete.

BURROUGHS MIGRATION AIDS

To ease the process of migration, Burroughs has available a wide range of aids for programs and data files. Tools, where they are required, exist for most functions and cover all the commonly used programming languages.

The B1000 migration software aids include:

On-line Migration Manual

Provides a means to interactively view or obtain a hard copy of all or part of the B1000 to B2000/3000/4000 Series Systems Migration Guide and the Migration Aids User Guide as the need for referencing arises.

COBOL68 to COBOL74 Translation

Filters and translates B1000 COBOL68 to B2000/3000/4000 COBOL74.

COBOL74 to COBOL74 Translation

Filters and translates B1000 COBOL74 to B2000/3000/4000 COBOL74.

RPG Translation

Filters and translates B1000 RPG to B2000/3000/4000 RPG.

File Transfer Utilities

Provides a method of transferring library tapes between the

B1000 and B2000/3000/4000 series systems.

File Renaming and Reformatting Utility

Provides an interactive method of renaming and reformatting large number of files transferred from the B1000 to the new host system.

SECTION 2
FILE TRANSFER

File transfer is the process of moving files from one system to another. Because of differences between the systems, it is generally not possible to simply dump files from the host system and load them to the target system. Therefore, file transfer comprises both the physical movement of data between the two systems as well as the achievement of any data or file format changes that may be necessary on the target system.

This section describes the considerations in file transfer between B1000 and B2000/3000/4000 series systems, and the available methods for data transfer and format changes.

FILE TRANSFER CONSIDERATIONS

No data format changes are necessary between the two systems. All types of data with all sign formats accepted by B1000 series systems are also accepted by the B2000/3000/4000 series systems. File formats are, however, different in some cases.

File names on the B2000/3000/4000 series systems are single level names of six or fewer characters. Therefore, files usually require to be renamed when transferred from B1000 to B2000/3000/4000 series systems.

On the B2000/3000/4000 series systems, the record size of a file must be an even number of bytes. This implies that special care must be taken when transferring files with odd record sizes. These files may have to be reblocked for efficiency purposes and programs using these files may have to be suitably modified.

If a file is to be edited by the B2000/3000/4000 EDITOR program, the file must be changed to a format recognizable by the EDITOR. The EDITOR command COPY FROM can be used to convert a disk or pack file to the EDITOR format if the original file consistec of 80 character records, blocked between 1 and 9. If this is not the case, the DMPALL program may be used first. An easier method is available using the SSNAME utility program.

The B2000/3000/4000 series compilers accept cards, tape, disk and pack as valid source input media. The record sizes and block sizes of sequential files must be within the limits for that compiler. The B2000/3000/4000 DMPALL program or the SSNAME utility program may be used to reblock the source input files appropriately.

ISAM, Relative and DMS files have different formats on the two systems. The general method to transfer these files is to unload the data into sequential files on the source system, physically move the sequential files to the target system and recreate the files in the correct format on the target system. Utilities are available to aid the user at different stages.

RPG ADDROUT files may not be transferred. They must be recreated by the B2000/3000/4000 SORT UTILITY program.

FILE TRANSFER METHODS

The following are the utilities which may be used in the process of file transfer:

1. SSCOPY
2. SSNAME
3. CONV/ISAM and ISMUTL
4. CONV/DMS
5. CONV/MSCOPY
6. DMPALL
7. User written programs

Below is a list of preferred methods of file transfer for different types of files.

FILE TYPES	METHODS OF FILE TRANSFER
Sequential files	SSCOPY and SSNAME
Sequential files with odd record sizes	SSCOPY and SSNAME
Source programs	SSCOPY and SSNAME
ISAM and Relative files	CONV/ISAM, SSCOPY, SSNAME and ISMUTL
DMS files	CONV/DMS

SSCOPY

SSCOPY is a utility program that runs on the B2000/3000/4000 series systems. Its major function is to copy files from tapes created by the B1000 series systems library maintenance program SYSTEM/COPY. It is a recommended method for transferring B1000 files to the B2000/3000/4000 systems media.

To transfer files, use the B1000 SYSTEM/COPY to dump files from disk to tape. Then mount the tape on a B2000/3000/4000 and run SSCOPY from the ODT to load them to disk or disk pack.

SSCOPY can optionally copy files under generated six character file names which can later be changed to names desired by the user using the MCP control instruction CHANGE or by the SSNAME utility. SSCOPY does not normally change file formats. Where such reformatting is required, other utilities must be used after a successful run of SSCOPY. SSCOPY produces a report of the run and a names file , which contains the original and new names of the files copied. The names file will be used by the SSNAME program.

SSNAME

SSNAME is a B2000/3000/4000 utility program that is designed to simplify the task of changing names or formats of a large number of files transferred to the B2000/3000/4000 series systems during the migration process. A successful run of SSCOPY is required prior to the execution of the SSNAME program. SSNAME accepts the names file created by SSCOPY as input.

SSNAME provides the user the capabilities of renaming, reblocking and changing to EDITOR format of files copied by SSCOPY.

INDEXED AND RELATIVE FILE TRANSFER UTILITIES

The transfer of ISAM and relative files is a two-part process. The first part consists of changing such files to sequential files on the B1000 series systems by executing the CONV/ISAM utility program. This part is required for COBOL74 style ISAM and relative files. However, this part is not required for COBOL68 and RPG TAG implementations of indexed files.

After changing the files to sequential files, the files must be physically moved to the B2000/3000/4000 series systems by the use of SSCOPY. The second part consists of changing the sequential files to indexed sequential or relative files by executing the ISMUTL program. ISMUTL is a utility program that runs on the B2000/3000/4000 series systems. It generates a COBOL or RPG program which will be used to build indexed sequential or relative files from the sequential data files. The generator requires the presence of any source program which will use the file being transfered on the B2000/3000/4000 systems media. At least one COBOL/RPG program which will use the indexed sequential or relative file must be translated beforehand. The generator will extract information from the program for describing the file.

CONV/MSCOPY

During the migration process, there often arises a need to transfer files from the B2000/3000/4000 series systems to the B1000 series systems for the purpose of testing. CONV/MSCOPY is a utility program that runs on the B1000 series systems which copies files from library tapes created by the B2000/3000/4000 series systems SYSTEM/COPY program to 180-byte media on the B1000 series systems. Files will normally be copied without any changes. File attributes will be carried over wherever possible, so that the programs using these files will require minimum or no modifications. CONV/MSCOPY will recognize and change the B2000/3000/4000 series systems EDITOR files to sequential B1000 CANDE compatible files.

SECTION 3
RPG

The use of the Burroughs B1000 Series RPG to Burroughs B2000/3000/4000 Series RPG Filter is recommended to assist you with this migration. The RPG Filter is a fully supported program available from your local Burroughs representative. The filter accepts B1000 RPG source code and produces B2000/3000/4000 RPG source code. The filter filters most constructs and clearly flags the remaining ones for manual changes.

This section will help you to handle B1000 RPG constructs that are not translated by the filter to B2000/3000/4000 RPG.

Refer to Section 2 for some tips on transferring your programs from the B1000 to the B2000/3000/4000 series systems.

DOLLAR SPECIFICATIONS

Dollar specifications on B1000 series systems can be divided into two classes:

1. Compiler directing.
2. File attributes.

Compiler-directing dollar specifications direct the compiler to perform specific functions such as listing the program or suppressing warning messages. File attribute dollar specifications give the compiler information about files that is not included in the normal F specifications. B1000 examples include PACKID or FAMILY.

COMPILER-DIRECTING DOLLAR SPECIFICATIONS

On the B2000/3000/4000 series systems, compiler-directing dollar specifications are referred to as Compiler Control Images (CCI). There are two types of CCIs: temporary and permanent. A temporary CCI, which has only one $ in column 6 can be made permanent by adding another $ in column 7.

The following is a list of compiler directing dollar specifications which require manual changes:

B1000	B2000/3000/4000
BAZBON	The effect is the same as having an entry of S in Column 42 of the H specifications.
LIBR	No equivalent. This option is being evaluated for future implementation. Manual changes are required.

NAMES	No equivalent.
PARMAP	No equivalent.
STACK	Not applicable. B2000/3000/4000 RPG handles stack automatically. If more stack is required, the program must be executed with additional memory using the CORE option and the INSERT statement.
XMAP	No equivalent exists but the CODE option is similar and very useful.
ZBINIT	The effect is the same as having an entry of S in Column 42 of the H specifications.

FILE ATTRIBUTE DOLLAR SPECIFICATIONS

All the B1000 file attribute dollar specifications are changed to File Attribute Specifications by the RPG Filter. Most do not require further changes. Those not handled, or those requiring additional information are documented in the following table.

The defaults for the minimum and maximum values for attribute values may not be same on both the systems. For exact values, refer to the B2000/3000/4000 RPG Reference Manual.

B1000	B2000/3000/4000
AAOPEN	No equivalent.
CASSET	This hardware is not supported on the B2000/3000/4000 series systems.
CLOSE	Normally, all files are closed at end-of-job, as specified in the RPG program cycle. To close files at end-of-file, specify an entry of S in Column 53 of the H specifications.
DRIVE	No equivalent.
DNAME	No equivalent.
FAMILY	Refer to the File Naming subsection.
OPEN	Not needed. Normally, all files

	are opened at beginning-of-job, as specified in the RPG program cycle.
REFORM	No equivalent.
REORG	Not applicable.
TAG	Not applicable. All B2000/3000/4000 ISAM files are similar to B1000 COBOL-74 IXSEQ files.

FILE NAMING

The B2000/3000/4000 FAMILYNAME file attribute is the same as the B1000 PACKID file attribute and is not related to the B1000 systems FAMILY attribute. There is no direct equivalent to the B1000 systems FAMILY.

For pack files, the B1000 FAMILY attribute should be ignored and the DISKID (PACKID) attribute should be changed to FAMILYNAME.

For tape files, the FAMILY attribute should be changed to FAMILYNAME since this stands for the tape name.

The B2000/3000/4000 file names are restricted to six characters only. In addition, it will be simpler to limit all file names to upper case letters and numbers and begin with a letter. Among the special characters, the hyphen (-) and underscore (_) cause the fewest problems.

GENERAL LANGUAGE ELEMENTS

B1000 RPG supports up to 31 digit numbers, while B2000/3000/4000 RPG supports up to 23 digit numbers.

B1000 RPG supports up to 511 character alphanumerics, while B2000/3000/4000 RPG supports up to 256 character alphanumerics.

H SPECIFICATION

There are a number of additional features available on the H specifications in B2000/3000/4000 RPG. They are explained in the B2000/3000/4000 RPG Reference Manual. The H specification difference is:

COLUMN 15

If a 1 appears in this column and the operation code DEBUG does not appear in the C specifications, the B2000/3000/4000 RPG compiler emits an error. If debugging is desired, DEBUG must appear in the C specifications.

D SPECIFICATION

B1000 RPG allows a file attribute dollar specification (only $PACKID or $DISKID) before the D specification when library files reside on disk other than the system disk. B2000/3000/4000 RPG does not allow any file attribute dollar specifications for the D specification.

F SPECIFICATION

The F specification differences are :

COLUMNS 7-14

The B1000 and B2000/3000/4000 RPG compilers do not allow file names to exceed eight characters. B1000 RPG requires that the first seven characters be unique among file names. B2000/3000/4000 RPG requires that the first six characters be unique among file names. If the file is of type indexed (an I in column 32 of the F specification), the first four characters must be unique among file names.

COLUMN 15

Card files may not be declared as combined on the B2000/3000/4000 series systems. Changes in program functions may be required in this context.

COLUMNS 20-23 AND 24-27

The default block and record lengths for disk data files is 180 bytes on the B1000 series systems, while on the B2000/3000/4000 series systems it is 100 bytes. Pack data files on both systems have the same default value of 180 bytes.

On the B1000 series systems, the record sizes and block sizes of input files are taken from the disk file header. On the B2000/3000/4000 series systems, these attributes must be specified exactly in the programs.

The B2000/3000/4000 series systems do not support 96 column card devices.

On the B2000/3000/4000 series systems, the record length must be an even number of bytes.

On the B2000/3000/4000 series systems, if the file is of the indexed type, then the calculation of block length is involved. The block size is always greater than the size required to accomodate the data portion. Refer to Section 9 of this manual for details on indexed files.

COLUMNS 29-30

B1000 RPG allows a maximum key length of 99 characters for B-indexed files. B2000/3000/4000 RPG allows a maximum key length of 29 characters or digits.

COLUMN 39

If an entry of L appears in this column, the B2000/3000/4000 RPG compiler requires an L specification.

COLUMNS 40-46

B1000 RPG accepts many device names that B2000/3000/4000 RPG does not. It will be simpler to use only generic device names. These are :

1. READER
2. PUNCH
3. PRINTER
4. TAPE
5. DISK
6. PACK
7. CONSOLE
8. DATACOM

Refer to the B2000/3000/4000 RPG Reference Manual for a complete list of valid device names.

On the B2000/3000/4000 series systems, the cevice DISK implies 100-byte disk. The device name for 180-byte disk is PACK.

COLUMN 53

B2000/3000/4000 RPG does not support the L and R options. The option L is for locking the file; the option R is for making the file input only for other programs. This option is being evaluated for future implementation.

COLUMNS 71-72

On the B1000 series systems, program switches may be initialized with the EXECUTE statement or by the MODIFY control instruction. If the switches are not initialized, console input is requested at program execution time. On the B2000/3000/4000 series systems, external indicators must be set by the INSERT clause in the EXECUTE statement. Console input is not requested after program BOJ when all indicators are reset.

E SPECIFICATION

The E specification differences are :

COLUMN 43

On the B2000/3000/4000 series systems, binary format is not allowed for pre-execution-time vectors.

COMPILE-TIME VECTORS

On the B1000 series systems, the CARDS and SOURCE files contain the RPG source statements, and the TABCRD file contains the compile-time vectors. On the B2000/3000/4000 series systems, compile-time vector data is included at the end of the source statements. Each vector must start with a VVECTOR card. Because each vector starts with a VVECTOR card, it is possible to have other than the last vector as a short vector. For more information on vectors, refer to the Extension Specifications and Vectors sections in the B2000/3000/4000 RPG Reference Manual.

B1000 RPG allows compile-time vectors to be up to 96 characters long. B2000/3000/4000 RPG limits compile-time vectors to 80 characters.

T SPECIFICATION

The T specification differences are :

COLUMNS 16-18

The B2000/3000/4000 data communications subsystem does not support the concept of a maximum number of stations per remote file. Therefore, this field is ignored.

COLUMNS 19-21

The maximum messages field has been moved to columns 71-73. This field is not used currently and must be left blank.

COLUMNS 22-27

The station number field has been moved to columns 41-47. It is also necessary to enter an S in column 40.

COLUMNS 28-33

The message length field has been moved to columns 64-70. It is also necessary to enter an S in column 63.

I SPECIFICATION

The I specification differences are :

COLUMNS 19-20

B2000/3000/4000 RPG does not support spread card format. Therefore, the indicator TR should not be specified. Change your spread card format data files to regular data files.

COLUMN 42

The B2000/3000/4000 series systems do not support hardware that can select stackers on input files. The compiler ignores any entry in this column.

COLUMN 43

B2000/3000/4000 RPG does not support binary format.

COLUMNS 61-62

B1000 RPG allows a total match field size of 256 characters, while B2000/3000/4000 RPG allows 255 characters.

C SPECIFICATION

The C specification differences are:

CHAIN

B2000/3000/4000 RPG requires the type and length specified in FACTOR 1 for indexed sequential files to be the same as the type and length of the key field for the file named in FACTOR 2.

DMKEY

On the B1000 series systems, the key condition must be specified in terms of elementary items when a group item key is involved. The B2000/3000/4000 series systems allow only the group name. Refer to Group Item Keys in Section 7 of this manual.

MOVEA

B2000/3000/4000 RPG requires either FACTOR 2 or the RESULT field to be an array.

SETLL

B2000/3000/4000 RPG requires the type and length specified in FACTOR 1 for indexed sequential files to be the same as the type and length of the key field for the file named in FACTOR 2.

ZIP

The B1000 ZIPped text may differ from the B2000/3000/4000 ZIPped text. The B2000/3000/4000 series systems requires a period (.) to terminate the control text to be zipped. Text to be zipped must be examined carefully and changed appropriately if needed.

O SPECIFICATION

The O specification differences are:

COLUMN 16

The B2000/3000/4000 series systems support hardware capable of selecting only up to four stackers. Stacker selections of 4-9 causes selection to the default stacker.

COLUMNS 32-37

The B2000/3000/4000 series systems do not support *PRINT function.

COLUMN 39

A blank-after with constants, and with the keywords UDATE, UDAY, UMONTH, UYEAR, UTIME and JDATE are not supported.

COLUMNS 40-43

The B2000/3000/4000 series systems do not support hardware that can print on cards.

COLUMN 44

B2000/3000/4000 RPG does not support binary format.

SECTION 4
SORT

The B2000/3000/4000 SORT UTILITY (SRTUTL) program is designed to sort indexed sequential, relative, or sequential files by requested keys and to generate either an ADDROUT file acceptable to RPG, or a new sequential file of complete records.

This section documents the differences between the B1000 SORT program and B2000/3000/4000 SRTUTL program. The B2000/3000/4000 series systems do not implement the B1000 SORT/UTILITY.

SRTUTL ONLY AS A PROGRAM

The B1000 SORT program can be either "compiled to library" for subsequent execution or compiled and executed immediately. The B2000/3000/4000 SRTUTL program can not be compiled to library for subsequent execution.

Refer to the B2000/3000/4000 System Software Operation Guide Volume 2 for execution instructions.

UNSUPPORTED B1000 OPTIONS

No other options except for INCLUDE/DELETE, MEMORY, PARITY DISCARD, RECORDS, TAPESORT and TAGSORT options are supported on the B2000/3000/4000 series systems. The TAGSORT option is equivalent to the ADDROUT option on the B2000/3000/4000 series systems.

All unsupported B1000 options are currently being evaluated for future implementation in the B2000/3000/4000 SORT program to eliminate inconveniences on the part of the users. Hence, the B2000/3000/4000 SORT program will be able to accept B1000 SORT syntax.

FILE STATEMENT

The B1000 SORT program allows up to 16 B1000 file input parts to be sorted. The B2000/3000/4000 SRTUTL currently allows up to 8 files to be sorted or merged.

The B2000/3000/4000 series systems require a six-character file name. The B1000 file name must be changed appropriately.

The options OPTIONAL, MULTI and VARIABLE are not supported on the B2000/3000/4000 series systems.

The hardware PAPER is not supported on the B2000/3000/4000 series systems.

The parity O, ODD, E, EVEN for TAPE files are not supported on B2000/3000/4000 series systems.

An output file of the type PRINTER is not supported on B2000/3000/4000 series systems.

DATA TYPE DIFFERENCES

The data types RSA, RSN and NC are not supported on the B2000/3000/4000 series systems.

INCLUDE AND DELETE STATEMENT

The IN option of the INCLUDE and DELETE is not supported.

When both INCLUDE and DELETE are specified in the B1000 the last specification follows, while on the B2000/3000/4000 series systems the INCLUDE is applied first.

SORT STATEMENT

The B2000/3000/4000 HOTSORT or DISKSORT statement will read a relative, indexed or sequential data file sequentially and sort records using disk for the work files.

ADDROUT FILE

The TAGSORT option is equivalent to the B2000/3000/4000 ADDROUT option. ADDROUT will select the requested keys (from the KEY statement), concatenate them, write a file of records consisting of an 8-digit releative record number plus the concatenated keys.

No changes to either the RPG or SORT program is required for migration of ADDROUT files from the B1000 to the B2000/3000/4000 series systems. However, the ADDROUT file used on the B2000/3000/4000 series systems must be created by the B2000/3000/4000 SRTUTL.

SECTION 5
COBOL74

The majority of differences between B1000 COBOL74 and B2000/3000/4000 COBOL74 are due to machine dependent aspects.

GENERAL DIFFERENCES

RESERVED WORDS

B1000 COBOL74 allows hardware names such as DISK to be used as file names although these are reserved words. B2000/3000/4000 COBOL74 is stricter, and so these names must be changed.

The list of words which are reserved in B2000/3000/4000 COBOL74 is slightly different from the corresponding B1000 COBOL74 list. Below is a list of words, that are not reserved in B1000 COBOL74, but are reserved in B2000/3000/4000 COBOL74. These words must be changed if used as data names.

ASCII	DISMISS	REMOVE
B-2800	DMSTRUCTURE	SINGLE
thru	DUMP	SPO
B-4900	GCR	SW1
BACKUP	INTERROGATE	thru
CARDS	MODIFY	SW8
CASSETTE	ODT	SYSTEMERROR
CHANNEL	PAPERTAPE	TAPES
COLUMNS	PHASE-ENCODED	THEN
DEADLOCK	PRINTER	TRACE-OFF
DISK	PUNCH	TRACE-ON
DISKPACK	READER	ZIP

HEXADECIMAL LITERALS

On the B1000 series systems, hexadecimal literals are treated as numeric and can be used in arithmetic operations.

B2000/3000/4000 COBOL74 emits a syntax error if a hexadecimal literal is used as an operand of an arithmetic operation. For example, ADD 1 @8@ GIVING A1 is invalid. A run time error occurs on the B2000/3000/4000 series systems when an operand containing undigits (@A@ thru @F@) is used in an arithmetic operation. For example, ADD 1 TO A1 will result in a run-time error if A1 is PIC 9 COMP and contains @A@.

IDENTIFICATION DIVISION

No changes are required.

ENVIRONMENT DIVISION

OBJECT-COMPUTER

STACK SIZE is used to specify the size of PERFORM stack on the B1000 series systems. On the B2000/3000/4000 series systems, it is used to specify the size of program stack. Refer to the B2000/B3000/B4000 COBOL74 Reference Manual.

INPUT-OUTPUT SECTION

FILE-CONTROL.

HARDWARE NAMES

Most device names accepted by the B1000 COBOL74 compiler are accepted by the B2000/3000/4000 COBOL74 compiler. QUEUE and REMOTE files are not available on the B2000/3000/4000 series systems. For the migration of programs using these files, refer to Sections 10 and 11 of this manual.

The device DISK on B2000/3000/4000 series systems refers to 100-byte disk device. The device name for 180 byte disk is DISKPACK.

SORT/MERGE

The use of both disk and tape for the same sort file is not allowed on the B2000/3000/4000 series systems. Change as follows:

B1000:

```
ASSIGN TO SORT [DISK [AND integer {TAPE }]]
                                  {TAPES}
```

B2000/3000/4000:

```
ASSIGN TO SORT { DISK     }
               { DISKPACK }

            or

ASSIGN TO SORT integer { TAPE  }
                       { TAPES }
```

FILE NAMES in the SELECT clause must have the first six characters unique on the B2000/3000/4000 series systems in order to use MCP file equation.

INDEXED FILE NAMES must have the first 8 characters unique on the B1000 series systems. The first 4 characters must be unique on the B2000/3000/4000 series systems.

DATA DIVISION

FILE SECTION

FD

On the B1000 series systems, if the record size and block size are not specified for input or I-O disk files by the **RECORD CONTAINS** and the **BLOCK CONTAINS** clauses, they are taken from the disk file header. On the B2000/3000/4000 series systems, these attributes must always be explicitly specified.

The **DATA RECORDS** clause is used for documentation purposes only in B1000 COBOL74. Any inconsistencies are ignored by the compiler. In B2000/3000/4000 COBOL74, the names and the number of records must match with the specified ones. Any inconsistencies are flagged as errors.

RECORD DESCRIPTION

The B1000 COBOL74 compiler does not check the type compatibility of the literal with the data name for which that value is assigned. The B2000/3000/4000 COBOL74 compiler requires that literals be type compatible with the corresponding data names. For example, the sign must be removed from a literal assigned to an unsigned data name.

WORKING-STORAGE SECTION

Refer to Record Description above.

COMMUNICATION SECTION

This feature is not available in B2000/3000/4000 COBOL74.

PROCEDURE DIVISION

CALL

B2000/3000/4000 COBOL74 requires that the called program have the same PROGRAM-ID and object program name. The object program name must be a valid file name on the B2000/3000/4000 series systems.

The statement CALL SYSTEM DUMP has no parameters. B1000 COBOL74 programs using this statement must be changed.

CLOSE

The statement **CLOSE WITH RELEASE** must be followed by a comma, semi-colon or a period.

The B1000 series systems allow a temporary file to be reopened in OUTPUT mode as in the following sequence:

```
OPEN OUTPUT DISK-FILE
CLOSE DISK-FILE
OPEN INPUT DISK-FILE
CLOSE DISK-FILE
OPEN OUTPUT DISK-FILE
```

On the B2000/3000/4000 series systems, the last OPEN must be preceded by a CLOSE PURGE or changed to an OPEN EXTEND.

COPY

External file names used with the COPY statement must be changed to valid B2000/3000/4000 file names.

B1000	B2000/3000/4000
COPY F2/F3 ON F1	COPY F3 ON F1.

MERGE

The CLOSE option for the GIVING file must be changed so that for disk files, instead of RELEASE, the PURGE option is used.

SORT

The TAG-SEARCH and TAG-KEY options are not allowed on the B2000/3000/4000 series systems and must be deleted.

The close option for the USING file must be changed so that for disk files, SAVE and RELEASE are changed to DISMISS and PURGE respectively.

UNSTRING

On the B1000 series systems, the overflow caused by the POINTER does not affect the old value of "INTO" identifier. The old value is erased under such conditions on the B2000/3000/4000 series systems.

COMPILER CONTROL IMAGES (DOLLAR OPTIONS)

The following options must be changed to their equivalent options on the B2000/3000/4000 series systems:

B1000	B2000/3000/4000
<non-numeric literal>	NEWID "literal"
LIBRDOLLAR	LIBR$
LISTDOLLAR	LIST$
MERGE	MERGE [<mfid>/] <fid> [<device>]
NEW	NEW [<mfid>/] <fid> [<device>]

The following options are not applicable on the B2000/3000/4000 series systems and must be deleted.

DEBUG ERRMESS
DEBUG TIME
LIST1
SEQ without parameters
STATISTICS
XSEQ

SECTION 6
COBOL68

It is recommended that B1000 COBOL68 programs be converted to B2000/3000/4000 COBOL74 for the following reasons:

1. COBOL74 is being enhanced. No new features are being added to B2000/3000/4000 COBOL68.

2. B2000/3000/4000 COBOL68 does not support ISAM files and DMSII. The B2000/3000/4000 COBOL74 ISAM is fully compatible with RPG.

3. All features available in B1000 COBOL68 are either available with the B2000/3000/4000 COBOL74 compiler or missing from both COBOLs on the B2000/3000/4000 series systems.

4. Most of the desirable Burroughs extensions in COBOL68 have been added to COBOL74.

It is recommended that you use the Burroughs to Burroughs COBOL filter to assist you with the migration. The B2000/3000/4000 COBOL filter is a fully supported program product. The filter will translate most constructs and flag the remaining constructs for manual changes.

This section will help you to handle B1000 COBOL68 constructs that are not translated by the filter to B2000/3000/4000 COBOL74.

Refer to Section 2 for some tips on transferring your programs from the B1000 to B2000/3000/4000 series systems.

GENERAL INFORMATION

NUMERIC LITERALS

The maximum literal size is 160 digits in B1000 COBOL68. B2000/3000/4000 COBOL74 allows 160 digits if the literal is not used in arithmetic operations. Otherwise, 98 digits are allowed.

HEXADECIMAL LITERALS

On the B1000 series systems, hexadecimal literals and data names containing undigit values (@A@ thru @F@) are treated as numeric and can be used in arithmetic operations.

B2000/3000/4000 COBOL74 emits a syntax error if a hexadecimal literal is used as an operand of an arithmetic operation. For example, ADD 1 @8@ GIVING A1 is invalid. A run time error occurs on the B2000/3000/4000 series systems when an operand containing undigits (@A@ thru @F@) is used in an arithmetic operation. For example, ADD 1 TO A1 will result in a run-time error if A1 is PIC 9 COMP and contains @A@.

HIGH VALUE, LOW VALUE

In B1000 COBOL68, high values for all data items are all display 9's (@F9@) Low values represent the lowest internal coding sequence (blanks). For a signed numeric computational field they represent +0.

In B2000/3000/4000 COBOL74, high values for all data types are all @FF@. Low values for numeric and alphanumeric data types are all@00@. Low values for numeric computation and numeric edited fields are @F0@.

SPECIAL REGISTERS

In B1000 COBOL68, the TIME format is HHMMSST, i.e., PIC 9(7) COMP. In B2000/3000/4000 COBOL74, the format is HHMMSShh, i.e., PIC 9(8) COMP.

In B2000/3000/4000 COBOL74, program switches are handled differently. Refer to PROCEDURE DIVISION in this section.

IDENTIFICATION DIVISION

The MONITOR facility can be simulated in B2000/3000/4000 COBOL74 by the USE FOR DEBUGGING declarative. Refer to the B2000/B3000/B4000 COBOL74 Reference Manual for details.

ENVIRONMENT DIVISION

INPUT-OUTPUT SECTION must not be specified unless there are files declared in the program.

All sections and paragraphs, if specified, must appear in the order given in the B2000/B3000/B4000 COBOL74 Reference Manual.

CONFIGURATION SECTION

SOURCE-COMPUTER

B2000/3000/4000 COBOL74 provides an additional feature, WITH DEBUGGING MODE. This is used for conditional compilation. Refer to the B2000/B3000/B4000 COBOL74 Reference Manual for details.

OBJECT-COMPUTER

SORT MEMORY SIZE is to be specified with the SORT verb verb B2000/3000/4000 COBOL74. It is moved to the SORT verb by the COBOL filter. Changes may be neccessary to the actual memory size.

SPECIAL-NAMES.

Mnemonic names may be specified in this paragraph for printer channels in order to translate the B1000 construct

WRITE <file> BEFORE CHANNEL <number>.

CURRENCY SIGN characters "L", "=" and "/" are not allowed in B2000/3000/4000 COBOL74 but are allowed in B1000 COBOL68. The character "I" has a special meaning in the PICTURE string in B2000/3000/4000 COBOL74.

INPUT-OUTPUT SECTION

FILE-CONTROL

This must not be an empty paragraph.

RESERVE <n> ALTERNATE AREAS is equivalent to RESERVE <n+1> AREAS. The filter reduces any higher numbers to the B2000/3000/4000 series systems maximum of 9 areas (or 8 alternate areas).

FILE-LIMIT is not supported by B2000/3000/4000 COBOL74.

HARDWARE NAMES

Most device names accepted by B1000 COBOL68 are also accepted by B2000/3000/4000 COBOL74. The device names CARD96, CASSETTE and REMOTE have no equivalents on the B2000/3000/4000 series systems. Programs using them require changes.

The COBOL filter changes all disk device names such as DISK or PACK to DISKPACK since this stands for 180 byte disk on the B2000/3000/4000 series systems. The device DISK refers to 100 byte media on the B2000/3000/4000 series systems.

FILE ATTRIBUTE OPTIONS

The word SINGLE must immediately follow a hardware name. This option has a different meaning on the B2000/3000/4000 series systems but has no adverse effect on migration.

SORT/MERGE

On B2000/3000/4000 series systems, the use of both disk and tape is not allowed for the same SORT file. Also, the number of tapes for tape sort should be specified as a constant. If DISK was specified originally, the COBOL filter changes it to DISKPACK. And if tapes were used, 3 tapes are assigned. Further changes may be required if different options are desired.

In B2000/3000/4000 COBOL74, only a file assigned to SORT or MERGE can be described with an SD in the FILE SECTION. The filter detects this condition and emits a warning. However, the word SORT or MERGE must be inserted manually in this paragraph.

FILE NAMES in the SELECT clause must have the first six characters unique on the B2000/3000/4000 series systems in order to use MCP file equation.

INDEXED FILE NAMES must have the first 8 characters unique on B1000 series systems. The first 4 characters must be unique on B2000/3000/4000 series systems.

The MULTIPLE FILE CONTAINS clause has a different syntax in B2000/3000/4000 COBOL74. However, it is used for documentation purposes only on both systems.

B1000:

```
MULTIPLE FILE { DISKPACK <diskpack-id>   }
              { TAPE     <multi-file-id> }
CONTAINS <file-name> [ POSITION <integer> ] ...
```

B2000/3000/4000:

```
MULTIPLE FILE TAPE CONTAINS
<file-name> [ POSITION <integer> ]...
```

VALUE OF FAMILYNAME can also be specified.

DATA DIVISION

FILE SECTION

FD

RECORDING MODE is changed to the attribute clause VALUE OF EXTMODE by the COBOL filter. A warning is emitted if further changes are required. The recording mode may be specified in two ways on the B2000/3000/4000 series systems.

```
CODE SET IS <alphabet-name>

or

VALUE OF EXTMODE IS { EBCDIC      }
                    { NONSTANDARD }
                    { BINARY      }
```

Refer to the B2000/3000/4000 COBOL74 Reference Manual for details on the use of these two clauses. Also related is the specification of a PROGRAM COLLATING SEQUENCE in the OBJECT-COMPUTER paragraph and the specification of an alphabet name in the SPECIAL-NAMES paragraph.

It should also be noted that all the data-items in files recorded in modes other than EBCDIC must be declared as DISPLAY and if they are signed numeric then

SIGN IS SEPARATE must be specified.

For the migration of programs using QUEUE files and REMOTE files, refer to Sections 10 and 11 of this manual.

RECORD CONTAINS and BLOCK CONTAINS

In B1000 COBOL68, the RECORD CONTAINS clause is for documentation, and the length computed from the record descriptions is used. In B2000/3000/4000 COBOL74, a syntax error is given if the record length computed from the record descriptions does not match the value specified in the RECORD CONTAINS clause. This clause must be either removed or corrected to have the correct record length.

On the B1000 series systems, if the record size and block size are not specified for input or I-O disk files, they are taken from the disk file header. These attributes must always be explicitly specified on the B2000/3000/4000 series systems.

The DATA RECORDS clause is used for documentation purpose only in B1000 COBOL68. Any inconsistencies are ignored by the compiler. In B2000/3000/4000 COBOL74, the names and number of records must match with the specified ones. Any inconsistencies are flagged as errors.

RECORD DESCRIPTION

USAGE ASCII is not allowed by B2000/3000/4000 COBOL74.

In B1000 COBOL68, the OCCURS DEPENDING ON clause is used for documentation purposes only. B2000/3000/4000 COBOL74 requires this clause with variable length tables and range checking is performed.

PICTURE

In B1000 COBOL68, if J is not the left-most character in a picture string, it reinitiates zero suppression. This feature is not available in B2000/3000/4000 COBOL74.

B1000 COBOL68 allows up to 160 positions for decimal scaling whereas B2000/3000/4000 COBOL74 allows up to 98 positions only. Negative scaling is not allowed by B2000/3000/4000 COBOL74.

VALUE

The B1000 COBOL68 compiler does not check the type compatibility of the literal with the data name for which that value is assigned. The B2000/3000/4000 COBOL74 compiler requires that literals be type compatible with the corresponding data names. The filter handles most such cases. For example, the sign is removed from a literal assigned to an unsigned data name. Changes are required in other cases such as a numeric literal assigned to a numeric edited data item.

COPY

External file names used with the COPY statement must be changed to valid B2000/3000/4000 file names.

B1000	B2000/3000/4000
COPY F1/F2/F3	COPY F3 ON F1.

WORKING-STORAGE SECTION

Refer to Record Description above.

PROCEDURE DIVISION

Program switches

In B1000 COBOL68, program switches are special registers which can be set to any value either by the program or externally. They can be used like data-names. In B2000/3000/4000 COBOL74, program switches can only be tested by the program. Their values can be set only externally and only "on" and "off" values are allowed. Refer to SPECIAL-NAMES in the B2000/B3000/B4000 COBOL74 Reference Manual for the syntax relating to the use of switches. There are, however, system intrinsics available to simulate B1000 types of program switches. To use these intrinsics, eight data-items must declared in WORKING-STORAGE SECTION as below. They may be used as the <identifier> in the CALLs shown subsequently.

```
       01  PROGRAM-SWITCHES.
           05  SW8-XX  PIC 9 COMP.
           05  SW1-XX  PIC 9 COMP.

           :    :       :     :
           :    :       :     :

           05  SW7-XX  PIC 9 COMP.

CALL "INTRINSIC GETSWITCH" GIVING <identifier>
CALL "INTRINSIC SETSWITCH" USING <identifier>
```

With the &SWITCH option, the COBOL filter will insert the WORKING-STORAGE declarations and change SW1 thru SW8 to SW1-XX thru SW8-XX. The CALL to the intrinsics must be inserted manually at the appropriate places.

Comparisons

B2000/3000/4000 COBOL74 requires that the operands of comparisons are compatible. For example, a computational numeric data item cannot be compared with SPACES.

ELSE Clauses

B1000 COBOL68 allows ELSE clauses to be used in a number of situations. The ELSE clause can follow the ON SIZE ERROR clause, the INVALID KEY clause, the AT END clause, the AT END-OF-PAGE clause, and the IF verb. In B2000/3000/4000 COBOL74, the ELSE clause can be used only with IF and not with any of the other clauses. Also, while B1000 COBOL68 permits any statements to follow these conditional clauses, only imperative statements are allowed by B2000/3000/4000 COBOL74. Portions of programs may have to be recoded to handle this difference.

B1000:

```
READ <file>
    AT END <statement-1>
    ELSE   <statement-2>.
```

B2000/3000/4000:

```
READ <file>
AT END <imperative-statement>.
```

CLOSE

The meaning of LOCK in B2000/3000/4000 COBOL74 is different from that in B1000 COBOL68. A file, if CLOSEd with LOCK, cannot be reopened.

The word RELEASE must be followed by a comma, semi-colon, or period unless the next statement is a RELEASE statement.

The B1000 series systems allow a temporary file to be reopened in the OUTPUT mode as in the following sequence:

```
OPEN OUTPUT DISK-FILE
CLOSE DISK-FILE
OPEN INPUT DISK-FILE
CLOSE DISK-FILE
OPEN OUTPUT DISK-FILE
```

On the B2000/3000/4000 series systems, the last OPEN must be preceded by a CLOSE PURGE or changed to an OPEN EXTEND.

COPY

External file names used with the COPY statement must be changed to valid B2000/3000/4000 file names.

B1000	B2000/3000/4000
COPY F1/F2/F3	COPY F3 ON F1.

DIVIDE

Use of the B1000 COBOL68 option MOD must be changed as follows:

B1000 :

1. DIVIDE MOD <identifier-1> INTO <identifier-2>

2. DIVIDE MOD <identifier-1> BY <identifier-2>
   GIVING <identifier-3>

B2000/3000/4000:

1. DIVIDE <identifier-2> BY <identifier-1>
   GIVING <temp> REMAINDER <identifier-2>

2. DIVIDE <identifier-1> BY <identifier-2>
   GIVING <temp> REMAINDER <identifier-3>

DUMP

DUMP is not allowed in B2000/3000/4000 COBOL74. DEBUG can be used instead. Refer to USE FOR DEBUGGING in the B2000/B3000/B4000 COBOL74 Reference Manual.

MERGE

In B1000 COBOL68, the WITH <file-name> option can be used to define a collating sequence. In B2000/3000/4000 COBOL74, EBCDIC and ASCII are the pre-defined collating sequences. Other collating sequences can be defined in SPECIAL-NAMES.

MOVE

If the sending field is a literal (including figurative constants), B2000/3000/4000 COBOL74 requires that it must be type compatible as the receiving field. For example, SPACES cannot be moved to a numeric data item.

OPEN

The B1000 COBOL68 option O-I is not available in B2000/3000/4000 COBOL74. A program using this option must be recoded by the use of OPEN OUTPUT, CLOSE WITH DISMISS and OPEN INPUT or I-O at appropriate locations.

The B1000 options, PUNCH, PRINT128, INTERPRET and STACKERS relating to the use of 80 or 96 column data recorders are not supported by the B2000/3000/4000 series systems.

READ

In B1000 COBOL68, use of the AT END clause or the INVALID KEY clause is not mandatory. B2000/3000/4000 COBOL74 requires the AT END clause for sequential access and the INVALID KEY clause for random access unless a USE procedure has been specified for the file.

The other aspects of the AT END clause have been mentioned earlier in this section.

RETURN

In B1000 COBOL68, the AT END clause is optional. B2000/3000/4000 COBOL74 requires the AT END clause to be specified. Rules mentioned earlier in this section must be followed.

REWRITE

The INVALID KEY clause must be specified in B2000/3000/4000 COBOL74 unless a USE procedure has been coded for the file. As with the AT END clause, only imperative statements may follow the INVALID KEY clause, and ELSE is not allowed.

SEEK

The B1000 series systems allow positioning of a file at a particular record using an alphanumeric key which is changed into the ordinal number of the record by specifying

WITH KEY CONVERSION

On the B2000/3000/4000 series systems, the key conversion must be performed explicitly before the SEEK.

SORT

The WITH <file-name> clause is used to establish a user-specified collating sequence in B1000 COBOL68. In B2000/3000/4000 COBOL74 the pre-defined collating sequences are ASCII and EBCDIC. Other types of collating sequence can be defined in SPECIAL-NAMES.

The RESTART option in B1000 COBOL68 can be converted to the BREAKOUT option in B2000/3000/4000 COBOL74. Since this is not an exact equivalent, some changes in the program logic may be necessary. Refer to the B2000/B3000/B4000 COBOL74 Reference Manual for details.

START

As with READ, INVALID KEY must be specified in B2000/3000/4000 COBOL74 if a USE procedure is not specified for the file. The INVALID KEY can be followed by imperative satements only and ELSE is not allowed.

USE

USE FOR KEY CONVERSION is not allowed in B2000/3000/4000 COBOL74. This procedure must be moved out of DECLARATIVES and must be explicitly performed.

USE FOR Q-EMPTY or Q-FULL is invalid. Refer to Section 10 for the migration of QUEUE files.

WAIT

In B1000 COBOL68, the waiting time is specified in tenth of seconds. In B2000/3000/4000 COBOL74 it is specified in seconds.

WRITE

In B2000/3000/4000 COBOL74, the CHANNEL clause is not allowed. If channels other than channel 1 are used, they must be equated to mnemonic names in the SPECIAL-NAMES paragraph and these names should replace the CHANNEL clause.

In B2000/3000/4000 COBOL74, only imperative statements are allowed following the AT END-OF-PAGE or AT EOP clause. The ELSE clause must not be specified.

In B2000/3000/4000 COBOL74, the INVALID KEY clause is required for random writes if there is no appropriate USE procedure for this file. This clause can be followed by imperative statements only and an ELSE branch cannot be specified.

COMPILER CONTROL IMAGES (DOLLAR OPTIONS)

The syntax for specifying compiler control images differs between the two systems.

B1000	B2000/3000/4000
$ <option>	$ SET <option>
$ NO <option>	$ RESET <option>

The following options must be changed to their equivalent options on the B2000/3000/4000 series systems:

B1000	B2000/3000/4000
<non-numeric literal>	NEWID "literal"
ANSI	FEDLEVEL = 4
CONTROL	LIST$
MERGE	MERGE [<mfid>/] <fid> [<device>]
NEW	NEW [<mfid>/] <fid> [<device>]
SPEC	SUMMARY
SUPPRESS	WARNSUPR
REF	DEBUG

The following options are not applicable on the B2000/3000/4000 series systems and must be deleted:

CARD
NO DEBUG
NOCOP
RPGTAGS
SINGLE
STACK <number>

The XREF dollar option may be used to get a cross-reference listing.

SECTION 7
DASDL AND DMSII

B1000 DMSII is basically a subset of B2000/3000/4000 DMSII. This section provides the changes necessary to a DASDL source deck to make it acceptable to the B2000/3000/4000 DASDL compiler.

ADDITIONAL FEATURES

The following is a list of significant features available in B2000/3000/4000 DMSII that are not available in B1000 DMSII:

1. Up to 990 structures
2. STATISTICS option in the OPTIONS statement
3. DIRECT and RANDOM data sets
4. Embedded standard data sets with more than one spanning set
5. On-line dumps
6. Menu-driven integratec DMSII utilities
7. ROLLBACK recovery
8. CHECKSUM and BUFFERS attributes in physical attributes
9. Arithmetic expressions in WHERE conditions
10. BOOLEAN and FIELD data types
11. Duplicate Audit
12. On-line Reconstruct

COMPILER CONTROL IMAGES (DOLLAR OPTIONS)

The syntax for compiler control images differs as follows:

B1000	B2000/3000/4000
$ <option>	$ SET <option>
$ NO <option>	$ RESET <option>

The B1000 compiler options and their B2000/3000/4000 equivalents are listed below. Options not listed are the same on both systems.

B1000	B2000/3000/4000
COBOL	CHECKCOBOL

COBOLIB	Not applicable. The compilers on the B2000/3000/4000 series systems read the description (dictionary) file directly. This eliminates the need for library files.
CONVERT	UPDATE
DEBUG	Not available.
FILE	Most of this information is always produced.
INCLUDE	Not available.
INCLNEW	Not applicable.
INITIALIZE	Structures are initialized if UPDATE or REORGANIZE is not specified.
LISTINCL	Not available.
MERGE	MERGE "<file id>" <device> <file id> is NEWSOU by default. <device> is DISK by default.
NEW	NEW "<file id>" <device> <file id> is NEWSOU by default. <device> is DISK by default.
RPG, RPGII	CHECKPRG
RPGLIB	Not applicable. The compilers on the B2000/3000/4000 series systems read the description (dictionary) file directly. This eliminates the need for library files.
SEQUENCE or SEQ (Default inc are 1000).	SEQUENCE or SEQ (Default base and inc are 10).
SINGLE	Not needed. Set by default.
SOURCE	Not available.
SOURCEONLY	Not applicable.
STRUCTURE	Structure information is always produced.

STANDARD	Not available.
SUPPRESS	WARNSUPR
TABLESIZE	Not available.
TAPE	Not available.
VERSIONCHECK	Not available. Version checking cannot be suppressed.

OPTIONS

AUDIT

If you have AUDIT RESET, remove the RESTART DATA SET. B2000/3000/4000 DMSII does not support the SM AUDIT SET/RESET command.

KEYCOMPARE

KEYCOMPARE is reset by default by B1000 DMSII. It is set by default by B2000/3000/4000 DMSII.

PARAMETERS

SYNC POINT

The default is 10 transactions in B1000 DMSII and 5 transactions in B2000/3000/4000 DMSII.

CONTROL POINT

The default value is 5 sync points in B1000 DMSII and 10 sync points in B2000/3000/4000 DMSII.

In view of above defaults, you may have to make the necessary changes in the parameter specifications.

MAXWAIT

This option is not supported in B2000/3000/4000 DMSII and must be deleted.

AUDIT TRAIL

BLOCKSIZE

Examine the BLOCKSIZE specification. If only an integer is specified, then the word BYTES must be added after the integer.

KIND

To assign the audit trail to disk or pack, refer to <file assignment specification> in the B2000/3000/4000 DMSII Reference Manual. The audit trail cannot be assigned to tape in B2000/3000/4000 DMSII.

SECURITYTYPE and SECURITYUSE

Security options are not available in B2000/3000/4000 DMSII and must be deleted.

DATA SETS

ORDERED EMBEDDED DATA SETS

Ordered embedded data sets are not supported in B2000/3000/4000 DMSII. Convert the ordered embedded data sets to standard embedded sets. The associated ACCESS TO must be changed to SET OF.

This change means that the one structure and file on the B1000 series systems become two structures and files on the B2000/3000/4000 series systems. They are, the data set and the data set's set. Host language programs that access orcered data sets directly should be changed to FIND the data set via the set.

VARIABLE FORMAT RECORDS

Variable format records are not supported in B2000/3000/4000 DMSII. Change all such record descriptions to fixed format records by retaining the largest length format. All other formats must be coded in the host language programs.

INITIALVALUE

INITIALVALUE = must be changed to INITIALVALUE IS.

SETS AND AUTOMATIC SUBSETS

GROUP ITEM KEYS

B1000 DMSII allows group items as keys, but always requires the RPG or COBOL program to specify each of the elementary items in all selection expressions. B2000/3000/4000 DMSII requires the host language to specify the group item in the selection expression. There are two ways to solve this problem:

1. Change the DASDL to specify all the elementary items as keys.

2. Change the COBOL or RPG programs that use the group key.

Leaving the key as a group item reduces the usefulness of the general selection expression. Further, leaving the key as a group item and changing the programs may introduce semantic differences in group and elementary compares. Accordingly, it is recommended that you change the

DASDL, not the programs.

INDEX RANDOM ACCESS

INDEX RANDOM access is not available in B2000/3000/4000 DMSII. It must be changed to INDEX SEQUENTIAL.

PHYSICAL ATTRIBUTES

SPLITFACTOR must be changed to LOADFACTOR.

As INDEX RANDOM is not supported in B2000/3000/4000 DMSII, MODULUS is irrelevant and must be deleted.

In the case of list structures, the BLOCKSIZE attribute in B1000 DMSII are specified in TABLES. Change TABLES to the corresponding BYTES or ENTRIES after calculating the number of BYTES or ENTRIES.

TABLESIZE is specified for list attributes in B1000 as the number of entries that fit in one disk sector. This attribute is not supported in B2000/3000/4000 DMSII and must be deleted.

The security attributes SECURITYTYPE, SECURITYUSE, and SECURITYGUARD are not supported in B2000/3000/4000 DMSII and must be deleted.

REMAPS

RESTART DATA SET

The restart data set cannot be remapped in B2000/3000/4000 DASDL.

REMAP REGROUPING

Restructuring of the data in the remap or <remap regrouping> is not allowed in B2000/3000/4000 DMSII. All such regroupings have to be grouped according to the sequence in which they are described in the original structure. For example:

B1000:

```
EMPLOYEE DATA SET
          (NAME GROUP ( FIRSTNAME    ALPHA(10);
                        LASTNAME     ALPHA(10) );
           TITLE        ALPHA(6) ), MAXRECORDS = 1000;

RMEMP  REMAPS EMPLOYEE
          (RMNAME GROUP (LASTNAME  ALPHA(10);
                         FIRSTNAME ALPHA(10))
           RMTITLE = TITLE HIDDEN ) SELECT (RMTITLE NEQ " ");
```

The remap must be changed as follows:

```
B2000/3000/4000:
RMEMP REMAPS EMPLOYEE
      (RMNAME = NAME GROUP (FIRSTNAME  ALPHA(10);
                            LASTNAME   ALPHA(10));
       RMTITLE = TITLE HIDDEN ) SELECT (RMTITLE NEQ " ");
```

REMAP SUBSETS

The name of the object data set must not be specified in B2000/3000/4000 DASDL. Change as follows:

```
B1000           : R-SUBSET = A-SUBSET OF A-DATA-SET
B2000/3000/4000 : R-SUBSET = A-SUBSET
```

INITIALVALUE

INITIALVALUE must be changed to INITIALVALUE IS.

LOGICAL DATA BASES

SECURITYGUARD can be specified for the entire data base on B1000 DMSII. This feature is not supported in B2000/3000/4000 DMSII and must be deleted.

AUDIT AND RECOVERY

Recovery can be initiated by the RC command on B1000 series systems and the IR command on B2000/3000/4000 series systems.

B1000 CLEAR/START recovery is equivalent to B2000/3000/4000 HALT/LOAD recovery.

UPDATE AND REORGANIZATION

On the B1000 series systems, reorganization is initiated by executing the DMS/REORG.READ program which zip executes DMS/REORG.WRIT. On the B2000/3000/4000 series systems, it is initiated by the IR keyboard input command.

SECTION 8
DMSII HOST LANGUAGES

DMSII HOST LANGUAGE SYNTAX

B1000 DMSII host syntax is basically a subset of the B2000/3000/4000 DMSII host interface. A few differences are described below:

The defaults for AUDIT at BEGIN-TRANSACTION and END-TRANSACTION are different on the two systems. On B1000, the defaults are BEGIN-TRANSACTION AUDIT and END-TRANSACTION NO-AUDIT. On the B2000/3000/4000 series systems, the defaults are BEGIN-TRANSACTION NO-AUDIT and END-TRANSACTION AUDIT.

In B2000/3000/4000 DMSII, standard embedded data sets can be accessed only via a set or a subset. FIND or MODIFY (LOCK) by physical order is not allowed. Program source modification to include the set name which corresponds to the B1000 access must be made.

Variable format data sets are not supported by B2000/3000/4000 DMSII. This difference must be resolved while transfering the DASDL and the data base. Due to this, the CREATE and RECREATE verbs must not specify the record type. The COBOL REDEFINES clause can be used in the programs using the variable format data set. Thus, many of the original data names would be preserved and the program would normally require few other changes. RPG programs must not specify the record type in Columns 45-70 of the record line on the Output Specifications.

In B1000 DMSII, selection expressions in a random FIND or MODIFY (LOCK) via a set ordered by a group item must contain the elementary items, and not the group item itself. On the other hand, B2000/3000/4000 DMSII requires the group name to be used. Use of the elementary items will result in a syntax error. For the DASDL changes required to avoid this problem, refer to Group Item Keys in Section 7 of this manual.

DMSII HOST LANGUAGE SEMANTICS

DMSII Exceptions

The few semantic differences between the two systems lie mostly in the area of DMSII exception handling. In many cases, these differences do not require program modification because of the following reasons.

Most programs do not have elaborate logic to handle every kind of exception. Some exceptions are handled by program logic, and the action in the remaining cases is usually to print or display that an exception occurred and report the exception. In these cases, even if the exception were different or returned under different conditions, program modification will not be required.

In B1000 DMSII, the ABORT exception is not returned at END-TRANSACTION unless END-TRANSACTION SYNC was specified. B2000/3000/4000 DMSII can return an ABORT at END-TRANSACTION. Programs not equipped to handle such an exception may require modification.

In B1000 DMSII, NORECORD is returned only when an operation related to an embedded structure has no current parent record. In most cases, this is a programming error. In B2000/3000/4000 DMSII, NORECORD is returned in some additional situations. B2000/3000/4000 DMSII will return a NORECORD exception whenever there is no current parent as well as when there is no current record. Operations such as STORE without any record (B1000 DMSII returns NOTLOCKED) will get a NORECORD exception. Situations where B2000/3000/4000 DMSII returns a NORECORD exception normally arise due to programming errors.

Other DMSII Exceptions

Below is a list of other differences which are of little consequence in most programs:

1. B1000 DMSII returns an AUDITERROR exception when a CLOSE is attempted in transaction state. B2000/3000/4000 DMSII returns CLOSEERROR.

2. B1000 DMSII returns READONLY if a program opens a database for INQUIRY and attempts a BEGIN-TRANSACTION or END-TRANSACTION. B2000/3000/4000 DMSII returns AUDITERROR.

3. OPENERROR is returned for any DMSII operation except CLOSE by B1000 DMSII if the database is not opened. On the B2000/3000/4000 series systems, the program is DS-ed.

4. In B1000 DMSII, a syncpoint is not forced at CLOSE unless the program passes through a transaction state. In B2000/3000/4000 DMSII, a syncpoint is always forced at CLOSE.

5. CLOSE frees all records and waits for syncpoint on the B1000 series systems. On the B2000/3000/4000 series systems, records are not freed before the syncpoint.

6. If an INSERT or REMOVE is attempted outside transaction state and without having a current parent record, NORECORD is returned by B1000 DMSII. B2000/3000/4000 DMSII returns AUDITERROR.

Partial Key Search of COBOL 68 Host.

On B1000 COBOL68, the current path pointer is updated when a FIND AT KEY or a FIND NEXT AT KEY (or MODIFY) is done via an indexed sequential SET or SUBSET. This takes place whether the operation was successful or not. Thus a position is established in the index and subsequent FIND NEXT operations can retrieve records from this position onwards.

In B2000/3000/4000 COBOL74, the dollar option SAVECP must be set in the COBOL program in order for DMSII to save the current path pointer on an unsuccessful FIND, LOCK or DELETE via an indexed sequential set or subset. If SAVECP is not set when the data base is opened, the current path pointer is left untouched after unsuccessful accesses.

SECTION 9
ISAM AND RELATIVE FILES

The B1000 ISAM files are different from the B2000/3000/4000 ISAM files. The RELATIVE files on the two systems are the same. This section provides a detailed explanation of the differences between the B1000 and B2000/3000/4000 ISAM files, and some relevant information on RELATIVE files.

B1000 ISAM FILES

B1000 has three types of ISAM files:

1. B style ISAM files.
2. TAG style ISAM files.
3. COBOL74-style ISAM files.

B1000 programs using any of these implementations can migrate to B2000/3000/4000 ISAM.

ISAM FILE TRANSFER

DATA TRANSFER

Because of format differences, these data files cannot be transferred from the B1000 to B2000/3000/4000 series systems using the SSCOPY utility program. To transfer these files the following action should be taken.

B STYLE ISAM FILES

B style ISAM files are available only in RPG. Use SYSTEM/COPY and SSCOPY to transfer the file. Then use ISMUTL to generate a B2000/3000/4000 program to read the data file and create an ISAM file.

TAG STYLE ISAM FILES

TAG style ISAM files are available in RPG and COBOL68. Use SYSTEM/COPY and SSCOPY to transfer the data portion of the file. The key files need not be transferred. Then use ISMUTL to generate a B2000/3000/4000 program to read the data file and create an ISAM file.

COBOL74 STYLE ISAM FILES

COBOL74-style ISAM files are available in RPG and COBOL74. Use CONV/ISAM to read the ISAM file and create a sequential data file on the B1000 series system. Use SYSTEM/COPY and SSCOPY to transfer the file to the B2000/3000/4000 series systems. Then use ISMUTL to generate a B2000/3000/4000 program to read the data file and create an ISAM file.

B2000/3000/4000 ISAM FILE

The ISAM files on the B2000/3000/4000 series systems are multi-keyed. For purposes of adding, updating and deleting records in a file, each record is identified solely by the value of its prime record key. This value must therefore, be unique and cannot be changed when updating the record. (The value of an alternate record key can be non-unique.)

An indexed file can be meaningfully accessed only if it is declared to be of indexed organization. However, for RECOVERY purposes it can be accessed as a RELATIVE file in COBOL and a SEQUENTIAL file in RPG.

An indexed file declared in a program is really a group of files; a data file and one or more key files. The data files and key files are not related outside of the code file. (It is the user's responsibility to specify all files in dumps, loads and so on.)

If an ISAM file is to be updated, all the alternate keys must be declared. Any key declared in the program accessing the ISAM file must match the key in the declaration of the file when the file was created.

INDEXED FILE FORMATS

Each indexed file declared in a program results in multiple physical files for data and key information. The data file name is the same as the name of the file in the file declaration. The name of each key file is composed of the first 4 characters of the data file name plus the two digit key number.

An indexed data file is a single physical file of data and control information. The first block in the file is composed entirely of file control information and is called the CONTROL BLOCK. The second through the last blocks are the DATA BLOCKS. They differ from the conventional data blocks in that they have block control information appended after the data records.

An indexed key file is similarly structured as the indexed data file. Conceptually, it can be described as an inverted tree-like structure with different levels within it.

INDEXED FILE RECOVERY

When creating a file, for all files opened output, should the system fail or the program get DS'ed, that file and any records written to the file will be lost. To prevent the file from being lost, you can create a permanent disk file by opening the file output, closing the file SAVE, and then reopening the file I-O before writing to it. This can also be achieved by using the file PROTECTION = ABNORMALSAVE.

Whenever a program that has an indexed file open I-O fails to CLOSE, that file must be recovered. An indexed file is recovered by reading up the file records from the data file of the corrupted indexed file using

the RELATIVE crganization and writing them out tc a new indexed file using the INDEXED Organization. At worst the data block being modified in memory at the time of the failure will be lost and the changes to it will not be reflected in the recovered file.

In order to insure that the indexed files are recoverable, user's should check their compile listings for the warnings "BLOCKSIZE PADDED - FILE NOT RECOVERABLE." If encountered, the blocksize must be increased to at least the control block size.

A RECOVERY program source template is provided to assist users in recovering their indexed files. The user files in the appropriate file declarations, compiles the COBOL source; and then executes the compiled program to recover the corrupted indexed file.

PERFORMANCE CONSIDERATIONS

There are certain things that a user should pay attention to in order to insure good performance on his ISAM file.

The first is the number of keys declared for a file. Each file update requires that all the key files be updated (except REWRITES that do not change the keys) and the number of keys is directly proportional to the time it takes to perform an update.

Next is the data file blocking factor which is specified in the file declaration. Good random performance (other than WRITES) is achieved by using a small blocking factor. Good sequential performance (and WRITES) is achieved by using a large blocking factor. If both are necessary than an intermediate blocking factor determined by trial and error is called for.

The size of the control block is 252 digits plus 100 digits for each alternate key declared for the file. Because it must fit into a single data block, this is the minimum block-size the data-file can have.

Key length and key type also affect the performance by choosing the minimum key-length, also by using numeric rather than alpha keys if possible (RPG users only), and by avoiding using signed keys if possible (RPG users only) high performance can be achieved.

Allocating sufficient key buffers can reduce the overall number of key file I-O's significantly. The default is three, but this number is usually not enough and can cause a performance bottleneck.

PROGRAM CONVERSION

COBOL74 PROGRAMS

There are no changes converting COBOL74 programs.

COBOL68 PROGRAMS

Use the COFLTR filter to migrate your COBOL68 programs.

RPG PROGRAMS

The following describes some of the necessary changes to migrate your RPG files.

1. For ISAM files there are no changes.

2. For TAG files there are no changes, unless you are creating your own TAG file. If you are, you should make your ISAM file multi-keyed. That way all of the tags are available all of the time and you will not have to create them.

3. For B files, since the file is always ordered on the key, some programs may access the file sequentially knowing that the records are in order. These programs will require a change to describe the file as indexed. The same holds for programs creating the file.

RELATIVE FILES

The RELATIVE files on the two systems are the same and they are available in RPG and COBOL74.

DATA TRANSFER

Because of format differences, these data files cannot be transferred from the B1000 to B2000/3000/4000 series systems using the SSCOPY utility program.

To transfer these files the following action should be taken.

Use CONV/ISAM to read the RELATIVE file and create a sequential data file.

Use SYSTEM/COPY and SSCOPY to transfer the file to the B2000/3000/4000. Then use ISMUTL to generate a B2000/3000/4000 program to read the data file and create a RELATIVE file.

B2000/3000/4000 RELATIVE FILES

A relative file can be meaningfully accessed only if it is declared to be of RELATIVE organization.

RELATIVE FILE FORMATS

The first block in the file is composed entirely of file control information and is called the CONTROL BLOCK. The second through last blocks are the DATA BLOCKS. They differ from the conventional data blocks that they have block control information appended after the data records.

RELATIVE FILE RECOVERY

File recovery is necessary whenever a program that is WRITING records to a RELATIVE file beyond the previous EOF fails to close the file successfully.

Relative file recovery is performed automatically by the next program that OPENs the file. A message is displayed indicating that the recovery routine is being performed; and the program finds the true EOF and updates the control block EOF pointer.

PERFORMANCE CONSIDERATIONS

Changing the block size of the RELATIVE file can cause considerable change in performance. Remember that the block size can be changed only before the file is created.

The size of the CONTROL BLOCK is 112 digits (56 bytes). Because it must fit into a single data block, this is the minimum block size the RELATIVE file can have.

PROGRAM CONVERSION

COBOL74 PROGRAMS

There are no changes converting COBOL74 programs.

RPG PROGRAMS

There are no changes converting RPG programs.

SECTION 10
QUEUE AND PORT FILES

The B2000/3000/4000 series systems equivalent to queue files is port files. Version two of this manual will discuss the migration of B1000 queue and port file constructs to B2000/3000/4000 port file constructs.

SECTION 11
MIGRATION OF REMOTE FILES

Version two of this manual will discuss the migration of programs using remote files.

SECTION 12
MIGRATION OF READER SORTER FILES

The READER-SORTER interface in B2000/3000/4000 COBOL74 is implemented by INTRINSICS rather than file constructs as in B1000 COBOL68. Version two of this manual will discuss the migration of programs using READER-SORTER files.

SECTION 13
FILE ATRIBUTES

Certain file attributes differ between B1000 and B2000/3000/4000 Systems.

Version two of this manual will contain a detailed explanation of the migration differences between the B1000 and B2000/3000/4000 file attributes.

SECTION 14
GEMCOS

Version two of this manual will contain a detailed explanation of the migration differences between B1000 and B2000/3000/4000 GEMCOS.

SECTION 15
REPORTER

All REPORTER products run on both B1000 and B2000/3000/4000 series systems. There are a few considerations to keep in mind, none of which are major stumbling blocks.

On the B1000 series systems, an external file name contains a maximum of three levels, each of which may consists of 10 characters or less. On the B2000/3000/4000 series systems, an external file name can contain one or two levels, each of which may consists of 6 characters or less. Changes will be required in this case.

If the database name, structures or data names were changed while converting the DASDL source from B1000 to B2000/3000/4000 DMSII, corresponding changes will have to be made in the VOCAL, REPORTER and On-line REPORTER descriptions.

The B2000/3000/4000 series systems do not support 96-column card devices. Only 80-column card input will be accepted by VOCAL. The B2000/3000/4000 COBOL source file input must be in 80 character records, blocked between 1 and 9.

Certain differences exist between B1000 and B2000/3000/4000 On-line REPORTER which are inherent to each machine. These functional differences are discussed in Appendices A and B of the On-line REPORTER III User's Manual.

The use of WFL is recommended for audit-reporter execution. The migration to WFL is a simple process. The same workflow source can be used for all audit-reporter runs by changing the external filenames which must be unique to each run.

The use of WFL provides the freedom to specify task attributes, control attributes, control the files and so on.

MANUALS	FORM
On-line REPORTER User's Manual	1110228
Vocabulary Language (VOCAL) User's Manual	1097128
AUDIT-REPORTER Language User's Manual	1096831
REPORTER II and REPORTER II (Advanced) User's Manual	1100393
On-line REPORTER III User's Manual	1149937
REPORTER III Vocabulary Language (VOCAL)	1149861

User's Manual

REPORTER III Report Language User's Manual 1131836

The same manuals are used for both B1000 and B2000/3000/4000 series systems.

SECTION 16
ODESY

The migration from B1000 to B2000/B3000/B4000 ODESY is a simple process and should cause no interruption to the data entry operation. The data entry portion of ODESY is operationally identical, thereby requiring no retraining of data entry operators. Screen formats created on the B1000 series systems can be easily transferred to the B2000/B3000/B4000 series systems by the use of the DUMP/LOAD function of the Format Maintenance program. There are a few considerations to keep in mind, none of which are major stumbling blocks.

If user programs are used to interface with the Data Entry program in ODESY, some changes to these programs will be necessary. The MCS header will be different between systems and will require program changes. The ODESY header is also formatted differently but contains the same information.

There are a few formatting features available in B1000 ODESY that are not supported by B2000/B3000/B4000 ODESY. Fortunately, these features are extraordinary and are not included on the average screen format. There is no option in B2000/B3000/B4000 ODESY to have DISPLAY formats and associated TOTAL fields which allows displaying of batch totals on a special screen format. B1000 ODESY allows the default sign of a signed field to be negative. This is not supported by B2000/B3000/B4000 ODESY. Specific fields can be defined as NO VERIFY using B1000 ODESY. This feature is also not supported by B2000/B3000/B4000 ODESY.

While screen formats can be carried across systems, there is no provision to convert data in the active TANK file. Batches in progress must be completed on the B1000 series systems so that a new (empty) TANK file can be created on the B2000/B3000/B4000 series systems. The SYSTEM file must also be recreated on the B2000/B3000/B4000 series system. To bring across the ODESY screen formats, install the current B2000/B3000/B4000 ODESY level 2.30. The required level of the B1000 ODESY is 2.20.

Formats are transferred by dumping them to a backup file on the B1000 series systems using the /DUMP command of the Format Maintenance program (ODESY/FORMAINT). This file is then transferred to the B2000/B3000/B4000 series systems by tape and inserted into the B2000/B3000/B4000 ODESY system with the /LOAD command to the Format Maintenance program (ODYFOR).

While B1000 and B2000/B3000/B4000 ODESY are essentially identical, there are some additional features available with B2000/B3000/B4000 ODESY. A partial list follows:

1. B2000/B3000/B4000 ODESY has an ODT input program which allows input into the Format Maintanence program (ODYFOR) and/or the

Edit program (ODYEDT). This program (ODYSPO), gives the ODT operator the ability to bring the ODESY system up or down, send messages to the data entry operators or schedule batches.

2. A nice feature of B2000/B3000/B4000 ODESY is journal headings where up to three lines of title and/or column headings can be specified for output to a printed report (journal).

3. B2000/B3000/B4000 ODESY has an option to change the signal character from a "/" to another character. This is sometimes necessary when the MCS under which ODESY is executing has the "/" character reserved for another purpose.

4. B2000/B3000/B4000 ODESY also allows checking of crossfoot totals in the data entry program (ODYEDT) by means of the /STATUS XFOOT command.

It will be useful to refer to the following manuals:

B1000 MANUALS	FORM
ODESY Installation Manual	1129384
ODESY Terminal Operator's Manual	1131851

B2000/3000/4000 MANUALS	FORM
ODESY User's Information Manual	2014064
ODESY Console and Terminal Operator's Manual	2014072
ODESY Capabilities Manual	2014379

APPENDIX A

B1000 AND B2000/3000/4000 REFERENCE MANUALS

B1000 MANUALS	FORM
System Software Operation Guide, Volume 1	1108982
System Software Operation Guide, Volume 2	1108966
COBOL Reference Manual	1057197
COBOL74 Reference Manual	1108883
CANDE Reference Manual	1090586
GEMCOS User's/Reference Manual	1093499
GEMCOS Formatting Guide	1106531
DMSII Reference Manual	1127222
DMSII Inquiry Reference Manual	1108875
RPG Reference Manual	1057189
SORT Reference Manual	1090594
System Communication Module (SYCOM)	1108859
Network Definition Language (NDL) Reference Manual	1073715

B2000/3000/4000 MANUALS	FORM
System Software Operations Guide, Volume 1	1127529
System Software Operations Guide, Volume 2	1127321
System Software Programmer's Guide	1090685
COBOL Reference Manual	1108909
COBOL74 Reference Manual	1090735
CANDE Reference Manual	1108834
GEMCOS User's/Reference Manual	1121316
GEMCOS Formatting Guide	
GEMCOS Format Generator User's Reference Manual	
DMSII Reference Manual	1108925
DMSII Installations and Operations Guide	1127313
DMSII Host Reference Manual	1108818
RPG Reference Manual	1090768
WFL Reference Manual	1090743

APPENDIX B
PLANNING GUIDE

Advance preparation and planning make a migration run smoothly. This section is designed to aid you in defining and planning the data entry portion of the migration.

The planning guide provides a set of worksheets for use for record keeping and future reference. We recommend that you use the provided worksheets for keeping track of the migration process. This planning provides an orderly summary of all the relevant information required for the migration.

This section also includes a checklist to help you keep track of the completed forms.

On these forms, alphabetic entries are followed by an A, and numeric entries are followed by an N. Entries that can be alphabetic, numeric, or special characters are followed by A/N.

The number of lines on these forms does not necessarily correspond to the number of characters or digits indicated after each line.

As you complete each form, write your initials and the date the form was completed. This is useful for good record keeping and for future reference.

REQUIRED DATA ITEMS CHECKLIST

Here is a checklist to help you collect the data needed to fill out each form. To make sure you have all the data needed, complete the following steps. Definitions of data items follow each form. If you have any questions concerning what data is being requested, refer to the appropriate definition page.

1. Collect the data required for each form. You either need to provide specific data, or you need to answer a question.

2. When you have collected the data, check off that item.

MIGRATION PLANNING WORKSHEETS

CUSTOMER PROFILE

Account Information

Customer's Name ..... _____
Customer's Address ..... _____

Burroughs Branch Information

District Code ..... _____
Branch Code ..... _____
Line of Business Code ..... _____
Account Manager's Name ..... _____
Sales Respresentative ..... _____
Field Engineer ..... _____

CONFIGURATION DETAIL
(For each system number)

Source System

Processor Designation ..... _____
Operating System ..... _____
Data Communications System ..... _____
Data Management System ..... _____

Target System

Processor Designation ..... _____
Operating System ..... _____
Data Communication System ..... _____
Data Management System ..... _____

WORK SCHEDULE
(For each weekday)

Work Days and Hours ..... _____

MIGRATION PLANNING WORKSHEETS

HOLIDAY SCHEDULE

Holidays ..... _____

PERSONNEL RESOURCE DESCRIPTION
(For each resource)

Resource Class ..... _____
Power Factor ..... _____
Count ..... _____
Cost Per Unit Per Hour ..... _____
Utilization Loss ..... _____

COMPUTER RESOURCE SCHEDULE

Average Time Computer is Available Per Week

Source Computer ..... _____
Host Computer ..... _____

Cost Per Hour for the Use of Computer

Source Computer ..... _____
Host Computer ..... _____

SYSTEM CHARACTERISTICS
(For each system ID)

System ID ..... _____
Priority ..... _____
Documentation Level ..... _____

FIXED COSTS

Fixed Costs Per Day ..... _____

MIGRATION PLANNING WORKSHEETS

PROGRAM DETAILS

(One form for each system identification or source/target language combination)

System Identification	-----
Source Language	
Target Language	
Program Identifications	
Numbers of Statements	
Types of Migrations Required	
Program Volatility?	-----
Optimization?	-----
Estimated Migration Times	-----

FILE DETAILS

(One form for each system identification or source/target language combination)

System Identification	-----
File Identifications	-----
Source Files Organizations	-----
Target Files Organizations	-----
Variable Record Length?	-----
COBOL Record Format Availability?	-----
Computer Time Required?	-----
Estimated Length	-----

MISCELLANEOUS TASK DETAIL

(One form for each system identification)

System Identification	-----
Task Identification	-----
Computer Required?	-----
Estimated Length	-----

MIGRATION PLANNING WORKSHEETS

SUBSYSTEM RELATIONSHIPS

System IDs	_____
Subsystem IDs	_____
Resource Classes	_____

SUBSYSTEM RELATIONSHIPS

Subsystem IDs	_____
Successor Subsystem IDs	_____

MIGRATION PLANNING WORKSHEETS

1. CUSTOMER PROFILE

ACCOUNT INFORMATION:

NAME _____ (40 CHAR A/N)

STREET _____ (40 CHAR A/N)

CITY & STATE _____ (40 CHAR A/N)

BURROUGHS BRANCH INFORMATION:

DISTRICT CODE ___ (3 CHAR N)

BRANCH CODE ___ (3 CHAR N)

LINE OF BUSINESS CODE ___ (2 CHAR N)

ACCOUNT MANAGER _____ (20 CHAR A/N)

SALES REPRESENTATIVE _____ (20 CHAR A/N)

FIELD ENGINEER _____ (20 CHAR A/N)

DATE _____

INIT _____

MIGRATION PLANNING WORKSHEETS

2. CONFIGURATION DETAIL

	SOURCE SYSTEM	TARGET SYSTEM	
SYSTEM NUMBER	-		(1 CHAR N)
PROCESSOR	_____	_____	(15 CHAR A/N)
OPERATING SYSTEM	_____	_____	(15 CHAR A/N)
DATA COMMUNICATION	_____	_____	(15 CHAR A/N)
DATA MANAGEMENT SYSTEM	_____	_____	(15 CHAR A/N)
SYSTEM NUMBER	-		(1 CHAR N)
PROCESSOR	_____	_____	(15 CHAR A/N)
OPERATING SYSTEM	_____	_____	(15 CHAR A/N)
DATA COMMUNICATION	_____	_____	(15 CHAR A/N)
DATA MANAGEMENT SYSTEM	_____	_____	(15 CHAR A/N)
SYSTEM NUMBER	-		(1 CHAR A/N)
PROCESSOR	_____	_____	(15 CHAR A/N)
OPERATING SYSTEM	_____	_____	(15 CHAR A/N)
DATA COMMUNICATION	_____	_____	(15 CHAR A/N)
DATA MANAGEMENT SYSTEM	_____	_____	(15 CHAR A/N)
		DATE	_____
		INIT	_____

MIGRATION PLANNING WORKSHEETS

EXAMPLE

CONFIGURATION DETAIL

	SOURCE SYSTEM	TARGET SYSTEM	
SYSTEM NUMBER	1		(1 CHAR N)
PROCESSOR	IBM SYSTEM 3	B 4000/B 3000	(15 CHAR A/N)
OPERATING SYSTEM	DOS/VSI	B 4000 MCP	(15 CHAR A/N)
DATA COMMUNICATIONS	CICS	GEMCOS	(15 CHAR A/N)
DATA MANAGEMENT SYSTEM	IDMS	DMS II	(15 CHAR A/N)

MIGRATION PLANNING WORKSHEETS

CONFIGURATION DETAIL DEFINITIONS

FIELD NAME	FIELD SIZE	FIELD TYPE	DESCRIPTION
SYSTEM NUMBER	1	N	Enter the number to identify source/target processor pair.
PROCESSOR	15	A/N	Manufacturer's processor designation.
OPERATING SYSTEM	15	A/N	Name of the system's operating system software.
DATA COMMUNICATIONS	15	A/N	Name of the system's data communication software.
DATA MANAGEMENT SYSTEM	15	A/N	Name of the system's data management system.

MIGRATION PLANNING WORKSHEETS

3. PROGRAM DETAIL

SYSTEM IDENTIFICATION _____ (15 CHAR A/N)

SOURCE LANGUAGE _____ (30 CHAR A/N)

TARGET LANGUAGE _____ (30 CHAR A/N)

PROGRAM ID (10 CHAR A/N)	NO. OF STMTS (6 CHAR N)	TYPE OF CONVER (T,R,S,D)	PROG VOLIT'Y (Y, N)	OPTIMIZED (Y, N)	EST CONV TIME (4 CHAR N)
----------	------	---	---	---	----.--
----------	------	---	---	---	----.--
----------	------	---	---	---	----.--
----------	------	---	---	---	----.--
----------	------	---	---	---	----.--
----------	------	---	---	---	----.--
----------	------	---	---	---	----.--
----------	------	---	---	---	----.--
----------	------	---	---	---	----.--
----------	------	---	---	---	----.--
----------	------	---	---	---	----.--
----------	------	---	---	---	----.--

DATE _____

INIT _____

MIGRATION PLANNING WORKSHEETS

EXAMPLE

PROGRAM DETAIL

SYSTEM IDENTIFICATION PAYROLL (15 CHAR A/N)

SOURCE LANGUAGE IBM 360/370 BAL (30 CHAR A/N)

TARGET LANGUAGE BURROUGHS MS COBOL (30 CHAR A/N)

PROGRAM ID (10 CHAR A/N)	NO. OF STMTS (6 CHAR N)	TYPE OF CONVER (T,R,S,D)	PROG VOLIT'Y (Y,N)	OPTIMIZED (Y,N)	EST CONV TIME (4 CHAR N)
TAXES	99565	T	Y	N	020.1
DEDUCTIONS	37500	T	N	Y	033.3
PENSIONS	12300	D	N	N	000.5

MIGRATION PLANNING WORKSHEETS

PROGRAM DETAIL DEFINITIONS

FIELD NAME	FIELD SIZE	FIELD TYPE	DESCRIPTION
SYSTEM IDENTIFICATION	15	A/N	The name of the system the program is in.
PROGRAM IDENTIFICATION	10	A/N	The program's name.
NUMBER OF STATEMENTS	6	N	The number of statements in the program.
SOURCE LANGUAGE	30	A/N	The language the program is written in before migration.
TARGET LANGUAGE	30	A/N	The language the program will be converted to.
TYPE OF MIGRATION REQUIRED	1	A	TYPE OF MIGRATION: T = Translate R = Rewrite S = Substitute D = Drop
PROGRAM VOLATILITY	1	A	Y or N. Yes, if frequent changes are made to this program.
OPTIMIZATION	1	A	Y or N. Yes, if the translated program is to be optimized.
ESTIMATED MIGRATION TIME	4	N	Man-days in 1/10 of a day increments.

MIGRATION PLANNING WORKSHEETS

3. FILE DETAIL

SYSTEM IDENTIFICATION_____ (15 CHAR A/N)

FILE ID (10 CHAR A/N)	SOURCE FILE ORG (D,M,I, R,S,O)	TARGET FILE ORG (M,I,R, S,O)	VAR'BLE RECORD LENGTH (Y,N)	COBOL RECORD FORMAT (Y,N)	ESTIMATED CONVER TIME (4 CHAR N)
----------	---	---	---	---	---•-
----------	---	---	---	---	---•-
----------	---	---	---	---	---•-
----------	---	---	---	---	---•-
----------	---	---	---	---	---•-
----------	---	---	---	---	---•-
----------	---	---	---	---	---•-
----------	---	---	---	---	---•-
----------	---	---	---	---	---•-
----------	---	---	---	---	---•-
----------	---	---	---	---	---•-
----------	---	---	---	---	---•-
----------	---	---	---	---	---•-

MIGRATION PLANNING WORKSHEETS

EXAMPLE

FILE DETAIL

SYSTEM IDENTIFICATION PAYROLL (15 CHAR A/N)

FILE ID (10 CHAR A/N)	SOURCE FILE ORG (D,M,I, R,S,O)	TARGET FILE ORG (M,I,R, S,O)	VAR*BLE RECORD LENGTH (Y,N)	COBOL RECORD FORMAT (Y,N)	ESTIMATED CONVER TIME (4 CHAR N)
EMPLOYEE	D	S	N	N	023.5
--------	---	---	---	---	--- -
PRSHAR	R	R	Y	N	008.5
--------	---	---	---	---	--- -
--------	---	---	---	---	--- -
--------	---	---	---	---	--- -
--------	---	---	---	---	--- -

MIGRATION PLANNING WORKSHEETS

FILE DETAIL DEFINITIONS

FIELD NAME	FIELD SIZE	FIELD TYPE	DESCRIPTION
SYSTEM IDENTIFICATION	15	A/N	The name of the system this file is in.
FILE IDENTIFICATION	10	A/N	This file's name.
SOURCE FILE ORGANIZATION	1	A	Type of file organization: D = Direct R = Random S = Sequential I = Indexed M = Data Base O = Other
TARGET FILE ORGANIZATION	1	A	Type of file organization. D is not allowed.
VARIABLE RECORD LENGTH	1	A	Y or N. Does this file contain variable length records?
COBOL RECORD FORMAT AVAILABLE	1	A	Y or N. Is a COBOL description available?
ESTIMATED MIGRATION TIME	4	N	Man-days in 1/10 of a day increments.

MIGRATION PLANNING WORKSHEETS

4. MISCELLANEOUS TASK DETAIL

SYSTEM IDENTIFICATION _____ (15 CHAR A/N)

TASK ID (10 CHAR A/N)	COMPUTER TIME REQUIRED (Y,N)	ESTIMATED TIME (4 CHAR N)
_____	___	____
_____	___	____
_____	___	____
_____	___	____
_____	___	____
_____	___	____
_____	___	____
_____	___	____
_____	___	____
_____	___	____
_____	___	____
_____	___	____
_____	___	____
_____	___	____
_____	___	____

DATE _____

INIT _____

MIGRATION PLANNING WORKSHEETS

EXAMPLE

MISCELLANEOUS TASK DETAIL

SYSTEM IDENTIFICATION PAYROLL (15 CHAR A/N)

TASK ID (10 CHAR A/N)	COMPUTER TIME REQUIRED (Y,N)	ESTIMATED TIME (4 CHAR N)
GTIMECRDS	N	003.5
CALCPAY	Y	002.5
CALCTAXS	Y	004.0

MIGRATION PLANNING WORKSHEETS

MISCELLANEOUS TASK DETAIL DEFINITIONS

FIELD NAME	FIELD SIZE	FIELD TYPE	DESCRIPTION
SYSTEM IDENTIFICATION	15	A/N	The name of the system the task is in.
TASK IDENTIFICATION	10	A/N	The name you've assigned the task.
COMPUTER TIME REQUIRED	1	A	Y or N. Is computer time required for this task?
ESTIMATED LENGTH	4	N	Man-days in 1/10 of a day increments.

MIGRATION PLANNING WORKSHEETS

5. WORK SCHEDULE

(Hours per day, 2 CHAR, N)

SUNDAY ___

MONDAY ___

TUESDAY ___

WEDNESDAY ___

THURSDAY ___

FRIDAY ___

SATURDAY ___

DATE _____

INIT _____

MIGRATION PLANNING WORKSHEETS

EXAMPLE

WORK SCHEDULE

SUNDAY	00
MONDAY	08
TUESDAY	08
WEDNESDAY	08
THURSDAY	08
FRIDAY	04
SATURDAY	04

MIGRATION PLANNING WORKSHEETS

6. HOLIDAY SCHEDULE

MM DD YY (6 CHAR N)	MM DD YY (6 CHAR N)	MM DD YY (6 CHAR N)
/ /	/ /	/ /
/ /	/ /	/ /
/ /	/ /	/ /
/ /	/ /	/ /
/ /	/ /	/ /
/ /	/ /	/ /
/ /	/ /	/ /
/ /	/ /	/ /
/ /	/ /	/ /
/ /	/ /	/ /
/ /	/ /	/ /

DATE _____

INIT _____

MIGRATION PLANNING WORKSHEETS

EXAMPLE

HOLIDAY SCHEDULE

MM DD YY (6 CHAR N)	MM DD YY (6 CHAR N)	MM DD YY (6 CHAR N)
01/ 01/ 83	02/ 21/ 83	05/ 30/83
07/ 04/ 83	08/ 05/ 83	10/ 10/83
11/ 08/ 83	11/ 11/ 83	11/ 24/83
12/ 23/ 83	12/ 24/ 83	12/ 25/83
12/ 26/ 83	12/ 27/ 83	12/ 28/83
12/ 29/ 83	12/ 30/ 83	/ /
/ /	/ /	/ /
/ /	/ /	/ /
/ /	/ /	/ /
/ /	/ /	/ /

MIGRATION PLANNING WORKSHEETS

7. PERSONNEL RESOURCE DESCRIPTION

RESOURCE CLASS (15 CHAR A/N)	POWER FACTOR (3 CHAR N)	COUNT (4 CHAR N)	COST PER UNIT PER HOUR (4 CHAR N)	UTILIZATION LOSS (%) (2 CHAR N)
----------	-.---	----	-----	---
----------	-.---	----	-----	---
----------	-.---	----	-----	---
----------	-.---	----	-----	---
----------	-.---	----	-----	---
----------	-.---	----	-----	---
----------	-.---	----	-----	---
----------	-.---	----	-----	---

DATE ----

INIT ----

MIGRATION PLANNING WORKSHEETS

EXAMPLE

PERSONNEL RESOURCE DESCRIPTION

RESOURCE CLASS (15 CHAR A/N)	POWER FACTOR (3 CHAR N)	COUNT	COST (PER UNIT PER HOUR (4 CHAR N)	UTILIZATION LOSS (%) (2 CHAR N)
SYSTEMS ANALYST	1.50	005	0080	00
SAM WILLS	0.75	001	0024	10
JANE PETERS	1.75	001	0031	00
POLLY BROWN	0.75	001	0021	70
PROGRAMMER2	0.50	015	0012	15
BILL JONES	0.75	001	0018	25
PROGRAMMER	1.50	005	0080	00
MARY SMITH	0.75	001	0024	10

MIGRATION PLANNING WORKSHEETS

PERSONNEL RESOURCE DESCRIPTION DEFINITIONS

FIELD NAME	FIELD SIZE	FIELD TYPE	DESCRIPTION
RESOURCE CLASS	15	A/N	Name of the resource (either generic, e.g., SYSTEMS ANALYST or by name, e.g., JOAN SMYTHE.
POWER FACTOR	3	N	A numeric estimate relating to the resource's relative efficiency (1.0 IS AVERAGE).
COUNT	3	N	Number in this resource class available to work on this migration. If the resource is by name, this field should be 1.
COST (PER UNIT PER HOUR)	4	N	Resource cost per unit per hour (salary).
UTILIZATION LOSS	2	N	A percentage of time this resource is to be engaged in non-migration related activities.

MIGRATION PLANNING WORKSHEETS

8. COMPUTER RESOURCE SCHEDULE

AVERAGE TIME COMPUTER IS AVAILABLE PER WEEK

SOURCE COMPUTER SYSTEM

___ . __ (4 CHAR N)

HOST COMPUTER SYSTEM

___ . __ (4 CHAR N)

COST PER HOUR FOR THE USE OF COMPUTER

SOURCE COMPUTER SYSTEM

_____ (4 CHAR N)

HOST COMPUTER SYSTEM

_____ (4 CHAR N)

DATE _____

INIT _____

MIGRATION PLANNING WORKSHEETS

EXAMPLE

COMPUTER RESOURCE SCHEDULE

AVERAGE TIME COMPUTER IS AVAILABLE PER WEEK

SOURCE COMPUTER SYSTEM	HOST COMPUTER SYSTEM
038.0 (4 CHAR N)	055.2 (4 CHAR N)

COST PER HOUR FOR THE USE OF COMPUTER

SOURCE COMPUTER SYSTEM	HOST COMPUTER SYSTEM
0025 (4 CHAR N)	0013 (4 CHAR N)

MIGRATION PLANNING WORKSHEETS

COMPUTER RESOURCE SCHEDULE DEFINITIONS

FIELD NAME	FIELD SIZE	FIELD TYPE	DESCRIPTION
AVERAGE TIME COMPUTER IS AVAILABLE PER WEEK	4	N	Number of hours per week the source and host computers will be available per week given to the nearest 1/10 of an hour.
COST PER HOUR FOR THE USE OF COMPUTER	4	N	Rates of charge per hour for the use of the source and host computers.

MIGRATION PLANNING WORKSHEETS

9. SYSTEM CHARACTERISTICS

SYSTEM ID (15 CHAR A/N)	PRIORITY (2 CHAR N)	LEVEL (G,A,P,N)

DATE _____

INIT _____

MIGRATION PLANNING WORKSHEETS

EXAMPLE

SYSTEM CHARACTERISTICS

SYSTEM ID (15 CHAR A/N)	PRIORITY (2 CHAR N)	LEVEL (G,A,P,N)
PAYROLL	50	P
ACCOUNTS PAY	60	A
ACCOUNTS REC	90	G
WIP	10	N

MIGRATION PLANNING WORKSHEETS

SYSTEM CHARACTERISTICS DEFINITIONS

FIELD NAME	FIELD SIZE	FIELD TYPE	DESCRIPTION
SYSTEM ID	15	A/N	The name of this system.
PRIORITY	2	N	The priority of this system's migration as related to the other systems to be converted.
DOCUMENTATION LEVEL	1	A	Quality of the documentation available for this system. G = Good A = Average P = Poor N = None

MIGRATION PLANNING WORKSHEETS

10. SUBSYSTEM DEFINITION

SYSTEM IDENTIFICATION _____ (15 CHAR A/N)

SUBSYSTEM ID _____ (10 CHAR A/N)

RESOURCE CLASS _____ (15 CHAR A/N)

UNITS _____ _____ _____ _____

(10 CHAR A/N) _____ _____ _____ _____

SUBSYSTEM ID _____ (10 CHAR A/N)

RESOURCE CLASS _____ (15 CHAR A/N)

UNITS _____ _____ _____ _____

(10 CHAR A/N) _____ _____ _____ _____

SUBSYSTEM ID _____ (10 CHAR A/N)

RESOURCE CLASS _____ (15 CHAR A/N)

UNITS _____ _____ _____ _____

(10 CHAR A/N) _____ _____ _____ _____

DATE _____

INIT _____

MIGRATION PLANNING WORKSHEETS

EXAMPLE

SUBSYSTEM DEFINITION

SYSTEM IDENTIFICATION PAYROLL (15 CHAR A/N)

SUBSYSTEM ID PERS001 (10 CHAR A/N)

RESOURCE CLASS JOHN DOE (15 CHAR A/N)

UNITS	PERSJOB1	PERSJOB2	PERSJOB3	PERSJOB4
(10 CHAR A/N)	PERSJOB5	PERSJOB6	PERSJOB7	PERSJOB8

SUBSYSTEM ID TAXREC001 (10 CHAR A/N)

RESOURCE CLASS SYSTEM ANALYST (15 CHAR A/N)

UNITS	TAXRECCA	TAXRECNV	TAXRECUT	TAXRECNY
(10 CHAR A/N)	TAXRECNY	TAXRECNJ	TAXRECCT	TAXRECNM

SUBSYSTEM ID FEDTAXREC (10 CHAR A/N)

RESOURCE CLASS JANE SMYTHE (15 CHAR A/N)

UNITS	FEDTAX001	FEDTAX002	FEDTAX003	FEDTAX004
(10 CHAR A/N)	TAXRECNY	TAXRECNJ	TAXRECCT	TAXRECNM

MIGRATION PLANNING WORKSHEETS

SYBSYSTEM DEFINITION DEFINITIONS

FIELD NAME	FIELD SIZE	FIELD TYPE	DESCRIPTION
SYSTEM ID	15	A/N	The name of the system this subsystem is in.
SUBSYSTEM ID	10	A/N	The subsystem's name.
RESOURCE CLASS	15	A/N	Name of the resource working on this subsystem.
UNITS	10	A/N	The names of programs, files, or tasks which make up this subsystem.

MIGRATION PLANNING WORKSHEETS

11. SUBSYSTEM RELATIONSHIPS

SYSTEM IDENTIFICATION _____ (15 CHAR A/N)

SUBSYSTEM ID _____ (10 CHAR A/N)

SUCCESSOR SUBSYSTEM ID'S (10 CHAR A/N)

_____ _____

_____ _____

SUBSYSTEM ID _____ (10 CHAR A/N)

SUCCESSOR SUBSYSTEM ID'S (10 CHAR A/N)

_____ _____

_____ _____

SUBSYSTEM ID _____ (10 CHAR A/N)

SUCCESSOR SUBSYSTEM ID'S (10 CHAR A/N)

_____ _____

_____ _____

DATE _____

INIT _____

MIGRATION PLANNING WORKSHEETS

EXAMPLE

SUBSYSTEM RELATIONSHIPS

SYSTEM IDENTIFICATION PAYROLL (15 CHAR A/N)

SUBSYSTEM ID PERS001 (10 CHAR A/N)

SUCCESSOR SUBSYSTEM ID'S (10 CHAR A/N)

TAXREC001	FEDTAX001

SUBSYSTEM ID TAXREC001 (10 CHAR A/N)

SUCCESSOR SUBSYSTEM ID'S (10 CHAR A/N)

FEDTAX001	

SUBSYSTEM ID (10 CHAR A/N)

SUCCESSOR SUBSYSTEM ID'S (10 CHAR A/N)

MIGRATION PLANNING WORKSHEETS

SYSTEM RELATIONSHIPS DEFINITIONS

FIELD NAME	FIELD SIZE	FIELD TYPE	DESCRIPTION
SYSTEM IDENTIFICATION	15	A/N	The system's name that this subsystem belongs to.
SUBSYSTEM ID	10	A/N	The subsystem's name.
SUCCESSOR SUBSYSTEM ID'S	10	A/N	The subsystem IDs of the subsystems that must follow this one in the migration.

MIGRATION PLANNING WORKSHEETS

12. FIXED COST

FIXED COSTS PER DAY _____ (6 CHAR N)

DATE _____

INIT _____

MIGRATION PLANNING WORKSHEETS

EXAMPLE

FIXED COST

FIXED COSTS PER DAY 000130 (6 CHAR N)

MIGRATION PLANNING WORKSHEETS

PLANNING CHECKLIST

CUSTOMER PROFILE ..... _____

CONFIGURATION DETAIL ..... _____

WORK SCHEDULE ..... _____

HOLIDAY SCHEDULE ..... _____

PERSONNEL RESOURCE DESCRIPTION ..... _____

COMPUTER RESOURCE SCHEDULE ..... _____

SYSTEM CHARACTERISTICS ..... _____

FIXED COST ..... _____

MISCELLANEOUS TASK DETAIL ..... _____

SUBSYSTEM DEFINITION ..... _____

SUBSYSTEM RELATIONSHIPS ..... _____

Documentation Evaluation Form

Title:_____ Form No: _____
_____ Date: _____

Burroughs Corporation is interested in receiving your comments and suggestions regarding this manual. Comments will be utilized in ensuing revisions to improve this manual.

Please check type of Suggestion:

__ Addition __ Deletion __ Revision __ Error

Comments:

From:

Name _____
Title _____
Company _____
Address_____
_____
Phone Number _____ Date _____

Remove form and mail to:

Burroughs Corporation
Programming Activity - Migrations
460 Sierra Madre Villa
Pasadena, CA 91109