

**UNISYS**

**A Series  
Printing Utilities  
Operations  
Guide**

Release 3.9.0

September 1991

Priced Item

U S America  
8600 0692-000

**UNISYS**

**A Series  
Printing Utilities  
Operations  
Guide**

Copyright © 1991 Unisys Corporation  
All rights reserved.  
Unisys is a registered trademark of Unisys Corporation.

Release 3.9.0

September 1991



Printed on recycled paper

Priced Item

U S America  
8600 0692-000

The names, places, and/or events used in this publication are not intended to correspond to any individual, group, or association existing, living, or otherwise. Any similarity or likeness of the names, places, and/or events with the names of any individual, living or otherwise, or that of any group or association is purely coincidental and unintentional.

**NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THE DOCUMENT.** Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed Program Product License or Agreement to purchase or lease equipment. The only warranties made by Unisys, if any, with respect to the products described in this document are set forth in such License or Agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document or software material, including direct, indirect, special or consequential damages.

You should be very careful to ensure that the use of this information and/or software material 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.

Correspondence regarding this publication may be forwarded using the Product Information card at the back of the manual, or may be addressed directly to Unisys, Product Information, 25725 Jeronimo Road, Mission Viejo, CA 92691.

# Page Status

Page	Issue
iii	-000
iv	Blank
v through xiii	-000
xiv	Blank
xv	-000
xvi	Blank
1-1	-000
1-2	Blank
2-1 through 2-35	-000
2-36	Blank
3-1 through 3-24	-000
4-1 through 4-18	-000
A-1 through A-8	-000
Glossary-1 through 9	-000
Glossary-10	Blank
Bibliography-1	-000
Bibliography-2	Blank
Index-1 through 6	-000

Unisys uses an 11-digit document numbering system. The suffix of the document number (1234 5678-xyz) indicates the document system. The first digit of the suffix (x) designates a revision level; the second digit (y) designates an update level. For example, the first release of a document has a suffix of -000. A suffix of -130 designates the third update to revision 1. The third digit (z) is used to indicate an errata for a particular level and is not reflected in the page status summary.



# About This Guide

## Purpose

The *A Series Printing Utilities Operations Guide* describes how to use the three utilities associated with the Print System. Each utility is presented separately.

This guide explains the function of the printing utilities, and illustrates how each can be used to perform various tasks.

## Scope

This guide is concerned only with the three printing utilities. Although it does give some reminders about the use of file equation and file attributes, it does not repeat information that is contained in the *A Series Print System (PrintS/ReprintS) Administration, Operations, and Programming Guide*. The use of each utility is presented according to the utility's function.

## Audience

This guide was written for any programmer, operator, system administrator, or user needing to process or maintain backup files.

## Prerequisites

This guide was written with the assumption that you have read the *A Series Print System (PrintS/ReprintS) Administration, Operations, and Programming Guide*. You should know how to use either Command AND Edit (CANDE) or Menu-Assisted Resource Control (MARC), and should be familiar with the following concepts:

- Backup files and print requests
- Task and file equation
- Task and file attributes

Additionally, programmers should have some knowledge of Work Flow Language (WFL) jobs. Operators should have some familiarity with controlling peripheral devices and entering commands at an operator display terminal (ODT).

# How to Use This Guide

This guide is written so that the sections on each utility can be accessed separately. It would be helpful to read or scan through a section sequentially at first to get a feel for how the information is presented. However, the sections are also designed to provide easy access to specific topics.

Throughout this guide, the terms *print* and *printer* are used to mean both print and punch output and devices unless specifically noted otherwise. Also, the term *backup file* refers to both printer and punch backup files unless noted otherwise.

A number of other terminology conventions are used in this guide. Those terms that are specific to a particular utility are noted at the beginning of each section. Industry and Unisys standard terms and acronyms and their definitions, as well as some terms specific to the utilities, are listed in the glossary.

## Organization

This guide contains the following sections and appendix. In addition, a glossary, a bibliography, and an index appear at the end of this guide.

### Section 1. Overview

This section outlines the function of each utility presented in this guide.

### Section 2. The Backup Processor Utility

This section describes how to use the Backup Processor utility.

### Section 3. The SYSTEM/BACKUP Utility

This section describes how to use the SYSTEM/BACKUP utility.

### Section 4. The LTTABLEGEN Utility

This section describes how to use the LTTABLEGEN utility.

### Appendix A. Understanding Railroad Diagrams

This appendix describes how to read railroad diagrams, the syntax diagrams used in this guide.

## Related Product Information

The following documents should be read prior to or concurrent with the study of this guide. The bibliography contains a list of all documents referred to in this guide. Throughout this guide, if a manual title that contains an acronym is referenced many times, the manual is referred to by a shortened version of the title. For example, the *A Series Print System (PrintS/ReprintS) Administration, Operations, and Programming Guide* is referred to as the *Print System Guide*.

***A Series File Attributes Programming Reference Manual (form 8600 0064).  
Formerly A Series I/O Subsystem Programming Reference Manual***

This manual contains information about each file attribute and each direct I/O buffer attribute. The manual is written for programmers and operations personnel who need to understand the functionality of a given attribute. The *A Series I/O Subsystem Programming Guide* is a companion manual.

***A Series I/O Subsystem Programming Guide (form 8600 0056). Formerly  
A Series I/O Subsystem Programming Reference Manual***

This guide contains information about how to program for various types of peripheral files and how to program for interprocess communication, using port files. This guide is written for programmers who need to understand how to describe the characteristics of a file in a program. The *A Series File Attributes Programming Reference Manual* is a companion manual.

***A Series Print System (PrintS/ReprintS) Administration, Operations, and  
Programming Guide (form 8600 1039)***

This guide describes the features of the Print System and provides a complete description of its command syntax. This guide is written for programmers, operators, system administrators, and other interactive users of Menu-Assisted Resource Control (MARC) and CANDE.

***A Series System Commands Operations Reference Manual (form 8600 0395)***

This manual gives a complete description of the system commands used to control system resources and work flow. This manual is written for systems operators and administrators.

***A Series Task Attributes Programming Reference Manual (form 8600 0502).  
Formerly A Series Work Flow Administration and Programming Guide***

This manual describes all the task attributes available on A Series systems. It also gives examples of statements for reading and assigning task attributes in various programming languages. The *A Series Task Management Programming Guide* is a companion manual.

***A Series Work Flow Language (WFL) Programming Reference Manual  
(form 8600 1047)***

This manual presents the complete syntax and semantics of WFL. WFL is used to construct jobs that compile or run programs written in other languages and that perform library maintenance such as copying files. This manual is written for individuals who



## About This Guide

---

have some experience with programming in a block-structured language such as ALGOL and who know how to create and edit files using CANDE or the Editor.

# Contents

<b>About This Guide</b> .....	v
<b>Section 1. Overview</b>	
<b>Section 2. The Backup Processor Utility</b>	
<b>Initiating the Backup Processor Utility</b> .....	2-1
Initiation from MARC .....	2-2
Initiation from CANDE .....	2-2
Loading Files from the Current Session .....	2-2
Loading Files from a Previous CANDE or MARC Session .....	2-2
Loading Files from a Specific Job .....	2-3
Loading Files from a Directory .....	2-3
Loading Files from a Family .....	2-4
Setting Session Options When Initiating the Utility	2-5
Removing Backup Files .....	2-5
Initiation from a WFL Job .....	2-6
<b>Backup Processor Initiation Messages</b> .....	2-6
<b>Exiting a Backup Processor Session</b> .....	2-7
<b>Session Options</b> .....	2-8
Displaying Current Option Values .....	2-8
Assigning New Option Values .....	2-9
<b>Retrieving Files for Processing</b> .....	2-10
Loading a New Working Set .....	2-11
Viewing Large Directory Listings: The SKIP Option ...	2-13
<b>Selecting a File to Work on</b> .....	2-13
Using the SELECT Command .....	2-14
Automatic Selection .....	2-15
<b>Getting Information about Backup Files</b> .....	2-15
Determining the Search Sequence .....	2-16
Using the QUERY Command .....	2-17
Using the BACK and DIRECTORY Commands .....	2-18
<b>Viewing the Contents of Backup Files</b> .....	2-18
Using the LIST Command .....	2-18
Restricting the Displayed Column or Sequence Range .....	2-20
Combining Session Options with the LIST Command .....	2-20
Text Positioning Commands .....	2-21
FIRST Command .....	2-21
LAST Command .....	2-21
SAME Command .....	2-21
+ (Plus) Command .....	2-21

## Contents

---

– (Minus) Command .....	2-22
Using Backup Processor Commands While Viewing a File .....	2-22
<b>Copying Backup Files to Disk .....</b>	<b>2-23</b>
<b>Printing Backup Files .....</b>	<b>2-24</b>
Printing the Current File .....	2-25
Printing Any File in the Working Set .....	2-25
Printing More Than One File .....	2-25
<b>Removing Backup Files .....</b>	<b>2-26</b>
<b>Getting Help Information .....</b>	<b>2-27</b>
<b>Session Option Descriptions .....</b>	<b>2-29</b>
CARRIAGECONTROL Option .....	2-29
COLUMNS Option .....	2-30
COMPRESS Option .....	2-31
ET1100 Option .....	2-31
EXTRACT Option .....	2-32
NOCRUNCH Option .....	2-33
RECORDS Option .....	2-33
SUPPRESS Option .....	2-34
TRUNCATE Option .....	2-34
UNSEQUENCED Option .....	2-35
UPCASED Option .....	2-36
WAIT/CONTINUOUS Option .....	2-36

### Section 3. The SYSTEM/BACKUP Utility

<b>Initiating the SYSTEM/BACKUP Utility .....</b>	<b>3-1</b>
<b>Specifying a Backup File .....</b>	<b>3-3</b>
Printing Files and Directories .....	3-3
Printing Files from Disk .....	3-5
Printing Files from a Current Job .....	3-5
Printing All Files from a Job .....	3-6
Printing a Specific File from a Job .....	3-6
Printing Files from Tape .....	3-7
Printing a Single File from Tape .....	3-7
Printing Multiple Files from Tape .....	3-8
Printing Multiple Tape Reels .....	3-8
<b>SYSTEM/BACKUP Utility Options .....</b>	<b>3-9</b>
COPIES Option .....	3-10
DEVICE Option .....	3-10
ID Option .....	3-11
KEY Option .....	3-11
Key Part Specification .....	3-12
Range Part Specification .....	3-12
LSN Option .....	3-16
ND Option .....	3-16
NOINCL Option .....	3-17
RECORD Option .....	3-18
SAVE Option .....	3-19
SPACING Option .....	3-19
<b>Using SYSTEM/BACKUP to Copy Files .....</b>	<b>3-19</b>

<b>Using File Equation with SYSTEM/BACKUP</b> .....	3-20
<b>Repositioning the Input File</b> .....	3-21
<b>Compile-Time Options</b> .....	3-23
The \$IDOPTION Option .....	3-23
The \$INFOPTION Option .....	3-23
<b>Handling Parity Errors</b> .....	3-23

**Section 4. The LTTABLEGEN Utility**

<b>Basic Constructs of LTTABLEGEN Commands</b> .....	4-4
<b>Train Tables and Train Printers</b> .....	4-5
<b>Train Table Commands</b> .....	4-6
Creating and Defining Train Tables .....	4-6
ALPHATABLE Command .....	4-7
CHARSET Command .....	4-7
DEFAULT Command .....	4-7
HEXTABLE Command .....	4-7
INVCHR Command .....	4-8
MEDIUM Command .....	4-8
NAME Command .....	4-8
SLOW Command .....	4-8
TRAINID Command .....	4-9
Printing the Names or Contents of Train Tables .....	4-9
PRINTALL Command .....	4-9
PRINTNAMES Command .....	4-9
PRINTRAINID Command .....	4-10
Punching the Contents of Train Tables .....	4-10
PUNCHALPHA Command .....	4-10
PUNCHARRAY Command .....	4-10
PUNCHHEX Command .....	4-11
<b>Defining and Using Train Tables</b> .....	4-11
<b>Translation Tables and Translation Printers</b> .....	4-12
<b>Translation Table Commands</b> .....	4-13
Describing a Printset .....	4-14
Deleting a Printset .....	4-14
Printing Translation Tables and Related Information .....	4-15
PRINTID Command .....	4-15
PRINTNAMES Command .....	4-16
PRINTALL Command .....	4-16
<b>Creating and Using Translation Tables</b> .....	4-16
<b>Running the LTTABLEGEN Utility</b> .....	4-17

**Appendix A. Understanding Railroad Diagrams**

<b>What Are Railroad Diagrams?</b> .....	A-1
Constants and Variables .....	A-2
Constraints .....	A-2
<b>Following the Paths of a Railroad Diagram</b> .....	A-5
<b>Railroad Diagram Examples with Sample Input</b> .....	A-6

**Contents**

---

**Glossary** ..... 1

**Bibliography** ..... 1

**Index** ..... 1

# Figures

A-1.	Railroad Constraints .....	A-5
------	----------------------------	-----



# Tables

2-1.	Effects of Commands . . . . .	2-22
4-1.	Default Train Tables for 450- and 750-lpm Printers . . . . .	4-1
4-2.	Default Train Tables for 1100- and 1500-lpm Printers . . . . .	4-3
4-3.	Default Translation Table Names . . . . .	4-4





# Section 1

## Overview

The *A Series Printing Utilities Operations Guide* describes three utilities that are associated with the Print System:

- Backup Processor

The Backup Processor utility is primarily an interactive program that allows you to maintain and process backup files kept on disk. You can list, copy, print, and delete backup files with this utility. The utility can be used interactively through Menu-Assisted Resource Control (MARC) or Command and Edit (CANDE), or in batch mode from MARC, CANDE, or a Work Flow Language (WFL) job.

- SYSTEM/BACKUP

The SYSTEM/BACKUP utility is used primarily for processing backup files kept on magnetic tape. Although it can be used for disk backup files, in most cases the Backup Processor is easier to use. SYSTEM/BACKUP allows you to copy files from tape to tape, and from tape to disk. You can also print backup files in a variety of ways. SYSTEM/BACKUP has its own set of options in addition to allowing you to use file attributes. The utility does not, however, have an interactive mode. It can be initiated from MARC, CANDE, a WFL job, or an operator display terminal (ODT).

- LTTABLEGEN

The LTTABLEGEN utility is used to create and maintain train and translation tables for nonstandard trains, bands, or drums. The need to run this utility will seldom arise because the SYSTEM/TRAINTABLES file contains train tables for all the print trains available on Unisys printers. The utility can be run from a WFL job or an ODT.



## Section 2

# The Backup Processor Utility

The Backup Processor utility allows you to maintain and process both printer and punch backup files. By using the Backup Processor you can

- Display a list of the backup files created during the current Menu-Assisted Resource Control (MARC) or Command and Edit (CANDE) session or a previous session or job, or a list of the backup files in a particular directory (using the DIRECTORY command).
- Display information about a backup file, including the values of file attributes retained in the backup file's control record (using the QUERY command).
- Display and view the contents of a backup file (using the LIST command and the text positioning commands), and control the format of the display (using the Backup Processor options).
- Copy a backup file to disk as a data file (using the COPY command).
- Print one or more backup files (using the PRINT command).
- Remove one or more backup files from the current file list (using the REMOVE command).

The Backup Processor has options that control both the display of information during a session and the action of specific commands. These options are referred to as *session options* or *Backup Processor options*.

You can work with any number of backup files during a Backup Processor session; however, a file must be made available to the Backup Processor before you can work with it. Making files available is referred to as loading files. Files can be loaded with the BACK command or the DIRECTORY command.

The files currently available in a session are referred to as the *working set* of files. This working set can be changed any number of times during a Backup Processor session. A working set can consist of files from the following sources:

- Your current CANDE or MARC session
- Any previous CANDE or MARC session to which you have access
- Any directory to which you have access

## Initiating the Backup Processor Utility

You can initiate the Backup Processor from MARC, from CANDE, and, in a limited sense, from a Work Flow Language (WFL) job. If you initiate the Backup Processor from CANDE, you can specify an initial working set of files. This allows you to load files from any directory, session, or job as part of the initiation process. If you initiate the Backup

## The Backup Processor Utility

---

Processor from MARC, you initially access the backup files from your current MARC session.

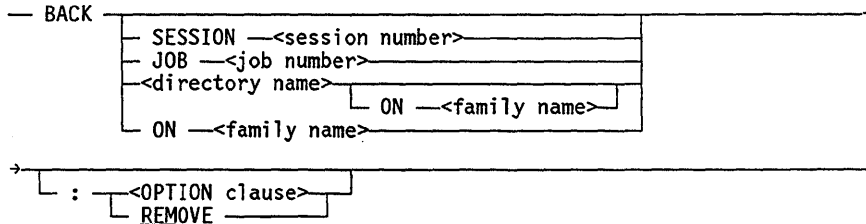
### Initiation from MARC

To use MARC to initiate the Backup Processor, enter and transmit *GO BACK* from the Action field of any MARC screen. This displays the Backup/Processor Utility screen, which has three fields. The first field allows you to enter any of the 12 session options that control how information is displayed during the Backup Processor session. These options are discussed in "Session Option Descriptions" in this section. The second field provides further control over how information is displayed. If you enter an X in the third field, the Backup Processor is initiated, all backup files created during the current MARC session are removed, and the Backup Processor is terminated. Once you have entered the required information, transmit the screen to initiate the Backup Processor utility.

Online documentation is available for each of the three fields on the Backup/Processor Utility screen.

### Initiation from CANDE

In CANDE, the Backup Processor is initiated with the BACK command. The syntax of the BACK command is



From CANDE, you can begin a Backup Processor session with any set of files. Thus, you can load files from any directory, session, or job as part of the initiation process. Additionally, you can change Backup Processor session option values during the initiation process.

### Loading Files from the Current Session

The BACK command, without any of the optional clauses, initiates the utility and loads all the backup files produced during your current CANDE session. In this case, the command syntax is simply

BACK

### Loading Files from a Previous CANDE or MARC Session

Backup files created during a previous CANDE or MARC session can be loaded as a group by using the following form of the BACK command:

BACK SESSION <session number>

MARC and CANDE session numbers go up to 9999 before they “turn over” and start again with some number greater than 100. If backup files are saved for extended periods, loading files by session number could result in a working set composed of files from two different sessions, each having the same session number.

## Loading Files from a Specific Job

Backup files created by a particular job can be loaded as a group by using the following form of the BACK command:

BACK JOB <job number>

*Note: Like session numbers, job numbers turn over when they reach 9999. Therefore, if backup files are saved for a long period of time, you could load files from more than one job, each having the same number.*

## Loading Files from a Directory

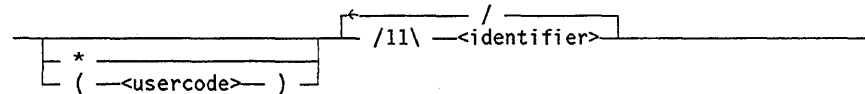
All backup files in a directory can be loaded into a Backup Processor session with either of the following commands:

BACK <directory name>

BACK <directory name> ON <family name>

The syntax of the <directory name> attribute is

<directory name>



The directory name can begin with a usercode within parentheses or with an asterisk (\*), followed by up to 11 identifiers. Each identifier must be separated from the next by a slash (/). An identifier can consist of from 1 to 17 alphanumeric characters, hyphens (-), and underscores (\_), or it can consist of a quoted string up to 17 characters long.

The following are examples of directory names:

MYDIRECTORY

(MYCODE)MYDIRECTORY

\*MYDIRECTORY

(USERA)MAINDIR/SUBDIR

## The Backup Processor Utility

---

files are found under a usercode, the Backup Processor is still initiated, but no files are loaded.

For example, assume that MYCODE is the usercode under which you are logged on, and that you want to initiate a Backup Processor session by loading the directory of system-named backup files; that is, the files named with BD as the first identifier (BD files). You can enter the following command:

```
BACK BD
```

This command causes the system to search for files of the following form:

```
(MYCODE)BD
```

If no files are found under (MYCODE)BD, no files are loaded.

When loading a file directory, you should include, if possible, the ON <family name> specification to reduce search time. If you do not include a family name clause when loading a directory, the system uses the following search order:

1. Your primary family
2. Your secondary family (if specified)
3. Your backup family

Once a backup file from the specified directory is found on a disk family, the search is discontinued.

For example, if you use a BACK \*BD command to initiate a Backup Processor session, then the system searches for the \*BD directory on your primary family. If the directory is found, all files to which you have access are loaded and the search is discontinued. If the directory is not found on your primary family, the search continues to your secondary family, and finally to your backup family.

Your primary and secondary families are specified with the CANDE *FAMILY* command. For more information about this command, refer to the *A Series CANDE Operations Reference Manual*. Your backup family is specified by the BACKUPFAMILY task attribute. For more information, refer to the *Print System Guide*.

### Loading Files from a Family

To load all the \*BD files created during your current session that are on a particular disk family, you can use the following command:

```
BACK ON <family name>
```

This command is useful for loading only the \*BD files created during your current CANDE or MARC session. If you use the command BACK \*BD instead, you could get \*BD files from a number of CANDE or MARC sessions.

### Setting Session Options When Initiating the Utility

There are 12 options you can use to control the output of information during a Backup Processor session. These options are described in detail in “Session Option Descriptions” in this section. They are mentioned here only to point out that it is possible to set one or more session options during the initiation of the Backup Processor.

The OPTION clause can be appended to any form of the BACK command. The following statements show the correct syntax for adding the OPTION clause to the various forms of the BACK command:

```
BACK:<OPTION clause>

BACK ON <family name>:<OPTION clause>

BACK <directory name>:<OPTION clause>

BACK <directory name> ON <family name>:<OPTION clause>

BACK SESSION <session number>:<OPTION clause>

BACK JOB <job number>:<OPTION clause>
```

### Removing Backup Files

If the only thing you want to do during a Backup Processor session is remove backup files, you can add the REMOVE clause to the CANDE BACK command. The removal of files can be limited to any of the following groups:

- Files from the current MARC or CANDE session
- Files from a previous MARC or CANDE session
- Files from a previous job
- Files belonging to any directory you can access

You can use any of the following forms of the BACK command to remove backup files. Which form you use depends on what files you want to remove.

```
BACK:REMOVE

BACK SESSION <session number>:REMOVE

BACK JOB <job number>:REMOVE

BACK <directory name>:REMOVE

BACK <directory name> ON <family name>:REMOVE

BACK ON <family name>:REMOVE
```



## The Backup Processor Utility

---

After a BACK command with the REMOVE clause has been processed, the utility returns you to the CANDE session that you entered the command from.

### Initiation from a WFL Job

The Backup Processor can be initiated from a WFL job, but the only action allowed is the removal of backup files. The following statement initiates the Backup Processor from a WFL job:

```
RUN SYSTEM/PRINT/BACKUP/PROCESSOR("<REMOVE clause>")
```

The REMOVE clause can take any of the following forms, though the directory name form is the most useful:

```
:REMOVE
```

```
SESSION <session number>:REMOVE
```

```
JOB <job number>:REMOVE
```

```
<directory name>:REMOVE
```

```
<directory name> ON <family name>:REMOVE
```

```
ON <family name>:REMOVE
```

Initiating the Backup Processor from a WFL job is useful when many backup files have been generated and you only want to print a few of them. As an example, assume you are running a program that produces a number of different classes of output files. Assume that one class of files is termed Diagnostics, that you do not want to print any of those files, and that DIAGNOSTIC is the directory name. The Backup Processor initialization command in the example WFL job that follows causes the DIAGNOSTIC directory to be removed before any print requests are constructed:

```
BEGIN SAMPLE/JOB;  
  RUN EXAMPLE/PROGRAM;  
  RUN SYSTEM/PRINT/BACKUP/PROCESSOR ("DIAGNOSTIC:REMOVE");  
END;
```

Running the Backup Processor from WFL is just like running any other program in terms of job flow. When REMOVE has been processed, the next statement in the job is evaluated.

### Backup Processor Initiation Messages

After a Backup Processor session has been initiated from either MARC or CANDE, a screen containing information similar to the following is displayed:

## The Backup Processor Utility

---

A15 Backup Processor (39.056.004).  
You have 2 backup files in your directory.  
Please enter a command or HELP for a list of commands.

```
Current file - File # 1: *BD/0005000/0005001/000OUT ON PACK
Created on:   Thurs, Sept 6, 1990 13:52:27
Job:         3687 "Session"
#<task number><pack name> <usercode><file name>
Task:        3837 *OBJECT/PRINTLP ON PACK
Print Status: Not yet queued           Length:    1080 lines
Printerkind: IP/LP                     PrintCopies: 1
%
```

If you make a mistake when initiating the Backup Processor, such as misspelling a name, specifying a nonexistent family or directory, or even just specifying a directory, session, or job that contains no backup files, then the Backup Processor is still initiated, but no files are loaded. In this situation, the following message is displayed upon initiation of the utility:

```
A15 Backup Processor (39.192.003).
You have no backup files in your directory.
Please enter a command or HELP for a list of commands.
%
```

If you use MARC or CANDE to initiate the utility merely to remove backup files, several messages appear to let you know that the job is progressing and what has been done. These messages appear only if the CANDE *MESSAGES* option is set. The following example shows the type of information displayed:

```
#RUNNING 1234
> Remove in progress...
#1234 PK47 BD/0007952/0008634/000OUTFILE REMOVED ON DISK
#1234 PK47 BD/0007952/0008635/000OUTFILE REMOVED ON DISK
The backup files from SESSION 7952 have been removed.
2 files removed.
#
```

## Exiting a Backup Processor Session

To terminate the Backup Processor utility and return to the MARC or CANDE session from which the utility was initiated, use the following command:

```
QUIT
```

The QUIT command can be abbreviated as Q.

You can request file removal when you exit a session; however, it applies only to your current working set. To remove a particular group of files, you first have to make that group your working set.

## The Backup Processor Utility

---

The file removal does not affect backup files that are in pending print requests generated by a PRINT command from the Backup Processor. Thus, if such a file has been queued for printing and has not been printed at the time the removal is initiated, then the file is not removed, even after it has been printed.

To remove all files in the working set when you end a Backup Processor session, use either of the following commands:

QUIT:REMOVE

Q:REMOVE

## Session Options

The Backup Processor has 12 session options that are used for controlling a Backup Processor session, or for specifying how files are displayed or copied:

CARRIAGECONTROL	RECORDS
COLUMNS	SUPPRESS
COMPRESS	TRUNCATE
ET1100	UNSEQUENCED
EXTRACT	UPCASED
NOCRUNCH	WAIT/CONTINUOUS

You can set or query session options with the OPTION command. Additionally, you can set some session options with the COPY and LIST commands. When you set an option with the COPY or LIST command, the value you specify overrides the previous setting, but only for the duration of the particular command. This method is further discussed with the COPY and LIST commands.

The general form of the OPTION command is

OPTION <option name>

The OPTION command can be abbreviated as OP and allows you to assign values to as many options as you want. Each option is described under "Session Option Descriptions" in this section.

## Displaying Current Option Values

To display the current setting of each session option, use the following command:

OPTION

The resulting display looks similar to the following:

Option	Value
-----	-----
WAIT/CONTINUOUS:	Wait for prompt
Carriagecontrol:	None
COLUMNS:	1 - 180
COMPRESS:	False
NOCRUNCH:	False
Records:	1 - end
SUPPRESS:	False
Truncate:	False
Unsequenced:	False
UPCASED:	False
MT983/ET1100:	False

Enter HELP <option name> for more information about an option.

%

The values shown for each option in the preceding example are the default values. These values apply to every Backup Processor session unless you change them by using either the OPTION clause of the BACK command or the OPTION command during a Backup Processor session. Also, note that the EXTRACT option is not listed. The EXTRACT option is a combination of the COLUMNS and RECORDS options. The values listed for those two options thus indicate the EXTRACT option setting.

### Assigning New Option Values

Option values can be changed during a session with the OPTION command. Individual options can be separated by a comma or a blank. Boolean-valued options can be set to TRUE by specifying the option name only.

#### Examples

The following three OPTION commands change several option values. All three forms of the OPTION command are equivalent.

```
OPTION UPCASED=TRUE, WAIT 10, RECORDS 50-500
```

```
OPTION UPCASED, WAIT 10, RECORDS 50-500
```

```
OPTION UPCASED WAIT 10 RECORDS 50-500
```

When the OPTION command is used to change option values, a list of the changed options is displayed. For the previous example, the resulting option list is displayed:

WAIT/CONTINUOUS:	10 SECONDS
RECORDS:	50 - 500
UPCASED:	TRUE

%

## The Backup Processor Utility

---

The values of options not specified in the OPTION command remain as they were, and are not displayed. Notice that because the UPCASED option was specified, the display appears in all uppercase characters.

## Retrieving Files for Processing

After you have initiated a Backup Processor session, you need to find out which files are in your current working set. These can be backup files from any directory, job, or MARC or CANDE session to which you have access. The set of files you have to work with in a session initially depends on what form of the BACK command you use to initiate the session. Refer to "Initiating the Backup Processor Utility" in this section for more information.

The DIRECTORY command allows you to display a list of the current working set of backup files, or to load a new working set and display it. The syntax of the DIRECTORY command is

```
— DIRECTORY —
  |-----|-----|-----|
  | SESSION —<session number> | SKIP —<integer> |
  |-----|-----|-----|
  | JOB —<job number> |
  |-----|-----|
  |<directory name> |
```

To display the current set of files, enter the following:

```
DIRECTORY
```

The DIRECTORY command can be abbreviated as DIR, and can be entered at any time during a session. For instance, you can enter a DIRECTORY command while you are listing a file; doing so does not terminate the LIST command. For more information, refer to "Viewing the Contents of Backup Files" in this section.

The listing of your working set of files includes the file number, the file title, the family name, and the size (in lines) of the file. The file number is an integer assigned to each file in the working set by the Backup Processor. The file title can be a user- or system-assigned name.

Following is an example of the display that results from a DIRECTORY command:

```
Current Directory: Session 5555 (6 files)

File # Title # Lines
# 1 BD/0005555/0003333/000LINE ON BACKPACK 99
# 2 BD/0005555/0003334/000OUT ON BACKPACK 236
# 3 BD/0005555/0004444/000PRNT ON BACKPACK 550
# 4 BD/0005555/0004444/001PRNT ON BACKPACK 11209
# 5 MY/FILE/1 ON MYPACK 3001
# 6 MY/FILE/2 ON MYPACK 684
%
```

### Loading a New Working Set

To load and display a new working set of backup files, use any of the following forms of the DIRECTORY command:

```
DIRECTORY SESSION <session number>
```

```
DIRECTORY JOB <job number>
```

```
DIRECTORY <directory name>
```

These commands load all the backup files in the specified MARC or CANDE session, job, or directory into the Backup Processor as your new working set, and display a list of the files. You can now use the Backup Processor to work with any of these files. The files in any previous working set are no longer available unless you recall them with the appropriate DIRECTORY command.

When you want to load files from a particular directory, you can specify the family the directory is on. Specifying the family of a directory saves search time. The syntax for this form of the command is

```
DIRECTORY <directory name> ON <family name>
```

If you do not include a family name clause when loading a directory, the system uses the following search order to find the specified directory:

1. Your primary family
2. Your secondary family (if specified)
3. Your backup family

Once a file from the specified directory is found on a family, the search is discontinued. If you specify a new directory, session, or job, and no files are found, then your current working set of files is retained.

### Examples

The following example situations show the use of the DIRECTORY command to load a new working set into a Backup Processor session. Assume the following conditions:

1. In a CANDE session numbered 4334, you ran three programs. These three programs generated the following files:

- Program 1:

```
REPORT/SALES  
REPORT/PURCHASES  
REPORT/INVENTORY  
REPORT/ORDERS
```

## The Backup Processor Utility

---

- Program 2:

\*BD/0009999/0007777/000LINE

\*BD/0009999/0007777/001LINE

- Program 3:

DATA/SAMPLE

GRAPH/SAMPLE

CHART/SAMPLE

2. All these backup files have been saved on disk under the family name of MYFAMILY.
3. Each program was run as a separate job, but you know only the job number for program 2, which is 9999.

These backup files can be loaded into a Backup Processor session in a number of ways, depending on which files you want to work with.

### Situation 1

To load all the files generated during the CANDE session, use the DIRECTORY command with the session number, as the following command shows:

```
DIR SESSION 4334
```

### Situation 2

To load only the files in the REPORT directory—those generated by program 1—use the following command:

```
DIR REPORT
```

### Situation 3

The \*BD files generated by program 2 can be loaded in several ways. If you want to work with only those two files, then you can load them by using either of the following commands:

```
DIR ON MYFAMILY
```

```
DIR *BD/0004334
```

The first command works only if you are still in CANDE session 4334. The second command loads all \*BD files generated in session 4334.

Your other choice is to load the \*BD directory, and then search through it to find the two files from job 4334. The \*BD directory can be loaded with either of the following commands:

```
DIR *BD
```

```
DIR *BD ON MYFAMILY
```

Note that if you use the DIR \*BD command and MYFAMILY is not your primary family, the backup files from job 4334 might not get loaded into the session. They will not be loaded if any \*BD files reside on a family that precedes MYFAMILY in the search order. Therefore, when you want to load specific files, it is always better to specify the family name. If you are a nonprivileged user, you can only load those \*BD files that were created by a program running under your usercode.

#### Situation 4

Loading the backup files generated by program 3 presents a problem because you cannot create a working set consisting only of those three files. Your options are to load each file singly by its directory name (DATA, GRAPH, and CHART), or to load all the files from the CANDE session. Note that if you load the files singly, then you have to process them singly.

## Viewing Large Directory Listings: The SKIP Option

The optional SKIP clause prevents the specified number of files from being listed in the DIRECTORY display. The DIRECTORY command displays only the first 20 files in the current directory, if there are that many. If there are more than 20 files in the directory, then you must use the SKIP clause to view them. Essentially, you use the SKIP clause to scroll through a large directory listing.

For example, if your current directory contains 25 files, then entering DIR SKIP 20 causes only the last five files (files 21 through 25) to be displayed. All other files in the directory are still available; they simply are not listed.

When scrolling through the listing of your current working set, do not repeat the job, session, or directory name. If you do, the whole search and setup procedure is repeated, which slows the response time.

Note that using the SKIP clause does not change the session status. For instance, if your current file is file 1 and you enter a DIR SKIP 20 command, your current file remains file 1. It is not changed to file 21.

## Selecting a File to Work on

To perform some action on a file, such as viewing its contents, making a copy of it, or printing it, you have to make it your current file. You can choose a current file with the SELECT command, or you can make a file your current file with commands such as COPY, LIST, and PRINT by specifying a file number. (NEXT is a synonym for SELECT.)

No matter which form of the BACK command you use to initiate the Backup Processor, file 1 (whatever it is) of the current working set is automatically selected as your current file. Entering a SELECT command immediately after starting a Backup Processor session makes file 2 your current file.



## The Backup Processor Utility

---

### Using the SELECT Command

The syntax of the SELECT command is

```
SELECT  
NEXT | # <file number>
```

Entering SELECT commands without a backup file number causes files to be selected sequentially from the current working set. Providing a backup file number with the SELECT command allows you to select files at random. If you specify a backup file number, a number sign (#) must precede the number, as in the following example:

```
SELECT #5
```

Whenever a file is selected, the Backup Processor displays information about that file, as follows:

```
File # 1:      MY/FILE/1 ON MYPACK  
Created on:   Thurs, Sept 6, 1990 13:52:27  
Job:         3687 SESSION SUMMARY  
Task:        3837 *OBJECT/PRINTLP ON PACK  
Print Status: Not yet queued      Length:    1080 lines  
Printerkind: IP/LP                PrintCopies: 1  
%
```

#### Example

The following example takes you through initiating a Backup Processor session, determining which files you have available, and selecting one to work with:

1. Enter the following BACK command to load all the backup files from your current session:

```
BACK
```

The utility responds with a display such as

```
A15 Backup Processor (39.050.005).  
You have 6 backup files in your directory.  
Current file - File # 1: MY/FILE/1 ON MYPACK  
Please enter a command or HELP for a list of commands.  
  
Created on:   Thurs, Sept 6, 1990 13:52:27  
Job:         3687 "Session"  
Task:        3837 *OBJECT/PRINTLP ON PACK  
Print Status: Not yet queued      Length:    1080 lines  
Printerkind: IP/LP                PrintCopies: 1  
%
```

2. Enter the following DIRECTORY command to display your current working set of files:

```
DIR
```

The utility lists the files in your directory as follows:

Current Directory: Session 3267 (6 files)

File #	Title	# Lines
# 1	MY/FILE/1 ON MYPACK	99
# 2	MY/FILE/2 ON MYPACK	236
# 3	MY/FILE/3 ON MYPACK	550
# 4	MY/FILE/4 ON MYPACK	11209
# 5	MY/FILE/5 ON MYPACK	3001
# 6	MY/FILE/6 ON MYPACK	684

%

- At this point, file 1, MY/FILE/1, is your current file. You can select other files in the directory sequentially by entering the SELECT or NEXT command. For example, if you now enter the SELECT command, your current file would be file 2. If you enter SELECT again, your current file would be file 3, and so on. If your current file is 6, then entering a SELECT command makes file 1 your current file.

### Automatic Selection

When a file is removed from the current working set, the Backup Processor automatically selects the next available file as the current file. (An available file is a file that belongs to the working set and has not been removed by the Backup Processor.)

#### Example

Assume that the following list of files is your current working set, and file 1 is your current file:

Current Directory: Session 3267 (6 files)

File #	Title	# Lines
# 1	MY/FILE/1 ON MYPACK	99
# 2	MY/FILE/2 ON MYPACK	236
# 3	MY/FILE/3 ON MYPACK	550
# 4	MY/FILE/4 ON MYPACK	11209
# 5	MY/FILE/5 ON MYPACK	3001
# 6	MY/FILE/6 ON MYPACK	684

%

If you remove file 1 (with the REMOVE command, discussed in this section), then file 2 becomes your current file. However, if file 1 is your current file and you remove file 4, then file 5 becomes your current file.

### Getting Information about Backup Files

You can use the QUERY command to display detailed information about the current backup file. The availability of some information, however, depends on the search sequence that the Backup Processor uses in response to the QUERY command. This, in

## The Backup Processor Utility

---

turn, depends on the way that you use the BACK or DIRECTORY commands to initiate the Backup Processor session, as described in the following subsections.

### Determining the Search Sequence

Information that the Backup Processor displays in response to a QUERY command is retrieved from either the file header or the SYSTEM/BACKUPFILELIST file that the Print System uses to store information about current or pending print requests. The file header, however, is the default source, and it is the only place that the Backup Processor will search for backup file information if you used any of the following versions of the BACK or DIRECTORY commands to initiate the Backup Processor session:

```
BACK <directory name>
BACK ON <family name>
DIR <directory name>
DIR ON <family name>
```

When these commands are used, the Backup Processor will not display information for any of the following fields in response to a subsequent QUERY command:

After	Host
Checkpoint	PrintCopies
Destination	Print Status
Disposition	Save Status

Because this attribute information is stored in the SYSTEM/BACKUPFILELIST file—and not in the file header—the resulting QUERY display will indicate a status of Not Specified for the Destination field and Unknown for the others. Also, the values of the following fields may be displayed differently, depending on whether the information was obtained from the file header or the SYSTEM/BACKUP file:

```
Job
Origin
Task
```

To retrieve the complete file status display, use one of the following versions of the BACK or DIRECTORY commands to initiate the Backup Processor session:

```
BACK
BACK SESSION <session number>
BACK JOB <job number>
DIR SESSION <session number>
DIR JOB <job number>
```

These versions of the BACK or DIRECTORY commands cause the Backup Processor to search the SYSTEM/BACKUPFILELIST file first, in response to a subsequent QUERY command. The Backup Processor will then search the header file only if the SYSTEM/BACKUPFILELIST file does not have any entries corresponding to the currently specified backup files.

**Note:** The SYSTEM/BACKUPFILELIST file will not contain entries for backup files created by a specific job or session if the BDNNAME task attribute was used or if the PRINTDISPOSITION file attribute was set to DONTPRINT. These attributes are used to prevent print requests from being created automatically.

Refer to the *Print System Guide* for information about the BDNNAME and PRINTDISPOSITION attributes and the SYSTEM/BACKUPFILELIST file.

### Using the QUERY Command

The QUERY command displays detailed information about the current backup file, such as its title, destination, originating job and task, print status, and permanent file attribute values. WHAT is a synonym for QUERY.

The syntax of the QUERY command is

```
QUERY _____  
WHAT _____
```

Following is a sample display that results from a QUERY command:

```
File # 1:      *BD/0004269/0005470/0005476/000OUT ON PACK  
Created on:   Fri, Feb 15, 1991 16:23:00  
Job:         4269 "Session"  
Task:        5476 *OBJECT/PRINTLP ON PACK  
  
Print Status: Not yet queued      Length:      80 lines  
PrinterKind:  IP/LP               PrintCopies: 1  
Usercode:     (ACCT)              TrimBlanks: False  
Chargecode:   6822                Checkpoint:  False  
Origin:       LSN 320             Trainid:     EBCDIC96  
Host:         A15A                Banner:      False  
Disposition:  EOJ                 After:       As soon as possible  
Save Status:  Remove after printing Alignment:    False  
  
Pagecomp:    Not Specified  
Destination: LP11  
FormID:      Not Specified  
Transform:   Not Specified  
Note:       Not Specified  
Alignfile:   Not Specified
```

%

## The Backup Processor Utility

---

### Using the BACK and DIRECTORY Commands

When the Backup Processor searches for information about a backup file, the file header provides the default source of information. The file header is the only place that the Backup Processor will search for information if you use any of the following forms of the BACK or DIRECTORY commands:

```
BACK <directory name>
BACK ON <family name>
DIR <directory name>
DIR ON <family name>
```

When you use these commands, the following fields in the display produced by a subsequent QUERY command will not be assigned any values:

After	Host
Checkpoint	PrintCopies
Destination	Print Status
Disposition	Save Status

### Viewing the Contents of Backup Files

You can display the contents of a backup file at your terminal by using the LIST command and text positioning commands. The format and appearance of the listing can be modified by using certain session options. Changing session option values with the LIST command causes the current values to be overridden, but when the LIST command terminates, the previous values are restored.

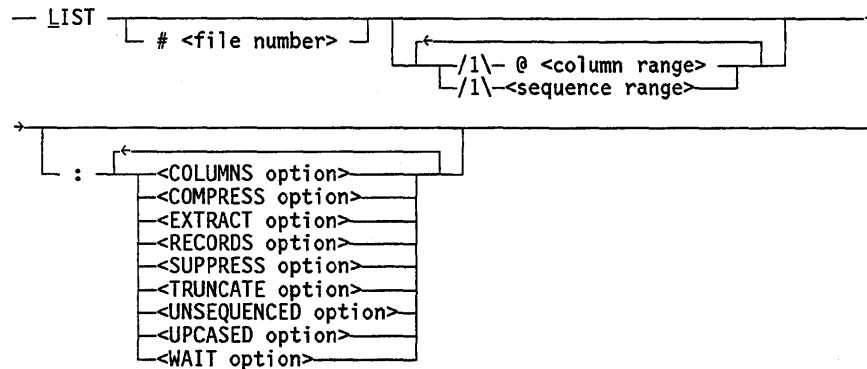
The five text positioning commands allow you to view various parts of a listed file. These commands do not disrupt the listing, and you cannot view more than the first page of a file without them.

Backup Processor commands can be entered while a file is being listed. Different commands have different effects on the listing. These effects are discussed under "Using Backup Processor Commands While Viewing a File" in this section.

### Using the LIST Command

The LIST command allows you to view a backup file at your terminal. When displayed at a terminal, the record numbers appear to the left of the text. The record numbers are assigned by the Backup Processor, beginning with 1 for the first record. Each record in the display corresponds to a print line. The backup file records that exceed the limits of the screen are wrapped around to the next line. However, remember that you can modify how a backup file is displayed with the Backup Processor options.

The syntax of the LIST command is



You can display the current file simply by entering LIST, or you can display any file in the working set by including a file number; for example, LIST #5. If you enter LIST and you do not have a current file selected, the utility selects and displays the first available file in your working set.

Unless you limit the display in some fashion, the specified file is listed beginning at the first page. A text positioning command is displayed at the cursor position in the upper left corner. The + (plus) command is displayed unless you are at the last page of the file, in which case the - (minus) command is displayed. If you want to scroll by lines or by more than one page, or if you want to use one of the other three text positioning commands, then enter the new command at the cursor position.

The following example shows the first page of a sample backup file listing:

```

+ % #1: MYFILE ON MYFAMILY (Lines 01 - 13 of 1500)
000000: This is just some example text for the purpose of
        showing you how
000001: the display of a backup file is going to appear when
        you list it with
000002: the Backup Processor LIST command.
000003:
000004: The previous record shows that you can have a blank
        line in the file;
000005: you might even have several consecutive blank lines
        in the file.
000006:
000007:
000008:
000009: You could, however, use the SUPPRESS option to reduce
        occurrences of multiple
000010: blank lines to a single blank line.
000011: Note that the cursor will appear at the plus sign (+)
        in the upper left corner
000012: of the display. This is the text positioning command.
        You can use the one
000013: displayed by pressing the [XMIT] key, or you can
        enter another text positioning command.
  
```

### Restricting the Displayed Column or Sequence Range

There are three ways to specify the range of columns you want displayed when listing a file:

- Use the @ <column range> optional clause.
- Use the COLUMNS option.
- Use the EXTRACT option.

There are also three ways to specify the sequence range you want displayed when listing a file:

- Use the <sequence range> optional clause.
- Use the RECORDS option.
- Use the EXTRACT option.

The use of the COLUMNS, RECORDS, and EXTRACT options is discussed under the appropriate headings in “Session Option Descriptions” in this section. The important thing to note here is that both the @ <column range> clause and the COLUMNS option perform the same function. Specifying both in the same LIST command causes a syntax error. Also, both the <sequence range> clause and the RECORDS option perform the same function. Specifying both in the same LIST command causes a syntax error.

#### Examples

The following are all valid statements for specifying column ranges and record ranges in a LIST command:

```
LIST @ 5-50
```

```
LIST 100-700
```

```
LIST @ 17-72 5-70
```

```
L @ 5 13
```

```
L #4 105-118
```

```
LIST #3 @ 7-80
```

### Combining Session Options with the LIST Command

As shown in the syntax diagram for the LIST command, there are nine options that can be used in conjunction with LIST. The syntax and effect of each of these options is described under “Session Option Descriptions” in this section.

## Text Positioning Commands

After the LIST command is used to initiate the listing of a file, the text positioning commands are used to scroll through the file. Text positioning commands can also initiate listing of the current file. Text positioning commands do not change the option values declared in a LIST command. However, if you start a listing with a text positioning command, then the default option values are used.

### FIRST Command

The FIRST command displays the first page of the backup file. The syntax is

— FIRST \_\_\_\_\_|

### LAST Command

The LAST command displays the last page of the backup file. The syntax is

— LAST \_\_\_\_\_|

### SAME Command

The SAME command redisplay the current page of text. If the current file has not been displayed before, the SAME command has the same effect as the FIRST command. The syntax is

— SAME \_\_\_\_\_|

### + (Plus) Command

The + (plus) command scrolls the text forward the specified number of lines or pages. The syntax is

— + \_\_\_\_\_|  
    ┌<number of lines>\_\_\_\_\_┐  
    └<number of pages> PAGES ┘

If you do not specify the number of lines or pages, the default is to scroll forward one page.

The following command, for example, scrolls the text forward five pages:

+5 P



## The Backup Processor Utility

---

### - (Minus) Command

The - (minus) command scrolls the text backward the specified number of lines or pages. The syntax is

```
-----|
|<number of lines>|
|<number of pages> PAGES|
```

If you do not specify the number of lines or pages, the default is to scroll backward one page.

The following command, for example, scrolls the text backward 500 lines:

```
-500
```

### Using Backup Processor Commands While Viewing a File

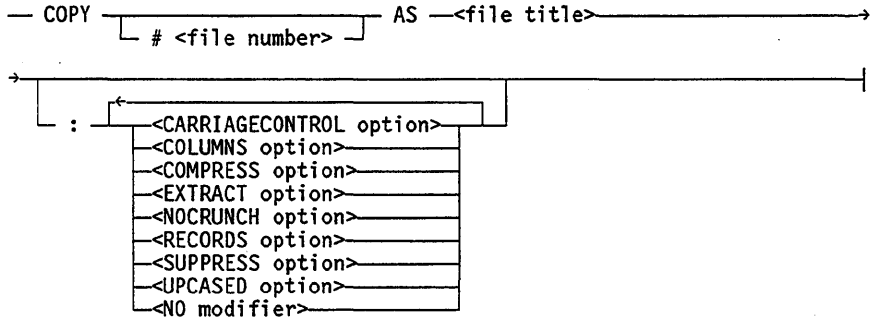
Any Backup Processor command can be entered while a file is being listed. However, different commands have different effects on a listing. The commands and their effects on a listed file are presented in Table 2-1.

Table 2-1. Effects of Commands

Command Name	Effect on a Listed File
DIRECTORY, HELP, OPTION and QUERY	The response for the entered command scrolls onto the screen. The listing is not terminated, and the position in the file is not changed. A text positioning command continues the listing.
COPY, PRINT and REMOVE	The specified file is copied, printed, or removed; the listed file is reset to the beginning, except if you remove the file. In this case, a message tells you the file has been removed and displays the name and number of the new current file. A text positioning command begins the listing of the new current file.
SELECT and LIST	The current listing is terminated, and a new file becomes the current file. In the case of the LIST command, the first page of the new file is displayed.

## Copying Backup Files to Disk

The COPY command copies a backup file to disk as a data file. Various options can be applied to the file being copied. The syntax is

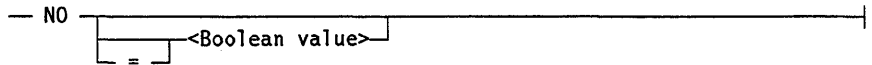


You can copy only the files in your current working set. The current file in your working set is copied unless you specify a file number. To find the file number of a file you want to copy, use the DIRECTORY command. If no backup file is currently selected, the first available backup file is copied.

The file title is the title to be given to the data file. The specification of a title can also include an *ON <family name>* clause to inform the system where to copy the file. The default is to copy the file to your primary family.

There are eight options and a NO modifier that can be invoked through the COPY command. The syntax and effect of the eight options are described under "Session Option Descriptions" in this section.

The NO modifier, when TRUE, suppresses the status report normally given by the COPY command every 10 seconds while a backup file is being copied to disk. The status report shows the number of records processed and the current percentage completed. The syntax of the NO modifier is



The default value for this modifier is FALSE.

The default record size used when copying backup files is 180 characters (30 words); however, the maximum record size is 1530 characters (255 words). The record size of the data file is determined as follows:

Backup File Column Range	Data File Record Size
<= 180	180 (30 words)
> 180	Best fit

The best fit is calculated as some multiple of the value of the BLOCKSIZE attribute. (The BLOCKSIZE attribute is described in the *A Series File Attributes Programming Reference Manual*.) As an example of the best fit, assume you are copying a backup file

## The Backup Processor Utility

---

with a column range of 290 and the value of BLOCKSIZE is 30. To accommodate 290 characters, a record size of 49 words is required. However, 49 is not a multiple of the BLOCKSIZE of 30. Therefore, the record size must be enlarged to the next multiple of 30, which in this case is 60. Therefore the record size is calculated to be 60 words.

The options specified in a COPY command are applied for that command only. Once the COPY command has completed, any specified option values return to their previous settings.

### Examples

The following are all valid COPY commands:

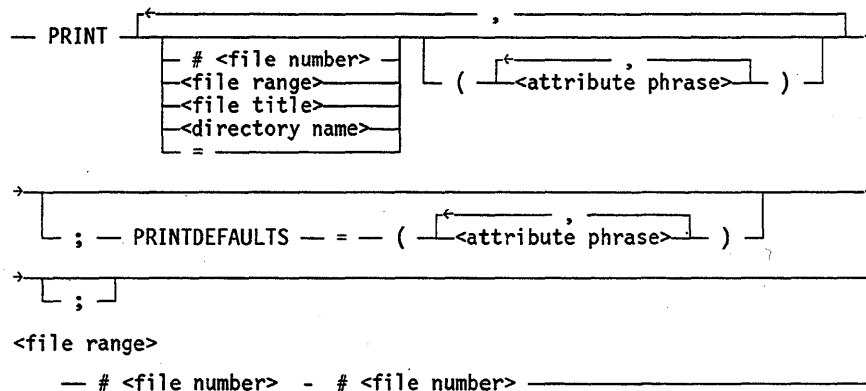
```
COPY #3 AS MYFILE:EXTRACT RECORDS 100-800
```

```
COPY AS CURRENTFILE:SUPPRESS UPCASED COMPRESS
```

```
COPY #10 AS BACKUPFILE
```

## Printing Backup Files

The PRINT command generates one or more print requests and immediately queues them for printing. A single PRINT command allows you to specify one file, several files, a directory, several directories, or a combination of each. By default, the backup file is not removed after printing. The syntax is



Any attribute phrases that immediately follow a file number, file title, or directory name are applied solely to that file or directory. Attribute phrases in the PRINTDEFAULTS clause are applied to all print requests generated by the PRINT command.

Refer to the *Print System Guide* for more information about file attribute assignment and the PRINTDEFAULTS statement.

The equal sign (=) selects all of the backup files in the current directory. The <file range> construct selects a consecutive group of backup files, which must be identified in ascending order, such as:

```
#2-#5
```

## Printing the Current File

To print the current file, use the following command:

```
PRINT
```

If you enter PRINT and there is no current file, the first available file is selected and printed. The PRINTDEFAULTS clause is the only parameter you can include when printing the current file in this way. The PRINTDEFAULTS clause can be used to assign a value to any print attribute or print modifier.

For example, to print the current file at LP4 with double-spacing, you would use the following command:

```
PRINT; PRINTDEFAULTS = (DESTINATION="LP4",DOUBLESPEACE=TRUE)
```

## Printing Any File in the Working Set

The easiest way to print a file in the current working set is to specify it by its file number, but you can also specify files by title. For example, assume the file TEST/NEW/PROGRAM belongs to the current working set, and that it is file #3. You can print the file with either of the following commands:

```
PRINT #3
```

```
PRINT TEST/NEW/PROGRAM
```

If you specify a file by its title, you must include an ON <family name> clause if the file is not on your primary family.

No matter whether you specify a file by number or by title, the file can be followed by its own set of file attribute assignments. For example, if you want to print MY/SAMPLE/PROGRAM at LP6, but defer printing until 8:00 p.m., you can use the following form of the PRINT command:

```
PRINT MY/SAMPLE/PROGRAM (DESTINATION="LP6",AFTER="20:00")
```

## Printing More Than One File

When printing multiple files with a single PRINT command, you can refer to the files by number, title, or directory. You can also select a range of files.

To print more than one file with the PRINT command, separate the file specifications with a comma. Each of the following examples prints two files. The first example prints the files 3 and \*BD/2148/3907/000PRNT, and the second example prints the files STATUS/REPORT and PROJECT/GUIDELINES.

```
PRINT *BD/2148/3907/000PRNT ON BACKPACK, #3
```

```
PRINT STATUS/REPORT ON DISK, PROJECT/GUIDELINES ON MYPACK
```

## The Backup Processor Utility

---

Each file (or range) you specify in a PRINT command can have its own set of file attribute assignments. The following example prints four files: 3, 4, 5, and STATUS/REPORT. Files 3 through 5 are not printed until 9:30 p.m. and include a banner page with the note THESE ARE MY FILES. Five copies of STATUS/REPORT are printed, but they do not wait until 9:30 p.m. to print, and no banner or note is included in the printout.

```
PRINT #3-#5 (AFTER="21:30", BANNER, NOTE="THESE ARE MY FILES."),
STATUS/REPORT ON DISK (PRINTCOPIES=5),
```

In addition to specifying attribute values for each file or file directory, you can set default values that apply to every file specified in the PRINT command by using the PRINTDEFAULTS clause. In the following example, two files are printed: 4 and REPORT/INVENTORY. File 4 is printed on LP4, but the file REPORT/INVENTORY is printed on the first available printer and includes a banner with the note ROUTE IMMEDIATELY!!. By including the PRINTDEFAULTS clause, three copies of both files 4 and REPORT/INVENTORY are printed, and both files are removed after they are printed.

```
PRINT #4 (DESTINATION=LP4),
REPORT/INVENTORY (BANNER,NOTE="ROUTE IMMEDIATELY!!");
PRINTDEFAULTS = (PRINTCOPIES=3,SAVEBACKUPFILE=FALSE)
```

## Removing Backup Files

Backup files you no longer need can be removed from the system with the REMOVE command. The Backup Processor removes only entire files. Session options are not applied during file removal.

You can choose from five ways of removing backup files:

- Use the REMOVE clause in the CANDE BACK command.
- Mark the appropriate field on the MARC Backup/Processor Utility screen.
- Initiate the Backup Processor through a WFL job.
- Use the REMOVE clause in the QUIT command.
- Use the Backup Processor REMOVE command.

Of these five methods, the first three were discussed in "Initiating the Backup Processor Utility" at the beginning of this section, and the fourth method was discussed in "Exiting a Backup Processor Session." Following is a discussion of the final method.

The REMOVE command allows you to remove files that belong to your current working set. The syntax is

```
— REMOVE —————|
| # <file number> |
| =                 |
| ALL               |
```

The REMOVE command removes the current file unless you specify one of the optional clauses. Any file in your working set can be removed by specifying the file number. The ALL and = attributes are synonyms. Using either one removes all the files in your current working set.

If you enter the REMOVE command and you do not have a current file, the first available file is selected and removed, and the file following the removed one becomes your current file.

If you try to remove a single file that belongs to a pending print request, and that request was generated by a Backup Processor PRINT command, then the utility informs you that the file cannot be removed. If you try to remove a file directory containing files that belong to pending print requests generated by the PRINT command, the utility informs you of how many files were removed and how many were not. If every file specified to be removed belongs to a pending print request generated by the PRINT command, no files are removed.

When a file has been removed, a message similar to the following is shown. The first line of this message appears only if the CANDE MESSAGES option is set.

```
# 6251 PK67 BD/0006203/0006215/0006225/0000OUT REMOVED ON PACK
File *BD/0006203/0006215/0006225/0000OUT ON PACK removed.
Current file - File # 4: *BD/0005000/0005001/0000OUT ON PACK
%
```

If you remove all the files in your current working set, then the utility displays the following message:

```
There are no backup files left in your directory.
%
```

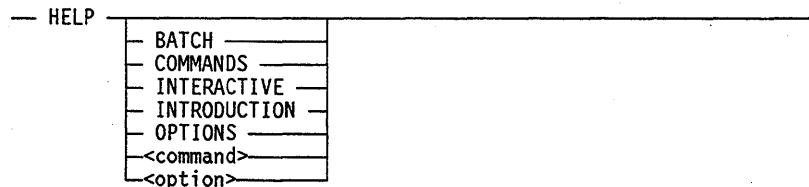
### Examples

The following are all valid REMOVE commands:

```
REMOVE
REMOVE #5
REMOVE ALL
REMOVE =
```

### Getting Help Information

The **HELP** command accesses online help information about the Backup Processor. The command gives you general information about the utility, as well as specific instructions about using the commands and options. The syntax is



Following is a brief description of the information displayed by each option of the **HELP** command:

Command Option	Information Displayed
HELP BATCH	Information is displayed about how to use the batch mode of the utility.
HELP COMMANDS	A list of the Backup Processor is displayed, including the text positioning commands, along with instructions on how to get information about each command.
HELP INTERACTIVE	Information is displayed about the tasks you can perform through the interactive mode of the utility.
HELP INTRODUCTION	A brief overview of the features of the Backup Processor is displayed.
HELP OPTIONS	A list of the Backup Processor options is displayed, along with instructions on how to get information about each option.

The **HELP <command>** and **HELP <option>** commands give you information about the specified Backup Processor command or option. A brief description of the command or option, along with the syntax diagram, is displayed. Any abbreviation for a command or option is noted.

### Session Option Descriptions

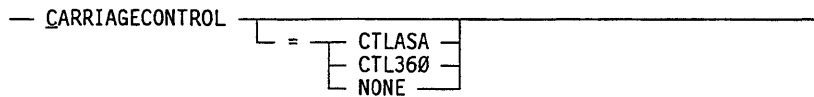
All the Backup Processor session options have default values that are used every time the utility is initiated. You can change these values anytime during a session. You can also change them temporarily in a **COPY** or a **LIST** command.

Session option values can be changed by using either the **OPTION** clause of the **BACK** command or the **OPTION** command from within a Backup Processor session. Both of these methods were discussed earlier in this section.

The following subsections describe each Backup Processor session option.

## CARRIAGECONTROL Option

The CARRIAGECONTROL option specifies the type of carriage control to be used when a backup file is copied to disk. Thus, this option applies only to the COPY command. A value can be assigned to it with the COPY or OPTION command. The syntax is



The CTLASA value specifies that the first character of the record determines the carriage motion and is not printed as data. When CTLASA is used, a carriage motion always occurs before each line is printed.

The CTL360 value specifies that the first field of the record controls the carriage motion. Unlike CTLASA, this value does not automatically perform a carriage motion before printing a line.

The default value for this option is NONE. This value specifies that no carriage control is copied to the data file. If such a data file is printed, then the printer's carriage control characteristics determine the carriage motion.

If you include CARRIAGECONTROL in your option list, but do not explicitly assign a value, the CTL360 value is assumed. That is, the following two assignment statements are equivalent:

```
CARRIAGECONTROL
```

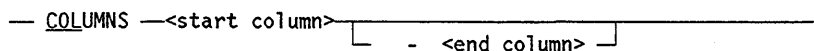
```
CARRIAGECONTROL = CTL360
```

However, if you do not include CARRIAGECONTROL in the option list at all, the default value of NONE is assigned.

The CTLASA characters and CTL360 fields are described under the CARRIAGECONTROL file attribute in the *A Series File Attributes Programming Reference Manual*.

## COLUMNS Option

The COLUMNS option specifies the range of columns to be displayed by a LIST command, or copied by a COPY command. No other commands are affected by this option. The syntax is



The <start column> attribute specifies the first column of the column range. The <end column> attribute specifies the last column of the column range. If a start column is specified without an end column, the Backup Processor utility uses 180 as the end column. The default column range is 1 through 180. The maximum column width is 1530.



## The Backup Processor Utility

---

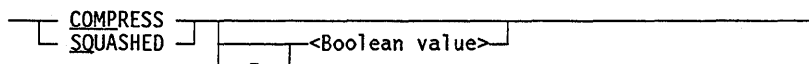
### Examples

Assume you have a backup file with columns from 1 to 500. The following table lists valid column specifications together with an explanation of their results:

Specification	Explanation
COLUMN 50	Uses columns 50 to 180
COLUMN 10-72	Uses columns 10 to 72
COL 100-200	Uses columns 100 to 200

### COMPRESS Option

The COMPRESS option, when set to TRUE, causes any occurrence of three or more consecutive blank characters to be displayed or copied as one blank character. The option does not change the source file. When the value of this option is FALSE, no compression of blank space occurs. SQUASHED is a synonym for COMPRESS. The syntax is



The default value for this option is FALSE.

### Example

Assume you have a source file that contains the following line of text:

This line contains three or more blank characters.

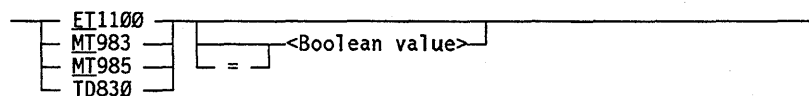
When the COMPRESS option is TRUE, the line is displayed or copied as follows:

This line contains three or more blank characters.

Notice that there are still two blanks occurring between the words *line* and *contains* and between *or* and *more*. The COMPRESS option does not affect occurrences of less than three blanks in a row.

### ET1100 Option

The ET1100 option, when TRUE, formats Backup Processor displays in screen mode, meaning that the screen is cleared before the response to a command is displayed. When the value of this option is FALSE, displayed information is in continuous mode, meaning that the previous display is not cleared before the new information is displayed. The syntax is



ET1100, MT983, MT985, and TD830 are synonyms. The default value for this option is FALSE.

### Examples

Each of the following ET 1100 option statements places the Backup Processor in screen mode:

```
ET1100=TRUE
```

```
TD=TRUE
```

```
MT=TRUE
```

Each of the following ET 1100 option statements places the Backup Processor in continuous mode:

```
MT983=FALSE
```

```
ET=FALSE
```

```
TD=FALSE
```

## EXTRACT Option

The EXTRACT option causes the specified range (records, columns, or both) of a backup file to be copied or displayed. This option is a combination of the COLUMNS and RECORDS options; therefore, the default values of these two options are the default for EXTRACT. The syntax is

```
— EXTRACT — [ /1\ - RECORDS — <range> ] , [ /1\ - COLUMNS — <range> ]
```

RECORDS specifies the range of records to be displayed or copied. COLUMNS specifies the column range to be displayed or copied. Valid <range> values for RECORDS and COLUMNS are the same as those for the RECORDS and COLUMNS options.

### Examples

The following examples show various valid forms of the EXTRACT option:

```
EXTRACT RECORDS 500-1500, COLUMNS 50-72
```

```
EX COL 12-66
```

```
EX R 250-750
```

```
EXTRACT R 5, COLUMNS 50
```

## The Backup Processor Utility

---

### NOCRUNCH Option

The NOCRUNCH option, when TRUE, prevents a backup file from being “crunched” when it is closed following a COPY command. Crunching is the process of compacting a file to eliminate any unused, allocated disk space. The syntax is

```
— NOCRUNCH [ = ] <Boolean value>
```

The default value for this option is FALSE.

### RECORDS Option

The RECORDS option specifies the record range of the backup file to be displayed or copied. The syntax is

```
— RECORDS [ = ] [ END ] [ <start record> ] [ - ] [ <end record> ] [ END ]
```

The default range is 1 through END. If a <start record> is specified without an <end record> attribute, only the starting record is displayed or copied. If the <end record> attribute is greater than the record number of the last record in the backup file, <end record> attribute is set to END.

### Examples

The following form of the RECORDS option copies or displays records 50 through 300 of a file:

```
R 50-300
```

The following form of the RECORDS option copies or displays all records in a file after record 24:

```
RECORDS 25-END
```

The following form of the RECORDS option copies or displays the first record of a file:

```
R 1
```

### SUPPRESS Option

The SUPPRESS option, when TRUE, displays or copies multiple occurrences of blank lines as a single blank line. The source file is not changed. When the value of this option is FALSE, the source file is displayed or copied without modification. The syntax is

```
— SUPPRESS [ = ] <Boolean value>
```

The default value for this option is FALSE.

### Example

Assume you have a source file that contains the following sequence of lines:

This example has three

lines of text with

multiple blank lines between each line of text.

When the value of the SUPPRESS option is TRUE, the preceding sequence of lines is displayed or copied as follows:

This example has three

lines of text with

multiple blank lines between each line of text.

## TRUNCATE Option

The TRUNCATE option, when TRUE, shortens records that cannot be displayed within the width of the screen. Any columns that extend beyond the screen width are not displayed. The option affects only the display of text. It has no effect on copied or printed files, and it does not alter the source file. When the value of this option is FALSE, the entire record is displayed. The syntax is

```
-- TRUNCATE  <Boolean value>
```

The default value for this option is FALSE.

### Example

Assume you have a source file that contains the following record:

This sentence has more characters in it than can fit on one line of the screen and would wrap around to the next line.

When the value of the TRUNCATE option is TRUE, the record shown above is displayed as follows:

This sentence has more characters in it than can fit on one

## The Backup Processor Utility

---

### UNSEQUENCED Option

The UNSEQUENCED option, when TRUE, causes backup files to be listed without record numbers. When the value of this option is FALSE, the record numbers are displayed to the left of the text. The syntax is

```
-- UNSEQUENCED  <Boolean value>
```

The default value for this option is FALSE.

Backup file records are numbered starting at 1 and are incremented by 1 for each following record. These numbers are generated by the Backup Processor; they are not equivalent to regular sequence numbers.

#### Example

When the UNSEQUENCED option is set to FALSE, a displayed backup file appears as follows:

```
000001 This is the first record of the example.  
000002 This is the second record of the example.  
000003 This is the third record of the example.  
.  
.  
.  
000100 This is the one hundredth record of the example.
```

When the UNSEQUENCED option is set to TRUE, the same backup file is displayed as follows:

```
This is the first record of the example.  
This is the second record of the example.  
This is the third record of the example.  
.  
.  
.  
This is the one hundredth record of the example.
```

### UPCASED Option

The UPCASED option, when TRUE, causes all lowercase characters to be displayed or copied as uppercase characters. This option is applied to all Backup Processor displays and messages, as well as listed and copied backup files. When the value of this option is FALSE, the Backup Processor displays and messages appear with a mixture of uppercase and lowercase characters, and backup files are displayed and copied with casing matching the source. The syntax is

```
-- UPCASED  <Boolean value>
```

The default value for this option is FALSE.

### Example

Assume you have a source file that contains the following record:

```
    this line contains no uppercase characters.
```

When the UPCASED option is TRUE, the record shown above is displayed or copied as follows:

```
    THIS LINE CONTAINS NO UPPERCASE CHARACTERS.
```

## WAIT/CONTINUOUS Option

The WAIT/CONTINUOUS option controls the rate at which text is displayed when you are listing a file. The WAIT option scrolls text by pages, and can be set to scroll pages at intervals (in seconds), or only upon entry of a text positioning command. The CONTINUOUS option scrolls text line by line, beginning at the first line of the file. There is no pause between the display of each line. The syntax is

```
  WAIT [ <integer> ]
  CONTINUOUS
```

The default for this option is WAIT for command, meaning that a page of text is retained until you enter a text positioning command or end the listing.

The optional integer value specifies the number of seconds the system waits before displaying the next page of text. You can override this time interval by entering any command or ?BRK.

If you specify the WAIT option without a time interval, then a text positioning command—FIRST, LAST, SAME, + (plus), or - (minus)—must be entered before the next page of text is displayed.

By default, terminals with a screen size greater than five lines wait for a prompt before displaying the next page of text. For all other devices, the display of the file is continuous.

### Example

The following example specifies that the system display the current page for 15 seconds before scrolling to the next page:

```
    WAIT 15
```







## The SYSTEM/BACKUP Utility

---

The value of <command phrase> attribute depends on the method of initiation, as shown in the following table:

Initiation Method	<command phrase>
MARC	WFL PB
WFL Job	PB
ODT	?PB
CANDE	WFL PB or U \$SYSTEM/BACKUP

Note that the values of the <command phrase> attributes for a WFL job and for the ODT have similar syntax. When initiating SYSTEM/BACKUP from an ODT, always confirm that the question mark (?) precedes the PB in <command phrase>.

Allowable values for the <file specification> and <options> attributes are discussed under "Specifying a Backup File" and "SYSTEM/BACKUP Utility Options" in this section. If the SYSTEM/BACKUP option ND is included as part of <options>, then <file equation statement> can be included in the initiation command. Possible values for <file equation statement> are presented under "Using File Equation with SYSTEM/BACKUP" later in this section.

Throughout this section, there are examples of initialization commands designed to focus on one particular aspect of the command. These examples use various values for <command phrase>. This convention is designed to familiarize you with all the ways you can run SYSTEM/BACKUP.

In addition to the four initiation methods just described, you can use the MARC SYSTEM/BACKUP Utility screen to initiate SYSTEM/BACKUP. To do this, enter GO BACKUP from the Action field of any MARC screen to display the SYSTEM/BACKUP Utility screen. Then enter any desired options in the Input to SYSTEM/BACKUP field. When using the SYSTEM/BACKUP Utility screen to initiate SYSTEM/BACKUP, you cannot include a file equation statement, even if you include the ND option.

### Examples

- Each of the following is a valid initialization command from MARC:

```
WFL PB "MYFILE"  
WFL PB D 1234  
WFL PB MT 10 FILE 6
```

- Each of the following is a valid initialization command from a WFL job:

```
PB "BACKUPFILE"  
PB MT 5 COPIES 5  
PB MT 10 ND SAVE; FILE BFILE (BACKUPKIND=TAPE)
```

- Each of the following is a valid initialization command from the ODT:

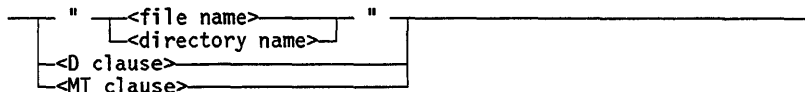
```
?PB D* DEVICE LP9
?PB "SAMPLE/FILE" KEY6 2 RANGE 00 20
?PB MT 10 ND SAVE; FILE BFILE(TITLE=NEWBACK, BACKUPKIND=TAPE)
```

- Each of the following is a valid initialization command from CANDE:

```
WFL PB "MYFILE"
U $SYSTEM/BACKUP "MYFILE"
U $SYSTEM/BACKUP MT 5 ID "TAPE FILES FROM UNIT 5"
```

## Specifying a Backup File

To obtain any results from the SYSTEM/BACKUP utility, you must include a file specification in the initialization command. The syntax is



SYSTEM/BACKUP prints and maintains backup files on both disk and tape. Disk files can be specified by file name, directory name, or the D clause. Tape files can be specified only with the MT clause.

File names and directory names must be enclosed by quotation marks. The D clause and MT clause have additional syntactical elements and are discussed under "Printing Files from Disk" and "Printing Files from Tape" in this section.

## Printing Files and Directories

A single disk file or directory of files can be specified as input to SYSTEM/BACKUP. Specifying a file name causes the named file to be printed; specifying a directory name causes all the files in that directory to be printed. Remember, file and directory names must be enclosed in quotation marks (" ").

*Note: If SYSTEM/BACKUP does not accept a file name or directory name entered in lowercase characters, then try entering the file name or directory name in uppercase characters.*

To print or punch a file or a directory, you only need to specify its name. However, you can include an ON <family name> clause, which is always advisable. (A family name can consist of up to 17 alphanumeric characters.) If you do not include the family name clause, then the system searches for the file or directory in the following order:

1. On the DISK family
2. On the PACK family
3. On the DLBACKUP family

## The SYSTEM/BACKUP Utility

---

Searching occurs on the DLBACKUP family only if the SB (Substitute Backup) system command has been used to divert backup files to the DLBACKUP family. See the *A Series System Commands Operations Reference Manual* for more information about the SB command.

If the file or directory exists on more than one family, the search is terminated after the first occurrence of the file or directory is found. If you specify a name that is both a file name and a directory name, SYSTEM/BACKUP processes the file, not the directory.

File names and directory names are composed of simple identifiers. Each identifier in a file name or a directory name must be separated by a slash (/), and each identifier can consist of up to 17 alphanumeric characters. File names and directory names can take any of the following forms:

- Up to 11 simple identifiers
- An asterisk (\*) and up to 11 simple identifiers
- A usercode (in parentheses) and up to 11 simple identifiers

If a file name is not preceded by an asterisk or a usercode, then the system searches for the file under the usercode under which you are logged on. However, if the file is not found under the usercode you are currently logged on, the system searched for the file with no usercode in the asterisk (\*) directory. A file or directory name preceded by an asterisk (\*) signifies that the file or directory is not associated with a particular usercode; however, that does not necessarily mean that the file is public. To print a nonpublic file or directory that is under someone else's usercode, you must be a privileged user, and you must precede the file name by the usercode, in parentheses.

### Examples

Assume the following hierarchy of backup files exists on a family called SALES:

```
(USER1)
  MONTH
    END
      PRELIM
      FINAL
    ESTIMATE
  BUDGET
    FINAL
    VERSION
  MONTHLY
```

The following commands print individual backup files:

```
PB "MONTH/END/FINAL"
WFL PB "BUDGET/MONTHLY"
U $SYSTEM/BACKUP "(USER1)MONTH/ESTIMATE"
WFL PB "MONTH/END/PRELIM ON SALES"
```

The following commands print entire directories:

```
PB "MONTH"
WFL PB "(USER1)BUDGET"
PB "MONTH/END"
?PB "BUDGET/FINAL"
```

## Printing Files from Disk

The *D* clause allows you to print all files generated by a current job, all files created by a completed job, or only one file. However, only files saved on disk that have standard system-assigned names can be specified through the *D* clause. The syntax is

<*D* clause>

```
— D * _____|
    |<job number>_____|
    |<system-assigned disk file title>_____|
```

## Printing Files from a Current Job

The *D \** form of the *D* clause prints all the backup files created by a job, up to the point where SYSTEM/BACKUP is initiated.

This command can be useful in a WFL job when you want to print backup files generated by certain programs immediately. In many instances, using the *D \** clause may be easier than file-equating a different value to the PRINTDISPOSITION attribute for each of a number of tasks.

### Example

Assume that you have written the following WFL job deck:

```
BEGIN SAMPLE/JOB;
  RUN EXAMPLE/PROGRAM/1;
  RUN EXAMPLE/PROGRAM/2;
  RUN EXAMPLE/PROGRAM/3;
  RUN EXAMPLE/PROGRAM/4;
END;
```

This job deck might be one that is run on a regular basis, but assume that in a particular instance, you need the output from the first two programs immediately. Rather than add a file equation statement for each of the first two programs (which would require that you know the internal file names), you could insert a SYSTEM/BACKUP initialization command into the job deck, using *D \** as input, as follows:

## The SYSTEM/BACKUP Utility

---

```
BEGIN SAMPLE/JOB;  
  RUN EXAMPLE/PROGRAM/1;  
  RUN EXAMPLE/PROGRAM/2;  
  PB D * LP4 ID "ROUTE IMMEDIATELY";  
  RUN EXAMPLE/PROGRAM/3;  
  RUN EXAMPLE/PROGRAM/4;  
END;
```

The inserted SYSTEM/BACKUP initialization command causes all the backup files from programs 1 and 2 to be routed directly to LP4. The ID option specifies that the print request is preceded by a header page containing the string, ROUTE IMMEDIATELY. Note that programs 3 and 4 will not be processed until the files from programs 1 and 2 have been printed.

### Printing All Files from a Job

To print all the backup files created by a particular job and named using the backup file standard naming conventions, you can use the D <job number> form of the D clause. This form allows you to print only files that have been saved on disk. Furthermore, this form does not work for backup files that were created with PRINTDISPOSITION = DONTPRINT, because the DONTPRINT value causes the backup file to have a usercode.

#### Example

The following initialization command prints all the backup files created by job 1234:

```
WFL PB D 1234
```

### Printing a Specific File from a Job

You can print a single \*BD file from a job with the D clause. To do so, you must provide the job number and the system-assigned file name. The file you want to print must be on disk, and it cannot have a usercode. You can use the SYSTEM/BACKUP options with this form of the D clause, or you can specify the ND option and use file equation.

If you are unfamiliar with the backup file naming conventions, refer to the information on the structure of backup files in the *Print System Guide*.

#### Example

To print a file named \*BD/0007676/0002981/000LINE from CANDE, you could use the following command:

```
WFL PB D 7676/2981/"000LINE"
```

Notice that you do not have to include the \*BD prefix or the leading zeros. And, if you do not use the SYSTEM/BACKUP SAVE option, the backup file is purged after printing. Also note that a WFL PRINT statement in place of this SYSTEM/BACKUP D clause accomplishes the same task, but the file name cannot be abbreviated—it must be \*BD/0007676/0002981/000LINE.

### Printing Files from Tape

The MT clause allows you to print backup files that have been saved on either printer or punch backup tape. By using the MT clause, you can perform the following tasks:

- Print a single backup file from tape.
- Print multiple backup files from tape.
- Print all backup files on multiple tape reels.

The syntax of the MT clause is

<MT clause>

```
— MT — " —<file name>— " —————|
      |<unit number>|<tape file spec>|
```

Tape files can be printed on a line printer, an image printer, or a card punch.

The <tape file spec> option allows you to be more explicit about the type of file to print, or where on the tape to start printing. All tapes accessed by the utility are automatically rewind.

### Printing a Single File from Tape

The MT <file name> form of the MT clause allows you to print a single file from a backup tape. When this form of the MT clause is used, the system makes a search for the specified file on every backup tape until the file is found.

#### Example

To print a backup file named MYFILE that has been saved onto a backup tape, you could use either of the following commands from CANDE:

```
WFL PB MT "MYFILE"
U $SYSTEM/BACKUP MT "MYFILE"
```

Note that the file name must be enclosed in quotation marks. Also, you must use the SAVE option if you want the file to be retained on the tape. Without the SAVE option, SYSTEM/BACKUP attempts to purge the tape after printing occurs.

## The SYSTEM/BACKUP Utility

---

### Printing Multiple Files from Tape

The MT <unit number> form of the MT clause allows you to print all the backup files on a particular tape unit. By adding FILE <file number> to the clause, you can omit printing of the files that precede the one specified by <file number>. In addition, you can include the ND option (described later in this section) to apply printing-related attributes to the print request.

#### Examples

Assume there is a backup tape mounted on unit 15. The following statement prints all the backup files on the tape:

```
WFL PB MT 15
```

The following statement prints every file except the first five:

```
WFL PB MT 15 FILE 6
```

The following statement applies the printing-related attributes, TRANSFORM and PRINTCOPIES, to the print request. Because the ND option is specified, the output does not go directly to a printer. Instead, it gets copied to the Print System where the printing-related attributes are applied.

```
WFL PB MT 15 FILE 6 ND; FILE BFILE (PRINTCOPIES=3,  
TRANSFORM="TAPETR IN SITE/TRANSFORMS")
```

### Printing Multiple Tape Reels

Sometimes an end-of-tape condition occurs during the recording of a file, so that the first part of the file is at the end of one tape reel and the remainder is at the beginning of the next tape reel. If you print a series of tape reels sequentially from the same device, then you do not have to make any specifications if files are split between reels. The system expects to find the same file type at the beginning of the next reel that was at the end of the reel just completed.

With SYSTEM/BACKUP you can print tape reels independently of one another; that is, reel 1 of a set of tapes can be printed on one device while reel 2 of the set can be printed on another. When printing tape reels independently, you must specify the file kind (printer or punch) of the first file on the reel.

The syntax of the MT clause that prints multiple reels is

<MT clause>

```
— MT —<unit number>— REEL —<number>—————→  
→ [ ( — KIND — = [ PRINTER ] ) ]
```

You do not need to specify **KIND** if a tape starts at the beginning of a file. Also, if you specify a file kind and it does not match the kind of the first file on the tape, the resulting output will, more than likely, be unrecognizable.

### Example

Assume you want to print the backup files contained on three reels of tape. All of the files are printer backup files, and the file that begins on reel 1 ends on reel 2. Also assume you have three tape drives, numbered 10, 20, and 30. You could use the following **CANDE** commands to print all the backup files:

```
WFL PB MT 10 REEL 1
WFL PB MT 20 REEL 2 (KIND = PRINTER)
WFL PB MT 30 REEL 3
```

## SYSTEM/BACKUP Utility Options

Using **SYSTEM/BACKUP** options is one way to control how the utility processes and prints backup files. Another way is to use file equation. To use file equation with the **SYSTEM/BACKUP** utility, you must specify the **ND** option. For more information, refer to the **ND** option and "Using File Equation with **SYSTEM/BACKUP**" later in this section.

The **SYSTEM/BACKUP** options are a small set of control parameters that provide both general and specialized functions. These options can be included as an optional part of the initialization command, and can be used no matter how the utility is initiated. The initialization command can contain any number of options, but each option can be specified only once. For more information about the initialization command, see "Initiating the **SYSTEM/BACKUP** Utility" earlier in this section.

Most of the options have default values that are used if the option is not specified. Other options default to the value of an equivalent or associated permanent file attribute. The permanent file attributes are maintained in the backup file's control record. Refer to the descriptions of each option for information about default values.

Some of the options cannot be used in combination with other options. Any such restrictions are noted in the descriptions of the individual options.

The 10 **SYSTEM/BACKUP** options are listed here and described in the following subsections:

COPIES	ND
DEVICE	NOINCL
ID	RECORD
KEY	SAVE
LSN	SPACING



## The SYSTEM/BACKUP Utility

---

### COPIES Option

The COPIES option specifies the number of copies of a file SYSTEM/BACKUP is to print. If more than one file is to be printed – which is possible when a directory name, the D clause, or the MT clause is used as input – then the value specified for the COPIES option applies to every file. The syntax of the COPIES option is

— COPIES —<integer>—————|

The default value for <integer> is 1.

#### Examples

The following examples show use of the COPIES option in both the WFL PB and CANDE U statements:

```
WFL PB "MYFILE" COPIES 3
U $SYSTEM/BACKUP "MYFILE" COPIES 3
```

### DEVICE Option

The DEVICE option specifies the device to use for printing or punching SYSTEM/BACKUP output. The syntax of the DEVICE option is

— CP —<unit number>—————|  
— IP —  
— LP —

With the DEVICE option, you can direct output to a card punch (CP), image printer (IP), or line printer (LP). The value of <unit number> specifies the particular device.

The DEVICE option has no default value. If the DEVICE option is not specified, the setting is determined from the file attribute KIND. If the value of KIND is PRINTER, the value of the file attribute PRINTERKIND is used to select the type of device (line printer or image printer). If the value of KIND is PUNCH, the files are routed to the first available card punch in the default pool.

The DEVICE option is ignored if the ND option has been specified. The ND option is described later in this section.

#### Examples

The following examples show use of the DEVICE option in SYSTEM/BACKUP initialization commands. Each command causes SYSTEM/BACKUP to print one copy of MYFILE at LP2.

```
WFL PB "MYFILE" LP2
U $SYSTEM/BACKUP "MYFILE" LP2
```

## ID Option

The ID option causes a header page to be printed prior to each file that SYSTEM/BACKUP prints. The text of the header page is printed in block characters, and can be up to 60 characters in length. Five lines of 10 block characters each can be printed on a page. A heading, therefore, can be more than one page long. The header page generated through the ID option is roughly equivalent to the header page generated through the Print System. The difference is that the Print System header page can contain more information.

The syntax of the ID option is

```
— ID — " <string> " _____|
```

The text of the heading is entered as the value of <string>, and it must be enclosed by quotation marks. The ID option has no default value. If the ID option is not specified, header page printing is controlled by the value of the IDOPTION compiler control option. IDOPTION is described under "Compile-Time Options" in this section.

If the ID option is specified in combination with the ND option, an error occurs that prevents the print request from being printed. The ND option is described later in this section.

### Examples

The following examples show use of the ID option in SYSTEM/BACKUP initialization commands. Each command causes a header page containing the string PRELIMINARY COPY to be printed preceding the file MYFILE.

```
WFL PB "MYFILE" ID "PRELIMINARY COPY"
```

```
U $SYSTEM/BACKUP "MYFILE" ID "PRELIMINARY COPY"
```

## KEY Option

The KEY option allows you to print portions of a file. With this option you can print defined sequences of records, or randomly selected records. The syntax of the KEY option is

```
— KEY —<key part><range part>_____|
```

The values of <key part> and <range part> combine to define a specific key to search for. A key is any part of a record or any numeric or alphanumeric string within a record that can be used to distinguish it from other records in the file. For example, one unique part of a record is its sequence number. When you want to print a specific, contiguous portion of a file, you should use a sequence number as the key. This is often the case when you are working with text files. In another situation you might want to print selected records that occur at random throughout a file. If all the records have some string in common, for example INSUFFICIENT FUNDS, you could specify that string as the key. This type of key is most useful when you are working with data files.



key length (maximum 120 characters). However, if <key part> is a compiler type, then <range part> must be specified with integers. Thus, you can print only contiguous sections of a compiler listing.

A key value or range of values must be compatible with the specified key length. That is, if the key length is specified as 3, you cannot specify a key value that requires more than or less than 3 positions to determine.

A range of values is specified with the RANGE clause. If the value at the key location falls within the range, inclusive of the boundary values, the record is printed. Range specifications must be of the same type. You cannot specify an integer as one boundary value and a character string as the other.

When specifying a range, you can use END as the second value. The END value, which must be in quotes, causes the last valid value of the specified range type to be used. For example, the following range specification causes the key to be H through Z:

```
RANGE "H" "END"
```

Similarly, the following example specifies the key to be in the range 500 through 999:

```
RANGE 500 "END"
```

If you separate the RANGE values with a colon (:), then all records included by a \$INCLUDE compiler control option are not checked for the key value. These records are printed regardless of the key value. This feature is often applied to ALGOL compiler listings.

A single value is specified by the EQUAL clause. If the value in the key location of the record matches the key, the record is printed.

### Examples

Assume you are working with a data file, named SAMPLE/FILE, formatted as follows:

Columns	Contents
01 to 08	Sequence number
10 to 23	Name data
25 to 36	Location data
38 to 50	Division data

The SAMPLE/FILE file has the following records:

## The SYSTEM/BACKUP Utility

---

00000001	ANDERSONJAMES	PHILADELPHIA	SALES
00000002	ANDREWSMORTON	LOSANGELES	MANUFACTURING
00000003	HEATHBARRY	CINCINNATI	SALES
00000004	CHURCHILLRON	LONDON	DEVELOPMENT
00000005	WILSONHENRY	DENVER	DEVELOPMENT
00000006	SMITHADAM	NEWYORK	SALES
00000007	GREENDAVE	LOSANGELES	MANUFACTURING
00000008	SHINERMARY	CHICAGO	SALES
00000009	HOLMESSTEVE	DENVER	DEVELOPMENT
00000010	MADDOXCAROL	NEWYORK	SALES
00000011	CHADWICKBRIAN	LONDON	SALES
00000012	TORVALDSARAH	DENVER	DEVELOPMENT
00000013	WATSONGEORGE	CINCINNATI	SALES
00000014	SMITHDENNIS	LOSANGELES	MANUFACTURING
00000015	KENTMELINDA	LONDON	DEVELOPMENT
00000016	JAMESHOLLY	CHICAGO	SALES
00000017	FORDHAMJIM	AUSTIN	MANUFACTURING
00000018	OLSENGRACE	NEWYORK	SALES
00000019	COLEDOUGLAS	AUSTIN	MANUFACTURING
00000020	BATESONLEWIS	LONDON	SALES
00000021	WESTROBERT	LOSANGELES	SALES
00000022	AHRONSENJAKE	CINCINNATI	SALES
00000023	HARBOUGHSID	AUSTIN	MANUFACTURING
00000024	WAKEFIELDJAMES	LONDON	DEVELOPMENT
00000025	SMITHLEONARD	PHILADELPHIA	SALES
00000026	RAWLINGSJANET	MIAMI	SALES
00000027	YOUNGSUSAN	AUSTIN	MANUFACTURING
00000028	DENTONALBERT	NEWYORK	SALES
00000029	GANTLARRY	PHOENIX	SALES
00000030	JORDANLISA	LONDON	DEVELOPMENT

To print the first 20 records of the file, you could use the following initialization command from CANDE:

```
WFL PB "SAMPLE/FILE" KEY 7 2 RANGE 00 20
```

Similarly, to print the last 10 records of the sample file, you could use either of the following commands:

```
WFL PB "SAMPLE/FILE" KEY 7 2 RANGE 21 30
```

```
WFL PB "SAMPLE/FILE" KEY 7 2 RANGE 21 "END"
```

## The SYSTEM/BACKUP Utility

---

To print the records of every person in the file whose name begins with the letter W, you could use the following commands:

```
WFL PB "SAMPLE/FILE" KEY 10 1 EQUAL "W"
```

The resulting output would be as follows:

```
00000005 WILSONHENRY    DENVER    DEVELOPMENT
00000013 WATSONGEORGE    CINCINNATI    SALES
00000021 WESTROBERT    LOSANGELES    SALES
00000024 WAKEFIELDJAMES    LONDON    DEVELOPMENT
```

Similarly, to print the records of every person in the file whose last name is SMITH, you could use the following command:

```
WFL PB "SAMPLE/FILE" KEY 10 5 EQUAL "SMITH"
```

The resulting output would be as follows:

```
00000006 SMITHADAM    NEWYORK    SALES
00000014 SMITHDENNIS    LOSANGELES    MANUFACTURING
00000025 SMITHLEONARD    PHILADELPHIA    SALES
```

To print the records of every person in the file whose last name begins with the letters A through H, you could use the following command:

```
WFL PB "SAMPLE/FILE" KEY 10 1 RANGE "A" "H"
```

The resulting output would be as follows:

```
00000001 ANDERSONJAMES    PHILADELPHIA    SALES
00000002 ANDREWMORTON    LOSANGELES    MANUFACTURING
00000003 HEATHBARRY    CINCINNATI    SALES
00000004 CHURCHILLRON    LONDON    DEVELOPMENT
00000007 GREENDAVE    LOSANGELES    MANUFACTURING
00000009 HOLMESSTEVE    DENVER    DEVELOPMENT
00000011 CHADWICKBRIAN    LONDON    SALES
00000017 FORDHAMJIM    AUSTIN    MANUFACTURING
00000019 COLEDOUGLAS    AUSTIN    MANUFACTURING
00000020 BATESONLEWIS    LONDON    SALES
00000022 AHRONSENJAKE    CINCINNATI    SALES
00000023 HARBOUGHSID    AUSTIN    MANUFACTURING
00000028 DENTONALBERT    NEWYORK    SALES
00000029 GANTLARRY    PHOENIX    SALES
```

There are numerous other useful keys you could specify to access the records in the file. For example, you could use the name of the city or the type of work each person does.

## The SYSTEM/BACKUP Utility

---

### LSN Option

The LSN option is used to print files created from a particular logical station number (LSN). This option causes SYSTEM/BACKUP to search the control record of each input file for the LSN. If the LSN in the control record matches the LSN specified by the option, then the file is printed.

The LSN option is used primarily for searching through entire backup tapes or large directories for backup files generated from a specific LSN. The LSN option is of very little use when a single file is specified as the input to SYSTEM/BACKUP. The syntax of the LSN option is

```
— LSN —<integer>—————|
```

The integer identifies a particular LSN.

#### Examples

The following initialization command searches the tape on unit 10 and prints all the backup files generated at LSN 4:

```
WFL PB MT 10 LSN 4
```

Similarly, the following command searches through a directory, called MYDIRECTORY, for backup files generated from LSN 8:

```
U $SYSTEM/BACKUP "MYDIRECTORY" LSN 8
```

### ND Option

The ND option overrides the utility's direct output mode, and causes output to be written to a SYSTEM/BACKUP internal file instead. This file is named BFILE. By using file equation, you can control what happens to the contents of BFILE. Information about file attributes can be found in the *A Series File Attributes Programming Reference Manual* and the *Print System Guide*.

The syntax of the ND option is

```
— ND ——————|
```

When you specify the ND option, observe the following constraints:

- Do not use *D \** as the file specification. Doing so results in an invalid input error.
- Do not use the DEVICE option. An error occurs.
- Do not use the ID option. An error occurs.

You should not use any other SYSTEM/BACKUP options in combination with the ND option, with the exception of the SAVE option. Most of the SYSTEM/BACKUP options have functional counterparts among the file attributes. Use file equation to achieve the desired effect of an option.

The two most common reasons to use the ND option are to copy files from one medium to another (such as tape to disk) and to apply printing-related file attributes to a print request. Copying files is discussed under "Using SYSTEM/BACKUP to Copy Files" later in this section.

To apply printing-related features to a file, you must use file equation. For example, the following initialization command applies the printing-related attributes TRANSFORM and PRINTCOPIES to the sixth file on the tape mounted on unit 15:

```
WFL PB MT 15 FILE 6 ND SAVE; FILE BFILE (PRINTCOPIES=3,  
TRANSFORM="TAPETR IN SITE/TRANSFORMS")
```

You can also use the ND option and file equation to route print requests more quickly. For example, the following command prints all the backup files on the tape mounted on unit 10:

```
WFL PB MT 10
```

However, in many cases printing will be faster if the following command is used instead:

```
WFL PB MT 10 ND; FILE BFILE(PRINTDISPOSITION=DIRECT,  
BUFFERS=10)
```

If you use the ND option when printing more than one file, SYSTEM/BACKUP separates each output file with a blank page by inserting a skip-to-channel-1 between files, regardless of the kind of output device used.

When the ND option is used for a punch file, the input punch file cannot be a binary card file.

## NOINCL Option

The NOINCL option is used in combination with the KEY option to prevent printing of any records with a particular key. The syntax of the NOINCL option is

```
— NOINCL _____|
```

NOINCL is most commonly used to exclude any records generated by the \$INCLUDE compiler control option. The \$INCLUDE option instructs a compiler to include certain records from another source into a listing. When a compiler includes a record in a listing, the system marks it by placing a digit two columns before the start of the sequence number. The KEY option can then be used to determine whether a record has been included in a program by the \$INCLUDE compiler control option, and the NOINCL option prevents those records from being printed.

### Examples

Assume you have a compiler listing, named MYLISTING, that contains included records, and that the sequence numbers for the file are located in columns 75 through 82. Any included records have a digit in column 73. To print this file without the included records, you must use an initialization command similar to the following:



## The SYSTEM/BACKUP Utility

---

```
PB "MYLISTING" NOINCL KEY 73 1 RANGE 0 9
```

In the preceding example statement, the **KEY** option specifies that the key begins in column 73, is one column in length, and is a digit in the range 0 through 9.

Now assume you have a data file, named **CHECKING**, that contains various information about checking accounts. Any accounts that are overdrawn contain the string **OVER** in columns 65 through 68. To print this file without overdrawn accounts, you could use an initialization command similar to the following:

```
WFL PB "CHECKING" NOINCL KEY 65 4 EQUAL "OVER"
```

### RECORD Option

The **RECORD** option prints a particular record or group of records. The syntax of the **RECORD** option is

```
— RECORD —<integer> —————|
                        |<integer>|
```

If you specify only one number for the **RECORD** option, only that record is printed. If you specify two numbers with the **RECORD** option, all the records in that range are printed.

In backup files, the first one or two records usually contain control information. Therefore, the first data record is not the first relative record of the file. However, in most cases you can use a value of 1 as the starting record because the control records are either passed over or blank. If you are not getting the correct range of records printed, you will have to offset the specified range to account for the control records.

#### Examples

To print the sixth record of a file named **MYFILE**, you should use an initialization command similar to the following:

```
?PB "MYFILE" RECORD 6
```

Remember that there might be a one- or two-record offset for any particular file, which is most noticeable when you are trying to print just one record. Using the previous example, if you specify the sixth record and it is not the one you want, try using the values 4, 5, 7, and 8. Once you have determined the offset for a particular file label type, that value is accurate for all other files of that label type.

To print the first 20 records of the file, you should use a command similar to the following:

```
WFL PB "MYFILE" RECORD 1 20
```

Again, you might have to offset the record numbers to account for the control records.

## SAVE Option

The SAVE option prevents backup files from being purged after printing. The syntax of the SAVE option is

```
— SAVE _____|
```

### Example

To print a file named DONTPURGE without removing it from the system, you must use a command similar to the following:

```
U $SYSTEM/BACKUP "DONTPURGE" SAVE
```

## SPACING Option

The SPACING option specifies whether a file is printed single-spaced or double-spaced. However, the option does not affect any text lines with file-designated spacing requirements. The syntax of the SPACING option is

```
— [ SINGLE ] _____|  
   [ DOUBLE ]
```

SINGLE is the default.

### Example

To print a file named TWOSKIP and apply double-spacing, you should use a command similar to the following:

```
WFL PB "TWOSKIP" DOUBLE
```

If TWOSKIP contains any spacing instructions for particular lines, the DOUBLE option is not applied to those lines, and they are printed according to the spacing instructions.

## Using SYSTEM/BACKUP to Copy Files

Besides printing backup files, the SYSTEM/BACKUP utility also can be used to copy backup files. Files can be copied between any of the following media:

- Tape to tape
- Disk to tape
- Tape to disk

You must use the ND option to copy backup files. The ND option overrides the direct output mode of the utility, and causes output to be written to a SYSTEM/BACKUP internal file instead. This file is named BFILE. You can then use file equation to write BFILE on either tape or disk. More information about the ND option can be found earlier in this section.

## The SYSTEM/BACKUP Utility

---

You can use almost any file specification when copying files. However, you cannot use the *D \** specification because using *D \** in conjunction with the ND option results in an invalid input error.

When you copy files from tape to tape, the system searches for any available backup tape. If one is available, the files are written to that tape. If a backup tape is not available, the system searches for a scratch tape. If a scratch tape is available, it is designated as a backup tape and the files are written on it. If there is no tape available at the time, a message is sent to the ODT, and SYSTEM/BACKUP waits until a tape is made available.

### Examples

The following command copies all the files on the tape mounted on unit 10 to another tape:

```
WFL PB MT 10 ND SAVE; FILE BFILE (BACKUPKIND=TAPE)
```

The following command copies the backup files generated by job 5555 to tape:

```
?PB D 5555 ND; FILE BFILE (BACKUPKIND=TAPE)
```

The following command copies the sixth file on the backup tape mounted on unit 15 to disk. The new disk file is named FILE6.

```
U $SYSTEM/BACKUP MT 15 FILE 6 ND;FILE BFILE (BACKUPKIND=DISK,  
USERBACKUPNAME=TRUE,  
TITLE=FILE6)
```

The USERBACKUPNAME file attribute must be set to TRUE for the TITLE attribute to be applied. If USERBACKUPNAME is set to FALSE, the TITLE attribute is ignored and the file is assigned a system-generated file name.

## Using File Equation with SYSTEM/BACKUP

File equation allows you to apply file attributes to a program's internal files. The SYSTEM/BACKUP utility uses four internal files — one for input, one for status information, and two for output:

Internal File Name	Purpose	Notes
FILEIN	Input file	FILEIN can accept either a backup tape file or a backup disk file.
LIN	Status file	LIN contains information such as the level and version of the utility, and the titles of the files printed. If any parameters of the utility contain syntax errors, the information is written to this file.
FILOUT	Output file for direct output when the ND option is not specified	Do not file equate the KIND attribute to this file.

*continued*

*continued*

Internal File Name	Purpose	Notes
BFILE	Output file for indirect output when the ND option is specified	Any file attribute can be equated to BFILE. The BFILE has a MAXRECSIZE of 255 characters, so the BLOCKSIZE can be specified only as a multiple of 255.

You will rarely, if ever, need to file equate to FILEIN, FILOUT, or LIN. You do not have to specify the ND option to file equate to these files.

If you are using SYSTEM/BACKUP in the direct output mode, then you might find it useful to file equate the PRINTERKIND file attribute to FILOUT. However, if this attribute has been specified in the backup file (in the control record), then its value cannot be changed by file equation.

### Examples

The following command copies a disk backup file named MYBACKUP as a disk file named NEWBACK on the family MYFAMILY. The file MYBACKUP is removed after it has been copied.

```
WFL PB "MYBACKUP" ND; FILE BFILE(TITLE=NEWBACK,
FAMILYNAME=MYFAMILY)
```

The following command prints all the \*BD files from job number 4376 on the first available image printer:

```
WFL PB D 4376 ND; FILE FILOUT (PRINTERKIND=IP)
```

Refer to the *Print System Guide* and the *A Series File Attributes Programming Reference Manual* for more information about file attributes and file equation as they relate to printing.

## Repositioning the Input File

If a printer or punch jams or fails while a direct file is being printed, several lines of text can be lost. To print these lost lines, the input file must be repositioned.

Input file repositioning can be used only when SYSTEM/BACKUP is using direct output. You cannot reposition the input file when the ND option has been used.

To reposition a file, you must perform the following steps:

1. Enter the HI (Cause EXCEPTIONEVENT) system command. This command causes all tasks associated with the specified mix number to be exception events.

## The SYSTEM/BACKUP Utility

---

The syntax of the HI command is

```
—<mix number>— HI —————  
                    |  
                    |<integer>|
```

The mix number is the number of your printer job. For more information about the HI command and mix numbers, refer to the *A Series System Commands Operations Reference Manual*.

After the HI command is entered, the SYSTEM/BACKUP utility is stopped and the following message is displayed:

```
ACCEPT: ENTER SKIP OR UNIT NO.
```

2. Respond to the message by entering an AX (Accept) system command. Use the following syntax:

```
—<mix number>— AX —————  
                    |  
                    | LP |<unit number>|  
                    | IP |  
                    | CP |  
→ | + |<skip count>| —————  
  | - |<skip count>| |<channel number>|
```

The *LP* <unit number>, *IP* <unit number>, and *CP* <unit number> options specify the device where the file is being printed. The plus (+) sign or minus (-) sign specifies to scroll the file forward or backward by the number of lines specified by the skip count. If the value of <skip count> positions the tape or disk before the beginning of the file, SYSTEM/BACKUP starts printing from the beginning of the file.

The channel number option allows you to specify a skip to a particular channel number (a value between 1 and 11). When you specify a channel number in addition to a skip count, the SYSTEM/BACKUP utility reads forward or backward until it has found the number of control words specified by the skip count. These words must contain the specified skip-to-channel number.

### Examples

- The following command causes the SYSTEM/BACKUP utility to read forward until it finds four control words containing a skip-to-channel-3. Printing is then resumed from that point on the same unit:

```
4567 AX +4 3
```

- The following command causes a skip forward to channel 1 once, and then resumes printing on LP5:

```
5788 AX LP5 +1 1
```

- The following command skips back to channel 3 four times, and resumes printing on LP15:

```
5784 AX LP 15 -4 3
```

- The following command scrolls the file backward three lines and then resumes printing on LP13:

```
6823 AX LP13 -3
```

## Compile-Time Options

The following compiler control options were set when the SYSTEM/BACKUP utility was initially compiled for your system. You can, however, change the settings of these options by recompiling SYSTEM/BACKUP.

### The \$IDOPTION Option

The SYSTEM/BACKUP utility was initially compiled with the \$IDOPTION option set to FALSE, meaning that a heading is not printed for a file unless the ID option has been specified.

If you recompile the SYSTEM/BACKUP utility with this option set to TRUE, a header page is printed at the beginning of each file. The header page is printed in block characters.

If disk files are being printed, the header contains the title of the file. If tape files are being printed, the header page contains the following:

```
BACKUP TAPE - UNIT nnnnn
```

The unit number of the tape drive where the backup tape is mounted is given by the string nnnnn.

This default heading is overridden if you use the ID option.

### The \$INFOPTION Option

The SYSTEM/BACKUP utility was initially compiled with the \$INFOPTION option set to FALSE. When set to FALSE, this option writes such information as the level and version of the SYSTEM/BACKUP utility, and the title of each file printed, to an internal file named LIN. The LIN file is always printed with SYSTEM/BACKUP output.

If you recompile the utility with \$INFOPTION set to TRUE, this information is displayed at your terminal, and is also included in the WFL job summary.

## Handling Parity Errors

If the system encounters an irrecoverable parity error when running the SYSTEM/BACKUP utility, it displays the following ACCEPT message:

```
ACCEPT: PARITY ERROR - GO OR DS?
```

## The SYSTEM/BACKUP Utility

---

If you respond with GO, the system continues processing the current backup file and flags all lines that contain a parity error. If you respond with DS, the system discontinues processing. If you respond with anything else, the ACCEPT message is repeated.

# Section 4

## The LTTABLEGEN Utility

LTTABLEGEN is a utility program used to create and maintain train tables and translation tables for Unisys printers. Each printer train that Unisys supports has a corresponding train table contained in the SYSTEM/TRAINTABLES file, which is supplied with the bundled software.

The SYSTEM/TRAINTABLES file also contains a number of translation tables. These tables are used on translation printers. A translation table can be used for a variety of purposes, including checking the validity of the input to a printer, or translating the input characters to produce a desired output, such as printing all lowercase characters as uppercase characters.

The LTTABLEGEN utility generates train tables for the 450- and 750-lines-per-minute (lpm) printers and the 1100- and 1500-lpm printers—these two printer types are hereafter referred to as *train printers*. The utility also generates translation tables for all other types of printers, which are hereafter referred to as *translation printers*. Translation printers are available only for Universal Input/Output (UIO) systems.

To obtain the character set required by a specific printer, the operating system indexes the SYSTEM/TRAINTABLES file using either the TRAINID value and the printer speed for train printers, or the PRINTSET ID value for translation printers.

Tables 4-1, 4-2, and 4-3 provide the names and identification numbers of all the default train and translation tables contained in the SYSTEM/TRAINTABLES file.

Table 4-1 shows the default train table names and their corresponding TRAINID values available for slow-speed (450- and 750-lpm) printers.

Table 4-1. Default Train Tables for 450- and 750-lpm Printers

TRAINID Value	Train Table Name
1	EBCDIC18
2	FORTAN48
3	B300B500
4	EBCDIC48
5	EBCDIC72
6	UK3500
7	UK6500
8	LATINPORTUGAL

continued



Table 4-1. Default Train Tables for 450- and 750-lpm Printers (cont.)

TRAINID Value	Train Table Name
9	LATINSPAIN3
10	SWEDENFINLAND3
11	DENMARK
12	BCL64
13	TURKEY
14	SWEDENOCR-B
15	ASCII72
16	EBCDIC96
17	KATAKANA
20	RPG48
21	OCRANUMERIC
22	OCRBNUMERIC
23	OCRAALPHANUMERIC
24	UK
25	GERMANYAUSTRIA
26	ITALY
27	SWEDENFINLAND2
28	LATINSPAIN2
29	ASCII64A
30	BRAZIL
31	DENMARKNORWAY
32	YUGOSLAVIA
33	EBCDIC64A
34	ASCII96A
35	EBCDIC64B
36	ASCII64B
37	ASCII96B

Table 4-2 shows the default train table names and their corresponding TRAINID values for medium-speed (1100- and 1500-lpm) printers.

Table 4-2. Default Train Tables for 1100- and 1500-lpm Printers

TRAINID Value	Train Table Name
1	EBCDIC18
2	FORTAN48
3	B300B500
4	EBCDIC48
5	EBCDIC72
6	UK3500
7	UK6500
8	LATINPORTUGAL
9	LATINSPAIN3
10	SWEDENFINLAND
11	DENMARK
12	BCL64
13	TURKEY
15	ASCII72
16	EBCDIC96
17	KATAKANA
18	ALPHAEBDIC
19	NUMERICEBDIC
20	RPG48
21	OCRANUMERIC
22	OCRBNUMERIC
37	ASCII96B

## The LTTABLEGEN Utility

---

Table 4-3 shows the default translation table names and their corresponding translated PRINTSET ID values. These translation tables can be used on any translation printer.

Table 4-3. Default Translation Table Names

TRAINID Value	Translation Table Name
5	EBCDIC64
8	LATINPORTUGAL
9	LATINSPAIN
16	EBCDIC96
21	OCRANUMERIC
23	OCRAALPHANUMERIC
29	ASCII64A
128	KATAKANACI
129	KATAKANACII
130	KATAKANACIII

The SYSTEM/TRAINABLES file is created on the disk family where the LTTABLEGEN code file resides. You should keep the LTTABLEGEN code file on the halt/load family because this is where the SYSTEM/TRAINABLES file must reside. If the SYSTEM/TRAINABLES file is generated on a different family, then it must be copied to the halt/load family before it can be used.

Because the SYSTEM/TRAINABLES file is supplied, you do not need to run LTTABLEGEN to generate it. Therefore, the need to run this utility should seldom arise. Probably the most likely reason to run this utility would be to create a translation table. However, the function of a translation table often can be accomplished with a transform function. Refer to the *Print System Guide* for information about transform functions.

## Basic Constructs of LTTABLEGEN Commands

The following metatokens are common to both train table commands and translation table commands:

**<blank>**

The hex byte 40, the hex code representation of a blank.

**<charset ID>**

Any positive integer in the range 0 (zero) to 255. The value specifies a particular character set, which is a subset of a printset.

This **< charset ID >** is treated as the TRAINID of a translation table printer. Since the system only recognizes the trainid values listed in Tables 4-1 and 4-2, the use of other values will cause the printer to be marked as using an unknown TRAINID in response to interrogatory system commands such as OL. This does not affect the functionality of the unit, however.

### **< comment character >**

The hex byte 60, the hex code representation of a percent sign (%). Scanning of an input card discontinues when a percent sign is encountered. Comments are usually placed after the percent sign.

### **< hex byte >**

A value in the range 00 to FF, inclusive.

### **< printset ID >**

Any positive integer in the range 0 (zero) to 255. The value is used to identify the physical train, drum, or band installed on a translation printer.

### **< subset ID >**

Any positive integer in the range 0 (zero) to 255. The value specifies a particular subset of a character set.

### **< table name >**

A character string of up to 17 characters. No restrictions apply in regard to embedded blanks or special characters.

### **< TRAINID >**

Any positive integer in the range 0 (zero) to 255. The value is used to identify the physical train installed in a train printer.

## Train Tables and Train Printers

Train printers need a train table to describe the physical organization of the characters on the print train. The train table is a one-to-one mapping of the print train. For instance, if the first character of a print train is an uppercase A, then an uppercase A must occupy the first position in the corresponding train table.

A train printer is a type of line printer that uses a removable or replaceable "train" on which a set of printing characters is arranged. A train typically contains 150 or more character locations. For any given character set (for instance, EBCDIC), many of the most common characters are repeated at one or more locations along the train so that printing can be completed as quickly as possible.

## The LTTABLEGEN Utility

---

On Unisys systems, train printers are the 450- and 750-lpm (slow-speed) and the 1100- and 1500-lpm (medium-speed) printers. The length of the print train differs on these two printer types, causing the speed at which each type operates to be different. Because of the difference in length, the train tables for each printer type are different and cannot be interchanged.

The following subsections describe the commands for creating and defining train tables.

### Train Table Commands

Train table commands are used as LTTABLEGEN input to accomplish the following tasks:

- Create and define train tables in either alphanumeric or hexadecimal characters.
- Print the contents or list the names of train tables in the SYSTEM/TRAINTABLES file.
- Punch the contents of train tables in the SYSTEM/TRAINTABLES file in either alphanumeric or hexadecimal formats, or as a value array.

The train table commands are grouped by function and listed alphabetically as follows:

<b>Create and Define Commands</b>	<b>Print Commands</b>	<b>Punch Commands</b>
ALPHATABLE	PRINTALL	PUNCHALPHA
CHARSETE	PRINTNAMES	PUNCHARRAY
DEFAULT	PRINTRAINID	PUNCHHEX
HEXTABLE		
INVCHR		
MEDIUM		
NAME		
SLOW		
MEDIUM		

The train table commands, their functions, and syntax are discussed on the following pages. Examples of how to use the commands can be found in "Creating and Using Train Tables" in this section. Commands can be split between records, provided that you do not split individual identifiers, numbers, or strings.

### Creating and Defining Train Tables

Use the commands described in the following subsections to create and define train tables.

### ALPHATABLE Command

The ALPHATABLE command defines a train table or a character set in alphanumeric characters. The syntax of the ALPHATABLE command is

```
ALPHATABLE <character string>
```

The character string describes the physical layout of characters on the print train. The string must include every character on the train, and every character must be in the correct position relative to the other characters on the train. The character string must be concluded with the word END.

To represent a blank character on the train, a blank character must be included in the character string. The blank character may not be the first or the last string character on an input line, however, because preceding and trailing blanks (and blank lines) are ignored.

The ALPHATABLE command must be the last command in the LTTABLEGEN input. (Note that you can use either the ALPHATABLE or HEXTABLE command to define a train table.)

### CHARSET Command

The CHARSET command defines the size of a character set — the number of characters on the print train, and, correspondingly, the number of characters in the train table. The syntax of the CHARSET command is

```
CHARSET <integer>
```

The value of the integer can be in the range 0 (zero) through 999.

### DEFAULT Command

The DEFAULT command is combined with the NAME command to designate the name of the train table to be used as the default. The syntax of the DEFAULT command is

```
DEFAULT NAME <train table name>
```

The train table specified by the DEFAULT command is loaded into segment 0 (zero) of the SYSTEM/TRAINABLES file (that is, its TRAINID value is 0), and is used by train printers that do not return a TRAINID. If the DEFAULT command is not used to specify a default train table, then the system uses the EBCDIC64B train table as the default.

### HEXTABLE Command

The HEXTABLE command defines a train table or a character set in hexadecimal (hex) characters. The syntax of the HEXTABLE command is

```
HEXTABLE <hex character string>
```

## The LTTABLEGEN Utility

---

The hex character string describes the physical layout of hex characters on the print train. The string must include every character on the train, and every character must be in the correct position relative to the other characters on the train. The hex character string must be concluded with a semicolon (;). Embedded blanks between characters, blank lines, and preceding and trailing blanks on input lines are allowed and are ignored.

The HEXTABLE command must be the last command in the LTTABLEGEN input. (Note that you can use either the HEXTABLE or ALPHATABLE command to define a train table.)

### INVCHR Command

The INVCHR command specifies the character to be printed if a character sent to the printer is not contained on the print train. The syntax of the INVCHR command is

```
INVCHR <alphanumeric character>
```

The specified alphanumeric character is printed whenever the printer receives a request to print a character that is not on the current print train. The default value is a question mark (?).

### MEDIUM Command

The MEDIUM command specifies that a train table or character set is to be associated with the 1100- and 1500-lpm train printers, which are the default train printers for train tables that do not allow the 450- and 750-lpm train printers. The syntax of the MEDIUM command is

```
MEDIUM
```

You must specify this command in order to associate the train table with the 1100/1500 lpm printers.

### NAME Command

The NAME command assigns a name to a train table. The syntax of the NAME command is

```
NAME <alphanumeric string>
```

The alphanumeric string can be up to 17 characters long.

### SLOW Command

The SLOW command specifies that a train table or character set is to be associated with the 450/750 lpm train printers, which are the default train printers unless they are not allowed for that train table. The syntax of the SLOW command is

```
SLOW
```

If you define a train table and do not specify which printer type it is for, SLOW is used by default. The table is thus designated for the 450- and 750-lpm train printers.

### TRAINID Command

The TRAINID command assigns an identification number to a train table. The syntax of the TRAINID command is

```
TRAINID <integer>
```

The value of the integer can be in the range 0 (zero) through 255.

### Printing the Names or Contents of Train Tables

Use the commands described in the following subsections to print the names or contents of train tables.

### PRINTALL Command

The PRINTALL command prints every train table in the SYSTEM/TRAINTABLES file in both alphanumeric and hexadecimal characters. The syntax of the PRINTALL command is

```
PRINTALL
```

This command causes the SYSTEM/TRAINTABLES file to be removed and regenerated. When the file is regenerated, any nonstandard (custom) train tables are lost. To prevent the loss of your nonstandard tables, copy and save the SYSTEM/TRAINTABLES file under another name before using the PRINTALL command.

The following information is printed preceding each train table:

- The TRAINID value
- The train table length (total number of characters)
- The number of characters in the character set
- The train table name
- The printer type (slow or medium) to which the train table belongs

### PRINTNAMES Command

The PRINTNAMES command prints the names and the corresponding TRAINID values of every train table in the SYSTEM/TRAINTABLES file. The syntax of the PRINTNAMES command is

```
PRINTNAMES
```



## The LTTABLEGEN Utility

---

This command causes the SYSTEM/TRAINABLES file to be removed and regenerated. Only the default (standard) train tables are regenerated. Any user-defined train tables are lost.

To avoid losing any nonstandard train tables, you should always copy and save the SYSTEM/TRAINABLES file under another file name before using the PRINTNAMES command.

### PRINTRAINID Command

The PRINTRAINID command prints the specified train table. The syntax of the PRINTRAINID command is

```
PRINTRAINID <trainid>
```

The TRAINID value is an integer in the range 0 (zero) through 255. The specified train table is printed in both its hexadecimal and alphanumeric representations. Preceding the table data are the TRAINID, the table length, the train table name, the printer type (slow or medium) to which the table belongs, and the number of characters in the character set.

### Punching the Contents of Train Tables

Use the commands described in the following subsections to punch the contents of train tables.

#### PUNCHALPHA Command

The PUNCHALPHA command punches a card deck of the specified train table in alphanumeric form. The syntax of the PUNCHALPHA command is

```
PUNCHALPHA <trainid>
```

The TRAINID value is an integer in the range 0 (zero) through 255. For example, the command PUNCHALPHA 42 causes the train table identified by the TRAINID value of 42 to be punched in alphanumeric characters.

#### PUNCHARRAY Command

The PUNCHARRAY command punches a value array of the specified train table. The syntax of the PUNCHARRAY command is

```
PUNCHARRAY <trainid>
```

The TRAINID value is an integer in the range 0 (zero) through 255.

A value array can be used by other programs (mostly system programs such as SYSTEM/LOADER) so they can communicate with a printer. In the case of

SYSTEM/LOADER, it may be desirable to print dump information. To do so, the LOADER program must have the train table information because it does not have access to the operating system.

### PUNCHHEX Command

The PUNCHHEX command punches a card deck of the specified train table in hexadecimal form. The syntax of the PUNCHHEX command is

```
PUNCHHEX <trainid>
```

The TRAINID value is an integer in the range 0 (zero) through 255. For example, the command PUNCHHEX 42 causes the train table identified by the TRAINID value of 42 to be punched in hexadecimal characters. The deck generated by the PUNCHHEX command can be altered by adding or removing cards, or by retyping a card or cards to change the hex digits.

## Defining and Using Train Tables

With the train table commands, you can define an entire train table, or just character sets (subsets) of the entire train.

A train table or character set can be defined in either alphanumeric or hexadecimal symbols. If the table or character set is defined in alphanumeric symbols, then you are limited to the symbols represented on the keyboard you are using. By using a hexadecimal definition, you are not limited to the symbols on your keyboard. For example, the ASCII character set contains many characters that are not represented on most keyboards.

The following examples show some of the various ways to use the train table commands. Note that in each case, the example WFL job would remove the existing SYSTEM/TRAINABLES file and create a new one. Also, in each case, since a DEFAULT train table is not specified, the new SYSTEM/TRAINABLES file would use EBCDIC64B as the new default.

### Examples

- The following WFL job removes the existing SYSTEM/TRAINABLES file and creates a new one using EBCDIC64B as the new default:

```
BEGIN JOB;  
  RUN SYSTEM/LTTABLEGEN; DATA CARD  
  PRINTALL PRINTNAMES  
  PUNCHARRAY 5  
END JOB
```

## The LTTABLEGEN Utility

---

This example also prints every train table in the SYSTEM/TRAINABLES file in both alphanumeric and hexadecimal characters, and prints the names and corresponding TRAINID values for all of the trains in the file.

The PUNCHARRAY 5 command punches the train table corresponding to TRAINID 5 onto cards as a value array declaration that can be used by other programs.

- The following WFL job also replaces the existing SYSTEM/TRAINABLES file with a new one using EBCDIC64B as the new default:

```
BEGIN JOB;
  RUN SYSTEM/LTTABLEGEN; DATA CARD
    TRAINID 40 CHARSET 48
    NAME A48CHARACTERSET
    ALPHATABLE
    235689$@4"71.*,-?MNOPQRBCDGHIVY) : (+AEFKLJSTUWXZØEND
  END JOB
```

In addition, this example defines a new train table named A48CHARACTERSET in the new SYSTEM/TRAINABLES file. Its TRAINID value will be 40, and its set size will be 48. Since a speed is not specified, by default, the new train table will be associated with 450- and 750-lpm printers.

- Similar to the previous two examples, the following WFL job replaces the existing SYSTEM/TRAINABLES file with a new one using EBCDIC64B as the new default:

```
BEGIN JOB;
  RUN SYSTEM/LTTABLEGEN; DATA CARD
    TRAINID 38 MEDIUM
    CHARSET 72 NAME USER/TABLE
    HEXTABLE
    FØF1F2F3F4F5F6F7F8F94B5B5C6Ø6B6F4EC1C2C3C4C5C6C7C8C9
    D1D2D3D4D5D6D7D8D9E2E3E4E5E6E7E8E94D5D7A7C7F4A4C5Ø5A
    5E616C6E7B7EEØ4F6DDØ7D5F;
  END JOB
```

This example, however, defines a new train table named USER/TABLE in the new SYSTEM/TRAINABLES file. Its TRAINID value will be 38, and its character set size will be 72. Because of the MEDIUM speed specification, the new train table will be explicitly associated with 1100- and 1500-lpm printers.

## Translation Tables and Translation Printers

Translation printers use data link processor (DLP) controllers, and are of a later design than train printers. The printer DLP contains, internally, an artificial feature that is referred to as a translation table. When data is sent to a translation printer, the DLP indexes the translation table. Whatever character is indexed in the table is the character that gets printed. In this way, character translation is handled automatically and without processor overhead.

A translation table is a 16- by 16-element matrix, where each element is one byte in length and represents one character. A translation table is loaded into the printer's DLP,

whereupon the DLP uses each subsequent write data character as an index into the table whenever a WRITE command is issued by the operating system.

The DLP performs table indexing by using the first digit of the write data character as the row index and the second digit of the character as the column index. For example, if the write data character is 82, the DLP prints the character found at row 8, column 2 of the current translation table.

The image a printer generates is determined by the symbols that comprise the internal mechanism (a train, band, or drum). However, for translation printers, the mapping of write data characters to translation table indices can be controlled. For instance, you might not have a printer that can print lowercase characters, but you can still process and output data that contains lowercase characters by writing a translation table that maps lowercase characters to uppercase characters.

A translation table does not describe the physical layout of a train, drum, or band. It simply maps characters in an output file to the characters on a printer. Data to be printed on a translation printer should always be processed through a translation table, even if the table does nothing other than mimic the file's character set. Using such a table is useful for detecting errors and does not affect system performance.

Translation table commands can be used as input to LTTABLEGEN only on A Series systems using translation printers. Translation printers are all printers except the 450- and 750-lpm and the 1100- and 1500-lpm train printers.

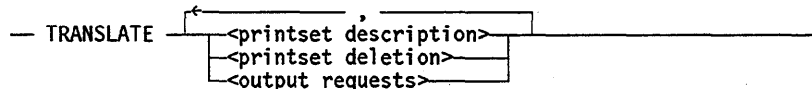
## Translation Table Commands

The translation table commands can be used as LTTABLEGEN input to accomplish the following tasks:

- Describe a printset.
- Delete a printset.
- Print the contents of translation tables, along with other information about each table.

The TRANSLATE command is the main command for specifying translation table input to LTTABLEGEN. All other translation table specifications are part of the TRANSLATE command. The syntax of the TRANSLATE command is

<translation table commands>



**Note:** The keyword TRANSLATE must be specified first, and it must be entered on a separate line from any commands that follow.

The following example shows the previous restriction:

## The LTTABLEGEN Utility

---

TRANSLATE  
<All other translation table commands>

Other than the TRANSLATE command, the remaining translation table commands can be entered in free format. For clarity, however, you should enter one command on each line. The examples given in the remainder of this section reflect this structure.

The three parts of the TRANSLATE command are described in the following subsections.

### Describing a Printset

A printset is described and assigned a number and a name using the PRINTSET command. Additionally, the following tasks can be accomplished:

- Changing the name of a printset, character set, or subset
- Creating character sets and subsets for a particular printset
- Deleting character sets and subsets from a particular printset

A printset is the set of all graphic characters found on the physical printing medium, whether it be a train, drum, or band. For translation printers, a translation table is used to map the printset of a train, drum, or band to the character set used by the computer.

The syntax of the PRINTSET command is

```
<printset description>  
— PRINTSET —<printset ID>—————→  
→ NAME —<set name>—<charset specs>—|  
  |<charset specs>—————→
```

The PRINTSET ID value is an integer in the range 0 (zero) to 255. A set name is an alphanumeric character string of up to 17 characters. The <charset specs> attribute can be either a character set deletion statement, a character set description statement, or both. Up to 12 occurrences of each statement can be included in a <charset specs> clause.

### Deleting a Printset

A printset can be deleted from the SYSTEM/TRAINABLES file by using the DELETE PRINTSET command. To delete a printset, you enter the command keywords DELETE PRINTSET followed by the PRINTSET ID value.

The syntax of the DELETE PRINTSET command is

```
DELETE PRINTSET <printset ID>
```

For example, assume there is a translation table with a PRINTSET ID value of 99. To delete that translation table you would enter the following command:

```
DELETE PRINTSET 99
```

An optional form of the DELETE PRINTSET command is to include the name of the translation table after the PRINTSET ID value. The syntax for this form of the command is

```
DELETE PRINTSET <printset ID> NAME <table name>
```

For example, assume that the name of the translation table having the PRINTSET ID value of 99 is MYEBCDIC. To delete that translation table you would enter the following command:

```
DELETE PRINTSET 99 NAME MYEBCDIC
```

### Printing Translation Tables and Related Information

You can print listings of all or various parts of the SYSTEM/TRAINTABLES file by using the following commands:

- PRINTID
- PRINTNAMES
- PRINTALL

#### PRINTID Command

The PRINTID command prints information about the specified printset. The syntax of the PRINTID command is

```
PRINTID <printset ID>
```

The following information is returned by the PRINTID command:

- The PRINTSET ID value and the name of the printset
- The numbers and names of all the character sets belonging to the specified printset
- The numbers and names of all the subsets belonging to each character set of the specified printset
- The translation table in both hexadecimal and alphanumeric form, including the specification of the pad character

Note that the printed output of the translation table is determined by the train, drum, or band being used on the printer where the listing is produced. Consequently, the alphanumeric images are accurate only if the device for which the translation table was generated is used to produce the listing.

## The LTTABLEGEN Utility

---

### PRINTNAMES Command

The PRINTNAMES command prints a listing of all the names and ID numbers for every printset, character set, and subset in the SYSTEM/TRAINABLES file. There are no parameters for this command, so the syntax is simply

```
PRINTNAMES
```

Note that the PRINTNAMES command does not print the translation tables themselves or the pad character. (See the PRINTALL and PRINTID command descriptions.)

### PRINTALL Command

The PRINTALL command generates a listing of all the translation tables contained in the SYSTEM/TRAINABLES file. The listing contains the following information:

- The PRINTSET ID value and the name of the printset
- The numbers and names of all the character sets belonging to the specified printset
- The numbers and names of all the subsets belonging to each character set of the specified printset
- The translation table in both hexadecimal and alphanumeric form, including the specification of the pad character

The PRINTALL command does not have any parameters, so the syntax is simply

```
PRINTALL
```

## Creating and Using Translation Tables

Translation tables are organized by printset, charset, and subset. A printset is the set of all graphic characters found on a physical printing medium; for example, a train, drum, or band. A charset is a set of characters all of whose elements can be found in a given printset; thus, a charset is a subset of a printset. A subset is a set of characters all of whose elements can be found in a given charset. Thus a subset is a subset of a charset and a sub-subset of a printset.

For example, the following table shows the relationship between the KATAKANA2 printset and the charsets and subsets that compose it:

Printset	Charsets	Subsets
KATAKANA2	KATAKANA	KATAKANA
		DEFAULT
	EBCDIC64	EBCDIC64
		EBCDIC48
		EBCDIC18

The preceding table depicts the KATAKANA2 printset and its associated character sets and subsets. If the KATAKANA2 print train is mounted on a printer, then both KATAKANA and EBCDIC characters can be printed (those are the character sets the print train contains). The KATAKANA charset can print either a KATAKANA or DEFAULT subset; the EBCDIC64 charset can print an EBCDIC64, EBCDIC48, or EBCDIC18 subset.

Using charsets and subsets allows you to define specific groups of characters to use for formatting output from several available groups defined for a given train, band, or drum. When specifying the character set to use for printing, the printset, charset, and subset names must be given; otherwise a system-generated default is used.

## Running the LTTABLEGEN Utility

The LTTABLEGEN utility can be initiated from a Work Flow Language (WFL) job or from an operator display terminal (ODT). Initiating LTTABLEGEN without any input data causes the utility to generate the SYSTEM/TRAINABLES file. This file contains all the default train and translation tables.

```
BEGIN MY/JOB;  
  RUN SYSTEM/LTTABLEGEN;  
END MY/JOB
```

You can specify input to LTTABLEGEN in a WFL job in two ways:

- You can provide the input as a data card, as in the following example:

```
BEGIN MY/JOB;  
  RUN SYSTEM/LTTABLEGEN;  
    DATA CARD  
      <LTTABLEGEN input statements>  
END MY/JOB
```

- You can provide the LTTABLEGEN input as a file card; that is, the input is specified in an external file that is called by the WFL job. The following example illustrates this method:

```
BEGIN MY/JOB;  
  RUN SYSTEM/LTTABLEGEN;  
    FILE CARD (KIND = DISK, TITLE = <file title>);  
END MY/JOB
```

The BEGIN and END statements in a WFL job might need to be preceded by an invalid character. If LTTABLEGEN is run from an ODT or a remote terminal, the use of an invalid character (in this case, a question mark) is optional. When run from a card reader, BEGIN and END must be preceded by any invalid punch.



## The LTTABLEGEN Utility

---

LTTABLEGEN is controlled by entering commands through a card reader, a remote terminal, or an ODT. Normally, you enter a table into the system, then translate that table as it is needed. The following steps explain how to specify a train table for use by a train printer:

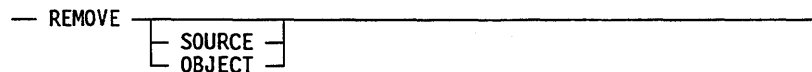
1. Specify the name of the train table. The name can be specified in alphanumeric or hexadecimal characters. The name of the default train table also can be specified.
2. Specify the number of characters in the character set and the invalid character.
3. Specify the invalid character.
4. Select the printer.
5. Print all the train tables in hexadecimal and alphanumeric characters. Included is information about each train table's ID, the number of characters in the character set, table length, train table name, and speed of the printer to which the table belongs. You can also print specific information only.
6. Create a value array that can be used by other programs to produce train tables.

# Appendix A

## Understanding Railroad Diagrams

### What Are Railroad Diagrams?

Railroad diagrams are diagrams that show you the rules for putting words and symbols together into commands and statements that the computer can understand. These diagrams consist of a series of paths that show the allowable structure, constants, and variables for a command or a statement. Paths show the order in which the command or statement is constructed. Paths are represented by horizontal and vertical lines. Many railroad diagrams have a number of different paths you can take to get to the end of the diagram. For example:



If you follow this railroad diagram from left to right, you will discover three acceptable commands. These commands are

- REMOVE
- REMOVE SOURCE
- REMOVE OBJECT

If all railroad diagrams were this simple, this explanation could end here. However, because the allowed ways of communicating with the computer can be complex, railroad diagrams sometimes must also be complex.

Regardless of the level of complexity, all railroad diagrams are visual representations of commands and statements. Railroad diagrams are intended to

- Show the mandatory items.
- Show the user-selected items.
- Present the order in which the items must appear.
- Show the number of times an item can be repeated.
- Show the necessary punctuation.

To familiarize you with railroad diagrams, this explanation describes the elements of the diagrams and provides examples.

Some of the actual railroad diagrams you will encounter might be more complex. However, all railroad diagrams, simple or complex, follow the same basic rules. They

## Understanding Railroad Diagrams

---

all consist of paths that represent the allowable structure, constants, and variables for commands and statements.

By following railroad diagrams, you can easily understand the correct syntax for commands and statements. Once you become proficient in the use of railroad notation, the diagrams serve as quick references to the commands and statements.

### Constants and Variables

A constant is an item that cannot be altered. You must enter the constant as it appears in the diagram, either in full or as an allowable abbreviation. If a constant is partially underlined, you can abbreviate the constant by entering only the underlined letters. In addition to the underlined letters, any of the remaining letters can be entered. If no part of the constant is underlined, the constant cannot be abbreviated. Constants can be recognized by the fact that they are never enclosed in angle brackets (< >) and are in uppercase letters.

A variable is an item that represents data. You can replace the variable with data that meets the requirements of the particular command or statement. When replacing a variable with data, you must follow the rules defined for the particular command or statement. Variables appear in railroad diagrams enclosed in angle brackets.

In the following example, BEGIN and END are constants while <statement list> is a variable. The constant BEGIN can be abbreviated since it is partially underlined. Valid abbreviations for BEGIN are BE, BEG, and BEGI.

— BEGIN —<statement list>— END —————|

### Constraints

Constraints are used in a railroad diagram to control progression through the diagram. Constraints consist of symbols and unique railroad diagram line paths. They include

- Vertical bars
- Percent signs
- Right arrows
- Required items
- User-selected items
- Loops
- Bridges

A description of each item follows.

#### Vertical Bar

The vertical bar symbol (|) represents the end of a railroad diagram and indicates the command or statement can be followed by another command or statement.

— SECONDWORD — ( —<arithmetic expression>— ) —————|

## Percent Sign

The percent sign (%) represents the end of a railroad diagram and indicates the command or statement must be on a line by itself.

— STOP —————| %

## Right Arrow

The right arrow symbol (>) is used when the railroad diagram is too long to fit on one line and must continue on the next. A right arrow appears at the end of the first line, and another right arrow appears at the beginning of the next line.

— SCALERIGHT — ( —<arithmetic expression>— , —————>  
><arithmetic expression>— ) —————|

## Required Items

A required item can be either a constant, a variable, or punctuation. A required item appears as a single entry, by itself or with other items, on a horizontal line. Required items can also exist on horizontal lines within alternate paths or nested (lower-level) diagrams. If the path you are following contains a required item, you must enter the item in the command or statement; the required item cannot be omitted.

In the following example, the word EVENT is a required constant and <identifier> is a required variable:

— EVENT —<identifier>—————|

## User-Selected Items

User-selected items appear one below the other in a vertical list. You can choose any one of the items from the list. If the list also contains an empty path (solid line), none of the choices are required. A user-selected item can be either a constant, a variable, or punctuation. In the following railroad diagram, either the plus sign (+) or the minus sign (-) can be entered before the required variable <arithmetic expression>, or the symbols can be disregarded because the diagram also contains an empty path.

— [ + ] [ - ] [ ] <arithmetic expression>—————|

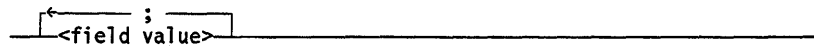
## Understanding Railroad Diagrams

---

### Loop

A loop represents an item or group of items that you can repeat. A loop can span all or part of a railroad diagram. It always consists of at least two horizontal lines, one below the other, connected on both sides by vertical lines. The top line is a right-to-left path that contains information about repeating the loop.

Some loops include a return character. A return character is a character—often a comma (,) or semicolon (;)—required before each repetition of a loop. If there is no return character, the items must be separated by one or more blank spaces.

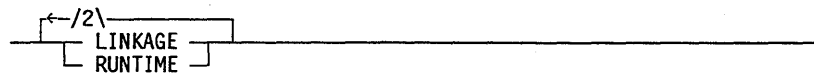
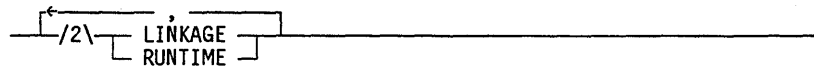


### Bridge

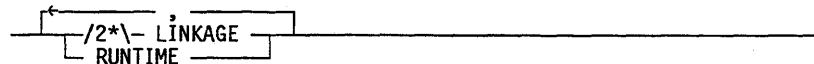
Sometimes a loop also includes a bridge, which is used to show the maximum number of times the loop can be repeated. The bridge can precede the contents of the loop, or it can precede the return character (if any) on the upper line of the loop.

The bridge determines the number of times you can cross that point in the diagram. The bridge is an integer enclosed in sloping lines (/ \). Not all loops have bridges. Those that do not can be repeated any number of times until all valid entries have been used.

In the first bridge example, you can enter LINKAGE or RUNTIME no more than two times. In the second bridge example, you can enter LINKAGE or RUNTIME no more than three times.



In some bridges an asterisk (\*) follows the number. The asterisk means that you must cross that point in the diagram at least once. The maximum number of times that you can cross that point is indicated by the number in the bridge.



In the previous bridge example, you must enter LINKAGE at least once but no more than twice, and you can enter RUNTIME any number of times.

The following figure shows the types of constraints used in railroad diagrams.

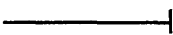
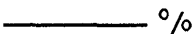


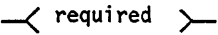
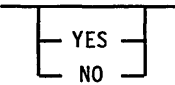
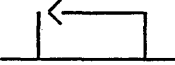
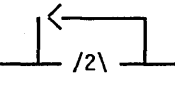
SYMBOL/PATH	EXPLANATION
	Vertical bar. Indicates that the command or statement can be followed by another command or statement.
	Percent sign. Indicates that the command or statement must be on a line by itself.
 	Right arrow. Indicates that the diagram occupies more than one line.
	Required items. Indicates the constants, variables, and punctuation that must be entered in a command or statement.
	User-selected items. Indicates the items that appear one below the other in a vertical list. You select which item or items to include.
	A loop. Indicates an item or group of items that can be repeated.
	A bridge. Indicates the maximum number of times a loop can be repeated.

Figure A-1. Railroad Constraints

## Following the Paths of a Railroad Diagram

The paths of a railroad diagram lead you through the command or statement from beginning to end. Some railroad diagrams have only one path, while others have several alternate paths. The following railroad diagram indicates there is only one path that requires the constant LINKAGE and the variable <linkage mnemonic>:

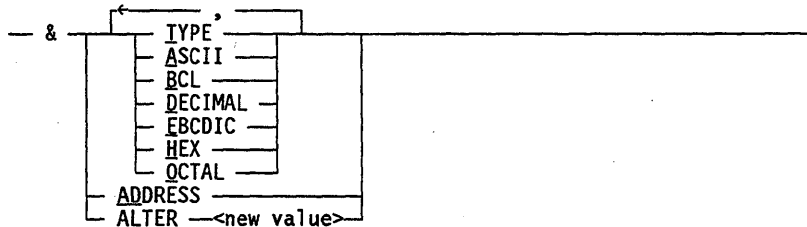
— LINKAGE —<linkage mnemonic>—————|

Alternate paths provide choices in the construction of commands and statements. Alternate paths are provided by loops, user-selected items, or a combination of both. More complex railroad diagrams can consist of many alternate paths, or nested (lower-level) diagrams, that show a further level of detail.

For example, the following railroad diagram consists of a top path and two alternate paths. The top path includes an ampersand (&) and the constants (that are

## Understanding Railroad Diagrams

user-selected items) in the vertical list. These constants are within a loop that can be repeated any number of times until all options have been selected. The first alternate path requires the ampersand and the required constant ADDRESS. The second alternate path requires the ampersand followed by the required constant ALTER and the required variable <new value>.

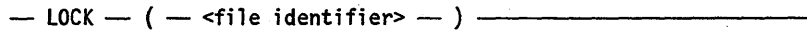


## Railroad Diagram Examples with Sample Input

The following examples show five railroad diagrams and possible command and statement constructions based on the paths of these diagrams.

### Example 1

<lock statement>



#### Sample Input

LOCK (FILE4)

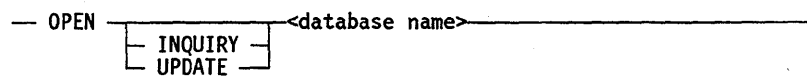
#### Explanation

LOCK is a constant and cannot be altered. Because no part of the word is underlined, the entire word must be entered.

The parentheses are required punctuation, and FILE4 is a sample file identifier.

### Example 2

<open statement>



#### Sample Input

OPEN DATABASE1

#### Explanation

The constant OPEN is followed by the variable DATABASE1, which is a database name.

The railroad diagram shows two user-selected items, INQUIRY and UPDATE. However, because there is an empty path (solid line), these entries are not required.

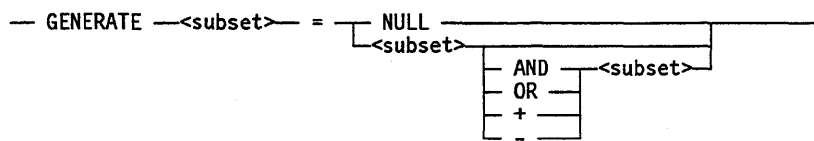
*continued*

*continued*

Sample Input	Explanation
OPEN INQUIRY DATABASE1	The constant OPEN is followed by the user-selected constant INQUIRY and the variable DATABASE1.
OPEN UPDATE DATABASE1	The constant OPEN is followed by the user-selected constant UPDATE and the variable DATABASE1.

### Example 3

<generate statement>



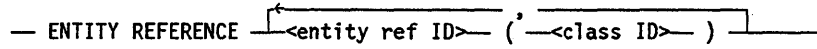
Sample Input	Explanation
GENERATE Z = NULL	The GENERATE constant is followed by the variable Z, an equal sign (=), and the user-selected constant NULL.
GENERATE Z = X	The GENERATE constant is followed by the variable Z, an equal sign, and the user-selected variable X.
GENERATE Z = X AND B	The GENERATE constant is followed by the variable Z, an equal sign, the user-selected variable X, the AND command (from the list of user-selected items in the nested path), and a third variable, B.
GENERATE Z = X + B	The GENERATE constant is followed by the variable Z, an equal sign, the user-selected variable X, the plus sign (from the list of user-selected items in the nested path), and a third variable, B.



# Understanding Railroad Diagrams

## Example 4

<entity reference declaration>



### Sample Input

ENTITY REFERENCE ADVISOR1 (INSTRUCTOR)

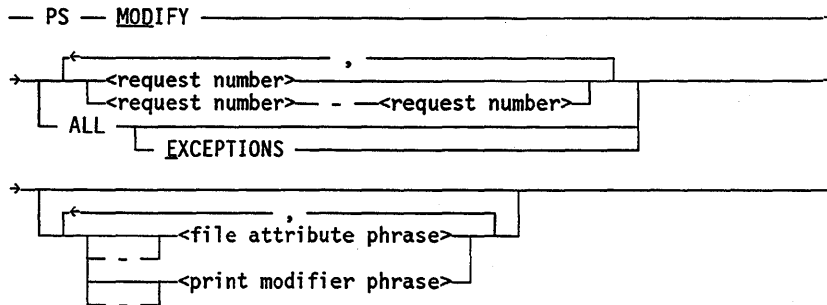
### Explanation

The required item ENTITY REFERENCE is followed by the variable ADVISOR1 and the variable INSTRUCTOR. The parentheses are required.

ENTITY REFERENCE ADVISOR1 (INSTRUCTOR),  
ADVISOR2 (ASST\_INSTRUCTOR)

Because the diagram contains a loop, the pair of variables can be repeated any number of times.

## Example 5



### Sample Input

PS MODIFY 11159

### Explanation

The constants PS and MODIFY are followed by the variable 11159, which is a request number.

PS MODIFY 11159,11160,11163

Because the diagram contains a loop, the variable 11159 can be followed by a comma, the variable 11160, another comma, and the final variable 11163.

PS MOD 11159-11161 DESTINATION =  
"LP7"

The constants PS and MODIFY are followed by the user-selected variables 11159-11161, which are request numbers, and the user-selected variable DESTINATION = "LP7", which is a file attribute phrase. Note that the constant MODIFY has been abbreviated to its minimum allowable form.

PS MOD ALL EXCEPTIONS

The constants PS and MODIFY are followed by the user-selected constants ALL and EXCEPTIONS.

# Glossary

## A

### ALGOL

Algorithmic language. A structured, high-level programming language that provides the basis for the stack architecture of the Unisys A Series systems. ALGOL was the first block-structured language developed in the 1960s and served as a basis for such languages as Pascal and Ada. It is still used extensively on A Series systems, primarily for systems programming.

### ASCII

American Standard Code for Information Interchange. A standard 7-bit or 8-bit information code used to represent alphanumeric characters, control characters, and graphic characters on a computer system.

### attribute

(1) A characteristic or property. (2) The information that describes a characteristic of an entity.

## B

### backup file

(1) A printer or punch file assigned to a backup peripheral for subsequent output. The default backup peripheral is a disk. (2) A copy of a file that is stored offline so that it can be copied back onto the system if the original file becomes corrupted or inaccessible.

### backup file name

The name of a backup file.

### Backup Processor

A Print System utility used to view, copy, remove, or print printer backup files.

## C

### CANDE

*See* Command and Edit.

### card punch

A device that punches holes in cards to represent data.

### checkpoint

(1) A place in a program where the program is to be stopped so that its current state can be written to disk. After the state of the program has been recorded, program execution resumes where it left off. If the system halts unexpectedly before the program finishes, the program can be restarted at its most recent checkpoint instead of at the beginning.

## Glossary

---

### **COBOL**

Common Business-Oriented Language. A widely used, procedure-oriented language intended for use in solving problems in business data processing. The main characteristics of COBOL are the easy readability of programs and a considerable degree of machine independence. COBOL is the most widely used procedure-oriented language.

### **Command and Edit (CANDE)**

A time-sharing message control system (MCS) that enables a user to create and edit files, and to develop, test, and execute programs, interactively.

### **Communications Management System (COMS)**

A general message control system (MCS) that controls online environments on A Series systems. COMS can support the processing of multiprogram transactions, single-station remote files, and multistation remote files.

### **COMS**

See Communications Management System.

### **crunch**

To return the unused portion of the last row of disk space of a disk file (beyond the end-of-file indicator) to the system. Only disk files can be crunched.

## **D**

### **data link processor (DLP)**

A processor that serves as the system interface to a specific peripheral device, controller, or communications network.

### **default printer pool**

A group of printers used for print requests with unspecified destinations.

### **direct printer file**

A file that is routed directly to the printer.

### **DLP**

See data link processor.

## **E**

### **EBCDIC**

Extended Binary Coded Decimal Interchange Code. An 8-bit code representing 256 graphic and control characters that are the native character set of most mainframe systems.

### **exception**

Anything unusual that happens to a print request and prevents the request from being carried out.

## F

### **file attribute**

An element that describes a characteristic of a file and provides information the system needs to handle the file. Examples of file attributes are the file title, record size, number of areas, and date of creation. For disk files, permanent file attribute values are stored in the disk file header.

### **file equation**

A mechanism for specifying the values of file attributes when a program is compiled or executed. A file equation implicitly assigns a value to the FILECARDS task attribute.

### **FORTRAN**

Formula Translation. A high-level, structured programming language intended primarily for scientific use.

## I

### **I/O**

Input/output. An operation in which the system reads data from or writes data to a file on a peripheral device such as a disk drive.

### **image printer**

A printer that prints a page of images at one time, as opposed to a line printer, which prints a line of images at one time.

### **interactive mode**

An execution mode in which each command or item of data is validated and executed at the time it is entered at a terminal or workstation, allowing the user to see immediate results and correct errors as they are made.

## J

### **job**

A group of one or more tasks under the control of a single Work Flow Language (WFL) program. The system assigns each job a mix number and treats each job as a discrete unit of work.

### **job summary**

A file, produced after a job completes execution, that lists information such as the tasks initiated by the job, the beginning and ending times for each task, and the termination information for each task.

## K

### **key**

A field used to locate or identify a record in an indexed file.

## Glossary

---

### **KIND**

The file attribute that indicates the type of device on which the file is stored.

## **L**

### **library**

A collection of one or more named routines or library objects that are stored in a file and can be accessed by other programs.

### **line printer (LP)**

A peripheral device that prints all characters of a line as a unit.

### **lines per minute (lpm)**

A measure of the printing speed of a printer.

### **logical file**

A file variable declared in a program, which represents the file and its structure to the program. A logical file has no properties of its own until it is described by file attributes or associated with a physical file.

### **logical station number (LSN)**

(1) In Network Definition Language II (NDLII), a unique number assigned to each station in a network. Each station has an LSN assigned according to the order in which the stations are defined in NDLII. The first defined station is 0000. (2) In the Interactive Datacomm Configurator (IDC), a unique number assigned to each station structure. When IDC creates the DATACOMINFO file from the network information file II (NIFII), it assigns an LSN to each structure sequentially, beginning with the number 2. The numbers allocated by IDC are the same as those used by the Master Control Program (MCP) to identify a station.

### **LP**

*See* line printer.

### **lpm**

*See* lines per minute.

### **LSN**

*See* logical station number.

## **M**

### **MARC**

*See* Menu-Assisted Resource Control.

### **medium**

A device used to store data. Examples are a disk pack or a magnetic tape (MT).

### **Menu-Assisted Resource Control (MARC)**

A menu-driven interface to A Series systems that also enables direct entry of commands.

**metatoken**

An item that appears in syntax notation as a variable item.

**N**

**NEWP**

A structured, high-level programming language used in developing some Unisys system software. Based on the ALGOL language, NEWP contains facilities necessary for the operating system to interact with A Series hardware.

**O**

**ODT**

See operator display terminal.

**operator display terminal (ODT)**

(1) A terminal or other device that is connected to the system in such a way that it can communicate directly with the operating system. The ODT allows operations personnel to accomplish system operations functions through either of two operating modes: system command mode or data comm mode. (2) The name given to the system control terminal (SCT) when it is used as an ODT.

**output**

(1) Pertaining to a device, process, or channel that delivers data from a computer.  
(2) Data delivered from a program or device.

**output file**

A file created or written to by an executing program.

**P**

**pad character**

A character, usually a blank, used to extend any character string to its maximum length. Pad characters are appended to text to completely fill a field or to stretch a line to the rightmost column.

**parameter**

(1) A quantity or item of information that can be given a different value each time a process is repeated. (2) An identifier associated in a special way with a procedure. A parameter is declared in the procedure heading and is automatically assigned a value when the procedure is invoked. (3) An object or value that is passed from an actual parameter and received by a formal parameter. (4) An element of a command, statement, or procedure that enables a user to determine the exact functionality of that command, statement, or procedure. A parameter can be variable or constant, and required or optional.

**physical file**

A file as it is stored on a particular recording medium such as a disk or a tape.

## Glossary

---

**physical unit number**

The permanent identification for a device by which it is known to the system.

**print request**

(1) A request that contains information describing the location, time, and method of printing a printer file or group of printer files. (2) A group of one or more files to be printed, along with the attributes that describe when, where, and how to print them.

**print request list**

A list containing information about all outstanding print requests. Outstanding print requests include those that are queued for printing, those with exceptions, those that are waiting for resources, and those scheduled for printing selection in the future.

**Print System (PrintS)**

A Unisys software product used to control when, where, and how printer backup files are printed on A Series systems.

**printer backup file**

A system-formatted file that contains data to be printed and carriage control information. A printer backup file refers to both printer and card punch output.

**printer file**

A file assigned to a printer. *See* printer backup file.

**printer pool**

*See* default printer pool.

**PrintS**

*See* Print System.

**process**

A software application; that is, any activity or systematic sequence of operations that produces a specified result.

## R

**railroad diagram**

A graphic representation of the syntax of a command or statement.

**remote device**

An I/O unit or other piece of equipment that is physically removed from the computer but connected by a communications line.

**remote host system**

The computer system that contains the message control system (MCS) to which a given station is connected, but that is not the system to which the station is physically connected.

**S****session**

(1) The interactions between a user and a message control system (MCS) during a particular period of time that is assigned an identifying session number. Logging on initiates a new session; logging off terminates a session. Each Menu-Assisted Resource Control (MARC) or Command and Edit language (CANDE) dialogue at a terminal accesses a different session. (2) In Command and Edit language (CANDE), the time measured from when a CANDE user enters a valid usercode/password combination until that user enters a CANDE *SPLIT*, *BYE*, or *HELLO* command. Each CANDE session is assigned a unique number. No output is printed until the session is ended.

**spooling**

The process of indirectly sending output files to peripheral devices such as printers. The output file is logically written to the peripheral device, but actually is sent to a tape or disk file known as a backup file. The backup file is written to the peripheral device when the job finishes. Spooling enables many users to share peripherals efficiently by preventing any one job from monopolizing a peripheral.

**syntax**

The rules or grammar of a language.

**T****task**

(1) A dependent process. (2) A single, complete unit of work performed by the system, such as compiling or executing a program, or copying a file from one disk to another. Tasks are initiated by a job, by another task, or directly by a user.

**task attribute**

Any of a number of items that describe and control various aspects of the execution of a process. Various operator commands, message control system (MCS) commands, and programming language statements can be used to interrogate and modify the values of task attributes. The task attributes of a process are stored in the process information block (PIB).

**TPDLP**

See train printer data link processor.

**train printer**

A type of line printer that uses a removable or replaceable train, band, or drum on which a set of characters is arranged.

**train printer data link processor (TPDLP)**

A data link processor (DLP) that serves as the controller of a train printer and provides the I/O interface between the I/O processor (IOP) and the printer.

**train table**

A table, composed of alphanumeric or hexadecimal characters, that describes the physical organization of printable characters on a print train.



## Glossary

---

### **transform function**

In the Print System (PrintS), a general-purpose function that can manipulate a print line before it is printed on a device. An example is translating lowercase characters to uppercase before printing.

### **translation printer**

A line printer that uses translation tables. Translation printers are also known as buffered printers.

### **translation table**

An array that, given the character output by the system, defines the physical data byte sent to printers. Normally, the character output and the physical data byte are the same. However, invalid characters are usually translated to a question mark (?).

## U

### **UIO**

See universal I/O (UIO) subsystem.

### **unit**

A peripheral device such as a disk drive or a tape drive.

### **unit number**

A number assigned by an installation to a peripheral device, such as a disk drive, and used to identify the device. The unit number commonly appears in conjunction with an acronym indicating the type of unit, which provides a unique identifier for a particular peripheral.

### **universal I/O (UIO) subsystem**

A physical I/O architecture that uses the message level interface (MLI) protocol to connect systems to peripherals by means of I/O base modules and data link processors (DLPs).

### **utility**

A system software program that performs commonly used functions.

## V

### **virtual**

Pertaining to an item whose existence is simulated by the system software. For example, a pseudostation is a virtual station, and a session is a virtual job.

## W

### **WFL**

See Work Flow Language.

**WFL job**

(1) A Work Flow Language (WFL) program, or the execution of such a program. (2) A collection of Work Flow Language (WFL) statements that enable the user to run programs or tasks.

**Work Flow Language (WFL)**

A Unisys language used for constructing jobs that compile or run programs on A Series systems. WFL includes variables, expressions, and flow-of-control statements that offer the programmer a wide range of capabilities with regard to task control.



# Bibliography

- A Series CANDE Operations Reference Manual* (form 8600 1500). Unisys Corporation.
- A Series COBOL ANSI-68 Programming Reference Manual* (form 8600 0320). Unisys Corporation.
- A Series COBOL ANSI-74 Programming Reference Manual, Volume 1: Basic Implementation* (form 8600 0296). Unisys Corporation.
- A Series File Attributes Programming Reference Manual* (form 8600 0064). Unisys Corporation. Formerly *A Series I/O Subsystem Programming Reference Manual*.
- A Series I/O Subsystem Programming Guide* (form 8600 0056). Unisys Corporation. Formerly *A Series I/O Subsystem Programming Reference Manual*.
- A Series Print System (PrintS/ReprintS) Administration, Operations, and Programming Guide* (form 8600 1039). Unisys Corporation.
- A Series System Commands Operations Reference Manual* (form 8600 0395). Unisys Corporation.
- A Series Task Attributes Programming Reference Manual* (form 8600 0502). Unisys Corporation. Formerly *A Series Work Flow Administration and Programming Guide*.
- A Series Work Flow Language (WFL) Programming Reference Manual* (form 8600 1047). Unisys Corporation.



# Index

## A

ALPHATABLE command, 4-7

## B

BACK command, 2-2, 2-18

- loading files with, 2-2
- removing files with, 2-5
- setting options with, 2-5

backup files

- automatic selection, 2-15
- copying, 2-23, 3-19
  - carriage control, 2-29
  - record size used, 2-23
  - specifying columns, 2-31
  - specifying records, 2-29, 2-31, 2-32
- current file, 2-13, 2-15

- displaying, 2-19

- deleting, 2-26

- displaying, 2-21

- controlling rate of, 2-35
  - specifying columns, 2-31
  - specifying records, 2-29, 2-31, 2-32

- displaying information about, 2-14, 2-15

- establishing current file, 2-13

- file numbers, 2-10

- file titles, 2-10

- loading, 2-1

- during initiation, 2-2
  - from a directory, 2-3
  - from a family, 2-4
  - from a specific job, 2-3
  - from current session, 2-2
  - from previous session, 2-2

- \*BD files only, 2-4

- making available, 2-1

- on tape, 3-7

- printing many files, 3-8
  - printing many reels, 3-8
  - printing single file, 3-7

- printing, 2-24, 3-3

- any file, 2-25

- current file, 2-25

- from current job, 3-5

- from disk, 3-5

- from particular job, 3-6

- from tape, 3-7

- header pages, 3-11, 3-23

- multiple copies, 3-10

- multiple files, 2-25, 3-8

- omitting records, 3-17

- particular records, 3-18

- single file, 3-6, 3-7

- specific portions of, 3-11

- specifying device for, 3-10

- record numbers of, 2-18

- removing, 2-26

- at end of session, 2-7

- messages displayed, 2-7

- through a WFL job, 2-6

- through CANDE, 2-5

- through MARC, 2-2

- with BACK command, 2-5

- selecting, 2-13, 3-3

- specifying, 3-3

- viewing, 2-18

- and entering commands, 2-22

- current file, 2-19

- partial, 2-18

- specifying columns, (See < column range >)

- specifying sequence numbers, 2-20

- working set

- changing, 2-11

- defined, 2-1

- displaying, 2-10

- displaying portions of, 2-13

- loading, 2-10, 2-11

Backup Processor

- capabilities, 2-1

## Index

---

- commands
    - BACK, 2-2, 2-18
    - DIRECTORY, 2-10, 2-18
    - during listing, 2-22
    - FIRST, 2-21
    - LAST, 2-21
    - LIST, 2-18
    - NEXT, 2-13
    - OPTION, 2-8
    - QUERY, 2-17
    - QUIT, 2-7
    - REMOVE, 2-26
    - SAME, 2-21
    - SELECT, 2-13
    - text positioning, (See text positioning commands)
    - WHAT, 2-17
      - +, 2-21
      - , 2-22
  - copying backup files, 2-23
  - deleting backup files, 2-26
  - exiting, 2-7
  - file numbers in, 2-10
  - file titles in, 2-10
  - initiating, 2-1
    - from a WFL job, 2-6
    - from CANDE, 2-2
    - from MARC, 2-2
    - messages, 2-6
    - removing files, 2-5
    - removing files and, 2-2
  - loading files, 2-1
    - during initiation, 2-2
    - from a directory, 2-3
    - from a family, 2-4
    - from a specific job, 2-3
    - from current session, 2-2
    - from previous session, 2-2
  - printing backup files, 2-24
  - removing backup files, 2-26
  - text positioning commands, 2-21
- Backup/Processor Utility screen, 2-2
- BFILE file, 3-20
- ### C
- C option, (See CARRIAGECONTROL option)
  - CARRIAGECONTROL option, 2-29
  - character set
    - defining, 4-7
    - size of, 4-7
  - CHARSET command, 4-7
  - COL option, (See COLUMNS option)
  - <column range>, 2-19, 2-20
  - COLUMNS option, 2-29
    - examples of, 2-30
  - COMP option, (See COMPRESS option)
  - COMPRESS option, 2-30
  - CONT option, (See CONTINUOUS option)
  - CONTINUOUS option, 2-35
  - COPIES option, 3-10
  - COPY command, 2-23
    - CARRIAGECONTROL option, 2-29
    - COLUMNS option, 2-29
    - COMPRESS option, 2-30
    - examples of, 2-24
    - EXTRACT option, 2-31
    - NO modifier, 2-23
    - NOCRUNCH option, 2-32
    - RECORDS option, 2-32
    - setting options with, 2-8
    - SUPPRESS option, 2-32
    - UPCASED option, 2-34
  - copying backup files
    - with Backup Processor, 2-23
    - with SYSTEM/BACKUP, 3-19
  - current file, 2-13
- ### D
- D clause, 3-5
  - DEFAULT command, 4-7
  - DELETE PRINTSET command, 4-14
  - deleting backup files
    - with Backup Processor, 2-26
  - DIR command, (See DIRECTORY command)
  - DIRECTORY command, 2-10, 2-18
    - examples of, 2-11
    - sample output, 2-10
    - SKIP option, 2-13
    - <directory name>, 2-3
      - examples of, 2-3
      - used with BACK command, 2-3
      - used with DIRECTORY command, 2-11
- ### E
- <end column>, 2-29
  - ET option, (See ET1100 option)

ET1100 option, 2-30  
 examples of, 2-31

EX option, (See EXTRACT option)

EXTRACT option, 2-9, 2-31

## F

FI command, (See FIRST command)

file equation

with SYSTEM/BACKUP, 3-20

file number

in Backup Processor, 2-10

in PRINT command, 2-25

in REMOVE command, 2-27

file specification, 3-3

file title

in PRINT command, 2-25

file titles

in Backup Processor, 2-10

FILEIN file, 3-20

files

backup, (See backup files)

FILOUT file, 3-20

FIRST command, 2-21

## H

HELP command, 2-28

BATCH option, 2-28

< command name > option, 2-28

COMMANDS option, 2-28

INTERACTIVE option, 2-28

INTRODUCTION option, 2-28

< option name > option, 2-28

OPTIONS option, 2-28

HEXTABLE command, 4-7

## I

ID option, 3-11

IDOPTION compiler option, (See \$IDOPTION)

INFOPTION, (See \$INFOPTION)

invalid character

specifying, 4-8

INVCHR command, 4-8

## J

job numbers

turning over, 2-3

used with BACK command, 2-3

used with DIRECTORY command, 2-11

## K

KEY option, 3-11

< key part >, 3-12

NOINCL option and, 3-17

< range part >, 3-12

## L

L command, (See LIST command)

LA command, (See LAST command)

LAST command, 2-21

LIN file, 3-20

LIST command, 2-18

COLUMNS option, 2-29

COMPRESS option, 2-30

CONTINUOUS option, 2-35

examples of, 2-20

EXTRACT option, 2-31

RECORDS option, 2-32

setting options with, 2-8

SUPPRESS option, 2-32

TRUNCATE option, 2-33

UNSEQUENCED option, 2-34

UPCASED option, 2-34

WAIT option, 2-35

with session options, 2-20

loading backup files

during initiation, 2-2

from a directory, 2-3

from a family, 2-4

from a specific job, 2-3

from current session, 2-2

from previous session, 2-2

\*BD files only, 2-4

LSN option, 3-16

LTTABLEGEN utility, 4-1

commands, 4-6

ALPHATABLE, 4-7

basic constructs of, 4-4

CHARSET, 4-7

DEFAULT, 4-7



## Index

---

DELETE PRINTSET, 4-14  
HEXTABLE, 4-7  
INVCHR, 4-8  
MEDIUM, 4-8  
NAME, 4-8  
PRINTALL, 4-9, 4-16  
PRINTID, 4-15  
PRINTNAMES, 4-9, 4-16  
PRINTRAINID, 4-10  
PRINTSET, 4-14  
PUNCHALPHA, 4-10  
PUNCHARRAY, 4-10  
PUNCHHEX, 4-11  
SLOW, 4-8  
TRAINID, 4-9  
TRANSLATE, 4-13  
example uses, 4-11  
reasons to use, 4-4  
running, 4-17

## M

MEDIUM command, 4-8  
minus command, (See - command)  
modifier  
    Backup Processor  
        NO, 2-23  
MT clause, 3-7  
    use of options with, 3-9

## N

N command, (See NEXT command)  
NAME command, 4-8  
ND option  
    copying files with, 3-19  
NEXT command, 2-13  
NO modifier, 2-23  
NOCRUNCH option, 2-32  
NOINCL option, 3-17

## O

OPTION clause, 2-5  
    with BACK command, 2-5  
OPTION command, 2-8  
    examples of, 2-9  
options, 2-8, 2-28

Backup Processor, 2-8, 2-28  
    CARRIAGECONTROL, 2-29  
    changing, 2-9  
    COLUMNS, 2-29  
    COMPRESS, 2-30  
    CONTINUOUS, 2-35  
    default values, 2-8, 2-9  
    displaying, 2-8  
    ET1100, 2-30  
    EXTRACT, 2-9, 2-31  
    NOCRUNCH, 2-32  
    RECORDS, 2-32  
    setting during initiation, 2-5  
    setting with BACK command, 2-5  
    SUPPRESS, 2-32  
    TRUNCATE, 2-33  
    UNSEQUENCED, 2-34  
    UPCASED, 2-34  
    WAIT, 2-35  
SYSTEM/BACKUP, 3-9  
    COPIES, 3-10  
    ID, 3-11  
    LSN, 3-16  
    NOINCL, 3-17  
    RECORD, 3-18  
    SAVE, 3-19  
    SPACING, 3-19  
    used with MT clause, 3-9  
    \$IDOPTION, 3-23  
    \$INFOPTION, 3-23

## P

parity errors in SYSTEM/BACKUP, 3-23  
plus command, (See + command)  
PRINT command, 2-24  
    attribute phrases with, 2-24, 2-25, 2-26  
    examples of, 2-25  
    PRINTDEFAULTS clause, 2-24, 2-26  
PRINTALL command, 4-9, 4-16  
PRINTDEFAULTS clause, 2-24, 2-26  
printer jams, 3-21  
PRINTID command, 4-15  
printing backup files  
    with Backup Processor, 2-24  
PRINTNAMES command, 4-9, 4-16  
PRINTRAINID command, 4-10  
printset, 4-14  
    deleting, 4-14  
    describing, 4-14  
    ID of, 4-14

obtaining information about, 4-15  
 PRINTSET command, 4-14  
 PUNCHALPHA command, 4-10  
 PUNCHARRAY command, 4-10  
 PUNCHHEX command, 4-11

## Q

Q command, (See QUIT command)  
 QUERY command, 2-17  
 QUIT command, 2-7

## R

R option, (See RECORDS option)  
 railroad diagrams, explanation of, A-1  
 record numbers, 2-18  
   preventing display of, 2-34  
 RECORD option, 3-18  
 RECORDS option, 2-32  
 REM command, (See REMOVE command)  
 REMOVE clause, 2-5  
   with BACK command, 2-5  
 REMOVE command, 2-26  
   examples of, 2-27  
 removing backup files  
   at end of session, 2-7  
   messages displayed, 2-7  
   through a WFL job, 2-6  
   through CANDE, 2-5  
   through MARC, 2-2  
   with Backup Processor, 2-26

## S

S command, (See SELECT command)  
 SA command, (See SAME command)  
 SAME command, 2-21  
 SAVE option, 3-19  
 SELECT command, 2-13  
 <sequence range>, 2-19, 2-20  
 session numbers  
   turning over, 2-3  
   used with BACK command, 2-3  
   used with DIRECTORY command, 2-11  
 SKIP option, 2-13  
 SLOW command, 4-8  
 SPACING option, 3-19

SQ option, (See COMPRESS option)  
 SQUASHED option, (See COMPRESS option)  
 <start column>, 2-29  
 SUPP option, (See SUPPRESS option)  
 SUPPRESS option, 2-32  
 SYSTEM/BACKUP utility, 3-1  
   backup file specification, 3-3  
   compile-time options, 3-23  
   copying files, 3-19  
   D clause, 3-5  
   file equation in, 3-20  
   file specification, 3-3  
   initiating, 3-1  
     example commands for, 3-2  
   internal files of, 3-20  
   MT clause, 3-7  
     use of options with, 3-9  
   options, 3-1, 3-9  
     COPIES, 3-10  
     ID, 3-11  
     LSN, 3-16  
     NOINCL, 3-17  
     RECORD, 3-18  
     SAVE, 3-19  
     SPACING, 3-19  
     used with MT clause, 3-9  
     \$IDOPTION, 3-23  
     \$INFOPTION, 3-23  
   parity errors, 3-23  
   printer jams, 3-21  
   printing files, 3-3  
     from current job, 3-5  
     from disk, 3-5  
     from particular job, 3-6  
     from tape, 3-7  
     header pages, 3-11, 3-23  
     multiple, 3-8  
     multiple copies, 3-10  
     omitting records, 3-17  
     particular records, 3-18  
     single file, 3-6, 3-7  
     specific portions of, 3-11  
     specifying device for, 3-10  
   repositioning the input file, 3-21  
   uses, 3-1  
 SYSTEM/BACKUP Utility screen, 3-2  
 SYSTEM/TRAINABLES file, 4-1, 4-4  
   generating, 4-17  
   printing, 4-15

## Index

---

### T

T option, (See TRUNCATE option)

tape backup files

copying, 3-19

printing many files, 3-8

printing many reels, 3-8

printing single file, 3-7

text positioning commands, 2-18, 2-21

entering, 2-19

FIRST, 2-21

LAST, 2-21

SAME, 2-21

+, 2-21

-, 2-22

train printer, 4-5

train tables, 4-5

commands, 4-6

ALPHATABLE, 4-7

CHARSET, 4-7

DEFAULT, 4-7

HEXTABLE, 4-7

INVCHR, 4-8

MEDIUM, 4-8

NAME, 4-8

PRINTALL, 4-9

PRINTNAMES, 4-9

PRINTRAINID, 4-10

PUNCHALPHA, 4-10

PUNCHARRAY, 4-10

PUNCHHEX, 4-11

SLOW, 4-8

TRAINID, 4-9

default names, 4-1, 4-3

defining, 4-7, 4-12

naming, 4-8

printing, 4-9, 4-10

punching, 4-10, 4-11

specifying default, 4-7

TRAINID values

assigning, 4-9

for medium-speed printers, 4-3

for slow-speed printers, 4-1

TRAINID command, 4-9

TRANSLATE command, 4-13

translation tables, 4-12

commands, 4-13

DELETE PRINTSET, 4-14

PRINTALL, 4-16

PRINTID, 4-15

PRINTNAMES, 4-16

PRINTSET, 4-14

TRANSLATE, 4-13

creating, 4-16

default names, 4-4

deleting, 4-14

TRAINID values, 4-4

uses, 4-1

TRUNCATE option, 2-33

### U

U option, (See UNSEQUENCED option)

UNSEQUENCED option, 2-34

UP option, (See UPCASED option)

UPCASED option, 2-34

### W

WAIT option, 2-35

WHAT command, 2-17

working set

changing, 2-11

defined, 2-1

displaying, 2-10

displaying portions of, 2-13

loading, 2-10, 2-11

examples of, 2-11

+ command, 2-21

\$IDOPTION, 3-23

\$INFOPTION, 3-23

- command, 2-22

## Help Us To Help You

Publication Title \_\_\_\_\_

Document Number \_\_\_\_\_

Date \_\_\_\_\_

Unisys Corporation is interested in your comments and suggestions regarding this manual. We will use them to improve the quality of your Product Information. Please check type of suggestion:

Addition

Deletion

Revision

Recommended Change (Please identify affected pages) \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Name \_\_\_\_\_

Telephone Number \_\_\_\_\_

Title \_\_\_\_\_

Company \_\_\_\_\_

Address (Street, City, State, Zip) \_\_\_\_\_

## Help Us To Help You

Publication Title \_\_\_\_\_

Document Number \_\_\_\_\_

Date \_\_\_\_\_

Unisys Corporation is interested in your comments and suggestions regarding this manual. We will use them to improve the quality of your Product Information. Please check type of suggestion:

Addition

Deletion

Revision

Recommended Change (Please identify affected pages) \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Name \_\_\_\_\_

Telephone Number \_\_\_\_\_

Title \_\_\_\_\_

Company \_\_\_\_\_

Address (Street, City, State, Zip) \_\_\_\_\_

## Help Us To Help You

Publication Title \_\_\_\_\_

Document Number \_\_\_\_\_

Date \_\_\_\_\_

Unisys Corporation is interested in your comments and suggestions regarding this manual. We will use them to improve the quality of your Product Information. Please check type of suggestion:

Addition

Deletion

Revision

Recommended Change (Please identify affected pages) \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Name \_\_\_\_\_

Telephone Number \_\_\_\_\_

Title \_\_\_\_\_

Company \_\_\_\_\_

Address (Street, City, State, Zip) \_\_\_\_\_



# BUSINESS REPLY MAIL

First Class      Permit No. 817      Detroit, MI 48232

Postage Will Be Paid By Addressee

Unisys Corporation  
ATTN: Publications  
25725 Jeronimo Road  
Mission Viejo, CA 92691-9826 USA

No Postage  
necessary  
if mailed in the  
United States



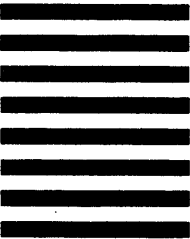
# BUSINESS REPLY MAIL

First Class      Permit No. 817      Detroit, MI 48232

Postage Will Be Paid By Addressee

Unisys Corporation  
ATTN: Publications  
25725 Jeronimo Road  
Mission Viejo, CA 92691-9826 USA

No Postage  
necessary  
if mailed in the  
United States



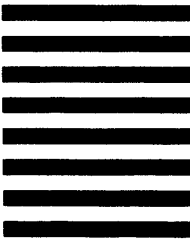
# BUSINESS REPLY MAIL

First Class      Permit No. 817      Detroit, MI 48232

Postage Will Be Paid By Addressee

Unisys Corporation  
ATTN: Publications  
25725 Jeronimo Road  
Mission Viejo, CA 92691-9826 USA

No Postage  
necessary  
if mailed in the  
United States







8600692-000