

IBM VSE/Enterprise Systems Architecture



System Utilities

Version 2 Release 4

IBM VSE/Enterprise Systems Architecture



System Utilities

Version 2 Release 4

Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page xi.

Second Edition (June 1999)

This edition applies to Version 2 Release 4 of IBM Virtual Storage Extended/Enterprise Systems Architecture (VSE/ESA), Program Number 5690-VSE, and to all subsequent releases and modifications until otherwise indicated in new editions.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the addresses given below.

A form for readers' comments is provided at the back of this publication. If the form has been removed, address your comments to:

IBM Corporation
Attn: Dept ECJ-BP/003D or to:
6300 Diagonal Highway
Boulder, CO 80301
U.S.A.

IBM Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1984, 1999. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	vii
Tables	ix
Notices	xi
Trademarks and Service Marks	xi
About This Book	xiii
Who Should Use This Book	xiii
How to Use This Book	xiii
Where to Find More Information	xiii
Summary of Changes	xv

Part 1. VSE/Fast Copy and Overview of Stand-Alone Environment 1

Chapter 1. Overview of Saving Data	3
Chapter 2. VSE/Fast Copy (Online and Stand-Alone)	5
DUMP (Backup), RESTORE, and COPY Functions	6
VSE/Fast Copy Control Statements (Online and Stand-Alone)	7
DUMP, RESTORE, COPY – ALL	7
DUMP, RESTORE, COPY – VOLUME	8
Special VOLUME Functions	9
DUMP, RESTORE, COPY – FILE	10
Special FILE Functions	12
Description of VSE/Fast Copy Control Statement Parameters	12
Integrity of System Files and Shared Disks	16
Sharing DASD	16
Restoring onto VM Minidisks	16
Calculating Device Capacity for Restore	16
VSE/Fast Copy Error Handling	18
Error Handling for FBA Disk Devices	18
Error Handling for CKD/ECKD Disk Devices	18
Return Codes	18
Summary of What You Should Consider	19
Summary of When to Use VSE/Fast Copy Online or Stand-Alone	20
Parameters Not Supported in VSE/Fast Copy Stand-Alone	20
Creating a Backup of the VSE/ESA System (DOSRES and SYSWK1)	21
Chapter 3. Running VSE/Fast Copy Online	23
Job Control Statements to be Entered	23
Device Assignments and Label Information	24
Storage and Run Requirements to Improve Performance	26
Space Requirements for an Intermediate Data Set on Disk	26
Running in Virtual or Real Mode (Performance Considerations)	26
Reading from and Writing to an FBA Disk Device	27
Examples for Running VSE/Fast Copy Online	27
Example 1 – Copy Partial FBA Volume	27
Example 2 – Dump a CKD/ECKD Volume	28
Example 3 – Dump a Volume with EXCLUDE and OPTIMIZE	28
Example 4 – Restore a Volume	28
Example 5 – Selective Restore to Same Location	29

Example 6 – Selective Restore to Other Location (Using REORGANIZE)	29
Example 7 – Copy a Multivolume File	30

Chapter 4. VSE/ESA Stand-Alone Environment (Including VSE/Fast Copy Stand-Alone)	31
Functional Characteristics of the Stand-Alone Environment.	31
Performing an IPL from the Stand-Alone Tape	32
Using VSE/Fast Copy Stand-Alone	33
VSE/Fast Copy Stand-Alone Control Statements	34
Examples for Running VSE/Fast Copy Stand-Alone	35
Overview of VSE Librarian RESTORE (Stand-Alone Environment)	37
Example of a Librarian Restore Run	37
Overview of Device Support Facilities (Stand-Alone Environment)	38
Example of an ICKDSF Run	39
Overview of DITTO/ESA for VSE (Stand-Alone Environment)	40
Example of a DITTO/ESA Run	40
Customizing Stand-Alone Utilities Using the Customization Table	41
Customized Answer Table	42
Customized Message Table	42
What You Should Consider when Changing Messages or Replies	42
Use of Macro SASTCK	43
Macro SASTCK	43

Part 2. Migrating to a New Release 49

Chapter 5. VSE/ESA Migration Utility IESBLDUP	51
Introducing the IESBLDUP Utility	51
Planning and Concept Information	52
Migrating VSE/ICCF DTSFILE and VSE Control File to VSE/ESA 2.4	52
Copying DOS/VSE User Definitions of VSE/ICCF DTSFILE Only	52
Migrating VSE/ICCF DTSFILE Only (DOS/VSE).	52
VSE/ICCF DTSFILE	52
VSE/ESA Required VSE/ICCF Options	52
Reserved VSE/ICCF Libraries	53
CICS Sign-On Table (DFHSNT).	53
Using the IESBLDUP Utility	54
Assignments.	54
Running IESBLDUP	54
The Status Report.	55
Model User Profiles	55
Adding New Users	55
Migrating to VSE/ESA 2.4	56
Backups Required from Your Old System	56
Precautionary Measures	57
Migration Steps.	57
The Control Statement	61
Control Statement Format	62
Keyword Description	62
Recovering from Migration Errors	64

Part 3. Miscellaneous Utilities and Programs 67

Chapter 6. Batch Program IESUPDCF.	69
Preparing to Use IESUPDCF.	69
Planning for User Profiles	69

Preparing Skeleton IESUPDCF	69
Setting the ICCF Parameter in Skeleton IESUPDCF	70
Adding a User ID in Skeleton IESUPDCF	71
Altering a User ID in Skeleton IESUPDCF	75
Deleting a User ID in Skeleton IESUPDCF	76
Skeleton IESUPDCF	76
Using IESUPDCF	78
Return Codes Issued by IESUPDCF	78
Example of Completed Skeleton IESUPDCF	80
Chapter 7. Utilities DTRSETP, DTRIINIT, and JCL Analyzer	81
DTRSETP Utility	81
Using SETPARM Procedures	81
Using DTRSETP	81
DTRSETP Operations	82
Restrictions When Using DTRSETP	83
Return Codes	83
DTRIINIT Utility.	84
Loading Jobs During Normal Processing	84
Loading Jobs During a COLD Startup	85
Loading Jobs into a System with Security Active	85
Additional Considerations	85
JCL Analyzer	86
Chapter 8. Device Utilities	87
Device Support.	87
Disks and Diskettes	87
Magnetic Tapes.	87
Control Statement Input	87
Job Control	87
Utility Control	87
Storage Requirements	88
Error Messages	88
Clear Disk (CLRDK)	88
Job Control Statements.	89
Utility Control Statement	89
Using the Clear Disk Utility	89
Copy and Restore Diskette (CRDR)	90
Job Control Statements.	91
Utility Control Statement	91
Using the Copy and Restore Diskette Utility	93
Initialize Tape (INTTP)	96
Job Control Statements.	97
Utility Control Statement	97
Using the Initialize Tape Utility	99
Chapter 9. Printing or Displaying System Information	101
Displaying the Label Information Area (LSERV)	101
Job Control Statement.	101
Displaying the Volume Table of Contents (VTOC).	102
Using the LVTOC Program	103
Using the Display VTOC Dialog	103
Printing and Handling the Recorder File	104
Printing the Hardcopy File (PRINTLOG)	105
Printing Job Information (LISTLOG).	108

Chapter 10. Batch Access to the VSE/ESA Host Transfer File	109
Overview	109
Command Description	109
LOAD Command	109
UNLOAD Command	110
DELETE Command	110
DIRECT Command	111
Sample Scenario	111
Exchanging Data between the HTF and a VSE/VSAM ESDS File	111
List HTF Directories.	113

Part 4. Appendixes	115
-------------------------------------	------------

Appendix A. Job Control Scanner Program	117
Program Description	117
Operating Environment	117
Preparing the Input	118
Description of Output	119
Starting a Scan Run	123
Messages	124
Appendix B. Understanding Syntax Diagrams	127
Glossary	131
Index	139

Figures

1. Functions of VSE/Fast Copy	5
2. Device Assignments and Label Information	24
3. Using CKD/ECKD or FBA Disk Devices with VSE/Fast Copy Online	25
4. Loading Stand-Alone Programs	33
5. Example Job for Backing Up the VSE Control File	56
6. Job Stream Example for Migrating from VSE/SP or Previous VSE/ESA System	58
7. Job Stream Example for Migrating from DOS/VSE	58
8. Job Stream Example for Status Report.	59
9. Example for a Status Report Output.	60
10. Restoring the VSE Control File.	65
11. Skeleton IESUPDCF	76
12. Example of a Completed Skeleton IESUPDCF	80
13. VSE/ESA System Utilities and their Run Sizes (Real Mode)	88
14. Track 0 Table	94
15. Sample LSERV Output	102
16. Job to Define a VSE/VSAM ESDS File	112
17. Job to Load VSE/VSAM File into HTF.	112
18. Job to Unload a HTF File into a VSE/VSAM ESDS File	113
19. Job to List HTF User Directories.	113
20. Sample Listing of HTF Directory	113
21. Example for Preparing Scanner Input under VM	119
22. Example of a Scanner SYSLST Output for Input of Type D	120
23. Example of a Scanner SYSLST Output for Input of Type U	121
24. Example of Records in a Scanner SYSPCH Output	122
25. Sample Job for Starting a Scan Run	123

Tables

1.	Overview on Programs for Saving Data	3
2.	Return Codes	18
3.	VSE/Fast Copy Control Statements with Description	20
4.	Values of OPTIMIZE for Storage and Run Requirements	26
5.	Overview of Stand-Alone Environment	31
6.	Functions of DITTO/ESA in a Stand-Alone Environment	40
7.	Defining IBM 3480/3490 Tape Devices	87

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785, U.S.A.

Any pointers in this publication to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement. IBM accepts no responsibility for the content or use of non-IBM Web sites specifically mentioned in this publication or accessed through an IBM Web site that is mentioned in this publication.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Deutschland Informationssysteme GmbH
Department 0215
Pascalstr. 100
70569 Stuttgart
Germany

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

Trademarks and Service Marks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

CICS
CICS/VSE
ECKD
IBM
NetView
OS/2
VisualAge
VM/ESA
VSE/ESA
VTAM

Windows NT is a trademark of Microsoft Corporation in the United States and/or other countries.

About This Book

This book describes VSE/ESA 2.4 system utilities and related functions.

Who Should Use This Book

This book addresses mainly system administrators and operators who need to perform tasks that require VSE/ESA system utilities. This includes, for example, saving data or migrating to a new release of VSE/ESA.

How to Use This Book

This book describes VSE/ESA system utilities and provides examples for using them. You can use the examples as a base for your own system.

Where to Find More Information

For an overview and details on the VSE/ESA 2.4 support, refer to the *VSE/ESA Planning* manual.

VSE/ESA Home Page

VSE/ESA has a home page on the World Wide Web, which offers up-to-date information about VSE-related products and services, new VSE/ESA functions, and other items of interest to VSE users.

You can find the VSE/ESA home page at:

<http://www.ibm.com/s390/vse/>

Summary of Changes

This edition of the manual documents VSE/ESA 2.4 enhancements and changes as follows:

- Chapter 5 has been partly rewritten to reflect the latest changes for migrating user profiles.
- The following topics are new in this edition of the manual:
 - JCL Analyzer
 - Batch Access to the Host Transfer File

Both topics were formerly included in the *VSE/ESA Enhancements* manual.

- Editorial changes, technical corrections, and Year 2000 adaptations have also been included.

Part 1. VSE/Fast Copy and Overview of Stand-Alone Environment

Chapter 1. Overview of Saving Data

VSE/ESA consists of different data and library formats. Because of this, VSE/ESA offers different backup and restore programs (or dialogs) which are designed to fit these formats. Table 1 lists the main data and library formats and gives recommendations on how to save the respective data.

It is possible to use VSE/Fast Copy to back up most of your data. This manual describes the VSE/Fast Copy program. Thus Table 1 shows recommendations for the use of VSE/Fast Copy.

If you want to do a backup of specific data, such as VSE/POWER queues, VSE/VSAM data, or VSE/ICCF libraries, it is recommended to use the backup utilities (or dialogs) provided for these programs. For a description, refer to the IBM manuals indicated in the following figure.

Table 1. Overview on Programs for Saving Data

Kind of data	Backup	Restore	Reorganize	Recommended	Where described
Total volumes (for example, DOSRES, SYSWK1)	VSE/Fast Copy or DITTO/ESA	VSE/Fast Copy or DITTO/ESA	(See note)	VSE/Fast Copy	<i>VSE/ESA System Utilities</i> <i>DITTO/ESA User's Guide and Reference</i>
VSE libraries	Librarian BACKUP	Librarian RESTORE	Librarian BACKUP/ RESTORE	Librarian or Interactive Interface dialogs	<i>VSE/ESA Guide to System Functions</i> <i>VSE/ESA System Control Statements</i>
Non-VSE/VSAM data	VSE/Fast Copy or DITTO/ESA	VSE/Fast Copy or DITTO/ESA	–	VSE/Fast Copy	<i>VSE/ESA System Utilities</i>
VSE/VSAM data	VSE/VSAM, BACKUP	VSE/VSAM, RESTORE	VSE/VSAM, EXPORT and IMPORT	VSE/VSAM, BACKUP/RESTORE or Interactive Interface dialogs	<i>VSE/VSAM Commands</i> <i>VSE/ESA Administration</i>
VSE/ICCF libraries	DTSUTIL, BACKUP	DTSUTIL, RESTORE	DTSUTIL, BACKUP FORMAT and RESTORE	DTSUTIL	<i>VSE/ICCF Administration and Operation</i>
VSE/POWER queue	POFFLOAD, BACKUP	POFFLOAD, LOAD	–	POFFLOAD or Interactive Interface dialogs	<i>VSE/POWER Administration and Operation</i>
DB2	ARCHIVE	ARCHIVE	–	ARCHIVE	<i>DB2 Data Base Administration</i>
DL/I	IMAGE COPY	IMAGE COPY	Reorganization UNLOAD and RELOAD	DL/I	<i>DL/I Resource Definition and Utilities</i>
System History File	MSHP or VSE/Fast Copy or DITTO/ESA	MSHP or VSE/Fast Copy or DITTO/ESA	–	MSHP	<i>VSE/ESA System Control Statements</i>

Note: Reorganizing data means a change of the physical arrangement of data to obtain a better correspondence of physical and logical structures in order to speed up access and to utilize storage more efficiently. The VSE/Fast Copy parameter REORGANIZE only relocates a file and does not rearrange data in general.

For a description of the dialogs that are available for performing a backup or a restore of data, refer to the IBM manual *VSE/ESA Operation*.

Chapter 2. VSE/Fast Copy (Online and Stand-Alone)

VSE/Fast Copy exists in two different versions:

- **VSE/Fast Copy online** and
- **VSE/Fast Copy stand-alone**

Generally, both versions work the same way and have the same purpose. The main difference is that VSE/Fast Copy stand-alone supports only a subset of the functions supported by VSE/Fast Copy online. “Summary of When to Use VSE/Fast Copy Online or Stand-Alone” on page 20 has more information.

As shown in Figure 1, you can perform the following tasks using VSE/Fast Copy:

1. Create a Fast Copy backup from disk to tape
2. Restore a Fast Copy backup from tape to disk
3. Copy from disk to disk

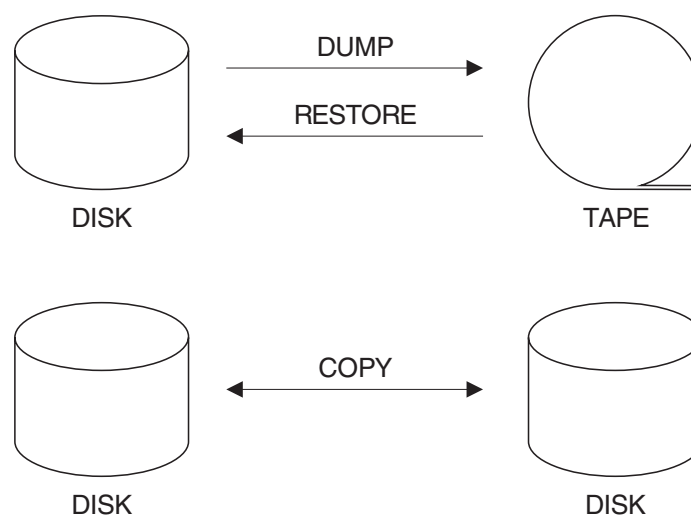


Figure 1. Functions of VSE/Fast Copy

Remarks:

1. The tape device must be supported by VSE/ESA. Alternate tapes are supported.
2. The disk device must be supported by VSE/ESA and may be of device type CKD, ECKD, or FBA.
3. VSE/Fast Copy can process tapes that were produced by:
 - The DOS/VS Fast Copy Disk Volume utility (for RESTORE VOLUME only)
 - VSE/Fast Copy Version 1 and Version 2.

VSE/Fast Copy stand-alone and VSE/Fast Copy online are compatible, as long as you use functions which are supported in both environments.

4. For dumping and restoring data in the VSE/Fast Copy online environment, see also Figure 3 on page 25.

DUMP (Backup), RESTORE, and COPY Functions

As shown before, with VSE/Fast Copy you can DUMP, RESTORE, and COPY data:

1. **DUMP from disk to tape**

You can backup data located on an FBA or CKD disk device to a tape, using the VSE/Fast Copy DUMP function. In this manual this is referred to as *dump* data. For dumping and restoring data in the VSE/Fast Copy online environment, see also Figure 3 on page 25. You later can restore the data to the same (or a different) disk **of the same device type**. You can dump either:

- A complete volume
- A partial volume
- A single file
- A multivolume file.

On the target disk the file occupies the same data extents as on the source disk if you did not request a relocation. For relocation of files, see “Reorganization of Files” on page 12.

2. **RESTORE from tape to disk**

You can restore data from an intermediate data set produced by a DUMP statement to a target disk. VSE/Fast Copy restores the data to exactly the same location as it occupied on the source disk, if you did not request relocation.

3. **COPY from disk to disk**

You can copy data from a source disk to a target disk directly (without using an intermediate data set). You can copy from an IBM FBA, CKD, or ECKD disk device type to any other IBM FBA, CKD, or ECKD disk device type, if:

- The capacity of the target device (CKD, ECKD, FBA) is equal to or larger than the capacity of the source disk.
- The target disk (CKD or ECKD) is of the same device type as the source disk. For example, VSE/Fast Copy can copy from an IBM 3380 Model D to an IBM 3380 Model E, but it cannot copy from an IBM 3390 to an IBM 9345. Also, you cannot copy from an IBM 3380 CKD disk device to an IBM 3380 ECKD disk device and vice versa.
- FBA disk devices have the same block size. The device type need not be the same.

Each of the system control statements (DUMP, RESTORE, and COPY) has specific operands. VSE/ESA uses the **operands**:

- DUMP, RESTORE, COPY – **ALL**
- DUMP, RESTORE, COPY – **VOLUME**
- DUMP, RESTORE, COPY – **FILE** (in the case of VSE/Fast Copy online).

“VSE/Fast Copy Control Statements (Online and Stand-Alone)” on page 7 describes the VSE/Fast Copy operands.

VSE/Fast Copy Control Statements (Online and Stand-Alone)

The utility control statements contain the information needed by VSE/Fast Copy to perform a desired function. You supply this information as one or more operands. The statement operands (DUMP, RESTORE, or COPY) tell the program which function is to be performed. For a description of the operands of these statements (in alphabetical sequence), refer to “Description of VSE/Fast Copy Control Statement Parameters” on page 12.

For input of a control statement, use only the character positions 1 through 72; positions 73 through 80 are reserved. If your statement does not fit on one line, indicate continuation of specifications with a hyphen (separated by a blank) after the last specification on the line. Do not make a line break within an operand. For an example of a line break, refer to 28.

Not all of the operands are available in VSE/Fast Copy stand-alone version. The “Summary of When to Use VSE/Fast Copy Online or Stand-Alone” on page 20 shows the differences.

DUMP, RESTORE, COPY – ALL

The ALL operand processes the entire disk volume physically. It moves all data to or from a disk, without using the Volume Table of Contents (VTOC), that is, data which is not registered in the VTOC. The ALL function also processes unused extents.

Use an ALL function if you:

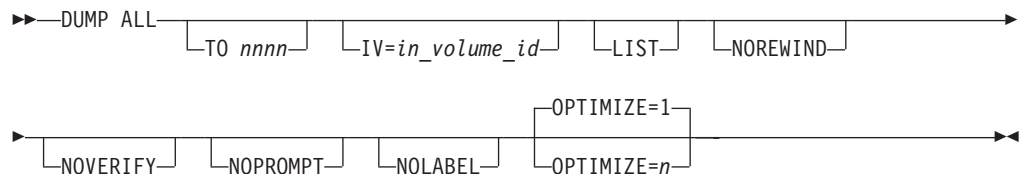
- Want data to be copied which has no VTOC entry.
- Cannot execute the VOLUME function, because of an invalid VTOC on the disk to be dumped or copied.

If you use a tape label (TLBL) when you DUMP data, you also must specify a tape label when you RESTORE the data.

DUMP ALL

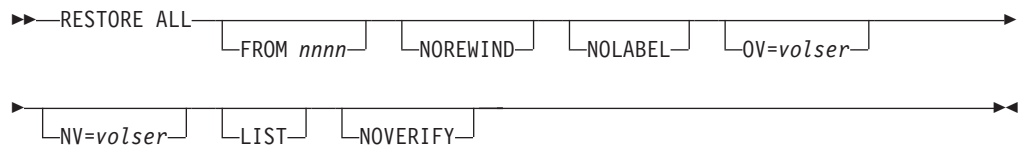
VSE/Fast Copy moves the data to an intermediate data set to be restored later; you can use this intermediate data set **only** for a RESTORE ALL operation.

The following syntax diagrams show the DUMP ALL, RESTORE ALL, and COPY ALL functions. For a description of the parameters used, refer to “Description of VSE/Fast Copy Control Statement Parameters” on page 12.



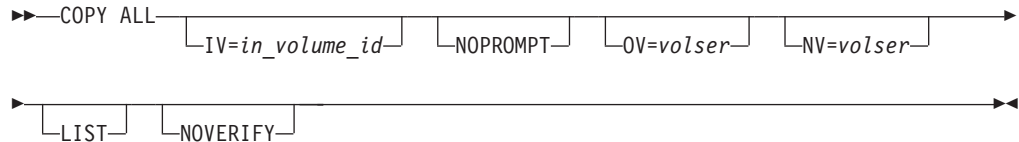
RESTORE ALL

RESTORE ALL restores all data. You can use it, when you have done the backup using DUMP ALL.



COPY ALL

The data is copied from the source disk directly to a target disk, without using an intermediate data set.



DUMP, RESTORE, COPY – VOLUME

The **VOLUME** operand moves data to or from a disk using the Volume Table of Contents (VTOC), that is, disk space which belongs to files recorded in the VTOC. Data which is not registered in the VTOC cannot be copied with the **VOLUME** operand. **VOLUME** does not process unused extents, which results in better performance compared to the **ALL** function.

If your source volume also contains active VSE/VSAM files, process either **all** of these VSE/VSAM files or **none**. Processing all VSE/VSAM files ensures that VSE/Fast Copy can process the VSE/VSAM catalog (if it is located on the same volume) or the VSE/VSAM catalog recovery area together with the data space(s). It ensures that catalog recovery (a function of VSE/VSAM) can be applied after the dumped volume has been restored.

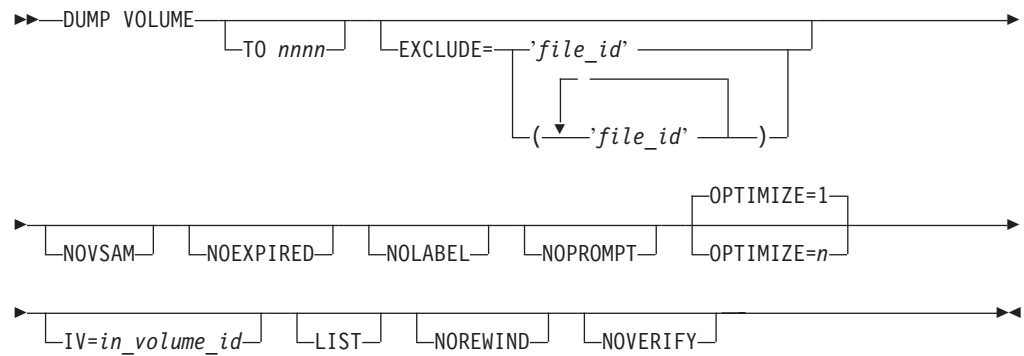
If you use a tape label (TLBL) when you **DUMP** data, you also must specify a tape label when you **RESTORE** the data.

The following syntax diagrams show the **DUMP VOLUME**, **RESTORE VOLUME**, and **COPY VOLUME** functions. For a description of the parameters used, refer to “Description of VSE/Fast Copy Control Statement Parameters” on page 12.

DUMP VOLUME

- For an FBA disk, **DUMP VOLUME** processes the IPL block and the VLB (volume label block).
- For CKD and ECKD disks, it processes track 0. Track 0 contains the IPL BOOTSTRAP records, the volume label of the disk, and the address of the VTOC.
- It processes the VTOC (Volume Table of Contents) and collects information on the limits of the data extents.
- It processes the data extents.

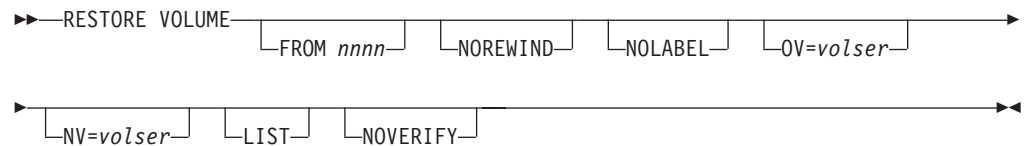
The backup can be used for **RESTORE VOLUME** or **RESTORE FILE** (for VSE/Fast Copy online) to the same or a different disk device of the same device type and with the same or larger capacity.



RESTORE VOLUME

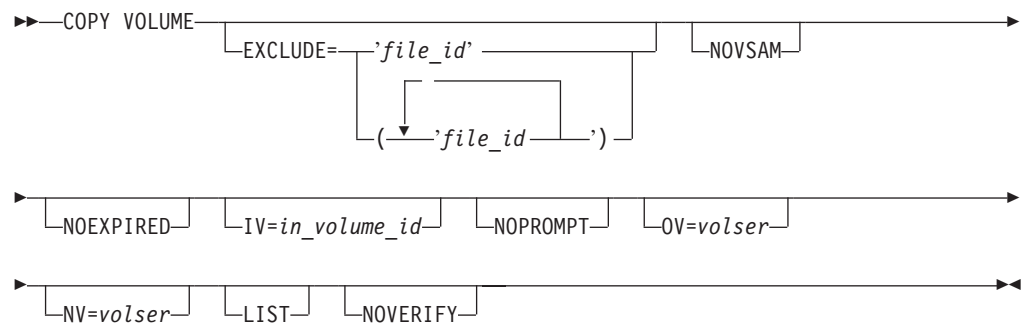
Before VSE/Fast Copy writes any data, it checks for data to be preserved on the target disk; that is, unexpired and/or data-protected files. For any such file, you can request the file identification which VSE/Fast Copy issues via SYSLOG. You may then either cancel the job or proceed with the RESTORE function which deletes any unexpired files.

RESTORE VOLUME is only possible, if you produced the backup with the DUMP VOLUME command.



COPY VOLUME

The data is copied from the source disk directly to a target disk, without using an intermediate data set.



Special VOLUME Functions

Processing Part of a Volume

You can dump or copy part of a volume. For example, you can:

- Copy all of a volume except for file A, file B, and so on.
- Exclude all expired files from dumping.
- Exclude all VSE/VSAM data sets from dumping.

- Exclude data extents for the Page Data Set (PDS), due to performance considerations.

This improves performance because VSE/Fast Copy excludes unwanted files from processing.

Note: For partial COPY VOLUME, the volume identification written on the output disk **must be different** from the one on the input disk.

For a partial DUMP VOLUME or COPY VOLUME with NOEXPIRED or EXCLUDE specified, VSE/Fast Copy does not update the VSE/VSAM-related Format 4 fields in the VTOC. This could cause a timestamp mismatch, and VSE/VSAM may not be able to access its catalog recovery area.

Copying or Restoring to a Larger Volume

VSE/Fast Copy can copy (or restore) data to a disk volume of a larger capacity than the source disk volume if:

- The source and target volumes are of the same type in case of CKD or ECKD disk devices. For example, VSE/Fast Copy can copy from an IBM 3380 Model D to an IBM 3380 Model E, but it cannot copy from an IBM 3390 to an IBM 9345 disk. Also, you cannot copy from an IBM 3380 CKD disk device to an IBM 3380 ECKD disk device and vice versa. The source and target volumes must have the same track layout.
- Both the source and the target volumes are on FBA devices. They need not have the same device type.

VSE/VSAM Restriction:

Do **not** copy one volume to another with larger capacity if the source volume includes VSE/VSAM files. The VSE/VSAM catalog on the target volume will not be compatible with the VSE/VSAM catalog on the source volume.

To bypass the problem, consider:

- A copy run with NOVSAM specified in your COPY VOLUME statement.
- A VSE/VSAM backup/restore run for VSE/VSAM files on the source volume.

DUMP, RESTORE, COPY – FILE

The FILE operand is only available for VSE/Fast Copy online.

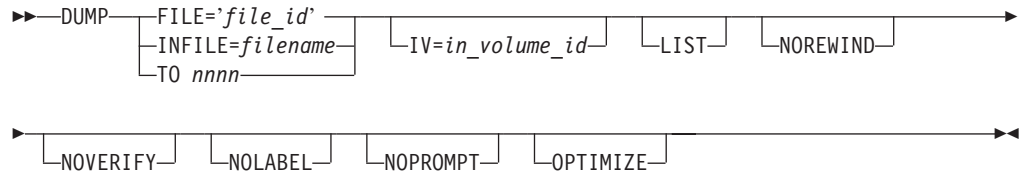
Using the FILE operand you can process a file located on a disk. On the target disk the file occupies the same data extents as on the source disk. If you wish, you can relocate this file ("Reorganization of Files" on page 12 has more information). If the file is access-protected, VSE/Fast Copy notifies you for permit access.

The file you process cannot be a VSE/VSAM data set.

If you use a tape label (TLBL) when you DUMP data, you also must specify a tape label when you RESTORE the data.

DUMP FILE

When you dump a file, the data making up the file (the labels and data extents described by these labels) is copied to the intermediate data set. You can use this data set only as **input** to the RESTORE FILE function to restore the dumped file to a disk.



RESTORE FILE

RESTORE FILE can be used if the dump was created with DUMP VOLUME or DUMP FILE. Before VSE/Fast Copy initiates restoring of the data, the target VTOC is checked for:

- Presence of an unexpired and/or a data-protected file with identical file-id.
- Overlap on any extent with that of an unexpired file.

If any such condition exists, VSE/Fast Copy notifies you and you can either cancel the job or proceed after deleting the conflicting file or renaming the file you want restored.

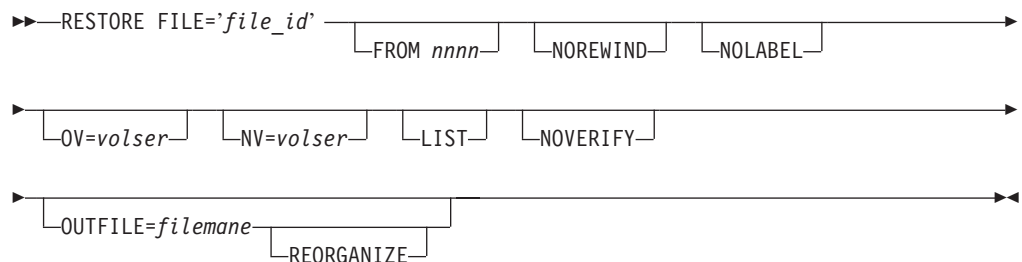
VSE/Fast Copy also checks for:

- Overlap of any extent with the target VTOC.
- Overlap of any extent with that of a data-protected file.

If any such condition exists, VSE/Fast Copy issues a message and cancels the job.

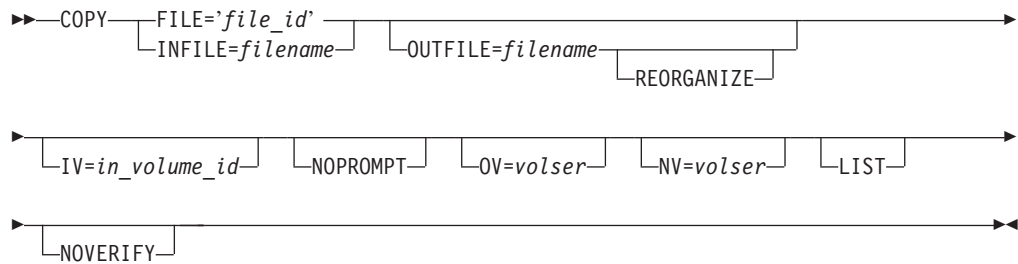
VSE/Fast Copy restores all labels exactly as they were dumped. In particular, it does not modify the creation date and the file serial number.

You can restore individual non-VSE/VSAM files from one DUMP VOLUME to disks of the same type as the source disk. If, during normal operation, one of the files on this disk pack happens to be destroyed, you might want to restore only the destroyed file. VSE/Fast Copy searches for file (*filename*) on the tape and restores it while maintaining the integrity of the other files. VSE/Fast Copy restores the file to exactly the same extents it occupied on the source disk, if you did not request relocation (with the REORGANIZE operand). See “Reorganization of Files” on page 12 for more information.



COPY FILE

VSE/Fast Copy copies the file to exactly the same extents it occupied on the source disk, if you did not request relocation (with the REORGANIZE operand). See “Reorganization of Files” on page 12 for more information.



Special FILE Functions

Handling of Multiple Volume Files

You can copy a single file (specified via DLBL/EXTENT statements) which could span several volumes (multivolume file) to one or several volumes of the same device type, or you can backup (DUMP FILE) an intermediate data set on tape(s) or disk(s). See 30 for more information. Accordingly, you can restore this intermediate data set to one or several volumes of the same device type.

Reorganization of Files

The extents on the output disk are the original extents by default. You can, however, request relocation of the file by:

1. Providing DLBL/EXTENT statements for the target extents and
2. Specifying the keyword REORGANIZE in the RESTORE or COPY statement of VSE/Fast Copy.

Not all files on disk can be relocated during a copy or a restore run of VSE/Fast Copy. For example, a SAM (sequential access method) file or a DAM (direct access method) file with relative record addressing can be relocated. A file cannot be relocated if it includes direct addresses, such as pointers to locations within the file. A typical example of such a file is a DAM file that includes synonym records on overflow tracks.

Description of VSE/Fast Copy Control Statement Parameters

ALL Specifies that all of the data on the involved volume is to be processed.

EXCLUDE='file-id'('file-id1' 'file-id2'...'file-idn')

Not supported for VSE/Fast Copy stand-alone. File IDs *must* be separated by blanks and set in single quotes. It identifies files that are to be excluded during a partial COPY VOLUME or DUMP VOLUME operation.

When specifying EXCLUDE for a partial COPY VOLUME operation, you must also specify the NV=volser operand.

FILE='file-id'

Not supported for VSE/Fast Copy stand-alone. File ID must be set in single quotes. It specifies the identifier of the file to be dumped, restored, or copied. For a restore or a selective restore, the specified identifier is checked against that of the file on the intermediate data set.

FROM nnnn

Not supported for VSE/Fast Copy stand-alone. Applies only to the restore function from an intermediate data set onto an FBA disk. For nnnn, specify either the four-digit IBM device type code for a CKD device, FBA for an FBA device, or TAPE for a tape unit. If you omit the operand, VSE/Fast Copy assumes TAPE.

INFILE=*filename*

Not supported for VSE/Fast Copy stand-alone. Specifies a file function that refers to a DLBL statement and the related EXTENT statement(s). The file name for INFILE and the DLBL statement *must* match. For examples see 29 and 30.

Note: If you copy an ISAM file using the INFILE or OUTFILE keyword, use ISE in the DLBL statement for the input file and ISC for the output file.

IV=*in-volume-id*

Causes the input volume to be checked for a dump or a copy function; it is ignored for a restore function. If the specified volume identifier does not match the identifier recorded on the volume, VSE/Fast Copy informs you by a message. You can then cancel the run or proceed with the same or a different volume.

LIST Requests VSE/Fast Copy to print, on SYSLST, the extents that are copied, dumped, or restored.

NOEXPIRED

Not supported for VSE/Fast Copy stand-alone.

Specifies that all expired files are to be excluded from being dumped or copied.

When specifying NOEXPIRED for a partial COPY VOLUME operation, you must also specify the NV=*volser* operand.

NOLABEL

By specifying the parameter NOLABEL within the VSE/Fast Copy control statement, no standard labels are assigned during dump or restore functions. If you specify NOLABEL during a dump function, then you also must specify it during the restore function of that dump. In the VSE/Fast Copy online environment, it is not necessary to use the UPSI statement when NOLABEL is specified. NOLABEL overwrites the UPSI statement.

NOPROMPT

Applies only to the messages 8F06D and 8F20D, which are displayed when using VSE/Fast Copy to dump or copy disks containing data secured files (for example, VSE/VSAM files). By specifying NOPROMPT in the VSE/Fast Copy control statement, these messages will be suppressed, and VSE/Fast Copy will continue as if PROCEED had been entered.

NOREWIND

When you specify NOREWIND and one intermediate data set is dumped to or restored from tape, VSE/Fast Copy positions the tape at the end of this intermediate data set and does not rewind the tape. If you omit the operand, VSE/Fast Copy rewinds the tape at the beginning and rewinds and unloads the tape at the end of the function which is performed.

Do not use NOREWIND together with alternate tapes. Specify NOREWIND when dumping or restoring multiple volumes to the same tape.

You can have several intermediate data sets (volume or file) on *one* tape using the NOREWIND operand for the dump or for the restore function.

Note: If you use a supervisor that supports the Access Control Facility, you must have specified **standard labels** for each intermediate data set on the tape.

If you want to skip an intermediate data set on tape, you must skip files as follows:

Creation of Intermediate Data Set from	No. of Files to be Skipped	
	Labeled	Unlabeled
CKD/ECKD disk	3	1
FBA disk		
Using VSE/Fast Copy Version 1 Release 1	3	1
Using VSE/Fast Copy Version 1 Release 2 or higher	4	2

NOVERIFY

Applies to the functions dump, restore, and copy to disk. Specifies that verification of output to disk is suppressed. If you omit this operand, VSE/Fast Copy writes and verifies this output.

NOVSAM

Not supported for VSE/Fast Copy stand-alone. Specifies that all VSE/VSAM spaces are to be excluded from being dumped or copied.

When specifying NOVSAM for a partial volume-copy operation, you must also specify the *NV=volser* operand.

NV=volser

Gives the new volume a different serial number. It inserts the new volume identifier into the VOL1 label. The operand is ignored for a dump function.

When specifying the EXCLUDE, NOEXPIRED, or NOVSAM operands for a partial volume-copy operation, you must specify the *NV=volser* operand.

If you omit the operand for a copy or restore, VSE/Fast Copy writes the serial number read from the input disk onto the output disk.

Retain the identifier of the input volume if you want to use the output disk as SYSRES volume, libraries, or data files. To do this, just omit the *NV=volser* keyword from the utility control statement. Do not restore SYSRES to a running system because the labels could be lost.

Files created on a non-DOS/VSE Advanced Functions system cannot always be restored.

OPTIMIZE=1|n

OPTIMIZE is only valid for dump functions. It influences the amount of data taken to the target tape. It has a value of 1,2,3, or 4, where the default value is 1. See Table 4 on page 26 for more information. If you specify a higher value for n, the Fast Copy program can perform faster. At the same time however, the program needs more storage. OPTIMIZE=4 results in the least number of I/O requests; OPTIMIZE=3 generally results in near optimum performance.

Note: OPTIMIZE=3 is often a good compromise between required storage and performance. For CKD, values greater than 1 are only effective for disks with ECKD attachments. For FBA, tape block sizes also are increased.

Using OPTIMIZE With CKD/ECKD Disk Devices:

OPTIMIZE enables you to read multiple tracks per I/O by allowing additional space to allocate buffers. OPTIMIZE is only valid for dump functions and influences the amount of data taken to the target tape.

If OPTIMIZE is specified, it is interpreted only if the input disk is using a storage controller that accepts the READ TRACK channel command word. If READ TRACK is not accepted and OPTIMIZE is specified, OPTIMIZE is ignored.

For cached storage controllers, OPTIMIZE=1 may result in a performance improvement due to sequential caching set in the corresponding ECKD channel program. For OPTIMIZE ≥2, inhibit cache load is used.

Using OPTIMIZE With FBA Disk Devices:

For a dump from an FBA disk device to tape using OPTIMIZE=1, the program has a default block size of approximately 16KB. Specifying a larger block size will improve tape capacity in general. It may improve performance, especially for tape units with streaming capability.

The program requires a **minimum** of three I/O buffers. If the amount of available virtual storage is insufficient to allocate three buffers, the program run will end with a cancel message.

The storage requirement is calculated as follows:

$$\begin{array}{l} \text{Program size (including partition GETVIS space)} \\ + \text{Buffer size multiplied by the number of buffers} \\ \hline = \text{Requirement} \end{array}$$

The program requires about 120KB, and an I/O buffer of about 16KB, multiplied by the value of OPTIMIZE. See Table 4 on page 26 for the values of OPTIMIZE.

OUTFILE=*filename*

Not supported for VSE/Fast Copy stand-alone.

Refers to a DLBL statement and the related EXTENT statement(s). OUTFILE can be specified for a selective restore run only (together with FILE). See also the note under INFILE. The file name for OUTFILE and the DLBL statement *must* match. For examples see 29 and 30.

OV=*volser*

Causes the output volume to be checked before any data transfer takes place. The operand verifies that you mounted the correct output disk volume. It is ignored by the dump function.

REORGANIZE

Not supported for VSE/Fast Copy stand-alone.

Specifies that the extents on the target disk differ in size from the extents used for the original file. The operand applies only to SAM and DAM files with relative track addressing.

A restore run with reorganization fails if the data to be restored includes a file which:

1. Was created on an FBA disk using physical IOCS (file defined by a DTFPH macro), and
2. No CISIZE was specified for the file.

TO *nnnn*

Applies only to the dump function from an FBA disk onto an intermediate data set. For *nnnn*, specify either the four-digit IBM device type code for a CKD device, FBA for an FBA device, or TAPE for a tape unit. If you omit the operand, VSE/Fast Copy assumes TAPE.

VOLUME

Specifies that the desired function is to be performed on a volume.

Integrity of System Files and Shared Disks

Do not use VSE/Fast Copy to dump system files while these files are active. System tasks might modify pointers within these files while VSE/Fast Copy dumps them onto an intermediate data set. If this happens, the files will be inconsistent and unusable after having been restored.

To safeguard data contained in system files, use system-file specific backup and restore functions, some of which are listed below:

Type of System File	Backup/Restore Function
Private library	Librarian BACKUP and RESTORE commands
Library of optional programs	MSHP BACKUP and RESTORE
VSE/POWER queues	POFFLOAD function of VSE/POWER
VSE/ICCF library	VSE/ICCF utility program DTSUTIL

If you do use VSE/Fast Copy to dump a system file, do not restore the file with relocation because this makes the file unusable.

Sharing DASD

When you use shared DASD, the device should only be in use by the system and partition in which VSE/Fast Copy runs.

Restoring onto VM Minidisks

Make sure that the minidisk is cleared during formatting. For example, you can use CMS FORMAT and then run the ICKDSF INIT command. If you have an FBA disk, make sure that you have complete cylinders.

For further details, refer to the IBM manual *Device Support Facilities, User's Guide and Reference*

Calculating Device Capacity for Restore

When dumped data is to be restored, VSE/Fast Copy issues the messages:

```
8F55I DUMPED VOLUME = volid FOR MORE INFORMATION SEE 8F56I
8F56I DATE = mm/dd/yyyy DEVICE = type CAPACITY = capacity
```

For a dump from a minidisk, the displayed device-capacity value is rounded down to the next lower (FBA disk internal) cylinder boundary. The data on the dump tape, however, may extend beyond this capacity limit. Therefore, to avoid a failure of a

restore of this data to a minidisk, be sure to define that disk large enough (the number of displayed blocks plus the number of blocks that represent a cylinder).

The following is an example of how to calculate space requirements:

- Disk device type: *FBA* (744 FBA blocks use up one physical cylinder).
- Device capacity displayed by the message: *11160* FBA blocks.
- Size to be specified for the restore minidisk: *11904* FBA blocks ($11160 + 744$).

VSE/Fast Copy Error Handling

Error Handling for FBA Disk Devices

Any permanent error on an FBA disk device gives you a message with the option to either continue or cancel the job.

If you do not cancel the job, VSE/Fast Copy does the following:

- In case of a permanent error on the **input disk**:
 1. It attempts to read the block range where the error occurred. VSE/Fast Copy performs a series of single block reads in order to save as much data as possible.
 2. It transfers the data of the block in error, as read, to the intermediate data set.

This might make the backup of a single file of the volume unusable. However, it does not impact the backup of all other files, except when the block in error is the volume label block or within the VTOC.

- In case of a permanent error on the **output disk**, VSE/Fast Copy attempts to write the block range where the error occurred by a series of single block writes in order to save as much data as possible. The data of the block in error is left "as is." This might make a single file of the volume useless. However, it does not impact all other files, except when the block in error is the volume label block or within the VTOC.

Error Handling for CKD/ECKD Disk Devices

Any permanent I/O error on a CKD/ECKD input or output disk terminates VSE/Fast Copy. If VSE/Fast Copy abnormally terminates during a copy or a restore function (because of an error on the intermediate data set, for example), you may have to re-initialize your output disk before using it again.

Return Codes

VSE/Fast Copy will set return codes depending on the messages displayed during a VSE/Fast Copy run.

Table 2. Return Codes

Return Code	Message causing Return Code						Meaning
0	8F01I 8F60I	8F02I	8F30D	8F33I	8F55I	8F56I	requested function successfully executed
4	8F06D 8F48I 8F99D 8X12I 8X46D 8X61D	8F20D 8F49I 8FA0I 8X20A 8X52D 8X63D	8F21D 8F50D 8X25D 8X54D 8X64I	8F22I 8F52I 8X42D 8X56D 8X65D	8F23A 8F54D 8X43A 8X58D 8X66D	8F32I 8F98I 8X45D 8X59D 8X75D	at least one message has been displayed
8	8F35I						requested function partially executed

Table 2. Return Codes (continued)

Return Code	Message causing Return Code						Meaning
12	8X03I	8X04I	8X05I	8X06I	8X07I		requested function not executed, i.e. in case of a syntax error
16	8F03I 8F10I 8F24I 8F39I 8F45I 8X02I 8X33I 8X53I	8F04I 8F11I 8F25I 8F40I 8F46I 8X10I 8X44I 8X55I	8F05I 8F12I 8F28I 8F41I 8F47I 8X21I 8X46I 8X57I	8F07I 8F13I 8F36I 8F42I 8F53I 8X22I 8X47I 8X60I	8F08I 8F17I 8F37I 8F43I 8F59I 8X30I 8X50I 8X74I	8F09I 8F19I 8F38I 8F44I 8X32I 8X51I	requested function has been cancelled.

The display of a decision or action message leads to a return code of 4. Depending on the answer given by the user, the return code may change to 12 for a syntax error in the utility control statement, or to 16 if the reply was CANCEL. Entering PROCEED leaves the return code unchanged.

VSE/Fast Copy keeps track of the largest return code reached in a job step and returns this value to Job Control.

Summary of What You Should Consider

Please make sure that you considered the following items:

1. Disk Device Considerations

- When DUMPing, RESTORing or COPYing data, make sure that the target disk is of the same device type and has the same capacity (or more) as the source disk. That means, the target disk must have the same characteristics (FBA) or track layout (CKD/ECKD) as the source disk from which the data was copied. See “Copying or Restoring to a Larger Volume” on page 10 for more information.
- The exact matching between source and target disk is of special importance, if VSE/VSAM files reside on your source disk. VSE/VSAM catalogs are no longer compatible, if disk device capacities are different.
- When VSE/Fast Copy restores or copies data from a disk device to another, the content of the source disk will be the same on the target disk. This is valid for functions that operate VTOC-driven or process the entire disk volume physically (VOLUME and ALL). Any information that resided on the target disk before will be **deleted**.

2. VSE/VSAM Considerations

- VSE/Fast Copy does not copy single VSE/VSAM files. When you use the same disk devices with the same capacity, you can restore or copy the entire data, even if the data contains VSE/VSAM files. If you want to restore or copy single VSE/VSAM files, you may use the *VSE/VSAM Backup/Restore* dialog or an appropriate VSE/VSAM function. For more information, refer to the IBM manual *VSE/VSAM User's Guide and Application Programming*.

3. VSE/Fast Copy Considerations

- Do not run VSE/Fast Copy to dump or restore data which can be modified during the VSE/Fast Copy run (see “Integrity of System Files and Shared Disks” on page 16). System tasks might modify pointers within these files while VSE/Fast Copy dumps them into an intermediate data set. If this happens, **the files will be unusable after having been restored.**
- You must specify the **same** first operand for the RESTORE command as you did for the DUMP command (and vice versa). This means, when you DUMP data with the ALL operand, you also must RESTORE the data with the ALL operand. The same applies for the FILE operand. For VOLUME backup you can do a complete or a selective restore (RESTORE VOLUME or RESTORE FILE).
- When you perform a DUMP from disk to tape, you may either select the tape to be unlabeled or use standard labels for it. When you perform the DUMP using standard labels, you also must perform the RESTORE using standard labels (this also must be done when you perform the DUMP/RESTORE with an unlabeled tape).

Summary of When to Use VSE/Fast Copy Online or Stand-Alone

As described, VSE/Fast Copy stand-alone provides a subset of the VSE/Fast Copy online functions. The following table lists VSE/Fast Copy control statements that are different in the VSE/Fast Copy online and VSE/Fast Copy stand-alone versions:

Table 3. VSE/Fast Copy Control Statements with Description

Control statement:	Examples provided in this manual, on page:	VSE/Fast Copy Online	VSE/Fast Copy Stand- Alone
DUMP ALL	–	YES	YES
RESTORE ALL	–	YES	YES
COPY ALL	–	YES	YES
DUMP VOLUME	28	YES	YES
RESTORE VOLUME	28	YES	YES
COPY VOLUME	–	YES	YES
Partial DUMP VOLUME	27 and 28	YES	NO
Partial COPY VOLUME	–	YES	NO
DUMP FILE	–	YES	NO
RESTORE FILE	–	YES	NO
COPY FILE	–	YES	NO
Selective RESTORE	29	YES	NO
Multivolume DUMP FILE	–	YES	NO
Multivolume RESTORE FILE	–	YES	NO
Multivolume COPY FILE	30	YES	NO

Parameters Not Supported in VSE/Fast Copy Stand-Alone

The following parameters are not supported in VSE/Fast Copy stand-alone. For a description of the parameters refer to “Description of VSE/Fast Copy Control Statement Parameters” on page 12.

FILE parameter

```
EXCLUDE='file-id'('file-id1' 'file-id2'...'file-idn')
FILE='file-id'
FROM nnnn
INFILE=filename
NOEXPIRED
NOVSAM
OUTFILE=filename
REORGANIZE
```

Notes:

1. VSE/Fast Copy stand-alone and VSE/Fast Copy online are compatible, as long as you use functions which are supported in both environments.
2. For using CKD/ECKD or FBA devices to backup data, see also Figure 3 on page 25.

Creating a Backup of the VSE/ESA System (DOSRES and SYSWK1)

When you do a backup of the VSE/ESA system disks (DOSRES and SYSWK1), both disks must be kept consistent. Therefore, it is recommended to do the backup of the two disk devices at the same time (one disk after the other), using VSE/Fast Copy stand-alone.

To do the backup, you can either use:

- DUMP/RESTORE ALL or
- DUMP/RESTORE VOLUME

If you want to do a backup of the VSE/ESA system disks online, you must ensure that the system files (for example, VSE/POWER queues and data files) are **not active**. Therefore, you must shut down CICS, VTAM, and VSE/POWER before executing VSE/Fast Copy online. System task might modify pointers within active files, while VSE/Fast Copy dumps them into an intermediate data set.

Refer to “Step 3 – Performing a Backup of DOSRES on the IPL Tape” on page 47 for an example.

Chapter 3. Running VSE/Fast Copy Online

This chapter includes performance-related information and recommendations for running VSE/Fast Copy online. It also provides examples for it.

VSE/Fast Copy online works with *two (alternating) buffers* for:

- All CKD/ECKD applications.
- FBA applications as follows:
 - Involving an intermediate data set on disk.
 - With file reorganization specified.
 - If the copy function is used.

VSE/Fast Copy online works with *multibuffering* (up to 24 buffers depending on the size of the partition used by VSE/Fast Copy) for FBA applications not listed above. Multibuffering results in better performance:

- In a shared-channel environment (source and target devices are on the same channel.)
- With an increasing number of defective tracks on the disk.

Job Control Statements to be Entered

You need certain job control statements to start a VSE/Fast Copy run:

1. EXEC statement

The program is invoked by an // EXEC statement as shown below, depending on the involved disk device type:

```
For CKD or ECKD disks: // EXEC FCOPY
For FBA disks:         // EXEC FCOPYFB
```

Note: After the EXEC statement you can specify one or more VSE/Fast Copy control statements.

2. UPSI statement

You may have to supply a // UPSI statement to indicate whether the intermediate data set has standard labels or no labels. Instead of the // UPSI statement, the VSE/Fast Copy control statement NOLABEL can be used. See NOLABEL on page 13 for more details.

- Bit 0 is associated with the input tape in a restore operation:
 - ON (100) to indicate unlabeled tape.
 - OFF (000) to indicate standard labels, which is the default.
- Bit 2 is associated with the output tape in a dump operation:
 - ON (001) to indicate unlabeled tape.
 - OFF (000) to indicate standard labels, which is the default.

3. DLBL and EXTENT, TLBL statements

Use UIN as file name for a restore function and UOUT as file name for a DUMP function.

Use // DLBL and // EXTENT statements for UIN or UOUT if the intermediate data set is located on a disk (FBA only).

Use a // TLBL statement for the respective file name if a labeled tape is to be read or written.

Device Assignments and Label Information

Figure 2 shows the device assignments and label information which you must provide.

Required Device Assignments	SYS004	SYS005	SYSnnn
Dump to intermediate data set: source disk target tape * target disk	X	X	X
Copy from disk to disk: source disk target disk	X	X	
Restore from tape: source tape target disk	X	X	
Restore from disk: * source disk target disk		X	X
Note: * applies only if the source device is an FBA disk.			

Figure 2. Device Assignments and Label Information (Part 1 of 2)

Required Label-information	File name is		Statement is	
	UIN	UOUT	TLBL	DLBL
Dump to intermediate data set: target tape (UPSI bit 2=0) * target disk		X X	X	X
Copy from disk to disk:	n/a	n/a	n/a	n/a
Restore from tape: source tape (UPSI bit0=0)	X		X	
Restore from disk: * source (intermediate) disk target disk	X	X		X X
Note: * applies only if the source for the intermediate data set was an FBA disk device. n/a means not applicable				

Figure 2. Device Assignments and Label Information (Part 2 of 2)

The logical unit (SYS*nnn*) for an intermediate data set on disk can be any programmer logical unit that is defined in the partition.

If you specify a file via DLBL/EXTENT statements (if you do a RESTORE FILE or selective RESTORE and request relocation, for example, or have a multivolume file processed), you indicate this by the INFILE or the OUTFILE operand in the utility control statement. Such EXTENT statements then can refer to any programmer logical units (SYS*nnn*) that are defined in the partition.

For FBA disk devices or VSAM-owned volumes, VSE/Fast Copy uses the CFSIZE value which is recorded in the format-1 label for the input file. If you specify a CFSIZE for the output file, it must be the same as the CFSIZE given in the format-1 label of the input file.

Include CFSIZE=16384 on the DLBL UOUT statement to write an intermediate data set on an FBA disk device.

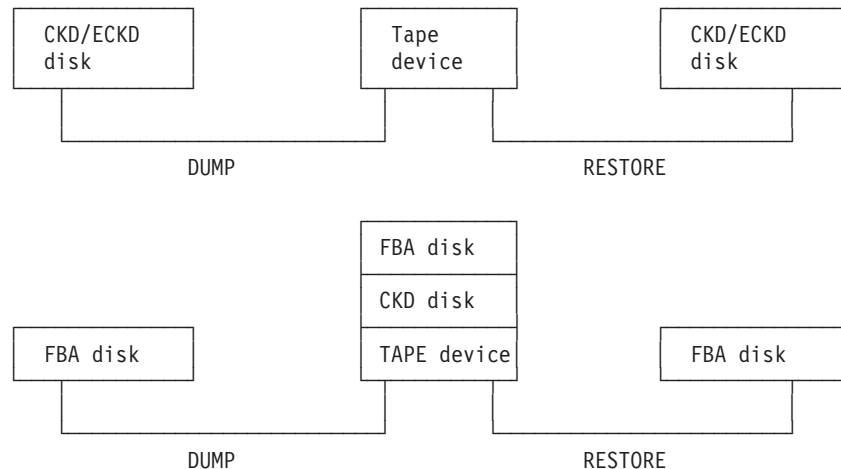


Figure 3. Using CKD/ECKD or FBA Disk Devices with VSE/Fast Copy Online

Accepting Control Statements from SYSLOG

VSE/Fast Copy will accept control statements from SYSLOG. After invoking VSE/Fast Copy, you will be prompted to enter a control command or the keyword END, which terminates the VSE/Fast Copy session.

The given control statement is scanned for syntax errors. If one is detected, you are prompted to re-enter the control statement. You may re-enter multiple commands during a session, provided the // ASSGN, // DLBL, // TLBL and // UPSI statements are still appropriate.

For example, a possible console input for a VSE/Fast Copy is:

```
// ASSGN SYS004,151
// ASSGN SYS005,152
// EXEC FCOPY
```

VSE/Fast Copy displays the following message:

```
8F30D ENTER FAST COPY CONTROL STATEMENT OR END
```

Now a control statement can be entered, for example:

```
COPY FILE='file-id'
```

After VSE/Fast Copy executes the command, it displays message 8F30D again to prompt for a second command or END.

Storage and Run Requirements to Improve Performance

VSE/Fast Copy runs in any partition below 16MB of a VSE/ESA system, taking into account storage and run requirements. You may run VSE/Fast Copy in real mode to minimize run time. This requires you to specify:

- A minimum partition size in an ALLOC R (or ALLOCR) command before the program run. The partition size increases depending on the value of OPTIMIZE. See the following table for possible values of OPTIMIZE.
- The corresponding value for program size in the SIZE operand of the EXEC statement that starts the run, depending on whether you are using a CKD or an FBA disk device. See the following table.

Table 4. Values of OPTIMIZE for Storage and Run Requirements

OPTIMIZE	For CKD	For FBA
OPTIMIZE=1	$200K^1 + 2 * \text{track capacity}$	$200K^1 + (n * \text{OPTIMIZE} * 16K^2)$
OPTIMIZE=2	$200K + 4 * \text{track capacity}$	n=number of buffers desired; it can be any number from 2 to 24.
OPTIMIZE=3	$200K + 6 * \text{track capacity}$	
OPTIMIZE=4	$200K + 10 * \text{track capacity}$	

¹Where 200K is the program size, including partition GETVIS space.

²Where 16K is the size of an I/O buffer.

Note: VSE/Fast Copy for FBA devices uses multibuffering when the intermediate data set is on tape. In this case, you can implicitly control the number of buffers by selecting a partition size based on the above formula.

Space Requirements for an Intermediate Data Set on Disk

The space you need depends on:

- The amount of data (the number of FBA blocks to be dumped).
- The device type of the intermediate data set.

VSE/Fast Copy uses the largest possible blocking on CKD disk devices.

The table below shows how many FBA blocks (of 512 bytes each) fit, for example, into a 3380 CKD track or into an FBA control interval as used by VSE/Fast Copy:

Disk for Intermediate Data Set	Blocks per Track	Blocks per Control Interval
IBM 3380	93	—
IBM FBA	—	31

Running in Virtual or Real Mode (Performance Considerations)

For optimum performance of VSE/Fast Copy, consider running the program in real mode. If VSE/Fast Copy runs in virtual mode, it attempts to fix I/O related storage.

VSE/Fast Copy may be unable to fix required storage because not enough page frames are available. If this occurs you get the message:

FIXING OF I/O STORAGE FAILED.

You should then proceed as recommended for the message in the *VSE/ESA Messages and Codes* manual.

The amount of storage required for page fixing varies with the type of the disk device from which VSE/Fast Copy reads or to which VSE/Fast Copy writes.

Reading from and Writing to an FBA Disk Device

VSE/Fast Copy can operate with two or more (up to 24) buffers if:

1. The partition for VSE/Fast Copy is large enough.
2. The intermediate data set is not on tape.
3. No file reorganization is requested.

This reduces run time, especially if the source and target disks are on the same channel or if you have an increasing number of defective blocks on your FBA disk device.

Examples for Running VSE/Fast Copy Online

The examples in this section assume that the sublibrary holding the VSE/Fast Copy program is accessible from the partition that is being used. Normally, this is done by a permanent LIBDEF statement.

To make use of the examples, substitute your own values for the lowercase values, names, and device addresses. The statement numbers in the job and utility control statements point to the numbered explanations after each example.

Example 1 – Copy Partial FBA Volume

Copy partial volume excluding all expired files.

```
// JOB jobname
(1) // ASSGN SYS004,140
(2) // ASSGN SYS005,141
(3) // EXEC FCOPYFB
(4) COPY VOLUME IV=SYSRES OV=xxxxxx NV=zzzzzz NOVERIFY -
    NOEXPIRED
(5) /*
(5) /&
```

1. Assigns SYS004 to FBA input disk on 140.
2. Assigns SYS005 to FBA output disk on 141.
3. Starts the VSE/Fast Copy program for FBA disk.
4. Causes VSE/Fast Copy to:
 - a. Copy the files.
 - b. Check the input and output volume.
 - c. Change the volume serial number.
 - d. Exclude all expired files.

All of this without write-and-check.

5. Ends utility control input and job.

Example 2 – Dump a CKD/ECKD Volume

Dumping a CKD/ECKD volume to an unlabeled tape.

```
// JOB jobname
(1) // UPSI 101
(2) // ASSGN SYS004,230
(3) // ASSGN SYS005,280
(4) // EXEC FCOPY
(5) DUMP VOLUME LIST
(6) /*
(6) /&
```

1. Indicates unlabeled tape for both the Dump and the Restore operation.
2. Assigns SYS004 to CKD/ECKD input disk on 230.
3. Assigns SYS005 to output tape on 280.
4. Starts the VSE/Fast Copy CDK program.
5. Dumps the volume to tape and prints the extent limits on SYSLST.
6. Ends utility control input and job.

Note: This same function can be achieved by omitting (1) and adding the parameter NOLABEL to (5).

Example 3 – Dump a Volume with EXCLUDE and OPTIMIZE

Dumping a volume excluding VSAM spaces, the SYSRES extent, and the extents of a work file.

```
// JOB jobname
(1) // ASSGN SYS004,140
(2) // ASSGN SYS005,181
(3) // ASSGN SYS005,182,ALT
(4) // TLBL UOUT,'SYSRES.BACKUP.PRT',2001/365
(5) // EXEC FCOPYFB
(6) DUMP VOLUME IV=SYSRES EXCLUDE=('DOS.SYSRES.EXTENT' -
'DOS.WORKFILE.NO1') NOVSAM OPTIMIZE=3
(7) /*
(7) /&
```

1. Assigns SYS004 to FBA input disk on 140.
2. Assigns SYS005 to output tape on 181.
3. Assigns alternate tape drive to 182.
4. Specifies standard label output tape.
5. Starts the VSE/Fast Copy program for FBA disk.
6. Dumps the volume SYSRES excluding VSAM spaces, SYSRES extent, and work file.
7. Ends utility control input and job.

Note: OPTIMIZE=3 is often a good compromise between required storage and performance. The tape block size is increased to 48K.

Example 4 – Restore a Volume

Restoring a volume from the tape created by the job in Example 3.

```
// JOB jobname
(1) // ASSGN SYS004,181
(2) // ASSGN SYS004,182,ALT
(3) // ASSGN SYS005,141
(4) // TLBL UIN,'SYSRES.BACKUP.PRT',2001/365
```

```

(5) // EXEC FCOPYFB
(6) RESTORE VOLUME FROM TAPE OV=WORK01
(7) /*
(7) /&

```

1. Assigns SYS004 to input tape on 181.
2. Assigns alternate tape drive to 182.
3. Assigns SYS005 to FBA output disk on 141.
4. Specifies standard label input tape.
5. Starts the VSE/Fast Copy program for FBA disk.
6. Restores the volume from tape to disk and checks if the volume with the serial number WORK01 is mounted.
7. Ends utility control input and job.

Example 5 – Selective Restore to Same Location

Selective restore of a file with the file-id 'A.Ledger' from the above tape(s) to its original location.

```

// JOB jobname
(1) // ASSGN SYS004,181
(2) // ASSGN SYS004,182,ALT
(3) // ASSGN SYS005,141
(4) // TLBL UIN,'SYSRES.BACKUP.PRT',2001/365
(5) // EXEC FCOPYFB
(6) RESTORE FILE='A.LEDGER' OV=SYSRES NOVERIFY
(7) /*
(7) /&

```

1. Assigns SYS004 to input tape on 181.
2. Assigns alternate tape drive on 182.
3. Assigns SYS005 to output FBA disk on 141.
4. Specifies standard label input tape.
5. Starts the VSE/Fast Copy program for FBA disk.
6. Restores the file from tape to SYSRES with no write-and-check.
7. Ends utility control input and job.

Example 6 – Selective Restore to Other Location (Using REORGANIZE)

Selective restore with the file-id 'A.Ledger' from the above tape(s) to two extents on two FBA disk devices.

```

// JOB jobname
(1) // ASSGN SYS004,181
(2) // ASSGN SYS004,182,ALT
(3) // ASSGN SYS020,140
(4) // ASSGN SYS021,141
(5) // DLBL OLDFIL,'A.LEDGER.OLDSTATUS',2001/365
(6) // EXTENT SYS020,SYSRES,1,0,120000,2000
(7) // EXTENT SYS021,WORK01,1,1,200,5000
(8) // TLBL UIN,'SYSRES.BACKUP.PRT',2001/365
(9) // EXEC FCOPYFB
(10) RESTORE FILE='A.LEDGER' OUTFILE=OLDFIL REORGANIZE
(11) /*
(11) /&

```

1. Assigns SYS004 to input tape on 181.
2. Assigns alternate tape drive on 182.
3. Assigns SYS020 to first FBA output disk on 140.
4. Assigns SYS021 to second FBA output disk on 141.

5. Defines target file (where, in this example, file name is OLDFIL and 'file-id' is 'A.LEDGER.OLDSTATUS').
6. Describes first extent.
7. Describes second extent.
8. Specifies standard label input tape.
9. Starts the VSE/Fast Copy program for FBA disk.
10. Restores the file to two extents referring to file name 'OLDFIL' and reorganizing the output.
11. Ends utility control input and job.

Example 7 – Copy a Multivolume File

Copying a sequential multivolume file located on two volumes to a file located on four volumes.

```
// JOB jobname
(1) // ASSGN SYS006,231
(2) // ASSGN SYS007,232
(3) // ASSGN SYS010,231
(3) // ASSGN SYS011,232
(3) // ASSGN SYS012,233
(3) // ASSGN SYS013,234
(4) // DLBL FILE1,'PAYR.FILE.1',2001/365,SD
(5) // EXTENT SYS006,,1,000,00100,1900
(6) // EXTENT SYS007,,1,000,00100,1900
(7) // DLBL FILE2,'PAYR.FILE.2',2001/365,SD
(8) // EXTENT SYS010,,1,000,03800,1900
(8) // EXTENT SYS011,,1,001,03800,1900
(8) // EXTENT SYS012,,1,002,03800,1900
(8) // EXTENT SYS013,,1,003,03800,1900
(9) // EXEC FCOPY
(10) COPY INFILE=FILE1 OUTFILE=FILE2 REORGANIZE NOVERIFY LIST
(11) /*
(11) /&
```

1. Assigns SYS006 to first input disk on 231.
2. Assigns SYS007 to second input disk on 232.
3. Assigns SYS010 through SYS013 to the four output disks on 231 through 234.
4. Defines the first input file (where, in this example, file name is FILE1 and 'file-id' is 'PAYR.FILE.1').
5. Describes the first extent of the input disk.
6. Describes the second extent of the input disk.
7. Defines the output file (where, in this example, file name is FILE2 and 'file-id' is 'PAYR.FILE.2').
8. Describes the four output extents.
9. Starts the VSE/Fast Copy program for CKD disk.
10. Copies the two-volume file 'FILE1' to a four-volume file 'FILE2'. It reorganizes, does not verify disk output, and lists the extent limits.
11. Ends utility control input and job.

Chapter 4. VSE/ESA Stand-Alone Environment (Including VSE/Fast Copy Stand-Alone)

The stand-alone environment (IPL from tape) is a special purpose VSE/ESA environment and serves mainly for:

- Initial installation
- Data saving
- Disk initialization
- Tape and disk handling

The following utilities can be used in the stand-alone environment:

Table 5. Overview of Stand-Alone Environment

Stand-Alone utility:	Described in:
VSE/Fast Copy stand-alone	"VSE/Fast Copy Control Statements (Online and Stand-Alone)" on page 7, "Using VSE/Fast Copy Stand-Alone" on page 33
RESTORE	<i>VSE/ESA Guide to System Functions</i>
Device Support Facilities (ICKDSF)	<i>Device Support Facilities User's Guide and Reference</i>
DITTO/ESA for VSE	<i>DITTO/ESA User's Guide and Reference</i>

After having used the stand-alone utilities, you may consider performing a software REIPL from disk. In this case, you are prompted for the *cuu* from which you want to perform the IPL.

Functional Characteristics of the Stand-Alone Environment

- You perform an IPL of VSE/ESA from a *stand-alone tape*, which you have previously created online with the Librarian BACKUP command. You can use the VSE/ESA installation tape for stand-alone operation. In this case, you have to answer NO to the system prompt asking, if you want to do an automatic installation.
- You can call the utilities shown in Table 5 in the sequence you want. For example, after you have invoked ICKDSF, you can invoke the Librarian RESTORE program, then DITTO/ESA, and finally VSE/Fast Copy stand-alone.
- In a stand-alone environment VSE/ESA runs without the system library IJSYSRS.
- The storage layout in the stand-alone environment corresponds to a minimum size of a VSE/ESA system:
 - The stand-alone programs run in a predefined single partition (BG).
 - As minimum, 8MB processor storage is required.

You cannot change these values.

- You request the stand-alone utilities to be executed by responding to messages from the system console. Alternatively, a predefined sequence of responses, assembled in a customization table and stored on the stand-alone tape, is processed. "Customizing Stand-Alone Utilities Using the Customization Table" on page 41 provides more information.
- All messages and responses are optionally logged to a printer.
- Messages issued in the stand-alone environment are prefixed, just like in the online environment (IPL from disk) by a partition ID and a reply ID, although only

one partition (BG) is active. If you want to enter a reply, you have to specify the reply ID in the same way as in the online environment. If you do not enter a reply ID, the system handles the reply as an Attention Routine (AR) command.

- The following subset of AR commands is supported in the stand-alone environment:
 - AUTOIPL (query function only)
 - CACHE (query function only)
 - CANCEL
 - DUMP
 - GETVIS
 - OFFLINE
 - ONLINE
 - MTC
 - REPLID
 - STATUS
 - VOLUME
- The stand-alone environment mainly supports the same devices that are supported by VSE/ESA. For a list of supported devices and restrictions refer to “Device Support” on page 87.

Performing an IPL from the Stand-Alone Tape

When you want to perform an IPL from a stand-alone tape, you are prompted from VSE/ESA. You may use two different types of tapes:

1. The **VSE/ESA installation tape**

If you perform a stand-alone IPL from it, the following message appears on your screen:

```
BG 0000 SI01D DO YOU WANT TO DO AN AUTOMATIC INSTALLATION (YES/NO)?  
BG-0000
```

Reply **0 no** to get the prompt for the stand-alone environment. After having entered 0 no, the stand-alone programs are loaded, as shown in Figure 4 on page 33.

2. A **stand-alone tape that you have created using the online Librarian BACKUP command (without CUSTTABLE parameter).**

If you perform an IPL from it, the message SA01I appears on your screen and the stand-alone programs are loaded (as shown in the following figure).

```
BG 0000 SA01I ***** STAND ALONE PROGRAMS LOADED *****
BG 0000 SA02D IF YOU WANT A LISTING, SPECIFY CUU OF PRINTER, ELSE (ENTER)
BG-0000
```

This message prompts for a printer address. You can enter the cuu of a printer, if you want a listing for all following stand-alone activities. Enter **0** and press END/ENTER, if you do not want a listing. Please note that it is not possible to specify a printer later during utility selection. If you specify a printer address, processing continues with the prompt for a printer device type.

You may enter: **0 00e**

```
BG 0000 SA03D DEVICE TYPE IS nnnn. ACCEPT (ENTER) OR SPECIFY ALTERNATE TYPE
BG-0000
```

Enter **0**

```
BG 0000 SA08D DATE IS mm/dd/yyyy. ACCEPT DATE (ENTER) OR SPECIFY DATE MM/DD/YYYY
BG-0000
```

Accept system date as job date and enter: **0**

```
BG 0000 SA09I SELECT ONE OF THE FOLLOWING PROGRAMS, OR TYPE END
BG 0000 SA10D FASTCOPY, RESTORE, ICKDSF, DITTO, REIPL
BG-0000
```

This message prompts you to select a utility. If you want to continue, specify one of the listed utilities. If you select REIPL, stand-alone processing terminates with a software REIPL from disk.

If you want to perform a VSE/Fast Copy stand-alone run, enter:

0 FASTCOPY

You can terminate the stand-alone processing (without a software REIPL), if you enter:

0 END

Figure 4. Loading Stand-Alone Programs

The following section describes the utility VSE/Fast Copy stand-alone.

Using VSE/Fast Copy Stand-Alone

After having invoked VSE/Fast Copy stand-alone (as shown in Figure 4) the program prompts you for the following information:

- Address (cuu) and type of the input device.
If the input device is a tape device, the utility prompts you for the address of an alternate tape. This may be specified if the input is on two or more tape volumes. Press END/ENTER if no alternate tape is required.
- Address (cuu) and type of the output device.
If the output device is a tape device, the utility prompts you for the address of an alternate tape. This may be specified if the output will be on two or more tape volumes. Press END/ENTER if no alternate tape is required.
- For performing a DUMP or RESTORE, you are prompted for tape label (TLBL) information. When you enter a TLBL statement, the filename on the tape is UOUT for a DUMP function and UIN for a RESTORE function.

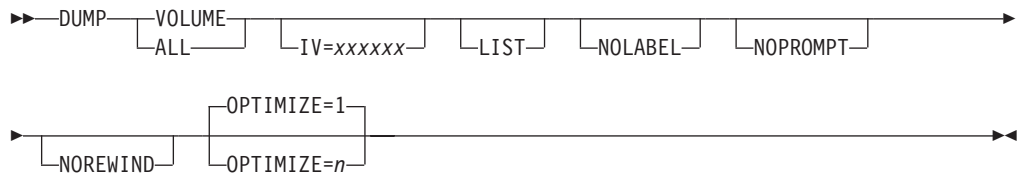
VSE/Fast Copy Stand-Alone Control Statements

The utility control statements, one for each major function, contain the information required to run the utility.

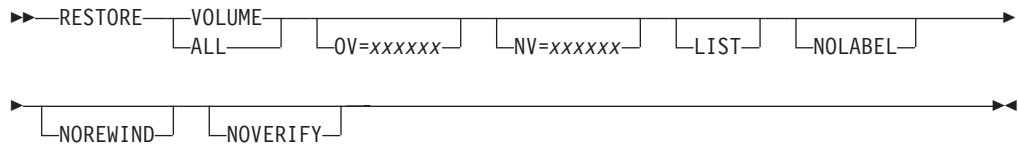
You must enter the function you want to perform and the operands on one line from position 1 to 72. Indicate continuation in a subsequent card image by a hyphen separated by a blank character following the last specification in the current card image. You can make a line break only between operands, not within operands.

The VSE/Fast Copy stand-alone control statement formats are shown below. For a description of the parameters, refer to “Description of VSE/Fast Copy Control Statement Parameters” on page 12.

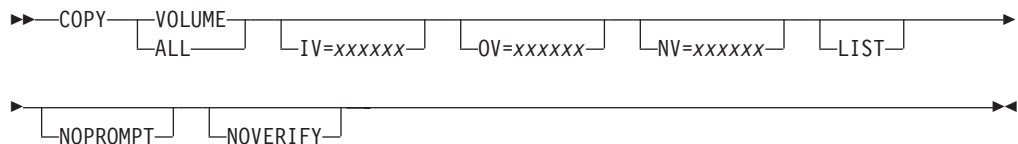
DUMP Statement (Stand-Alone)



RESTORE Statement (Stand-Alone)



COPY Statement (Stand-Alone)



Examples for Running VSE/Fast Copy Stand-Alone

The following samples show you how to use the VSE/Fast Copy stand-alone.

Example 1 - Dump a Volume Onto Tape

This example shows the prompting sequence for a disk to be dumped to a labeled tape. The utility prompts you for the control statements. In case of a syntax error, it prompts you again. If the device assignments are not consistent with the specified functions, the job is canceled.

If you enter a specific input or output device type, do so left-justified. The lines preceded by an 'I' show your input; all others are system output.

```

BG 0000 SA01I ***** STAND ALONE PROGRAMS LOADED *****
BG 0000 SA02D IF YOU WANT A LISTING, SPECIFY CUU OF PRINTER, ELSE (ENTER)
BG-0000
I 0 00e
BG 0000 SA03D DEVICE TYPE IS 1403 . ACCEPT (ENTER) OR SPECIFY ALTERNATE TYPE
BG-0000
I 0
BG 0000 SA08D DATE IS 08/17/1998.ACCEPT DATE (ENTER) OR SPECIFY DATE MM/DD/YYYY
BG-0000
I 0
BG 0000 SA09I SELECT ONE OF THE FOLLOWING PROGRAMS, OR TYPE END
BG 0000 SA10D FASTCOPY, RESTORE, ICKDSF, DITTO, REIPL
BG-0000
I 0 fastcopy
BG 0000 SA11D SPECIFY ADDRESS OF INPUT DEVICE CUU
BG-0000
I 0 110
BG 0000 SA03D DEVICE TYPE IS 3380 . ACCEPT (ENTER) OR SPECIFY ALTERNATE TYPE
BG-0000
I 0
BG 0000 SA12D SPECIFY ADDRESS OF OUTPUT DEVICE CUU
BG-0000
I 0 480
BG 0000 SA03D DEVICE TYPE IS 3480 . ACCEPT (ENTER) OR SPECIFY ALTERNATE TYPE
BG-0000
I 0
BG 0000 SA13D TAPE MODE IS 00. ACCEPT (ENTER) OR SPECIFY ALTERNATE MODE YY
BG-0000
I 0
BG 0000 SA14D SPECIFY ADDRESS OF ALTERNATE TAPE (CUU), OR (ENTER)
BG-0000
I 0
BG 0000 SA15D FOR TAPE LABEL CHECKING SPECIFY // TLBL, ELSE (ENTER)
BG-0000
I 0 // t1b1 uout,'ckd.dump'
BG 0000 8F30D ENTER FASTCOPY CONTROL STATEMENT OR END
BG-0000
I 0 dump volume list
BG 0000 DUMP VOLUME LIST
BG 0000 8F01I PROCESSING STARTS
BG 0000 8F02I END OF PROCESSING.      420 TRACKS DUMPED
BG 0000 SA09I SELECT ONE OF THE FOLLOWING PROGRAMS, OR TYPE END
BG 0000 SA10D FASTCOPY, RESTORE, ICKDSF, DITTO, REIPL
BG-0000
I 0 end
BG 0000 SA17W ***** END OF STAND ALONE PROCESSING *****
BG-0000
```

Example 2 - Restore a Volume

This example shows the prompting sequence for a CKD disk dumped to the IPL tape (unlabeled, stored behind the VSE/Fast Copy stand-alone utility) to be restored to a CKD disk. The IPL address of the tape is 481. The data to be restored is assumed to be on the same tape. In this example message 8F34D is issued, because the same cuu is used for IPL from tape.

Note: If you want to enter a specific input or output device type, do so left-justified.

The lines preceded by an 'I' show your input; all others are system output.

```

BG 0000 SA01I ***** STAND ALONE PROGRAMS LOADED *****
BG 0000 SA02D IF YOU WANT A LISTING, SPECIFY CUU OF PRINTER, ELSE (ENTER)
BG-0000
I 0 00e
BG 0000 SA03D DEVICE TYPE IS 1403 . ACCEPT (ENTER) OR SPECIFY ALTERNATE TYPE
BG-0000
I 0
BG 0000 SA08D DATE IS 08/17/1998.ACCEPT DATE (ENTER) OR SPECIFY DATE MM/DD/YYYY
BG-0000
I 0
BG 0000 SA09I SELECT ONE OF THE FOLLOWING PROGRAMS, OR TYPE END
BG 0000 SA10D FASTCOPY, RESTORE, ICKDSF, DITTO, REIPL
BG-0000
I 0 fast
BG 0000 SA11D SPECIFY ADDRESS OF INPUT DEVICE CUU
BG-0000
I 0 481
BG 0000 SA03D DEVICE TYPE IS 3480 . ACCEPT (ENTER) OR SPECIFY ALTERNATE TYPE
BG-0000
I 0
BG 0000 SA14D SPECIFY ADDRESS OF ALTERNATE TAPE (CUU), OR (ENTER)
BG-0000
I 0
BG 0000 SA12D SPECIFY ADDRESS OF OUTPUT DEVICE CUU
BG-0000
I 0 241
BG 0000 SA03D DEVICE TYPE IS 3380 . ACCEPT (ENTER) OR SPECIFY ALTERNATE TYPE
BG-0000
I 0
BG 0000 SA15D FOR TAPE LABEL CHECKING SPECIFY // TLBL, ELSE (ENTER)
BG-0000
I 0
BG 0000 8F30D ENTER FASTCOPY CONTROL STATEMENT OR END
BG-0000
I 0 restore volume norewind
BG 0000 RESTORE VOLUME NOREWIND
BG 0000 8F34D DUMP DATA ON IPL TAPE? (YES/NO). DEFAULT IS NO.
BG-0000
I 0 yes
BG 0000 8F55I DUMPED VOLUME = ESA141. FOR MORE INFORMATION SEE MESSAGE 8F56I.
BG 0000 8F56I DATE = 08/17/1998. DEVICE = 3380 . CAPACITY = 50
CYLINDERS.
BG 0000 8F01I PROCESSING STARTS
BG 0000 8F02I END OF PROCESSING. 420 TRACKS RESTORED
BG 0000 SA09I SELECT ONE OF THE FOLLOWING PROGRAMS, OR TYPE END
BG 0000 SA10D FASTCOPY, RESTORE, ICKDSF, DITTO, REIPL
BG-0000
0 end
BG 0000 SA17W ***** END OF STAND ALONE PROCESSING *****
BG-0000
```

Overview of VSE Librarian RESTORE (Stand-Alone Environment)

The stand-alone RESTORE function allows to restore a single SYSRES file. This file may be stored on a distribution tape, for example, or on any tape created by using the Librarian backup functions with RESTORE=STANDALONE specified, or in the first backup file on the input tape.

If the tape contains more than one SYSRES file in the first backup file on the input tape, you have to select *one file*. If private libraries, sublibraries, or members are part of the backup tape, you can restore them online. The backup tape created for stand-alone restore need not include a SYSRES file. The SYSRES file can be in the first backup file on another tape.

The following example shows you how to proceed, after you have invoked the Librarian RESTORE function.

Example of a Librarian Restore Run

This example shows you a communication example of a restore run with an unlabeled tape.

The lines preceded by an "I" show your input; all others are system output.

```
BG 0000 SA01I ***** STAND ALONE PROGRAMS LOADED *****
BG 0000 SA02D IF YOU WANT A LISTING, SPECIFY CUU OF PRINTER, ELSE BYPASS
I (ENTER)
BG-0000
I 0
BG 0000 SA08D DATE IS 03/12/1998. ACCEPT DATE (ENTER) OR SPECIFY DATE MM/DD/YYYY
BG-0000
I 0
BG 0000 SA09I SELECT ONE OF THE FOLLOWING PROGRAMS, OR TYPE END
BG 0000 SA10D FASTCOPY, RESTORE, ICKDSF, DITTO, REIPL
BG-0000
I restore
BG 0000 SA11D SPECIFY ADDRESS OF INPUT DEVICE CUU
BG-0000
I 0 493
BG 0000 SA03D DEVICE TYPE IS 3480. SPECIFY ALTERNATE TYPE, OR ACCEPT (ENTER)
BG-0000
I 0
BG 0000 SA15D IF TAPE LABEL CHECKING IS DESIRED SPECIFY // TLBL, ELSE BYPASS (ENTER)
BG-0000
I 0
BG 0000 SA16D SPECIFY ADDRESS OF SYSRES DISK CUU
BG-0000
I 0 863
BG 0000 SA03D DEVICE TYPE IS 3380. SPECIFY ALTERNATE TYPE, OR ACCEPT (ENTER)
BG-0000
I 0
BG 0000 L302A ENTER YES TO RESTORE SYSRES FILE IJSYSRS OR NO TO SKIP TO NEXT SYSRES
BG-0000
I 0 yes
BG 0000 L315I ORIGINAL FILE ID= VSE.SYSRES.LIBRARY
BG 0000 L316A ENTER YES TO KEEP OR NO TO RESPECIFY THE SYSRES FILE ID
BG-0000
I 0 yes
```

```

BG 0000 L309I ORIGINAL ALLOCATION=      954 TRACKS =      63 CYLINDERS 9 TRACKS
BG 0000 L310A ENTER YES TO KEEP OR NO TO RESPECIFY THE ALLOCATION
BG-0000
I 0 no
BG 0000 L312I MINIMUM ALLOCATION=      667 TRACKS =      44 CYLINDERS 7 TRACKS
BG 0000 L304I ENTER THE DESIRED ALLOCATION AS NUMBER OF TRACKS OR CYLINDERS
TRACKS
BG 0000 L313A ALLOC=
BG-0000
I 0 959
BG 0000 L329A ENTER YES TO RESTORE ALL SUBLIBRARIES OR NO FOR SELECTIVE
RESTORE
BG-0000
I 0 yes
BG 0000 L338I SUMMARY OF RESTORE PARAMETERS:
BG 0000 L318I FILE NAME = IJSYSRS
BG 0000 L319I FILE ID = VSE.SYSRES.LIBRARY
BG 0000 L321I ALLOCATION=      959 TRACKS
BG 0000 L344I START= CYLINDER 0 TRACK 1 - END= CYLINDER      63 TRACK 14
BG 0000 L327I RESTORE ALL SUBLIBRARIES
BG 0000 L322A ENTER YES IF THE SPECIFICATION IS CORRECT OR NO TO RESPECIFY
BG-0000
I 0 yes
BG 0000 L300I FORMATTING OF LIBRARY IJSYSRS IN PROGRESS
BG 0000 L306I RESTORE OF LIBRARY IJSYSRS IN PROGRESS
BG 0000 L325I RESTORE OF SUBLIBRARY IJSYSRS.SYSLIB IN PROGRESS
BG 0000 L326I RESTORE COMPLETE FOR LIBRARY IJSYSRS
BG 0000 SA09I SELECT ONE OF THE FOLLOWING PROGRAMS, OR TYPE END
BG 0000 SA10D FASTCOPY, RESTORE, ICKDSF, DITTO, REIPL
BG-0000
I 0 end
BG 0000 SA17W ***** END OF STAND ALONE PROCESSING *****

```

Overview of Device Support Facilities (Stand-Alone Environment)

When you select ICKDSF as the utility which you want to use, ICKDSF prompts directly for the ICKDSF control statement (according to the function you want to execute).

When you execute ICKDSF in the stand-alone environment, a message soliciting an input command is sent to the console. The input command must provide the *cuu* (UNIT parameter) of the device address instead of the programmer logical unit (SYSnnn). The input command can be only of 70 characters per line. If the command goes over 70 characters, then the 70th character (at the latest) has to be a continuation character, showing that the command continues.

A message requiring confirmation will be sent to the console before function execution, when applicable. You need to reply for the 'U' or 'T' prompt. The command output will be sent to the console. When you enter 'END' or 'EOF' ICKDSF processing is terminated.

ICKDSF in the VSE/ESA Stand-Alone Environment

The stand-alone (SA) version of ICKDSF is no longer available in the VSE/ESA stand-alone environment. The (VSE) online version of ICKDSF is loaded instead, when you select ICKDSF as the utility you want to use. For documentation on the list of supported commands and parameters refer to the **VSE version** of ICKDSF in the IBM manual *Device Support Facilities User's Guide and Reference*.

The following example shows how to use ICKDSF from the VSE/ESA stand-alone environment.

Example of an ICKDSF Run

After having invoked ICKDSF, it prompts you for the ICKDSF control statement. You have to enter the control statement according to the function you want to perform. The following example assumes a device with cuu 241.

The lines preceded by an 'I' show your input; all others are system output.

```
BG 0000 SA09I SELECT ONE OF THE FOLLOWING PROGRAMS, OR TYPE END
BG 0000 SA10D FASTCOPY, RESTORE, ICKDSF, DITTO, REIPL
BG 0000
I 0 ickdsf
BG 0000 ICKDSF - VSE          DEVICE SUPPORT FACILITIES 16.0   TIME
BG 0000 : zz:zz:zz          mm/dd/yyyy   PAGE   1
BG 0000
BG 0000
BG 0000 ENTER INPUT COMMAND:
BG-0000
I 0 init unit(241) nvfy nomap purge dosvtoc(end) volid(dosres)
BG 0000 INIT UNIT(241) NVFY NOMAP PURGE DOSVTOC(END) VOLID(DOSRES)
BG 0000
BG 0000 ICK00700I DEVICE INFORMATION FOR 0241 IS CURRENTLY AS FOLLOWS:
BG 0000
BG 0000          PHYSICAL DEVICE = 3380
BG 0000
BG 0000          STORAGE CONTROLLER = 3880
BG 0000
BG 0000          STORAGE CONTROL DESCRIPTOR = 03
BG 0000
BG 0000          DEVICE DESCRIPTOR = 06
BG 0000
BG 0000 ICK00703I DEVICE IS OPERATED AS A MINIDISK
BG 0000
BG 0000 ICK003D REPLY U TO ALTER VOLUME 0241 CONTENTS, ELSE T
BG-0000
I 0 u

BG 0000 ICK01313I VOLUME CONTAINS 0      ALTERNATE TRACKS -- 0      AVAILABLE.
BG 0000
BG 0000 ICK01314I VTOC IS LOCATED AT CCHH=X'0031 0000' AND IS      15 TRACKS.
BG 0000
BG 0000 ICK00001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
BG 0000
BG 0000          zz:zz:zz      mm/dd/yyyy
BG 0000
BG 0000
BG 0000
BG 0000 ENTER INPUT COMMAND:
BG-0000
I 0 end
BG 0000 ICK00002I ICKDSF PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
:
:
```

Overview of DITTO/ESA for VSE (Stand-Alone Environment)

DITTO/ESA for VSE belongs to the VSE/ESA base programs. It provides many functions for working with tape devices, disk devices, Volume Table of Contents (VTOCs) and catalogs, VSE/VSAM data, VSE library members, and card images. For more information on DITTO/ESA, refer to the IBM manual *DITTO/ESA User's Guide and Reference*.

Example of a DITTO/ESA Run

After having invoked DITTO/ESA, you are prompted for control statements. According to the function you want to perform, you enter the required statement. For all DITTO/ESA functions supported in the stand-alone environment, see Table 6.

The lines preceded by an 'I' show your input; all others are system output.

```
      BG 0000 SA09I SELECT ONE OF THE FOLLOWING PROGRAMS, OR TYPE END
      BG 0000 SA10D FASTCOPY, RESTORE, ICKDSF, DITTO, REIPL
      BG 0000
I 0 ditto
      BG 0000 *** DITTO/ESA active ***
      BG 0000 When prompted, enter GO to skip optional prompts, or U to exit from function
      BG 0000 DITTO function ?
      BG-0000
I 0 xxx
      BG 0000 Printing...
      BG 0000 XXX completed
      BG 0000 DITTO function ?
      BG-0000
I 0 did
      BG 0000 Input disk ? (Device number or Vxxxxxx VOLID)
      BG-0000
I 0 300
      BG 0000 Existing VOLSER is DOSRES on device 0300
      BG 0000 Enter new volume serial xxxxxx
      BG-0000
I 0
      BG 0000 DID completed
      BG 0000 DITTO function ?
      BG-0000
I 0 end
```

Table 6. Functions of DITTO/ESA in a Stand-Alone Environment

TAPE CONTROL	
BSF	Backward Space File
BSR	Backward Space Record
ERT	Erase Tape
FSF	Forward Space File
FSR	Forward Space Record
INT	Initialize Tape
REW	Rewind Tape
RUN	Rewind Unload
SET	Set Processing
VER	Version Command
WTM	Write Tape Mark
XXX	Command Summary Listing (stand alone update)

Table 6. Functions of DITTO/ESA in a Stand-Alone Environment (continued)

TAPE	
TCN	Tape to Console
TLB	Tape to Label Display
TLT	Tape to Labeled Tape
TMP	Tape Map
TP	Tape Print
TRL	Tape Record Load
TRS	Tape Record Scan
TT	Tape to Tape Copy
TTC	Tape to Tape Compare
TTR	Tape to Tape Reblocked
VTOC	
DSE	Data Set Extents
DVT	Disk VTOC display
PVT	Process VTOC
DISK	
DCN	Disk to Console
DID	Alter Disk Volume Identifier
DSE	Data Set Extents
DP	Disk Print
DRL	Disk Record Load
DRS	Disk Record Scan
EOF	End of File Record

Customizing Stand-Alone Utilities Using the Customization Table

You can customize the stand-alone utilities to reduce operator intervention when restoring a system from a backup tape to disk.

How to proceed to customize stand-alone utilities:

- In the VSE/ESA online environment, you generate a customization table by using macro SASTCK. The resulting phase will be part of the stand-alone utilities on the IPL tape.
- To put the customization table on the IPL tape, you have to specify the Librarian BACKUP command with the CUSTTABLE operand to support the customization table and the changing of message texts. The following is an example:

```
BACKUP TAPE=480 RESTORE=STANDALONE CUSTTABLE=IJSYSRS.SYSLIB.FCOREST.PHASE
```

Refer to the IBM manual *VSE/ESA System Control Statements* for a detailed description of the BACKUP command.

The customization table logically consists of two parts:

1. Answer table

The answer table contains all console input requested during a stand-alone run.

2. Message table

The message table defines modified message texts and message answers.

The default name of the table is IJWCUST.PHASE. It contains no answers and no modified messages. A customization table containing only the characters /* is considered as empty. Message prompting is not suppressed, and the operator must answer the messages as usual. It is recommended that you keep the (empty) system-provided customization table (IJWCUST.PHASE) and use your own name when creating a table.

For creating the table, the system provides:

Macro SASTCK in PRD1.MACLIB.

Sample jobs for VSE/Fast Copy stand-alone in Z-book SACUSTFC in system library IJSYSRS.SYSLIB.

Sample jobs for Librarian RESTORE in Z-book SACUSTRS in system library IJSYSRS.SYSLIB.

Customized Answer Table

This table contains answers to console-requested input issued during a stand-alone Librarian RESTORE or VSE/Fast Copy stand-alone in order of their occurrence.

To create an answer text, you use the macro SASTCK to build the corresponding table entry. The sequence of defined answers will be used as console input during stand-alone processing. In one customized answer table you can invoke the stand-alone utilities sequentially several times. If a defined answer does not match the expected console input, processing checks for the next stacked answer. If no further stacked messages and replies are available, the system switches to prompting mode; that is, console prompts are issued, and you have to answer them manually. The maximum answer length is 80 characters (for ICKDSF only 70 characters are used as valid input).

If modified messages or messages that are forced with the '%DSPLY' parameter require an answer, the answer must be entered from the console.

Console messages are suppressed as long as no abend or cancel condition occurs and the message text is not modified or forced with the '%DSPLY' parameter. (See "Macro SASTCK" on page 43.) Supervisor emergency and action messages are not suppressed and must be answered from the system console.

Customized Message Table

To modify a message text, use the SASTCK macro. It defines the number of the message to be changed, the modified message text, the modified reply to the original message and the original reply. The maximum length is 80 characters (for ICKDSF only 70 characters are used as valid input).

Page 46 shows an example for a customized message table.

What You Should Consider when Changing Messages or Replies

- You cannot stack Attention Routine (AR) commands. AR commands are not passed through customization tables to the system.
- You should not enter AR commands on the system console during a customized stand-alone run, because the AR command output would go to the printer (if specified) and not to the system console.
- You must specify two single quotes, if you want to stack an answer that has to be entered with a single quote itself.

- If a changed message requires an answer, and if it is forced to the system console, this message must be answered from the system console. You cannot stack the answer to such a message into your customization table.
- If you have customized the stand-alone utilities, all decision messages (D messages) are suppressed on the system console. If you specify a printer in your customization table and have changed messages and replies, the printer will nevertheless show IBM-shipped messages and replies. You will see **all** messages or replies issued, although you may have suppressed them on the system console. This helps solve problems in case of an error situation.

Use of Macro SASTCK

Macro SASTCK

Macro SASTCK is used to provide answer text or to modify messages and replies.

Control Statement Format:

```

>>—SASTCK 'answertext' [ ,PNAME=IJWCUST ] [ ,EOD=NO ]
                        [ ,PNAME=phasename ] [ ,EOD=YES ]
>>—SASTCK 'msg_text' ,MNO=msg_no
> [ ,MRPL='modified_reply',ORPL='original_reply' ] [ ,PNAME=IJWCUST ]
                                     [ ,PNAME=phasename ]
> [ ,EOD=NO ]
  [ ,EOD=YES ]
>>—SASTCK '%DSPLY' ,MNO=msg_no
> [ ,MRPL='modified_reply',ORPL='original_reply' ]
>>—SASTCK PNAME=phasename
>>—SASTCK EOD=YES

```

Parameter description:

'answertext'	Text of stacked message text. The maximum reply length is 8 characters.
'msg text'	Text of modified message text.

MNO	Message number to be modified. MNO can consist of 5 positions only.
MRPL	Modified reply text. You can only specify MRPL for messages requiring a reply (generally A and D messages).
ORPL	Original reply text. You can only specify ORPL for messages requiring a reply (generally A and D messages).
PNAME	Name of phase to be used in LIBRARIAN backup. Default is IJWCUST. This parameter is only evaluated in the first macro call.
'%DSPLY'	With this parameter you preserve original message text which may include variables, for example, an error message with specific error code, or a device address. The keyword '%DSPLY' in SASTCK macro stands for the original message. If you specify '%DSPLY' you must also specify MNO.
EOD	This parameter is for the last macro call. Default is NO. If you specify EOD=YES, subsequent macro calls will not generate any code.

Example 1 – Using Macro SASTCK to Modify Message Text and Reply

In the following example, the message text:

```
4140A NO ALTERNATE DRIVE ASSIGNED UIN
```

is requested to be changed to:

```
INSERT NEXT TAPE IN DRIVE AND ENTER GO
```

The answer to this message is GO and results in the following macro call:

```
SASTCK 'INSERT NEXT TAPE IN DRIVE AND ENTER GO',      X
      MNO=4140A,MRPL='GO',ORPL='NEWTAP'
```

Example 2 – Building a Stand-Alone Librarian Restore Customization Table

This example shows how to create an answer table. Note that the console prompts are included as comments.

Step 1 – Creating a Customization Table for Automatic Stand-Alone Restore (Phase SAREST):

Note: Code responses for RESTORE statements and messages are only issued if they are actually needed during the RESTORE process.

```

// JOB SACUSTRS BUILD SA RESTORE CUSTOMIZING PHASE
// OPTION CATAL
// LIBDEF PHASE,CATALOG=IJSYSRS.SYSLIB
// EXEC ASMA90
  SASTCK PNAME=SAREST
* SPECIFY PRINTER (MSG SA02D) - ENTER MEANS NO PRINTER DESIRED
  SASTCK '
* SPECIFY DATE (MSG SA08D) - ACCEPT SYSTEM DATE AS JOB DATE
  SASTCK '
* SELECT ONE OF THE FOLLOWING PROGRAMS OR TYPE END (MSG SA09I AND SA10D)
  SASTCK 'RESTORE'
* SPECIFY ADDRESS OF INPUT CUU (MSG SA11D)
  SASTCK '480'
* SPECIFY TYPE OF INPUT CUU (MSG SA03D) - ACCEPT SENSED DEVICE TYPE
  SASTCK '
*
* NOTE 1: YOU HAVE TO SPECIFY THE DEVICE TYPE IF THE DEVICE TYPE
*         CAN NOT BE SENSED AT IPL-TIME.
*
* TAPE LABEL CHECKING REQUIRED? (MSG SA15D) - ENTER MEANS NO TLBL
  SASTCK '
* SPECIFY ADDRESS OF SYSRES DISK (MSG SA16D)
  SASTCK '300'
* SPECIFY TYPE OF SYSRES DISK (MSG SA03D) - ACCEPT SENSED DEVICE TYPE
  SASTCK '
* ANSWER MESSAGE L320A
* L320A ENTER YES TO RESTORE SYSRES FILE IJSYSR1 OR NO TO SKIP
*   TO NEXT SYSRES
  SASTCK 'YES'
* ANSWER MESSAGE L316A
* L316A ENTER YES TO KEEP OR NO TO RESPECIFY THE SYSRES FILE ID
  SASTCK 'YES'
* ANSWER MESSAGE L310A
* L310A ENTER YES TO KEEP OR NO TO RESPECIFY THE ALLOCATION
  SASTCK 'NO'
* ANSWER MESSAGE L313A
*
* NOTE 2: TYPE IN THE ALLOCATION PERTAINING TO YOUR DASD DEVICE.
*         FOR DETAILS PLEASE REFER TO VSE/ESA INSTALLATION MANUAL.
*
  SASTCK '959'
* ANSWER MESSAGE L329A
* L329A ENTER YES TO RESTORE ALL SUBLIBRARIES OR NO FOR SELECTIVE
*   RESTORE
  SASTCK 'YES'
* ANSWER MESSAGE L322A
* L322A ENTER YES IF THE SPECIFICATION IS CORRECT OR NO TO RESPECIFY
  SASTCK 'YES'
  SASTCK 'RESTORE COMPLETE FOR OPERATING SYSTEM. IPL FROM DISK 300.', *
        MNO=L326I
* SELECT ONE OF THE FOLLOWING PROGRAMS OR TYPE END (MSG SA09I AND SA10D)
  SASTCK 'END'
  SASTCK EOD=YES
  END
/*
// EXEC LNKEDT,PARM='MSHP'
/*
/&

```

Step 2 – Creating Stand-Alone Tape Including the Customization Table (Phase SAREST):

```
// JOB SATAPE BUILD STAND ALONE TAPE
// MTC REW,480
// EXEC LIBR
BACKUP LIBRARY=IJSYSRS TAPE=480 RESTORE=STANDALONE -
CUSTTABLE=IJSYSRS.SYSLIB.SAREST.PHASE
/*
// MTC REW,480
/ &
```

Example 3 – Building a Stand-Alone VSE/Fast Copy Customization Table

The following sample job, SACUSTFC, shows how to build the customized answer table for use with customized stand-alone utilities and a changed message (8F02I).

Step 1 – Creating the Customization Table (Phase FFOREST):

```
..// JOB SACUSTFC BUILD SA FASTCOPY CUSTOMIZING PHASE
// OPTION CATAL
// LIBDEF PHASE,CATALOG=IJSYSRS.SYSLIB
// EXEC ASMA90
  SASTCK PNAME=FFOREST
* SPECIFY PRINTER (MSG SA02D) - ENTER MEANS NO PRINTER DESIRED
  SASTCK '
* SPECIFY DATE (MSG SA08D) - ENTER MEANS ACCEPT SYSTEM DATE AS JOB DATE
  SASTCK '
* SELECT ONE OF THE FOLLOWING PROGRAMS OR TYPE END (MSG SA09I AND SA10D)
  SASTCK 'FASTCOPY'
* SPECIFY ADDRESS OF INPUT DEVICE (MSG SA11D)
  SASTCK '480'
* SPECIFY TYPE OF INPUT DEVICE - SENSED DEVICE TYPE ACCEPTED (MSG SA03D)
  SASTCK '
* SPECIFY ADDRESS OF ALTERNATE TAPE (MSG SA14D)
  SASTCK '
* SPECIFY ADDRESS OF OUTPUT DEVICE (MSG SA12D)
  SASTCK '300'
* SPECIFY TYPE OF OUTPUT DEVICE - SENSED DEVICE TYPE ACCEPTED (MSG SA03D)
  SASTCK '
* TAPE LABEL CHECKING REQUIRED? (MSG SA15D)
  SASTCK '
* 8F30D ENTER FASTCOPY CONTROL STATEMENT OR END
  SASTCK 'RESTORE VOLUME NOVERIFY NOREWIND'
* 8F34D DUMP DATA ON IPL TAPE
  SASTCK 'YES'
* 8X63D UNIT=CUU VOLID=VOLID OWNED BY VSAM
  SASTCK 'PROCEED'
*
* NOTE 1: THIS REPLY IS ONLY NEEDED WHEN RESTORING TO A DISK THAT IS
* ALREADY A VSE FORMATTED VOLUME WITH DATA. DO NOT CODE IF DISK IS
* NEWLY FORMATTED AND EMPTY.
*
* 8X65D UNIT=CUU VOLID=VOLID REPLY CANCEL SHOW OR PROCEED
  SASTCK 'PROCEED'
* 4140A NO ALTERNATE DRIVE ASSIGNED UOUT
  SASTCK 'NEWTAP'
```

```

* NOTE 2: CODE THIS REPLY ONLY IF A SECOND TAPE IS ACTUALLY NEEDED AND
* YOU DID NOT SPECIFY AN ALTERNATE TAPE. THE MESSAGE S120A,
* 'INTERVENTION REQUIRED DEVICE=XXX' MUST BE ANSWERED MANUALLY AT THE
* CONSOLE
  SASTCK 'RESTORE DONE. NOW IPL FROM 300',MNO=8F02I
* SELECT ONE OF THE FOLLOWING PROGRAMS OR TYPE END (MSG SA09I AND SA10D)
  SASTCK 'END'
  SASTCK EOD=YES
  END
/*
// EXEC LNKEDT,PARM='MSHP'
/*
/&

```

Code responses for FASTCOPY statements and messages only if they will actually be needed during the restore process. Coding a response to a message that will not occur, or is in error, may result in message:

```
8X01D  INCORRECT REPLY,RETRY
```

Step 2 – Creating a Stand-Alone Tape Including Customization Table (Phase FCoreST): Create an IPL tape using the Librarian BACKUP command and specifying the operand CUSTTABLE with the following job, where FCoreST is the phase name for the CUSTTABLE built with job SACUSTFC.

```

// JOB SATAPE    BUILD STAND ALONE TAPE
// MTC REW,480
// EXEC LIBR
  BACKUP TAPE=480 RESTORE=STANDALONE -
  CUSTTABLE=IJSYSRS.SYSLIB.FCOREST.PHASE
/*
/&

```

Step 3 – Performing a Backup of DOSRES on the IPL Tape: To ensure consistency of system files, shut down CICS, VTAM, and VSE/POWER before you use FASTCOPY.

When shutdown is complete, the following message appears:

```
F1-0001 1C10D  PLEASE ASSIGN SYSRDR
```

You have to enter the following job in partition BG, preceded by a 0:

```

// ASSIGN SYSLST,IGN
// JOB FCOTAPE    BUILD STAND ALONE FASTCOPY TAPE
// ASSIGN SYS004,DISK,VOL=DOSRES,SHR
// ASSIGN SYS005,480
// EXEC FCOPY
  DUMP VOLUME NOREWIND LIST NOLABEL
/*
// MTC REW,480
/&

```

Part 2. Migrating to a New Release

Chapter 5. VSE/ESA Migration Utility IESBLDUP

This chapter describes the VSE/ESA migration utility IESBLDUP. IESBLDUP is called during **initial installation** when you request migration. However, you can use IESBLDUP also at a later time, for example if you have installed a test system and want to migrate your user profiles and other control data after you have tested your new system.

Introducing the IESBLDUP Utility

IESBLDUP migrates DOS/VSE, VSE/SP, or previous VSE/ESA **user profiles** as well as user-tailored **selection panels** and **application profiles** to VSE/ESA 2.4. IESBLDUP updates and adds user profile or other data to the **current** VSE Control File (IESCNTL) and the **current** VSE/ICCF DTSFILE.

During migration, the sources of input are processed in the following sequence:

1. Current VSE/ICCF DTSFILE.
2. Copy of old VSE Control File and/or old VSE/ICCF DTSFILE.
3. CICS/VSE sign-on table DFHSNT (from CICS/VSE 2.3 or a previous CICS system).

If duplicate user IDs are found, only the first occurrence of an user ID will be added. The user IDs defined for VSE/ICCF and CICS/VSE provide, together with a model VSE/ESA user profile, the data to define new users to VSE/ESA.

You can also use IESBLDUP to create a **status report** of your system's users. A status report lists all user IDs and related data defined in the VSE Control File, the VSE/ICCF DTSFILE, and the CICS sign-on table (DFHSNT). Such a status report provides valuable information for maintaining user profiles and data.

Note on DFHSNT

CICS/VSE 2.3 was the last CICS system that had a DFHSNT (Sign-On Table). Its successor system, the CICS Transaction Server for VSE/ESA, does not include a DFHSNT.

If you want to migrate a DFHSNT, the easiest way is to migrate it first to your old VSE Control File. Run IESBLDUP to achieve this, and then create a backup copy of your old VSE Control File for the actual migration run. You can apply this method for CICS/DOS/VS 1.7 up to CICS/VSE 2.3. In case you want to migrate an even older CICS version, you must create a CICS coexistence environment under VSE/ESA 2.4 by installing CICS/VSE 2.3 in addition to the CICS Transaction Server. You must then recompile your old DFHSNT under CICS/VSE 2.3.

The IESBLDUP Control Statement

You control the processing of IESBLDUP through options defined in the IESBLDUP control statement. If no control statement is used, the default options are in effect. The statement and its options are described under "The Control Statement" on page 61.

Planning and Concept Information

Migrating VSE/ICCF DTSFILE and VSE Control File to VSE/ESA 2.4

If you migrate from a **VSE/SP** or previous **VSE/ESA** system to VSE/ESA 2.4, the old VSE Control File and old DTSFILE are processed together when **CF=YES** is specified in the control statement.

The user profiles are copied from the old files. The records from the old VSE Control File are reformatted and merged into the current VSE Control File. No model profile and DFHSNT values are used as input. The old VSE Control File is also scanned for user selection panels which reference VSE/ESA reserved names (beginning with IES). If your panels contain names of VSE/ESA panels or applications which have been deleted, the names are changed to the corresponding VSE/ESA 2.4 functions and listed in the status report.

Copying DOS/VSE User Definitions of VSE/ICCF DTSFILE Only

The old DTSFILE is read from SYS004. SYS005 must be assigned with IGN (//ASSGN SYS005,IGN) to ensure that the VSE Control File is not read even if CF=YES is specified. The user profiles from the old DTSFILE are copied directly into the current DTSFILE without change. You can add the VSE/ESA user profiles directly from the DTSFILE to the current VSE Control File (DTSFILE=YES). *This statement may be used to copy user definitions of the old DTSFILE to the current DTSFILE without using model profiles.*

Migrating VSE/ICCF DTSFILE Only (DOS/VSE)

If migrating from **DOS/VSE**, the only input is a copy of your old DTSFILE. If a user ID is found that does not exist on the current VSE Control File, several actions take place:

- The user ID is added to the current VSE Control File.
- A user profile is created and added to the current VSE Control File.

The user's ID, password, and VSE/ICCF library number are taken from the old DTSFILE records. The VSE/ESA options and VSE/ICCF specifications are taken from the model user profile specified by the ADMN or PROG keyword in the control statement.
- A DTSUTIL ADD statement is created to add the user to the DTSFILE.
- If you do not want to use the new model profiles (SYSA and PROG, for example) you must copy the users of the old DTSFILE, as described in "Copying DOS/VSE User Definitions of VSE/ICCF DTSFILE Only".

VSE/ICCF DTSFILE

If **DTSFILE=YES** is specified in the control statement, users are added from the current DTSFILE. The user records (B\$ and E\$ records) are read from the current DTSFILE. For each unique ID found, a user profile is created and added to the current VSE Control File. The user's ID, password, VSE/ICCF library number, and VSE/ICCF specifications are taken from the current DTSFILE records. VSE/ESA options are taken from the model user profile.

VSE/ESA Required VSE/ICCF Options

The VSE/ESA Interactive Interface requires user profiles to be defined with certain VSE/ICCF options in order to operate correctly. For example, if a type 1 user profile was specified as a model for a type 2 user, the VSE/ICCF option byte settings

would be incorrect. If a new user profile is added to the current VSE Control File, incompatible VSE/ICCF options are listed in the status report.

If **ALT=YES** is specified in the control statement and a new user is found to be ADDED, it will be added with the VSE/ESA default values. The values for an existing user in the current DTSFILE will be ALTERed to the default values.

The VSE/ESA default values are as follows:

- OPTA, OPTB, and OPTC bytes.

User type 1 (Administrator):

```
OPTA - 011*00*1
OPTB - **111010
OPTC - **000*0*
```

User type 2 (Programmer):

```
OPTA - 000*01*0
OPTB - **000000
OPTC - **000*0*
```

* bits not checked or changed

```
TIMEOUT - 600
LOGONRTN - @L$LOGON
DEL, TAB, BS, ESC, END, and HEX - Set to OFF
```

For more information on the VSE/ICCF options refer to “Maintaining User Profiles” in the IBM manual *VSE/ESA Administration*.

Reserved VSE/ICCF Libraries

When adding VSE/ICCF users, the user’s library number is checked to determine if one of the VSE/ESA reserved libraries (50 - 68) has been specified. If so, the library is listed on the status report as being incompatible.

If **ALT=YES** has been specified in the control statement, the library number is altered to 10 for administrators and to 9 for programmers. If an alternate library has been specified as one of the VSE/ESA reserved libraries, it is not changed but indicated in the status report.

CICS Sign-On Table (DFHSNT)

If **SNT=YES** is specified in the control statement, DFHSNT is scanned for entries. If there is no matching user ID defined in the current VSE Control File, the DFHSNT user ID is added as a type 3 user (non-VSE/ICCF user).

Refer also to “Note on DFHSNT” under “Introducing the IESBLDUP Utility” on page 51.

Using the IESBLDUP Utility

IESBLDUP performs the following steps:

- It updates the VSE Control File with user profile and control data information.
- It creates job DTRMIGR and places it into the VSE/POWER reader queue.

DTRMIGR invokes the VSE/ICCF utility DTSUTIL to perform the following functions:

- Add new user profiles to the current VSE/ICCF DTSFILE.
- Alter VSE/ICCF DTSFILE options which are incompatible with VSE/ESA 2.4.
- Catalog a dummy table ADM\$USRT into VSE/ICCF library 50 for performance reasons. Table ADM\$USRT will be deleted after the *Maintain User Profiles* dialog is used for the first time.

To ensure that DTRMIGR is processed immediately after IESBLDUP has finished, run IESBLDUP as a CLASS=A job (DTRMIGR is created as CLASS=A, DISP=K).

Assignments

For the IESBLDUP job stream, the following assignments are used:

- SYS004

Defines the tape drive where the copy of the old VSE/ICCF DTSFILE (DTSRSTR) is mounted.

- SYS005

Defines the tape drive where the copy of the old VSE Control File is mounted. If only one physical tape drive is available and CF=YES is specified in the control statement, SYS005 must be unassigned (// ASSGN SYS005,UA). The operator will then be instructed to mount the old VSE Control File copy on SYS004 after processing of the DTSFILE copy has been completed.

In case of DOS/VSE, which uses as input the DTSFILE only, SYS005 must be defined with // ASSGN SYS005,IGN to avoid reading of the old VSE Control File (even if CF=YES).

- SYS010

Defines the disk device on which the current VSE/ICCF DTSFILE resides. As shipped, the DTSFILE resides on the SYSWK1 system disk. The assignment is done by procedure DTRICCF.

Running IESBLDUP

When running IESBLDUP, you must observe the following:

- IESBLDUP must run in a batch partition controlled by VSE/POWER. VSE/POWER must be generated with the SPOOL=YES option, as delivered with VSE/ESA 2.4.
- If UPDATE=YES is specified and the Interactive Interface is active, enter the following CEMT command to close the VSE Control File (IESCNTL).:
CEMT SET FILE(IESCNTL) CLOSE

As long as IESCNTL is closed, you cannot sign on to the Interactive Interface.

You also are requested to disconnect the DTSFILE before DTRMIGR runs. Enter the following command:

/DISC DTSFILE

When DTRMIGR has finished, you should reconnect the DTSFILE using the command:

/CON DTSFILE

- If UPDATE=NO is specified in the control statement, both the current VSE Control File and DTSFILE are used in read only mode and need not be closed or disconnected.
- The VSE/ICCF modules DTSFILRT and DTSUTIL must be in a VSE/ESA library accessible to IESBLDUP. The same is true for the CICS sign-on table (DFHSNT), if you have set SNT=YES (see Figure 7 on page 58).

The Status Report

A status report, which shows the results of the migration process, is printed on SYSLST. The following items are listed:

- The existing user IDs defined in the VSE Control File, VSE/ICCF DTSFILE, and the CICS sign-on table DFHSNT (if DFHSNT exists).
- Each user ID processed together with the VSE/ICCF library assigned, the model profile used, and an indication whether a matching DFHSNT entry was found in case SNT was specified.

If the VSE/ICCF options found are incompatible with VSE/ESA 2.4, the old values are shown on the status report.

Model User Profiles

For each user profile added to the current VSE Control File (except for users from the old VSE Control File), an existing VSE/ESA user profile is used as a model for the new user. The model profile provides default values when creating a new user profile.

VSE/ESA provided user profiles (for example SYSA, PROG, and OPER) may be used as model profiles. You may also define your own model profile with the *Maintain User Profiles* dialog.

For users added to the current VSE Control File from the current or old DTSFILE, the model selected is based on the user class defined in the VSE/ICCF DTSFILE user record. The profile specified in the control statement for ADMN will be used as the model for the administrator (bit 2 of OPTB=1), and the name specified for PROG will be used for all other users. The defaults, which may be changed, are ADMN=SYSA and PROG=PROG. If you use different model users, the related users must be defined in the **new** (VSE/ESA 2.4) VSE Control File.

If users are to be added from the CICS sign-on table (DFHSNT) as type 3 users, a profile should be defined with the GENL parameter in the control statement. If no name is specified, the default is GENL=PROG with the user type changed from type 2 to type 3. When PROG is used as a default profile, the initial selection panel will offer selections that require access to VSE/ICCF. Type 3 users, however, do not have access to VSE/ICCF functions. It is therefore recommended that you create your own default profile for type 3 users.

Adding New Users

If you want to add new user IDs to VSE/ESA, you can also use a batch program described in “Chapter 6. Batch Program IESUPDCF” on page 69. It is recommended to add new users **after** you have migrated your system.

Migrating to VSE/ESA 2.4

Backups Required from Your Old System

To perform migration you need a copy on tape of your old DTSFILE and your old VSE Control File.

If you want to migrate users defined in DFHSNT (the latest is the CICS/VSE 2.3 DFHSNT), you must ensure that the DFHSNT is in a library accessible to IESBLDUP.

1. If migrating from **VSE/SP** or an earlier version of **VSE/ESA**, create a copy of the old DTSFILE by using the *Backup/Restore Library Objects* dialog.

If you are a **DOS/VSE** user, use the VSE/ICCF utility DTSUTIL to create a copy of the DTSFILE. Note that step 2 does **not** apply to **DOS/VSE**, since in this case the VSE Control File is not used as input.

2. If migrating from **VSE/SP** or an earlier version of **VSE/ESA**, create a copy of the old VSE Control File (IESCNTL).

You can create a backup copy either by:

- a. **Using the Interactive Interface**

Select the *Resource Definition* dialog from the *VSE Selection Panel*. In the *File and Catalog Management* dialog select the *Display or Process a File* panel for the VSE/VSAM user catalog VSESPUC. On the next panel, select the COPY function to copy the VSE Control File (IESCNTL). The COPY function uses the VSE/VSAM REPRO command. When requested, the following should be specified:

- Volume serial number (CF0001)
- RECORD FORMAT=3 (variable-length blocked records)
- BLOCK SIZE=4000

Note: Do *not* use VSE/VSAM Backup/Restore to create the VSE Control File copy. The VSE/VSAM REPRO command must be used.

- b. **Creating a jobstream for batch processing**

```
* $$ JOB JNM=BACKCNTR,CLASS=A,DISP=D,NTFY=YES
// JOB COPY FILE
// DLBL COPYIN,'VSE.CONTROL.FILE',,VSAM,          X
           CAT=VSESPUC
// TLBL COPYOUT,'CONTROL.FILE',,CF0001
// ASSGN SYS005,181
// EXEC IDCAMS,SIZE=AUTO
      REPRO INFILE (COPYIN) -
           OUTFILE (COPYOUT -
           ENVIRONMENT (BLOCKSIZE (4000 ) -
           RECORDFORMAT (VARBLK) -
           STDLABEL -
           PRIMEDATADEVICE (2400) REW)) -
           NOREUSE
/*
/&
* $$ E0J
```

Figure 5. Example Job for Backing Up the VSE Control File

Precautionary Measures

The following measures help you recover in case migration failed during initial installation or later. Even if migration failed during initial installation, you can take a backup copy of the VSE Control File which is not damaged since no update takes place in case of a failure. The best point for creating a backup copy is after initial installation post-base processing.

1. A **backup** the DTSFILE of your new VSE/ESA 2.4 system using the *Backup/Restore Library Objects* dialog. Perform this step to ensure that you still have a correct version of the DTSFILE available in case of migration errors. If you encounter migration errors, see page 64. However, before backing up the DTSFILE a reformat run will be necessary if you want to migrate more libraries or users than the current DTSFILE allocation allows. The VSE/ESA-supplied DTSFILE contains an allocation for 199 libraries and 199 users.
2. A **backup** of the VSE Control File (IESCNTL) of your new VSE/ESA 2.4 system. For details on how to do this, see step 2 on page 56 under “Backups Required from Your Old System” on page 56.

Refer also to “Recovering from Migration Errors” on page 64.

Migration Steps

The following summary of migration steps applies if you did **not** migrate during initial installation.

1. Create backup tapes of the old DTSFILE and the old VSE Control File.
If you migrate a DFHSNT, ensure that it is in a library accessible to IESBLDUP.
2. Prepare the job stream to run IESBLDUP.
The control statement must be placed immediately after the // EXEC statement. Refer to “The Control Statement” on page 61 for the default values active.
3. Run IESBLDUP first with UPDATE=NO in the control statement to get a status report.
4. Review the status report and correct any errors.
5. Change the control statement to UPDATE=YES and resubmit the job. If UPDATE=YES, close the VSE Control File if the corresponding message is issued.
6. Job DTRMIGR will be submitted as soon as IESBLDUP has finished processing. Disconnect the DTSFILE when instructed to do so.
7. Review the status report.

Example 1: Migrating to VSE/ESA 2.4 from a VSE/SP or Previous VSE/ESA System

The following example shows migration to VSE/ESA 2.4 from a VSE/SP or VSE/ESA system. The user profiles SYSA and PROG are implicitly used as user models for administrators and programmers.

This sample job stream is provided as skeleton IESBLDUP in VSE/ICCF library 59.

Migration Utility IESBLDUP

```
* $$ JOB JNM=IESBLDUP,CLASS=A,DISP=D
// JOB IESBLDUP  MIGRATION FROM VSE/SP or VSE/ESA
// EXEC PROC=DTRICCF
// TLBL DTSRSTR,,ICCF01      **DTSFILE BACKUP**
// ASSGN SYS004,300
// TLBL CFCOPY,,CF0001      **CONTROL FILE COPY**
// ASSGN SYS005,301
// PAUSE  IESCNTRL MUST BE CLOSED IF UPDATE=YES IS SPECIFIED
// EXEC IESBLDUP,SIZE=64K
* CONTROL STATEMENT FOR MIGRATION FROM A VSE/SP OR VSE/ESA SYSTEM
CF=YES,DTSRSTR=YES,UPDATE=YES
/*
/&
* $$ E0J
```

Figure 6. Job Stream Example for Migrating from VSE/SP or Previous VSE/ESA System

Example 2: Migrating to VSE/ESA 2.4 from a DOS/VSE System

The following example shows migration to VSE/ESA 2.4 from a DOS/VSE system. User profiles are created from the IDs defined in the old DTSFILE and the CICS sign-on table DFHSNT (if a DFHSNT exists). USRA, USRB, and USRC are used as model user profiles. You must define them in the current VSE Control File before submitting the job stream.

Note that the DFHSNT must be CICS/DOS/VS version 1.7 or later. If you migrate from an earlier version, you must first create a job stream to process your old DFHSNT with macro DFHSNT of CICS/VSE 2.3 in the VSE/ESA 2.4 coexistence environment. Refer also to "Note on DFHSNT" under "Introducing the IESBLDUP Utility" on page 51.

```
* $$ JOB JNM=IESBLDUP,CLASS=A,DISP=D
// JOB IESBLDUP  MIGRATION FROM DOS/VSE
// EXEC PROC=DTRICCF
// TLBL DTSRSTR,,ICCF01      **DTSFILE BACKUP**
// ASSGN SYS004,300
// PAUSE  UPDATE=YES IS SPECIFIED, CLOSE IESCNTRL
        BEFORE PROCEEDING
// EXEC IESBLDUP,SIZE=64K
CF=NO,DTSRSTR=YES,SNT=YES,ALT=YES,UPDATE=YES
ADMN=USRA,PROG=USRB,GENL=USRC
/*
/&
* $$ E0J
```

Figure 7. Job Stream Example for Migrating from DOS/VSE

Example 3: Creating a Status Report of Existing User IDs

The following example produces a status report which lists the user IDs defined in the VSE Control File, VSE/ICCF DTSFILE, and the CICS sign-on table DFHSNT (if a DFHSNT exists).

```
* $$ JOB JNM=IESBLDUP,CLASS=A,DISP=D
// JOB IESBLDUP STATUS REPORT OF USER IDS
// EXEC PROC=DTRICCF
// EXEC IESBLDUP,SIZE=64K
CF=NO,DTSRSTR=NO,UPDATE=NO
/*
/&
* $$ E0J
```

Figure 8. Job Stream Example for Status Report

Migration Utility IESBLDUP

Example 4: Status Report Example

Figure 9 shows an example of a status report.

```
// JOB MIGRAT
// EXEC PROC=DTRICCF
// ASSIGN SYS010,DISK,VOL=SYSWK1,SHR DTSFILE EXTENT
IT201 SYS010 HAS BEEN ASSIGNED TO X'141' (TEMP)
EOP DTRICCF
// EXEC IESBLDUP,SIZE=64K

10/13/98 VSE/ESA MIGRATION UTILITY -INITIALIZATION- PAGE 1
CF=NO,DTSRSTR=NO,UPDATE=NO,DTSFILE=YES
IESU00021 VSE CONTROL FILE 'IESCNTL' OPENED FOR INPUT
IESU00041 ICCF DTSFILE OPENED
10/13/98 VSE/ESA MIGRATION UTILITY -STATUS OF EXISTING VSE USER IDS- PAGE 2
CONTROL
USER ID DTSFILE FILE
$SRV N 2
AAAA A
AAA2 A
AAA3 A
ABCDE 1
ADAM A
ADMN A
AMAD A 1
?AMPR 2 WARNING, TYPE 1 OR 2 USER WITH NO ICCF USER PROFILE
ARMI A
ASEC A 2
AZZZ A
A102 A
A103 A
A104 A
:
:
A138 A
A139 A
A140 A
10/13/98 VSE/ESA MIGRATION UTILITY -STATUS OF EXISTING VSE USER IDS- PAGE 3
CONTROL
USER ID DTSFILE FILE
A141 A
A170 A
A171 A
A172 A
BGIG A
CALL A
CHIN 3
DAVE A
DAVI A 1
DAV8 A
DAV9 A
DISP A 1
DUVE N
D112 A
ENGL 3
?FAC1 2 WARNING, TYPE 1 OR 2 USER WITH NO ICCF USER PROFILE
?FAC2 2 WARNING, TYPE 1 OR 2 USER WITH NO ICCF USER PROFILE
?FAC3 2 WARNING, TYPE 1 OR 2 USER WITH NO ICCF USER PROFILE
?FAC4 2 WARNING, TYPE 1 OR 2 USER WITH NO ICCF USER PROFILE
FORSEC 1
GLAE A
```

Figure 9. Example for a Status Report Output (Part 1 of 2)

```
GOEB      A
HAL0      A
HANS      A
HART      A
HEIN      A
HELD      A
HSCZ      A      1
IBMS      A      1
JAPA      A      3
JIM2      N
JOER      A
JORG      A      1
JSCH      A      1
?JSC1     A      1      WARNING, TYPE 1 OR 2 USER WITH NO ICCF USER PROFILE
JUST      A
MOST      N
NETT      A
NLST      A      1
NOICCF    A      1
OPER      A      2
POST      A      1
PROG      N      2
SCHA      A
SPAN      A      3
SYSA      A      1
TES2      A
USCH      A      1
VOGL      A
10/13/98 VSE/ESA MIGRATION UTILITY          -STATUS OF EXISTING VSE USER IDS-      PAGE 4
CONTROL
USER ID   DTSFILE  FILE
VOLK      A      1
VOL0      A
VOL1      A
VOL2      A
V111      A
V222      A
?WACK     A      1      WARNING, TYPE 1 OR 2 USER WITH NO ICCF USER PROFILE
WURS      A
ZELL      A      1
ZEL1      A
1234      A
2222      A
9999      A
TOTAL     74      32
ALLOC     199
DTSFILE USER TYPE:
A = ADMINISTRATOR
N = NORMAL
CONTROL FILE USER TYPE:
1 = ADMINISTRATOR
2 = PROGRAMMER
3 = GENERAL
10/13/98 VSE/ESA MIGRATION UTILITY          -CREATE USER PROFILES-      PAGE 5
CONTROL  DEFAULT  ICCF  -----ICCF OPTIONS INCOMPATIBLE WITH VSE/ESA-----
USER ID  DTSFILE  FILE  PROFILE  LIBR  LIBR ALT  OPTA  OPTB  OPTC  LOGON  TIMEO  DEL  TAB  BS  ESC  END  HEX
USERS ADDED FROM DTSFILE:
*$SRV          PROG  11  IGNORED, DUPLICATE ID ON CONTROL FILE
* = INDICATES THAT USER ID WAS NOT ADDED
IESU0150I      0 RECORDS WERE ADDED TO VSE CONTROL FILE
IESU0010I IESBLDUP EXECUTION COMPLETED, END OF JOB
1S55I LAST RETURN CODE WAS 0000
EOJ MIGRAT MAX.RETURN CODE=0000                                DATE 10/13/98,CLOCK 07/40/05,DURATION 00/00/17
```

Figure 9. Example for a Status Report Output (Part 2 of 2)

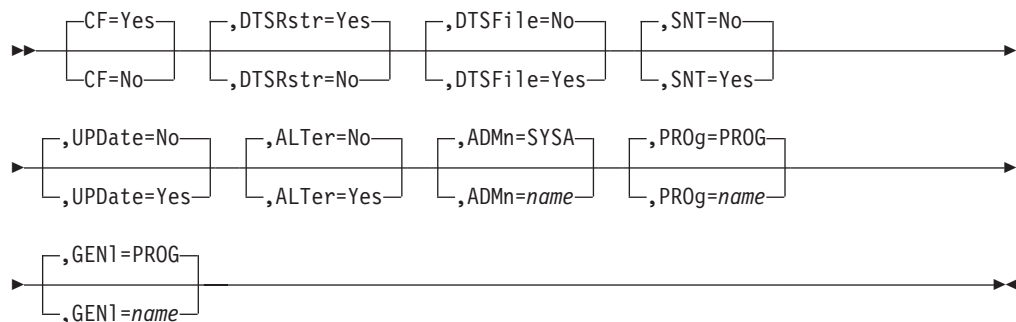
The Control Statement

The control statement must follow directly the // EXEC IESBLDUP statement. Processing options are selected by keywords and may be specified on multiple statements.

Migration Utility IESBLDUP

Control Statement Format

The control statement format is shown below.



Keyword Description

Default values are underlined.

CF=Yes|No

Specifies whether a copy of an old VSE Control File is to be processed.

Yes A copy of an old VSE Control File will be read from the tape drive assigned to SYS005. If only one tape drive is available SYS005 must be unassigned. After reading the DTSFILE copy, the VSE Control File copy is mounted on the drive assigned to SYS004. The records of the old VSE Control File are added to the current VSE Control File. Records with duplicate keys and reserved VSE/ESA record types (IES records) are ignored. If this option is specified, DTSRSTR=YES must also be specified.

No No copy of an old VSE Control File is to be processed.

DTSRstr=Yes|No

Specifies whether a copy of an old VSE/ICCF DTSFILE is to be processed.

Yes If CF=YES is specified:

A copy of an old VSE/ICCF DTSFILE will be read from the tape unit assigned to SYS004. The VSE/ICCF user profiles are copied to the current DTSFILE. If there is a user ID on the old DTSFILE which does not exist on the current DTSFILE, it is copied to the current DTSFILE from the old DTSFILE.

If CF=NO is specified:

A copy of an old VSE/ICCF DTSFILE will be read from the tape unit assigned to SYS004. For each ID found in the file, a user profile is created and added to the current VSE Control File and current DTSFILE. If the ID is already in the current VSE Control File, the ID of the old DTSFILE is ignored.

No No copy of an old DTSFILE is to be read.

DTSFile=Yes|No

Specifies whether VSE/ESA user profiles are to be created for user IDs defined on the current VSE/ICCF DTSFILE.

Yes The current DTSFILE will be read and for each ID found, a user

profile will be created and added to the current VSE Control File. If the ID is already on the VSE Control File, it is ignored.

No No user profiles are to be created from the current DTSFILE.

SNT=Yes|No

Specifies whether the CICS sign-on table (DFHSNT) is to be processed. Refer also to "Note on DFHSNT" under "Introducing the IESBLDUP Utility" on page 51.

Yes Table DFHSNT is searched for a matching user ID when creating and adding user profiles to the current VSE Control File. The ID is compared to the USERID field in DFHSNT. If the IDs and passwords match, the DFHSNT entry is used to provide the CICS data for the new user ID (like CICS TIMEOUT and security keys). DFHSNT is not scanned for users being added from the old VSE Control File. The CICS data for them is taken from the old user profile.

If a DFHSNT entry contains a 4-8 character user ID and a three to six character password, and if there is no matching user ID defined on the current VSE Control File, the ID is added as a type 3 user.

No Table DFHSNT is not be processed. Default values are used for the CICS data in the current VSE Control File. No users are added from the DFHSNT.

UPDate=Yes|No

Specifies whether the current VSE Control File and VSE/ICCF DTSFILE are to be updated.

Yes New users are added to the current VSE Control File and the current DTSFILE is updated with new and changed VSE/ICCF users.

No The current VSE Control File is not updated and the job DTRMIGR is not submitted to update the current DTSFILE. The job is executed in edit mode only.

ALTer=Yes|No

Specifies whether VSE/ICCF options that are incompatible with VSE/ESA are to be altered.

Yes When adding VSE/ICCF users (type 1 and 2), the user's VSE/ICCF options are checked for VSE/ESA incompatible options. If the user being added has incompatible options specified, they will be altered to be VSE/ESA compatible. Refer to "Maintaining User Profiles" in *VSE/ESA Administration* for a description of the VSE/ICCF options.

No Incompatible VSE/ICCF options are not to be altered.

ADMn=SYSAlname

Specifies the user profile to be used as a "model" for administrators (bit 2 of OPTB=1) if the input is from the current DTSFILE or an old DTSFILE. This profile will supply the default data for the new user. The model user profile specified must be defined in the current VSE Control File before IESBLDUP is run.

PROg=PROGIname

Specifies the user profile to be used as a "model" users of type programmer (bit 2 of OPTB=0) if the input is from the current DTSFILE or

Migration Utility IESBLDUP

old DTSFILE. This model user profile will supply the default data for the new user. The user profile specified must be defined in the current VSE Control File before IESBLDUP is run.

GENI=PROGName

Specifies the user profile to be used as a "model" when adding users to VSE/ESA from the CICS sign-on table (DFHSNT) as type 3 users. The user profile specified must be defined in the current VSE Control File before IESBLDUP is run. If the profile specified is not of type 3, the VSE/ICCF options are ignored and the user is added as a type 3 user.

Recovering from Migration Errors

If you encounter an error while migrating the (old) VSE Control File, you may have to perform the following steps to **restore** the previously saved (new) VSE Control File. Refer also to "Precautionary Measures" on page 57.

1. On the VSE/ESA Selection Panel, select the dialog *Resource Definition*.
2. Select the COPY function from the *Display or Process a File* dialog.
3. At the *Copy Files* panel specify the file as an output file and the second (input) file as tape.
4. Enter the file ID and name.
5. The following *must* be specified at the *Copy Files - Tape Specification* panel:
 - RECORD FORMAT=3 (variable-length blocked records)
 - BLOCK SIZE=4000

Note: Do **not** submit the job for immediate execution because you must edit it before submission.

6. Edit the job and insert the REPLACE option as shown in Figure 10 on page 65.
7. Submit the job.

Additional VSE Control File Information:

If migration fails during initial installation, the VSE Control File is not damaged. The status of the new control file is: hardware configuration information has been migrated but not user profiles and related data. User profiles may be migrated later in a separate job step using skeleton IESBLDUP. Refer also to Figure 6 on page 58.

Example Job for Restoring the VSE Control File:

```

* $$ JOB JNM=RESCNTR,CLASS=A,DISP=D,NTFY=YES
// JOB  AMAD  COPY FILE
// DLBL COPYOUT,'VSE.CONTROL.FILE',,VSAM,                X
           CAT=VSESPUC
// TLBL COPYIN,'CONTROL.FILE',,CF0001
// ASSGN SYS004,181
// EXEC IDCAMS,SIZE=AUTO
  REPRO INFILE (COPYIN -
    ENVIRONMENT (BLOCKSIZE (4000 ) -
      RECORDFORMAT (VARBLK) -
      STDLABEL -
      PRIMEDATADEVICE (2400) REW)) -
    OUTFILE (COPYOUT) -
    NOREUSE REPLACE
/*
/&
* $$ EOJ

```

Figure 10. Restoring the VSE Control File

Migration Utility IESBLDUP

Part 3. Miscellaneous Utilities and Programs

Chapter 6. Batch Program IESUPDCF

The batch utility program **IESUPDCF** allows the system administrator to maintain user profiles in the VSE Control File (IESCNTL) and in the VSE/ICCF DTSFILE. With this program, you can **ADD**, **ALTER**, and **DELETE** user profiles. IESUPDCF helps you save time when configuring user profiles.

Preparing to Use IESUPDCF

The following section describes the procedures you should perform before using IESUPDCF.

Planning for User Profiles

With VSE/ESA you can use three types of user profiles.

A VSE/ICCF (short form: ICCF) user profile is a type 1 or type 2 user profile with a 4 character user ID. It is defined in the VSE control file (IESCNTL) and also in the VSE/ICCF DTSFILE.

Model profiles for type 1 and type 2 user profiles are provided:

Type 1 User Profile

Valid for the System Administrator. Access to all VSE/ESA functions, including ICCF.

Type 2 User Profile

Valid for Operators and Programmers. Access to most of the VSE/ESA functions, including ICCF.

Type 3 User Profile

Valid for general users (and Type 1 and Type 2 users with a user ID of 4 to 8 characters). Access to selected functions, but not to ICCF.

Information for ICCF users is recorded in the IESCNTL and in the DTSFILE. Information for type 3 user profiles is only recorded in the IESCNTL file. For the following discussion you should know that ICCF-related definitions (PASSWORD and LIBRARY) are recorded in two places: in the DTSFILE and in the IESCNTL file.

Figure 11 on page 76 shows skeleton IESUPDCF. It is shipped in ICCF library 59. You have to change this skeleton to add, alter, or delete user profiles. Before you change skeleton IESUPDCF, you should carefully plan for the types of users you want to create.

Preparing Skeleton IESUPDCF

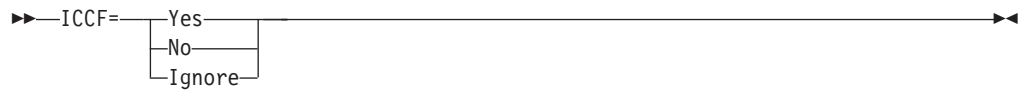
You have to prepare skeleton IESUPDCF according to your needs. This may include:

- Set the **ICCF** parameter for all users referred to in the job.
- Insert **ADD** statements for adding user profiles.
- Insert **ALTER** statements for altering user profiles.
- Insert **DELETE** statements for deleting user profiles.

The following sections have more details.

Setting the ICCF Parameter in Skeleton IESUPDCF

With the setting of the ICCF parameter, you control the generation of job DTRUPD, which updates the DTSFILE. You must enter either Yes, No, or Ignore. There is no default.



ICCF=YES

IESUPDCF updates user profiles in the control file (IESCNTL). For ICCF users, IESUPDCF updates user profiles in the DTSFILE. Therefore, a new job DTRUPD is generated.

The following describes how specifying ICCF=YES affects the ADD, ALTER, and DELeTe statements:

ADD The new user is added to the IESCNTL control file. The definitions of the model user profile are used as default.

If the model profile is for an ICCF user and the new user ID is 4 characters long, then the new user will also be an ICCF user. Thus, the DTSUTIL statement is generated for job DTRUPD.

ALTER The user definition is altered in the IESCNTL control file. If the user profile is for an ICCF user, a DTSUTIL statement is generated for job DTRUPD.

DELeTe

The user definition in the IESCNTL control file is deleted. If the user profile is an ICCF user, the DTSUTIL statement for job DTRUPD is generated.

ICCF=No

No update of the DTSFILE is performed. This means that you cannot ADD or DELeTe ICCF users. In addition, you cannot ALTER the password or the ICCF library of ICCF users.

The following describes how specifying ICCF=NO affects the ADD, ALTER, and DELeTe statements:

ADD If the model profile is for an ICCF user, then the ADD statement is ignored, and an error message is inserted into the listing. If the model profile is not for an ICCF user, the new user is added to the IESCNTL control file.

ALTER ICCF-related definitions (PASSWORD and LIBRARY) are not altered in the IESCNTL control file.

DELeTe

If the user profile is an ICCF user, the statement is ignored and an error message

Batch Program IESUPDCF

is inserted into the listing. For type 3 user profiles, the definition in the control file is deleted.

ICCF=Ignore

You can ADD, ALTER, or DELETE any user in the control file (IESCNTL). For VSE/ICCF users, however, the DTSFILE is not updated. IGNORE must be used if the control file is not related to an ICCF subsystem. With VSE/ESA 2.4, however, a single VSE Control File is recommended.

The following describes how specifying ICCF=IGNORE affects the ADD, ALTER, and DELETE statements:

ADD The new user is added to the control file.

ALTER All specified parameters are altered in the control file. If specified, the LIBRARY parameter is ignored, since it is only relevant for ICCF subsystems.

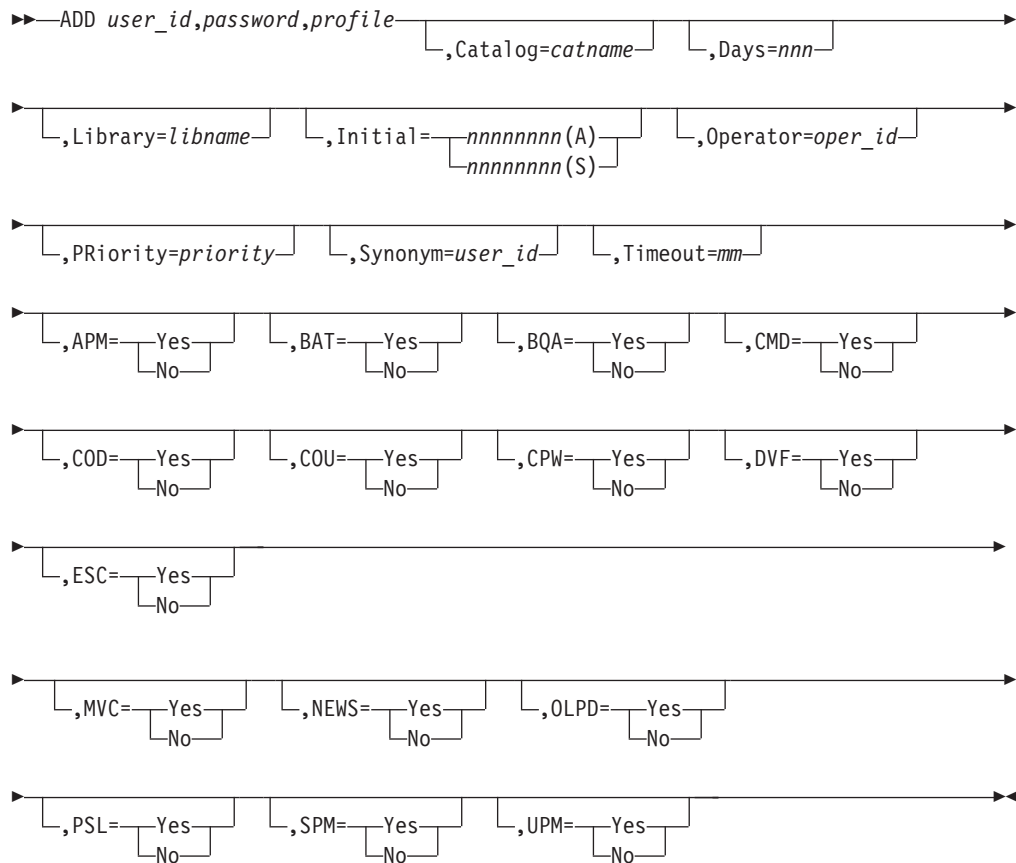
DELETE The user definition in the control file is deleted, independent of the user profile type.

Refer also to item 2 in skeleton IESUPDCF, shown in Figure 11 on page 76.

Adding a User ID in Skeleton IESUPDCF

To **ADD** a user ID, you insert the following statement into skeleton IESUPDCF:

Batch Program IESUPDCF



The first three parameters are mandatory; the rest are optional. **Do not change the order of the mandatory parameters.**

Mandatory Parameters

Note: Each ADD statement may use one or more physical lines. A continuation line is indicated by the continuation character “-” as the last character in the previous line. The continuation character must be preceded by a blank or a comma. The required parameters must be specified together with the ADD statement on *one* line.

user-id

The user ID which identifies the user to the system. It must be 4-8 alphanumeric characters long and may include the characters @, #, or \$. Blanks are not allowed.

Note: For ICCF users, the user ID can only be 4 characters long.

password

This is the password associated with the user ID. It can be 3 - 8 alphanumeric characters long and may include the characters @, #, or \$. Blanks are not allowed.

profile

This is the identification (user ID) of a user already defined to the system and used as a model for the new user. It must be 4-8 characters long. Using optional parameters you can alter the defaults for the new user ID.

Optional Parameters

This section describes the optional parameters you can use with either the ADD statement, or the ALTER statement (described on page 75).

Catalog=*catname*

The name of the user's default VSE/VSAM catalog (IJSYSCT). This parameter is not available for type 3 users.

Days=*nnn*

The number of days before the user's password expires. Specify a number between 0 and 365. If you enter 0, the password will **not** expire.

Library=*libname*

The user's primary ICCF library. This value may be 4 digits in length. When specifying:

- ICCF=IGNORE, the LIBRARY parameter is ignored, since it is only relevant for ICCF subsystems.
- ICCF=NO, you cannot change the library for an VSE/ICCF user.

Initial=*nnnnnnnn(A)nnnnnnnn(S)*

Initial function performed at sign on. You can use up to eight alphanumeric characters. The value must be followed by the type specification:

- (A) - if the initial function is an application, or
- (S) - if the initial function is a selection panel.

For example:

INITIAL=FUNCNAME(A) for an application.

Operator=*oper-id*

3-character operator identification for CICS. The ID must be unique.

PWDIPassword=*password*

It can be 3 - 8 alphanumeric characters long and may include the characters @, #, or \$. Blanks are not allowed. You cannot change the password for an ICCF user when specifying ICCF=NO. This parameter is only to be used together with the ALTER statement.

Priority=*priority*

The value which CICS uses for the dispatching priorities of the user. Enter a number from **0 - 255**. 0 is the highest priority; 255 the lowest.

Synonym=*user-id*

This defines the user ID to be used as a model for synonyms. VSE/ESA provides synonyms for users SYSA, PROG, and OPER. These can be used as models for other users.

Timeout=*mm*

Gives the value in minutes used by CICS to initiate sign off after the value specified has elapsed since the latest terminal activity. You can specify a value from 0 to 60. The value you specify is always rounded up to a multiple of 5 minutes. A value of 0 means no time out. 0 should be specified for ICCF users.

APM=Yes|No

When set to "Yes", you can create and maintain application profiles. This is only valid for the administrator (type 1) user profiles.

BAT=Yes|No

When set to "Yes", you can submit jobs for batch processing.

Batch Program IESUPDCF

BQA=Yes|No

When set to “Yes”, you can manage all VSE/POWER jobs of an ICCF user.

CMD=Yes|No

When set to “Yes”, you can enter system console commands from the *System Console* dialog. This authorization is **not** available for general (type 3) user profiles.

COD=Yes|No

When set to “Yes”, you will be asked for confirmation when you try to delete ICCF members.

COU=Yes|No

When set to “Yes”, all console output is shown.

CPW=Yes|No

When set to “Yes”, you can change your password (press PF10 at the SIGNON panel). Otherwise, your system administrator must change your password using User Profile Maintenance.

DVF=Yes|No

When set to “Yes”, you can define and delete VSE/VSAM files, libraries, alternate indexes, and alternate names. This authorization is **not** possible for general (type 3) user profiles.

ESC=Yes|No

When set to “Yes”, you can *escape* to CICS. This lets you leave the Interactive Interface and work directly with CICS.

MVC=Yes|No

When set to “Yes”, you can process VSE/VSAM catalogs and define and delete VSE/VSAM space. This authorization is **not** available for general (type 3) user profiles.

NEWS=Yes|No

When set to “Yes”, the system displays *news items* to you. News items are messages which the system displays when you sign on or when you are already signed on.

OLPD=Yes|No

When set to “Yes”, you can delete Online Problem Determination (OLPD) incident records from the system. This authorization is **not** available for general (type 3) user profiles.

PSL=Yes|No

When set to “Yes”, you will have a private sublibrary (primary user ID).

SPM=Yes|No

When set to “Yes”, you can create and maintain selection panels. This is only valid for the administrator (type 1) user profiles.

UPM=Yes|No

When set to “Yes”, you can create and maintain user profiles. This is only valid for the administrator (type 1) user profiles.

Altering a User ID in Skeleton IESUPDCF

To **ALTER** a user ID, you insert the following statement into skeleton IESUPDCF:

```

➤➤—ALTER userid—┐,Catalog=catname—┐,Days=nnn—┐,Library=libname—┐
┐,Initial=—┐nnnnnnnn(A)—┐┐,Operator=oper-id—┐┐,PWD=password—┐
┐nnnnnnnn(S)—┐┐PAssword=password—┐
┐,PRiority=priority—┐┐,Synonym=user-id—┐┐,Timeout=mm—┐
┐,APM=—┐Yes—┐┐,BAT=—┐Yes—┐┐,BQA=—┐Yes—┐┐,CMD=—┐Yes—┐
┐No—┐┐No—┐┐No—┐┐No—┐
┐,COD=—┐Yes—┐┐,COU=—┐Yes—┐┐,CPW=—┐Yes—┐┐,DVF=—┐Yes—┐
┐No—┐┐No—┐┐No—┐┐No—┐
┐,ESC=—┐Yes—┐┐,MVC=—┐Yes—┐┐,NEWS=—┐Yes—┐
┐No—┐┐No—┐┐No—┐
┐,OLPD=—┐Yes—┐┐,PSL=—┐Yes—┐┐,SPM=—┐Yes—┐┐,UPM=—┐Yes—┐
┐No—┐┐No—┐┐No—┐┐No—┐

```

ALTER checks the user types and performs those changes allowed for the specific user types. For the ALTER statement the first parameter (*userid*) is mandatory. The user ID identifies the user to the system. It must be 4-8 alphanumeric characters long and may include the characters @, #, or \$. Blanks are not allowed. You also must specify at least **one** additional parameter from the optional parameters shown in the ALTER statement syntax. The optional parameters you can use are described in detail under “Optional Parameters” on page 73. You may have more than one ALTER statement for the same user profile.

Each ALTER statement may use one or more physical lines. A continuation line is indicated by the continuation character “—” as the last character in the previous line. The continuation character must be preceded by a blank or a comma. The *user-id* must be specified on the same line as the ALTER statement.

Batch Program IESUPDCF

Deleting a User ID in Skeleton IESUPDCF

To **DE**lete a user ID, you insert the following statement into skeleton IESUPDCF:

```
►—DELeTe user_id—————►
```

For the **DE**lete statement, the user ID identifies the user which is to be deleted. It must be 4-8 alphanumeric characters long and may include the characters @, #, or \$. Blanks are not allowed.

Skeleton IESUPDCF

The following figure shows skeleton IESUPDCF, shipped in ICCF library 59. Use this skeleton to **ADD**, **ALTER**, or **DE**lete your user profiles.

```
* $$ JOB JNM=IESUPDCF,CLASS=0,DISP=D
* $$ PUN DISP=I,CLASS=0,PRI=9
// JOB IESUPDCF
// OPTION NOLOG
*
* THIS SKELETON MAY BE USED BY THE ADMINISTRATOR TO GENERATE A
* JOB FOR BATCH USER PROFILE MAINTENANCE.
* 1. IF THE CONTROL FILE BELONGS TO A CICS WITHOUT ICCF AND THIS
* CICS DOES NOT SHARE THE CONTROL FILE WITH CICS/ICCF,
* ADJUST THE '// DLBL' STATEMENT TO MAINTAIN
* USER PROFILES IN THE RELATED CONTROL FILE.
* 2. SUPPLY AN OPERAND FOR THE ICCF PARAMETER, VALID OPERANDS ARE:
*   Yes    ... UPDATE USER PROFILES IN CONTROL FILE (CICS) AND
*           IN THE DTSFILE (ICCF).
*   No     ... UPDATE USER PROFILES IN CONTROL FILE ONLY.
*           INHIBIT CHANGES TO ICCF RELATED INFORMATION.
*   Ignore ... UPDATE USER PROFILES IN CONTROL FILE ONLY.
*           THIS VALUE MUST BE USED IF THE CONTROL FILE
*           IS USED IN CICS SUBSYSTEMS RUNNING WITHOUT ICCF.
* 3. INSERT THE ADD, ALTER AND DELETE STATEMENTS THAT YOU NEED TO
* MAINTAIN USER PROFILES.
*   SAMPLE STATEMENTS:
*   =====
*   * TEXT    ... A COMMENT LINE
*   Add      USERID,PASSWD,PROFILE(,OPTIONAL PARAMETERS)
*   Alter    USERID(,OPTIONAL PARAMETERS)
*   Delete   USERID
*   EXPLANATION OF PARAMETERS:
*   =====
*   1. REQUIRED AND POSITIONAL PARAMETERS:
*   -----
*   USERID  ... THE ID OF THE USER ( ADD, ALTER, DELETE )
*             ( 4-8 CHARACTER / 4 CHARACTER FOR ICCF USER )
*   PASSWD   ... THE PASSWORD OF THE USER ( ADD )
*             ( 3-8 CHARACTERS )
*   PROFILE  ... THE ID OF THE USER USED AS PROFILE FOR
*             THE NEW USER ( ADD )
*             ( 4-8 CHARACTER / 4 CHARACTER FOR ICCF USER )
```

Figure 11. Skeleton IESUPDCF (Part 1 of 3)

```

*      2. OPTIONAL PARAMETERS IN ADD/ALTER STATEMENT:
*      -----
*      Catalog= ... THE DEFAULT CATALOG OF THE USER
*                EXAMPLE: CAT=VSESPUC
*      Days= ... NUMBER OF DAYS IN EXPIRATION INTERVAL
*                EXAMPLE: DAYS=20 ( RANGE: 0-365)
*      Library= ... Primary ICCF library ( only ICCF users )
*                EXAMPLE: LIB=20
*      Initial= ... Initial function at SIGNON
*                EXAMPLE: INIT=APPLNAME(A) ... FOR APPLICATION
*                        INIT=SELNAME(S) ... FOR SELECTION P.
*      Natlang= ... NATIONALLANGUAGE_INDICATOR
*                EXAMPLE: NAT=E ( for English)
*      OID = ... OID CHARACTERS
*                EXAMPLE: OID=ABC ( max. 20 Characters )
*      Operator= OPERATOR ID
*                EXAMPLE: OPER=OPE
*      PWD= ... USER PASSWORD
*      PAssword= EXAMPLE: PWD=PASSWD ( 3-8 Characters )
*      Priority= ... OPERATOR PRIORITY
*                EXAMPLE: PRIOR=5 ( RANGE: 0-255 )
*      Revoke= ... REVOKE DATE
*                EXAMPLE: R=01/31/99 ( Format mm/dd/yy )
*      Synonym= ... SYNONYMS MODEL
*                EXAMPLE: SYNONYM=SYNS ( 4-8 CHARACTERS )
*      Timeout= ... TIMEOUT INTERVAL
*                EXAMPLE: TIME=20 ( VALUES: 0,5,10,...,60 )
*      APM=Yes|No ... APPLICATION PROFILE MAINTENANCE
*      BQA=Yes|No ... MANAGE ALL BATCH QUEUES
*      CMD=Yes|No ... ENTER CONSOLE COMMANDS
*      COU=Yes|No ... FULL OUTPUT ON SYSTEM CONSOLE
*      COD=Yes|No ... CONFIRM ON DELETE
*      DVF=Yes|No ... DEFINE VSAM FILES
*      ESC=Yes|No ... ESCAPE TO CICS
*      MVC=Yes|No ... MANAGE VSAM CATALOGS
*      NEWS=Yes|No ... DISPLAY NEWS TO USER
*      OLPD=Yes|No ... DELETE OLPD INCIDENTS
*      PSL=Yes|No ... OWNS A PRIVATE SUBLIBRARY
*      BAT=Yes|No ... SUBMIT TO BATCH
*      SPM=Yes|No ... SELECTION PANEL MAINTENANCE
*      UPM=Yes|No ... USER PROFILE MAINTENANCE
*      XRF=Yes|No ... XRF SIGNOFF
*      - ... CONTINUATION CHARACTER
*      4. DELETE BLOCK 'UPDPL', IF YOU DO NOT WANT TO MAINTAIN
*          THE PRIMARY LIBRARY.
*
*      * =====> UPDATE NEXT LINE IF NECESSARY (SEE 1.)
*      // DLBL IESCNTRL,'VSE.CONTROL.FILE',,VSAM,CAT=VSESPUC
*
*      // EXEC PROC=DTRICCF
*      // EXEC IESUPDCF,SIZE=64K
*
*      * =====> SUPPLY AN OPERAND FOR THE ICCF PARAMETER (SEE 2.)
*      ICCF=
*
*      * =====> INSERT STATEMENTS HERE (NO COMMENT '*' IN FIRST COLUMN, SEE 3.)

```

Figure 11. Skeleton IESUPDCF (Part 2 of 3)

Batch Program IESUPDCF

```
* /*
// IF $RC=0 THEN
// GOTO STEP2
// IF $RC=4 THEN
// GOTO ERROR
// IF $RC>6 THEN
// GOTO END
// LOG
* ==> JOB 'DTRUPD' CREATED, ENSURE THAT THIS JOB IS EXECUTED NEXT
// NOLOG
// IF $RC=2 THEN
// GOTO STEP2
/. ERROR
// LOG
* ==> ERRORS IN INPUT DATA, STATEMENT(S) FLAGGED IN LISTING
// NOLOG
/. STEP2
*
* ==> DELETE BLOCK 'UPDPL', IF REQUIRED (SEE 4.)
* ***** BEGIN OF BLOCK 'UPDPL' *****
// EXEC PROC=IESUPDPL
* /*
* ***** END OF BLOCK 'UPDPL' *****
*
/. END
* /&
* $$ E0J
```

Figure 11. Skeleton IESUPDCF (Part 3 of 3)

Using IESUPDCF

After making changes in skeleton IESUPDCF, submit the job for processing. Once the job is processed, check the output listing to see whether the job DTRUPD was created. Please note that this job will only be created when you specify ICCF=YES. If so:

- Check the system console, since job DTRUPD prompts you to disconnect the DTSFILE and waits for a response.
- Disconnect the DTSFILE (/DISC DTSFILE) and reply to the suspended job.
- Reconnect the DTSFILE after the job has terminated (/CON DTSFILE).
- If you have specified ICCF=YES, you will have two listings with the name IESUPDCF. (With ICCF=NO or ICCF=IGNORE you will get one IESUPDCF listing). Check both of them for flagged statements and return codes.

Return Codes Issued by IESUPDCF

- | | |
|---|---|
| 0 | No error. Job DTRUPD was not generated. |
| 2 | No error. Job DTRUPD was generated. |
| | User action: |
| | <ul style="list-style-type: none">• Ensure that job (DTRUPD) is started immediately.• Disconnect the DTSFILE when prompted on the system console. |
| 4 | The program has detected one or more invalid user statements in the job. The invalid statements are flagged in the listing. All valid statements are processed. Job DTRUPD was not generated. |

User action:

- Examine the job listing.
- Correct the flagged job statements.
- Delete statements that are not flagged from the job, because they have been processed before.
- Submit the corrected job again.

- 6 The program has detected one or more invalid user statements in the job. The invalid statements are flagged in the listing. All valid statements are executed. Job DTRUPD was generated.

User action:

- Submit job DTRUPD.
- Examine the job listing.
- Correct the flagged job statements.
- Delete statements that are not flagged from the job, because they have been processed before.
- Submit the corrected job again.

- 8 The ICCF statements were ignored. ICCF=NO was specified in the job, but there was at least one statement that tried to alter an ICCF user definition. This statement was ignored.

User action:

If the erroneous statement is to be processed:

- Specify ICCF=YES.
- Delete all statements that are not flagged, because they have been processed before.
- Submit the corrected job again.

- 16 The program has been canceled due to severe errors.

User action:

- Examine the listing to determine the reason. The error might have been caused by one of the following:

CDLOAD

The program was unable to load the DTSFILE I/O routine DTSFILRT.

CONTROL FILE

A VSE/VSAM macro caused an error.

GETVIS

The partition GETVIS area is too small for the job.

Batch Program IESUPDCF

Example of Completed Skeleton IESUPDCF

The following is an example of a completed IESUPDCF skeleton, which shows ADDing, ALTering, and DELeting users.

```
* $$ JOB JNM=IESUPDCF,CLASS=0,DISP=D
* $$ PUN DISP=I,CLASS=0,PRI=9
// JOB IESUPDCF
// OPTION NOLOG
*
* THIS SKELETON MAY BE USED BY THE ADMINISTRATOR TO GENERATE A
* JOB FOR BATCH USER PROFILE MAINTENANCE.
*
* ...
* ... Description is deleted.
* ... (See skeleton IESUPDCF)
*
* =====> UPDATE NEXT LINE IF NECESSARY (SEE 1.)
// DLBL IESCNTRL,'VSE.CONTROL.FILE',,VSAM,CAT=VSESPUC
*
// EXEC PROC=DTRICCF
// EXEC IESUPDCF,SIZE=64K
*
* =====> SUPPLY AN OPERAND FOR THE ICCF PARAMETER (SEE 2.)
ICCF=YES
*
* =====> INSERT STATEMENTS HERE (NO COMMENT '*' IN FIRST COLUMN, SEE 3.)
ADD NEWUSR,PASSWD,OLDUSR, -
    DAYS=30,TIMEOUT=15, -
    CPW=YES,PSL=YES
ALT MYUSER, PWD=NEWPWD, -
    DAYS=30,TIMEOUT=15, -
    CPW=YES,PSL=YES
DEL OLDUSR
/*
// IF $RC=0 THEN
// GOTO STEP2
// IF $RC=4 THEN
// GOTO ERROR
// IF $RC>6 THEN
// GOTO END
// LOG
* =====> JOB 'DTRUPD' CREATED, ENSURE THAT THIS JOB IS EXECUTED NEXT
// NOLOG
// IF $RC=2 THEN
// GOTO STEP2
/. ERROR
// LOG
* =====> ERRORS IN INPUT DATA, STATEMENT(S) FLAGGED IN LISTING
// NOLOG
/. STEP2
*
* =====> DELETE BLOCK 'UPDPL', IF REQUIRED (SEE 4.)
* ***** BEGIN OF BLOCK 'UPDPL' *****
// EXEC PROC=IESUPDPL
* /*
* ***** END OF BLOCK 'UPDPL' *****
/. END
* /&
* $$ E0J
```

Figure 12. Example of a Completed Skeleton IESUPDCF

Chapter 7. Utilities DTRSETP, DTRIINIT, and JCL Analyzer

This chapter describes the VSE/ESA utilities DTRSETP, DTRIINIT, and introduces the JCL Analyzer. You can use DTRSETP to tailor or create CPUVAR n and SETPARM procedures, and DTRIINIT to load jobs into the VSE/POWER reader queue. The JCL Analyzer is a tool for JCL job stream analysis.

DTRSETP Utility

You use DTRSETP when tailoring or creating CPUVAR n procedures or user-written SETPARM procedures. The program must be run in a batch partition.

Using SETPARM Procedures

Each CPU requires its own CPUVAR n procedure, where n is the CPU number. For a single CPU system, VSE/ESA provides procedure CPUVAR1. VSE/ESA uses the SETPARM procedure CPUVAR1, or the appropriate CPUVAR n procedure, to save system variables for startup from one IPL to the next. The IBM manual *VSE/ESA Administration* shows the CPUVAR1 procedure shipped with VSE/ESA.

If you create SETPARM procedures for purposes other than startup, use names different from CPUVAR n .

A SETPARM procedure must be stored in a VSE library and its member type must be **PROC**. A SETPARM procedure can *only* contain SETPARM statements. Each record (statement) of such a procedure must follow the syntax for SETPARM statements as described in the IBM manual *VSE/ESA System Control Statements*. The only exception is that *no continuation sign* is allowed.

Using DTRSETP

The following sample statements set the environment variable (XENVNR) in CPUVAR1 to B. You can supply the operation parameters (SET, WAIT, and DELETE) of DTRSETP via SYSIPT (1) or as a parameter string (2):

1. Via SYSIPT:

```
// EXEC DTRSETP,SIZE=AUTO,PARM='CPUVAR1'  
    SET XENVNR=B  
/*
```

2. As a parameter string:

```
// EXEC DTRSETP,SIZE=AUTO,PARM='CPUVAR1;;SET XENVNR=B'  
/*
```

DTRSETP supports three parameters. They are enclosed in single quotes (') and separated by semicolons (;). A parameter string (in the EXEC statement) may contain up to 100 characters. Refer to the *VSE/ESA System Control Statements* manual for details.

- *Parameter 1* defines the name of the SETPARM procedure. In the example, it is the name of the startup procedure CPUVAR1.
- *Parameter 2* defines the sublibrary in which the procedure resides. The syntax is "libraryname.sublibraryname". The default is IJSYSRS.SYSLIB, as in the example.
- *Parameter 3* defines the **operation** to be performed (SET in the example used). This parameter can be repeated within one // EXEC statement until the maximum of 100 characters has been reached.

DTRSETP Utility

DTRSETP supports three operations:

SET Adds a new SETPARM statement to a procedure or changes the value of an existing SETPARM statement.

DELETE

Deletes a SETPARM statement from a procedure.

WAIT Waits until a SETPARM statement is present in a procedure with a specific value.

DTRSETP Operations

You must observe the rules for JCL symbolic parameters when specifying a SET, WAIT, or DELETE operation. Refer to the IBM manual *VSE/ESA Guide to System Functions* for details about symbolic parameters.

Operation SET

►►—SET *name=value* —————►
 └──comment──┘

The parameters have the following meaning:

name Specifies the name of the variable you want to define or to which a new value and a new comment is to be assigned. You can specify 1 to 7 characters (without leading &).

value Specifies the value you want to assign to the variable. You can specify a character string of up to 50 characters. If the string contains special or national characters, it must be enclosed in single quotes (').

comment

You can enter a comment of up to 56 characters. Any character is allowed. Do not enclose the comment in /* and */ characters; the comment would be ignored.

Note: The total length of a SET specification must not exceed 59 characters. That is, when adding the individual lengths of name, value, and comment, the maximum result allowed is 59.

Operation WAIT

WAIT allows you to synchronize VSE partitions at the JCL level. The SETPARM procedure specified is read at each compare interval until the maximum wait time is exhausted. DTRSETP compares the value of the specified variable with the value in the SETPARM procedure. If the values match, the condition is set to TRUE and the loop is left. If the values do not match, the condition is set to FALSE and the compare is repeated at the next interval.

To reach condition TRUE, a program in another VSE partition must SET the variable in the SETPARM procedure to the agreed value.

Refer also to the IBM manual *VSE/ESA Administration* for further details about partition synchronization.

```

▶▶—WAIT name=value,MAX=maximum—┐,INT=10
                                └─,INT=interval—▶▶

```

The parameters have the following meaning:

name Specifies the name of the variable you want to compare.

value Specifies the value to be compared. You can specify a character string of up to 50 characters. Observe the rules for symbolic parameters. If the string contains special or national characters, it must be enclosed in single quotes (').

maximum

Specifies the maximum wait time in seconds (1 to 999999).

interval

Specifies the compare interval time in seconds (1 to 60). The default value is 10 seconds.

Operation DELETE

```

▶▶—DELETE name—▶▶

```

name Specifies the name of the variable you want to delete.

Restrictions When Using DTRSETP

- In a SETPARM statement, no continuation sign is allowed. A SETPARM statement with a continuation sign will be ignored and will cause an error message.
- There is a maximum of 150 lines per SETPARM procedure. If an overflow occurs, proceed as follows:
 - Use the librarian program LIBR to list the SETPARM procedure.
 - Identify redundant lines.
 - Delete the redundant lines with the utility program DTRSETP.
- Within PARM=, parameters are separated by semicolons (;). Therefore, a semicolon cannot be used within a parameter.
- A WAIT operation cannot be specified together with a SET or DELETE operation for one call of DTRSETP. Such a statement would be ignored and would cause an error message.
- In case of certain errors, DTRSETP does not overwrite the source member but creates a new member. The member name is **S\$E\$T\$P\$** and the member type is **PROC**. This is true for the following error situations:
 - Syntax error in SETPARM statement definition.
 - Syntax error in operation definition.
 - WAIT and SET/DELETE operations are defined together.
 - The maximum number of SETPARM statements exceeds 150.

Return Codes

DTRSETP issues return codes to indicate the processing status:

00 Processing was completed successfully.

DTRSETP Utility

- 02** SETPARM statement to be deleted does not exist. No error message is issued and processing continues.
- 04** The SETPARM procedure of the specified name does not exist or is found to be empty. An error message is issued and processing continues.
- 06** The maximum wait time has elapsed while condition was still FALSE. An error message is issued and processing continues.
- 08** One of the following errors occurred:
- Maximum number of SETPARM statements (150) exceeded.
 - Syntax error in SETPARM statement.
 - Syntax error during a SET, WAIT, or DELETE operation.
 - WAIT operation mixed with SET/DELETE operation(s) during one program call.
- The function is partially completed. An error message is issued and processing continues.
- 12** One of the following errors occurred:
- Syntax error in procedure name or library/sublibrary name.
 - Library full condition.
 - Library/sublibrary not found.
- Processing is terminated and an error message is issued.
- 16** A phase was not found or GETVIS space is exhausted. Processing is terminated and an error message is issued.
-

DTRIINIT Utility

With the utility program DTRIINIT you can load jobs into the VSE/POWER reader queue.

You can use DTRIINIT for the following tasks:

1. To load any jobs into the VSE/POWER reader queue during normal processing.
2. To load your own jobs into the VSE/POWER reader queue during a COLD startup.

Loading Jobs During Normal Processing

To load any kind of job into the VSE/POWER reader queue you should use the skeleton SKLOAD and follow the instructions in the skeleton. Skeleton SKLOAD is documented in detail in the IBM manual *VSE/ESA Administration*.

The following is an example of an excerpt from the skeleton SKLOAD:

```
// JOB CATPAUSE
// EXEC DTRIINIT
  ACCESS IJSYSRS.SYSLIB
  LOAD PAUSEC.Z
/*
/ &
```

This job stream loads the job PAUSEC from VSE library IJSYSRS.SYSLIB into the VSE/POWER reader queue. Any job you want to load must first be cataloged into a VSE library.

The sample job shown above loads your jobs, by default, with SYSA as user ID. This means that only the system administrator can access the jobs while they are in the queue. You can, however, specify on the EXEC statement another user ID as shown below:

```
// EXEC DTRIINIT,PARM='ID01'
```

As a result, not only the system administrator but also user ID01 can access the jobs while they are in the VSE/POWER reader queue.

Loading Jobs During a COLD Startup

You can add your own jobs to the load list used by VSE/ESA during a COLD startup. Skeleton SKCOLD helps you do that. The skeleton updates procedure COLDJOBS which is processed during a COLD startup. Skeleton SKCOLD is documented in detail in the IBM manual *VSE/ESA Administration*.

The jobs you want to load must first be cataloged into a VSE sublibrary.

Loading Jobs into a System with Security Active

All jobs that are loaded into the reader queue by DTRIINIT inherit the access rights (security) of the job which is calling DTRIINIT. You can turn off this inheritance, if you add an ID statement before you call DTRIINIT, for example:

```
// ID USER=DUMMY,PWD=DUMMY
```

The user DUMMY has no access rights as defined in the access control table DTSECTAB.

For more information on secured systems, see the IBM manual *VSE/ESA Administration*.

Additional Considerations

A job must include all VSE/POWER JECL statements required. DTRIINIT does not add any statements. This is true for a COLD startup as well as for normal processing. To avoid problems when cataloging your jobs under control of VSE/POWER you may mask the source statements. These statements are unmasked when the job is loaded into the VSE/POWER reader queue. Examples of such statements are given below:

Member record	Job record in
in library:	reader queue:

\$\$\$\$...	* \$\$...
\$\$/*	/*
\$\$/&	/&

To avoid problems when using conditional JCL, you can mask the // JOB statement as shown below:

Member record	Job record in
in library:	reader queue:

\$\$// JOB	// JOB
------------	--------

JCL Analyzer

The JCL Analyzer for VSE/ESA scans JCL job streams stored in a VSE sublibrary. Its purpose is to create a Common Data Interchange Format (CDIF) file. Such a file can be used as input to the VisualAge Application Understanding tools based on OS/2 or Windows NT. The Application Understanding tool can then display a graphical analysis of the JCL job stream. You find further details at the following Web sites:

<http://www.software.ibm.com/ad/va2000>

<http://www.software.ibm.com/ad/cobol>

The JCL Analyzer is shipped as part of VSE Central Functions in VSE/ICCF library 59. It consists of the following members:

Member	Description
ARDWREAD	Detailed description of JCL Analyzer.
ARDWINST	Job stream to link and catalog the four phases that make up the JCL Analyzer.
ARDWVER	Sample job to verify installation.
ARDW101R	Sample job to run the JCL Analyzer logging.
ARDWSVAL	Sample job to load the JCL Analyzer logging module.
ARDWVSAM	Sample job to create and load the VSE/VSAM dataset used for logging.
ARDW104	CMS EXEC used to send JCL to a VSE sublibrary.

Chapter 8. Device Utilities

This chapter describes how to use the following device utilities provided by VSE/ESA:

- **CLRDK**, for clearing and preformatting extents on a CKD or an ECKD disk.
- **CRDR**, for copying the entire contents of a diskette onto another diskette.
- **INTTP**, for initializing a tape with IBM and ANSI standard volume labels.

Device Support

Disks and Diskettes

The CLRDK and CRDR utilities support all disks and diskettes supported by VSE/ESA, with the exception of the CRDR utility which does not support the IBM 3390 disk device. References to the IBM 3380 in this context imply that the IBM 3390 is used in IBM 3380 track compatibility mode.

Magnetic Tapes

The INTTP utility supports all magnetic tapes and cartridges supported by VSE/ESA. If you are using an IBM 3490E, 3490, or 3480 tape device refer to the following table:

Table 7. Defining IBM 3480/3490 Tape Devices

Tape Device	To be defined as 3490E	To be defined as 3490	To be defined as 3480
IBM 3480			x
IBM 3480 with IDRC feature		x	
IBM 3490			x
IBM 3490 with IDRC feature		x	
IBM 3490E	x		

Note: IDRC means Improved Data Recording Capability.

The INTTP utility can also be used for initializing tapes or cartridges of the IBM 9346, 9347, 9348 tape devices, and the IBM 3590 Tape Subsystem.

For more information concerning support of these tape devices, refer to the *VSE/ESA Planning* manual.

Control Statement Input

Job Control

Job control statements related to the assignment of logical units to physical devices and to label processing are used with the utilities. For more information on job control statements, see *VSE/ESA System Control Statements*.

Utility Control

Most utilities require utility control statements in addition to job control statements. Format and usage of the utility control statements are described with each utility.

Use SYSIPT for control statements and assign it to a card reader, tape, disk, or diskette extent. SYSIPT files with a record size of 81 characters (for example, produced via SYSPCH on tape) are not supported by the following utilities:

- Clear Disk (CLRDK)
- Copy and Restore Diskette (CRDR)

Storage Requirements

To run a utility in *virtual mode* (which is the normal mode for an online utility) you need no storage allocation beyond the minimum partition size.

Note: In the EXEC (job control) statement which invokes the utility, do not specify SIZE=AUTO.

If you run a utility in *real mode*, requests for GETVIS storage are issued just as in virtual mode. Therefore, an allocation of 48K of processor storage for the GETVIS area is recommended. If a utility does not require the entire 48K of processor storage allocated as partition GETVIS area, only the portion used by the utility is taken from the page pool.

On the EXEC statement which invokes an utility to be run in real mode, you must specify a SIZE value; for example:

```
// EXEC CLRDK,REAL,SIZE=24K
```

Figure 13 lists the device utilities and their respective run sizes in real mode.

VSE/ESA Utilities	Program Size (in bytes) (SIZE = Value)	Available ONLINE
Clear Disk (CLRDK)	24K (Note)	X
Copy and Restore Diskette (CRDR)	20K	X
Initialize Tape (INTTP)	10K	X
VTOC Display (LVTOC)	16K	X

Figure 13. VSE/ESA System Utilities and their Run Sizes (Real Mode)

The above figure shows the requirement for an IBM 3380 disk device. For other disks, the requirement may be different, depending on the individual track capacity.

Error Messages

Error messages are written to SYSLST and/or SYSLOG.

Clear Disk (CLRDK)

The purpose of this utility is:

- To clear one or more extents on a CKD (Count Key Data) or an ECKD (Extended Count Key Data) disk and/or create a file label in the VTOC (volume table of contents).

- To preformat the tracks of the cleared extents.

The area to be cleared can be as small as one track or as large as a complete disk. Any number of areas can be cleared with one run. When an area is cleared, it is formatted as fixed-length blocks containing count, key and data areas. The information needed by the utility to format the key and data areas must be supplied by you. If you omit it, the utility uses the default values.

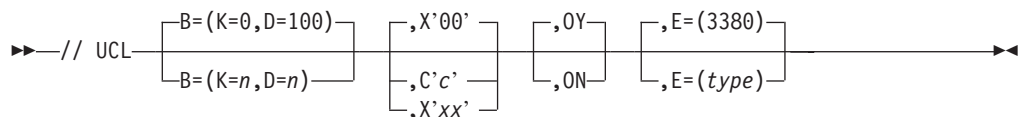
The key and data areas, except the first eight bytes of the data area of R0 (track descriptor record), are filled with a character which you must define.

Job Control Statements

The DLBL statement for UOUT in this utility must indicate a sequential file either explicitly or by default. Label information is checked for an unexpired file or parts of it. Expired labels are deleted from the VTOC. For the preformatted tracks, a file label is written in the VTOC of the disk.

Utility Control Statement

The format of the (optional) // UCL statement is:



B=(K=n|0,D=n|100)

Key length (K - default is 0) and data block
(D - default is 100) in bytes(n).

C'c'|X'xx'

For c enclosed in quotes, specify your fill character in character notation.
For xx enclosed in quotes, specify your fill character in hexadecimal notation. The default is X'00'.

ONIOY

Output option operand:

Y indicates write-disk check (default).
N no write-disk check.

E=(type|3380)

Output device type. Valid specifications for type entries are 3375, 3380, and ECKD.

END Statement

Supply to signal the end of the utility control statements. Its format is:

```
// END
```

Using the Clear Disk Utility

The following sample job stream shows you how to use the Clear Disk utility.

Example – Clear and Preformat One Extent

```

// JOB jobname
(1) // ASSGN SYS012,191
(2) // DLBL UOUT,'DISK LABEL',9999
(3) // EXTENT SYS012,001221,1,0,00310,00630

```

- ```

(4) // EXEC CLRDK
(5) // UCL B=(K=38,D=480),X'55',ON
(6) // END
(6) /&

```
1. // ASSGN SYS012,191  
Assigns SYS012 to CKD disk on 191.
  2. // DLBL UOUT,'DISK LABEL',9999
    - UOUT is the filename.
    - 'DISK LABEL' is the file-id.
    - Retention period is 9999 days.
  3. // EXTENT SYS012,001221,1,0,00310,00630
    - Extent for CKD output disk to be cleared.
    - 001221 is the volume serial number.
    - Type is 1 (data area - no split cylinder).
    - Sequence number is 0.
    - Clear 630 tracks starting at track 310.
  4. // EXEC CLRDK  
Starts Clear Disk operation.
  5. // UCL B=(K=38,D=480),X'55',ON
    - The key length is 38 and data length 480.
    - The fill character is X'55'.
    - Output option is ON. It indicates that a write-disk check is **not** to be performed.
    - The output disk is an IBM 3380 per default.
  6. Ends utility control input and job.

---

## Copy and Restore Diskette (CRDR)

The purpose of this utility is:

- To copy the entire contents of a diskette onto another diskette with or without an intermediate disk.
- To replace faulty labels on a diskette.

The utility is controlled by the following statements:

- Utility control statement.
- File descriptor statement.
- // END statement.
- /\* statement.

If your control input does not supply sufficient information for a successful copy, the utility cancels after printing track 0 (volume and file information) together with an error indication. Complete track 0 information must be present to start copying.

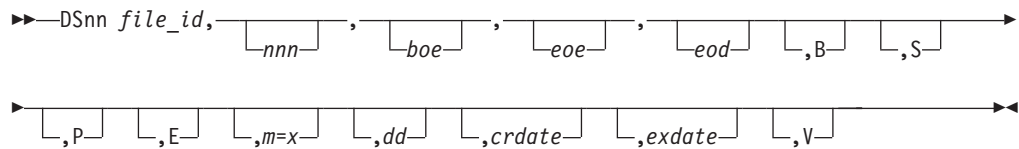
### ***Input Diskette***

The utility reads the VOL1 label of the input diskette. If that label cannot be read, you must supply label information in the utility control statement.

If the input diskette contains faulty file descriptor labels, you can specify replacement information in file descriptor statements.

### ***Output Diskette***





A blank character must follow the last operand.

Put all required and optional information for one file in one card image (position 2 - 72). Continuation is not allowed.

**DSnn** *nn* specifies the file number. This must be the two-digit number that was written by the utility to SYSLST (as part of a message) during a failing previous run.

*file-id* Identifies the logical file. Can be from one to eight alphanumeric characters.

*nnn* Indicates the length of the data records on the file (from 1 to 128 bytes).

*boe* Indicates the Beginning of the Extent (BOE). For *boe* specify the diskette address in the form *cchrr*, where:

cc = cylinder number  
h = head number (always zero)  
rr = record (sector) number

For example:

|                |                 |
|----------------|-----------------|
| cchrr (lowest) | cchrr (highest) |
| 01001          | 73026           |

The range covered by *boe* and *eoe* must not overlap any other file.

*eoe* Indicates the End of the Extent (EOE). Must be a valid five digit diskette address in the form *cchrr* (see also *boe* above).

The range covered by *boe* and *eoe* must not overlap any other file. The EOE must not be lower than the BOE.

*eod* Indicates the End of Data (EOD). This also takes the form of *cchrr* as described above.

The EOD address must be higher than the BOE address and not more than one record higher than the EOE address.

**B** Indicates that a file is to be bypassed.

**S** Indicates that you must reply YES to access the file, if it is an input file.

**P** Indicates that a file may not be overwritten or deleted after expiration.

**E** Indicates fixed, unblocked, sequential records &lesym 128 bytes in length.

**m=x** Indicates whether the specific file is continued on another volume or ends on the currently processed volume:

x=C the file is continued  
x=L the file ends

If you omit the operand, the file starts and ends on this volume.

*dd* Indicates the order of the volume in a multivolume file. It must be a two-digit number. The first or only volume is 01.

The default value is 01.

*crdate* Indicates the creation date of the file. The format is *yymmdd* (where: *yy* = year, *mm* = month, *dd* = day).

The default creation date is the system date.

*exdate* Indicates the expiration date of the file. The format is *yymmdd* (where: *yy* = year, *mm* = month, *dd* = day).

The default expiration date is the system date plus seven days.

**V** Indicates that the data has been subjected to a verification procedure.

**END and /\* Statements:** You must supply these statements to signal the end of the control statement set. Code them as shown:

```
// END
/*
```

## Using the Copy and Restore Diskette Utility

The following sample job streams show you how to use the Copy and Restore Diskette utility.

### Example 1 – Copy an Error-free Diskette to Another Diskette on a Different Diskette Unit

```
// JOB jobname
(1) // ASSGN SYS004,001
(2) // ASSGN SYS005,002
(3) // EXEC CRDR
(4) // UCD TV,VOL=123456
(5) // END
(5) /*
(5) /&
```

1. // ASSGN SYS004,001

Assigns SYS004 to the input diskette.

2. // ASSGN SYS005,002

Assigns SYS005 to the output diskette.

3. // EXEC CRDR

Starts the Copy operation.

4. // UCD TV,VOL=123456

Controls the Copy operation and checks the volume serial number 123456 on SYS004.

5. Ends utility control input and job.

If the input diskette has one or more faulty labels, no copy operation takes place because the control statement stream does not include any descriptor statements and therefore, the faulty labels cannot be replaced. The track-0 table is printed instead (see the example in Figure 14 on page 94) below. This example of a track-0 table is discussed subsequently.

|       |     |    |   |       | (6)   | (7)   | (8) | (9) | (10) | (11) | (12)   | (13)   | (14) | (15)  |
|-------|-----|----|---|-------|-------|-------|-----|-----|------|------|--------|--------|------|-------|
| DS01  | VTL | TC | 1 | 00080 | 01001 | 03004 |     |     | C    | 01   | 730101 | 730101 |      | 03005 |
| 02    | VTL | TC | 3 | 00080 | 08001 | 08026 |     |     | L    | 02   | 730101 | 781224 |      | 08024 |
| ** 03 | VTL | TC | 4 | 00080 | 10004 | 10026 |     |     |      |      |        |        |      |       |
| 04    | VTL | TC | 2 | 00080 | 05001 | 06008 |     |     | C    | 01   | 730101 | 781224 |      | 06008 |
| 05    | VTL | TC | 5 | 00008 | 12001 | 12009 |     |     | C    | 01   | 730101 | 781224 |      | 12009 |
| 06    | VTL | TC | 6 | 00080 | 14001 | 15003 |     |     |      | 01   | 730101 | 731224 |      | 15004 |
| ** 07 |     |    |   |       |       |       |     |     |      |      |        |        |      |       |
| 08    |     |    |   |       |       |       |     |     |      |      |        |        |      |       |
| 09    |     |    |   |       |       |       |     |     |      |      |        |        |      |       |
| 10    |     |    |   |       |       |       |     |     |      |      |        |        |      |       |
| 11    |     |    |   |       |       |       |     |     |      |      |        |        |      |       |
| 12    |     |    |   |       |       |       |     |     |      |      |        |        |      |       |
| 13    |     |    |   |       |       |       |     |     |      |      |        |        |      |       |
| 14    |     |    |   |       |       |       |     |     |      |      |        |        |      |       |
| 15    |     |    |   |       |       |       |     |     |      |      |        |        |      |       |
| 16    |     |    |   |       |       |       |     |     |      |      |        |        |      |       |
| 17    |     |    |   |       |       |       |     |     |      |      |        |        |      |       |
| 18    |     |    |   |       |       |       |     |     |      |      |        |        |      |       |
| 19    |     |    |   |       |       |       |     |     |      |      |        |        |      |       |

|                           |                             |                           |
|---------------------------|-----------------------------|---------------------------|
| (1) DSN                   | (6) BYPASS INDICATOR        | (11) VOLUME SEQUE #       |
| (2) DATA SET IDENTIFIER   | (7) DATA SET SECURITY       | (12) CREATION DATE YYMMDD |
| (3) BLOCK LENGTH          | (8) WRITE PROTECTION        | (13) EXPIR. DATE YYMMDD   |
| (4) BEGIN OF EXTENT CCHRR | (9) EXCHANGE INDICATOR      | (14) VERIFY INDICATOR     |
| (5) END OF EXTENT CCHRR   | (10) MULTI VOLUME INDICATOR | (15) END OF DATA CCHRR    |

Figure 14. Track 0 Table

Files 3 and 7 are marked defective by means of \*\*. You need a second run of the CRDR utility with a job stream containing file descriptor statements for files 3 and 7.

File 3: A read check occurred after the end-of-extent field was read. The lost information must be recreated. The information for four of the five required operands is still present in the track 0 table. Code the statement as follows:

```
DS03 VTLTC4,80,10004,10026
```

Eod is the only required operand to be added. If this diskette is *not* the output of a previous CRDR run, eod points to the record beyond eoe, which is the first record on the next track.

```
DS03 VTLTC4,80,10004,10026,11001
```

If the input diskette is the output of a previous CRDR run, it is possible that eod is an address lower than 11001 and, therefore, the new copy will contain too many records in file 3. Delete the superfluous records after the Copy and Restore job.

File 7: No information at all has been passed on to the track 0 table. Therefore, the read check occurred in the first bytes of the label. You may now do either of the following:

- Delete the file by coding DS07 only, so that file 7 will not be copied to the output diskette (see explanation number 8 in "Example 2 – Using an Intermediate Disk" on page 95) or
- Re-create file 7.

If you want to re-create file 7, you must determine the extent limits and the record length. In this case you can use the CRDR utility to make all records between file 6 and the end of the diskette accessible by coding:

DS07 ANYNAME,128,15004,73026,74001

> end of data  
 > end of extent  
 > beginning of extent  
 > block length

With the extent and record values known to you, you can code a 'DS07' statement similar to number (7) below in order to copy this file to the diskette or intermediate disk, as the case may be.

## Example 2 – Using an Intermediate Disk

If you want to use an intermediate disk in the second run, and X'190' has been added to the system as an FBA disk drive, the following may be used:

```

 // JOB jobname
(1) // ASSGN SYS004,001
(2) // ASSGN SYS005,190
(3) // DLBL UTEMP,,98/240,SD,,CISIZE=2048
(4) // EXTENT SYS005,222222,1,1,200,520
(5) // EXEC CRDR
(6) // UCD TV
(7) DS03 VTLTC4,80,10004,10026,11001,,,,,,,,981231
(8) DS07
(9) // END
(9) /*
(9) /&

```

### 1. // ASSGN SYS004,001

Assigns SYS004 to input and output diskette.

### 2. // ASSGN SYS005,190

Assigns SYS005 to intermediate FBA disk.

### 3. // DLBL UTEMP,,98/240,SD,,CISIZE=2048

- UTEMP is the filename.
- 98/240 is year and day for the expiration date.
- SD is sequential disk.
- CISIZE is control interval size of 2048 bytes.

### 4. // EXTENT SYS005,222222,1,1,200,520

- FBA disk on SYS005 has volume serial number 222222.
- Type is 1 (data area).
- First extent of an indexed sequential file.
- Extent starts at relative block 200.
- Block is 520 bytes long (see Note below).

### 5. // EXEC CRDR

Starts Copy operation.

### 6. // UCD TV

Controls Copy operation.

### 7. DS03 VTLTC4,80,10004,10026,11001,,,,,,,,981231

Needed for defective file number 3.

- *file-id* is VTLTC4.
- 80-byte records extending from cylinder 10 record number 4 to cylinder 10 record number 26.
- End of data at cylinder 11 record number 1.
- 981231 is year-month-day.

### 8. DS07

Prevents file number 7 from being copied to the output diskette.

9. Ends utility control input and job.

**Note:** The extent size depends on the type of the disk device used. A full diskette requires disk storage, for example, as follows:

| Device   | No. of tracks | No. of blocks<br>(with CFSIZE of 2K in DLBL) |
|----------|---------------|----------------------------------------------|
| IBM 3380 | 15            |                                              |
| IBM FBA  |               | 520                                          |

---

## Initialize Tape (INTTP)

The purpose of this utility is to write volume labels on magnetic tape or data cartridge for standard label checking:

- One to eight IBM standard tape volume labels in numerical sequence, followed by one dummy header label and one tape mark on EBCDIC tapes.
- One American National Standard label, followed by one dummy header label and one tape mark on ASCII tapes.

ASCII = American Standard Code for Information Interchange.

You have two options to create standard labels on tape:

- Use a single control statement to provide the initial volume serial number, the owner identification, and the protection code for the first and all succeeding tapes. The volume serial number is incremented by one for each new tape.
- Write unique volume labels on each tape by using a volume label image statement. Supply up to eight volume label image statements for each EBCDIC tape but only one volume label image statement for each ASCII tape.

You must use this option when the volume serial number field contains alphameric values.

The utility writes a dummy header label, immediately followed by a tape mark.

As each tape is initialized, its volume label(s) and tape unit number are written to SYSLOG to indicate the completion of the initialization. You can then rewind or rewind and unload the tape.

To initialize two or more tapes, assign the tape units in ascending order, starting with SYS000. After you have assigned the last unit, unassign the next sequential logical unit (or assign to a unit other than a tape drive) to ensure proper termination. Thus you can avoid unintentional initialization (see “Example 3 – Initialize two ASCII Tapes with the Volume Label Image Option” on page 100).

The utility initializes one tape after another starting with SYS000.

If there is only SYS000 and no additional units, you can take one of the following two steps:

- If you specified the rewind option, the job is terminated after initializing one tape.
- If you did not specify the rewind option, the utility waits until you mount a new tape on the first output unit (SYS000). Then tape initialization begins a new cycle of the output units.

If the utility reads the last statement before a cycle of assigned units has been read, the utility completes the tape on the current unit and terminates the job.

## Job Control Statements

Use the following job control statements to assign a system:

**// JOB** Must be assigned.

**// ASSGN SYSIPT**  
Must be assigned.

**// ASSGN SYSLOG**  
Must be assigned for output messages.

**// ASSGN SYS000**  
Must be assigned as the first output tape unit.

**// ASSGN SYS<sub>nnn</sub>**  
Optional. Up to 15 ASSGN statements for additional output devices may be given. The programmer logical unit names must be in ascending order and contiguous, starting with SYS001. The lowest unused logical unit should be assigned UA.

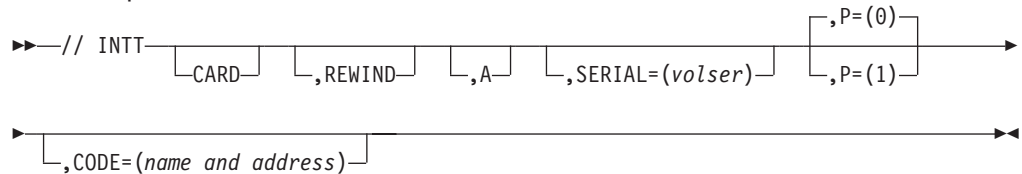
**// EXEC INTTP**  
Required.

## Utility Control Statement

The statement you must supply contains the control information required by the utility. Its format is shown below. A discussion of the operands follows.

Specify the operands in the order stated. If an optional operand is not used, do not code a comma for that operand.

For example:



```
// INTT REWIND,SERIAL=(123456),CODE=(J DOE ABC CHEM)
```

**CARD** Indicates the volume label image option operand.

If specified, volume label image statements (only one for ASCII tapes) must be supplied (see Volume Label Image Statement and “Example 3 – Initialize two ASCII Tapes with the Volume Label Image Option” on page 100).

If omitted, the utility assumes that volume information for every initialized tape must be taken from an INTT control statement.

**REWIND**

Indicates that each tape must be rewound at completion of initialization.

If omitted, the utility assumes that the tapes must be rewound and unloaded at completion of initialization.

**A**

Indicates that all tape labels must be recorded in the ASCII mode and will observe the ANSI standards (ANSI = American National Standards Institute).

If omitted, the utility assumes EBCDIC mode and IBM standards.

**SERIAL=(volser)**

Required if you omitted the CARD operand. For *volser*, specify a six-digit

number that represents the volume serial number. This number is incremented by 1 for each additional tape initialized after the first.

**P=(1)|(0)**

Applies only if you omitted the CARD operand. Identifies the protection option operand.

For EBCDIC tapes, this operand indicates that the volume security fields of all volumes to be initialized must be given the specified protection value:

- 1 = Protection (no accessibility).
- 0 = No protection (unlimited accessibility - default).

For ASCII tapes, this operand indicates that the accessibility fields of all volumes must be given the specified protection value. If you omit it, a space is written.

**CODE=(name-and-address)**

Required if you omit the CARD operand. For EBCDIC tapes, you must supply a 10-character value, and for ASCII tapes, a 14-character alphanumeric value (0-9, A-Z, \$, #, @, blank) to state the owner's name-and-address code.

**Volume Label Image Statement**

If you specified the volume label image option (the CARD operand) you need additional statements:

- For EBCDIC tapes from one to eight volume label image statements in ascending order (VOL1, VOL2, etc.).  
For ASCII tapes one volume label image statement. This statement must contain VOL1 in the first four positions.
- // END statement. This statement is used to separate the volume label image statements for the different tapes. If eight volume label image statements are supplied for a given EBCDIC tape, you may omit the END statement.

The volume label image statement is described below:

**Positions**

**Contents**

- |                       |                                                                                                                                                                                                                                               |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>1-3</b>            | Label identifier. Must contain VOL to indicate that it is a volume label.                                                                                                                                                                     |
| <b>4</b>              | Volume label number. Indicates the relative position (1 - 8) of a volume label within a group of volume labels.<br><br><i>The following fields apply to VOL1 labels only.</i>                                                                 |
| <b>5-10</b>           | Volume serial number. An identification code assigned to a volume when it enters an installation. Normally a numeric value from 000001 through 999999, although any or all of the six bytes may be alphabetic. Must always be six characters. |
| <b>11</b>             | Volume security. Indicates the security status of the volume. Not used by VSE.                                                                                                                                                                |
| <b>12-37 or 12-41</b> | Reserved for ASCII or EBCDIC, respectively.                                                                                                                                                                                                   |
| <b>38-51 or 42-51</b> | Owner's name-and-address code in ASCII or EBCDIC, respectively.<br>Indicates a specific customer, installation, and/or system.                                                                                                                |
| <b>52-79</b>          | Reserved.                                                                                                                                                                                                                                     |



- Volume serial number is 000001.
  - Protected against unlimited access.
  - Owner's name-and-address code in position 38.
5. Ends utility control input and job.

### Example 3 – Initialize two ASCII Tapes with the Volume Label Image Option

```
// JOB jobname
(1) // ASSGN SYS000,181
(2) // ASSGN SYS001,182
(3) // ASSGN SYS002,UA
(4) // EXEC INTTP
(5) // INTT CARD,A
(6) VOL1000001 AB COMPANY NYC
(7) // END
(8) VOL1000002 AB COMPANY NYC
(9) // END
(10) /*
(10) /&
```

1. // ASSGN SYS000,181  
Assign SYS000 to first tape on 181.
2. // ASSGN SYS001,182  
Assigns SYS001 to second tape on 182.
3. // ASSGN SYS002,UA  
Unassigns next sequential tape unit.
4. // EXEC INTTP  
Starts the Initialize operation.
5. // INTT CARD,A  
States that volume label image statements follow.
6. VOL1000001 AB COMPANY NYC  
Volume label image statement with
  - Vol=label identifier.
  - 1=volume label number.
  - 000001=volume serial number.
  - Owner's name-and-address code in position 38.
7. // END  
Separates volume image label statements.
8. VOL1000002 AB COMPANY NYC  
See 6.
9. // END  
See 7.
10. Ends utility control input and job.

---

## Chapter 9. Printing or Displaying System Information

This chapter describes the following utility programs for printing or displaying system information:

- **LSERV** (label information area)
- **LVTOC** (volume table of contents)
- **EREP** (recorder file)
- **PRINTLOG** (hardcopy file)
- **LISTLOG** (job information).

---

### Displaying the Label Information Area (LSERV)

The “operator action” for a number of system messages includes listing the label information area. This listing helps programmers to determine and correct the error that caused the message.

#### Job Control Statement

Use the LSERV program to print the contents of the label information area on the device assigned to SYSLST. The job control statement for LSERV is as follows:

```
▶▶ // EXEC LSERV [,PARM='STDLABEL',
 [PARSTD
 [PARSTD=syslogid
 [CLASSTD
 [CLASSTD=class]]]] ▶▶
```

The following is a description of the parameters used in the job control statement:

#### **STDLABEL**

prints the system standard labels only.

#### **PARSTD**

prints all partition standard labels only.

#### **PARSTD=*syslogid***

prints the partition standard labels of the specified **static** partition only.

#### **CLASSTD**

prints all class standard labels only.

#### **CLASSTD=*class***

prints the class standard labels of the specified class only.

If no parameter is specified, all labels are printed. User labels from static or dynamic partitions can only be printed when no parameter is specified. User labels change from job to job and, therefore, no special support is needed in the LSERV program.

A sample partial output of an LSERV run for the above control statements is shown in Figure 15 on page 102. The output indicates the files whose labels have been stored in the label information areas for the various partitions, whether these labels have been stored permanently or temporarily, and (for the labels on a CKD/ECKD volume) the label information area track.

The output shows the relationship between job control DLBL and EXTENT statements (for further information see the *VSE/ESA Guide to System Functions*).

#### For VSE/VSAM Files Only:

There is an additional label information record following the VSAM label record if, in the DLBL statement, at least one of the operands DISP, RECORDS, or RECSIZE is specified.

**Note:** A warning message is issued on SYSLST if you request LSERV while another partition is updating the label area. The free usage subarea which is internally used by VSE/ICCF is not shown by the LSERV program.

#### Example of an LSERV Output

```

 LABEL INFORMATION DISPLAY PAGE nnn
EXAMPLE
 FILE IDENTIFIER EXAMPLE
 FILE SERIAL NUMBER OMITTED
 VOLUME SEQUENCE NUMBER 01
 CREATION DATE OMITTED
 RETENTION PERIOD (DAYS) 0999
 FILE TYPE SEQUENTIAL

 EXTENT INFORMATION
 EXTENT SEQUENCE NUMBER 000
 EXTENT TYPE 1 (PRIME DATA)
 RELATIVE START ADDRESS 002
 NUMBER OF TRACKS/BLOCKS 045107
 SYMBOLIC UNIT SYSRES LOGICAL UNIT FORMAT
 TYP#00,NUM#06
 VOLUME SERIAL NUMBER OMITTED

 ADDITIONAL INFORMATION
 DISPOSITION (OLD,KEEP)
 RECORDS (00000000500,0000000100)
 RECORD SIZE 0000000080

SALARY
 FILE IDENTIFIER SALARY.1999.FILE
 FILE SERIAL NUMBER DASD02
 VOLUME SEQUENCE NUMBER 01
 CREATION DATE OMITTED
 EXPIRATION DATE 99/365
 FILE TYPE SEQUENTIAL

 EXTENT INFORMATION
 EXTENT SEQUENCE NUMBER 000
 EXTENT TYPE 1 (PRIME DATA)
 RELATIVE START ADDRESS IN TRACKS/BLOCKS 010000
 NUMBER OF TRACKS/BLOCKS 001000
 SYMBOLIC UNIT SYS019 LOGICAL UNIT FORMAT
 TYP#01,NUM#13
 VOLUME SERIAL NUMBER DASD02

```

Figure 15. Sample LSERV Output

---

## Displaying the Volume Table of Contents (VTOC)

The operator action for a number of system messages includes the listing of the volume table of contents (VTOC) of a certain disk pack. A volume table of contents (VTOC) is an index of all files, and the remaining space, on a disk volume. This listing helps your programmer to determine the error that caused the message. You

can display the VTOC either by using the LVTOC program or by using the *Display VTOC* dialog. Both methods are described below.

**Note:** You can also display the Volume Table of Contents using the Interactive Interface dialog *Display VTOC* or the utility program DITTO/ESA. For more information on DITTO/ESA, see the IBM manual *DITTO/ESA for VSE, User's Guide and Reference*

## Using the LVTOC Program

A VTOC display can be requested by executing the LVTOC program with SYS004 assigned to the applicable disk drive and SYS005 to a printer.

LVTOC lists the file labels contained in a VTOC in alphabetic sequence by file name. It also provides a listing of free space on the volume, with the start and end addresses and sizes of the unused space. The creation and expiration dates of a file that is considered "never expiring" are prefixed by asterisks (\*\*). The control statements needed to invoke that program may be submitted in one of two ways:

1. Using SYSRDR.

The job control statements that are required to display the VTOC from SYSRDR are:

```
// JOB jobname
// ASSGN SYS004,cuu
// ASSGN SYS005,cuu
// EXEC LVTOC
/
```

SYS004 *must* be assigned to the physical address of the disk drive on which the volume in question is mounted. SYS005 *must* be assigned to the printer on which the VTOC is to be displayed.

2. Using the console (SYSLOG).

To display the VTOC from the printer-keyboard or display console, enter the following statements:

```
// ASSGN SYS004,cuu
// ASSGN SYS005,cuu
// EXEC LVTOC
```

A display of a VTOC can be requested also in response to messages as indicated in the *VSE/ESA Messages and Codes* manual. Such a response is CANCELV or DSPLYV. Use CANCELV if you intend to cancel the job, or DSPLYV if the condition allows program execution to be continued after the VTOC display.

Frequently, during resource definition and system maintenance, you need information about the status and contents of your disk volumes. This information is stored for each **disk volume** in its Volume Table of Contents (VTOC).

## Using the Display VTOC Dialog

The *Display VTOC* dialog interactively displays information stored in a VTOC. To access the dialog, start with the initial *Function Selection* panel and select:

- 2 (Resource Definition)
- 3 (Display VTOC)

**Note:** The above path selection is for an Administrator.

|                                       |                                 |        |
|---------------------------------------|---------------------------------|--------|
| Administrator<br>Fast Path: <b>23</b> | Synonym<br>Default: <b>vtoc</b> | Yours: |
|---------------------------------------|---------------------------------|--------|

The *Display VTOC* panel displays a list of volumes on your system. The following information for each volume is shown on the panel:

#### **VOLUME**

This is the VOLID. If asterisks (\*\*\*\*\*) are displayed, either the device is not available to the system or it is a diskette. You cannot display VTOC information for diskettes.

#### **ADDRESS**

This is the device address.

#### **DEVICE TYPE**

This is the type of device.

#### **SHARED**

This shows whether the disk is accessible by more than one VSE system.

#### **RESERVED**

This indicates whether the device is currently unavailable.

You may select one of the three options shown on the *Display VTOC* panel. Type the number of one of the following options next to the device for which you want a VTOC listing.

#### **Option Result**

##### **1 = FREE SPACE**

This option lists all free (unused) extents on the device, in ascending order beginning with the lowest address.

##### **2 = FILES LIST**

This option lists all files (file IDs), in alphabetical order, allocated on the device.

##### **3 = VOLUME LAYOUT**

This option lists all files (file IDs) and free extents, in ascending order beginning with the lowest address.

---

## **Printing and Handling the Recorder File**

The system recorder file is used to save information about machine or device related hardware error or device related statistical information. With the **Environmental Recording, Editing and Printing (EREP)** program you can:

- Save the recorder file contents,
- Reinitialize the recorder file,
- Build and update recorder file history tapes, and
- Produce machine and device-related error reports.

Saving and clearing of the recorder file can be done at any time, but it *must* be done in cases where you receive a recorder file full condition. If you have this condition, you must save the information as soon as possible; otherwise, further reports occurring after the full condition was detected will not be saved.

EREP is invoked by issuing the following job stream at the system console:

```
// TLBL HISTOT
// ASSGN SYS009,cuu
// EXEC IFCOFFLD,SIZE=AUTO
```

The TLBL and ASSGN statement define the tape on which the recorder file is saved. *cuu* is the tape unit. The job stream does the following:

- It creates a system summary report of the data stored in the recorder file. The system summary report is printed on SYSLST.
- It writes the contents of the recorder file to the tape defined.
- It reinitializes the recorder file.

The EREP program produces several different types of reports:

- All records from the recorder file,
- Statistics of device failures tailored to each device type,
- Statistics of device failures of all devices attached to your system,
- Statistics of machine failures, and
- Statistics of channel failures.

EREP will accept as input:

- The system recorder file,
- A history tape containing the contents of the previously saved recorder file, or
- The combination of the system recorder file and a history tape.

How to request a particular EREP report is described in the IBM manual *EREP User's Guide*

To save device statistical information within the recorder file, especially before powering down your system, use **Record On Demand (ROD)**. This Job Control Command (JCC) records specific hardware information that might otherwise be lost. If you do *not* have a recorder file full condition, you may use the ROD command to store this information in the recorder file. Enter at the system console:

```
ROD
```

To print the current recorder file, run the following job:

```
// JOB jobname
* PRINT RECORDER FILE AND CLEAR IT
// EXEC IFCEREP1,SIZE=200K
PRINT=PT
ACC=N
ZERO=Y
TABSIZ=50K
/*
/&
```

---

## Printing the Hardcopy File (PRINTLOG)

Each line that appears on the screen of the display console is written to the hardcopy file, which resides on SYSREC. It may be necessary to print the hardcopy file or parts of it if before it becomes full. You should print its contents before it is overwritten (see the topic “Hardcopy File Full Condition” in the IBM manual *VSE/ESA Guide for Solving Problems*). Also, there may be a need to check what happened during the day or which messages were issued for a certain partition. The IBM utility program (PRINTLOG) prints the hardcopy file from disk onto SYSLST.

To print the hardcopy file, proceed as follows:

1. Type // EXEC PRINTLOG and press ENTER. The following message appears on the screen:  

```

ENTER OPTIONS FOR PRINTLOG OR ? FOR A LIST OF OPTIONS
VALID OPTIONS: (CHOOSE MAX ONE OF EACH LINE)
ALL OR NEW TOTAL FILE OR ONLY NEW RECORDS
CMD, IPL, AR, BG, F1... COMMANDS, IPL-MSGs OR PARTITION ID
A, S OR N ACTION, SUPPRESSED OR 'NETVIEW' RECORDS
JOBNAME=NAME ONLY RECORDS OF THAT JOB
MM/DD/YYYY ONLY RECORDS OF THAT DATE
ENTER OPTIONS FOR PRINTLOG OR ? FOR A LIST OF OPTIONS

```
2. Select the desired options and enter them. The options must be separated by commas; intervening blanks are not permitted.  
You may enter a null line (just press ENTER). The system then uses the option **ALL** by default.

| <b><i>Operation Mode Operand:</i></b> | <b><i>Description:</i></b>                                                                                                                                                                                                                                                                                                                                                                          |
|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ALL</b>                            | Specifies printing of all messages that are on the hardcopy file. (See Note 1.)                                                                                                                                                                                                                                                                                                                     |
| <b>NEW</b>                            | Specifies printing of only those messages that have accumulated since PRINTLOG was last run with only the NEW or ALL option specified, or since the last IPL with SET HC=CREATE. (See Note 1).<br><br>If ALL or NEW is combined with one or more of the following options, its function applies only to the messages printed by these options. If neither ALL nor NEW is specified, ALL is assumed. |
| <b><i>Filter Operand:</i></b>         | <b><i>Description:</i></b>                                                                                                                                                                                                                                                                                                                                                                          |
| <b>CMD</b>                            | Specifies printing of all entered commands (Attention routine commands, VSE/POWER, VM, or CP) together with the command responses. If, for example, R RDR was entered, not only the command itself, but also the related responses are printed.                                                                                                                                                     |
| <b>IPL</b>                            | Specifies printing of all commands entered during IPL and their command responses. Only items up to the first message with prefix "BG 0000" are displayed.                                                                                                                                                                                                                                          |
| <b>AR</b>                             | Specifies printing of messages issued by the attention routine.                                                                                                                                                                                                                                                                                                                                     |
| <b>BG - Fn</b>                        | Specifies printing of messages issued by a particular static partition. Only <i>one</i> partition identifier or AR may be specified. n can be from 1 up to the number of foreground partitions supported in your system.                                                                                                                                                                            |
| <b>&lt;class&gt;&lt;pno&gt;</b>       | Specifies printing of messages issued by a particular dynamic partition. <b>class</b> specifies a partition class and <b>pno</b> specifies a particular partition number. If * is specified, all messages of the selected dynamic partition class are displayed (for example, all partitions within that specific class).                                                                           |
| <b><i>Subfilter Operand:</i></b>      | <b><i>Description:</i></b>                                                                                                                                                                                                                                                                                                                                                                          |
| <b>A</b>                              | Specifies printing of action messages which requested a reply or which have one of the descriptor codes 1, 2, 3, or 11. (See Note 2.)                                                                                                                                                                                                                                                               |

|                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>N</b>                                                                                                                                                                                          | Specifies printing of all logging items directed to or entered on an automation console, such as a NetView operator station.                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>S</b>                                                                                                                                                                                          | Specifies printing of all messages that were suppressed or replied by an automation product, like VSE/OCCF.                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>JOBNAME=</b>                                                                                                                                                                                   | Specifies printing of messages pertaining to the job identified by NAME.                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>MM/DD/YYYY</b>                                                                                                                                                                                 | Specifies printing of messages issued on a particular day. Enter the date in the format month, day, year (if this format was specified in the STDOPT command). For day and month, leading zeros may be omitted. The year can be specified as a 1, 2 or 4-digit number. In the case of 4 digits, the year is accepted as is. In case of 2 digits, a window of 20,80 years is in effect. For example: 01/01/15 results in 01/01/2015, and 01/01/16 results in 01/01/1916. In case of 1 digit, 01/01/1 results in 01/01/2001, for example. |
| <b>Note:</b> Please note that you can specify one operation mode operand plus one filter and or one subfilter operand. If you specify more than one of each, the additional operands are ignored. |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

#### Notes:

1. If PRINTLOG with the option ALL or NEW is canceled, the following PRINTLOG with the option NEW starts at the position where the PRINTLOG ALL/NEW was interrupted.
2. Important reply messages, which are flagged with a plus or minus sign after the partition identifier, are always printed if you specify A.

#### Examples

Print all action messages: ALL, A

Print all action messages for F1: ALL,F1,A

Print all new messages issued by the Attention routine: NEW,AR

Print all messages issued on April 28, 1998: 4/28/1998

**Note:** Enter the date according to the format specified in the STDOPT command. If you enter an option that does not exist, or if you do not adhere to the prescribed format, the following message appears on the screen:

```
THE FOLLOWING OPTIONS ARE INCORRECT: xxxxxxxx -
PLEASE REENTER
```

The incorrect option(s) are displayed (xxxxxxx) and you need correct only the option(s) in error.

All options can also be entered directly in the EXEC PRINTLOG statement as PARM values. For example:

```
// EXEC PRINTLOG,PARM='ALL,F1,A'
```

---

## Printing Job Information (LISTLOG)

To gather information about how a particular job has run on the system, use the utility program LISTLOG. This will provide a listing on SYSLST of the following items:

- All job control statements which are written to the console.
- All console messages for the job.
- All operator responses for the job.
- Any attention routine messages and commands issued while the job was running.

The LISTLOG utility program can be invoked in two ways:

- Explicitly, by including the job control statement  
    `// EXEC LISTLOG`  
in the job stream. The items listed above will be printed for this job.
- Automatically, when a job is canceled by the operator or by program request. In this case, job control generates a `// EXEC LISTLOG` statement which will not, however, appear on the console or be written to the hardcopy file.

LISTLOG derives the information to be printed from the hardcopy file. It prints all messages and commands relevant to the partition in which the job ran.

Printing stops when the current hardcopy file address is reached (the EXEC LISTLOG statement) and not necessarily at the end-of-job for the currently active job.

If a second job cancels due to program request while LISTLOG is running, it will not be invoked again.

---

## Chapter 10. Batch Access to the VSE/ESA Host Transfer File

---

### Overview

The VSE/ESA Host Transfer File (HTF) is a repository for workstation users to exchange data between their workstation and the VSE/ESA host system.

VSE/ESA includes program INWMUTIL which can be run in a VSE/ESA batch partition and provides the following maintenance functions for the HTF:

|               |                                                          |
|---------------|----------------------------------------------------------|
| <b>LOAD</b>   | To load a VSE/VSAM ESDS or KSDS file into the HTF.       |
| <b>UNLOAD</b> | To unload a file from the HTF into a VSE/VSAM ESDS file. |
| <b>DELETE</b> | To delete a file from the HTF.                           |
| <b>DIRECT</b> | To print a directory of the files stored in the HTF.     |

The main purpose of INWMUTIL is to exchange files between VSE/VSAM and the HTF without going through the *Personal Computer Move Utilities* dialog. In addition, INWMUTIL allows the VSE/ESA administrator to check who owns the files stored in the HTF.

The VSE/ESA Host Transfer File (HTF) is a repository which workstation users can use to exchange data between a workstation and the VSE/ESA host system. The file transfer function is provided in conjunction with a 3270 emulator program running on the workstation. (for more information see "VSE/ESA Programming and Workstation Guide").

A workstation user can "SEND" a file to the HTF, or "RECEIVE" a file from the HTF. Each file in the HTF belongs to a user's directory, associated with a particular VSE/ESA user ID. A file in the user's directory is identified by its file name and by its optional file type (8 characters each).

**Note:** Do not access the HTF simultaneously from the CICS Transaction Server partition and a batch partition with the same user ID.

---

### Command Description

Each function is invoked via a control card from SYSIPT. All specified parameters have to be separated by commas without imbedded blanks.

#### LOAD Command

The LOAD command copies a VSE/ESA ESDS or VSE/ESA KSDS file into a user directory in the HTF. If a file of the same name already exists in the HTF, it is replaced.

Parameters supported:

|                  |                                                                                                                                                                                                                                                                      |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>FILENAME=</b> | (required) 1 to 8 characters for the file name of the file loaded into the HTF. The file name (first 7 characters) will be also used as file name in the DLBL statement given for the VSE/VSAM input file. A DLBL statement has to be provided using this file name. |
| <b>FILETYPE=</b> | (optional) 1 to 8 characters for the file type of the file loaded into the HTF. If not specified, a blank file type is assumed.                                                                                                                                      |
| <b>USERID=</b>   | (required) the file is stored into this user's directory in the HTF.                                                                                                                                                                                                 |

## Batch Access to HTF

**PUBLIC** (optional) the file is flagged as "public" in the HTF. "Public" means that all VSE/ESA users can access this file in the HTF. The default is "private".

INWMUTIL displays a message on the console showing the number of records loaded and the date and time when the file was loaded.

Return Code

**0** Load successful  
**16** Severe error (see message on console)

## UNLOAD Command

The UNLOAD command copies a file from the HTF into a VSAM ESDS file. The VSAM ESDS file has to be defined as work file (REUSE) and must exist prior to the UNLOAD.

Parameters supported:

**FILENAME=** (required) 1 to 8 characters for the file name unloaded from the HTF. The file name (first 7 characters) will be also used as file name in the DLBL statement given for the VSAM ESDS output file. A DLBL statement has to be provided using this file name.  
**FILETYPE=** (optional) 1 to 8 characters for the file type of the file unloaded from the HTF. If not specified, a blank file type is assumed.  
**USERID=** (required) the file is searched in this user's directory in the HTF.

INWMUTIL displays a message on the console showing the number of records unloaded.

Return Code

**0** Unload successful  
**8** File does not exist in HTF  
**16** Severe error (see message on console)

## DELETE Command

The DELETE command deletes a file from the HTF.

Parameters supported:

**FILENAME=** (required) 1 to 8 characters for the file name to be deleted from the HTF.  
**FILETYPE=** (optional) 1 to 8 characters for the file type of the file to be deleted from the HTF.  
**USERID=** (required) the file is searched in this user's directory in the HTF.

Return Code

**0** Delete successful  
**4** File does not exist in HTF  
**16** Severe error, see message on console.

## DIRECT Command

Display the contents of a user's directory, or list all user directories.

Parameters supported:

**USERID=** (optional) the directory of this user printed. If not specified, all user directories found in the HTF are printed.

**SORT** (optional) the directory is printed in sorted order. If this parameter is not specified the directory is printed in directory entry order (physical sequence of entries in directory).

Return Code

**0** Directory print successful

**16** Severe error (see message on console)

---

## Sample Scenario

### Exchanging Data between the HTF and a VSE/VSAM ESDS File

The following is a sample usage of the INWMUTIL program. Before running the INWMUTIL program you have to do prepare:

1. VSAM ESDS file to be loaded and/or unloaded from VSE/ESA Host Transfer File.
2. a job to execute INWMUTIL with the required commands.

INWMUTIL supports VSAM ESDS files. The UNLOAD function requires that the file is defined with the REUSE attribute. Also, the VSE/VSAM file used as target file with the UNLOAD function must be defined with the maximum record size used.

## Batch Access to HTF

### Define a VSE/VSAM ESDS File

```
* $$ JOB JNM=F$WACK,CLASS=0,DISP=D,NTFY=YES
// JOB WACK DEFINE FILE
* DEFINE VSAM ESDS FILE FOR HTF LOAD/UNLOAD
// EXEC IDCAMS,SIZE=AUTO
 DEFINE CLUSTER (-
 NAME (HTFDATA2.ESDS) -
 CYLINDERS(2 2) -
 SHAREOPTIONS (1) -
 RECORDSIZE (80 120) -
 VOLUMES (DOSRES) -
 REUSE -
 NONINDEXED -
 TO (99366)) -
 DATA (NAME (HTFDATA2.ESDS.@D@) -
 CONTROLINTERVALSIZE (4096)) -
 CATALOG (VSESP.USER.CATALOG)
 IF LASTCC NE 0 THEN CANCEL JOB
/*
// OPTION STDLABEL=ADD
// DLBL HTFDAT2,'HTFDATA2.ESDS',,VSAM, X
 CAT=VSESPUC
/*
// EXEC IESVCLUP,SIZE=AUTO
A HTFDATA2.ESDS HTFDAT2 VSESPUC STDLABUP
/*
/&
* $$ E0J
```

Figure 16. Job to Define a VSE/VSAM ESDS File

Next, prepare a job to run INWMUTIL. For each LOAD or UNLOAD command a corresponding DLBL statement has to be provided for the file name given.

**Note:** The DLBL file name allows only 7 characters, which must match the first 7 characters of the file name from the HTF.

### Load a VSE/VSAM ESDS File into the HTF

```
* $$ JOB JNM=INWBAT01,CLASS=0,DISP=D
// JOB INWBAT01
// DLBL HTFDAT2,'HTFDATA2.ESDS',0,VSAM,CAT=VSESPUC
// EXEC INWMUTIL,SIZE=AUTO
LOAD,FILENAME=HTFDAT2,FILETYPE=DATA,USERID=BORI
/*
/&
* $$ E0J
```

Figure 17. Job to Load VSE/VSAM File into HTF

The file in the HTF is stored in the directory of user BORI. Only user BORI can download the file (from the HTF to a workstation).

**Note:** If the file already exists in the HTF it is replaced without further warning.

To move the HTF file back into the VSE/VSAM file, use the UNLOAD command as shown in the following job stream.

## Unload a HTF File into a VSE/VSAM ESDS File

```
* $$ JOB JNM=INWBAT02,CLASS=0,DISP=D
// JOB INWBAT02
// DLBL HTFDAT2,'HTFDATA2.ESDS',0,VSAM,CAT=VSESPUC
// EXEC INWMUTIL,SIZE=AUTO
UNLOAD,FILENAME=HTFDAT2,FILETYPE=DATA,USERID=BORI
/*
/&
* $$ E0J
```

Figure 18. Job to Unload a HTF File into a VSE/VSAM ESDS File

## List HTF Directories

To check for existing files in the HTF use the following job stream:

```
* $$ JOB JNM=INWBAT00,CLASS=0,DISP=D
// JOB INWBAT00
* DISPLAY ALL HTF DIRECTORIES
// EXEC INWMUTIL,SIZE=AUTO
DIRECT, SORT
/*
/&
* $$ E0J
```

Figure 19. Job to List HTF User Directories

It is recommended to specify "SIZE=AUTO" for the "EXEC INWMUTIL" to satisfy the program's requirement for partition GETVIS for the SORT function (INWMUTIL requires 220 KB of partition GETVIS).

If you find files belonging to users already deleted from the VSE/ESA user list, you can use the DELETE command to remove those files in the HTF.

```
INWMUTIL PC/HOST TRANSFER FILE DIRECTORY 09/12/98 10.23.54 PAGE 1
 AF=ACCESS FLAG 1=PRIVATE, 2=PUBLIC; X=SHARED
 DT=DATA TYPE BLANK=TEXT, 1=BINARY, 2=BIN/CRLF
 DIRECTORY FOR USERID BORI
```

| FILENAME | FILETYPE | MAX<br>RECLEN | A<br>F | D<br>T | LOAD<br>DATE   | JUL<br>DAY | LOAD<br>TIME | RECORD<br>COUNT |
|----------|----------|---------------|--------|--------|----------------|------------|--------------|-----------------|
| HTFDAT2  | DATA     | 80            | 1      |        | 01/09/1998/009 |            | 9.51.54      | 64              |
| KSDS001  | VSAM     | 31            | 1      |        | 09/06/1998/340 |            | 14.42.19     | 20              |
| TEST1    | KSDS     | 31            | 1      |        | 09/06/1998/340 |            | 14.37.33     | 20              |

```
* END OF DIRECTORY
```

Figure 20. Sample Listing of HTF Directory



---

## Part 4. Appendixes



---

## Appendix A. Job Control Scanner Program

This chapter describes how to use the Job-Control Scanner program. This program, subsequently referred to as “scanner”, assists in migrating from the pre-Version 2 **VSE/SP library format** to the format introduced with **VSE/SP Version 2**.

---

### Program Description

The scanner processes VSE/SP Version 1 job streams that have been copied onto a tape. The job streams on the tape may be one of the following:

- One or more CMS files (applies to a virtual machine environment).
- One or more VSE/ICCF libraries.
- One or more VSE/SP Version 1 source statement or procedure libraries.
- An undefined source.

Based on its input, the scanner produces output as follows:

- List output on the SYSLST device  
The format of this output depends on the type of input. Independent of which input type is used, this output flags the job control statements that have to be changed or replaced. It flags, in addition, the statements which should be changed eventually.
- Tape output on the SYSPCH device  
The scanner makes this output available if the job streams on the input tape originated from a VSE/SP Version 1 procedure or source statement library.

#### Notes:

1. The scanner processes the input line by line. It does not check for any kind of non-matching specifications.
2. Processing may fail for a member copied from a VSE/SP Version 1 library and containing more than 2000 statements.  
If the scanner finds no migration hit for the first 2000 statements, the program's buffer space is exhausted. As a result, the scanner discontinues processing the member and continues scanning the next member. The scanner informs you by a message.

### Operating Environment

The program needs a partition of:

- At least 240KB if job streams copied from a VSE/SP Version 1 procedure or source statement library are to be processed.
- Minimum size (128KB) if any other input is to be processed.

The possible types of input are discussed under “Preparing the Input” on page 118. You may, if this is desirable, use the scanner under VSE/SP Version 1. To do this, proceed as follows:

1. Punch out the program's object code to a tape whose drive is assigned to SYSPCH. This code is cataloged in the system sublibrary by this member name and type:

JCLSCAN.OBJ

Use the Librarian PUNCH function with the FORMAT=OLD specification for this punch-out run.

## Job Control Scanner

2. Link the punched out phase to a core image library of your VSE/SP Version 1 system.

## Preparing the Input

The scanner reads its input from the tape mounted on the drive to which the logical unit SYS007 is assigned. It expects the input records to have a length of either 80 or 81 bytes. The scanner reads the tape from beginning until it finds a tape mark on the tape. How to prepare the input is discussed by input type:

- CMS files (under VM) – see “CMS Files”.
- VSE/SP Version 1 source statement or procedure libraries – see “Members from Procedure or Source Statement Libraries”.
- VSE/ICCF libraries – see “VSE/ICCF Library Members”.
- Undefined source – see “Undefined Source Input”.

### CMS Files

This input is referred to as *type C*. You prepare the scanner-input tape by way of a MOVEFILE command. Figure 21 on page 119 shows a sequence of commands that creates a suitable input tape. The sequence assumes that the operator has mounted a scratch tape on the device known to VM as TAP1.

### Members from Procedure or Source Statement Libraries

This input is referred to as *type D*. You prepare the scanner-input tape by using the PSERV program to copy procedures and the SSERV program to copy source books. To ensure that a tape mark follows the last member on the tape, issue an MTC WTM command or a CLOSE command. For the scanner, a CATALP or CATALS statement marks the beginning of a member.

### VSE/ICCF Library Members

This input is referred to as *type I*. You prepare the scanner-input tape by using the DTSUTIL program with a PUNCH PUNCTL command. For the scanner, the statements ADD MEMBER and END OF MEMBER mark the beginning and end of a member.

### Undefined Source Input

This input is referred to as *type U*. The scanner expects a tape with a file containing sequences of job control statements. Make sure to write a tape mark at the end of the file.

---

```

(1) CH RDR CL J HOLD
(2) SP PU * CL J
(3) SP RDR CL J CONT

(4) FI INMOVE READER
(4) FI OUTMOVE TAP1 (LRECL 80 BLOCK 80 RECFM F DEN 6250
(5) MOVEFILE
(6) TAPE WTM
(6) TAPE REW
(7) SP PU OFF CL A
(7) SP RDR NOCONT CL A

```

---

Punch the members (CMS files) that are to be processed by the scanner

```

(1) CH RDR CL J HOLD
 Causes any files currently in your reader to be placed in the hold
 status. Do this as a precaution to avoid loss of data.
(2) SP PU * CL J
 Causes punch files to be spooled into the own reader, class J.
(3) SP RDR CL J CONT
 Ensures that all CMS files that are to be punched to the reader will be
 continuous (all contained in one file).
(4) FI INMOVE READER
 FI OUTMOVE TAP1 (LRECL 80 BLOCK 80 RECFM F DEN 6250
 They define the source and the target for the write-to-tape operation.
(5) MOVEFILE
 Initiates writing the spooled file to tape.
(6) TAPE xxx
 TAPE WTM causes a tape mark to be written at the end of the last member
 (CMS file) that is copied onto tape.
(7) SP PU OFF CL A
 SP RDR NOCONT CL A
 They restore the original spool controls see references (2) and (3) above.

```

---

Figure 21. Example for Preparing Scanner Input under VM

## Description of Output

### Output to SYSLST

The scanner always produces output to SYSLST. This output is essentially the same for input types C, D, and I. The scanner structures this output by members and prints only the affected statements. Each one of these statements is followed by an action-flag message. See Figure 22 on page 120 for an example of SYSLST output for input of type D.

The SYSLST output for input of type U is a list of *all* statements in the same sequence as they are stored in the input file. The scanner prints an action-flag message behind any statement that is affected by migration to VSE/ESA. The output differs from another scanner SYSLST output primarily by the absence of a structure by members. See Figure 23 on page 121 for an example of output for input of type U.

## Job Control Scanner

```
// JOB SCANTST
// ASSGN SYS007,280
// EXEC JCLSCAN,PARM='D,TRM6,155'
/ &
```

### Output on SYSLST:

```
LIBRARY 155 MEMBER STRTPR2
// ASSGN SYSSLB,300
*** STATEMENT IS NO LONGER SUPPORTED AND MUST BE REPLACED *****
// ASSGN SYSCLB,310
*** STATEMENT IS NO LONGER SUPPORTED AND MUST BE REPLACED *****
// LIBDEF CL,FROM=IJSYSRS
*** STATEMENT WILL BE MIGRATED BY THE SYSTEM, BUT SHOULD BE MODIFIED ***
// LIBLIST CL
*** STATEMENT WILL BE MIGRATED BY THE SYSTEM, BUT SHOULD BE MODIFIED ***
// EXEC BACKUP
*** PROGRAM NO LONGER SUPPORTED. REPLACE ALSO SYSIPT DATA *****
// EXEC MAINT
*** CHECK SYSIPT DATA *****
// EXEC MSHP
*** CHECK SYSIPT DATA *****
// OPTION LINK
*** CHECK IF MULTIPHASE OR OVERLAY PROGRAM IS BEING LINKED *****
END OF MEMBER STRTPR2
 E N D O F S C A N P R O G R A M
```

*Figure 22. Example of a Scanner SYSLST Output for Input of Type D*

```
// JOB SCANTST
// ASSGN SYS007,280
// EXEC JCLSCAN,PARM='U'
/ &
```

**Output on SYSLST:**

```
// JOB STRTPR2
// ASSGN SYSSLB,300
*** STATEMENT IS NO LONGER SUPPORTED AND MUST BE REPLACED *****
// ASSGN SYSCLB,310
*** STATEMENT IS NO LONGER SUPPORTED AND MUST BE REPLACED *****
// LIBDEF CL, FROM=IJSYSRS
*** STATEMENT WILL BE MIGRATED BY THE SYSTEM, BUT SHOULD BE MODIFIED ***
// LIBLIST CL
*** STATEMENT WILL BE MIGRATED BY THE SYSTEM, BUT SHOULD BE MODIFIED ***
DVCDN 180
// ASSGN SYS005,300
// ASSGN SYS006,280
// ASSGN SYS007,300
// ASSGN SYS008,300
// ASSGN SYS009,300
// EXEC BACKUP
*** PROGRAM NO LONGER SUPPORTED. REPLACE ALSO SYSIPT DATA *****
// EXEC MAINT
*** CHECK SYSIPT DATA *****
 DELETE MODUL1
// EXEC MSHPI
*** CHECK SYSIPT DATA *****
 RETRACE
// OPTION LINK
*** CHECK IF MULTIPHASE OR OVERLAY PROGRAM IS BEING LINKED *****
/*
/ &

 E N D O F L I S T I N G
 E N D O F S C A N P R O G R A M
```

Figure 23. Example of a Scanner SYSLST Output for Input of Type U

**Output to SYSPCH**

The scanner provides output to tape (assigned to SYSPCH) if the input is of type D. This output is suitable for being loaded into a VSE/ICCF library using the VSE/ICCF DTSUTIL program. You can then make changes as required using the editor of VSE/ICCF. The output of the scanner includes for this purpose:

- An ADD MEMBER statement before each member.
- An END OF MEMBER statement behind each member.

These statements are needed by the VSE/ICCF DTSUTIL program in order to load the scanner output.

- The applicable action-flag message behind each statement that is affected by migration to VSE/ESA. This is the same as for a scanner output to SYSLST.

CATALP and CATALS statements contained in the scanner input are included unchanged in this output. Following your update effort, you can submit the job(s) for recataloging in one of your system's VSE/SPLibraries. For ease of job submission to VSE/POWER, the scanner has inserted job delimiting JECL place holders.

Figure 24 on page 122 shows an example of records in a SYSPCH output of the scanner.

## Job Control Scanner

**Note:** If you do not want the scanner to provide output on the assigned SYSPCH tape, run the job with the statement

```
// ASSGN SYSPCH,IGN
```

```
// JOB SCANTST
// ASSGN SYS007,280
// ASSGN SYSPCH,281
// EXEC JCLSCAN,PARM='D,TRM6,155'
/&
```

### Output on SYSPCH:

```
ADD MEMBER 155 STRTPR2 TRM6
..$$ JOB JNM=...
// JOB ...
// EXEC LIBR
ACCESS SUBLIB=...
CATALP STRTPR2
// JOB STRTPR2
// ASSGN SYSSLB,300
*** STATEMENT IS NO LONGER SUPPORTED AND MUST BE REPLACED *****
// ASSGN SYSLB,310
*** STATEMENT IS NO LONGER SUPPORTED AND MUST BE REPLACED *****
// LIBDEF CL,FROM=IJSYSRS
*** STATEMENT WILL BE MIGRATED BY THE SYSTEM, BUT SHOULD BE MODIFIED ***
// LIBLIST CL
*** STATEMENT WILL BE MIGRATED BY THE SYSTEM, BUT SHOULD BE MODIFIED ***
DVCDN 180
// ASSGN SYS005,300
// ASSGN SYS006,280
// ASSGN SYS007,300
// ASSGN SYS008,300
// ASSGN SYS009,300
// EXEC BACKUP
*** PROGRAM NO LONGER SUPPORTED. REPLACE ALSO SYSIPT DATA *****
// EXEC MAINT
*** CHECK SYSIPT DATA *****
DELETE MODUL1
// EXEC MSHP
*** CHECK SYSIPT DATA *****
RETRACE
// OPTION LINK
*** CHECK IF MULTIPHASE OR OVERLAY PROGRAM IS BEING LINKED *****
/*
/&
../*
../&
..$$ EOJ
END OF MEMBER
```

Figure 24. Example of Records in a Scanner SYSPCH Output

## Starting a Scan Run

To do this, submit a job similar to the example shown and explained in Figure 25. The example assumes that permanent assignments exist for: SYSRDR, SYSIPT, and SYSLST.

Explanations for Figure 25:

```
(1) * $$ JOB JNM=SCNJOB1,CLASS=9,DISP=D
 // JOB SCANJCL
(2) // ASSGN SYS007,280
(3) // ASSGN SYSPCH,281
(4) // EXEC JCLSCAN,PARM='D,TRM6,155'
 /*
 /&
(1) * $$ E0J
```

The numbers within parentheses refer to explanations given following this sample job.

Figure 25. Sample Job for Starting a Scan Run

### (1) \* \$\$ ...

VSE/POWER statements. The system ignores these statements if you have the scanner run executed outside VSE/POWER control.

### (2) // ASSGN SYS007,280

Assign SYS007 to the drive on which your operator is to mount the input tape.

### (3) // ASSGN SYSPCH,281

Instruct your operator to mount a scratch tape on the drive that you specify. *However*, if you do not need the scanner's SYSPCH output, supply instead:

```
// ASSGN SYSPCH,IGN
```

and, as scanner control data, a dummy user identifier and a dummy VSE/ICCF library number. Example:

```
D,XYZ,999
```

### (4) // EXEC JCLSCAN,PARM='D,TRM6,155'

Supplies, by way of the PARM operand, the control data needed by the scanner.

If you use the scanner under Version 1 of VSE/Advanced Functions, this control data is to be submitted as SYSIPT data. In this case, your coding is like this:

```
// EXEC JCLSCAN
D,TRM6,155
```

The scanner requires its control data in a format as follows:  
where:



**t =** One of the following:

- C** A tape with input of type C (containing CMS files) is to be processed.
- I** A tape with input of type I (containing members of one or more VSE/ICCF libraries) is to be processed.
- U** A tape with input of type U (containing one or more job streams from an undefined source) is to be processed.

## Job Control Scanner

- D** A tape with input of type D (containing members from a VSE/SP Version 1 procedure or source statement library) is to be processed. If you specify this input type, you must also specify a user identifier and a library number as described below.

**user-id**

For user-id, specify an identifier of one to four characters. The scanner inserts this identifier, TRM6 in the example, into the ADD MEMBER statements of the SYSPCH output for the run.

**library-no**

For library-no, supply the three-digit number of the VSE/ICCF library into which the scanner's SYSPCH output is to be loaded for later editing.

## Messages

The scanner issues two kinds of messages:

- Action-flag messages

A message of this type follows an individual job control statement of the job stream(s) being scanned. It indicates the action to be taken in order to safely migrate the affected job stream to VSE/ESA.

- Error messages

A message of this type indicates an error condition found by the scanner. They are documented in the manual *VSE/ESA Messages and Codes*.

All messages are directed to the device assigned to SYSLST.

### Action-Flag Messages

**\*\*\* STATEMENT WILL BE MIGRATED BY THE SYSTEM BUT SHOULD BE MODIFIED**

**Explanation:** A statement so flagged is to be changed if no library-migration table exists.

**\*\*\* STATEMENT WILL BE MIGRATED BY THE SYSTEM BUT SHOULD BE REPLACED**

**Explanation:** A statement so flagged is to be changed if it requests a function not available with the VSE/ESA Librarian.

**\*\*\* CHECK IF MULTIPHASE OR OVERLAY PROGRAM IS BEING LINKED**

**Explanation:** A // OPTION LINK statement was found in the job stream, which indicates immediate program execution. VSE/ESA does not support the link-and-go function for a multiphase or an overlay program.

**Bypass:** Link your phases permanently (// OPTION CATAL) into a sublibrary available for program testing.

**\*\*\* STATEMENT IS NO LONGER SUPPORTED AND MUST BE REPLACED**

**Explanation:** Self-explanatory.

**\*\*\* NO INCOMPATIBLE STATEMENTS**

**Explanation:** In the scanned member, the scanner found no statements that should be changed or replaced.

**\*\*\* FOLLOWING MEMBER DOES NOT CONTAIN INCOMPATIBLE STATEMENTS**

**Explanation:** In the member named after this message, the scanner found no statements that should be changed or replaced.

**\*\*\* PROGRAM NO LONGER SUPPORTED. REPLACE ALSO SYSIPT DATA**

**Explanation:** The flagged statement is to be replaced, including the program control statements that follow the flagged statement.

**\*\*\* CHECK SYSIPT DATA**

**Explanation:** The SYSIPT data for the involved // EXEC request may have to be changed.



---

## Appendix B. Understanding Syntax Diagrams

This section describes how to read the syntax diagrams in this manual.

To read a syntax diagram follow the path of the line. Read from left to right and top to bottom.

- The ►— symbol indicates the beginning of a syntax diagram.
- The —► symbol, at the end of a line, indicates that the syntax diagram continues on the next line.
- The ►— symbol, at the beginning of a line, indicates that a syntax diagram continues from the previous line.
- The —►◄ symbol indicates the end of a syntax diagram.

Syntax items (for example, a keyword or variable) may be:

- Directly on the line (required)
- Above the line (default)
- Below the line (optional)

### Uppercase Letters

Uppercase letters denote the shortest possible abbreviation. If an item appears entirely in uppercase letters, it can not be abbreviated.

You can type the item in uppercase letters, lowercase letters, or any combination. For example:

►►—KEYWOrd—►►

In this example, you can enter KEYWO, KEYWOR, or KEYWORD in any combination of uppercase and lowercase letters.

### Symbols

You **must** code these symbols exactly as they appear in the syntax diagram

|    |              |
|----|--------------|
| *  | Asterisk     |
| :  | Colon        |
| ,  | Comma        |
| =  | Equal Sign   |
| -  | Hyphen       |
| // | Double slash |
| () | Parenthesis  |
| .  | Period       |
| +  | Add          |

For example:

\* \$\$ LST

### Variables

Highlighted lowercase letters denote variable information that you must substitute with specific information. For example:



Here you must code `USER=` as shown and supply an ID for `user_id`. You may, of course, enter `USER` in lowercase, but you must not change it otherwise.

### Repetition

An arrow returning to the left means that the item can be repeated.



A character within the arrow means you must separate repeated items with that character.



A footnote (1) by the arrow references a limit that tells how many times the item can be repeated.



### Notes:

- 1 Specify `repeat` up to 5 times.

### Defaults

Defaults are above the line. The system uses the default unless you override it. You can override the default by coding an option from the stack below the line. For example:



In this example, `A` is the default. You can override `A` by choosing `B` or `C`.

### Required Choices

When two or more items are in a stack and one of them is on the line, you **must** specify one item. For example:



Here you must enter either A or B or C.

### Optional Choice

When an item is below the line, the item is optional. Only one item **may** be chosen. For example:



Here you may enter either A or B or C, or you may omit the field.

### Required Blank Space

A required blank space is indicated as such in the notation. For example:

\* \$\$ E0J

This indicates that at least one blank is required before and after the characters \$\$.



---

# Glossary

This glossary includes terms and definitions related primarily to IBM VSE/ESA with the base program VSE Central Functions. For other VSE/ESA component programs whose terms are not included in this glossary, such as VTAM or CICS/VSE, refer to the respective General Information manuals. This glossary includes terms and definitions from the IBM Dictionary of Computing, New York: McGraw-Hill, 1994.

The glossary includes definitions with:

- Symbol \* where there is a one-to-one copy from the IBM Dictionary of Computing.
- Symbol (A) from the *American National Dictionary for Information Systems* ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42 Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- Symbol (E) from the ANSI/EIA Standard-440-A: *Fiber Optic Terminology*. Copies may be purchased from the Electronic Industries Association, 2001 Pennsylvania Avenue, N.W., Washington DC 20006. Definitions are identified by the symbol (E) after the definition.
- Symbols (I) or (T) from the *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.

**access control.** A function of VSE that ensures that the system and the data and programs stored in it can be accessed only by authorized users in authorized ways.

**access control table (DTSECTAB).** A table used by the system to verify a user's right to access a certain resource.

**access method.** A program, that is, a set of commands (macros), to define files or addresses and to move data to and from them; for example VSE/VSAM.

**address space.** A subdivision of the total of virtual storage.

**alternate library.** A library which becomes accessible from a terminal when the user of that terminal issues a connect or switch (library) request.

\* **alternate tape.** A tape drive to which the operating system switches automatically for tape read or write operations if the end of the volume has been reached on the originally used tape drive.

**alternate track.** On a CKD disk, a track designated to contain data in place of a defective track.

\***American National Standard Code for Information Interchange (ASCII).** The standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters. (A)

\* **application profile.** A control block in which the system stores the characteristics of one or more application programs.

**application program.** A program written for or by a user that applies directly to the user's work. See also *batch program* and *online application program*.

**assembler.** A computer program used to assemble.

**attention routine.** A routine of the system that receives control when the operator presses the Attention key. The routine sets up the console for the input of a command, reads the command, and initiates the system service requested by the command.

**auxiliary storage.** Addressable storage that is not part of the processor, for example storage on a disk unit. Synonymous with *external storage*.

**batch processing.** 1. Serial processing of computer programs. 2. Pertaining to the technique of processing a set of computer programs in such a way that each is completed before the next program of the set is started. (A)

**batch program.** A program that is processed in series with other programs and therefore normally processes data without user interaction.

**block.** Usually, a block consists of several records of a file that are transmitted as a unit. But if records are very large, a block can also be part of a record only. See also *control block*.

**blocking.** The process of combining (or cutting) records into blocks.

**catalog.** 1. A directory of files and libraries, with reference to their locations. A catalog may contain other information such as the types of devices in which the files are stored, passwords, blocking factors. (I) (A) 2. To store a library member such as a phase, module, or book in a sublibrary.

See also *VSAM master catalog*, *VSAM user catalog*.

**\* cataloged procedure.** A set of control statements placed in a library and retrievable by name.

**\* catalog recovery area (CRA).** In systems with VSAM, an entry-sequenced data set that exists on each volume owned by a recoverable catalog, including the catalog volume itself. The CRA contains copies of the catalog records and can be used to recover a damaged or invalid catalog.

**channel command word (CCW).** One or more CCWs make up the channel program that directs data channel operations.

**channel program.** One or more channel command words that control a sequence of data channel operations. Execution of this sequence is initiated by a single start I/O (SIO) instruction.

**CKD device.** Count-key-data device.

**\* compile.** To translate a source program into an executable program (object program). See also *assembler*.

**component.** 1. Hardware or software that is part of a computer system. 2. A functional part of a product, identified by a component identifier. 3. In VSE/VSAM, a named, cataloged group of stored records, such as the key-sequenced file or alternate index.

**computer system.** A functional unit of one or more computers and their associated software.

**configuration.** The devices and programs that make up a system, subsystem, or network.

**control block.** An area within a program or a routine defined for the purpose of storing and maintaining control information.

**control interval (CI).** A fixed length area of disk storage where VSE/VSAM stores records and distributes free space. It is the unit of information that VSE/VSAM transfers to or from disk storage. For FBA, it must be an integral multiple, to be defined at cluster definition, of the block size.

**count-key-data (CKD) device.** A disk device that stores data in the record format: count field, key field, data field. The count field contains, among others, the address of the record in the format: cylinder, head (track), record number and the length of the data field. The key field, if present, contains the record's key or search argument. CKD disk space is allocated by tracks and cylinders. Contrast with *FBA disk device*. See also *extended count-key-data device*.

**\* data block (DBLK).** In VSE/POWER, the unit of transfer for spooling job input and job output.

**data file.** See *file*.

**data set.** See *file*.

**deblocking.** The process of making each record of a block available for processing. Contrast with *blocking*.

**default value.** A value assumed by the program when no value has been specified by the user.

**\* device address.** 1. The identification of an input/output device by its channel and unit number. 2. In data communication, the identification of any device to which data can be sent or from which data can be received.

**\* device class.** The generic name for a group of device types, for example, all display stations belong to the same device class. Contrast with *device type*.

**\* Device Support Facilities.** A program supplied for performing operations on disk volumes so that they can be accessed by IBM and user programs. Examples of these operations are initializing a disk volume and assigning an alternate track.

**\* device type code.** The four- or five-digit code to be used for defining an I/O device to a computer system.

**\* dialog.** 1. In an interactive system, a series of related inquiries and responses similar to a conversation between two people. 2. For VSE/ESA, a set of panels that can be used to complete a specific task, for example, defining a file.

**direct access.** Accessing data on a storage device using their address and not their sequence. This is the typical access on disk devices as opposed to magnetic tapes. Contrast with *sequential access*.

**directory.** 1. A table of identifiers and references to the corresponding items of data. (I) (A) 2. In VSE, specifically, the index for the program libraries. See also *library directory* and *sublibrary directory*.

**disk operating system residence volume (DOSRES).** The disk volume on which the system sublibrary IJSYSRS.SYSLIB is located including the programs and procedures required for system startup.

**disposition.** A means of indicating to VSE/POWER how job input and output is to be handled. A job may, for example, be deleted or kept after processing.

**DOSRES.** Disk operating system residence volume.

**dynamic partition.** A partition created and activated on an 'as needed' basis that does not use fixed static allocations. After processing, the occupied space is released. Contrast with *static partition*.

**EBCDIC.** Extended binary-coded decimal interchange code.

**ECKD device.** Extended count-key-data device.

**emulation.** The use of programming techniques and special machine features that permit a computer system to execute programs written for another system or for the use of I/O devices different from those that are available.

**Enterprise Systems Architecture (ESA).** See *ESA/390*.

**environmental record editing and printing (EREP) program.** The program that makes the data contained in the system recorder file available for further analysis.

**EREP program.** Environmental record editing and printing program.

**ESA/390.** IBM Enterprise Systems Architecture/390. The latest extension to the IBM System/370 architecture which includes the advanced addressability feature and advanced channel architecture.

**\* escape.** To return to the original level of a user interface.

**extended count-key-data (ECKD) device.** A disk storage device that has a data transfer rate faster than some processors can utilize. A specialized channel program is needed to convert ordinary CKD channel programs for use with an ECKD device.

**extent.** Continuous space on a disk or diskette occupied by or reserved for a particular file or VSAM data space.

**extended binary-coded decimal interchange code (EBCDIC).** A coded character set consisting of 8-bit coded characters.

**external storage.** Storage that is not part of the processor.

**FBA disk device.** Fixed-block architecture disk device.

**file.** A named set of records stored or processed as a unit. (T) Synonymous with *data set*.

**fixed-block architecture (FBA) disk device.** A disk device that stores data in blocks of fixed size. These

blocks are addressed by block number relative to the beginning of the file. Contrast with *CKD device*.

**\* forms control buffer (FCB).** In the 3800 Printing Subsystem, a buffer for controlling the vertical format of printed output.

**\* fragmentation (of storage).** Inability to allocate unused sections (fragments) of storage in the real or virtual address range of virtual storage.

**generate.** To produce a computer program by selecting subsets of standardized code under the control of parameters. (A)

**generation.** See *macro generation*.

**\* GETVIS space.** Storage space within a partition or the shared virtual area, available for dynamic allocation to programs.

**hard-copy file.** A system file on disk, used to log all lines of communication between the system and the operator at the system console, to be printed on request.

**hard wait.** The condition of a processor when all operations are suspended. System recovery from a hard wait is impossible without performing a new system startup.

**hardware.** Physical equipment used in data processing, as opposed to programs, procedures, rules, and associated documentation. (I) (A) Contrast with *software*.

**High Level Assembler for VSE.** A VSE/ESA base program. This is the most sophisticated assembler available for VSE, VM, and MVS.

**\* initial program load (IPL).** The process of loading system programs and preparing the system to run jobs.

**interactive.** A characteristic of a program or system that alternately accepts input and then responds. An interactive system is conversational, that is, a continuous dialog exists between user and system. Contrast with *batch*.

**Interactive Computing and Control Facility (ICCF).** Part of the VSE/ESA base program VSE Central Functions, that serves as interface, on a time-slice basis, to authorized users of terminals linked to the system's processor.

**interactive interface.** A system facility which controls how different users see and work with the system by means of user profiles. When signing on, the interactive interface makes available those parts of the system authorized by the profile. The interactive interface has sets of selection- and data-entry panels through which users communicate with the system.

**interactive partition.** An area of virtual storage for the purpose of processing a job that was submitted interactively via VSE/ICCF.

**interface.** A shared boundary between two hardware or software units defined by common functional or physical characteristics. It might be a hardware component or a portion of storage or registers accessed by several computer programs.

\* **intermediate storage.** Any storage device used to hold data temporarily before it is processed. See also *buffer storage*.

**job control language (JCL).** A language that serves to prepare a job or each job step of a job to be run. Some of its functions are: to identify the job, to determine the I/O devices to be used, set switches for program use, log (or print) its own statements, and fetch the first phase of each job step.

**job control statement.** A particular statement of JCL.

**job step.** One of a group of related programs complete with the JCL statements necessary for a particular run. Every job step is identified in the job stream by an EXEC statement under one JOB statement for the whole job.

**job stream.** The sequence of jobs as submitted to an operating system.

\* **kilobyte (KB).** 1024 bytes.

**label information area.** An area on a disk to store label information read from job control statements or commands. Synonymous with *label area*.

**language translator.** A general term for any assembler, compiler, or other routine that accepts statements in one language and produces equivalent statements in another language.

\* **librarian.** The set of programs that maintains, services, and organizes the system and private libraries.

**library.** See *VSE library* and *VSE/ICCF library*.

\* **library directory.** The index that enables the system to locate a certain sublibrary of the accessed library.

\* **library member.** The smallest unit of data to be stored in and retrieved from a sublibrary.

\* **licensed program.** A separately priced program and its associated materials that bear an IBM copyright and are offered to customers under the terms and conditions of the IBM Customer Agreement (ICA).

\* **local address.** In SNA, an address used in a peripheral node in place of an SNA network address and transformed to or from an SNA network address by the boundary function in a subarea node.

\* **logging.** The recording of data about specific events.

**logical record.** A user record, normally pertaining to a single subject and processed by data management as a unit. Contrast with *physical record* which may be larger or smaller.

**logical unit (LU).** 1. A name used in programming to represent an I/O device address. 2. In SNA, a port through which a user accesses the SNA network, a) to communicate with another user and b) to access the functions of the SSCP. An LU can support at least two sessions - one with an SSCP and one with another LU - and may be capable of supporting many sessions with other LUs. See also *network addressable unit (NAU)*, *peripheral LU*, *physical unit (PU)*, *system services control point (SSCP)*, *primary logical unit (PLU)*, and *secondary logical unit (SLU)*.

**logical unit name.** In programming, a name used to represent the address of an input/output unit.

**macro expansion.** See *macro generation*.

**macro generation.** An assembler operation by which a macro instruction gets replaced in the program by the statements of its definition. It takes place before assembly. Synonymous with *macro expansion*.

\* **maintain system history program (MSHP).** A program used for automating and controlling various installation, tailoring, and service activities for a VSE system.

\* **megabyte (MB).** 1 024 KB or 1 048 576 bytes.

\* **member.** The smallest unit of data that can be stored in and retrieved from a sublibrary.

**message.** 1. In VSE, a communication sent from a program to the operator or user. It can appear on a console, a display terminal or on a printout. 2. In telecommunication, a logical set of data being transmitted from one node to another.

\* **migrate.** To move to a changed operating environment, usually to a new release or version of a system.

\* **module.** A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading; for example, the input to, or output from, an assembler, a compiler, linkage editor, or executive routine. (A)

**NAU.** Network addressable unit.

**NetView.** An IBM licensed program to monitor a network, manage it, and diagnose its problems.

**network.** 1. An arrangement of nodes (data stations) and connecting branches. 2. The assembly of equipment through which connections are made between data stations.

**network address.** In SNA, an address, consisting of subarea and element fields, that identifies a link, link station, or NAU. Subarea nodes use network addresses; peripheral nodes use local addresses. The boundary function in the subarea node to which a peripheral node is attached transforms local addresses to network addresses and vice versa. See *local address*. See also *network name*.

**network addressable unit (NAU).** In SNA, a logical unit, a physical unit, or a system services control point. It is the origin or the destination of information transmitted by the path control network. Each NAU has a network address that represents it to the path control network. See also *network name*, *network address*.

**network name.** 1. In SNA, the symbolic identifier by which users refer to a NAU, link, or link station. See also *network address*. 2. In a multiple-domain network, the name of the APPL statement defining a VTAM application program. This is its network name which must be unique across domains.

\* **object code.** Output from a compiler or assembler which is itself executable machine code or is suitable for processing to produce executable machine code. (A)

**object module (program).** A program unit that is the output of an assembler or compiler and is input to a linkage editor.

**online application program.** An interactive program used at display stations. When active, it waits for data. Once input arrives, it processes it and sends a response to the display station or to another device.

\* **operating system.** Software that controls the running of programs; an operating system may provide services such as resource allocation, scheduling, input/output control, and data management. (I) (A)

**page data set (PDS).** One or more extents of disk storage in which pages are stored when they are not needed in processor storage.

\* **page fixing.** Marking a page so that it is held in processor storage until explicitly released. Until then, it cannot be paged out.

**page frame.** An area of processor storage that can contain a page.

\* **page pool.** The set of page frames available for paging virtual-mode programs.

**partition.** A division of the virtual address area available for running programs. See also *dynamic partition*, *static partition*.

\* **peripheral LU.** In SNA, a logical unit in a peripheral node.

\* **physical record.** The amount of data transferred to or from auxiliary storage. Synonymous with *block*.

\* **physical unit (PU).** In SNA, the component that manages and monitors the resources of a node, such as attached links and adjacent link stations, as requested by an SSCP via an SSCP-SSCP session.

**PLU (primary logical unit).** In SNA, the LU that contains the primary half-session for a particular LU-LU session. A particular LU may contain both primary and secondary half-sessions for different active LU-LU sessions. Contrast with *SLU*.

\* **primary library.** A VSE library owned and directly accessible by a certain terminal user.

**primary logical unit (PLU).** In SNA, the LU that contains the primary half-session for a particular LU-LU session. A particular LU may contain both primary and secondary half-sessions for different active LU-LU sessions. Contrast with *SLU*.

**priority.** A rank assigned to a partition or a task that determines its precedence in receiving system resources.

\* **private library.** A user-owned library that is separate and distinct from the system library.

**procedure.** See *cataloged procedure*.

\* **processing.** The performance of logical operations and calculations on data, including the temporary retention of data in processor storage while this data is being operated upon.

**processor.** The hardware component that interprets and executes instructions. (I) (A)

**processor storage.** The storage contained in one or more processors and available for running machine instructions. Synonymous with *real storage*.

**profile.** A description of the characteristics of a user or a computer resource.

\* **programmer logical unit.** A logical unit available primarily for user-written programs. See also *logical unit name*.

**program product.** See *licensed program*.

**prompt.** To issue messages to a terminal or console user, requesting information necessary to continue processing.

**PU.** Physical unit.

**real mode.** In VSE, a processing mode in which a program may not be paged. Contrast with *virtual mode*.

**real storage.** See *processor storage*.

\* **record.** A collection of related data or words, treated as a unit. See *logical record*, *physical record*.

\* **restore.** To write back on disk data that was previously written from disk to an intermediate storage medium such as tape.

\* **routine.** Part of a program, or a sequence of instructions called by a program, that may have some general or frequent use. (I) (A)

\* **search chain.** The order in which chained sublibraries are searched for the retrieval of a certain library member of a specified type.

**secondary logical unit (SLU).** In SNA, the LU that contains the secondary half-session for a particular LU-LU session. An LU may contain secondary and primary half-sessions for different active LU-LU sessions. Contrast with *PLU*.

**security.** See *access control*.

\* **selection panel.** A displayed list of items from which a user can make a selection. Synonymous with *menu*.

**sequential access.** The serial retrieval of records in their entry sequence or serial storage of records with or without a premeditated order. Contrast with *direct access*.

**sequential access method (SAM).** A data access method that writes to and reads from an I/O device record after record (or block after block). On request, the support performs device control operations such as line spacing or page ejects on a printer or skip a certain number of tape marks on a tape drive.

**sequential file.** A file in which records are processed in the order in which they are entered and stored.

\* **service program.** A program in general support of computer processes, for example, a diagnostic program, a trace program, or a sort program. (T) Synonymous with *utility program*.

\* **shared virtual area (SVA).** In VSE, a high address area that contains the system directory list (SDL) of frequently used phases, resident programs shared between partitions, and an area for system support.

**skeleton.** A set of control statements and/or instructions that requires user-specific information to be inserted before it can be submitted for processing.

**SLU.** Secondary logical unit.

\* **software.** Programs, procedures, rules, and any associated documentation pertaining to the operation of a computer system.

**source statement.** A statement written in symbols of a programming language.

\* **spooling.** The use of disk storage as buffer storage to reduce processing delays when transferring data between peripheral equipment and the processor of a computer. In VSE, this is done under the control of VSE/POWER.

**SSCP.** System services control point.

**stand-alone program.** A program that runs independently of (not controlled by) the VSE system.

\* **standard label.** A fixed-format record that identifies a volume of data such as a tape reel or a file that is part of a volume of data.

**startup.** The process of performing IPL of the operating system and of getting all subsystems and application programs ready for operation.

**static partition.** A partition, defined at IPL time and occupying a defined amount of virtual storage that remains constant. Contrast with *dynamic partition*.

**station.** 1. One of the input or output points of a network that uses communication facilities; for example, the telephone set in the telephone system or the point where the business machine interfaces with the channel on a leased private line. 2. One or more computers, terminals, or devices at a particular location.

\* **subarea.** A portion of the SNA network consisting of a subarea node, attached peripheral nodes, and associated resources. Within a subarea node, all NAUs, links, and adjacent link stations in attached peripheral or subarea nodes that are addressable within the subarea share a common subarea address and have distinct element addresses.

**sublibrary.** In VSE, a subdivision of a library. Members can only be accessed in a sublibrary.

**sublibrary directory.** An index for the system to locate a member in the accessed sublibrary.

**submit.** A VSE/POWER function that passes a job to the system for processing.

\* **subsystem.** A secondary or subordinate system or program, usually capable of operating independently of, or asynchronously with, the operating system.

\* **supervisor.** The part of a control program that coordinates the use of resources and maintains the flow of processor operations.

**SYSRES.** System residence file.

\* **system console.** A console, usually equipped with a keyboard and display screen for control and communication with the system.

**system directory list (SDL).** A list containing directory entries of frequently-used phases and of all phases resident in the SVA. The list resides in the SVA.

\* **system file.** In VSE, a file used by the operating system, for example, the hard-copy file, the recorder file, the page data set.

**system recorder file.** The file that is used to record hardware reliability data. Synonymous with *recorder file*.

**system residence file (SYSRES).** The system sublibrary IJSYSRS.SYSLIB that contains the operating system. It is stored on the system residence volume DOSRES.

**system services control point (SSCP).** In SNA, a focal point within an SNA network for managing the configuration, coordinating network operator and problem determination requests, and providing directory support and other session services for users of the network. Multiple SSCPs, cooperating as peers, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units and logical units within its domain.

**system sublibrary.** The sublibrary that contains the operating system. It is stored on the system residence volume (DOSRES).

\* **tailor.** A process that defines or modifies the characteristics of the system.

\* **telecommunication.** Transmission of data between computer systems and between such a system and remote devices.

**terminal.** A point in a system or network at which data can either enter or leave. (A) Usually a display screen with a keyboard.

\* **utility program.** 1. A program in general support of computer processes, for example, a diagnostic program, a trace program, or a sort program. (T) Synonymous with *service program*. 2. A program that performs an everyday task such as copying data from one storage device to another. (A)

**virtual address.** An address that refers to a location in virtual storage. It is translated by the system to a processor storage address when the information stored at the virtual address is to be used.

**virtual address area.** The virtual range of available program addresses.

**virtual address space.** In VSE, a subdivision of the virtual address area available to the user for the allocation of private (non-shared) partitions.

\* **virtual machine.** A functional simulation of a computer system and its associated devices.

**Virtual Machine/Enterprise Systems Architecture (VM/ESA).** The most advanced VM system currently available.

\* **virtual mode.** The operating mode of a program which may be paged.

**virtual storage.** Addressable space image for the user from which instructions and data are mapped into processor (real) storage locations.

**VM/ESA.** Virtual Machine/Enterprise Systems Architecture.

**volume.** A data carrier that is mounted and demounted as a unit, for example, a reel of tape or a disk pack. (I) Some disk units have no demountable packs. In that case, a volume is the portion available to one read/write mechanism.

**volume ID.** The volume serial number, which is a number in a volume label assigned when a volume is prepared for use by the system.

**volume table of contents (VTOC).** A table on a disk volume that describes every file on it.

**VSE (Virtual Storage Extended).** A system that consists of a basic operating system and any IBM supplied and user-written programs required to meet the data processing needs of a user. VSE and the hardware it controls form a complete computing system. Its current version is called VSE/ESA.

**VSE/Advanced Functions.** Part of VSE Central Functions, a base program of VSE/ESA.

**VSE/ESA (VSE/Enterprise Systems Architecture).** The most advanced VSE system currently available.

**VSE/Fast Copy.** Part of the VSE/ESA base program VSE Central Functions. It is a utility for fast copy data operations from disk to disk and dump/restore operations via an intermediate dump file on magnetic tape or disk.

\* **VSE/ICCF (VSE/Interactive Computing and Control Facility).** Part of the VSE/ESA base program VSE Central Functions, that serves as interface, on a time-slice basis authorized users of terminals linked to the system's processor.

**VSE/ICCF library.** A file composed of smaller files (libraries) including system and user data which can be accessed under the control of VSE/ICCF.

**VSE library.** A collection of programs in various forms and storage dumps stored on disk. The form of a program is indicated by its member type such as source code, object module, phase, or procedure. A VSE library consists of at least one sublibrary which can contain any type of member.

**VSE/OCCF (Operator Communication Control Facility).** An IBM optional program that helps reduce operator interaction in the operation of a VSE-controlled installation and helps centralize data processing skills.

\* **VSE/POWER.** Part of the VSE/ESA base program VSE Central Functions. Primarily used to spool input and output. The program's networking functions enable a VSE system to exchange files with or run jobs on another remote processor.

**wait state.** The condition of a processor when all operations are suspended. System recovery from a hard wait is impossible without performing a new system startup. Synonym for *hard wait*.

---

# Index

## A

- ADD statement (in IESUPDCF) 71
- adding a user ID (in IESUPDCF) 69, 71
- ALL dump/copy/restore 20
- ALL function 20
- ALL operand
  - VSE/Fast Copy online 7, 12
  - VSE/Fast Copy stand-alone 34
- ALTER statement (in IESUPDCF) 75
- altering
  - a user ID (in IESUPDCF) 69, 75
- alternating buffers 23
- ASSGN statement (EREP program) 105
- assigning
  - devices 24
  - SYSIPT 88
- assignments, device 24
- Attention Routine (AR) commands 32

## B

- backup of system disks 21
- batch access to HTF 109
- blocking for
  - CKD disks 26
  - FBA disks 26
- buffers, alternating 23

## C

- CICS/VSE sign-on table (DFHSNT) 53
- CISIZE value 25
- CKD disk
  - blocking 26
  - with OPTIMIZE 14
- CLASSTD= (LSERV parameter) 101
- CLASSTD (LSERV parameter) 101
- clear disk utility (CLRDK)
  - DLBL job control statement for UOUT 89
  - END statement 89
  - format of 89
  - purpose 88
  - using 89
- command symbols 127
- compatibility modes for disks 87
- continuation of statements 7
- control statement input, device utilities
  - job control 87
  - utility control 87
- conventions, command 127
- copy and restore diskette utility (CRDR)
  - file descriptor statement 91
  - format 91
  - operands 91
  - job control statements 91
  - purpose 90
  - using 93
  - utility control statement 91

- copy and restore diskette utility (CRDR) *(continued)*
  - format 92
- COPY utility control statement
  - stand-alone FASTCOPY 34
  - VSE/Fast Copy online 10
- copying
  - a file 20
  - a multivolume file 12
  - a partial volume file 10
  - a volume 20
  - to a larger volume 10
  - with the ALL function 20
- creating
  - a backup of system disks 21
  - a FASTCOPY backup tape (multiple disk volume dumps) 47
  - a stand-alone FASTCOPY tape using CUSTTABLE 46
- customization table for stand-alone utilities 41
- customizing stand-alone utilities 41
  - creating FASTCOPY tape using CUSTTABLE 46
  - customization table (CUSTTABLE) 41
    - customized answer table 42
    - customized message table 42
  - examples 44
  - macro SASTCK 43
    - example for using 44
    - format 43
    - parameters 43

## D

- DAM file
  - direct addressing 12
  - relocation of 12
- data set, intermediate 26
  - skipping of 14
  - space requirements on disk 26
  - used with FBA applications for VSE/Fast Copy online 23
- DELETE operation (DTRSETP utility) 83
- DELETE statement (in IESUPDCF) 76
- deleting a user ID (in IESUPDCF) 69, 76
- description 117
- device assignments 24
- device support
  - disks 87
  - magnetic tapes 87
- device utilities
  - disk compatibility modes 87
  - storage requirements 88
- DFHSNT (CICS/VSE sign-on table) 53
- dialog, Display VTOC 103
- direct addressing, DAM file 12
- disk compatibility modes for device utilities 87
- disk error handling 18
  - for CKD 18
  - for FBA 18

- disk space for intermediate data set 26
- Display VTOC dialog 103
- displaying
  - label information area (LSERV) 101
  - program description 101
  - volume table of contents (VTOC)
    - using LVTOC program from SYSLOG 103
    - using LVTOC program from SYSRDR 103
    - using the dialog 103
- DITTO/ESA for VSE (stand-alone) 40
- DLBL statement 23, 102
- DTRIINIT utility
  - additional considerations 85
  - purpose 84
  - SKCOLD skeleton 85
- DTRSETP utility 81
  - operations 82
  - using DTRSETP 81
  - using SETPARM procedures 81
- DUMP utility control statement
  - stand-alone FASTCOPY 34
  - VSE/Fast Copy online 7
- dumping
  - a file 20
  - a multivolume file 12
  - a volume 20
  - with EXCLUDE 28
  - with the ALL function 20

## E

- END statement (CLRDK utility) 89
- EREP (Environmental Recording Editing and Printing)
  - program 104
- EREP reports, types of 105
- error handling 18
  - CKD disks 18
  - during migration 64
  - FBA disks 18
  - return codes 18
- examples 119, 120, 121, 122
  - clear and preformat one extent using CLRDK 89
  - completed skeleton IESUPDCF 80
  - console input for VSE/Fast Copy online 25
  - copy multivolume file using VSE/Fast Copy online 30
  - copy partial volume using VSE/Fast Copy online 27
  - copy to a different diskette unit using CRDR 93
  - creating Status Report of existing user IDs 58
  - dump a CKD/ECKD volume using VSE/Fast Copy online 28
  - dump a volume onto tape using stand-alone FASTCOPY 35
  - dump with EXCLUDE using OPTIMIZE with VSE/Fast Copy online 28
  - EREP job stream 104
  - examples for Job Control Scanner 120
  - EXEC statement for CKD/ECKD disks (VSE/Fast Copy online) 23
  - EXEC statement for FBA disks (VSE/Fast Copy online) 23

- examples 119, 120, 121, 122 *(continued)*
  - EXEC statement with SIZE value specified 88
  - formats for VSE/Fast Copy online utility control statements 7, 8
  - initialize one ASCII tape without volume label image option using INTTP 99
  - initialize one EBCDIC tape using INTTP 99
  - initialize two ASCII tapes with volume label image option using INTTP 100
  - Job Control Scanner Outputs 120
  - LSERV output 102
  - LVTOC job streams 103
  - migrating to VSE/ESA from DOS/VSE 58
  - migrating to VSE/ESA from VSE/SP 57
  - preparing Job Control Scanner input under VM 119
  - printing the recorder file 105
  - records in a Job Control Scanner SYSPCH output 122
  - restore a volume using stand-alone FASTCOPY 36
  - restore a volume using VSE/Fast Copy online 28
  - selective restore to other location using VSE/Fast Copy online 29
  - selective restore to same location using VSE/Fast Copy online 29
  - Starting a Job Control Scanner run 123
  - track table for CRDR 94
  - using an intermediate disk for the second run of CRDR 95
  - using macro SASTCK (stand-alone utilities) 44
  - using the customization table (stand-alone utilities) 44
- EXCLUDE operand (VSE/Fast Copy online) 12
- EXEC statement (for invoking VSE/Fast Copy online) 23
  - EXEC FCOPY 23
  - EXEC FCOPYB 23
- EXTENT statement 23, 102

## F

- Fast Copy program, online
  - ALL dump/copy/restore 20
  - COPY utility control statement 10
  - copying or restoring to a larger volume 10
  - device assignments 24
  - DUMP utility control statement 7
  - error handling 18
    - for CKD disks 18
    - for FBA disks 18
    - return codes 18
  - examples for using VSE/Fast Copy online 27
  - file copy 20
  - file dump 20
  - file restore 20
  - for CKD disks 18
  - for FBA disks 18
  - for using OPTIMIZE 15
  - formats of
    - COPY utility control statement 8
    - DUMP utility control statement 7
    - RESTORE utility control statement 7
  - integrity of system files 16

Fast Copy program, online (*continued*)

- job control statements 23
  - accepting control statements from SYSLOG 25
  - device assignments and label information 24
  - DLBL 23
  - EXEC 23
  - EXTENT 23
  - TLBL 23
  - UPSI 23
- label information 24
- multivolume file dump/copy/restore 12
- NOREWIND 13
- operands 12
- OPTIMIZE 14
- performance considerations 26
  - reading from and writing to an FBA disk 27
  - running in virtual mode 26
- program description 5
- program functions 5
  - ALL dump/copy/restore 7
  - COPY FILE 11
  - COPY VOLUME 9
  - DUMP FILE 10
  - DUMP VOLUME 8
  - multivolume file dump/copy/restore 12
  - NOREWIND 13
  - OPTIMIZE 14
  - overview 3, 5
  - partial volume, dump/copy 9
  - reorganization of files 12
  - RESTORE FILE 11
  - RESTORE VOLUME 9
  - selective restore 11, 20
- restoring onto VM minidisks 16
- restrictions for 10
  - copying or restoring to a larger volume 10
  - integrity of system files 16
  - restoring onto VM minidisks 16
  - restoring with relocation 12
  - using the NV operand 14
- return codes 18
- selective restore 11
- storage and run requirements 26
  - for real mode 26
  - for using OPTIMIZE 26
  - for using the ALLOC R command 26
  - SIZE operand 26
  - space for intermediate data set on disk 26
- using the NV operand 14
- using VSE/Fast Copy online, examples for 27
- utility control statements 7
- when to use VSE/Fast Copy online 20

Fast Copy program, stand-alone

- create a stand-alone FASTCOPY tape 46
- examples for using FASTCOPY 35
- parameters not supported 20
- restoring to VM FBA mini disks 16
- starting FASTCOPY 35
- utility control statements 34
  - operands 34
- when to use VSE/Fast Copy stand-alone 20

FBA disk

- blocking 26
- error handling 18
- using devices with OPTIMIZE 15

file

- copy a 20
- define a 25
- dump a 20
- names of (VSE/Fast Copy online) 23
- reorganization 12
- restore a 20
- specification 25

file descriptor statement, CRDR utility

- format 92
- operands 92

FILE operand (VSE/Fast Copy online) 12

FROM operand (VSE/Fast Copy online) 12

functions (of VSE/Fast Copy)

- copy volume 20
- dump volume 20
- file copy 20
- file dump 20
- file restore 20
- multivolume file 12, 20
- partial volume, dump/copy 20
- restore volume 20
- selective restore 11, 20

## G

GETVIS space for running in real mode 88

## H

handling errors 18

- CKD disks 18
- during migration 64
- FBA disks 18
- return codes 18

handling the recorder file

- ASSGN statement 105
- Record On Demand (ROD) 105
- TLBL statement 105
- using the EREP program for 104

hardcopy file, printing 105

Host Transfer File, batch access to 109

HTF, batch access to 109

## I

ICCF

- parameter for user profiles (in IESUPDCF) 70

ICKDSF/VSE (stand-alone) 38

IESBLDUP migration utility

- adding new users 55
- assignments 54
- CICS/VSE Sign-On Table (DFHSNT) 53
- control statement
  - format 62
  - keyword descriptions 62
- migrating to VSE/ESA 2.4
  - examples 58

- IESBLDUP migration utility (*continued*)
  - precautionary measures 57
  - requirements 56
  - VSAM REPRO command 56
  - migration errors, handling of 64
  - model user profiles 55
  - performing migration 57
  - SPOOL=YES option 54
  - status report 55
  - using 54
  - VSE/ICCF DTSFILE 52
    - reserved VSE/ICCF libraries 53
    - VSE/ESA required VSE/ICCF options 52
- IESUPDCF batch program
  - adding a user ID (ADD statement) 71
    - mandatory parameters for 72
    - optional parameters for 73
    - statement syntax for 72
  - altering a user ID (ALTer statement) 75
    - statement syntax for 75
  - deleting a user ID (DELeTe statement) 76
  - description 69
  - example of completed skeleton IESUPDCF 80
  - planning for user profiles 69
  - preparing skeleton IESUPDCF 69
  - return codes 78
  - setting the ICCF parameter 70
  - skeleton IESUPDCF 76
  - using 76, 78
- INFILE operand (VSE/Fast Copy online) 13
- initialize tape (INTTP utility)
  - job control statements 97
  - purpose 96
  - using 99
  - utility control statement
    - format 97
    - operands 97
  - volume label image statement 98
- input preparation 118
- integrity
  - of system files 3, 16
- intermediate data set 26
  - skipping of 14
  - space requirements on disk 26
  - used with FBA applications for VSE/Fast Copy
    - online 23
- invoking VSE/Fast Copy
  - online 27
  - stand-alone 35
- IV= operand
  - VSE/Fast Copy online 13

## J

- JCC (Job Control Command) ROD 105
- JCL analyzer 86
- Job Control Command (JCC) ROD 105
- Job Control Scanner Outputs 121
- Job Control Scanner Program 117
- job control statements for
  - CLRDK utility 89

- job control statements for (*continued*)
  - CRDR utility 91
  - introduction to 87
  - INTTP utility 97
  - LSERV utility 101
  - VSE/Fast Copy online 23
- job information, printing 108

## L

- label information area, displaying 101
- label information for VSE/Fast Copy online 24
- LIBRARIAN Restore (stand-alone) 37
- LIST operand
  - VSE/Fast Copy 13

## M

- macro SASTCK (stand-alone utilities) 43
- magnetic tapes supported by device utilities 87
- maintaining user profiles
  - via IESUPDCF batch program 69
- messages 124
- messages, Job Control Scanner program 124
- migrating to VSE/ESA 2.4 56
  - backups required 56
- migration utility IESBLDUP 56
  - user profiles 51
- modes, disk compatibility 87
- modifying messages and replies (macro SASTCK) 43
- multibuffering with FBA disk devices
  - run/storage requirements 27
  - using 27
- multivolume file dump/copy/restore 12

## N

- NOEXPIRED operand (VSE/Fast Copy online) 13
- NOLABEL operand 13
- NOPROMPT operand 13
- NOREWIND operand 13
- notations, command 127
- NOVERIFY operand 14
- NOVSAM operand (VSE/Fast Copy online) 10, 14
- NV= operand
  - restrictions concerning use 14

## O

- operands for
  - stand-alone VSE/Fast Copy utility control
    - statements 34
  - VSE/Fast Copy online utility control statements 7, 12
- operating environment 117
- operation under VM (VSE/Fast Copy) 16
- OPTIMIZE
  - description 15
  - storage requirements for using 15
  - values for 26
  - VSE/Fast Copy online 14

- OPTIMIZE (*continued*)
  - with CKD disk devices (VSE/Fast Copy) 14
  - with FBA disk devices (VSE/Fast Copy) 15
- OUTFILE operand (VSE/Fast Copy online) 15
- OV= operand 15
- overview of saving data 3
- overview of stand-alone environments
  - Attention Routine commands 32
  - DITTO/ESA for VSE 40
    - example of a DITTO/ESA run 40
  - functional characteristics 31
  - ICKDSF 38
    - example of an ICKDSF run 39
  - Librarian Restore 37
    - example of a Librarian Restore run 37, 44
  - VSE/Fast Copy 3, 33

## P

- parameters 12
  - description of 12
  - not supported in VSE/Fast Copy stand-alone 20
- parameters for LSERV job control statement 101
- PARSTD= (LSERV parameter) 101
- PARSTD (LSERV parameter) 101
- partial volume, dump/copy 27
- performance considerations for VSE/Fast Copy
  - online 26
    - reading from and writing to an FBA disk 27
    - running in virtual mode 26
- preparing skeleton IESUPDCF 69
- printing job information (LISTLOG) 108
- printing the hardcopy file (PRINTLOG)
  - examples 107
  - options 106
- printing the recorder file
  - ASSGN statement 105
  - Record On Demand (ROD) 105
  - TLBL statement 105
  - using the EREP program for 104
- PRINTLOG (printing the hardcopy file) 105
- protection code (INTTP utility) 96

## R

- real mode
  - run sizes for system utilities 88
  - specifying SIZE value on EXEC statement 88
  - storage requirements for running in 88
- real mode run (VSE/Fast Copy online)
  - performance consideration 26
  - storage requirements for 26
- Record On Demand (ROD) 105
- recorder file
  - handling the 104
  - printing the 104
- recorder file, handling
  - ASSGN statement 105
  - Record On Demand (ROD) 105
  - TLBL statement 105
  - using the EREP program for 104
- REIPL (stand-alone) 31

- relative addressing of
  - DAM files 12
  - SAM files 12
- reorganization, file
  - restriction 12
  - specification for 12
  - summary 12
- REORGANIZE operand (VSE Fast/Copy online) 15
- restore all data 20
- RESTORE utility control statement
  - stand-alone FASTCOPY 34
  - VSE/Fast Copy online 10
- restoring
  - a file 20
  - a multivolume file 12
  - a volume 20
  - selectively 11, 20
  - selectively with relocate 29
  - selectively without relocate 29
  - to VM FBA mini disk (stand-alone FASTCOPY) 16
  - with relocation 12
  - with the ALL function 20
- restrictions for
  - copying to a larger volume 10
  - copying VSAM files 19
  - direct addressing 12
  - dumping VSAM files 20
  - partial volume, dump/copy 9, 10
  - relocating a file 12
  - restoring onto VM minidisks 16
  - restoring SYSRES 14
  - restoring VSAM files 10
  - restoring with relocate 12
  - selective restore 11
  - system files 3, 16
  - using NV operand 14
- return codes
  - for IESUPDCF batch program 78
  - for VSE/Fast Copy online 18
- ROD (Record On Demand) 105
- running VSE/Fast Copy online
  - in real mode 26
  - in virtual mode 26

## S

- SAM file
  - direct addressing 12
  - relocation of 12
- SASTCK macro (stand-alone utilities) 43
- scan run (Job Control Scanner program), starting 123
- selective restore 11
- SET operation (DTRSETP utility) 82
- SETPARM procedures (DTRSETP utility) 81
- setting ICCF parameter (in IESUPDCF) 70
- SIZE value for EXEC statement 88
- skeleton
  - IESUPDCF 69
    - example of 76
    - preparing 69
- stand-alone environment (overview) 31

- stand-alone environment (overview) 31 (*continued*)
  - functional characteristics 31
- stand-alone tape 32
  - performing IPL from stand-alone tape 32
- stand-alone utilities
  - customizing 41
- starting
  - stand-alone FASTCOPY 35
  - VSE/Fast Copy online 27
- starting a scan run 123
- STDLABEL (LSERV parameter) 101
- storage requirements for device utilities
  - in real mode 88
  - in virtual mode 88
- storage requirements for VSE/Fast Copy online 26
- support for devices
  - disks 87
  - magnetic tapes 87
- support return codes for VSE/Fast Copy online 18
- syntax, of commands 127
- syntax symbols 127
- SYSIPT, assigning 88
- SYSLST, Job Control Scanner output to 119
- SYSPCH, Job Control Scanner output to 121
- system files, integrity of 3, 16

## T

- TLBL statement 23, 105
- TO operand 16
- type 1/2/3 user profiles 69

## U

- UIN file name 23
- UOUT file name 23
- UPSI statement (for label setting in VSE/Fast Copy online) 23
- user profile types 69
- user profiles
  - adding a user ID 71
  - altering a user ID 75
  - deleting a user ID 76
  - migrating 51
  - planning 69
  - skeleton IESUPDCF 76
  - types of 69
- using
  - CLRDK utility 89
  - CRDR utility 93
  - Display VTOC dialog 103
  - DTRIINIT utility 84
  - DTRSETP utility 81
  - EREP program 104
  - IESBLDUP migration utility 54
  - IESUPDCF batch program 78
  - INTTP utility 99
  - JCC command ROD 105
  - LISTLOG utility 108
  - LSERV program 101
  - LVTOC program 103
  - PRINTLOG utility 105

- using (*continued*)
  - SETPARM procedures 81
  - stand-alone FASTCOPY 35
  - VSE/Fast Copy online 27
- utilities, device
  - disk compatibility modes 87
  - storage requirements 88
- utility control statements
  - CLRDK utility 89
  - CRDR utility 91
  - introduction to 87
  - INTTP utility 97
  - stand-alone FASTCOPY 33
  - VSE/Fast Copy online 12

## V

- values for
  - CISIZE 25
  - OPTIMIZE parameter 26
- virtual mode
  - specifying SIZE=AUTO 88
  - storage requirements for running in 88
- virtual mode run for VSE/Fast Copy online 26
- VM FBA mini disks, restore to (stand-alone FASTCOPY) 16
- volume copy
  - description 20
  - VSAM restriction 8
- volume dump
  - description 20
  - VSAM restriction 19
- volume label image statement for the INTTP utility 98
- VOLUME operand
  - VSE/Fast Copy (online, stand-alone) 16
- volume restore
  - description 20
  - VSAM restriction 9
- Volume Table of Contents (VTOC), displaying 102
- VSAM file restrictions for
  - copy, dump, restore volume 10
  - selective restore 11
- VSAM REPRO command 56
- VTOC (Volume Table of Contents), displaying 102

## W

- WAIT operation (DTRSETP utility) 82





File Number: S370/S390-34  
Program Number: 5690-VSE



Printed in the United States of America  
on recycled paper containing 10%  
recovered post-consumer fiber.

SC33-6717-00

