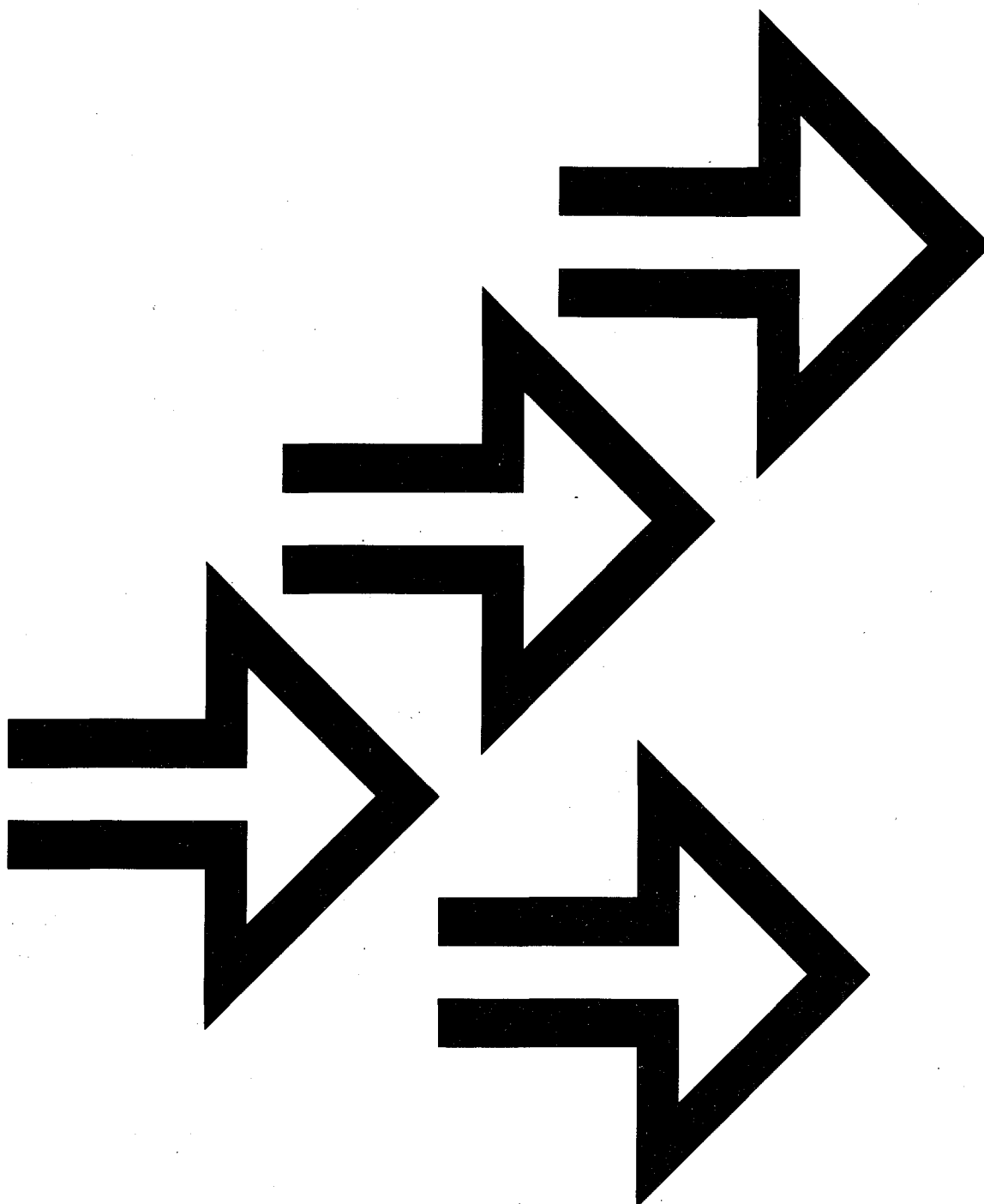




Document Library Facility Guide

Program
Product

SH20-9165-5



Program Number 5748-XXE

Release 3

This publication was produced using the IBM Document
Composition Facility (program number 5748-XX9).
The master was printed on the IBM 3800 Printing Subsystem Model 3.

Sixth Edition (June 1985)

This book applies to Release 3 of the IBM Document Library Facility, Program Product 5748-XXE, that runs under OS/VS2 MVS S.P. Release 1.3.0, VSE with VSE/AF 3.1.x or VSE/AF 2.1.0 and to any subsequent releases unless otherwise indicated in new editions or technical newsletters.

This is a major revision of SH20-9165-4, which is now obsolete.

The changes for this edition are summarized under "Summary of Amendments" in the preface of this book.

Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest IBM System/370 and 4300 Processors Bibliography, GC20-0001, for the editions that are applicable and current.

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 program product in this publication is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

Publications are not stocked at the address given below. Requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Information Products Division, Box 1900, Department 580, Boulder, Colorado, U.S.A. 80301. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

(c) Copyright International Business Machines Corporation 1978, 1980, 1981, 1982, 1984, 1985

PREFACE

This publication describes the functional characteristics and operation of the Document Library Facility (DLF) Release 3 that runs under OS/VS2 MVS S.P. Release 1.3.0 and VSE with VSE/Advanced Function Release 3.

This manual addresses two types of users, general users and administrators. When applicable, the terms user or library user will represent both the general user and the administrator. The general users should understand basic data processing terminology, text processing, and data storage concepts. Administrators, in addition to the preceding, should know how to create and maintain the document library using standard data management facilities of the above operating systems.

This publication is both a guide and a reference manual for administrators and general users. With the aid of this guide, general users can properly store and control documents assigned to them. Administrators can set up the document library (the area in which active documents are stored), define library users to DLF, and manage storage allocation.

ORGANIZATION OF THIS MANUAL

This manual is divided into four parts: Introduction, The General User, The Administrator, and Additional Uses of DLF.

- Part 1, "Introduction," introduces DLF functions, structure, and security. Syntax conventions for DLF commands and required JCL are also introduced.
- Part 2, "The General User," lists alphabetically each general user command and explains its function. This part offers examples of each general user command.
- Part 3, "The Administrator," lists alphabetically each administrator command and explains its function. Setting up and administering the document library are discussed in this part.
- Part 4, "Additional Uses of DLF," includes explanations of using DLF to call processors and using DLF as a subroutine. This part also includes user exit routines.

The term processor, as used in this Document Library Facility Guide, refers to a non-DLF program that can be called by the library program to perform special processing for the DLF commands IMPORT, EXPORT, SCRIPT, and READ.

The eight appendixes include:

- Appendix A, "Converting from Release 1 to Release 3," describes the process of migrating the library contents of DLF Release 1 to DLF Release 3. No conversion from Release 2 to Release 3 is required.
- Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics," summarizes the job control language (JCL) statements necessary to invoke DLF, describes how external data sets can be dynamically allocated, and specifies default data set characteristics of nonlibrary data sets used by DLF.
- Appendix C, "DCF Formatter (SCRIPT/VS) and DLF Interface," describes the parameter list that SCRIPT/VS and DLF share.

- Appendix D, "Format of Sequential Data Set for COPY OUT," summarizes the content, storage, and record format of data sets created by the COPY OUT command.
- Appendix E, "Format of the Accounting Record," is a figure showing the offset, field size, data format, and content of a System Management Facility (SMF) type 47 record.
- Appendix F, "Output of LIST Commands," shows examples of how DLF users can list their documents.
- Appendix G, "Relationships of ATMS-III and the Document Library Facility," presents the concept of archiving ATMS-III documents in DLF.

PREREQUISITE PUBLICATIONS

FOR LIBRARY USERS

The messages for DLF are supplied in printed form, SH35-0049.

FOR ADMINISTRATORS

To include VSAM in the operating system you use, read the publications that apply:

- OS/VS2 System Programming Library: System Generation Reference, GC26-3792.
- VSE/Advanced Functions System Generation, SC33-6096.
- Using VSE/VSAM Commands and Macros, SC24-5144.
- OS/VS2 Access Method Services, GC26-3841.
- VSE/VSAM Programmer's Reference, SC24-5145.
- VSE/VSAM Messages and Codes, SC24-5146.

The following publication describes the Authorized Program Facility:

- OS/VS2 System Programming Library: Supervisor, GC28-1046.

RELATED PUBLICATIONS

FOR ALL LIBRARY USERS

For information about the Document Composition Facility (DCF) formatter (SCRIPT/VS) read the following publications:

- Document Composition Facility: SCRIPT/VS Text Programmer's Guide, SH35-0069, describes SCRIPT/VS control words, symbols, and macro facilities and how to use them to format documents.
- Document Composition Facility: SCRIPT/VS Language Reference, SH35-0070, lists SCRIPT/VS commands and control words and gives installation guidelines and 3800 line printer fonts available with SCRIPT/VS.
- Document Composition Facility: Generalized Markup Language Starter Set User's Guide, SH20-9186, describes the Generalized Markup Language (GML) and how to use it. Particular emphasis is placed on the starter set of GML, provided with DCF to give users a starting point in using GML and developing their own GML to meet their specific needs.

- Document Composition Facility: Generalized Markup Language Starter Set Reference, SH20-9187, describes the GML starter set provided with the Document Composition Facility (DCF), provides information on using SCRIPT/VS to process documents with GML tags, and explains tailoring the starter set for the requirements of your data-processing center.
- Document Composition Facility Generalized Markup Language: Concepts and Design Guide, SH20-9188, introduces GML concepts, describes GML formatting symbols, and provides guidelines for customizing GML.
- Document Composition Facility Messages, SH35-0048, gives an explanation for and response to the messages DCF issues.

Additional information about the Document Library Facility and the Document Composition Facility can be found in the following publication.

- Document Composition Facility and Document Library Facility: General Information, GH20-9158, introduces DCF, a text formatting and processing system, and gives a general description of GML. The manual also introduces DLF, a program that stores and controls access to many types of data. Programming requirements and storage estimates for these products are described.

You can find information about using the Composed Document Printing Facility (CPDF - Program Number 5668-997) to produce 4250 Printer output with the DLF SCRIPT command in this publication:

- Composed Document Printing Facility Installation and Operation, SC33-6135.

In the VSE environment, the following publications contain information that is needed when DLF is called as a subroutine:

- VSE/Advanced Functions Macro User's Guide, SC24-5210, contains information about the use of VSE macro instructions.
- VSE/Advanced Functions Macro Reference, SC24-5211, contains reference material about VSE macro instructions.

FOR ADMINISTRATORS

For valid route codes for the ROUTE keyword of the DEFINE SYSTEM and ALTER SYSTEM commands, see the following manual:

- OS/VS2 MVS Supervisor Services and Macro Instructions, GC28-0683. (OS/VS2 Release 3.7)
- OS/VS2 MVS Supervisor Services and Macro Instructions, GC28-1114. (OS/VS2 Release 3.8)

If you use Resource Access Control Facility (RACF) protection, the following manuals provide information:

- Resource Access Control Facility (RACF) General Information Manual, GC28-0722.
- OS/VS2 MVS Resource Access Control Facility (RACF) Installation Reference Manual, SC28-0734.
- OS/VS2 MVS Resource Access Control Facility (RACF) Command Language Reference, SC28-0733.
- OS/VS2 MVS Resource Access Control Facility (RACF) Messages and Codes, SC38-1014.

When accounting information is to be processed, use the following publication:

- OS/VS2 System Programming Library: System Management Facilities (SMF), GC28-0706.

For additional VSAM information, see the following manuals:

- OS/VS2 Virtual Storage Access Method (VSAM) Programmer's Guide, GC26-3838.
- Planning For Enhanced VSAM Under OS/VS, GC26-3842.

RESTRICTED PUBLICATIONS

- Document Library Facility Diagnosis Guide, SY35-0071, and Document Library Facility Diagnosis Reference, LY35-0072, provide information for customers and IBM service personnel who diagnose programming errors.

SUMMARY OF AMENDMENTS

SIXTH EDITION

Significant changes in this edition are:

- The discussion of NAMELIST entries, formerly an appendix in this book, was added to the ARCHIVE NAMELIST and RETRIEVE NAMELIST command discussions.
- New examples were added showing how to use the DLF SCRIPT command to format and print documents on page printers in a batch environment (including the 3820 Page Printer).
- The information about protecting documents and DLF users with the Resource Access Control Facility (RACF) was revised.
- The addition of information pertinent to the advent of two Small Program Enhancements (SPEs):
 1. The first involves changes to the DLF LIST commands so that users can specify the document library (input or output) from which to list information and the addition of a LIST SYSTEM command.
 2. The second SPE provides a new operand pair (called SPACE/NOSPACE) for the LIST DOCUMENT, and LIST CLUSTER commands so users can produce Space Usage reports. This SPE also adds the CLUSTER(name) operand to the LIST DOCUMENT command.
- JCL changes for VSE/AF 2.1.0.

FIFTH EDITION

Significant changes in this edition of Release 3 are listed below:

- Additional completion information can be returned to the calling program when DLF is called as a subroutine. This information includes a reason code if an error condition is detected and an indication of document version number if the requested function read in a document via the IMPORT command.
- Additional devices can be specified on the ENVIRONMENT command:
 - The 3375 and DASD (generic) are supported for both the library and external files.
 - 3350 support has been extended to external files in the VSE environment.
 - TAPE (generic) support has been added to external files in the VSE environment.
 - The 3310 and 3370 FBA devices are supported for external files in the VSE environment.

Note: For FBA devices, a CISIZE parameter is necessary for SAM support. The Document Library Facility provides CISIZE=18432 on the DTFs used. This is the correct size for the ARCHIVE and COPY OUT commands. For the EXPORT command, this can be overridden on the DLBL JCL statement if desired.

- General users can specify a range of user numbers on the LIST DOCUMENT command.

FOURTH EDITION

Significant changes in this edition of Release 2 are as follows:

- Clarifications have been made to the descriptions of the following general user commands:

ARCHIVE ALL
ARCHIVE DOCUMENT
COPY
COPY DOCUMENT
COPY IN
COPY OUT
LIST DOCUMENT
RETRIEVE ALL
RETRIEVE DOCUMENT
SCRIPT.

- Several sample problems have been clarified and expanded.
- Clarifications have been made to the descriptions of the following administrator commands:

ACCOUNT
ALTER MAP
ALTER USER
ARCHIVE ALL
COPY
COPY IN
COPY OUT
LIST DOCUMENT
RETRIEVE ALL.

CONTENTS

Part 1. Introduction	1
Chapter 1. Document Library Facility Concepts and Conventions	3
Overview of DLF	3
Overview of the Document Library Structure	3
Directory Data Set	4
Source Data Sets	4
Storing Data in the Document Library	5
Accessing Data in the Document Library	5
Private Libraries	5
Project Libraries	5
Public Libraries	6
Example of a Simple Document Library	7
Types of Users	8
General Users	8
Administrators	9
Documents	9
Document Attributes	10
Class	10
Data Type	10
Source	10
Overview of Data Security	11
VSAM Password Protection	11
Identification of Valid Users	11
Controlled Access of Valid Users	11
Document Protection	12
Chapter 2. JCL and Command Syntax Conventions	13
JCL Statements	13
Authorizing Access to the Document Library	15
Syntax and Conventions	16
Command Structure	16
Placement	17
Abbreviations	17
Separators	17
Continuation	17
Comments	18
Quoted Strings	18
Notational Conventions	18
Commonly Used Command Operands	19
Variable Parameter Operands	19
Keyword Operands	21
Part 2. The General User	25
Chapter 3. General User Commands	27
ALTER DOCUMENT	29
Examples	30
ARCHIVE	31
ARCHIVE ALL	32
Example	33
ARCHIVE DOCUMENT	34
Example	35
ARCHIVE NAMELIST	36
Creating a Namelist for the ARCHIVE Command	37
Namelist Format	37
Examples	38
AUTH	39
Example	39
CLOSE	40
COPY	41
Preparing to Execute a COPY Command	41
COPY DOCUMENT	42
Ownership of A Copied Document	42
Examples	45
COPY IN	46

Example	47
COPY OUT	48
Example	49
ENVIRONMENT	50
Examples	52
EXPORT	54
Examples	56
IMPORT	58
Examples	65
LIST	67
LIST CLASS	68
Example	68
LIST CLUSTER	69
Examples	70
LIST DOCUMENT	71
Examples	75
LIST PROCESS	76
Example	76
LIST USER	77
Examples	78
PASSWORD	79
Examples	79
PROTECT	80
Examples	81
PURGE	82
PURGE DOCUMENT	82
Example	83
READ	84
Example	86
RETRIEVE	87
RETRIEVE ALL	87
Example	88
RETRIEVE DOCUMENT	89
Example	90
RETRIEVE NAMELIST	91
Creating a Namelist for the RETRIEVE Command	92
Namelist Format	92
Examples	93
SCRIPT	94
Examples	98
 Chapter 4. Using the Document Library	 101
Putting Documents into Your Own Library	101
Producing Copies of a Single Document	105
Backing Up Documents and Copying Another DLF Library	107
Purging Documents and Getting Lists of Document Names	108
Formatting a Document with SCRIPT/VS	110
Processing ATMS Documents	115
Archiving and Retrieving Your Documents and their Versions	119
Copying Versions of Documents	121
Changing Document Names and Attributes	123
 Part 3. The Administrator	 125
 Chapter 5. Administrator Commands	 127
ACCOUNT	129
Examples	131
ALTER	132
ALTER CLASS	133
Example	133
ALTER CLUSTER	134
Example	135
ALTER DOCUMENT	136
Examples	137
ALTER MAP	138
Example	139
ALTER PROCESS	140
Example	141
ALTER SYSTEM	142
Example	142
ALTER USER	143
Examples	145
ARCHIVE ALL	146
Examples	148

COPY	149
Preparing to Execute a Copy Command	149
Administrator Capabilities	150
COPY IN	151
Examples	153
COPY LIBRARY	154
Example	155
COPY OUT	156
Examples	158
COPY USER	159
Example	159
DEFINE	160
DEFINE CLASS	161
Example	161
DEFINE CLUSTER	162
Example	163
DEFINE MAP	164
Examples	164
DEFINE PROCESS	165
Example	166
DEFINE SYSTEM	167
Example	168
DEFINE USER	169
Examples	171
LIST	172
LIST DOCUMENT	173
Example	177
LIST MAP	178
Example	178
LIST SYSTEM	179
Example	179
LIST USER	180
Example	181
PROTECT	182
Examples	183
PURGE	184
PURGE CLASS	185
Example	185
PURGE CLUSTER	186
Example	186
PURGE DOCUMENT	187
Example	188
PURGE MAP	189
Example	189
PURGE PROCESS	190
Example	190
PURGE USER	191
Examples	191
RETRIEVE	192
RETRIEVE ALL	192
Example	193
 Chapter 6. Setting Up the Document Library	 195
Defining the VSAM Data Sets	195
Estimating Space Requirements	196
Authorized Program Facility	198
Installing DLF	198
Installing Processors	200
Defining Required Profiles	201
PROVIDING RACF SECURITY FOR DATA AND USERS	203
Protecting DLF Libraries with RACF	203
Protecting DLF User Numbers with RACF	204
 Chapter 7. Administering the Document Library	 205
Authorizing New Users	205
Copying Profiles and Documents	206
Ensuring Adequate Space For the Document Library	206
Listing Library Space Usage	206
Deleting Obsolete Documents	208
Deleting a User Profile	208
Allocating Additional Space in the Document Library	208
Reorganizing the Document Library	210
Reorganizing the Directory	211
Reorganizing an ESDS Source Cluster	211

Modifying an Attribute Profile	211
Defining User Maps for External Systems	212
Defining Classes of Data	212
Part 4. Additional Uses of the Document Library Facility	213
Chapter 8. Using DLF to Call Processors	215
Input to Processor	215
Processor Activities	216
How a Processor Returns to DLF	216
Calling Sequence Between DLF and a Processor	216
Register Conventions	216
Parameter Area	217
Messages	218
The Communication Buffer	219
Examples	219
Chapter 9. Using DLF as a Subroutine	221
Calling DLF From an Application Program	221
Example of Use	223
SCRIPT FROM(*) - Input Data from Another Program	223
Example of Use	224
Chapter 10. Writing User Exit Routines	225
Uses for Exit Routines	225
Taking an Exit	225
Calling Sequence Between DLF and a User Exit Routine	228
Register Conventions	228
Parameter Area	229
Flags	229
Return Codes	230
Messages	230
Exits at Each Command	231
ARCHIVE User Exit	231
DEFINE MAP and ALTER MAP User Exits	232
DEFINE USER AND ALTER USER User Exits	232
EXPORT User Exit	232
IMPORT User Exit	232
PURGE DOCUMENT User Exit	232
PURGE MAP User Exit	232
PURGE USER User Exit	232
RETRIEVE User Exit	232
Appendix A. Converting from Release 1 to Release 3	235
Converting a Document Library	235
Conversion Process	235
Administrators	235
General Users	235
Sequencing Data	235
DATE keyword	236
Appendix B. JCL Statements, Dynamic Allocation, and Data Set Characteristics	237
JCL Requirements	237
Data Set Naming for Dynamic Allocation	239
Default Data Set Characteristics	239
Appendix C. DCF Formatter (SCRIPT/VS) and DLF Interface	243
SCRIPT/VS Formatter Requirements	243
Read Module	243
Additional Information for OS/VS2	245
Additional Information for VSE	246
Appendix D. Format of Sequential Data Set for Copy out .	249
Appendix E. Format of the Accounting Record	251
Appendix F. Output of LIST Commands	253
Appendix G. Relationship Of ATMS-III And The Document Library Facility	259
Glossary of Terms and Abbreviations	261

Index	267
--------------	------------

FIGURES

1.	DLF Document Library Structure	7
2.	Predefined ddnames	14
3.	Sample JCL Procedures for DLF	16
4.	Summary of the General User Commands	27
5.	Flowchart of how DLF tests version characteristics.	61
6.	Examples of the IMPORT Command	102
7.	IMPORT Command Using the FROM(*) Keyword	104
8.	Examples of the EXPORT Command	105
9.	Examples of the COPY OUT and COPY DOCUMENT Commands	107
10.	Examples of the PURGE DOCUMENT and LIST DOCUMENT Commands	109
11.	Example of the JCL statements and SCRIPT Command for 3800 Model 1	111
12.	Example of the JCL statements and SCRIPT Command for IBM 4250 printer	112
13.	Example of the JCL statements and SCRIPT Command for IBM 3800 Model 3	113
14.	Example of the JCL and JECL statements and SCRIPT Command for IBM 3820 Page Printer	114
15.	The SCRIPT/VS Utility Data Sets	115
16.	Examples of ATMS Conversion Processing	117
17.	Examples of the ARCHIVE DOCUMENT Command	120
18.	Examples of the COPY DOCUMENT Command	122
19.	Examples of the ALTER DOCUMENT Command	123
20.	Summary of the Administrator Commands	127
21.	VSAM Parameters for the Document Library Data Sets	196
22.	Examples of Defining Document Library Data Sets (OS/VS2)	199
23.	Example of Defining Document Library Data Sets (VSE)	200
24.	Examples of Defining Minimum Profiles	202
25.	Examples of Allocating and Formatting Additional Library Space	209
26.	Examples of Altering an Attribute Profile	212
27.	Processor Return and Reason Codes	217
28.	The Format of the User Mapping Record (UMR)	226
29.	The Format of the Document Entry Record (DER)	227
30.	The Format of the User Profile Record (UPR)	228
31.	The Parameter Area of the User Exit Routine	229
32.	JCL Data Set Names	237
33.	OS/VS2 Default Data Set Characteristics	240
34.	VSE Default Data Set Characteristics	241
35.	Read Module Parameter List	244
36.	Format of the Accounting Record	251
37.	Example of output from LIST commands	254

PART 1. INTRODUCTION

Part 1 of the Document Library Facility Guide provides an overview of the product through the following chapters:

- Chapter 1 "Document Library Facility Concepts and Conventions" — contains an overview of library structure, its users, and data security features.
- Chapter 2 "JCL and Command Syntax Conventions" — contains information about DLF-related JCL, authorizing access to the document library, and commonly used command operands.

CHAPTER 1. DOCUMENT LIBRARY FACILITY CONCEPTS AND CONVENTIONS

OVERVIEW OF DLF

The Document Library Facility (DLF) is a program that stores and accesses many types of data in a protected environment. All discrete units of data are called documents and are stored in the document library, an area of auxiliary storage made up of virtual sequential access method (VSAM) data sets. DLF controls access to these data sets with library user numbers and document classifications.

If the Document Composition Facility (DCF) formatter, SCRIPT/VS, is installed, DCF (in a background environment) can format documents from the document library or from an external sequential data set.

DLF can be executed as a batch job or called as a subroutine of a user-written program. It operates in the OS/VS2 and VSE environments.

The basic DLF capabilities are:

- Storing data (primarily text documents).
- Protecting data with security levels.
- Maintaining and managing different versions of a document.
- Archiving (by user commands) inactive documents and their descriptions while retaining their corresponding index information in the directory data set.
- Copying and backing up single documents, all of the documents in a library user's library, or those in a complete document library.
- When DCF is installed, formatting documents in a background environment using SCRIPT/VS.
- When DCF is installed, converting to SCRIPT/VS controls the text-processing controls of the Advanced Text Management System III (ATMS-III) to process documents using SCRIPT/VS.
- With user-written conversion programs, converting other text-processing controls and data formats into SCRIPT/VS controls for formatting in a batch job.
- With user-written application programs, calling DLF as a subroutine. All DLF functions are available through this interface.

OVERVIEW OF THE DOCUMENT LIBRARY STRUCTURE

DLF manages the storage, use, and accessibility of various data. These data, also called documents, are stored in virtual sequential access method (VSAM) data sets, which make up the document library. Two kinds of VSAM data sets are used: (1) the directory data set, which is a series of descriptive records (profile records) that states the kinds of information being stored in the document library, and (2) the source data sets that store all the text documents and their respective versions.

Note: This overview chapter examines the structure of a single document library. However, DLF can create as many document libraries as a customer chooses to define. In all

cases, document libraries will have the same characteristics as the one described in this publication.

DIRECTORY DATA SET

The directory data set consists of descriptive records that provide managerial information for the document library. This information resides in a VSAM key-sequenced data set (KSDS) of descriptive profiles for all aspects of document library operation. The directory data set supports the entire storage system and must be defined before the document library is used. The information contained in these records is continuously updated, so library users have an accurate picture of the document library's status at all times.

The directory data set includes the following profile records:

- A system profile record (SPR) defining the document library itself
- A user profile record (UPR) for each library user that defines each user's library user number, password, access and function limitations, default characteristics, and library space allocation and usage
- Cluster profile records (CPRs) describing VSAM entry-sequenced data sets (ESDS) of various control-interval sizes for the storage of documents
- Class entry records (CERs) defining the categories of documents contained in the document library, such as books, letters, and memoranda and, optionally, assigning those categories to a specific cluster for storage
- Document entry records (DERs) specifying each document in the document library and its corresponding description (such as the document owner, password, attributes)
- User mapping records (UMRs) that can associate valid library users with any names associated with those library users in an external operating system
- Library entry records (LERs) each indicating the number of documents a library user has placed in a library owned by another library user (and indicating that library number)
- A processor profile record (PPR) for each translating program used to process data stored in the library.

SOURCE DATA SETS

The source data sets contain all the active documents that are stored in the document library. These data sets are VSAM entry-sequenced data sets. Initially, as new documents are placed into the source data sets, they are added after any existing documents in the space (called free space) at the end of the data in the data set. After the end of the data set has been reached and unused documents have been removed from the data set, new documents can be added in the space that is made available. At least one ESDS with a control-interval size of 4K bytes (K equals 1 024) and the DLF cluster name DSMPTLIB must be defined before a document library can be used. Control intervals are fixed length areas of direct access storage in which VSAM stores records and distributes free space. A control interval is roughly equivalent to a physical record in a data set.

The means by which DLF allocates documents to storage in the following two sections.

STORING DATA IN THE DOCUMENT LIBRARY

DLF uses storage space by recognizing each ESDS as a cluster that is to be used to store documents of the same general size. A cluster can be thought of as a storage area containing VSAM control intervals of the same size. Only one document can reside in a control interval of a cluster. If a document is larger than the control interval, it continues into the next control interval. If a document does not fill a control interval, the unused space remains empty.

Control interval storage capacity is allocated when the cluster is initially defined (see "Source Data Set (Entry-Sequenced)" in Chapter 6). A document can be assigned to a particular cluster on the basis of its class attribute (an identifier that can be used to describe a document's size or form, such as book, letter, memo). A user can cause the proper cluster to be used for storage of a document by specifying a CLASS name when the document is imported, in other words, stored in the library.

Although an installation can define any type of connection between a class and a cluster, the most efficient relationships are those of a cluster with large control intervals for books, and a cluster with smaller control intervals for documents such as letters and memoranda. In this way, very little storage space is unused.

Before a document library can be used, it must contain at least one source data set (cluster). The initial ESDS defined must be the default cluster with a control interval size (CISIZE) of 4K bytes.

ACCESSING DATA IN THE DOCUMENT LIBRARY

The document library contains three types of libraries with differing degrees of accessibility: private, project, and public.

Private Libraries

A private library represents the storage space allocated to only one user. The private-library owner can store, read, copy, or delete documents in this library. An administrator can read, copy, or delete documents in a private library belonging to another user, but cannot store data there. All of the documents in a private library are owned by the library user who has the corresponding library user number. Therefore, if library 123 and library 234 are private libraries, library user 123 will own library 123 (and all documents therein), and library user 234 will own library 234 (and all documents therein).

Project Libraries

A project library represents the storage space owned by one library user which contains documents owned and accessible by a group of library users known as project members. The project-library owner can store, read, copy, or delete any documents in this library.

A project member will have his own library user number but will be given access to a project library when his user profile record (UPR) is defined. A library user can belong to only one project library at a time. A project library member can access any document in the project library assigned regardless of ownership of the document.

Project members can store documents in the project library and delete the documents they personally own from that library. A project member can add documents to be owned by another member of the project library; however, he cannot delete a document

unless he is the owner of that document, an administrator, or the owner of the project library.

Controlled Library

If the project library is controlled (access is limited), only the project library owner or an administrator can delete documents from that library, and only the library owner can store documents in that project library.

Public Libraries

A public library represents the storage space owned by one user which contains documents owned or accessible by any user of DLF. The public-library owner can store, read, copy, or delete documents in this library.

All valid library users have access to public libraries. Like members of a project library, public-library users can put documents into the library or delete the documents they own. A library user can access any document in any public library even if he does not own that document.

Controlled Library

If the public library is controlled, then only the public-library owner or an administrator can delete documents, and only the library owner can store documents in that library.

DLF DOCUMENT LIBRARY STRUCTURE

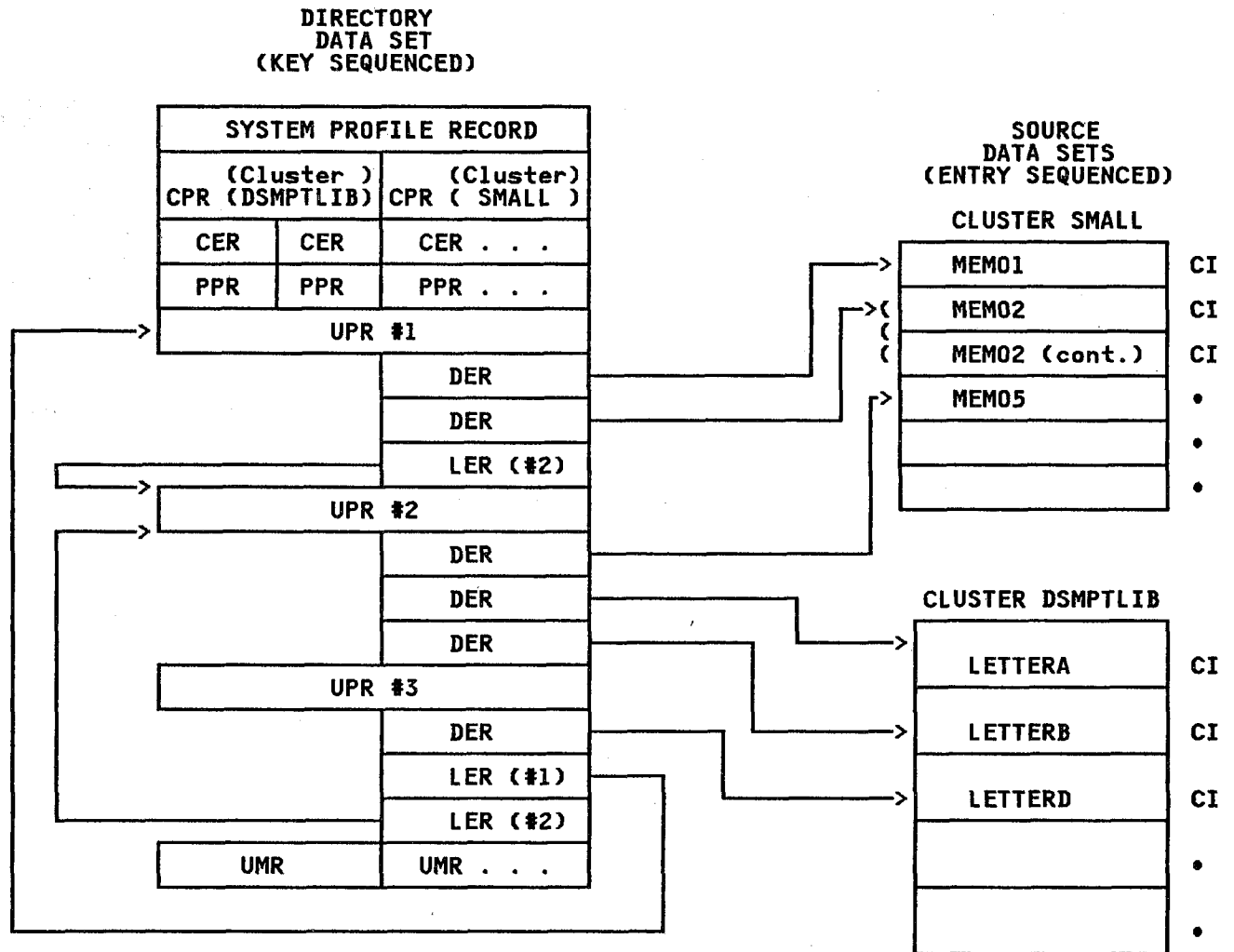


Figure 1. DLF Document Library Structure

EXAMPLE OF A SIMPLE DOCUMENT LIBRARY

Figure 1 is a simplified drawing of a single DLF document library. It shows only three users and two clusters in order to make it easier to understand the basic structure of a document library. Please use this drawing as an aid and not as a total picture.

In Figure 1, one user (UPR #1) owns a public library that has two documents (two DER's) stored in it. The third user owns one or more of these documents. The second user also owns a public library and may own one or none of the documents in his library.

The drawing shows the third user as having a private library in which he owns one document (pointed to by the DER). This third user also owns other documents which are stored in the public libraries owned by user one and user two, respectively (referred to by the two LER's).

Notes:

1. A document can fill more than one control interval (CI) (such as MEM02 in Figure 1 on page 7); but one control interval cannot hold more than one document. The document does not have to fill the entire control interval.
2. DSMPTLIB is the required basic cluster that was created with the document library. The size of each of DSMPTLIB's control intervals must be 4K bytes.
3. There is one cluster profile record (CPR) for each cluster in the document library.
4. There can be many users and their associated user profile records (UPRs) in a document library.
5. See "Directory Data Set" on page 4 for descriptions of the SPR, CER, PPR, and UMR record types referred to in Figure 1.

TYPES OF USERS

DLF has two types of library users: general users and administrators. General users control only the documents and the library they own. Administrators manage many of the document library operations and control other library users as well as their own documents and library.

General Users

A general user can be assigned ownership of a public, a project, or a private library. A general user is identified by the library user number that corresponds to his owned library. A general user is given read-only access to other library users' documents in a project library (if assigned to one), to all documents in all public libraries, and to any documents with share status residing in another user's library.

A general user can store a document in the library he owns, in a noncontrolled project library to which he is assigned, or in any noncontrolled public library. A general user cannot store or delete data in a controlled library unless he is the owner of that library.

The general user cannot perform any administrative operations. Because of this restriction, this manual separates discussion of general user commands and tasks from discussion of administrative duties.

With DLF you, as a general user, can perform the following tasks:

- Put documents into your own library
- Produce copies of a document
- Back up your documents and copy them into another document library
- Purge documents
- Get lists of document names
- Format a document with SCRIPT/VS
- Process ATMS documents

- Archive documents to and retrieve documents from external data sets
- Copy versions of documents
- Change document names and attributes.

Administrators

An administrator can own a public, a project, or a private library and can control his library and documents in the same manner as the general user does. A document library can have more than one administrator.

In addition, administrators have the authority to read any document in the document library, regardless of ownership. They can also copy documents into any private, project, or public library. However, an administrator cannot use the IMPORT command to store a document into another user's private library. An administrator can alter, list, or delete documents from any library or delete users from the document library.

Using the applicable commands, an administrator can:

- Define a new document library (DEFINE SYSTEM command). This is done when DLF is first used against a set of VSAM data sets that are to make up a document library.
- Define a new document library user, including himself when DLF is first used with this document library (DEFINE USER command).
- Alter the profiles of existing library users (ALTER USER command).
- Copy any library user profiles, descriptive records, or documents (COPY commands).
- List any library user profiles, descriptive records, or documents (LIST commands).
- Produce accounting statistics relating to library space usage (ACCOUNT command).

DOCUMENTS

The texts and other discrete units of data stored in the document library are called documents. As the library user imports a document, a document entry record (DER) is created and stored in the directory data set. The DER includes the following information:

- Number of the library in which the document is stored
- Document name
- Document owner
- Document password (if assigned)
- Date of entry
- Location in the source data set

- Version of the document
- Document attributes:
 - class
 - data type
 - source.

Note: The library number, the document name, and the data type uniquely identify a document. The document version number identifies a specific version of such a uniquely identified document.

DOCUMENT ATTRIBUTES

To help identify a document or prepare it for processing, DLF assigns attributes. The three general types of attributes are class, data type, and source.

Class

A class attribute is an optional name that typically describes the size of a document. Although class can describe size in terms of form (such as book, letter, memo), it can also be named as the user chooses.

Through the class attribute, a library user can cause DLF to allocate storage space best suited to the document by matching it with the appropriate cluster. A cluster profile record (CPR) defines a VSAM ESDS cluster with a particular control interval (CI) size. A class entry record (CER) defines the cluster associated with a particular class attribute. Through this matching process, the library user can make the most efficient possible use of storage space. For example, a class named memo would best be suited to a cluster that accommodates small documents.

Data Type

Data type joins other qualifiers such as document name and library number to make the document unique. The data type attribute usually indicates the type of formatting controls contained within a document. As used here, controls are instructions in the text of a document that specify how that document is to be formatted. These instructions are interpreted by a text-formatting program (ATMS formatter or SCRIPT/VS, for example). The same data type must be assigned to all versions of a document.

Source

The source attribute typically refers to the system on which a document was created, but it can also be user-defined. This information is passed using the calling sequence to a DLF processor (when used) and can therefore cause processing unique for that source. A source attribute can also help to identify documents in a document list. TS0, ATMS, CMS, and VM are all examples of external source attributes.

OVERVIEW OF DATA SECURITY

Library users can use four methods of data security to protect their documents and libraries:

- VSAM password protection
- Identification of valid users
- Controlled access of valid users
- Document protection.

VSAM PASSWORD PROTECTION

When you use this method of security, the document library can be protected by VSAM password controls. In this way, only those who know the password can invoke the library program to access the document library. Detailed information about specifying VSAM passwords is in the Access Method Services manuals cited under "Prerequisite Publications" in the preface to this Guide.

IDENTIFICATION OF VALID USERS

With this method of security, you use the profiles of valid users in the directory data set to control access to the document library. For each DLF operation, the library user must first execute the AUTH (authorization) command to identify himself as a valid user. Upon recognizing the library user number, DLF gives access to this library user. A library user's authority to use the document library remains current until invalidated by an administrator. An administrator cannot invalidate himself. This ensures that there will always be at least one administrator authorized for the document library.

In addition to this form of control, user passwords provide protection. When an administrator defines a user's profile, a password can be assigned to the user. If a user password is designated, access to the library will not be given unless the user specifies the password in the AUTH command. An authorized user can change or cancel his own password at any time.

In OS/VS2, the Resource Access Control Facility (RACF), program product number 5740-XXH, provides additional security. This protection can include VSAM data sets and input/output data sets used with the document library. DLF can use RACF to protect its users by restricting user access.

CONTROLLED ACCESS OF VALID USERS

The document library structure controls the access by valid library users to other users' documents. A general user can access documents only from his own library (the one corresponding to his library user number), a project library assigned to him in his user profile record, any document with share status, and any public libraries. A project or public library can be additionally protected by defining it as controlled. Defining a library as controlled:

- Prevents documents from being deleted by any user other than the library owner or an administrator
- Prevents any user other than the library owner from storing documents in that library.

DOCUMENT PROTECTION

A document can be protected in several ways:

- Its owner can assign a document password. When accessing a document with a password, the library user must supply the password. (However, under certain circumstances, an administrator can access the document without using the password.)
- The owner can assign sharable or nonsharable status to a document. Sharable or nonsharable status protects a document by limiting user access. Each document in a public library has share status, that is, any valid user can access it. Documents in project or private libraries can be given share status by the document or library owner or by an administrator. Share status extends accessibility of the document to all users of the document library.
- A document can be locked (made unavailable) by the document owner, the library owner, or an administrator, so no other general users can access it. The document owner, the library owner, or an administrator can unlock (make available) the document to reinstate access to it.

CHAPTER 2. JCL AND COMMAND SYNTAX CONVENTIONS

For each specific task performed by DLF, the library user must supply the required host system job control language (JCL, for OS/VS2) or job control statements (JCS, for VSE). Hereafter, in this manual, the term JCL includes VSE JCS. For each command, the library user also must observe certain syntax rules.

This chapter discusses the JCL rules that apply to every job. It presents the notational conventions that are used for this guide. It also explains the syntax rules you will need to execute DLF commands. However, this chapter is not a comprehensive discussion of all JCL or commands. For further information, consult Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics" and Chapter 3, "General User Commands." If you are an administrator, also consult Chapter 5, "Administrator Commands."

Note: For library user tasks, the required JCL statements can be placed in a JCL procedure and stored in the host system's procedure library (see the example in Figure 2). This measure simplifies job setup.

JCL STATEMENTS

These JCL rules apply to every job:

1. **To execute DLF:** Specify the load module DSMSPEXC on the EXEC statement. For OS/VS2, you can specify the LIST or NOLIST parameter on the EXEC statement. For VSE you can specify LISTX or NOLISTX on the OPTION statement. The LIST or LISTX keyword requests that, during the execution of certain commands (COPY, EXPORT, and IMPORT), DLF print the contents of the documents processed. The NOLIST or NOLISTX (the default values) requests that only the document names be printed.

The LIST/NOLIST parameters also control the printing of the description of newly defined or altered profiles.

2. **To identify minimum VSAM data sets:** Each job that executes DLF must include at least two DD statements (for OS/VS2) or DLBL and EXTENT statement pairs (for VSE) identifying the minimum VSAM data sets (the KSDS directory data set and the basic 4K ESDS source data set) that make up the document library. Together, these two DD statements identify a minimum document library for either input or output.

Note: Data sets are described by JCL DD statements in OS/VS2 environments and by DLBL or TLBL statements in VSE environments. The ddnames identify DD statements, and DLBL/TLBL names identify DLBL/TLBL statements. Hereafter in this manual, ddname is used to refer to both ddnames and DLBL/TLBL names in order to reduce the complexity of descriptive information.

The ddnames you use to refer to the document library determine whether it is to be used for input or output. Whether you include the input or output JCL statements depends on which commands you plan to issue. For example, if you take a document out of your library (that is, read a document), you need input JCL statements. If you put a document into your library or change its share status or password, you need output JCL statements. See Figure 2 on page 14 for a list of ddnames that are predefined by DLF.

OS/VS2 ddname	VSE DLBL name	In/Out	Data Set Type
DSMINDIR	DSMIDIR	Input	Directory
DSMINLIB	DSMILIB	Input	Source
DSMPTDIR	DSMPDIR	Output	Directory
DSMPTLIB	DSMPLIB	Output	Source
SYSIN	SYSRDR	Input	Reader
DSMLIST	SYSLST	Output	Printer

Figure 2. Predefined ddnames. Every invocation of DLF requires some combination of these DD statements.

It is possible that four DD statements, identifying both the minimum input and the minimum output document libraries, will be required in a job step. For example, specific functions of the COPY command require that both the input and output JCL statements be included. They may or may not specify the same document library. Usually, however, the input and output DD statements do refer to the same document library.

For input in OS/VS2, the ddname for the directory data set is fixed as DSMINDIR; that for the source data set is DSMINLIB. In VSE, the DLBL name for the directory data set must be DSMIDIR, and for the source data set must be DSMILIB. The disposition should be SHR to allow for multiple access to the data sets. For example:

```
//DSMINDIR DD DSN=DSMLIB.DIRECTRY,DISP=SHR
//DSMINLIB DD DSN=DSMLIB.SOURCE,DISP=SHR
```

When required for output, the ddnames must be DSMPTDIR and DSMPTLIB (for OS/VS2) and DSMPDIR and DSMPLIB (for VSE).

The disposition should be OLD if access to the document library is to be through DLF only. If, however, the document library can also be accessed by a long-running interactive task such as ATMS-III or the Document Interchange Facility (DIF), then DISP=SHR should be used to allow DLF to access the document library concurrently with the long-running task. If there are multiple DLF job submissions with the long-running task, all DLF jobs should be submitted to the same job class (with a single initiator active for that class).

The DISP=SHR parameter allocates the document library to concurrent jobs. It does not prevent an OPEN for update that coincides with an OPEN for access. To avoid not found or lost data situations that may occur, DLF enqueues (locks in VSE) the document library for UPDATE and SHARE depending on the command being processed. On jobs with DISP=SHR specified for the output library, the LIBRARY keyword must have been specified on a DEFINE SYSTEM or ALTER SYSTEM command before running the job.

The examples in this guide assume that only DLF has access to the document library and that DISP=OLD is used. For example:

```
//DSMPTDIR DD DSN=DSMLIB.DIRECTRY,DISP=OLD
//DSMPTLIB DD DSN=DSMLIB.SOURCE,DISP=OLD
```

A job stream usually includes multiple commands, the combination of which may require both input and output JCL statements. In all cases for the general user (there are exceptions for administrators), it is never wrong to include both sets of JCL statements even though both may not be required.

In addition to the basic 4K ESDS, you can define additional source clusters for the document library. These clusters

```

*****
* Summary Page
*****
IR                II04540
STATUS            INTRAN
SYNPTOM           UN
KEYWORD           INCORROUT
INITIAL COMP ID   INFOPALIB      PA LIB INFO ITE
FINAL COMP ID
Severity          4

```

ABSTRACT:
CLARIFICATION OF DLF LIBRARY SERIALIZATION

```

*****
* Submittor Page
*****

```

```

SYSRES=          SYSIN=          SYSOUT=          CPU=          RE-IPL=
OPTYE=           SPECIAL ACTIVITY=        REGRESSION=
PRE-SCREEN NO.=  RSCP= RS000

```

ERROR DESCRIPTION:

THE PURPOSE OF THIS APAR IS TO CLARIFY THE TOPIC OF DLF LIBRARY SERIALIZATION AS DESCRIBED IN DLF GUIDE, CHAPTER 2, "JCL STATEMENTS", ITEM 2.

IN THE TEXT OF THIS APAR, "DLF LIBRARY" OR "LIBRARY" WILL REFER TO A COMPLETE SET OF VSAM CLUSTERS WHICH MAKE UP A DLF LIBRARY SYSTEM, THAT IS, THE DIRECTORY CLUSTER AND ALL OF THE SOURCE OR LIBRARY CLUSTERS DEFINED TO IT. FOR THE PURPOSES OF SERIALIZATION, A DLF LIBRARY IS A SINGLE ENTITY, THAT IS, LOCAL FIX:
SERIALIZATION IS PERFORMED ON THE COMPLETE LIBRARY NOT ON AN INDIVIDUAL CLUSTER.

THE FACTORS WHICH AFFECT SERIALIZATION ARE THE "DISP" PARAMETER CODED IN THE JCL FOR EACH DLF JOB, THE "SHAREOPTIONS" SPECIFIED WHEN THE VSAM CLUSTERS ARE DEFINED WITH AMS, AND THE LIBRARY PARAMETER CODED ON THE DLF "DEFINE SYSTEM" COMMAND.

THE "DISP" PARAMETER CONTROLS SERIALIZATION BY ALLOCATION. DLF JOBS SHOULD BE CODED WITH "DISP=SHR" ON ALL INPUT LIBRARY DD CARDS. OUTPUT LIBRARY DD CARDS SHOULD BE INCLUDED IN DLF JOBS ONLY IF REQUIRED. IN AN ALL BATCH DLF ENVIRONMENT, "DISP=OLD" SHOULD BE CODED ON OUTPUT LIBRARY DD CARDS. THIS WILL CAUSE THE JOBS TO BE SERIALIZED, THAT IS IF TWO OUTPUT JOBS ARE SUBMITTED FOR THE SAME LIBRARY AT THE SAME TIME, MVS WILL MAKE ONE OF THEM WAIT UNTIL THE OTHER HAS COMPLETED. IF DLF BATCH IS TO RUN ALONG WITH A LONG RUNNING TASK SUCH AS ATMS WHICH ACCESSES THE DLF LIBRARY, THEN "DISP=SHR" WILL HAVE TO BE CODED ON THE OUTPUT LIBRARY DD CARDS FOR BATCH JOBS AND A DIFFERENT FORM OF SERIALIZATION IS REQUIRED AND WILL BE DISCUSSED LATER.

VSAM CHECKS FOR PROPER SERIALIZATION OF THE LIBRARY CLUSTERS AT "OPEN" TIME. IF A CLUSTER IS OPENED FOR INPUT, THE ACCESS IS "READ". IF A CLUSTER IS OPENED FOR OUTPUT THE ACCESS IS WRITE. DLF REQUIRES THAT SHAREOPTIONS(2), AMS DEFINE CLUSTER COMMAND, BE SPECIFIED FOR ALL VSAM CLUSTERS WHICH MAKE UP A DLF LIBRARY. THIS ALLOWS MULTIPLE READ ACCESS AND A SINGLE WRITE ACCESS TO THE CLUSTERS TO BE PROCESSED AT THE SAME TIME. THIS OPTION ALSO REQUIRES THE USER, DLF, TO INSURE THE READ INTEGRITY OF THE LIBRARY, WHICH IS THE PURPOSE OF THE LIBRARY NAME PARAMETER ON THE DLF "DEFINE SYSTEM" COMMAND. DLF OPENS THE DEFAULT CLUSTERS DURING INITIALIZATION AND PERFORMS NO INTERNAL SERIALIZATION. IF TWO DLF JOBS ARE EXECUTED AT THE SAME TIME FOR THE SAME OUTPUT DLF LIBRARY (DISP=SHR IN JCL), ONE OF THEM WILL FAIL IN INITIALIZATION WITH MSGDSMPIN024E "UNABLE TO OPEN DSMPTDIR" AND MSGIEC161I "052-84". THE "052" IS A VSAM OPEN RETURN CODE MEANING THAT THE DATA SET IS ALL READY OPEN FOR OUTPUT BY ANOTHER JOB. THIS IS WHY IN AN ALL BATCH ENVIRONMENT, THE OUTPUT LIBRARY DD CARDS SHOULD BE CODED WITH "DISP=OLD".

IN AN ALL BATCH ENVIRONMENT, SERIALIZATION IS ACCOMPLISHED BY CODING THE PROPER "DISP" PARAMETERS IN THE JCL. HOWEVER, IN AN ATMS ENVIRONMENT OR ANY OTHER ENVIRONMENT WHERE ONE OR MORE INPUT TYPE FUNCTIONS ARE TO RUN CONCURRENTLY WITH AN OUTPUT OPERATION, THIS WILL NOT WORK. MVS WILL NOT LET AN OUTPUT JOB WITH "DISP=OLD" CODED IN THE JCL RUN UNTIL ALL CURRENTLY RUNNING "SHR" REQUESTS HAVE COMPLETED AND NO NEW "SHR" REQUESTS CAN BE STARTED UNTIL THE "OLD" REQUEST HAS COMPLETED. ATMS IS A LONG RUNNING CICS APPLICATION THAT REQUIRES READ ACCESS TO THE DLF LIBRARY. THE STARTUP JCL DD CARDS FOR THE DLF LIBRARY ARE CODED WITH "DISP=SHR". USUALLY ATMS RUNS FROM MVS IPL TO MVS IPL AND, THEREFORE, NO OUTPUT DLF BATCH JOBS CAN EXECUTE IF "DISP=OLD" WERE CODED IN THE JCL. TO ALLOW OUTPUT JOBS TO RUN, USERS MUST CODE "DISP=SHR" ON THEIR OUTPUT LIBRARY DD CARDS. HOWEVER THIS INTRODUCES TWO NEW PROBLEMS. FIRST, TWO OUTPUT JOBS MAY ATTEMPT TO EXECUTE AT THE SAME TIME BECAUSE MVS WILL NOT SERIALIZE THE JOBS WITH "DISP=SHR" AND ONE OF THE JOBS WILL FAIL IN VSAM OPEN. THE SECOND PROBLEM IS THAT ONE JOB MAY ATTEMPT TO READ A DOCUMENT WHICH IS BEING UPDATED BY A SECOND JOB. THE RESULTS OF THIS WOULD BE HIGHLY UNPREDICTABLE AND IS WHAT THE VSAM DOCUMENTATION IS REFERRING TO WHEN IT SAYS THAT THE USER IS RESPONSIBLE TO PROVIDE READ INTEGRITY WITH "SHAREOPTION(2)".

THE SUGGESTED WAY TO SERIALIZE OUTPUT DLF JOBS IN THIS ENVIRONMENT IS TO SETUP A UNIQUE MVS JOB CLASS WITH A SINGLE INITIATOR. ALL DLF JOBS WITH OUTPUT LIBRARY DD CARDS, SHOULD BE RUN IN THIS CLASS AND WOULD THEREFORE RUN SERIALY UNDER THE ONE INITIATOR. THE SECOND PROBLEM IS RESOLVED BY CODING THE "LIBRARY" PARAMETER ON THE DLF "DEFINE SYSTEM" COMMAND. A DIFFERENT LIBRARY NAME SHOULD BE USED FOR EACH DLF LIBRARY. IF A LIBRARY NAME IS SPECIFIED BOTH ATMS AND DLF WILL DO ENQ'S TO SERIALIZE THE LIBRARY. THE MAJOR NAME FOR THESE ENQ'S IS "DSMSPEXC" AND THE MINOR NAME IS THE SPECIFIED LIBRARY NAME. WHEN READING A DOCUMENT, ATMS WILL ENQ SHARED ON THE RESOURCE FOR THE OPERATION DURATION. DLF ENQ'S AT THE BEGINNING OF EACH COMMAND AND DEQ'S WHEN THE COMMAND HAS COMPLETED. A DLF JOB MAY HAVE AN INPUT LIBRARY AND AN OUTPUT LIBRARY. THESE MAY BE DIFFERENT LIBRARIES OR THEY MAY BE THE SAME LIBRARY. DLF ONLY ENQ'S ON THE INPUT AND/OR OUTPUT LIBRARY BASED ON THE REQUIREMENTS OF THE CURRENT COMMAND. THAT IS, FOR A COMMAND WHICH REQUIRES ACCESS TO THE INPUT LIBRARY, DLF WILL ENQ SHARED ON THE INPUT LIBRARY NAME. FOR A COMMAND THAT REQUIRES ACCESS TO THE OUTPUT LIBRARY, DLF WILL ENQ EXCLUSIVE ON THE OUTPUT LIBRARY. IF THE COMMAND REQUIRES ACCESS TO BOTH LIBRARIES AND THE LIBRARIES ARE NOT THE SAME, DLF WILL ENQ SHARED ON THE INPUT LIBRARY AND EXCLUSIVE ON THE OUTPUT LIBRARY. IF THE LIBRARIES ARE THE SAME, DLF WILL ENQ EXCLUSIVE ON THE LIBRARY NAME.

also must be represented in the JCL if they are used during an invocation of DLF. The input and output ddnames that must be specified for these additional clusters are those that were specified when the clusters were defined to DLF (see "DEFINE CLUSTER" in Chapter 5, "Administrator Commands").

3. **To direct data stream input and printed output:** JCL statements are required that contain ddnames for data stream input and printed output. In OS/VS2, the ddnames are SYSIN and DSMLIST respectively. In VSE, the labels are SYSRDR and SYSLST, but need not be included in the JCL statements if they are permanently assigned.
4. **To receive error messages:** If any required JCL statements are missing, DLF prints an error message and stops processing. (You must specify DSMLIST or SYSLST to receive error messages from DLF.) Most non-JCL errors will not end the job; however, DLF will end if the time stamp maintained in the directory data set does not match the source data set time stamp or if DLF cannot get enough storage to execute its functions.
5. **To specify external data sets:** Additional JCL statements are required for the ACCOUNT, COPY, IMPORT, ARCHIVE, RETRIEVE, and EXPORT commands, or for the DCF external files (imbeds, appends, or profiles) if the FROM or TO keywords are specified for data sets (in OS/VS2) or files (in VSE) that are external to the document library.

In OS/VS2 environments, a partitioned data set member can be used as the external sequential data set in most of the commands that require external input or output. In this case, the DD statement defining the partitioned data set member is specified:

```
//DOC2DD DD DSN=PDSLIB(DOC2MEM),DISP=SHR,UNIT=3330,
// VOL=SER=DISKA
```

Because DLF tries to open both the input and output document libraries, VSAM system messages will be printed unless both input and output libraries are defined in the JCL. These messages should be regarded as normal. The message for OS/VS2 is '...DD STATEMENT MISSING...'. The message for VSE is 'OPEN ERROR X'80'...'.

Figure 3 shows a sample JCL procedure for executing DLF. The STEPCAT statement (for OS/VS2) or IJSYSUC statements (for VSE) are not required if the document library is defined in the host system's master catalog. The STEPLIB statement (for OS/VS2) or IJSYSCL statements (for VSE) are not required if DLF is located in SYS1.LINKLIB (for OS/VS2) or the system core image library (for VSE). Expand this procedure to include JCL statements for any additional source clusters that you define to DLF.

AUTHORIZING ACCESS TO THE DOCUMENT LIBRARY

The AUTH command must always be the first command in the input job stream (with the exception of DEFINE SYSTEM). It must specify a library user number (and a password, if one applies). Through this command, DLF verifies the library user number and permits access to the document library. It is possible to have more than one AUTH command in a job stream. Each time DLF encounters an AUTH command, it repeats the verification process. The library user number associated with the most recent AUTH command becomes the current library user number.

```

//DLF          PROC                                     [for OS/VS2]
//STEP1        EXEC   PGM=DSMSPEXC,PARM='LIST'
//STEP1CAT     DD     DSN=USERCAT,DISP=SHR
//STEP1LIB     DD     DSN=LOADLIB,DISP=SHR
//SYSPRINT     DD     SYSOUT=A
//DSMLIST      DD     SYSOUT=A
//DSMPDIR      DD     DSN=DSMLIB.DIRECTRY,DISP=OLD
//DSMPLIB      DD     DSN=DSMLIB.SOURCE,DISP=OLD
//DSMINDIR     DD     DSN=DSMLIB.DIRECTRY,DISP=SHR
//DSMINLIB     DD     DSN=DSMLIB.SOURCE,DISP=SHR
//             PEND

```

```

CATALP        DLF,VM=01.0                               [for VSE]
// DLBL        DLFLOAD,'LOADLIB'
// EXTENT      ,USRDSK
LIBDEF        CL,SEARCH=DLFLOAD,TEMP
// DLBL        IJSYSUC,'USERCAT',,VSAM
// DLBL        DSMPDIR,'DSMLIB.DIRECTRY',,VSAM
// EXTENT      ,USRDSK
// DLBL        DSMPLIB,'DSMLIB.SOURCE',,VSAM
// EXTENT      ,USRDSK
// DLBL        DSMIDIR,'DSMLIB.DIRECTRY',,VSAM
// EXTENT      ,USRDSK
// DLBL        DSMILIB,'DSMLIB.SOURCE',,VSAM
// EXTENT      ,USRDSK
// OPTION      LISTX
// EXEC        DSMSPEXC,SIZE=AUTO
/*

```

Note: For VSE/AF 2.1.0, the format of the LIBDEF statement would be LIBDEF PHASE, SEARCH=DLFDCF.LIBRARY - where DLFDCF.LIBRARY is the sublibrary containing the DLF program phases.

Figure 3. Sample JCL Procedures for DLF

SYNTAX AND CONVENTIONS

COMMAND STRUCTURE

Each DLF command is identified by a command verb that describes the task to be done. The command verb can consist of one word (such as IMPORT or EXPORT) or two words (such as DEFINE USER or LIST DOCUMENT). In the case of a two-word command like DEFINE or LIST, the second word of the two-word command must always be present for the command to be recognized.

In addition to the command verb, some DLF commands allow or require operands that further qualify the task to be done. These operands follow the command verb and are separated from it and each other by an arbitrary number of separator characters (see "Separators" on page 17).

Command operands can be keywords or variable parameters. Variable parameter operands are positional and must immediately follow the command verb. DLF commands can have only one variable parameter operand. Keyword operands are identified by a unique word that is recognized by DLF. That keyword may or may not have variable subparameters associated with it. These keyword operands can indicate the presence or absence of a condition, for example, LOCK and UNLOCK or ADM and NOADM. Or, they can identify an attribute that is specified as a variable subparameter such as CLASS(name) or USER(number).

Commands and their following operands can be entered in either uppercase or lowercase letters. DLF folds the command verb and all operands (with the exception of SCRIPT commands and variable values enclosed in quotes) to uppercase.

PLACEMENT

Each command must begin on a new line (a new record). It can begin in column 1 or can be preceded by an arbitrary number of separator characters. The end column is column 72.

ABBREVIATIONS

Every word in a command verb and every keyword operand can be shortened starting from the right. The shortest abbreviation must be unique to identify a DLF command. For example, DEFINE can be shortened to D (also, DE, DEF, DEFI, and DEFIN) because it is the only command beginning with the letter D. However, the shortest unique abbreviations for ACCOUNT, ALTER, and AUTH are AC, AL, and AU, respectively.

Variable parameters (values), cannot be shortened.

SEPARATORS

A separator can be an arbitrary number of blanks or commas. Separate the command verb and operands from one another by one or more separator characters. Also separate the parameters in each operand by one or more separator characters.

Exceptions:

1. A parameter ending with a right parenthesis does not need to be separated from the next parameter.
2. A password separated from a preceding document name or user number by a slash (/) must not have a separator between the slash and the password.

CONTINUATION

A command can be continued by making a minus (-) or a plus (+) sign the last nonblank character in the record. The maximum command length is 1 020 bytes.

A minus sign as the last nonblank character in a record indicates that the command is continued (after the character preceding the minus sign) with the first character of the next line or record. For example,

```
DEFINE USER 123 -
ADM -
MODEL(321)-
PASSW-
ORD(/OCTOBER)
```

is equivalent to

```
DEFINE USER 123 ADM MODEL(321)PASSWORD(/OCTOBER)
```

Notice that a blank character before a minus sign appears in the equivalent command. The continuation from MODEL(321) to PASSWORD is valid, even though there is no blank space between operands, because the last character of the first operand is a right parenthesis.

A plus sign as the last nonblank character indicates that the command is continued (after the character preceding the plus sign) with the first nonseparator character of the next line or record.

For example,

```
DEFINE USER 123 +  
  ADM +  
  MODEL(321) PASSW+  
  ORD(/OCTOBER)
```

is equivalent to

```
DEFINE USER 123 ADM MODEL(321) PASSWORD(/OCTOBER)
```

COMMENTS

A comment can be inserted wherever a separator can be used; DLF treats it as a separator. A comment must begin with the two characters /* and end with the two characters */.

Comments are continued with either a plus or minus sign. For example:

```
DEFINE/*THIS IS A -  
CONTINUED +  
COMMENT*/USER 123 MODEL(321)
```

is equivalent to

```
DEFINE USER 123 MODEL(321)
```

DLF treats each non-continued comment line, or blank line, as a null command and returns a return code of 0 and a blank line. Use continuation characters to prevent return codes after every line of a multi-line comment.

Do not begin a comment in column 1 because /* in the first two columns can be interpreted by the host system as the end of data in the input stream.

QUOTED STRINGS

Some parameters are defined by a series of characters contained in single quotation marks. These characters make up a quoted string. Each command limits the number of characters that can be contained in a quoted string. However, the string may contain any character. If a single quote is used within the string, it must be represented by two single quotation marks (').

NOTATIONAL CONVENTIONS

This guide uses the following notational conventions to describe DLF commands:

1. Words shown in CAPITAL letters can be entered as shown, in either upper or lowercase letters, or shortened to a unique abbreviation.
2. Variables for which you supply the value are shown in lowercase letters.
3. If there is a default value for an operand, it is underscored.
4. Optional operands are enclosed in brackets ([]). Only one can be specified.
5. A list of operands enclosed in small braces ({ }) indicates that one and only one of the choices must be specified.
6. A list of operands enclosed in bold braces indicates that at least one of the choices must be made.
7. Required parameters are shown without brackets or braces.

COMMONLY USED COMMAND OPERANDS

A command can refer to many names, numbers, or attributes of documents, library users, or other entities of the document library. Each item is usually referred to by the same keyword or variable parameter operand whenever it is used. The characteristics of the names, numbers, or attributes most frequently used are fully described in this chapter and apply to all commands in this guide. Options, dependencies, and any deviations from the characteristics cited are included in the individual command descriptions.

DLF command operands are of two types: (1) variable parameter operands and (2) keyword operands.

VARIABLE PARAMETER OPERANDS

Variable parameter operands are positionally dependent, that is, their position in the command identifies them. Each DLF command has no more than one such operand, and it must immediately follow the command verb because it has no identifying keyword. However, this one variable parameter operand can have multiple parts or subparameters.

Variable parameter operands are shown in the command descriptions by a string of lowercase characters that are usually descriptive of the information to be entered. The most frequently used are docid, name, and number.

docid This variable parameter operand has three parts (subparameters) and identifies a document in the document library. The three subparameters are dlib#, dname, and dpass.

DOCID	<div><div>[dlib#]</div><div>{ dname }</div><div>{ 'dname' }</div><div>[/dpass]</div><div>[/'dpass']</div></div>
--------------	---

dlib# This is an optional subparameter. It indicates that a 1-to-7 character library number (dlib#) can be entered and that the value of this number is from 0 to 9 999 999. This number normally identifies the specific library in which the document being identified resides or will reside. In many commands, however, a zero can be entered, which means that the library specified is the current user's project library. If dlib# is not specified, the library owned by the current user is implied. The current user is defined by the most recent AUTH command processed.

dname This is a required subparameter. It is a 1-to-16 character document name that identifies a document in a user's library. A document name can contain any characters if it is enclosed in single quotation marks. (These enclosing quotation marks are not included in the limit of 16 characters.) If a single quotation mark is to be included in a document name that is enclosed in quotation marks, it must be represented by

two adjacent single quotation marks. If a document name is not enclosed in single quotation marks, it is folded to uppercase and can only contain alphanumeric characters or the characters \$, #, or @. The first character of an unquoted document name must not be numeric. There is no default for this subparameter.

dpass This is an optional subparameter and must be entered only if a password is (or will be) associated with the document. It indicates that a 1-to-8-character document password can be (and in some cases, must be) entered. When specified, the document password must be separated from the document name by a slash (/). Separator characters are invalid between the slash (/) and the beginning of the password or quoted password. A document password can contain any characters if it is enclosed in single quotation marks. (These enclosing quotation marks are not included in the limit of 8 characters.) If a single quotation mark is to be included in a document password that is enclosed in quotation marks, it must be represented by two adjacent single quotation marks. If the password is not enclosed in single quotation marks, it is folded to uppercase and can only contain alphanumeric characters or the characters \$, #, or @. Unlike document names, passwords can begin with a number. There is no default for this subparameter.

Examples of valid docid formats:

```
DOCNAME
'docname'
1234567 DOCNAME
1234567 'docname'
DOCNAME/password
'docname'/PASSWORD
1234567 DOCNAME/PASSWORD
1234567 'docname'/PASSWORD
1234567 'docname'/'password'
1234567 DOCNAME/'password'
'doc name is 14'/'pass wrd'
1234567DOCNAME
1234567'docname'
'docname' /PASSWORD
1234567 DOCNAME /PASSWORD
```

name	This variable parameter operand indicates that a name is to be entered. The type of name is determined by the command. For example, DEFINE CLUSTER name indicates that the name of the cluster is to be specified. The characteristics of the various names are defined either in "Keyword Operands" in this chapter or in the applicable command description.
number	This variable parameter operand indicates that a number is to be entered. The command determines the type of number. For example, DEFINE USER number indicates that a library user number is to be specified. The characteristics of the various numbers are defined either in "Keyword Operands" in this chapter or in the applicable command description.

KEYWORD OPERANDS

A keyword operand may or may not have an associated variable. If a variable is required, it is indicated in parentheses following the keyword. Two examples are CLASS(name) and VERSION(number).

The descriptions below show only the keyword because the variable qualifies the keyword and can differ among the commands. For example, VERSION(number), VERSION(number1:number2), and VERSION(:number2) are all variations of the keyword operand that refer to document version number(s).

Note: On commands where CLASS, DATA, or SOURCE operands are optional, the attribute values specified in the user's profile will be used as the default. However, it is important to remember that DLF usually treats a null attribute as it does any other value specified for the attribute - rather than as an absence of a value for the attribute.

Therefore, if, on an EXPORT command for example, the current user's default data type is not exactly the same as the data type of the document selected, the user must explicitly specify the data type of the document, and they cannot be the same if only one of them is null.

CLASS This operand shows the class attribute associated with, or to be associated with, a document or set of documents. The class attribute can contain alphanumeric characters or the characters \$, #, or @ and is folded to uppercase if entered in lowercase. It must not begin with a numeric character. It can be 1 to 8 characters long. If a zero is specified for a class name, the document or profile referred to by the command has a null class attribute associated with it.

A class attribute can be associated with a document in order to identify it as a member of a group of documents with some similar characteristic (usually size). The user or installation defines the characteristic associated with class. Documents of like class can, at your option, be stored together on a particular cluster in the document library. This option is elected by the DEFINE CLASS command.

DATA This operand shows the data attribute, also known as data type, associated with, or to be associated with, a document or set of documents. The data attribute can contain alphanumeric characters or the characters \$, #, or @, and is folded to uppercase if entered in lowercase. It must not begin with a numeric character. It can be 1 to 8 characters long. If a zero is specified for the data type, the document or profile referred to by the command has a null data attribute associated with it.

A data attribute can be associated with a document in order to identify it as a member of a group of documents with some similar content. You define the content associated with data type, but the operand usually describes the type of formatting controls contained in the document, for example, DATA(script) or DATA(atms). For some commands, the document library uses a document's data attribute to determine if a special processor is used during the handling of this command.

SOURCE This operand shows the source attribute associated with, or to be associated with, a document or set of documents. The source attribute can contain alphanumeric characters and the characters \$, #, and @, and is folded to uppercase if entered in lowercase. It must not begin with a numeric character. It can be 1 to 8 characters long. If a zero is specified for a

source name, the document or profile referred to by the command has a null source attribute associated with it.

A source attribute is associated with a document to identify the external source from which it originated. The source names used are defined by you, and are normally descriptive of the external source system from which the documents originated. For example, SOURCE(TSO) shows that a document is from a TSO system.

VERSION This operand specifies a particular version or range of versions of a document. The number(s) specified can be from 1 to 32 767. Some commands and messages allow a zero to be designated for version to indicate a document that has no versions. For further information on the use of this operand, consult Chapter 3, "General User Commands."

CLUSTER This operand shows the cluster name associated with an ESDS source data set in the document library. The cluster name can contain alphanumeric characters or the characters \$, #, and @, and is folded to uppercase when entered in lowercase. It must not begin with a numeric character. It can be 1 to 8 characters long. A cluster name is assigned to each document library cluster (other than the basic 4K cluster) at the time it is defined.

The basic 4K cluster is automatically assigned the name DSMPTLIB (DSMPLIB in VSE) when the system is defined. All other clusters must be named by using the DEFINE CLUSTER command. Documents of a particular class can be routed to a specific cluster at your option (elected by the DEFINE CLASS command or the CLUSTER keyword on commands like IMPORT). If the cluster is not specified or implied, a document will be routed to the default 4K cluster.

T0 This operand frequently specifies an external data set to which information is to be written. If a specific ddname is specified as the variable for the T0 operand, DLF opens the data set defined by the JCL statement that is identified by that ddname.

The ddname variable parameter indicates the label of the DD statement (DLBL or TLBL statement in VSE) that must be included in the JCL to define the T0 data set. In OS/VS2 environments, ddname can contain 1 to 8 alphanumeric characters and the characters \$, #, or @. It must not begin with a numeric character. In VSE environments, the ddname (actually DLBL/TLBL filename) is limited to 1 to 7 characters but otherwise has the same characteristics as those in OS/VS2 environments.

This manual will use the term ddname to refer to both the OS/VS2 ddname and the VSE filename.

In the OS/VS2 environment only, the T0 data set can be dynamically allocated.

Instead of a ddname, the variable parameter associated with the T0 operand can name a specific data set name (cataloged or created in this job step). This is indicated by T0(DSN(xxxxx)) where xxxxx is a specific data set name. xxxxx can be fully qualified and therefore is to be used as it is (indicated by placing it in quotes) or, for some commands, can be unqualified in which case DLF adds qualifiers as defined for the individual commands and as described in Appendix B under "Data Set Naming For Dynamic Allocation."

FROM

The same characteristics that apply to the TO operand apply to the FROM operand, except that the FROM operand refers to an external data set from which information or data is to be read.

PART 2. THE GENERAL USER

Part 2 of the Document Library Facility Guide provides an alphabetic listing of all general user commands, their functions, and detailed examples of their use.

General users are authorized to use the library program; but have limited capabilities. As a general user, you do not have administrator authority and you control only the documents and the library you own.

Chapter 3, "General User Commands" on page 27 gives detailed information about the use of each DLF command a general user can issue.

Chapter 4, "Using the Document Library" on page 101 discusses how to perform commonly required tasks.

CHAPTER 3. GENERAL USER COMMANDS

This chapter gives detailed information on the use of each DLF command a general user can issue.

Note: Chapter 4, "Using the Document Library" on page 101 contains examples of how to use each of the commands in the table below.

DLF commands are read from SYSIN (for OS/VS2) or SYSRDR (for VSE) JCL statements. These data sets must contain 80-character records and can be blocked for OS/VS2, but must be unblocked for VSE. All 80 characters of each record are scanned for input; no part of the record is reserved for sequence numbers.

Command	Function
ALTER DOCUMENT	Locks or unlocks a document or changes its attributes or name. General users can alter any document in the library they own.
ARCHIVE ALL	Archives all selected documents from a user's library to an external sequential data set. The document entry records (DERs) are archived with the documents but a copy of each DER is retained in the directory data set.
ARCHIVE DOCUMENT	Archives a document, along with its DER, from the document library to an external sequential data set. A copy of the DER is retained in the directory data set.
ARCHIVE NAMELIST	Archives documents specified in a name list from the document library to an external sequential data set. The DERs are archived with the documents but a copy of each DER is retained in the directory data set.
AUTH	Identifies someone as an authorized user of DLF and establishes the current user number.
CLOSE	Ends the reading of one or more documents when DLF is used as a subroutine of an application program.
COPY DOCUMENT	Copies a document from one user's library to another user's library.
COPY IN	Restores all or some of a user's documents by using the backup copy created during the COPY OUT command.
COPY OUT	Makes a backup copy of some or all of a user's documents on an external data set.
ENVIRONMENT (Changed)	Describes the characteristics of a VSE sequential file that is used for input or output. Not a valid command in OS/VS2.
EXPORT	Makes an external copy of a document in the document library.
IMPORT	Reads a document from a sequential file and stores it in the document library.
LIST CLASS (Changed)	Lists all of the class entry records (CERs) that are defined in the directory data set.
LIST CLUSTER (Changed)	Lists the cluster profile records (CPRs) defined in the directory data set.

Figure 4 (Part 1 of 2). Summary of the General User Commands

Command	Function
LIST DOCUMENT (Changed)	Lists the attributes of selected documents stored in the user's own library, project library, or any public library.
LIST PROCESS (Changed)	Lists the processor profiles that are defined in the directory data set.
LIST USER (Changed)	Lists information from the requester's user profile record (UPR), project library, and/or any public library.
PASSWORD	Changes or deletes a user's password.
PROTECT	Changes the password or share status of a document or a version of a document.
PURGE DOCUMENT	Purges a document (any or all versions) from a user's library. Library users can purge documents they own in their project or any public library as well as any documents in their own library.
READ	Requests records from the document library when DLF is used as a subroutine of an application program.
RETRIEVE ALL	Retrieves (copies into the document library) all of the documents in a user's library that were previously archived to the data set specified in the command.
RETRIEVE DOCUMENT	Retrieves a single document that was previously archived to an external sequential data set.
RETRIEVE NAMELIST	Retrieves the documents specified in a name list. All the listed documents must have previously been archived to the same external sequential data set.
SCRIPT	Invokes the SCRIPT/VS formatter when the Document Composition Facility is also installed.

Figure 4 (Part 2 of 2). Summary of the General User Commands

ALTER DOCUMENT

This command locks or unlocks a document, changes its attributes, or changes its name. If only one version of a document exists, all the operands of this command can be used to alter it. If more than one version exists, only the DATA, VERSION, and LOCK or UNLOCK operands can be used.

The general user can alter any document he owns and any document in his library regardless of ownership.

To understand the description of the ALTER DOCUMENT command that follows, the reader must have read and understood the material in the section "Commonly Used Command Operands" in Chapter 2.

ALTER DOCUMENT	docid [DATA(name)] [VERSION(number)] [NEWNAME ({ dname }) { 'dname' }] [NEWDATA(name)] [NEWCLASS(name)] [NEWSOURCE(name)] [LOCK UNLOCK]
----------------	--

- docid** This operand specifies the current document identification of the document to be altered.
- DATA(name)** This operand specifies the current data type of the document to be altered. If not specified, the data type default for the current authorized user is used. A zero placed in the (name) field, indicates that a null data type has been assigned to this document.
- VERSION(number)** This operand specifies the particular version of the document to be altered. If a document has versions, then VERSION(number) must be specified. If a document has more than one version, only its lock status can be altered by this command.
- NEWNAME(dname)** This operand specifies a document's new name and may or may not be entered as a quoted string. If NEWNAME is not specified, the document name is not changed.
- NEWDATA(name)** This operand specifies the new data type for the document. If a zero is placed in the (name) field, the current data name is nullified. If NEWDATA is not specified, the data type attribute for the document is not changed.

- NEWCLASS(name)** This operand specifies a new class for this document. If a zero is placed in the (name) field, the current class name is nullified. If NEWCLASS is not specified, the class attribute for the document is not changed.
- NEWSOURCE(name)** This operand specifies a new source attribute for this document. If a zero is specified in the (name) field, the current source attribute in the document entry record is nullified. If NEWSOURCE is not specified, the source attribute of the document is not changed.
- LOCK** This operand specifies that the document is to be locked against normal access. It can be accessed only by the use of COPY, ARCHIVE, or RETRIEVE commands or the LOCK option of the LIST DOCUMENT command until it is unlocked by another ALTER command. Only one version of a multiple-version document can be locked by each ALTER DOCUMENT command.
- If you omit the LOCK/UNLOCK operand pair from the ALTER DOCUMENT command, no change is made to the lock status of the document.
- UNLOCK** This operand specifies that this document is to be unlocked to become accessible to authorized library users. Only one version of a multiple-version document can be unlocked by each ALTER DOCUMENT command.
- If you omit the LOCK/UNLOCK operand pair from the ALTER DOCUMENT command, no change is made to the lock status of the document.

Notes:

1. The output document library JCL statements are required.
2. When a document or a version of a document is locked, UNLOCK is the only keyword that can be used to alter the document.
3. Documents with null data types (expressed as DATA(0)), can be copied into libraries whose owners do not have a default data type of null. In order to conveniently access these documents after they are copied, change their data type to the default data type of the library receiving the document. This is done with an ALTER DOCUMENT command as shown in the example below:

ALTER DOCUMENT DOCA DATA(0) NEWDATA(SCRIPT).
4. ALTER DOCUMENT cannot be used on documents that are archived.

Examples

- This example locks version 10 of document MULTIPLE:

AUTH 100/WORKER
ALTER DOCUMENT MULTIPLE VERSION(10) LOCK
- This example changes the name and class of document VARIABLE:

AUTH 100/WORKER
ALTER DOCUMENT VARIABLE NEWNAME(STABLE) -
NEWCLASS(MEMO)

ARCHIVE

There are three ARCHIVE commands:

- ARCHIVE ALL
- ARCHIVE DOCUMENT
- ARCHIVE NAMELIST.

These commands move documents from the document library to an external sequential data set. Included in the move is the text of the document and its document entry record (DER). However, a copy of the DER is retained in the directory data set. The text no longer resides in the source data set. A general user can archive all the documents he owns.

When a document has been archived, the only commands that access the document itself are the RETRIEVE commands, which move the document back to the document library from the external sequential data set. A number of commands, however, can act on the copy of the DER retained in the directory data set; these include COPY IN, COPY OUT, COPY LIBRARY, PURGE USER, PURGE DOCUMENT, and LIST DOCUMENT (with the ARCHIVE option). Locked documents can be archived. When documents are archived, the user profile record (UPR) reflects the increase in available DASD space.

The following operands apply to each ARCHIVE command:

NOTE('44 CHARACTERS') This operand specifies that up to 44 characters of information in a quoted string is placed into the DERNOTE field of the DER of each document archived. Typically, this field is used to keep track of the location of the archived data set. This information remains available in the directory data set because a copy of the DER of each archived document is retained in the document library.

OPERATOR This operand specifies that a message is written from the archive program to the console operator requesting the volume serial number of the volume containing the sequential data set in which the documents are archived. The console operator's response to this request is retained in the DERNOTE field of the DER of each document archived. NOTE and OPERATOR are mutually exclusive. Each retains information in the same field of the DER.

If you use TO(ddname), you can also enter the NOTE or OPERATOR keyword. If you use TO(DSN...) in any one of its forms, you cannot enter NOTE or OPERATOR. TO(DSN...) in any of its forms dynamically allocates the data set, and DLF places the resulting data set location information in the DERNOTE field (which is also used by NOTE and OPERATOR) of the DER of each document archived. TO(DSN...) is valid under OS/VS2 MVS only.

To understand the description of the ARCHIVE commands that follow, the reader must have read and understood the material in the section "Commonly Used Command Operands" in Chapter 2.

Each command and its unique parameters are described in this chapter.

ARCHIVE ALL

This command archives all documents from the general user's library that meet the selection criteria specified in the command. The documents can be those this user owns or those owned by other users that are stored in this user's library (public or project).

ARCHIVE ALL	<pre> { ddname } { DSN(dsname) } TO ({ DSN(dsname(member)) }) { DSN('dsname') } { DSN('dsname(member)') } [DATA(name)] [DATED ({ date } { date1:date2 }) { date1: } :date2 }] [NOTE('44 characters') OPERATOR] </pre>
-------------	--

TO This required operand specifies the external sequential data set or partitioned data set member receiving the archived documents and DERs.

ddname is the name on the DD, DLBL, or TLBL JCL statement that identifies the TO data set.

DSN... is valid in OS/VS2 MVS only and shows that the TO data set is dynamically allocated. If the specified data set name is in quotation marks, it is used as given. Otherwise it is qualified as described in Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics." ARCHIVE is the suffix qualifier. If TO(DSN...) in any of its forms is used, do not specify NOTE or OPERATOR.

DATA(name) This operand specifies that only documents with this data type are archived. The default is all data types that meet other selection criteria.

DATED This operand specifies that only documents stored in the document library on or between the dates specified are archived. If DATED is not specified, then all documents that meet other selection criteria are archived regardless of the dates they were stored in the document library. Each date specified must be in the form mm/dd/yy where mm is the month (1 to 12), dd is the day of month (1 to 31), and yy is the last two digits of the year.

date specifies that only documents stored in the document library on a particular date are to be archived.

date1:date2 specifies that only documents stored in the document library within an inclusive range of dates are to be archived.

date1: This operand specifies that only documents stored in the document library on or after the specified date are to be archived.

:date2 This operand specifies that only documents stored in the document library on or before the specified date are to be archived.

NOTE See "ARCHIVE" on page 31.

OPERATOR See "ARCHIVE" on page 31.

Notes:

1. The input and output document library JCL statements are required.
2. Only documents within the current general user's own library (which meet other selection criteria) are archived.
3. For VSE, you must use the ENVIRONMENT command to specify the characteristics of the external data set if the default values are not applicable to you. See Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics" for further details.
4. Locked documents can be archived.
5. If any document being archived has a password, the archive command will fail.

Example

This example archives all selected documents from the current authorized user's library to a sequential data set identified by the ddname ARCDs. Only documents with a data type of ATMS that were stored between January 2, 1979 and January 2, 1980, inclusive, are archived.

```
AUTH 100/WORKER
ARCHIVE ALL DATA(ATMS) DATED(01/02/79:01/02/80) TO(ARCDs)
```

ARCHIVE DOCUMENT

This command archives a specified document.

ARCHIVE DOCUMENT	<p>docid</p> <p>[DATA(name)]</p> <p>[VERSION ({ number } { number1:number2 } { :number2 })]</p> <p>[TO ({ ddname } { DSN(dsname) } { DSN(dsname(member)) } { DSN('dsname') } { DSN('dsname(member)') })]</p> <p>[NOTE('44 characters') OPERATOR]</p>
------------------	---

docid	This operand identifies the document to be archived.
DATA(name)	This operand specifies the data type of the document to be archived. The default is the data type defined in the current user's user profile record (UPR).
VERSION	<p>This operand specifies which version(s) of the document are to be archived. Some form of the VERSION operand must be entered if the document being archived is a document that has versions. If VERSION is used without the number field, all versions of the document, except the latest, are archived.</p> <p>number specifies a specific version to be archived.</p> <p>number1:number2 gives the inclusive range of versions archived.</p> <p>:number2 shows that all versions are to be archived up to and including the one specified.</p>
TO	<p>This operand identifies the external sequential data set or partitioned data set member to which the document(s) and associated DER(s) are to be archived. The TO operand is required for VSE.</p> <p>If the TO keyword is omitted (valid for OS/VS2 MVS only), DLF dynamically allocates a previously defined external sequential data set with a data set name based on the document name of the document being archived. The first eight characters of the document name are folded to uppercase and any special characters converted to the character @. This 8-character name is then qualified as described in Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics." ARCHIVE is the suffix qualifier. <u>If TO is omitted, do not specify NOTE or OPERATOR.</u></p>

ddname is the name on the DD, DLBL, or TLBL JCL statement that identifies the TO data set.

DSN... is valid in OS/VS2 MVS only and shows that the TO data set is dynamically allocated. If the specified data set name is in quotation marks, it is used as given. Otherwise it is qualified as described in Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics." ARCHIVE is the suffix qualifier. If TO(DSN...) in any of its forms is used, do not specify NOTE or OPERATOR.

NOTE See "ARCHIVE" on page 31.

OPERATOR See "ARCHIVE" on page 31.

Notes:

1. The input and output document library JCL statements are required.
2. Multiple documents with the same name but different data types can be stored in a user's library.

Example

This example archives document OLDEST to a sequential disk data set identified by DD statement DISK. The characters VOLID are placed into the DERNOTE field of the DER of the document.

```
AUTH 100/WORKER
ARCHIVE DOCUMENT OLDEST TO(DISK) NOTE('VOLID')
```

ARCHIVE NAMELIST

This command archives all the documents identified in a user-created list. See "Creating a Namelist for the ARCHIVE Command" on page 37 for the format of NAMELIST entries.

ARCHIVE NAMELIST	<pre> { * } ({ ddname }) { DSN('dsname') } { DSN('dsname(member)') } { ddname } { DSN(dsname) } TO ({ DSN(dsname(member)) }) { DSN('dsname') } { DSN('dsname(member)') } [NOTE('44 characters') OPERATOR] </pre>
------------------	--

() The first parameter is positional and required. It shows the location of the namelist, which can be in the job input stream or in a data set or partitioned data set member. See "Creating a Namelist for the ARCHIVE Command" on page 37 for the format of NAMELIST entries. This first parameter must be one of the following:

- * specifies that the namelist follows the command in the job input stream. This is the only valid option in VSE.
- ddname is valid only in OS/VS2 MVS and specifies the name on the JCL DD statement that describes the data set containing the namelist.
- DSN... is valid in OS/VS2 MVS only and shows that the namelist data set is dynamically allocated. The specified data set name must be in quotation marks and it is used as given.

TO This required keyword operand identifies the external sequential data set or partitioned data set member receiving the archived documents and DERs.

- ddname is the name on the DD, DLBL, or TLBL JCL statement that describes the TO data set.
- DSN... is valid in OS/VS2 MVS only and shows that the TO data set is dynamically allocated. If the specified data set name is in quotation marks, it is used as given. Otherwise it is qualified as described in Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics." ARCHIVE is the suffix qualifier. If TO(DSN...) in any of its forms is used, do not specify NOTE or OPERATOR

NOTE See "ARCHIVE" on page 31.

OPERATOR See "ARCHIVE" on page 31.

Note: The input and the output document library JCL statements are required.

CREATING A NAMELIST FOR THE ARCHIVE COMMAND

The ARCHIVE NAMELIST command requires a user-created namelist either following the command in the job input stream or in a sequential data set. This namelist contains one or more entries, each of which must contain at least the name of the document to be archived. Other information in a namelist entry may be required to uniquely identify that document.

A namelist entry can specify any document owned by the current user as indicated by the most recent AUTH command. If the current user is an administrator, a namelist entry can specify any document owned by any user of the document library.

The structure of a namelist entry is:

docid
[DATA(name)]
[VERSION(number)]

docid This operand is positional within a namelist entry and must be specified first. For docid, the reader must be familiar with the information found in Chapter 2, "JCL and Command Syntax Conventions" on page 13.

DATA(name) This operand specifies the data type of the document. If not specified, the default in the current user's UPR is used.

VERSION(number) This operand specifies the version of the document to be archived. VERSION is required for documents that have versions. It is an error if:

1. You specify this operand for a document without versions
2. You specify this operand but do not specify a (number).

NAMELIST FORMAT

1. The namelist format always starts in position 2. A blank in position 1 indicates the start of a new logical record. Record text must not begin in position 1.
2. Namelist entries may be continued on separate 80-byte logical records if you follow these rules:
 - A plus sign (+) in position 1 means that the first character in this record immediately follows the last nonblank character of the previous record.
 - A minus sign (-) in position 1 means that this record is appended to the end of the previous record with no adjustments for blanks.
3. The end of a namelist is indicated by a period in position 1 or an EOF during reading. The period must be in a separate record following the last record of the namelist entries. Any other information in the period record is ignored.

4. No other nonblank characters will be allowed in position 1 of an input record. If a nonblank character encountered in position 1 is not a period, a plus sign, or a minus sign, command processing ends.

Examples

- In this example, the documents OLDDOC1, OLDDOC2, and OLDDOC3 have no versions and are owned by user 100. They have a data type equal to the default data type in user 100's UPR. None of these documents has a password. Because the list is in the job input stream, it must be ended by a record with a period in column 1. The documents are archived to an external sequential data set identified by the DD, DLBL, or TLBL name ARCD5.

```
AUTH 100/WORKER
ARCHIVE NAMLIST (X) TO(ARCD5)
OLDDOC1
OLDDOC2
OLDDOC3
.
```

- This example archives documents named in a list residing in an external sequential data set identified by the DD name DISKNAME. The documents are archived to the tape data set identified by the DD name TAPED5. A request goes to the operator for the volume number of the tape to which the documents are archived. This example is only valid in OS/VS2 MVS.

```
AUTH 100/WORKER
ARCHIVE NAMLIST (DISKNAME) TO(TAPED5) OPERATOR
```

AUTH

The AUTH command is used to authorize a DLF user to use the document library. The specified user number then becomes the "current user."

Each DLF session must begin with an AUTH command.

DLF will authorize a user:

- If there is a previously defined User Profile Record (UPR) for the specified DLF user in the directory data set; and
- If the user has correctly supplied any password specified in that user's UPR.

Defaults for some DLF operands, such as document data type, are based on the value assigned to the current DLF user.

To understand the description of the AUTH command that follows, you should read and understand the material in the section "Commonly User Command Operands" in Chapter 2.

AUTH	{ number[/password] } { number['password'] }
-------------	---

number This operand specifies the library user number of the user requesting authorization.

password This operand is the password associated with this user. User passwords have the same characteristics as document passwords except that they are associated with a user rather than a document. If a user password has been assigned, then it must be specified in the AUTH command. If a password has not been assigned and one is specified, it is treated as an invalid password.

Notes:

1. Either the input or output document library JCL statements are required. If both are included and they do not identify the same document library, both are searched for the specified user profile. If the profile is found in both and the passwords (if any) are identical, the command completes normally. Otherwise, an error message results.
2. Subsequent commands are verified against the information in the most recent AUTH command.
3. An AUTH command must be the first DLF command in all sessions except when you initialize DLF (DEFINE SYSTEM command).

Example

This example authorizes a general user to access the document library as user 55 490 and to access any documents owned by that user. It also gives him any additional privileges, such as access to a project library or administrator authority, if specified in the UPR. General user 55 490 has a password of SECRET.

AUTH 55490/SECRET

CLOSE

This command is used only by an application program that calls DLF as a subroutine. It requests that DLF close one or more document files that are currently open for access by the calling program. See the READ command later in this chapter for information on opening document files.

Chapter 9, "Using DLF as a Subroutine," discusses the parameter list used by the calling program and describes how this parameter list is set up for the READ and CLOSE commands.

To understand the description of the CLOSE command that follows, the reader must have read and understood the material in the section "Commonly Used Command Operands" in Chapter 2.

CLOSE	<div data-bbox="558 576 695 655">[ALL docid]</div> <div data-bbox="558 670 773 727">[DATA(name)]</div> <div data-bbox="558 742 935 800">[VERSION [(number)]]</div>
--------------	--

- | | |
|------------------------|--|
| docid | This operand identifies the document to be closed. |
| ALL | This operand specifies that the application program wants to close all its open document files. This is the default. If ALL is specified, DATA and VERSION cannot be used. |
| DATA(name) | This operand specifies the data type name of the document. If not specified, the default is the data type in the current user's UPR. This parameter cannot be used with ALL. |
| VERSION(number) | This operand specifies the version of the document. If VERSION is specified without (number), the highest version of the document that is open for access by the calling program is closed. This is the default. This parameter cannot be used with ALL. |

Notes:

1. The closing of a document file assumes that a READ was issued earlier to open the document file. The file being closed must have previously been opened, or an error will result.
2. If no document is named or the ALL operand is specified, all open document files are closed.
3. CLOSE ALL, in addition to closing all open document files and their associated ESDSs, closes the directory data set and the basic 4K source data set.

COPY

COPY commands create duplicate documents and their associated document entry records (DERs).

There are three COPY commands a general user can use:

- COPY DOCUMENT
- COPY IN
- COPY OUT.

There are two ways to copy documents. A document can be copied from one user's library to another user's library either within the same document library or in another document library. A general user can copy documents in either direction between the document library and a sequential data set.

To understand the description of the COPY commands that follow, the reader must have read and understood the material in the section "Commonly Used Command Operands" in Chapter 2.

Note: The COPY commands may be used to copy documents from one cluster to another cluster that has the same CI size. The EXPORT/IMPORT commands must be used to move documents between clusters of different CI sizes.

Preparing to Execute a COPY Command

1. **Copying from a document library:** When you are copying from a document library, the input document library JCL statements must refer to the document library from which the documents are being copied. The output JCL statements can refer to:
 - An existing document library, into which documents and profiles are written with or without the REPLACE option. This may, in fact, be the same as the input document library.
 - A newly defined external sequential data set on either tape or disk.
2. **Copying to a document library:** When you are copying to a document library, the output JCL statements must refer to that document library. The input JCL statements can refer to:
 - The same document library.
 - Another document library.
 - An external sequential data set residing on either tape or disk, this data set is created by the COPY OUT command and contains profiles and documents that make up all or part of a document library.
3. **Copying from one document library to another:** When copying from one document library to another, the current general user, as specified by the AUTH command, must be authorized in both document libraries and have the same password in both.

To copy a document from one document library to another, you can copy the document to an external sequential data set and then to the other document library, or you can copy the document directly to the other document library with the JCL statements referring to both document libraries.

COPY DOCUMENT

This command is used to copy a specified document from one user's library to another user's library. The user's libraries can be in the same or different document libraries. The term "source library" refers to the user's library from which the document is copied. "Source document" and "original document" refer to the document being copied. The COPY DOCUMENT command copies the document into the target library where it is referred to as the "copy" or "target document."

Any user can copy any document he has the authority to read. A general user can copy a document into any library except a controlled public or project library owned by another user. This allows a user to give another library user one of his documents without requiring him to export and reimport that document. A document that has been archived cannot be copied unless it is first retrieved.

If the document has versions, the latest version is copied unless one of the VERSION keywords is specified. The user can change the document name or version during the copy process. If the original document has a document password, the copy has the same password.

When the source and target libraries are in the same document library, the source and target library numbers must be different unless the document name or version is changed.

Ownership of A Copied Document

Private library: If the document is copied into a private library, the owner of the private library becomes the owner of the copied document.

Project library: If the document is copied into a project library, the ownership remains the same as long as the document libraries are the same and the user is copying into his own project library. If they are not the same document libraries or the current user is copying a document into a project library other than his own, the ownership is always changed to the owner of the target library, the TOUSER.

Public library: If the document is copied into a public library, the ownership remains the same as long as the document libraries are the same. A library entry record (LER) is created in the source library if the input and output document libraries are the same. If they are not the same document libraries, the ownership is changed to the TOUSER number and no LER is created. When the document ownership is changed space usage is changed appropriately.

COPY DOCUMENT	docid <code>[DATA(name)]</code> <code>[VERSION ({ number })]</code> <code>[{ number1:number2 }]</code> <code>[{ ALL }]</code> <code>[NEWVERSION ({ number })]</code> <code>[TOUSER(number)]</code> <code>[REPLACE]</code> <code>[NOREPLACE]</code> <code>[CLUSTER(name)]</code> <code>[NEWNAME ({ dname })]</code> <code>[{ 'dname' }]</code>
----------------------	---

docid

This operand specifies the document identification of one source document.

DATA(name)

This operand specifies the data type of the original document. The default is the data type specified in the current user's UPR. The data type must be specified if it is not the same as the current user's default data type.

VERSION

This operand specifies which version(s) of the source document are to be copied. The latest version will be copied if the source document is versioned and this operand is omitted or included with no subparameter(s). If this operand is used and the source document is not versioned, an error message will be written and the command will not process.

number specifies the particular version of the source document that is to be copied.

number1:number2 specifies the inclusive range of versions of the source document that are to be copied.

ALL specifies that all versions of the source document are to be copied.

NEWVERSION(number)

This operand specifies that a version number is to be assigned to the target document. The source document need not have versions for the copy to be given a version number. If a range of documents is copied, the NEWVERSION(number) is the starting version number of the copies.

The NEWVERSION operand is valid when the document does not currently exist in the target library or when it is versioned in the target library. The NEWVERSION operand cannot be used to copy a document into an existing document that has no versions.

When the document being copied does not exist in the target library, the version number of the copy will be either the specified NEWVERSION(number) or, for NEWVERSION without number, the same version number as the original versioned document or version one for an unversioned original. If NEWVERSION is not specified, the copy will be unversioned when the source document is unversioned or the copies will have the same version numbers as those copied from the source library.

When the copy already exists as a versioned document in the target library, the first, or only, version number of the copy is defined as follows:

- The NEWVERSION(number) specified must be greater than any existing version number for that document in the target library.
- If NEWVERSION is specified without number, the copy(s) will receive the next highest unused version number(s).
- If NEWVERSION is not specified, the copy(s) will receive the same version numbers as the original document and those version numbers must be higher than any existing version number for the document in the target library for the copy to occur.

TOUSER(number)

This operand specifies the user's library to which the document is copied: the target library. Documents can be copied to any library except a controlled public or project library not owned by the current user. The default is the library owned by the current user.

REPLACE

This operand specifies that the unversioned document in the target library is to be replaced with the unversioned document being copied. If the document does not currently exist in the target library, the REPLACE operand is ignored. If the target library has an unversioned document but the source document is versioned, the command fails.

Note: The owner of the copy must be the same as the owner of the current unversioned document in the target library or the copy will not occur.

NOREPLACE

This operand is the default. It specifies that the unversioned document from the source library is not to replace any existing unversioned document of the same name and data type in the target library. If such a document is found in the target library, the copy will fail.

CLUSTER(name)	This operand specifies the cluster to which the document is to be copied. <u>Both clusters must have the same control interval size or the copy does not occur.</u> If this keyword is omitted, the copy is made to the output library with the same name as the input library cluster containing the source document.
NEWNAME(dname)	This operand specifies a new document name for the copy.

Notes:

1. The input and output document library JCL statements are required.
2. A versioned source document cannot be copied to a target library which has a non-versioned document with the same name and data type as the document being copied.
3. A non-versioned source document can be copied to a target library which has a non-versioned document of the same name and data type only if the REPLACE operand is specified and the owner of the target document after the copy will be the same as the original owner of the document in the target library.
4. A non-versioned source document can be copied to a target library which has a versioned document with the same name and data type provided an appropriate NEWVERSION operand is included in the COPY DOCUMENT command.

Examples

- This example copies a document called SHAREDOK from the private library of the current authorized user to his project library, 1 000. The new document is called OWNDOC in library 1 000.

```
AUTH 2000/GIVER
COPY DOCUMENT SHAREDOK -
TOUSER(1000) NEWNAME(OWNDOC)
```

- This example copies ONLYONE, a document with no versions, to library 1 314 151 and gives it the name of MAYBEMORE in library 1 314 151. It will become a document with versions in the new library. The copy is version one of the document with newly created versions.

```
AUTH 2000/GIVER
COPY DOCUMENT ONLYONE TOUSER(1314151) -
NEWNAME(MAYBEMORE) NEWVERSION
```

i

You can copy in documents you own from a backup data set created by another user (administrator). When a general user is copying from such a data set, none of the attribute records (PPR,CER,UMR) in the data set are copied.

If a general user is copying all of the documents he owns into the document library and a UPR record does not exist for public or project libraries into which some of his documents are to be placed, those documents are not copied and a message identifies each missing UPR.

```

COPY IN      [ { ddname                } ]
              [ { DSN(dsname)          } ]
FROM ( [ { DSN(dsname(member))        } ] )
        [ { DSN('dsname')            } ]
        [ { DSN('dsname(member)')     } ] ]

[ SELECT ( { dname                      } ]
          [ { dname1:dname2            } ]
          [ { 'dname'                  } ]
          [ { 'dname1':'dname2'        } ] ]

[ DATA ( { name                        } ) ]
          [ { name1:name2              } ] ]

[ CLUSTER(name) ]

[ REPLACE ]
[ NOREPLACE ]

```

The FROM operand is required in VSE. If FROM is not provided in OS/VS2, the data set name Usernumber.BACKUP.COPY is used, where Usernumber is U followed by the user number of the current user. The data set must be previously defined.

DSN... is the name of the previously defined sequential data set used for input. COPY is the suffix qualifier, if the dsname is not in quotation marks. This subparameter is valid for dynamic allocation in OS/VS2 MVS only. See Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics" for details.

SELECT	<p>This operand names documents to be copied. The default is that all documents owned by the current user that meet other selection criteria are copied. <u>The password cannot be used when specifying the name of the document copied.</u> Document names enclosed in quotes can contain any character and will not be folded to uppercase.</p> <p>dtype specifies the document name of the document to be copied.</p> <p>dtype1:dtype2 specifies that an inclusive range of document names is to be copied.</p>
DATA	<p>This operand specifies that a particular data type is copied. The default copies all data types.</p> <p>name specifies that a data type is copied.</p> <p>name1:name2 specifies that an inclusive alphabetical range of data types is copied.</p>
CLUSTER(name)	<p>This operand specifies the name of a cluster copied. Only documents originally copied from a cluster with the same name are selected for copying to this cluster name. <u>Both clusters must have the same control interval size or the copy does not occur.</u></p> <p>If you do not specify the CLUSTER operand, the default is all documents that meet the other selection criteria, regardless of the cluster in which they are stored.</p>
REPLACE	<p>This operand specifies that the incoming document replaces an existing document in the library to which it is copied.</p>
NOREPLACE	<p>This operand specifies that the incoming document does not replace an existing document. This operand is the default.</p>

Note: The output document library JCL statements are required.

Example

- This example copies in all the documents owned by general user 6 100 on the sequential data set C111111.DLFLIB.BACKUP.D04081. These documents are only part of the data set which is actually a backup of the complete document library. Because user 6 100 is not an administrator, only the documents user 6 100 owns will be copied. Any documents already in the library will be replaced.

```
AUTH 6100/WORKER
COPY IN FROM(DSN('C111111.DLFLIB.BACKUP.D04081')) REPLACE
```

COPY OUT

This command provides a backup for part of a document library. General users can copy only those documents they own (that is, documents they own in their owned library, their assigned project library, and any public library).

The data set generated by the COPY OUT command is described in Appendix D.

Note: In VSE, if you have multiple COPY commands in a job step, you may want to put multiple files on a single tape (because you can specify only one tape unit in a step). In this case, use NOREWIND to override the REWIND and UNLOAD tape defaults in the ENVIRONMENT command. Then each COPY command creates a separate file on the tape in the order processed.

```

COPY OUT
[
    { ddname                }
    { DSN(dsname)           }
    TO ( { DSN(dsname(member)) } )
    { DSN('dsname')        }
    { DSN('dsname(member)') }
]

[
    { dname                }
    SELECT ( { dname1:dname2 } )
    { 'dname'              }
    { 'dname1':'dname2'    }
]

[
    DATA ( { name          } )
           { name1:name2    }
]

[
    CLUSTER(name)
]

```

TO

This operand identifies the sequential data set or member of a partitioned data set receiving the data. This operand is required in the VSE environment but is optional in OS/VS2.

If TO is not provided in OS/VS2, the data set name Uusernumber.BACKUP.COPY is used, where Uusernumber is U followed by the user number of the current user. This data set must be previously defined.

ddname is the name on the DD, DLBL, or TLBL statement that describes the data set that receives the data.

DSN... is the name of the previously defined sequential data set that receives the data. COPY is the suffix qualifier if dsname is not in quotation marks. See Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics," for details. This parameter is valid for dynamic allocation in OS/VS2 MVS only.

SELECT

This operand names document(s) to be copied. All versions are copied. If this operand is omitted, all the user's documents that meet other selection criteria are copied, including those locked and unlocked and including the DERs for any archived documents. Only the document name is used; neither the library number nor the password is entered. Document names enclosed in quotes may contain any character and will not be folded to uppercase.

dtype specifies the name of a single document to be copied.

dtype1:dtype2 specifies an inclusive alphabetical range of document names to be copied. All versions are copied, including those owned by the user in project or public libraries and any that are locked or archived.

DATA

This operand specifies the particular data type to be copied. The default is all data types.

name specifies a particular data type to be copied.

name1:name2 specifies an inclusive alphabetical range of data types to be copied. The default is all data types.

CLUSTER(name)

This operand specifies the name of a cluster from which the copy takes place. Only documents in this cluster are to be copied.

If you do not specify the CLUSTER operand, the default is all documents that meet the other selection criteria, regardless of the cluster in which they are stored.

Notes:

1. The input document library JCL statements are required.
2. All versions of each document are always written to the backup data set.

Example

This example creates a backup copy of the current authorized general user's documents. The documents are copied to the sequential data set identified by ddname BACKUP. Only documents with names that fall in the inclusive alphabetical range from ESSENTIAL through NONESSENTIAL are copied. These documents can be copied back into the document library with the COPY IN command.

```
AUTH 100/WORKER
COPY OUT TO(BACKUP) SELECT(ESSENTIAL:NONESSENTIAL)
```

ENVIRONMENT

This command applies to VSE only. It defines device and data set characteristics of an external file. It is required when these characteristics are different from the defaults used by DLF. See Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics," for the default characteristics of external data sets used by various DLF commands. The ENVIRONMENT command cannot be used to describe SYSLST, SYSRDR, SYSPCH, or SYSIPT files. A job step can contain no more than 32 different ENVIRONMENT commands.

ENVIRONMENT	<pre>{ DLBL(name) } { TLBL(name) } { FILENAME(name) } [DEVICE(type [unit] [disposition])] [LABEL NOLABEL] [FORMAT(format)] [BLOCKSIZE(number)] [RECORDSIZE(number)]</pre>
--------------------	---

DLBL(name) This operand specifies the name of the DLBL statement identifying the file on disk.

TLBL(name) This operand specifies the name of the TLBL statement identifying the file on tape.

FILENAME(name) This operand specifies the same name as the FILE option on the SCRIPT command. FILENAME is used in place of a DLBL name to identify the 3800 printer.

Note: The (name) specified in the three operands above has the same characteristics as those of ddname, which is described in detail under TO and FROM operands in the "Commonly Used Command Operands" section of Chapter 2.

DEVICE This operand describes valid devices:

type specifies the type of device: 2314, 2319, 2400, 3310, 3330, 3340, 3350, 3370, 3375, 3800, DASD, or TAPE. Any supported direct-access device can be specified by indicating DASD. All tape devices supported by VSE are included by specifying either 2400 or TAPE. The default is either 2400 or printer (see Note below).

unit is a 1- to 3-digit number from the label of the logical unit. For example, unit value 005 means a logical unit of SYS005 defined by a JCS statement. The default is SYS001 to SYS014 or SYSLST (see Note below).

disposition (tape only) specifies the disposition of the tape volume as REWIND, UNLOAD, or NOREWIND. UNLOAD is the default.

Note: The actual defaults for "type" and "unit" are determined by which DLF command is associated with a particular ENVIRONMENT command and can be found in Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics" on page 237. A DLF command is associated with an ENVIRONMENT command if the name specified in the TLBL or DLBL operand of the ENVIRONMENT command also appears in the FROM or TO operand of the DLF command.

For example, if IMPORT is the associated DLF command, the ENVIRONMENT command implicit defaults for device would be:

type=2400
unit=SYS005

If SCRIPT is the associated DLF command (with the FILE option), the ENVIRONMENT command implicit defaults for device would be:

type=printer
unit=SYSLSST

See Figure 34 on page 241 for a listing of default values.

LABEL(tape only) This operand specifies that the tape volume has a label. This is the default.

NOLABEL(tape only) This operand specifies that the tape volume does not have a label.

FORMAT(format) This operand specifies the organization of the file:

F	Fixed, unblocked
FB	Fixed, blocked
V	Variable, unblocked
VB	Variable, blocked
S	Spanned, unblocked
SB	Spanned, blocked
U	Undefined

If you do not specify the FORMAT operand, the default is one of the following: VBT, V, FB, FM, F, or VB. See **Note:** below.

Note: The actual default is determined by which DLF command is associated with a particular ENVIRONMENT command and can be found in Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics" on page 237. A DLF command is associated with an ENVIRONMENT command if the name specified in the TLBL or DLBL operand of the ENVIRONMENT command also appears in the FROM or TO operand of the DLF command.

For example, if IMPORT is the associated DLF command, the ENVIRONMENT command default for format would be:

format(FB)

If SCRIPT is the associated DLF command (with the FILE option), the ENVIRONMENT command default for format would be:

format(FM)

See Figure 34 on page 241 for a listing of default formats.

BLOCKSIZE(number) This operand specifies the block size of the physical records. Enter the keyword followed by a decimal number from 32 to 32 767 (inclusive). (See Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics," for the default block-size of external data sets used by various DLF commands.)

RECORDSIZE(number) This operand specifies the maximum logical record size. Enter the keyword, followed by a decimal number from 32 to 32 767 (inclusive). (See Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics," for the default record-size of external data sets used by various DLF commands.)

Notes:

1. If an ENVIRONMENT command is not specified for an input or output file, default values are assumed. For these defaults, see Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics."
2. This command is required to describe the IBM 3800 Printing Subsystem when the SCRIPT command FILE option refers to an IBM 3800 Printer.
3. The block size, record size, and format cannot be changed (from the default as specified in Appendix B) for the COPY IN, COPY OUT, ARCHIVE, ACCOUNT, or RETRIEVE commands.
4. The DLBL, TLBL, or FILE required operands must be specified immediately following the ENVIRONMENT command verb.
5. If a fixed block architecture (FBA) device is being used, DLF uses a default CISIZE of 18 432. If you desire a smaller CISIZE for output commands such as EXPORT, you can specify it on the DLBL statement.

Examples

- This example identifies a DLBL name of CHAP1 for a data set residing on a 3330 disk. The logical unit is SYS005, and the data set format is variable-blocked, with a block size of 2 600 bytes. The maximum record size is 2 596. The document on the identified data set is read and stored into the document library with a document name of REPORT.

```
AUTH 100/WORKER
ENVIRONMENT DLBL(CHAP1) DEVICE(3330 005) -
          FORMAT(VB) BLOCKSIZE(2600) RECORDSIZE(2596)
IMPORT REPORT FROM(CHAP1)
```

- This example defines a tape with the TLBL name of NEWTAPE and a disposition of NOREWIND. The tape is unlabeled and is on logical unit SYS022. The data set format is fixed-blocked, with a block size of 800 and a maximum record size of 80. The document named REPORT is copied (with EXPORT) to this data set.

```
AUTH 100/WORKER
ENVIRONMENT TLBL(NEWTAPE) DEV(2400 022 (NOREWIND)) -
  NOLABEL BLOCKSIZE(800) RECORDSIZE(80) -
  FORMAT(FB)
EXPORT REPORT TO(NEWTAPE)
```

- This example identifies an IBM 3800 Printer as PRT3800 on logical unit SYS028. SCRIPT/VS formats the document named DOC1 and sends it to the 3800 printer.

```
AUTH 100/WORKER
ENVIRONMENT FILENAME(PRT3800) DEVICE(3800 028)
SCRIPT DOC1 ( TWO DEV(3800N8) FILE(PRT3800) -
  CHARS(GT12))
```

- This example identifies a DLBL name of D4250 for a file residing on disk. The output of the SCRIPT command for DOC1 is written to this file for later processing by the Composed Document Print Facility (CDPF) for final output on the IBM 4250 Printer.

```
AUTH 100/WORKER
ENVIRONMENT DLBL(D4250) DEV(DASD 007) -
  FORMAT(VB) BLOCKSIZE(2056) RECORDSIZE(2048)
SCRIPT DOC1 (TWO DEV(4250A) FILE(D4250))
```

- This example identifies a DLBL name of D3820 for a file residing on disk. The output of the SCRIPT command for NEWDOC is written to the file for later processing by a program product such as VSE/DITTO (VSE/Data Interfile Transfer, Testing and Operations Utility) for final output by Print Services Facility on the IBM 3820 Page Printer.

```
AUTH 100/WORKER
ENVIRONMENT DLBL(D3820) DEV(DASD 007) -
  FORMAT(VB) BLOCKSIZE(8205) RECORDSIZE(8201)
SCRIPT NEWDOC ( DEV(3820A) FILE(D3820))
```

EXPORT

This command copies a document from a general user's library to an external data set (tape or disk) or to a printer. DERs are not copied. A processor can also be called to process each record before it is exported. The EXPORT command deblocks the control intervals into individual records.

To understand the description of the EXPORT command that follows, the reader must have read and understood the material in the section "Commonly Used Command Operands" in Chapter 2.

EXPORT	docid [DATA(name)] [VERSION [(number)]] [PROCESS(name)] [TO ({ ddname } { DSN (dsname) } { DSN(dsname(member)) }) { DSN('dsname') } { DSN('dsname(member)') }]] [OUTDATA(name)] [OUTSOURCE(name)] [PARM('64 characters')]
---------------	---

- docid** This operand identifies the document exported.
- DATA(name)** This operand specifies the document's data type. The default is the data type in the current user's UPR.
- VERSION(number)** This operand specifies a particular version of the document exported. If VERSION is specified without (number) the latest version is exported. If VERSION is omitted for a document with versions, the latest version of the document is exported. These are the defaults. If VERSION is specified for a document without versions, an error message results and the document is not exported.
- PROCESS(name)** This operand specifies the entry name of a processor called for each record of the exported document. This processor supersedes any processor implied by input data type (DATA) and output data type (OUTDATA). See Chapter 8, "Using DLF to call Processors", for information about processors.
- TO** This operand identifies the external sequential data set or data set member receiving the data.
- If the TO operand is omitted, a ddname (OS/VS2) or DLBL/TLBL name (VSE) is constructed based on the first eight characters of the document name in OS/VS2 (first seven characters in VSE). These

characters are folded to uppercase and the special characters are converted to @ to arrive at the name. The data set must be defined in the JCL by a DD statement (OS/VS2) or DLBL/TLBL (VSE) statement with a name identical to this constructed name.

ddname specifies the name on the DD, DLBL, or TLBL JCL statement receiving the data. A DLBL or TLBL statement is not required if SYSLST (VSE) is specified and if SYSLST is permanently assigned.

DSN... is valid in OS/VS2 only. It indicates that the output data set is dynamically allocated. If the specified data set name is in quotation marks, it is used as given. Otherwise it is qualified as described in Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics..." SCRIPT is the suffix qualifier.

OUTDATA(name) This operand specifies the data type of an exported document after being processed by a special processor. This operand implies that a processor profile record (PPR) exists with a data type pair equal to the data type of the document and the OUTDATA name respectively. (See the DEFINE PROCESS command for additional information about the INDATA and OUTDATA pair.) If the implied PPR or its specified processor is not found, the document is not exported (however, see PROCESS keyword above). If OUTDATA is omitted, no search is made for a PPR.

If you do not specify the OUTDATA operand, no processor is implied. You can still explicitly specify a processor with the PROCESS(name) operand even if you did not specify OUTDATA.

OUTSOURCE(name) This operand specifies the name of the source attribute an exported document has after it is processed by a special processor. This operand has meaning only if OUTDATA(name) or PROCESS(name) is specified. If you omit OUTSOURCE or if you specify OUTSOURCE(0), no output source attribute is sent to the processor.

PARM('64 characters') This operand has an accompanying string up to 64 characters long enclosed in quotes. The string may include any values (the default is binary zeros). It is passed to the invoked processor in the initial call. You can invoke a processor in two ways:

1. Explicitly — using the PROCESS(name) operand.
2. Implicitly — through the presence of a processor profile record (created by a DEFINE PROCESS command) and the use of the appropriate pair of data types in the DATA and OUTDATA operands.

See "DEFINE PROCESS" on page 165 and Chapter 8, "Using DLF to Call Processors" on page 215 for more information.

Notes:

1. Input document library JCL statements are always required. The output document library JCL is required if there is a user exit that might change the application data. When output document library JCL is present, the input and output document libraries must be the same.
2. The OUTDATA, OUTSOURCE, and PARM parameters contain information that is passed to a processor. This information is not interpreted, analyzed, or used by DLF.

DLF ignores the PARM and OUTSOURCE operands if a processor has not been invoked (implicitly or explicitly).
3. When a special processor that was predefined using a DEFINE PROCESS command is invoked by the EXPORT command, the DATA attribute of the document (indicated in the DATA parameter of EXPORT) corresponds to the INDATA parameter of the processor; likewise, OUTDATA of EXPORT corresponds to OUTDATA for the processor.
4. Locked documents cannot be exported.
5. A general user can export documents from his own library and from any other library containing documents he can access. These include documents in any public library and in his project library. Also included are shared documents in any private or project libraries. If a document has a password, the general user must specify it.

Note: Administrators can use this command to export any document from any library.
6. The EXPORT command can be used to print a document by specifying a printer for the output data set.
7. The original document is not deleted after being exported. To delete a document, use the PURGE DOCUMENT command.
8. Using JCL, you must specify the characteristics of the external data set or use the default values. In VSE, if the characteristics of the external data set or the device type differ from the defaults, include an ENVIRONMENT command describing them.
9. If either SYSLST or SYSPCH in VSE is assigned to an output data set, the printer or punch control character will be present in the file. The VSE CLOSE JCS statement is required to close the punch.
10. The receiving file (tape or DASD) must have a LRECL equal to or greater than the largest record in the document. Otherwise, truncation will occur, resulting in lost data.

Examples

- This example exports a copy of the document named EXODUS. It is taken from library 12 345 and written to the data set identified by the DD statement OUTGO. The processor named EXPTPROC is invoked to process each record of the document before it is exported.

```
AUTH 12345
EXPORT EXODUS TO(OUTGO) -
  PARM('passed parameter') PROCESS(EXPTPROC)
```


- This OS/VS2 example exports a copy of the document STUDENT from library 12 to the data set named SCHOOL. The data set is dynamically allocated using the name SCHOOL because it was enclosed in quotation marks.

```
AUTH 12345
EXPORT 12 STUDENT TO(DSN('SCHOOL'))
```

- This example exports a copy of the document named TMEMO (with a password of TIME) from library 87 to the data set referred to by the DD statement, TCASE. If a PPR is found that has a data type pair equal to ATMS and ENCRYPT (as the INDATA/OUTDATA pair), a special processor is called before each record of the document is exported. The outsource attribute, CIPHER, is passed to the special processor.

```
AUTH 1234,
EXPORT 87 TMEMO/TIME TO(TCASE) DATA(ATMS) -
OUTDATA(ENCRYPT) -
OUTSOURCE(CIPHER)
```

IMPORT

This command takes a document from an external sequential data set, a partitioned data set, or the input stream immediately following the IMPORT command and puts it into a user's library. The input records can be modified by a processor before they are stored in a library by specifying the PROCESS keyword or by specifying an INDATA/DATA pair for which a processor profile record (PPR) is defined.

A user can import documents into his own library, his project library (if it is uncontrolled), and any uncontrolled public libraries.

To understand the description of the IMPORT command that follows, the reader must have read and understood the material in the section "Commonly Used Command Operands" in Chapter 2.

Operands of the IMPORT command are used to:

1. Identify the document (docid, DATA, VERSION).
2. Direct the IMPORT operation (FROM, REPLACE, FOR, SUPPRESS, CLUSTER).
3. Direct special processor actions (PROCESS, INDATA, INSOURCE PARM).
4. Associate various characteristics (SHARE, CLASS, SOURCE, DOCINFO, APPLINFO) with the document being imported.

DLF retains a document's characteristics and identification as part of the document. If these characteristics are not specified and the document being imported already exists (i.e., you are replacing it or adding another version), by default DLF applies the characteristics associated with the existing document to the document being imported.

If a document's characteristics are not specified and you IMPORT a new document, DLF normally defaults to null characteristics or the characteristics of the currently authorized user.

VERSION(number) This operand specifies the version number assigned to the imported document. See Figure 5 on page 61 for a flowchart describing how DLF tests version characteristics.

Version Characteristics:

- If this is the first import of a document and the number field contains a zero or the VERSION keyword is omitted, a version number of 0 is assigned and the document at this time has no versions.

If this is the first import of a document and the VERSION keyword is used without a number variable, a version number of 1 is assigned.

- If the document already exists in the document library as a document with versions, a version number is assigned if either the VERSION keyword or its number field is omitted. The version assigned is the latest version number plus one. If you include the VERSION keyword and its associated number, assign any valid version number that is higher than the most recent one. If a lower or equal version number is specified, the document will be imported with the latest version number plus one.
- If you specify VERSION(number) with a number greater than zero when importing a copy of a document that already exists in the selected library without versions (version number 0), the existing copy becomes version 1, and the current import becomes the specified version number. An exception to this exists when VERSION(1) is specified. In this case the current import is imported as version 2. This method changes the characteristics of a document from one without versions to one with versions. The REPLACE operand, if specified in this circumstance, is ignored.

Note: If you want to replace an existing document without versions with a new document with versions, issue a PURGE DOCUMENT command against the existing document followed by the import of the document with versions.

- If you specify VERSION(0) when importing a copy of a document that already exists in the selected library, an error message results and the new copy is not imported.
- If you omit the VERSION operand and a copy of this document without versions already exists in the document library, the current import does not occur unless REPLACE is also specified.

PROCESS(name) This operand specifies the entry name of a processor for each record of the imported document. (name) is 1 to 8 alphanumeric characters (and the characters \$, #, and @) which DLF folds to uppercase. (name) cannot begin with a numeric character. If PROCESS is not specified, and the imported document has an INDATA (processor INDATA) and DATA (processor OUTDATA) type that match a data pair associated with a processor profile record (PPR), the

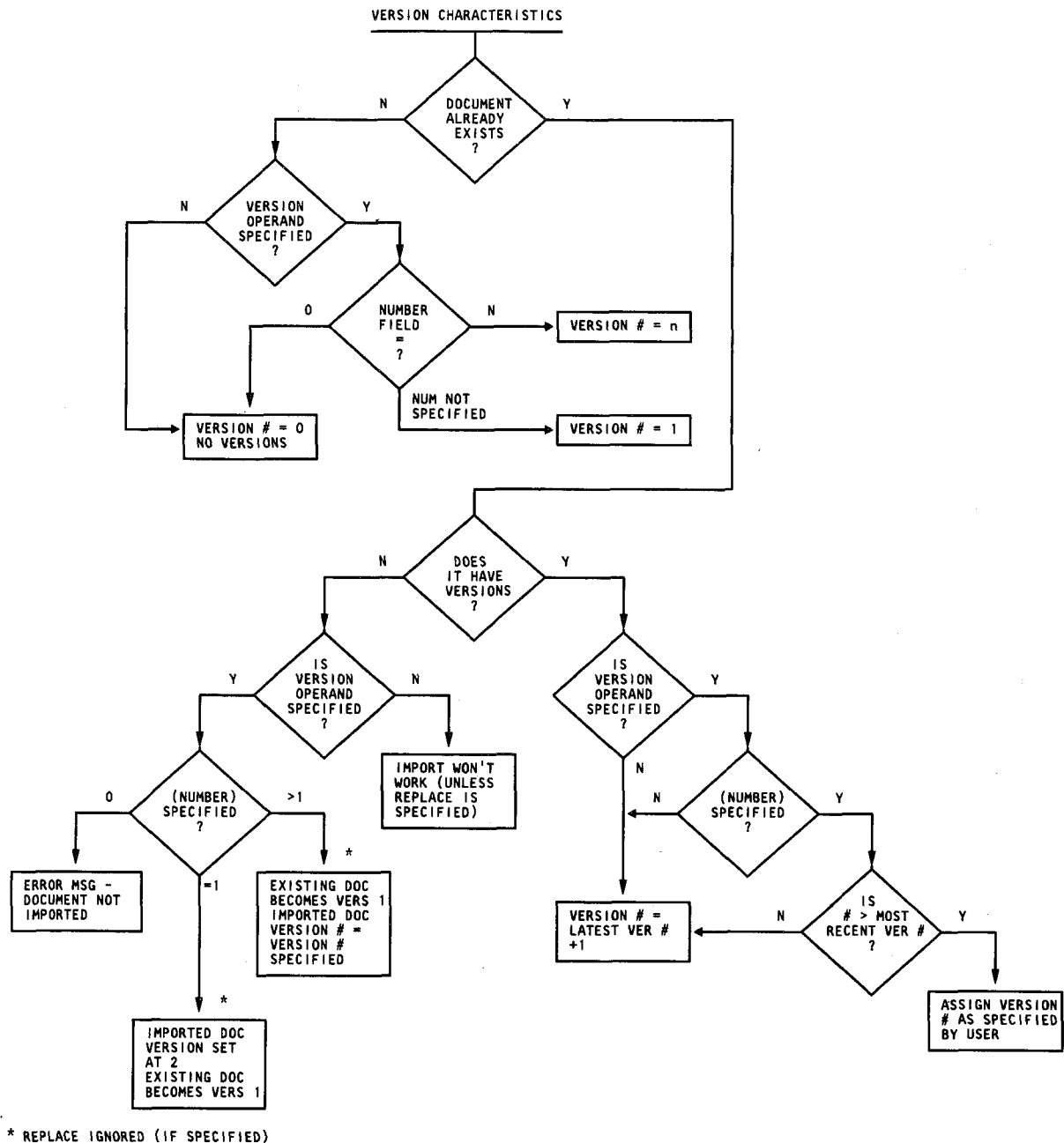


Figure 5. Flowchart of how DLF tests version characteristics.

FROM

processor indicated in that PPR is used. If PROCESS is not specified and no PPR is defined for the INDATA/DATA pair, no processor is used during this import.

This operand identifies the data set or partitioned data set member containing the imported data. The input records come from the processor if the FROM operand is omitted and a process has been specified by the PROCESS operand or implied by the INDATA operand. The

processor must either generate these records or obtain them from a source data set that it is set up to access. If a process has not been specified or implied and the FROM operand is omitted, an error message results and the data is not imported.

* indicates that the file to be imported is contained in the job input stream following this command. See "Putting Documents into Your Own Library" in Chapter 4 for information on mixing commands and data.

ddname is the name on the DD, DLBL, or TLBL JCL statement identifying the external sequential data set or member used for input.

DSN... is valid in OS/VS2 only. It indicates that the FROM data set is dynamically allocated. If the specified data set name is in quotation marks, it is used as given. Otherwise it is qualified as described in Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics." SCRIPT is the suffix qualifier.

REPLACE

This operand specifies that the imported document replaces any copy of this document without versions existing in the user's library. This operand cannot be used on documents with versions in the document library. The REPLACE operand is ignored if the imported document has versions and a document without versions by the same name already exists in the document library. See the discussion under the VERSION operand of this command for details.

NOREPLACE

This operand specifies that the imported document does not replace any existing copy of the document without versions. If a document without versions with the same name and data type exists in the user's library and NOREPLACE is specified, the document is not imported and an error message results. NOREPLACE is the default.

SHARE

This operand specifies that the imported document is shared by all users of the document library. If a shared document has no password, any user of the document library can access it. If a shared document has a password, any user of the library who knows the password can access the document. SHARE is the default for public libraries.

NOSHARE

This operand specifies that the document cannot be read by other users of the document library. It is the default for private and project libraries. However, users with access to the project library can read documents in that library without regard to share status.

CLASS(name)

This operand gives the class name used for this document. The name need not be predefined (by using a DEFINE CLASS command). If this is a new document and the CLASS(name) is omitted, the default class for the user (as specified in the UPR) is used. If this is not a new document, the class used in the previous version of the

document (or the document being replaced) is used.

See the discussion of the CLUSTER(name) operand below for additional information on how document class can affect the IMPORT command.

INDATA(name)

This operand specifies the current data type of the imported document. If this operand is specified, a processor is implied. The data type is indicated in the DEFINE PROCESS command corresponding to the DATA (processor OUTDATA) and INDATA (processor INDATA) pair in the IMPORT command. If the named processor is not located, the import is not performed.

If this operand is not specified, no processor is implied. You can still explicitly specify a processor with the PROCESS(name) operand even if INDATA is not specified.

INSOURCE(name)

This operand specifies the name of the system where the document originated. This operand is meaningful and allowed only if INDATA or PROCESS is specified. No input source is identified to the processor if the (name) field contains a zero or if the INSOURCE operand is not specified.

FOR(number)

This operand specifies the number of a user who is to be the owner of the document being imported. If FOR is not specified, the current user will be the owner. If FOR is specified, the library into which the document is to be imported (determined by specific document library number specification in the docid operand or implicitly as the current users library) must either be the FOR user's project library or a public library.

CLUSTER(name)

This operand specifies the name of a cluster into which this document is to be imported. If CLUSTER is not specified, there are three levels of default.

1. DLF first defaults to the cluster (if any) associated with the class (if any) specified on the IMPORT command.
2. DLF then defaults to the cluster (if any) associated with the default class (if any) of the currently authorized user.
3. DLF lastly defaults to the basic 4K cluster.

Different versions of a single document can be imported to different clusters.

Note: If a cluster is specified on an IMPORT command where REPLACE is also specified, the cluster specification is ignored because the replace must be done to the same cluster.

SOURCE(name)

This operand specifies a source name that is stored with the document. This name is passed to any processor invoked. If the SOURCE operand is omitted for a new document, DLF uses the default source of the currently authorized user. If this is not a new document and the SOURCE operand is omitted, DLF defaults to the source of the existing document (which you are adding a version to or replacing).

SUPPRESS

This operand directs DLF to delete trailing blanks from logical records prior to their being written to the document library. This is the default. See "Putting Documents into Your Own Library" in Chapter 4 for information on deleting trailing blanks from continued record segments.

NOSUPPRESS

This operand specifies that trailing blanks are maintained on all logical records written to the document library. See "Putting Documents into Your Own Library" in Chapter 4 for information on retaining trailing blanks on continued record segments.

PARM('64 characters') This operand has an accompanying string up to 64 characters long enclosed in quotes. The string may include any values (the default is binary zeros) and is passed to the invoked processor in the initial call. You can invoke a processor in two ways:

1. Explicitly — using the PROCESS(name) operand.
2. Implicitly — through the presence of a processor profile record (created by a DEFINE PROCESS command) and the use of the appropriate pair of data types in the DATA(name) and INDATA(name) operands.

See "DEFINE PROCESS" on page 165 and Chapter 8, "Using DLF to Call Processors" on page 215 for more information.

DOCINFO('50 characters') With this operand you specify a quoted string of up to 50 characters of additional information that you want to include in the DER.

For example, you might use the 50 characters to indicate when the document should be deleted from the library or a brief history of the document and who uses it.

The characters are retained in the DER of the imported document and the default is binary zeros. You can list this information using the INFO operand of the LIST DOCUMENT command.

APPLINFO('75 characters') This operand specifies a quoted string of up to 75 characters of any value. This information is placed in the DER of the imported document and contains application-oriented data. The default is binary zeros. The field is accessed or changed through a user exit routine. See Chapter 10, "User Exit Routines," for details. This field in the DER that is retained in the document library for an archived document can be overlaid with information about the location of the data set on which the document is archived. Overlay will occur if dynamic allocation is used for the archive data set or if either the OPERATOR or NOTE keywords are used on the archive command. The DER that is archived to the external sequential data set with the document itself will retain this APPLINFO field data. This field cannot be listed.

Notes:

1. The output document library JCL statements are required.
2. The maximum size for a document is 32 767 control intervals for the cluster in which the document is stored.
3. The INDATA, INSOURCE, and PARM parameters contain information that is passed to a processor. This information is not interpreted, analyzed, or used by DLF.

The INSOURCE and PARM operands cannot be used with the IMPORT command unless either the INDATA or the PROCESS operand is also specified.
4. The PROCESS(name) identifies a processor load module name. The load module is loaded and each record is passed to this load module during import processing. Specifying the PROCESS operand overrides any processor implied by the INDATA and DATA designation for the document.
5. If the INDATA operand is specified, a processor is implied. The processor entry point must be defined in a PPR. If a PPR with the necessary data type name pair is not found, the document is not imported and the command fails. No search is made for a PPR if the PROCESS operand is specified.
6. For VSE, use the ENVIRONMENT command to specify a data set's characteristics before importing a document if the default values for the installation are not applicable. See Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics," for further details.
7. To change the share status of a document, use the PROTECT command.
8. Each line of import data that follows an IMPORT command that is rejected with an error will be treated as a command.

Examples

- This example imports a document named CHAP2 into the private library of user 925 from a data set specified by the DD statement NEWDATA and assigns the document the password BLACK.

```
AUTH 925
IMPORT CHAP2/BLACK FROM(NEWDATA)
```

- This example imports an ATMS document named PEANUTS into private library 111 111. It is read from a data set identified by INDOC, replacing a document without versions named PEANUTS if PEANUTS exists in library 111 111. The document is assigned SHARE status. The records are converted by the load module named DSMACAIP. The document is stored in the document library with a data type of SCRIPT.

```
AUTH 111111
IMPORT PEANUTS FROM(INDOC) REPLACE INSOURCE(ATMS) -
INDATA(ATMS) SHARE PROCESS(DSMACAIP) DATA(SCRIPT)
```

- This example imports a document named CASHEWS from a sequential data set identified by INDOC1. The DER of this document is to contain the information "Nut Inventory" in the DOCINFO field. When the document is listed with the INFO operand specified, the DOCINFO field also is listed.

```
AUTH 111111
IMPORT CASHEWS FROM(INDOC1) DOCINFO('Nut Inventory')
LIST DOCUMENT SELECT (CASHEWS) INFO
```

- This VSE example imports a document into library 130 041 from a tape identified as TAPEIN. The document is stored in the library with a document name of STRATEGY.

```
// JOB      IMPORT FROM TAPEIN
// DLBL     DSMPDIR,'DSMLIB.DIRECTRY',,VSAM
// EXTENT   ,DSMVOL
// DLBL     DSMPLIB,'DSMLIB.SOURCE',,VSAM
// EXTENT   ,DSMVOL
// TLBL     TAPEIN
// ASSGN    SYS034,X'280'
// MTC      REW,SYS034
// OPTION   NOLISTX
// EXEC     DSMSPEXC,SIZE=AUTO
//          AUTH 130041/MYPWORD
//          ENV TLBL(TAPEIN) DEV(2400 034) NOLABEL
//          IMPORT STRATEGY FROM(TAPEIN)
/*
/8
```

- This VSE example imports a document into library 130 041 from a card reader identified as SYSIPT. The document is stored with a document name of STRATEGY.

```
// JOB      IMPORT FROM CARD READER
// DLBL     DSMPDIR,'DSMLIB.DIRECTRY',,VSAM
// EXTENT   ,DSMVOL
// DLBL     DSMPLIB,'DSMLIB.SOURCE',,VSAM
// EXTENT   ,DSMVOL
// ASSGN    SYSIPT,X'00C'
// OPTION   NOLISTX
// EXEC     DSMSPEXC,SIZE=AUTO
//          AUTH 130041/MYPWORD
//          IMPORT STRATEGY FROM(SYSIPT)
//          /* DOCUMENT TO BE IMPORTED IS HERE */
/*
/*
/8
```

Notes:

1. The first /* after the comment marks the end of the imported document, and the second /* marks the end of the job step.
2. Because DLF uses separate control blocks to manage SYSRDR and SYSIPT, you can assign (ASSGN SYSXXX, cuu) either, but not both, SYSRDR and SYSIPT to a DASD file. For example, do not assign SYSIN (ASSGN SYSIN, cuu) to a DASD file in a VSE environment.

LIST

The LIST command lists information from various profiles in the directory data set.

There are five list commands for general users:

- LIST CLASS
- LIST CLUSTER
- LIST DOCUMENT
- LIST PROCESS
- LIST USER.

Notes:

1. See Appendix F, "Output of LIST Commands" on page 253 for an example of output from LIST CLUSTER, LIST DOCUMENT, and LIST USER.
2. For LIST commands with selection criteria, the list becomes increasingly more specific (and shorter) as the number of criteria is increased.

LIST CLASS

This command lists class names and associated clusters from all the class entry records (CERs) defined in the directory data set.

LIST CLASS	[FROMLIB ({ INPUT }) { OUTPUT }]
------------	---

FROMLIB

Use this operand to specify which document library's class entry records you want to list. Your options are:

INPUT Indicates that you want to list all of the CERs in the input document library. This is the default.

OUTPUT Indicates that you want to list all of the CERs in the output document library.

Notes:

1. Input document library JCL statements are required:
 - If you specify FROMLIB(INPUT)
 - If you omit the optional FROMLIB operand (allowing the command to default to INPUT).
 - Output document library JCL statements are required if you specify FROMLIB(OUTPUT).

Example

This example lists all classes and their associated clusters defined in the input document library's directory data set.

```
AUTH 100/WORKER
LIST CLASS
```

LIST CLUSTER

This command lists all the cluster profile records (CPRs) defined in the directory data set.

LIST CLUSTER	<div data-bbox="816 327 1219 404">FROMLIB ({ INPUT }) { OUTPUT }</div> <div data-bbox="816 421 981 497">SPACE NOSPACE</div>
--------------	---

- FROMLIB** Use this operand to specify which document library's cluster profile records you want to list. Your options are:
- INPUT** Indicates that you want to list all of the CPRs in the input document library. This is the default.
- OUTPUT** Indicates that you want to list all of the CPRs in the output document library.
- SPACE** Use this operand to generate the Space Usage report for clusters rather than the Standard report that you would get by default.
- NOSPACE** Use this operand to generate the Standard report for clusters. This is the default.

Notes:

1. Input document library JCL statements are required:
 - If you specify FROMLIB(INPUT)
 - If you omit the optional FROMLIB operand (allowing the command to default to INPUT).
2. Output document library JCL statements are required if you specify FROMLIB(OUTPUT).
3. There are two LIST CLUSTER reports:
 - The Standard report - generated by default or by specifying NOSPACE on the LIST CLUSTER command.
 - The Space Usage report - generated by specifying SPACE on the LIST CLUSTER command.

Both reports include the cluster name, input and output ddnames, and the control interval size.

In addition, the Space Usage report shows the cluster size, the amount of the cluster already used, and the amount available for use. The available space is divided into space that was never used (contiguous space at the end of the cluster) and reusable space (space formerly occupied by documents and now probably fragmented).

4. The LIST CLUSTER command lists "CNTL INTRVL SIZE" in bytes. All other space usage data is listed in thousands of bytes rounded to the nearest thousand bytes.
5. All percentages are rounded with the exception of those values between 0% and 1% which are rounded to 1% and those values between 99% and 100% which are rounded to 99%.

6. See Appendix F, "Output of LIST Commands" on page 253 for an example of the Space Usage report and Chapter 7, "Administering the Document Library" on page 205 for more information about the space usage statistics provided by the LIST CLUSTER and LIST DOCUMENT commands.
7. When you request the Space Usage report, any cluster activation failure messages will appear before the Space Usage report and the failing clusters will be identified in the Space Usage report.

Examples

- This example creates the Standard report, listing all CPRs identifying the VSAM data sets in the input document library.

AUTH 100/WORKER
LIST CLUSTER

- This example requests the Space Usage report for LIST CLUSTER. The report will list data from all CPRs and their associated Cluster Control Blocks (CLBs) in the output document library.

AUTH 100/WORKER
LIST CLUSTER SPACE FROMLIB(OUTPUT)

LIST DOCUMENT

This command lists information from the document entry records (DERs) for the documents assigned to either the current library user or to the library user(s) specified in the USER operand.

To understand the description of the LIST DOCUMENT command that follows, the reader must have read and understood the material in the section "Commonly Used Command Operands" in Chapter 2.

LIST DOCUMENT

[FROMLIB ({ INPUT })
 { OUTPUT }]

[{ ddname }
 { DSN(dsname) }
TO ({ DSN(dsname(member)) })
 { DSN('dsname') }
 { DSN('dsname(member)') }]

[USER ({ number }
 { number1:number2 })]

[SELECT ({ dname }
 { dname1:dname2 }
 { 'dname' }
 { 'dname1':'dname2' })]

[DATED ({ date }
 { date1: }
 { date1:date2 }
 { :date2 })]

[CLASS(name)]

[DATA(name)]

[CLUSTER(name)]

[SOURCE(name)]

[ALL
OWN]

[SPACE
NOSPACE]

[LOCK]

[ARCHIVE]

[VERSION]

[INFO]

FROMLIB

Use this operand to specify which document library's document entry records you want to list. Your options are:

INPUT Indicates that you want to list all of the DERs (that also meet the other selection criteria) in the input document library. This is the default.

OUTPUT Indicates that you want to list all of the DERs (that also meet the other selection criteria) in the output document library.

TO

This operand identifies the external sequential data set or data set member receiving the data. If the TO operand is omitted, the data is written to a sequential data set with the ddname of DSMLIST (for OS/VS2) or SYSLIST (for VSE).

ddname is the name on the DD, DLBL, or TLBL JCL statement receiving the data.

DSN... is valid in OS/VS2 MVS only. It indicates that the list data set is dynamically allocated. If the specified data set name is in quotation marks, it is used as given, otherwise it is qualified as described in Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics.." LIST is the suffix qualifier.

USER

This operand specifies the user(s) whose libraries are searched for DERs meeting the other selection criteria for the list. The default is to search the current user's private library, project library, and any public library in which he owns documents.

number specifies a specific user's library searched for DERs meeting the other selection criteria for the list. General users can specify their own library, their project library, or any public library.

number1:number2 specifies a range of users' libraries to be searched for DERs meeting other selection criteria for the list. Only the current general user's own library, his project library, and any public library within the range are accessed.

SELECT

This operand specifies that selection is by document name. If not specified, all documents meeting other selection criteria are listed. Document names enclosed in quotes can contain any character and will not be folded to uppercase.

dname specifies that only the document(s) with this document name are selected for the list.

dname1:dname2 specifies that documents whose names fall in the inclusive alphabetical range are selected for the list.

DATED

This operand specifies that only documents stored in the document library on a specific date or between an inclusive range of dates are selected for the list. The default is to select documents stored on any or all dates that meet the other selection criteria. The date is specified in the

form mm/dd/yy where mm is the month, dd is the day of the month, and yy is the last two digits of the year.

date specifies the given date.

date1: specifies the given date and all subsequent dates.

date1:date2 specifies an inclusive range of dates.

:date2 specifies the given date and all preceding dates.

CLASS(name)	This operand specifies that only documents with the given class are to be listed. The default is all classes.
CLUSTER(name)	This operand specifies that only documents in the given cluster are to be listed. The default is all clusters.
DATA(name)	This operand specifies that only documents with the given data type are listed. The default is all data types.
SOURCE(name)	This operand specifies that only documents with the given source attribute are listed. The default is all source attributes.
ALL	This operand indicates that all documents in the designated libraries that meet other selection criteria are to be listed. That is, a user can list all documents he can read.
OWN	This operand specifies that only documents owned by the current user in the designated libraries are to be listed. This is the default.
SPACE	This operand specifies that you want to generate the Space Usage report rather than the Standard report you would receive by default. If you specify SPACE, you can specify any of the current LIST DOCUMENT options, but the INFO option would be ignored. If you specify both the ARCHIVE option and the SPACE option, the archived documents are listed and show the amount of physical space that would be used if the documents were retrieved; but the summary page does not include information about documents that are archived. The summary page is not generated unless more than one NON-ARCHIVED document is listed.
NOSPACE	Use this operand to indicate that you want the Standard report for documents. This is the default.
LOCK	This operand indicates that locked documents are listed as well as unlocked ones. The default is to list only unlocked documents.
ARCHIVE	This operand specifies that any archived documents, the date of their archive, the time of day, and the contents of the DERNOTE field are listed. The default is to list only document(s) that are not archived. If you also specify the SPACE option, archived documents are listed but not included in the summary statistics.
VERSION	This operand specifies that all versions of the document(s) are listed. The default is to list only the latest version of the document(s).

INFO

This operand specifies that the listing includes the information from the DERINFO field (specified as DOCINFO on the IMPORT command) of the document's DER if such information exists. The default is not to list the information in the INFO field. This operand is ignored if you specify the SPACE option.

Notes:

1. Input document library JCL statements are required:
 - If you specify FROMLIB(INPUT)
 - If you omit the optional FROMLIB operand (allowing the command to default to INPUT).
2. Output document library JCL statements are required if you specify FROMLIB(OUTPUT).
3. If the command is specified without any operands, information from the DERs of the latest, or only, version of all documents owned by the current user (which are not locked or archived) are listed. This is true for documents in the user's own library, his project library, and any public libraries in which he owns documents.
4. A general user can list information from the DERs of all documents in his private library, all documents in his project library, and all documents in any public library in which he owns documents.
5. There are two types of LIST DOCUMENT reports:
 - The Standard report - generated by default or by specifying NOSPACE on the LIST DOCUMENT command.
 - The Space Usage report - generated by specifying SPACE on the LIST DOCUMENT command.

Both reports display space in bytes and include: library number, document name, class and data type, version number, owner ID, and the name of the cluster in which the document is stored. Each report has a summary line indicating the number of documents listed.

The Standard report also includes the document source, control and archive indicators, creation date, file size, and number of records.

The additional information available in the Space Usage report shows the control interval SIZE of the cluster, the COUNT of control intervals a document is using, and the percentages with the number of bytes that are FILLED and VACANT. If there is more than one NON-ARCHIVED document listed, a summary page is also generated. It provides a count and other summary information about ONLY the NON-ARCHIVED documents listed. This includes the SMALLEST, AVERAGE, and LARGEST document sizes and totals for the number of CTL-INTRVLs and RECORDS. The summary page also lists a total number of TEXT BYTES that excludes the bytes used by DLF for control purposes. This is essentially a total for the field called FILE SIZE on a Standard LIST DOCUMENT report (excluding any archived documents). The last lines on the summary page are totals and overall percentages for the FILLED, VACANT, and TOTAL fields on the Space Usage report.

6. See Appendix F, "Output of LIST Commands" on page 253 for examples of the two reports and Chapter 7, "Administering the Document Library" on page 205 for a more detailed explanation of the space usage statistics from the LIST DOCUMENT and LIST CLUSTER commands.

7. Alphabetical order for listing is blank, \$, #, @, a-z, A-Z, and 0-9.
8. The listing can show unexpected documents from a library that is neither the user's private library, his project library, nor any public library. If this problem occurs, notify your administrator to handle these ghost DERs.

Examples

- This example lists the Standard report information from the DERs of documents owned by user 8 946 in the input document library. The example lists all documents (latest or only version of each) owned by this user that are not locked or archived. These documents can be stored in this user's private library, his project library, or in any public library in which this user owns documents. The list is written to DSMLIST (for OS/VS2) or to SYSLIST (for VSE).

```
AUTH 8946
LIST DOCUMENT
```

- This example requests the Space Usage report for LIST DOCUMENT and specifies that only documents in the cluster named DSMSTLIB are to be listed.

```
AUTH 18
LIST DOCUMENT CLUSTER(DSMPTLIB) SPACE
```

- This example lists the Standard report information from the DERs of documents that are stored in any library in which user 18 owns documents in the output document library. The example lists all documents (latest or only version of each) in these libraries that are not locked or archived, regardless of ownership. These documents are stored in user 18's private library, his project library, or in any public library in which he owns documents. The list is written to DSMLIST (for OS/VS2 MVS) or to SYSLIST (for VSE).

```
AUTH 18
LIST DOCUMENT ALL FROMLIB(OUTPUT)
```

- This example lists the Standard report information from the DERs of all documents in the input document library that:
 - Have a document name that falls in the alphabetic range from CONNIE through FRANCES inclusive.
 - Are stored in user 101's private library, project library, or any public library in which user 101 owns documents.
 - Were stored into the document library on or after October 11, 1980.
 - Have the data type name of SCRIPT.

```
AUTH 101
LIST DOCUMENT SELECT(CONNIE,FRANCES) -
DATED(10/11/80;) DATA(SCRIPT) ALL
```

- This example lists information from the DERs of all documents (including those that are locked or archived) owned by user 101 in the input document library. These are stored in any private, project, or public library in which user 101 owns documents. All versions of each document meeting this criteria are listed.

```
AUTH 101
LIST DOCUMENT ARCHIVE LOCK VERSION
```

LIST PROCESS

This command lists information from all the processor profile records (PPRs) that are defined to the document library.

LIST PROCESS	[FROMLIB ({ <u>INPUT</u> }) { <u>OUTPUT</u> }]
--------------	---

FROMLIB

Use this operand to specify which document library's processor profile records you want to list. Your options are:

INPUT Indicates that you want to list all of the PPRs in the input document library. This is the default.

OUTPUT Indicates that you want to list all of the PPRs in the output document library.

Notes:

1. Input document library JCL statements are required:
 - If you specify FROMLIB(INPUT)
 - If you omit the optional FROMLIB operand (allowing the command to default to INPUT).
2. Output document library JCL statements are required if you specify FROMLIB(OUTPUT).

Example

This example lists information from all PPRs previously defined to the output document library.

```
AUTH 123/WORKER
LIST PROCESS FROMLIB(OUTPUT)
```

LIST USER

This command lists information from the indicated user profile record (UPR). As a general user, you can only list information from the UPR of your own library, project library, or any public library.

To understand the description of the LIST USER command that follows, you must first read and understand the material in the section "Commonly Used Command Operands" in Chapter 2.

LIST USER	<div>[{ number } { number1:number2 } { ALL }]</div> <div>[FROMLIB ({ INPUT }) { OUTPUT }]</div>
-----------	---

number	Specifies a particular user.
number1:number2	Specifies an inclusive range of users. Information is listed only from the UPRs in this range that define libraries you are authorized to access. These include the UPRs of your own library, project library, and any public library.
ALL	Indicates that you want to list information about all DLF users in the range 1:9999999 whose libraries you are authorized to access.
FROMLIB	Use this operand to specify which document library's user profile records you want to list. Your options are: INPUT Indicates that you want to list all of the UPRs (that also meet the other selection criteria) in the input document library. This is the default. OUTPUT Indicates that you want to list all of the UPRs (that also meet the other selection criteria) in the output document library.

Notes:

1. Input document library JCL statements are required:
 - If you specify FROMLIB(INPUT)
 - If you omit the optional FROMLIB operand (allowing the command to default to INPUT).
2. Output document library JCL statements are required if you specify FROMLIB(OUTPUT).
3. If you do not specify number, number1:number2, or ALL, the current user's own UPR is the default.
4. The LIST USER command lists space in units of 1 000 bytes. "SPACE IN USE" and "PUB/PRJ" rounded upward to the nearest 1 000 bytes.
5. Refer to Appendix F, "Output of LIST Commands" on page 253 for an example of the output from a LIST USER command.

6. Zero (0) specified as the only user number refers to the current user's project library. Zero (0) specified as the first number of a range causes DLF to assume you want to search from the beginning (user #1) of the document library.
7. Note that input document library JCL statements are required.

Examples

- This example lists the UPR of user 100 in the input document library:

```
AUTH 100/WORKER  
LIST USER
```

- This example lists the UPRs of user 100's private library, project library, and all public libraries defined in the output document library:

```
AUTH 100/WORKER  
LIST USER ALL FROMLIB(OUTPUT)
```

PASSWORD

With this command, the current user can alter his password and update his UPR. This command can only change the password of the current user.

PASSWORD	{	/password	}
	{	/'password'	}
	{	NOPASS	}

/password
or
/'password'

The /password or /'password' operand specifies the new password for the current user. The user gave his original password on the previous AUTH command and need not enter it with this command. User passwords have the same characteristics as document passwords. See dpass under "Commonly Used Command Operands" in Chapter 2 for the characteristics of a password.

NOPASS

This operand specifies that the current user's password is deleted from his UPR.

Notes:

1. The output document library JCL statements are required.
2. If a password is assigned to a library user, it must be specified on all subsequent AUTH commands for that user.

Examples

- This example changes the password for user 18 from PASSIT to STOPPER.

```
AUTH 18/PASSIT
PASSWORD /STOPPER
```

- This example changes the password for user 20 from UNQUOTE to QUOTEDφ.

```
AUTH 20/UNQUOTE
PASSWORD /'QUOTEDφ'
```

- This example removes user 18's password.

```
AUTH 18/STOPPER
PASSWORD NOPASS
```

PROTECT

This command changes the accessibility of a general user's documents. The documents he owns or the documents in the library he owns (regardless of document ownership) are affected.

To understand the description of the PROTECT command that follows, the readers must have read and understood the material in the section "Commonly Used Command Operands" in Chapter 2.

PROTECT	docid [DATA(name)] [VERSION(number)] [PASSWORD({/password }) {/password'} NOPASS] [SHARE NOSHARE]
----------------	---

docid This operand identifies the document whose access status is changed.

DATA(name) This operand specifies the document's data type. The default is the data type in the current general user's UPR.

VERSION(number) This operand shows the version of the document whose access status is changed. The default is the latest version. If the version is not found, the document is not altered, and a warning message is issued.

PASSWORD Specifies whether you intend to supply a new password for this document (PASSWORD) or if you want the existing password deleted (NOPASS).

NOPASS If you specify PASSWORD, supply a password in the form of (/password or /'password'). If you want to delete or remove the password, specify NOPASS.

If you omit the PASSWORD/NOPASS operand pair, the document's password does not change.

SHARE This operand specifies that the document is made shareable. The default is the current share status of the document. Documents with share status may be accessed by any valid document library user who knows the passwords.

If you omit the SHARE/NOSHARE operand pair from the PROTECT command, no change is made to the share status of the document.

NOSHARE This operand specifies that the document is no longer to have share status.

If you omit the SHARE/NOSHARE operand pair from the PROTECT command, no change is made to the share status of the document.

Notes:

1. The output document library JCL statements are required.
2. The PROTECT command cannot change the accessibility of locked or archived documents.
3. General users can change the password or share status only of the documents they own, with the following exception: the owner of a project or public library can change the password or share status of any document in his library, regardless of ownership.
4. If SHARE is specified for documents in a project library, it extends document accessibility to those who are not members of the project.
5. SHARE and NOSHARE have no effect when specified for documents in a public library because all documents in a public library are automatically sharable.
6. NOSHARE is the default for documents being placed in private or project libraries with the IMPORT command. NOSHARE status assigned to a document in a project library limits access to the project members.

EXAMPLES

- This example assigns the password HIDEIT to the document named ACCOUNT and specifies that it is not to be shared.

AUTH 35/SAFE
PROTECT ACCOUNT PASSWORD(/HIDEIT) NOSHARE
- This example removes the password from the document RECORDS and makes it accessible to all users of the document library.

AUTH 35/SAFE
PROTECT RECORDS NOPASS SHARE

PURGE

The PURGE commands remove profiles and documents from the document library.

Only one PURGE command can be executed by general users:

- PURGE DOCUMENT.

All other PURGE commands require administrator authority.

PURGE DOCUMENT

This command eliminates documents from a user's library.

To understand the description of the PURGE DOCUMENT command that follows, the reader must have read and understood the material in the section "Commonly Used Command Operands" in Chapter 2.

PURGE DOCUMENT	docid [DATA(name)] [VERSION ({ number { number1:number2 } } { ALL })]
-----------------------	--

- docid** This operand identifies the document to be purged. General users can specify their own, their project, or any public library in which they own documents as the dlib# (part of the docid). If the document to be purged has a password, it must be specified.
- DATA(name)** This operand specifies the data type of the document to be purged. The default is the current authorized user's default data type located in his UPR.
- VERSION** This operand specifies the version(s) of the purged document. If this operand is omitted and the document has versions, the document is not purged. If the operand is specified and the document is without versions, an error message results and the document is not purged. The VERSION operand without a value is invalid for this command.
- number** indicates the specific version of the document to be purged.
- number1:number2** indicates an inclusive range of versions of the document to be purged.
- ALL** specifies that all versions of the document are to be purged.

Notes:

1. The output document library JCL statements are required.
2. A general user can purge any documents he owns. He also can purge any documents in the library he owns regardless of document ownership.
3. Locked documents cannot be purged.
4. The DERs of archived documents can be purged from the directory data set. The archived documents themselves and their DERs remain on the data set to which they were

archived. They can no longer be retrieved, however, because the document library no longer contains the information required to retrieve them.

5. If a document has a password, it must be specified.
6. Only the owner of a controlled project or public library, or an administrator, can purge documents from that library.

Example

This example purges version 2 of the document named HISTORY with a data type of SCRIPT, from user 100's own library.

```
AUTH 100/WORKER  
PURGE DOCUMENT HISTORY DATA(SCRIPT) VERSION(2)
```

READ

This command is used only by application programs that call DLF as a subroutine. The command is passed from the calling application program to DLF by way of a parameter list and requests that DLF pass to the application program the next (or first) record in the specified document. The first time the command is issued for a specific document, DLF sets up a control area from which it manages the sequential retrieval of this document's records from the document library. This procedure is called opening a document file. After the document file is opened, the first record is retrieved and passed back to the calling program. Subsequent retrievals for this document will retrieve the next record from the document library and pass it back to the calling program. The calling program issues a CLOSE command (also by way of a parameter list) when processing on the document is complete. This allows DLF to close the document file (free up its control area). For details on how this command is passed to DLF, see Chapter 9, "Using DLF as a Subroutine."

To understand the description of the READ command that follows, the reader must have read and understood the material in the section "Commonly Used Command Operands" in Chapter 2.

READ	docid
	[DATA(name)]
	[VERSION [(number)]]
	[OUTDATA(name)]
	[OUTSOURCE(name)]
	[PROCESS(name)]
	[PARM('64 characters')]

docid This operand identifies the document to be read.

DATA(name) This operand specifies the data type of the document to be read (such as SCRIPT or ATMS). The default is the data type in the user's UPR.

VERSION(number) This operand specifies the particular version of the document that is read. If the operand or the (number) field is omitted, the latest version of the document will be read. If VERSION is specified for a document that has no versions, the read does not occur and an error is indicated.

OUTDATA(name) This operand specifies the name of the output data type (such as SCRIPT or ATMS) that the document has after its records were converted by a processor. This processor is called for each record of the document being read and passed to the calling program. The OUTDATA operand implies that a PPR is defined with a data type name pair that matches the names specified or implied in the DATA (processor INDATA) and OUTDATA (processor OUTDATA) operands of this command. If the PPR is not found, the document is not read. If this operand is omitted or if a processor is explicitly

called for with the PROCESS operand in this command, no search is made for a PPR.

If you do not specify the OUTDATA operand, no processor is implied. You can still explicitly specify a processor with the PROCESS(name) operand even if you did not specify OUTDATA.

OUTSOURCE(name) This operand specifies the source name of the system for which the read and processed document is intended. This operand has meaning only if OUTDATA(name) or PROCESS(name) is specified. Binary zeros are passed to the processor as the outsource designation if the (name) field contains 0 or if OUTSOURCE is not specified.

PROCESS(name) This operand specifies an entry name of a processor for each document. This processor supersedes any other implied by the DATA/OUTDATA combination (equivalent to INDATA/OUTDATA of a predefined processor). This name is 1- to 8-characters long and can contain alphanumeric characters and the characters \$, #, and @; but the PROCESS name must not begin with a numeric character. The program will fold the characters to upper case.

PARM('64 characters') This operand has an accompanying string up to 64 characters long enclosed in quotes. The string may include any values (the default is binary zeros) and is passed to the invoked processor in the initial call. You can invoke a processor in two ways:

1. Explicitly — using the PROCESS(name) operand.
2. Implicitly — through the presence of a processor profile record (created by a DEFINE PROCESS command) and the use of the appropriate pair of data types in the DATA and OUTDATA operands.

See "DEFINE PROCESS" on page 165 and Chapter 8, "Using DLF to Call Processors" on page 215 for more information.

Notes:

1. The input document library JCL statements are required.
2. The OUTDATA, OUTSOURCE, and PARM operands contain information that is passed to a processor. This information is not interpreted, analyzed, or used by DLF.

DLF ignores the PARM and OUTSOURCE operands unless a processor has been invoked (implicitly or explicitly).
3. Each time this command is executed, one record is retrieved and placed in an area identified by the parameter list.
4. A return code of 4 in the parameter list indicates the end of the document. For more details, see Chapter 8, "Using DLF to Call Processors."
5. If PROCESS is not specified and if the OUTDATA keyword is not specified, no processor is given control prior to passing the record to the calling program. If the OUTDATA keyword is specified, a processor for processing each record is implied. A PPR must be created (with the DEFINE PROCESS command) that associates a pair of data type names with a processor. The first of this pair (INDATA) must be identical to the document's DATA designation, and the second of the pair (OUTDATA) must be identical to the OUTDATA name specified in the READ command.

6. DLF initially obtains enough working space (control and buffer area) for one document only. If only one document is read from the beginning to the end, then only the initial working space is required. If multiple documents are read concurrently, additional working space is required. A maximum of 16 documents can be open concurrently. If only part of a document is to be read, the CLOSE command must be issued for that document in order to free the working space. If a CLOSE command is not issued following one or more READ commands issued against a document, the working space for that document remains in main storage. If a SCRIPT command is then issued for that same document, SCRIPT command processing begins with the line following the last line read. The SCRIPT command closes the document after processing so any subsequent READ command issued against that document will start at the beginning of the document. Be sure to issue a CLOSE command after you issue the last READ command against a document if you intend to SCRIPT the same document immediately.
7. Because both READ and SCRIPT commands use the same storage for the 16 maximum open files, each document opened by the READ command reduces the possible level of nested imbeds within a SCRIPT operation.

Example

This example reads one record at a time from the document called STATUS that has a password of ONLYJOHN and resides in public library 123. Each record is passed to a processor previously associated with a data type pair of ATMS and SCRIPT.

```
AUTH 100/WORKER  
READ 123 STATUS/ONLYJOHN DATA(ATMS) OUTDATA(SCRIPT)
```

RETRIEVE

There are three RETRIEVE commands library users can execute:

- RETRIEVE ALL
- RETRIEVE DOCUMENT
- RETRIEVE NAMELIST.

These commands are used for returning archived documents to the document library. Documents that were locked at the time they were archived can be recovered; however, they remain locked after they are returned to the document library.

To understand the description of the RETRIEVE commands that follow, the reader must understand the material in the section "Commonly Used Command Operands" in Chapter 2.

RETRIEVE ALL

This command recovers all selected documents in the FROM data set (an external data set or partitioned data set member). All archived versions of the selected document will be retrieved. General users can retrieve any document that was archived from their own library, whether they own them or not. For VSE, use the ENVIRONMENT command to specify the archive data set's characteristics if the default values for the installation are not applicable. See Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics," for further details.

RETRIEVE ALL	<pre> { ddname } { DSN(dsname) } FROM ({ DSN(dsname(member)) }) { DSN('dsname') } { DSN('dsname(member)') }</pre> <p>[DATA(name)]</p> <p>[DATED(date)]</p>
--------------	--

FROM

This required operand identifies the external sequential data set or partitioned data set member that contains the archived documents.

ddname is the name on the DD, DLBL, or TLBL JCL statement defining the data set from which the documents are retrieved.

DSN... is valid in OS/VS2 MVS only. It indicates that the FROM data set is dynamically allocated. If the specified data set name is in quotation marks, it is used as given. Otherwise it is qualified as described in Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics." ARCHIVE is the suffix qualifier.

DATA(name) This operand specifies the data type of the documents retrieved. The default is all data types that meet other selection criteria.

DATED(date) This operand specifies that only documents originally stored in the document library on this particular date are to be retrieved. If DATED is

not specified, all documents meeting other selection criteria are retrieved. (date) must be in the form mm/dd/yy where mm is the month (1 to 12), dd the day of month (1 to 31), and yy the last two digits of the year.

Notes:

1. The output document library JCL statements are required.
2. The FROM operand is required in all systems.
3. Only a specific date is allowed in the DATED operand, not a range of dates as can be specified with the ARCHIVE ALL command.

Example

This example retrieves all documents (without regard to date stored) previously archived to the sequential data set defined by the DD (OS/VS2) or DLBL/TLBL (VSE) JCL statement named ARCDs. Because the user is not an administrator, only the documents he owns are retrieved.

```
AUTH 100/WORKER
RETRIEVE ALL FROM(ARCDs)
```


RETRIEVE DOCUMENT

This command retrieves a single document. For VSE, use the ENVIRONMENT command to specify a data set's characteristics before retrieving a document if the default values for your installation are not applicable. See Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics," for further details.

RETRIEVE DOCUMENT	<div data-bbox="867 405 948 432">docid</div> <div data-bbox="867 457 1084 512">[DATA(name)]</div> <div data-bbox="867 529 1211 583">[VERSION [(number)]]</div> <div data-bbox="867 600 1463 751"> <div data-bbox="899 617 1435 743"> <div data-bbox="1013 617 1403 644">{ ddname }</div> <div data-bbox="1013 644 1403 672">{ DSN(dsname) }</div> <div data-bbox="899 672 1435 699">FROM ({ DSN(dsname(member)) })</div> <div data-bbox="1013 699 1403 726">{ DSN('dsname') }</div> <div data-bbox="1013 726 1403 751">{ DSN('dsname(member)') }</div> </div> </div> <div data-bbox="867 768 1052 823">[OPERATOR]</div>
-------------------	---

docid This operand identifies the document to be retrieved.

DATA(name) This operand specifies the name of the data type of the document to be retrieved. The default is the data type specified in the current user's UPR.

VERSION(number) This operand gives the version number of the document retrieved. If VERSION or number is omitted, the latest version of the document is retrieved.

FROM This operand identifies the external sequential data set or partitioned data set member from which the archived documents are retrieved. In OS/VS2 only, if the FROM operand is omitted the document is retrieved from a dynamically allocated data set whose full name is found in the DERNOTE field of the DER. This parameter is valid only if the archived document was sent to the same dynamically allocated data set.

ddname is the name on the DD, DLBL, or TLBL JCL statement defining the data set containing the archived documents retrieved.

DSN... is valid in OS/VS2 MVS only. It indicates that the FROM data set is dynamically allocated. If the specified data set name is in quotation marks, it is used as given. Otherwise it is qualified as described in Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics." ARCHIVE is the suffix qualifier.

OPERATOR This operand specifies that a message is sent from the retrieve program to the console operator specified by the route code in the DEFINE SYSTEM command. The message requests the mounting of the volume serial number that was entered in the DER when the document was archived with the OPERATOR operand. If OPERATOR is specified, FROM must also be specified. It is valid only if the OPERATOR operand was specified on the ARCHIVE command(s).

Notes:

1. The output document library JCL statements are required.
2. Omitting the FROM operand is valid only in OS/VS2, and then only if the document was archived to a dynamically allocated data set.
3. The OPERATOR operand is valid only if the document was archived using the OPERATOR operand. FROM must be specified when using OPERATOR.
4. Unlike the ARCHIVE DOCUMENT command, the RETRIEVE DOCUMENT command does not allow a range of versions to be retrieved.

Example

This example retrieves the document named OLDEST from a data set defined by the DD (OS/VS2) or DLBL/TLBL (VSE) JCL statement named ARCHIVE. A message is written to the operator requesting the volume that was entered when the document was archived. OPERATOR must have been specified on the ARCHIVE command.

```
AUTH 100/WORKER
RETRIEVE DOCUMENT OLDEST FROM(ARCHIVE) OPERATOR.
```

RETRIEVE NAMELIST

This command retrieves archived documents specified in a namelist. See the section titled "Creating a Namelist for the RETRIEVE Command" on page 92 for details about the format of a namelist entry.

RETRIEVE NAMELIST	<pre> { * { { ddname { DSN('dsname') { DSN('dsname(member)') } } } FROM ({ { ddname { DSN(dsname) { DSN(dsname(member)) } } } { DSN('dsname') { DSN('dsname(member)') } }) </pre>
-------------------	---

() The first operand is positional and required. It shows the location of the namelist that specifies the documents to be retrieved. The namelist can be in the job input stream or in a data set or partitioned data set member. (See "Creating a Namelist for the RETRIEVE Command" on page 92 for the format of NAMELIST entries.) This first parameter must be one of the following:

* specifies that the namelist follows the RETRIEVE command in the job input stream. This is the only valid option in VSE.

ddname is valid only in OS/VS2 MVS and specifies the name on the JCL DD statement describing the data set containing the namelist.

DSN... is valid in OS/VS2 only. It indicates that the data set is dynamically allocated. The specified data set name must be in single quotation marks and is used as given.

FROM This required operand identifies the external sequential data set or partitioned data set member that contains the archived documents to be retrieved.

ddname is the name on the DD, DLBL, or TLBL JCL statement describing the FROM data set.

DSN... is valid in OS/VS2 only. It indicates that the FROM data set is dynamically allocated. If the specified data set name is in quotation marks, it is used as given. Otherwise it is qualified as described in Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics." ARCHIVE is the suffix qualifier.

Notes:

1. The output document library JCL statements are required.
2. The FROM operand must be specified.

3. With the RETRIEVE NAMELIST command, efficiency is increased if you retrieve the documents in the order in which they were archived. This order can be determined by using the LIST DOCUMENT command with the ARCHIVE operand, which lists the documents in alphabetical order with the date and time of the archive for each document. Alternatively, if you used a namelist to archive this set of documents and it is still available, edit the namelist to reflect only the documents you want retrieved.

CREATING A NAMELIST FOR THE RETRIEVE COMMAND

The RETRIEVE NAMELIST command requires a user-created namelist either following the command in the job input stream or in a sequential data set. This namelist contains one or more entries, each containing at least the name of the document to be retrieved. Other information might be required in a namelist entry to uniquely identify the document.

A namelist entry can specify any document owned by the current user as indicated by the most recent AUTH command. If the current user is an administrator, a namelist entry can specify any document owned by any user of the document library.

The structure of a namelist entry is:

<div style="margin-bottom: 10px;">docid</div> <div style="margin-bottom: 10px;">[DATA(name)]</div> <div>[VERSION(number)]</div>

- | | |
|------------------------|--|
| docid | This operand is positional within a namelist entry and must be specified first. For docid, the reader must be familiar with the information found in Chapter 2, "JCL and Command Syntax Conventions" on page 13. |
| DATA(name) | This operand specifies the data type of the document. If it is not specified, the default in the current user's UPR is used. |
| VERSION(number) | <p>This operand specifies the version of the document to be retrieved. It is considered an error if you:</p> <ol style="list-style-type: none"> 1. Specify this operand for a document without versions. 2. Use the VERSION operand without specifying (number). |

NAMELIST FORMAT

1. The namelist format always starts in position 2. A blank in position 1 indicates the start of a new logical record.
Record text must not begin in position 1.
2. You can continue namelist entries on separate 80-byte logical records if you follow these rules:
 - A plus sign (+) in position 1 means that the first character in this record immediately follows the last nonblank character of the previous record.

- A minus sign (-) in position 1 means that this record is appended to the end of the previous record with no adjustments for blanks.
- 3. A period in position 1 or an EOF during reading indicates the end of a namelist. The period must be in a separate record following the last record of the namelist entries. DLF ignores any other information in the period record.
- 4. No other nonblank characters will be allowed in position 1 of an input record. Command processing ends if a nonblank character encountered in position 1 is not a period, a plus sign, or a minus sign.

Examples

- This example retrieves DOCNAME1 which originally was stored in public library 456 and DOCNAME2 which originally was stored in user 100's library. They are retrieved from the data set identified by the DD, DLBL, or TLBL JCL statement named ARCD\$1. Both documents are without versions and have data types identical to the default in user 100's UPR. Neither document has an associated password. The name list immediately follows the RETRIEVE command in the input stream and is terminated by a record with a period in column 1.

```
AUTH 100/WORKER
RETRIEVE NAMEDLIST (*) FROM(ARCD$1)
  456 DOCNAME1
  DOCNAME2
```

- This example retrieves those documents identified in the namelist defined by the JCL DD statement named NAMEDS. Any number of documents can be identified for retrieval in this namelist. The documents are actually retrieved from an external sequential data set defined by the JCL DD statement named ARCD\$S. This example is valid only in OS/VS2 MVS.

```
AUTH 100/WORKER
RETRIEVE NAMEDLIST (NAMEDS) FROM(ARCD$S)
```

SCRIPT

This command formats documents with the SCRIPT/VS formatter in a batch environment if the Document Composition Facility (DCF) program product is installed. The document can come from either an input data set outside or from inside the document library. The output can be directed to either a printer or to an external data set.

A detailed description of the SCRIPT command is found in the Document Composition Facility: SCRIPT/VS Language Reference. Several options specifically applicable to a batch environment, however, are described here.

To understand the description of the SCRIPT command that follows, the reader must have read and understood the material in the section "Commonly Used Command Operands" in Chapter 2.

SCRIPT	<pre>{docid [DATA(name)] [VERSION(number)]}¹ { * }¹ { dname } FROM ({ DSN(dsname) }) { DSN(dsname(member)) } { DSN('dsname') } { DSN('dsname(member)') } [PROCESS(name)] [PARM('64 characters')] [options]</pre>
---------------	---

docid This operand identifies the document to be formatted. If a document library of 0 is specified, the user's project library is assumed to contain the document to be formatted.

DATA(name) This operand specifies the data type of the document that is to be formatted. If the data type is not specified, the default data type (that assigned to the current user) is used. If the data type is not specified and the current user has no default data type, then a null data type is assumed. Also, all imbed files used in the document being formatted must have a data type equal to the default data type of the current user unless .dd controls override this data type name.

VERSION(number) This operand identifies the version number of the document formatted. Without the VERSION operand, the latest version is formatted. This is the default.

An error message is printed and no formatting takes place if this operand is used for documents without versions.

¹ You must enter either ({docid [DATA(name)] [VERSION(number)]}) or the FROM operand (with appropriate options) - they are mutually exclusive.

FROM

This operand shows that the document to be formatted comes from an external source and not from the document library.

***** shows that an application program is calling DLF as a subroutine and will supply the data to be formatted. The special form FROM(*) is used only by an application program that calls DLF as a subroutine. See Chapter 9, "Using DLF as a Subroutine" for details.

ddname specifies the name on the DD, DLBL, or TLBL JCL statement identifying the sequential data set used for input when the document is not in the document library.

DSN... specifies the name of the data set used for input when the document library is not the source. This parameter is valid for dynamic allocation in OS/VS2 only. SCRIPT is the default suffix qualifier if the specified data set name is not in quotation marks. See Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics," for details.

PROCESS(name) This operand specifies the name of a load module that is loaded and called to process each line of text. This processor supersedes any processor implied by the DATA(name) keyword.

PARM('64 characters') This operand has an accompanying string up to 64 characters long enclosed in quotes. The string may include any values (the default is binary zeros). It is passed to the invoked processor in the initial call. You can invoke a processor in two ways:

1. Explicitly — using the PROCESS(name) operand.
2. Implicitly — through the presence of a processor profile record (created by a DEFINE PROCESS command) and the use of the appropriate data type in the DATA operand with an implied outdata data type of SCRIPT.

This operand is required if a called processor needs to receive parameters when invoked. If you do not specify the PARM operand, no parameter data is passed to any processor that might be called.

Note: See "DEFINE PROCESS" on page 165 and Chapter 8, "Using DLF to Call Processors" on page 215 for more information.

options

The following options have particular characteristics associated with a batch environment. These options and others are explained in detail in Document Composition Facility: SCRIPT/VS Language Reference.

FILE This operand specifies the sequential data set used for output from the formatter. If this option is omitted, and either PRINT is specified or implied or TERM is not specified or implied, output is written to DSMLIST (for OS/VS2) or SYSLST (for VSE).

ddname specifies the name on the DD, DLBL, or TLBL statement identifying the data set used for output.

DSN... specifies the name of the data set used for output. This parameter is valid for dynamic allocation only.

in OS/VS2. LIST is the default suffix qualifier if the specified data set name is not in quotation marks. For further details, see Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics." The four options for specifying DSN(...) are the same as those shown here for the FROM(DSN(...)) operand.

If you want FILE output from the formatter printed on a page printer you can specify the following options:

FONTLIB This operand specifies the name
(ddname) of the system library that contains the fonts for the page printer associated with the logical output device. In MVS, the FONTLIB is a partitioned data set. In VSE/AF 1.3.x, it is a core image library (CL) that you must specify on a JCL LIBDEF statement. In VSE/AF 2.1.0, it is a sublibrary that must be specified on a JCL LIBDEF statement. For MVS, the ddname entry is the file name on the DD JCL statement.

SEGLIB This operand, valid only for
(ddname) MVS, specifies the name of a library that contains page segments for documents that merge graphics with text for page printer output. If you are formatting a document under VSE that includes segments (.SI control words), then you must specify NOSEGLIB so that SCRIPT will not access the segment library.

PROFILE This operand specifies the document to be used as the profile.

Note: If no profile is specified, SCRIPT/VS defaults to the profile called "PROFILE" that is supplied by IBM with the DCF program product. This is a SCRIPT input file that is embedded by SCRIPT/VS before the document to be formatted is processed. This profile must have a data type equal to the default data type for the current user.

user-number identifies the library where the profile resides. It is not required if the profile resides in the current user's library as specified in the most recent AUTH command. A user-number of 0 refers to the project library of the current user. A specific project library user number is 1 to 7 numeric characters.

document-name identifies a profile that resides in the document library. document-name is 1 to 16 alphameric characters.

password is the profile password if one exists. Password is 1 to 8 alphameric characters.

1. The profile document's password (if any) must always be specified.

2. Library number must be specified if the profile resides in a public library other than 1 314 151.
3. Library number need not be specified if the profile resides in either the user's own library, his project library, or public library 1 314 151.
4. The profile cannot be accessed from a private library other than the authorized users even if explicitly specified in PROFILE or SEARCH options.
5. The library number (if specified in the PROFILE option) must be separated from the profile name by a blank.
6. The password (if specified in the PROFILE option) must be separated from the profile name by a slash which may or may not be preceded by a blank.

SEARCH

This operand specifies one additional user library to be searched for imbedded documents, symbol and macro definitions, or GML tags used in the source document. The operand can specify a public, project, or the current user's private library user-number and is 1 to 7 numeric digits.

If you do not specify the SEARCH option, no additional library is searched for imbedded documents, symbols, or macro definitions.

Notes:

1. The search order used by DLF to find imbedded files, symbol and macro definitions, and GML tags is:
 - a. The current user's private library
 - b. The current user's project library
 - c. The library specified in the SEARCH option
 - d. The SCRIPT/VS required public library (1 314 151).
2. DLF does not scan, interpret, or analyze the PARM string. It is meaningful only to the processor to which it is passed.
3. SCRIPT/VS assumes the existence of a required public library that contains the GML tags, profiles, macros, and symbols required for its processing. The library number shipped as part of DCF is 1 314 151. It is your option to change this number. Instructions for doing so are included in the customer documentation shipped with DCF. DLF always uses this public library as the last library in the search sequence.
4. The minimum JCL statements required are the input document library and the DSMUTMSG DD statement (for OS/VS2) or the DSMUMSG DLBL or TLBL statement (for VSE). Additional JCL statements may be required depending on the requirements of the document to be processed by SCRIPT/VS. See Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics," for details.

In the MVS environment, if output is to be printed on a page device, a FONTLIB JCL statement is required. If there are segment include (.SI) control words in the document, you

must provide a JCL statement specifying the SEGLIB operand, unless you specified NOSEGLIB.

In the VSE environment, if you provide a DLBL JCL statement for FONTLIB, you must also specify the DLBL name as a keyword for CL (core image library).

5. The source document being processed can have imbedded documents. These documents can be in a private, project, or public library, a library specified in the SEARCH option, or in an external sequential data set if the SCRIPT/VS control word of .dd is used.
6. A document can undergo preprocessing by a processor in two cases:
 - a. If the PROCESS keyword is specified, this processing takes place, regardless of the location of the input document or of the document's attributes. The document is processed by the load module whose entry name is specified in the PROCESS operand. This operand must be specified if a document read from a sequential data set requires preprocessing.
 - b. Otherwise, an implied processor is invoked. This processor must have been defined in a DEFINE PROCESS command with an INDATA definition equal to the DATA definition of the document and an OUTDATA definition equal to SCRIPT. If no such processor is found, the document is formatted without any processing.
7. Output is in 1403 printer format unless that format is overridden by the DEVICE option. See the Document Composition Facility: SCRIPT/VS Language Reference for the description of the DEVICE option.
8. The LIB, QUIET, NOSPIE, STOP, and TERM options of SCRIPT/VS are not valid in a batch environment. If any of these options is specified, a warning message is issued, the option is ignored, and processing continues.
9. Imbedded and appended files coming from the library must have been imported with the same data type as the primary file being formatted.
10. When you use utility files (DSMUTMSG, DSMUTTOC, DSMUTWTF, and DSMUTCTF), they must be allocated to a sequential dataset. See Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics" on page 237 for utility file attributes.

Examples

- This example directs DLF to format a document named ABBEY. The document is in user 181's private library. Before each record is given to the formatter, the text processing controls are converted from ATMS-III to SCRIPT/VS by the load module DSMACAIP. To format the document correctly, specify the profile document ATMSPRF2.

```
AUTH 181/MONK
SCRIPT ABBEY PROCESS(DSMACAIP) ( PROFILE(ATMSPRF2))
```

- This example formats a document from the data set U111111.EDEN.SCRIPT, which is dynamically allocated, and writes it to the sequential data set identified by the DSMLIST DD statement.

```
AUTH 111111/EVE
SCRIPT FROM(DSN(EDEN))
```

- This example formats a document that is in a data set with a ddname of PARAKEET. The load module is named BIRD.

```
AUTH 111111/EVE
SCRIPT FROM(PARAKEET) PROCESS(BIRD)
```

- This example formats a document named MEMO residing in the project library of user 5 001. Library 4 000 has been added to DLF's search sequence so any symbols, macros, GML tags, or imbedded documents in this library will also apply to MEMO. DLF will search library 4 000 after the user's private library and project library have been searched.

The profile DSMPROF3 controls the formatting. The indicated options are: formatting the document to an IBM 3800 printer, producing a memo on 8-1/2 x 11 inch paper, printing 8 lines per inch, using the GT15 font, and providing a 2-inch left margin for binding.

```
AUTH 5001/SECPool
SCRIPT 0 MEMO (SEARCH(4000) PROFILE(DSMPROF3) -
DEVICE(3800N8) CHARS(GT15) BIND(2I))
```

(See "Formatting a Document with SCRIPT/VS" on page 110 for more information.)

CHAPTER 4. USING THE DOCUMENT LIBRARY

With DLF you, as a general user, can perform the following tasks:

- Put documents into your own library
- Import data included within the job input stream
- Produce copies of a document
- Back up your documents and copy them into another document library
- Purge documents
- Get lists of document names
- Format a document with SCRIPT/VS
- Process ATMS documents
- Archive documents to and retrieve documents from external data sets
- Copy versions of documents
- Change document names and attributes.

This chapter describes how to do these tasks. In each case a problem and its solution is shown.

For the purpose of these discussions, assume you are general user 88 with password CITADEL. You own private library 88 and have access to project library 1 000. There is also a SCRIPT/VS system library 1 314 151, which is defined as a public library.

The following examples are not meant to be a comprehensive discussion of all library user commands, nor of all job control language (JCL, for OS/VS2) or job control statements (JCS, for VSE). For a complete discussion of library user commands see Chapter 3, "General User Commands." For detailed discussions of JCL and JCS requirements see Chapter 2, "JCL and Command Syntax Conventions" and Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics."

PUTTING DOCUMENTS INTO YOUR OWN LIBRARY

The Problem: You are general user 88 with three new documents located on different devices. You want to put these documents into the document library:

- DOC1 is a SCRIPT document residing on tape. You want to place it in your private library (88) and share it with other users.
- DOC2 is a document residing in a sequential data set on disk. It is to be named REPORT7 and put into your project library (1 000).
- DOC3 is an ATMS-III document residing in a sequential data set on disk. It will be converted to a SCRIPT document before it is put into DCF's system library, public library 1 314 151, and named ATMSREP with the password MOOSE.

The Solution: Use the IMPORT command. The JCL and IMPORT commands required for this task are shown in Figure 6 on page 102.

```

//IMPORT JOB
//STEP1 EXEC PGM=DSMSPEXC,PARM=LIST
//SYSPRINT DD SYSOUT=A
//DSMLIST DD SYSOUT=A
//DSMPDIR DD DSN=DSMLIB.DIRECTRY,DISP=OLD
//DSMPLIB DD DSN=DSMLIB.SOURCE,DISP=OLD
//DOC1DD DD UNIT=TAPE,LABEL=(1,SL),
//          VOL=SER=TAPE1,DISP=OLD,DSN=DOC1
//DOC2DD DD DSN=DSDOC2,VOL=SER=DISKA,
//          DISP=OLD,UNIT=3330
//DOC3DD DD DSN=DSDOC3,VOL=SER=DISKA,
//          DISP=OLD,UNIT=3330
//SYSIN DD *
AUTH 88/CITADEL
IMPORT DOC1 FROM(DOC1DD) SHARE DATA(SCRIPT)
IMPORT 1000 REPORT7 FROM(DOC2DD)
IMPORT 1314151 ATMSREP/MOOSE FROM(DOC3DD) -
DATA(SCRIPT) PROCESS(DSMACAIP)
/*

```

for OS/VS2

```

// JOB IMPORT
// DLBL DSMPDIR,'DSMLIB.DIRECTRY',,VSAM
// EXTENT ,USRDSK
// DLBL DSMPLIB,'DSMLIB.SOURCE',,VSAM
// EXTENT ,USRDSK
// TLBL DOC1DD,'DOC1',,TAPE1
// ASSGN SYS021,x'380'
// DLBL DOC2DD,'DSDOC2',0,SD
// EXTENT SYS022,DISKA
// ASSGN SYS022,3330,VOL=DISKA,SHR
// DLBL DOC3DD,'DSDOC3',0,SD
// EXTENT SYS023,DISKA
// ASSGN SYS023,3330,VOL=DISKA,SHR
// OPTION LISTX
// EXEC DSMSPEXC,SIZE=AUTO
AUTH 88/CITADEL
ENVIRONMENT TLBL(DOC1DD) DEV(2400 021)
IMPORT DOC1 FROM(DOC1DD) SHARE DATA(SCRIPT)
ENVIRONMENT DLBL(DOC2DD) DEV(3330 022)
IMPORT 1000 REPORT7 FROM(DOC2DD)
ENVIRONMENT DLBL(DOC3DD) DEV(3330 023)
IMPORT 1314151 ATMSREP/MOOSE FROM(DOC3DD) -
DATA(SCRIPT) PROCESS(DSMACAIP)
/*
/&

```

for VSE

Figure 6. Examples of the IMPORT Command

Notes:

1. If PARM=LIST is specified on the EXEC statement for OS/VS2 or if OPTION LISTX is designated for VSE, the contents of each document are printed out during the import processing.
2. In VSE you must supply an ENVIRONMENT command for each external data set you name on an IMPORT command. ENVIRONMENT commands describing a tape use the default block size (800) and record size (80). In addition, they assume that tapes are labeled.
3. Each IMPORT command has the keyword FROM, which in this example specifies a unique DD, TLBL, or DLBL statement. The DD, TLBL, or DLBL statement describes a sequential data set or data set member.

4. DOC1 was read from the first file of the tape labeled TAPE1. It was given the name DOC1 and was put into your private library by default because no other user number was specified. The document was stored in the document library with a data type of SCRIPT. Because DOC1 was assigned share status, all other users in the document library are able to use the EXPORT, COPY, and SCRIPT commands with this document.
5. DOC2 was read from the sequential data set DSDOC2 on volume DISKA, was named REPORT7, and was put into your project library (1 000). You still remain the document owner, however. Because the data type was not specified and no default data type is associated with user 88, the document was assigned a null data type by default. REPORT7 can be shared by other project members.
6. DOC3 was read from sequential data set DSDOC3. It was named ATMSREP and assigned password MOOSE. DLF called the processor identified by the load module named DSMACAIP. This processor converted the ATMS-III formatting controls to SCRIPT/VS symbols and macros. The document was then placed in the DCF system library (public library 1 314 151) with the data type of SCRIPT.

The Problem: You would like to compose a memo, put it in your private library, and look at its DER all in the same job step.

The Solution: Use the IMPORT command with the FROM(*) operand. This operand indicates that the imported file follows the command; that is, the file is to be imported from the input stream. The JCL and commands required to complete the job are shown in Figure 7 on page 104.

```
//IMPORT JOB
//STEP1 EXEC PGM=DSMSPEXC,PARM='LIST'
//SYSPRINT DD SYSOUT=A
//DSMLIST DD SYSOUT=A
//DSMINDIR DD DSN=DSMLIB.DIRECTRY,DISP=SHR
//DSMINLIB DD DSN=DSMLIB.SOURCE,DISP=SHR
//DSMPDIR DD DSN=DSMLIB.DIRECTRY,DISP=OLD
//DSMPDLIB DD DSN=DSMLIB.SOURCE,DISP=OLD
//SYSIN DD *
AUTH 88/CITADEL
IMPORT SPECIAL FROM (*)
TO: STAFF
FROM: COMMUNICATIONS
THIS IS AN EXAMPLE OF IMPORTING DATA AS PA
+RT OF THE COMMAND FORMAT
LIST DOCUMENT SELECT(SPECIAL)
/*
```

for OS/VS2

```
// JOB IMPORT
// DLBL DSMIDIR,'DSMLIB.DIRECTRY',,VSAM
// EXTENT ,USRDSK
// DLBL DSMILIB,'DSMLIB.SOURCE',,VSAM
// EXTENT ,USRDSK
// DLBL DSMPDIR,'DSMLIB.DIRECTRY',,VSAM
// EXTENT ,USRDSK
// DLBL DSMPLIB,'DSMLIB.SOURCE',,VSAM
// EXTENT ,USRDSK
// OPTION LISTX
// EXEC DSMSPEXC,SIZE=AUTO
AUTH 88/CITADEL
IMPORT SPECIAL FROM(*)
TO: STAFF
FROM: COMMUNICATIONS
THIS IS AN EXAMPLE OF IMPORTING DATA AS PA
+RT OF THE COMMAND FORMAT
LIST DOCUMENT SELECT(SPECIAL)
/*
```

for VSE

Figure 7. IMPORT Command Using the FROM(*) Keyword

Notes:

1. Record text always starts in position 2. DLF allows IMPORT commands and data to be intermixed in the input reader. A blank in position 1 indicates the start of a new logical record. Record text must not begin in position 1.
2. Because data in the operating system readers is limited to 80-character logical records and input to the document library may consist of larger records, it may be necessary to use a continuation character. A plus or minus sign in position 1 will indicate that this record is a continuation of the previous record. If you use a minus sign, the preceding record is taken in its entirety and the data of the present record is concatenated to it (trailing blanks in the previous record are retained). If you use a plus sign, the trailing blanks of the previous record will be overlaid by the data of the present record. A blank in position 1 indicates a new logical record.
3. Unless you specify the NOSUPPRESS option on the IMPORT command, trailing blanks ('40'X) are always suppressed from

the end of each logical record that is placed in the library.

4. The end of a document being imported is indicated by a period in position 1 or an EOF during reading. This period must be in a record following the last record of the input document. Any other information in the period record is ignored.
5. If a nonblank character is encountered in position 1 that is not a period, a plus sign, or a minus sign, the document will not be imported. No other nonblank characters will be allowed in position 1 of an input record.
6. Because of the buffer size of DSMSPRDC, data record size can not exceed 1 020 bytes per input record from the input stream.

PRODUCING COPIES OF A SINGLE DOCUMENT

The Problem:

- You would like a copy of document DOC1 printed from your private library.
- In addition, you want to copy document ATMSREP, which is in public library 1 314 151, to tape.

The Solution: Use the EXPORT command. The JCL and EXPORT commands required to accomplish this task are shown in Figure 8.

```
//EXPORT JOB
//STEP1 EXEC PGM=DSMSPEXC,PARM='NOLIST'
//SYSPRINT DD SYSOUT=A
//DSMLIST DD SYSOUT=A
//DSMINDIR DD DSN=DSMLIB.DIRECTRY,DISP=SHR
//DSMINLIB DD DSN=DSMLIB.SOURCE,DISP=SHR
//ATMSDD DD UNIT=TAPE,DISP=(NEW,KEEP),LABEL=(1,SL),
// VOL=SER=ATMSTP,DSN=ATMSREP
//SYSIN DD *
AUTH 88/CITADEL
EXPORT DOC1 TO(DSMLIST)
EXPORT 1314151 ATMSREP/MOOSE TO(ATMSDD) DATA(SCRIPT)
/*
```

for OS/V52

```
// JOB EXPORT
// DLBL DSMIDIR,'DSMLIB.DIRECTRY',,VSAM
// EXTENT ,USRDSK
// DLBL DSMILIB,'DSMLIB.SOURCE',,VSAM
// EXTENT ,USRDSK
// TLBL ATMSDD,'ATMSREP',,ATMSTP
// ASSGN SYS027,X'381'
// OPTION NOLISTX
// EXEC DSMSPEXC,SIZE=AUTO
AUTH 88/CITADEL
EXPORT DOC1 TO(SYSIST)
ENV TLBL(ATMSDD) DEV(2400 027)
EXPORT 1314151 ATMSREP/MOOSE TO (ATMSDD) DATA(SCRIPT)
/*
/&
```

for VSE

Figure 8. Examples of the EXPORT Command

Notes:

1. The EXPORT command requires that the TO keyword be specified unless the output ddname is the same as the first eight characters of the name of the document being exported. These eight characters are changed to uppercase and any special characters are changed to @ before being used as the ddname.
2. In the VSE example, an ENV command is required to describe an output data set on tape; it is not required when exporting to SYSLST. A JCL statement for SYSLST is not required if that SYSLST has been assigned as a system output data set.
3. A copy of DOC1 was read from your private library (88), the default, because no other library was specified. The copy was written to the sequential data set identified by the JCL statement labeled DSMLIST (OS/VS2) or SYSLST (VSE).
4. A copy of ATMSREP was read from public library 1 314 151 and written to a sequential file identified by the JCL statement labeled ATMSDD, which describes a tape. Because this document was password-protected, the correct password, MOOSE, had to be supplied to access the document.

BACKING UP DOCUMENTS AND COPYING ANOTHER DLF LIBRARY

The Problem:

- Having added many new documents to the document library, you would like to back up all the documents you own. That is, you would like to copy them all to an external data set to be retained as backup. This procedure gives you the latest copy of each document for recovery purposes. It also gives you a complete set of your documents to move to another document library. For this task you want to copy all your documents to a tape.
- In addition, you want to copy one of your documents (DOC1) without versions to an entirely separate document library associated with another department. User 88 is also an authorized user of that document library. You want the document to be stored in the other document library as DOC2, replacing any existing, non-versioned document with the same name.

The Solution: Use the COPY OUT and COPY DOCUMENT commands. The JCL and copy commands required to accomplish this task are shown in Figure 9.

```
//COPY      JOB                                           for OS/VS2
//STEP1 EXEC PGM=DSMSPEXC,PARM='NOLIST'
//SYSPRINT DD SYSOUT=A
//DSMLIST   DD SYSOUT=A
//DSMPDIR   DD DSN=DSMLIB1.DIRECTRY,DISP=OLD
//DSMPDLIB  DD DSN=DSMLIB1.SOURCE,DISP=OLD
//DSMIDIR   DD DSN=DSMLIB.DIRECTRY,DISP=SHR
//DSMLIB    DD DSN=DSMLIB.SOURCE,DISP=SHR
//BACKTP    DD DSN=BACKTAPE,DISP=(NEW,KEEP),
//           LABEL=(1,NL),UNIT=TAPE
//SYSIN     DD *
//           AUTH 88/CITADEL
//           COPY OUT TO(BACKTP)
//           COPY DOCUMENT DOC1 REPLACE NEWNAME(DOC2)
/*
```

```
// JOB      COPY                                           for VSE
// DLBL     DSMPDIR,'DSMLIB1.DIRECTRY',,VSAM
// EXTENT   ,USRDSK
// DLBL     DSMPDLIB,'DSMLIB1.SOURCE',,VSAM
// EXTENT   ,USRDSK
// DLBL     DSMIDIR,'DSMLIB.DIRECTRY',,VSAM
// EXTENT   ,USRDSK
// DLBL     DSMILIB,'DSMLIB.SOURCE',,VSAM
// EXTENT   ,USRDSK
// TLBL     BACKTP
// ASSGN    SYS034,X'381'
// OPTION    NOLISTX
// EXEC     DSMSPEXC,SIZE=AUTO
//           AUTH 88/CITADEL
//           ENV TLBL(BACKTP) DEV(2400 034) NOLABEL
//           COPY OUT TO(BACKTP)
//           COPY DOCUMENT DOC1 REPLACE NEWNAME(DOC2)
/*
/&
```

Figure 9. Examples of the COPY OUT and COPY DOCUMENT Commands

Notes:

1. Both COPY functions require the input document library JCL statements. The COPY DOCUMENT command also requires the output document library JCL statements.
2. All documents owned by you were written to the tape (identified by the JCL statement labeled BACKTP) as a sequential data set. This procedure included all documents in your private library and all documents you own in your project library 1 000 and in all public libraries.
3. DLF copied DOC1 to user 88 in the output document library. Because TOUSER was not specified, it defaulted to the authorized user. The document is given the new name of DOC2 in the output document library. If DOC2 already exists in the output document library, it is replaced.

While you, as user 88, did not have to be defined with the same privileges in both document libraries, you have to be defined in both and have the same password in both.

PURGING DOCUMENTS AND GETTING LISTS OF DOCUMENT NAMES

The Problem: You want to delete some documents you no longer need, and then you want a new updated list of your documents. You also want a list of all documents in public library 1 314 151 that fall between documents DSMAAAAA and DSMZZZZZ inclusive. You want only the ones that were put in the library on or before December 17, 1981.

The Solution: Use the PURGE DOCUMENT and LIST DOCUMENT commands. The JCL statements and DLF commands required to accomplish the tasks are shown in Figure 10 on page 109.

```

//LIST      JOB
//STEP1 EXEC PGM=DSMSPEXC
//SYSPRINT DD SYSOUT=A
//DSMLIST DD SYSOUT=A
//DSMINDIR DD DSN=DSMLIB.DIRECTRY,DISP=SHR
//DSMINLIB DD DSN=DSMLIB.SOURCE,DISP=SHR
//DSMPDIR DD DSN=DSMLIB.DIRECTRY,DISP=OLD
//DSMPTLIB DD DSN=DSMLIB.SOURCE,DISP=OLD
//SYSIN DD *
AUTH 88/CITADEL
PURGE DOCUMENT DOC1
PURGE DOCUMENT 1314151 ATMSREP/MOOSE DATA(SCRIPT)
PURGE DOCUMENT 1000 REPORT7
LIST DOCUMENT
LIST DOCUMENT USER(1314151)-
SELECT (DSMAAAA:DSMZXXX)-
DATED(:12/17/81) ALL
/*

// JOB      LIST
// DLBL     DSMIDIR, 'DSMLIB.DIRECTRY',,VSAM
// EXTENT   ,USRDSK
// DLBL     DSMILIB, 'DSMLIB.SOURCE',,VSAM
// EXTENT   ,USRDSK
// DLBL     DSMPDIR, 'DSMLIB.DIRECTRY',,VSAM
// EXTENT   ,USRDSK
// DLBL     DSMPLIB, 'DSMLIB.SOURCE',,VSAM
// EXTENT   ,USRDSK
// EXEC     DSMSPEXC,SIZE=AUTO
AUTH 88/CITADEL
PURGE DOCUMENT DOC1
PURGE DOCUMENT 1314151 ATMSREP/MOOSE DATA(SCRIPT)
PURGE DOCUMENT 1000 REPORT7
LIST DOCUMENT
LIST DOCUMENT USER(1314151)-
SELECT (DSMAAAA:DSMZXXX)-
DATED(:12/17/81) ALL
/*
/&

```

for OS/VS2

for VSE

Figure 10. Examples of the PURGE DOCUMENT and LIST DOCUMENT Commands

Notes:

1. Because PURGE DOCUMENT requires the output document library JCL statements and LIST DOCUMENT, by default, requires the input JCL statements, both sets of JCL were included.
2. Three of your documents were deleted from three different libraries, one document at a time. ATMSREP has a password, and you needed to specify it to delete the document because you are not an administrator. DATA type need not be specified unless it differs from the default DATA type for the current library user.
3. Because the TO keyword was not specified on the LIST DOCUMENT command, your output was written to the output data set identified in DD statement DSMLIST (for OS/VS2) or the assigned SYSLST (for VSE).
4. By omitting all optional keywords in the LIST DOCUMENT command, you got a list of all the documents you own in your private, project, and public libraries (your ownership set).

5. Public library 1 314 151 was searched by DLF and, regardless of ownership, all document names were listed that fell in the range of document names DSMAAAAA to DSMZZZZZ and that were put into the public library on or before December 17, 1981.

FORMATTING A DOCUMENT WITH SCRIPT/VS

The Problem: You want to format a document named DSMIVC30, which contains GML markup and resides in public library 1 314 151. It imbeds several documents that reside in your private library, in public library 1 314 151, and in one additional public library, 9 999. You don't know which control words are used in the document. Therefore, you don't know which of the SCRIPT/VS utility data sets will be required. See Figure 15 on page 115 for a table describing the SCRIPT/VS Utility data sets. You will need DD statements for all the likely Utility data sets. You will include all except those relating to terminal I/O, just in case. For VSE, ENVIRONMENT commands may also be required for those files.

The document will have a table of contents. You'd like to check the spelling, so you expect quite a few messages.

And, you'd like to have this document printed on a device such as:

- The IBM 3800 Printing Subsystem Model 1
- The IBM 4250 Printer
- The IBM 3800 Printing Subsystem Model 3 operating as an all-points-addressable printer
- The IBM 3820 Page Printer.

All system messages will be printed on the system printer.

This document must be formatted with the SCRIPT/VS profile document DSMPROF3, which resides in public library 1 314 151. It requires two fonts, and you've decided to use the fonts identified by the names GT12 and GB12.

The Solution: These tasks require the SCRIPT command, so DCF must be installed. The JCL and DLF commands required to accomplish these tasks are shown in Figure 11 on page 111, Figure 12 on page 112, and Figure 13 on page 113.

```

//SCRIPT JOB
//STEP1 EXEC PGM=DSMSPEXC,PARM='LIST'
//SYSPRINT DD SYSOUT=A
//D3800 DD SYSOUT=(F,,3800),CHARS=(GT12,GB12),
// DCB=(OPTCD=J,RECFM=VBM,BLKSIZE=260,LRECL=256)
//DSMLIST DD SYSOUT=A
//DSMINDIR DD DSN=DSMLIB.DIRECTRY,DISP=SHR
//DSMINLIB DD DSN=DSMLIB.SOURCE,DISP=SHR
//DSMUTMSG DD UNIT=SYSDA,SPACE=(TRK,(5,5))
//DSMUTOC DD UNIT=SYSDA,SPACE=(TRK,(5,5))
//DSMUTWTF DD UNIT=SYSDA,SPACE=(TRK,(5,5))
//SYSIN DD *
AUTH 88/CITADEL
SCRIPT 1314151 DSMIVC30 (PROF(DSMPROF3) CONT-
FILE(D3800) BIND(11) TWOPASS SPELLCHK SEARCH(9999)-
DEV(3800N8) CHARS(GT12 GB12))
/*

```

for OS/VS2

```

// JOB SCRIPT
// ASSGN SYS020,3800
// SETPRT SYS020,CHARS=(GT12,GB12),TRC=Y
// DLBL DSMIDIR,'DSMLIB.DIRECTRY',,VSAM
// EXTENT ,USRDSK
// DLBL DSMILIB,'DSMLIB.SOURCE',,VSAM
// EXTENT ,USRDSK
// DLBL DSMUMSG,'MESSAGE FILE',0,SD
// EXTENT SYS023,USRDSK,1,0,4978,10
// ASSGN SYS023,3330,VOL=USRDSK,SHR
// DLBL DSMUTOC,'TABLE OF CONTENTS',0,SD
// EXTENT SYS025,USRDSK,1,0,4993,5
// ASSGN SYS025,3330,VOL=USRDSK,SHR
// DLBL DSMUWTF,'UTILITY FILE',0,SD
// EXTENT SYS026,USRDSK,1,0,4998,5
// ASSGN SYS026,3330,VOL=USRDSK,SHR
// OPTION LISTX
// EXEC DSMSPEXC,SIZE=AUTO
AUTH 88/CITADEL
ENVIRONMENT DLBL(DSMUMSG) DEV(3330 023)
ENVIRONMENT DLBL(DSMUTOC) DEV(3330 025)
ENVIRONMENT DLBL(DSMUWTF) DEV(3330 026)
ENVIRONMENT FILENAME(D3800) DEV(3800 020)
SCRIPT 1314151 DSMIVC30 (PROF(DSMPROF3) CONT -
FILE(D3800) BIND(11) TWOPASS SPELLCHK SEARCH(9999) -
DEV(3800N8) CHARS(GT12 GB12))
/*
/&

```

for VSE

Figure 11. Example of the JCL statements and SCRIPT Command for 3800 Model 1. The FILE option of the SCRIPT command is used to indicate that formatted output is directed to the IBM 3800 Printing Subsystem instead of to the system printer indicated by DSMLIST (OS/VS2) or SYSLST (VSE). For OS/VS2, the 3800 printer is described by the DD statement D3800. For VSE, the FILENAME keyword of the ENVIRONMENT command associates the name D3800 with the system assignment. No DLBL or TLBL statement is used. Notice, however, that a SETPRT statement is required in VSE if you want to change the character sets or other default values. In addition, the CHARS parameter in the SCRIPT command must refer to the same fonts in the same order as specified on the D3800 DD statement (OS/VS2) or SETPRT statement (VSE).

DEV(3800N8) requested that the document be formatted for an IBM 3800 Model 1 printer, 8 lines per inch on 8 1/2 x 11 inch paper.

```

//SCRIPT JOB
//STEP1 EXEC PGM=DSMSPEXC,PARM='LIST'
//SYSPRINT DD SYSOUT=A
//D4250 DD DSN=DSMIVC30.LIST4250,DISP=(NEW,PASS),
// UNIT=SYSDA,SPACE=(TRK,(5,10),DCB=(RECFM=VB,
// LRECL=2052,BLKSIZE=2056)
//FONT4250 DD DSN=SYS1.FONT4250,DISP=SHR
//DSMLIST DD SYSOUT=A
//DSMINDIR DD DSN=DSMLIB.DIRECTRY,DISP=SHR
//DSMINLIB DD DSN=DSMLIB.SOURCE,DISP=SHR
//DSMUTMSG DD UNIT=SYSDA,SPACE=(TRK,(5,5))
//DSMUTTOC DD UNIT=SYSDA,SPACE=(TRK,(5,5))
//DSMUTWTF DD UNIT=SYSDA,SPACE=(TRK,(5,5))
//SYSIN DD *
AUTH 88/CITADEL
SCRIPT 1314151 DSMIVC30 (PROF(DSMPROF3) CONT-
FILE(D4250) BIND(11) TWOPASS FONTLIB(FONT4250)-
SEARCH(9999) DEV (4250A)
/*

```

for OS/VS2

```

// JOB SCRIPT
// DLBL D4250,'LIST4250 OUTPUT'
// EXTENT SYS007,USRDSK,1,1,4968,10
// ASSGN SYS007,3330,VOL=USRDSK,SHR
// DLBL FONTS,'FONT LIBRARY'
// EXTENT ,BFUDSK,1,0,7619,5700
// LIBDEF CL,SEARCH=(FONTS)
// DLBL DSMIDIR,'DSMLIB.DIRECTRY',,VSAM
// EXTENT ,USRDSK
// DLBL DSMILIB,'DSMLIB.SOURCE',,VSAM
// EXTENT ,USRDSK
// DLBL DSMUMSG,'MESSAGE FILE',0,SD
// EXTENT SYS023,USRDSK,1,0,4978,10
// ASSGN SYS023,3330,VOL=USRDSK,SHR
// DLBL DSMUTOC,'TABLE OF CONTENTS',0,SD
// EXTENT SYS025,USRDSK,1,0,4993,5
// ASSGN SYS025,3330,VOL=USRDSK,SHR
// DLBL DSMUWTF,'UTILITY FILE',0,SD
// EXTENT SYS026,USRDSK,1,0,4998,5
// ASSGN SYS026,3330,VOL=USRDSK,SHR
// OPTION LISTX
// EXEC DSMSPEXC,SIZE=AUTO
AUTH 88/CITADEL
ENVIRONMENT DLBL(DSMUMSG) DEV(3330 023)
ENVIRONMENT DLBL(DSMUTOC) DEV(3330 025)
ENVIRONMENT DLBL(DSMUWTF) DEV(3330 026)
ENVIRONMENT DLBL(D4250) DEV(3330 007) FORMAT(VB)-
BLOCKSIZE(2056) RECORDSIZE(2048)
SCRIPT 1314151 DSMIVC30 (PROF(DSMPROF3) CONT-
FILE(D4250) BIND(11) TWOPASS-
SEARCH(9999) DEV(4250A)
/*
/&

```

for VSE

Figure 12. Example of the JCL statements and SCRIPT Command for IBM 4250 printer. The JCL statements for the font library and file name D4250 are necessary to format the document. The FILE option of the SCRIPT command writes page formatted output to the file identified by the D4250 JCL statement.

Another step, or job, is also necessary to get the 4250 print. To get 4250 output, use the Composed Document Print Facility (CDPF - Program Number 5668-997) as described in Composed Document Printing Facility Installation and Operation. The values shown for logical record length and block size are necessary for input to CDPF. DEV(4250A) requests that the document be formatted for an IBM 4250 printer, with a logical device specification indicating a page size of 8 1/2 by 11 inches. The MVS FONTLIB keyword identifies the library (name provided by your installation) that contains the fonts to be used when formatting the document for a page printer.


```

//SCRIPT JOB
//STEP1 EXEC PGM=DSMSPEXC,PARM='LIST'
//SYSPRINT DD SYSOUT=A
//D38PP DD DSN=DSMIVC30.LIST38PP,DISP=(NEW,PASS),
// UNIT=SYSDA,SPACE=(TRK,(5,10),DCB=(RECFM=VBM,
// LRECL=8205,BLKSIZE=8209)
//FONT38PP DD DSN=SYS1.FONT38PP,DISP=SHR
//DSMLIST DD SYSOUT=A
//DSMINDIR DD DSN=DSMLIB.DIRECTRY,DISP=SHR
//DSMINLIB DD DSN=DSMLIB.SOURCE,DISP=SHR
//DSMUTMSG DD UNIT=SYSDA,SPACE=(TRK,(5,5))
//DSMUTTOC DD UNIT=SYSDA,SPACE=(TRK,(5,5))
//DSMUTWTF DD UNIT=SYSDA,SPACE=(TRK,(5,5))
//SYSIN DD *
AUTH 88/CITADEL
SCRIPT 1314151 DSMIVC30 (PROF(DSMPROF3) CONT-
FILE(D38PP) BIND(11) TWOPASS FONTLIB(FONT38PP)-
SEARCH(9999) DEV (38PPN)
/*

```

for OS/VS2

Figure 13. Example of the JCL statements and SCRIPT Command for IBM 3800 Model 3. In the SCRIPT command, the JCL statements for the font library and file name D38PP are necessary to format the document. The FILE option of the SCRIPT command writes page formatted output to the file identified by the D38PP JCL statement.

A subsequent step, or job, is necessary to get the 3800 Model 3 print (when the 3800 Model 3 is operating as an all-points-addressable printer). The procedure for obtaining output on the IBM 3800 Printing Subsystem Model 3 using the Print Services Facility (PSF, Program Number 5665-265) is described in the Document Composition Facility: SCRIPT/VS Text Programmer's Guide. The values shown for logical record length and block size are necessary for input to PSF.

DEV(38PPN) requests that the document be formatted for an IBM 3800 Printing Subsystem Model 3 whose logical device specification indicates a page size of 8 1/2 by 11 inches.

The FONTLIB keyword identifies the library (name provided by your installation) that contains the fonts to be used when formatting the document for a page device.

```

* $$ JNM=SCRIPT
// JOB      SCRIPT
// DLBL     IJSYSUC.'VSEP.USER.CATALOG',,VSAM
// LIBDEF   PHASE,SEARCH=(DCFDLF.LIBRARY,RESRCE.FONT)
// DLBL     DSMUTOC,'TABLE OF CONTENTS',0,SD
// EXTENT   SYS009,DOSRES,,,15340,10
// ASSGN    SYS009,3350,VOL=DOSRES,SHR
// DLBL     DSMUWTF,'UTILITY.FILE',0,SD
// EXTENT   SYS012,DOSRES,,,15350,10
// ASSGN    SYS012,3350,VOL=DOSRES,SHR
// DLBL     DSMUMSG,'MESSAGE.FILE',0,SD
// EXTENT   SYS013,DOSRES,,,15360,10
// ASSGN    SYS013,3350,VOL=DOSRES,SHR
// ASSGN    SYS080,000
* $$ LST    LST=00F,DEST=(,PR38201),FNO=STD,CURRS=L
// DLBL     DSMIDIR,'DIRECTRY',,VSAM
// DLBL     DSMILIB,'SOURCE',,VSAM
// DLBL     DSMPIR,'DIRECTRY',,VSAM
// DLBL     DSMPLIB,'SOURCE',,VSAM
// OPTION   LISTX
// EXEC     DSMSPXEC,SIZE=AUTO
          AUTH 88/CITADEL
          ENVIRONMENT DLBL(DSMUMSG) DEV(3350)
          ENVIRONMENT DLBL(DSMUTOC) DEV(3350)
          ENVIRONMENT DLBL(DSMUWTF) DEV(3350)
          SCRIPT 1314151 DSMIVC30 (PRINT PROF(DSMPROF3) DEV(3820A) CONT-
            TWOPASS SEARCH(9999)
/*
/&
* $$ E0J

```

for VSE/POWER

ADDRESS OF [PRCL]

Figure 14. Example of the JCL and JECL statements and SCRIPT Command for IBM 3820 Page Printer. This example can be used only in VSE for direct output to VSE/POWER. In this example, you must assign SYS080 to a dummy device previously defined to VSE as a 3800 (see your system programmer for the specific device address). The VSE/POWER JECL LST statement must reference the same dummy device address.

For MVS (and also as an alternative in VSE), use an output to file method similar to that for the IBM 4250 printer. The differences would be to specify a 3820 font library, a logical device type of 3820A, and a logical record length of at least 8 201.

Notes:

1. Because the FROM keyword was not specified on the SCRIPT command, the document was not to be read from an external data set. The input data set defaults to the input document library JCL statements, which are always required with the SCRIPT command.
2. The primary document, DSMIVC30, is found in public library 1 314 151. Thereafter, for all imbedded files and macros encountered in the document, DLF searched the following libraries in this order:
 - Your private library 88
 - Your project library 1 000
 - Public library 9 999 because it was identified in the SEARCH option
 - The SCRIPT/VS system library, which is 1 314 151, unless changed by your installation.
3. Because DSMIVC30 has a table of contents, the TWOPASS option was specified, causing SCRIPT/VS to make two passes through the document to resolve forward page number references.

4. When you use utility files (DSMUTMSG, DSMUTTOC, DSMUTWTF, DSMUTCTF), you must allocate them to a sequential data set. See Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics" on page 237 for the attributes of these utility files.
5. A utility data set was included in the JCL: DSMUTWTF (DSMUWTF in VSE). This permits the write-to-file functions to be performed, if requested in the document.
6. Two utility data sets were also included: DSMUTTOC (DSMUTOC in VSE), to accumulate the table of contents, and DSMUTMSG (DSMUMSG in VSE), to accumulate Document Composition Facility messages.
7. The required ddnames for the SCRIPT/VS utility data sets are listed in Figure 15. The data set default characteristics are included in Figure 33 on page 240 and Figure 34 on page 241.
8. DLF messages were printed in the data set identified by DSMLIST (OS/VS2) or SYSLST (VSE).
9. BIND (11) requests that the document be printed with a left binding of 1 inch.
10. CONT requests that formatting continue if no error occurs greater than severity 8.
11. SPELLCHK requests that the document be checked for spelling errors.

SCRIPT/VS Function	SCRIPT/VS Control Word	OS/VS2 ddname	VSE DLBL
Messages	.mg	DSMUTMSG	DSMUMSG
Table of Contents	.tc, .pt	DSMUTTOC	DSMUTOC
Write to File	.wf	DSMUTWTF	DSMUWTF
Terminal Input	.te, .rv	DSMTERMI	DSMITRM
Terminal Output	.ty	DSMTERMO	DSMOTRM

Figure 15. The SCRIPT/VS Utility Data Sets

PROCESSING ATMS DOCUMENTS

ATMS-III has the facility to send documents in fixed 80-character records to a sequential data set. This is done by the ATMS module DOKFT00. We will call such documents FT00 format.

Note: See Appendix G, "Relationship Of ATMS-III And The Document Library Facility" on page 259 for more information about processing ATMS documents using DLF.

The examples in this section pertain only to ATMS documents in FT00 format. They do not pertain to documents archived from ATMS-III that are to be imported into the document library.

The Problem: You have three documents in FT00 format. You would like to convert ATMSDOC1 so it can be formatted using SCRIPT/VS. You do not want to put ATMSDOC1 into the document library.

You would like to convert ATMSDOC2 to SCRIPT/VS symbols and macros and store it in the document library. You want to format it in a subsequent command.

You would like to put ATMSDOC3 in the document library, retaining its ATMS controls. In a subsequent command you want to convert and format the document.

The Solution: Use the SCRIPT and IMPORT commands. The JCL statements and DLF commands required to accomplish this task are shown in Figure 16 on page 117. This task requires that DCF be installed.

You should understand the material presented in Appendix G and the appropriate ATMS-III manual before proceeding with this example.

```

//ATMSCV JOB
//STEP1 EXEC PGM=DSMSPEXC
//SYSPRINT DD SYSOUT=A
//DSMLIST DD SYSOUT=A
//DSMUTMSG DD UNIT=SYSDA,SPACE=(TRK,(5,5))
//DSMINDIR DD DSN=DSMLIB.DIRECTRY,DISP=SHR
//DSMINLIB DD DSN=DSMLIB.SOURCE,DISP=SHR
//DSMPDIR DD DSN=DSMLIB.DIRECTRY,DISP=OLD
//DSMPTLIB DD DSN=DSMLIB.SOURCE,DISP=OLD
//ATMSTAP DD UNIT=TAPE,LABEL=(1,SL),
// VOL=SER=ATMS01,DISP=OLD,DSN=ATMS1
//ATMSDDA DD DSN=ADOC2,VOL=SER=ADISK,
// DISP=OLD,UNIT=3330
//ATMSDDB DD DSN=ADOC3,VOL=SER=ADISK,
// DISP=OLD,UNIT=3330
//SYSIN DD *
AUTH 88/CITADEL
SCRIPT FROM(ATMSTAP) PROCESS(DSMACAIP) (PROF(ATMSPRF2))
IMPORT ATMSDOC2 FROM(ATMSDDA) PROCESS(DSMACAIP) INDATA(ATMS) -
SOURCE(ATMS) DATA(SCRIPT)
SCRIPT ATMSDOC2 DATA(SCRIPT) (PROF(ATMSPRF2))
IMPORT ATMSDOC3 FROM(ATMSDDB) DATA(ATMS) SOURCE(ATMS)
SCRIPT ATMSDOC3 DATA(ATMS) (PROF(ATMSPRF2))
/*

// JOB ATMSCV
// DLBL DSMIDIR,'DSMLIB.DIRECTRY',,VSAM
// EXTENT ,USRDSK
// DLBL DSMILIB,'DSMLIB.SOURCE',,VSAM
// EXTENT ,USRDSK
// DLBL DSMPDIR,'DSMLIB.DIRECTRY',,VSAM
// EXTENT ,USRDSK
// DLBL DSMPLIB,'DSMLIB.SOURCE',,VSAM
// EXTENT ,USRDSK
// DLBL DSMMSG,'MESSAGE FILE',0,SD
// EXTENT SYS028,USRDSK,1,0,4978,10
// ASSGN SYS028,3330,VOL=USRDSK,SHR
// TLBL ATMSTAP,'ATMS1',,ATMS01
// ASSGN SYS021,x'380'
// DLBL ATMSDDA,'ADOC2',0,SD
// EXTENT SYS022,ADISK1
// ASSGN SYS022,3330,VOL=ADISK1,SHR
// DLBL ATMSDDB,'ADOC3',0,SD
// EXTENT SYS023,ADISK2
// ASSGN SYS023,3330,VOL=ADISK2,SHR
// EXEC DSMSPEXC,SIZE=AUTO
AUTH 88/CITADEL
ENVIRONMENT DLBL(DSMMSG) DEV(3330 028)
ENVIRONMENT TLBL(ATMSTAP) DEV(2400 021)
SCRIPT FROM(ATMSTAP) PROCESS(DSMACAIP) (PROF(ATMSPRF2))
ENVIRONMENT DLBL(ATMSDDA) DEV(3330 022)
IMPORT ATMSDOC2 FROM(ATMSDDA) PROCESS(DSMACAIP) INDATA(ATMS) -
SOURCE(ATMS) DATA(SCRIPT)
SCRIPT ATMSDOC2 DATA(SCRIPT) (PROF(ATMSPRF2))
ENVIRONMENT DLBL(ATMSDDB) DEV(3330 023)
IMPORT ATMSDOC3 FROM(ATMSDDB) DATA(ATMS) SOURCE(ATMS)
SCRIPT ATMSDOC3 DATA(ATMS) (PROF(ATMSPRF2))
/*
/&

```

Figure 16. Examples of ATMS Conversion Processing

Notes:

1. Because the ATMS conversion routine does not generate any of the SCRIPT/VS controls that require special utility data sets, it was not necessary to include JCL statements for SCRIPT/VS data sets other than the message data set

(DSMUTMSG/DSMUMSG). (For VSE, the ENVIRONMENT command is required for the message file if the defaults are not acceptable.)

2. The Processing of ATMSDOC1:

- ATMSDOC1 was read from a data set named ATMS1 on a tape labeled ATMS01, described by JCL statement ATMSTAP. Because the PROCESS operand was specified with the SCRIPT command, DLF first passed this document, one record at a time, to the load module specified in the process parameter. When the ATMS conversion routine completed the processing of each record, it returned the record to DLF, which, in turn, passed the converted record to SCRIPT/VS.
- SCRIPT/VS required profile ATMSPRF2 to format a document that has been processed by the ATMS conversion routine. The profile was used to resolve the SCRIPT/VS symbols placed in the converted document by the ATMS conversion routine.
- ATMSDOC1 (which is not known by any name to DLF because it is not coming from or going to the document library) was formatted for a line printer (such as the 1403) and printed on the device specified in DSMLIST (for OS/VS2) or SYSLST (for VSE).

3. The Processing of ATMSDOC2:

- ATMSDOC2 was read from a sequential data set (ADOC2) on ADISK, described by JCL statement ATMSDDA. Again, DLF passed this document, one record at a time, to the load module specified by the PROCESS keyword. When the ATMS conversion routine completed the processing of each record, it returned the record to DLF, which stored ATMSDOC2 in private library 88. ATMSDOC2 was assigned a data type of SCRIPT.
- The DSMACAIP (ATMS to SCRIPT conversion processor) replaces ATMS controls with SCRIPT/VS macro calls during its conversion processing. SCRIPT/VS uses the SCRIPT/VS system library 1 314 151 to resolve these macros when the document is formatted.
- ATMSDOC2 was read from private library 88 and formatted. SCRIPT/VS again required that the ATMS profile ATMSPRF2 be specified to resolve the symbols which were added by the ATMS conversion routine. The data type of the document to be formatted had to be SCRIPT.
- As with ATMSDOC1, document ATMSDOC2 took the formatting defaults. It was formatted for a line printer (such as the 1403) and printed on the device specified by DSMLIST (for OS/VS2) or SYSLST (for VSE), taking all the SCRIPT/VS defaults.

4. The Processing of ATMSDOC3:

- ATMSDOC3 was read from data set ADOC3 on ADISK, described by JCL statement ATMSDDB, and was stored in private library 88. It was named ATMSDOC3 and assigned the data type ATMS.
- ATMSDOC3 was then read from private library 88 to format. Because the data type is other than SCRIPT, a processor is implied. The processor must have previously been defined with the DEFINE PROCESS command specifying an INDATA type of ATMS and an OUTDATA type of SCRIPT.
- Each record of ATMSDOC3 was passed to the ATMS conversion routine for processing before being formatted.

- ATMSDOC3 was formatted, again requiring that the ATMSPRF2 profile be specified to resolve the symbols included in the converted document by the ATMS conversion routine.
- The document was formatted for a line printer (such as the 1403) and printed on the device specified by DSMLIST (for OS/VS2) or SYSLST (for VSE). All SCRIPT/VS defaults were taken.

ARCHIVING AND RETRIEVING YOUR DOCUMENTS AND THEIR VERSIONS

The Problem: You wish to archive some inactive documents from the document library to a sequential external data set (ARCD5) you use for that purpose.

- The first two documents, NOVERS and UNIQUE, are archived to your external data set. They have no versions.
- The third document, MEM01, which has five versions, is to have versions 3 and 4 archived to your external data set.
- The fourth document, MEM02, has several versions and you want to archive all but the latest version.
- The fifth document, LETTER1, has ten versions and you want to archive versions 1 through 5.

The Solution: Use the ARCHIVE DOCUMENT command. The JCL and ARCHIVE DOCUMENT commands required to accomplish this task are shown in Figure 17 on page 120.

```

//ARCHIVE      JOB
//STEP1 EXEC   PGM=DSMSPEXC
//SYSPRINT DD  SYSOUT=A
//DSMLIST DD   SYSOUT=A
//DSMINDIR DD  DSN=DSMLIB.DIRECTRY,DISP=SHR
//DSMINLIB DD  DSN=DSMLIB.SOURCE,DISP=SHR
//DSMPDIR DD   DSN=DSMLIB.DIRECTRY,DISP=OLD
//DSMPTLIB DD  DSN=DSMLIB.SOURCE,DISP=OLD
//ARCHIVDS DD  DSN=ARCDS,VOL=SER=DISKA,UNIT=3330,DISP=MOD
//SYSIN DD *
AUTH 88/CITADEL
ARCHIVE NAMELIST (*) TO(ARCHIVDS)
NOVERS
UNIQUE

ARCHIVE DOCUMENT MEMO1 VERSION(3:4) TO(ARCHIVDS)
ARCHIVE DOCUMENT MEMO2 VERSION TO(ARCHIVDS)
ARCHIVE DOCUMENT LETTER1 TO(ARCHIVDS) VERSION(:5)
/*

```

for OS/VS2

```

// JOB      ARCHIVE
// DLBL     DSMIDIR,'DSMLIB.DIRECTRY',,VSAM
// EXTENT   ,USRDSK
// DLBL     DSMILIB,'DSMLIB,SOURCE',,VSAM
// EXTENT   ,USRDSK
// DLBL     DSMPDIR,'DSMLIB.DIRECTRY',,VSAM
// EXTENT   ,USRDSK
// DLBL     DSMPLIB,'DSMLIB,SOURCE',,VSAM
// EXTENT   ,USRDSK
// DLBL     ARCHIVDS,'ARCDS',0,SD
// EXTENT   SYS025,USRDSK
// ASSIGN   SYS025,3330,VOL=USRDSK,MOD
// OPTION   NOLIST
// EXEC     DSMSPEXC,SIZE=AUTO
AUTH 88/CITADEL
ENVIRONMENT DLBL(ARCHIVDS) DEV(3330 025)
ARCHIVE NAMELIST (*) TO(ARCHIVDS)
NOVERS
UNIQUE

ARCHIVE DOCUMENT MEMO1 VERSION(3:4) TO(ARCHIVDS)
ARCHIVE DOCUMENT MEMO2 VERSION TO(ARCHIVDS)
ARCHIVE DOCUMENT LETTER1 TO(ARCHIVDS) VERSION(:5)
/*
/8

```

for VSE

Figure 17. Examples of the ARCHIVE DOCUMENT Command

Notes:

1. The JCL statements and commands shown in the example archive your documents to data set ARCDS, which is identified by the JCL statement labeled ARCHIVDS.
2. You must specify the TO keyword if you use the NAMELIST or ALL option. In non-OS/VS2 environments, TO must be specified at all times. See Chapter 3, "General User Commands," for the options of this command.
3. For MEMO2, by specifying only VERSION (without a number) on the ARCHIVE DOCUMENT command, you archive all versions except the latest.

4. All versions through the one specified (for LETTER1, five) are archived when you include the VERSION option and a single number with a preceding colon.
5. Documents that have been archived no longer exist in the document library. The DER, however, is retained in the directory data set to indicate that the document exists but has been archived. You can get a list of your archived documents by the ARCHIVE option of the LIST DOCUMENT command. See "Commonly Used Command Operands" in Chapter 2 for details.
6. Note that output document library JCL statements as well as input document library JCL statements were included in the archive example. They were required for ARCHIVE because DERs are written back to the directory data set to indicate that the documents exist but have been archived.
7. The name list shown in the job input stream of the example contains entries with only document name specified. Additional information in name list entries can be required to uniquely identify the document to be archived (Appendix H "NAMELIST Used in the ARCHIVE and RETRIEVE Commands" contains a detailed description of the namelist entries).

The Problem: You now wish to:

- Retrieve version 2 of archived document MEM02 from the archive data set into the active document library.
- Retrieve the latest archived version of document LETTER1 from the archive data set into the active document library.

The Solution: Use the RETRIEVE DOCUMENT command. The JCL statements required to retrieve the documents archived in the example are the same as those used to archive them. The required RETRIEVE DOCUMENT commands are shown below.

```
AUTH 88/CITADEL
RETRIEVE DOCUMENT MEM02 FROM(ARCHIVDS) VERSION(2)
RETRIEVE DOCUMENT LETTER1 FROM(ARCHIVDS)
```

The latest version is retrieved if the VERSION keyword is omitted.

COPYING VERSIONS OF DOCUMENTS

In the following situations, assume you own documents MAJOR and MINOR in your private library 88. MAJOR has five versions and MINOR has no versions. You will be copying to several users.

The Problem: You wish to:

- Copy version 1 of MAJOR within your own library with a new name of MAGIC.
- Copy all versions of MAJOR within your own library with a new name of MIDDLE.
- Copy the latest version of MAJOR to your project library 1 000.
- Copy MINOR to the system library, public library 1 314 151.
- Copy MINOR to your project library 1 000 and assign versions to MINOR in library 1 000.

The Solution: Use the COPY DOCUMENT command.

The JCL and commands required to complete these jobs are shown in Figure 18 on page 122.

```

//COPY      JOB
//STEP1 EXEC PGM=DSMSPEXC,PARM='LIST'
//SYSPRINT DD SYSOUT=A
//DSMLIST DD SYSOUT=A
//DSMINDER DD DSN=DSMLIB.DIRECTRY,DISP=SHR
//DSMINLIB DD DSN=DSMLIB.SOURCE,DISP=SHR
//DSMPDIR DD DSN=DSMLIB.DIRECTRY,DISP=OLD
//DSMPLIB DD DSN=DSMLIB.SOURCE,DISP=OLD
//SYSIN DD *
AUTH 88/CITADEL
COPY DOCUMENT MAJOR VERSION(1) TOUSER(88) NEWNAME(MAGIC)
COPY DOCUMENT MAJOR VERSION(ALL) NEWNAME(MIDDLE)
COPY DOCUMENT MAJOR VERSION TOUSER(1000)
COPY DOCUMENT MINOR TOUSER(1314151)
COPY DOCUMENT MINOR TOUSER(1000) NEWVERSION
/*

```

for OS/VS2

```

// JOB      COPY
// DLBL     DSMIDIR,'DSMLIB.DIRECTRY',,VSAM
// EXTENT   ,USRDSK
// DLBL     DSMILIB,'DSMLIB.SOURCE',,VSAM
// EXTENT   ,USRDSK
// DLBL     DSMPDIR,'DSMLIB.DIRECTRY',,VSAM
// EXTENT   ,USRDSK
// DLBL     DSMPLIB,'DSMLIB.SOURCE',,VSAM
// EXTENT   ,USRDSK
// OPTION   LISTX
// EXEC     DSMSPEXC,SIZE=AUTO
AUTH 88/CITADEL
COPY DOCUMENT MAJOR VERSION(1) TOUSER(88) NEWNAME(MAGIC)
COPY DOCUMENT MAJOR VERSION(ALL) NEWNAME(MIDDLE)
COPY DOCUMENT MAJOR VERSION TOUSER(1000)
COPY DOCUMENT MINOR TOUSER(1314151)
COPY DOCUMENT MINOR TOUSER(1000) NEWVERSION
/*
/&

```

for VSE

Figure 18. Examples of the COPY DOCUMENT Command

Notes:

1. Because user 88 is not an administrator, he can copy only into his private library, his project library (if not controlled), or any noncontrolled public library.
2. Version 1 of MAJOR will be copied from user 88 to himself with the new name, MAGIC. MAGIC will also have versions.
3. All versions of document MAJOR will be copied from user 88 to himself with the new name of MIDDLE. MIDDLE will also have versions. Note that if TOUSER is not specified, the default is the library of the current user as specified in the latest AUTH command.
4. The latest version of MAJOR will be copied to project library 1 000. The name will remain the same, and MAJOR (1 000) will also have versions.
5. MINOR will be copied to public library 1 314 151, and it will not have versions.
6. MINOR will be copied to project library 1 000. Because of the NEWVERSION keyword, MINOR will have versions in library 1 000 even though it had none in library 88. The first

version (and only current version) of MINOR in library 1 000 will be assigned version number 1.

7. A new version name, without a version number, will take the version number of the original document name. For example, if MAJOR(5) were copied with the new version name of MAGIC, the copy is MAGIC(5).

CHANGING DOCUMENT NAMES AND ATTRIBUTES

The Problem: You wish to change the attributes of several of your documents. They are OLDDOC, RED, and DELTA, which have no versions; and documents NEWDOC and CHARGE, which do have versions.

The Solution: Use the ALTER DOCUMENT command. The JCL and commands required to complete these jobs are shown in Figure 19.

<pre>//ALTER JOB //STEP1 EXEC PGM=DSMSPEXC,PARM='NOLIST' //SYSPRINT DD SYSOUT=A //DSMLIST DD SYSOUT=A //DSMINDER DD DSN=DSMLIB.DIRECTRY,DISP=SHR //DSMINLIB DD DSN=DSMLIB.SOURCE,DISP=SHR //DSMPDIR DD DSN=DSMLIB.DIRECTRY,DISP=OLD //DSMPLIB DD DSN=DSMLIB.SOURCE,DISP=OLD //SYSIN DD * AUTH 88/CITADEL ALTER DOCUMENT DELTA NEWCLASS(BOOK) ALTER DOCUMENT OLDDOC DATA(ATMSIII) NEWDATA(SCRIPT) ALTER DOCUMENT NEWDOC VERSION(5) LOCK ALTER DOCUMENT RED NEWNAME(BLUE) ALTER DOCUMENT CHARGE UNLOCK VERSION(13) /*</pre>	for OS/VS2
---	------------

<pre>// JOB ALTER // DLBL DSMIDIR,'DSMLIB.DIRECTRY',,VSAM // EXTENT ,USRDSK // DLBL DSMILIB,'DSMLIB.SOURCE',,VSAM // EXTENT ,USRDSK // DLBL DSMPDIR,'DSMLIB.DIRECTRY',,VSAM // EXTENT ,USRDSK // DLBL DSMPLIB,'DSMLIB.SOURCE',,VSAM // EXTENT ,USRDSK // OPTION NOLISTX // EXEC DSMSPEXC,SIZE=AUTO AUTH 88/CITADEL ALTER DOCUMENT DELTA NEWCLASS(BOOK) ALTER DOCUMENT OLDDOC DATA(ATMSIII) NEWDATA(SCRIPT) ALTER DOCUMENT NEWDOC VERSION(5) LOCK ALTER DOCUMENT RED NEWNAME(BLUE) ALTER DOCUMENT CHARGE UNLOCK VERSION(13) /* /&</pre>	for VSE
---	---------

Figure 19. Examples of the ALTER DOCUMENT Command

Notes:

1. You need not specify data type unless the document has a data type that is different from the current user's default data type as indicated in the UPR.
2. DELTA will be given a new class of BOOK.

3. OLDDOC, which has a current data type of ATMSIII, will be given a new data type of SCRIPT.
4. Version 5 of document NEWDOC will be locked, and no commands other than COPY, ARCHIVE, RETRIEVE, LIST or another ALTER DOCUMENT command (for UNLOCK option only) can be used to access it. Because NEWDOC has versions, no alterations can be made to it with the exception of lock status.
5. RED will be given a new name of BLUE.
6. The 13th version of CHARGE will be unlocked. CHARGE was previously locked and no alterations could be made to it except to unlock it. No other options of the ALTER DOCUMENT command can be used.

Part 3 of the Document Library Facility Guide gives DLF administrators the information they need to set up and control the document library.

- Chapter 5, "Administrator Commands" on page 127 provides an alphabetic list of the commands available to the library administrator.
- Chapter 6, "Setting Up the Document Library" on page 195 describes the steps required to:
 - Define the VSAM data sets DLF requires
 - Install DLF
 - Install processors
 - Define required profiles
 - Provide RACF security.
- Chapter 7, "Administering the Document Library" on page 205 provides information that allows a DLF administrator to:
 - Authorize new users
 - Copy profiles and documents
 - Ensure adequate space for the document library
 - List library space usage
 - Allocate additional space for the document library
 - Reorganize the document library
 - Modify an attribute profile
 - Define user maps for external systems
 - Define classes of data.

CHAPTER 5. ADMINISTRATOR COMMANDS

The library administrator creates and maintains the document library by using administrator commands. These commands are summarized in the table below.

In addition to the commands shown here, the administrator can issue all general user commands against his own library. The administrator can also use general user commands on his own documents and on documents belonging to any library user as well. See Chapter 3, "General User Commands," for a detailed discussion of those commands.

Command	Function
ACCOUNT	Creates accounting records (OS/VS2 SMF record format type 47) from information contained in user profiles. These profiles can be in the document library directory or in a sequential data set created by the COPY OUT command.
ALTER CLASS	Changes the cluster name associated with a class.
ALTER CLUSTER	Changes a cluster profile and formats the cluster if new space has been added to it.
ALTER DOCUMENT	Changes the name or characteristics of any document in the document library.
ALTER MAP	Changes the mapping record that associates an external user with a library user number.
ALTER PROCESS	Changes the entry name of a processor.
ALTER SYSTEM	Formats any additional space added to the basic 4K ESDS cluster since the last DEFINE SYSTEM or ALTER SYSTEM command was issued. Also can change the RACF protection, ENQ/LOCK RNAME, and the operator routing code used by the library program.
ALTER USER	Alters the description of a user or a user's library, or both.
ARCHIVE ALL	Archives the documents of any or all library users.
COPY IN	Restores all or part of a document library.
COPY LIBRARY	Copies a document library from one set of VSAM data sets to another.
COPY OUT	Provides a backup copy of all or part of a document library.
COPY USER	Copies all of a user's library from one document library to another or copies all documents from one user library to another user library within the same document library.
DEFINE CLASS	Defines a document class and associates it with an ESDS cluster.
DEFINE CLUSTER	Defines and formats new ESDS clusters.

Figure 20 (Part 1 of 2). Summary of the Administrator Commands

Command	Function
DEFINE MAP	Maps an external user to a library user number.
DEFINE PROCESS	Defines a processor to DLF.
DEFINE SYSTEM	Establishes the document library by formatting the space for the default 4K cluster, defining RACF protection, specifying the ENQ/LOCK RNAME, and specifying the number of the OS operator route code.
DEFINE USER	Defines a new user and his library to DLF.
LIST DOCUMENT (Changed)	Lists the attributes of any or all of the documents in the document library.
LIST MAP (Changed)	Lists all of the user mapping records (UMRs) that are defined in the directory data set.
LIST USER (Changed)	Lists the user profile records (UPRs) of any or all library users.
LIST SYSTEM (New)	Lists information from the system profile record (SPR).
PROTECT	Changes the password or share status of any document stored in the document library.
PURGE CLASS	Purges a class entry record (CER) from the directory data set.
PURGE CLUSTER	Purges a cluster profile record (CPR) from the directory data set.
PURGE DOCUMENT	Purges documents from any library in the document library.
PURGE MAP	Purges a user mapping record (UMR) associating an external user with a library user number.
PURGE PROCESS	Purges a processor profile record (PPR) from the directory data set.
PURGE USER	Purges a user profile record (UPR) and all of the documents in the user's library from the directory data set. Deletes mapping records to this user.
RETRIEVE ALL	Retrieves the documents of any or all library users from an external sequential data set.

Figure 20 (Part 2 of 2). Summary of the Administrator Commands

ACCOUNT

This command generates accounting records reporting the amount of storage space each user is currently using in private, project, and public libraries. Information is taken from the user profile record (UPR) and written in the same format as the System Management Facilities (SMF) type 47 records. SMF is a control program option of OS/VS2. For OS/VS2, records are written as a sequential data set or member of a partitioned data set. For VSE, the accounting file must be a predefined, VSAM entry-sequenced data set. The format for the accounting record can be found in Appendix E, "Format of the Accounting Record."

The ACCOUNT command requires administrator authority.

ACCOUNT	[{ ddname }]
	FROM ({ DSN(dsname) })
		{ DSN(dsname(member)) })
		{ DSN('dsname') }	
		{ DSN('dsname(member)') }	
	[{ ddname }]
	TO ({ DSN(dsname) })
		{ DSN(dsname(member)) })
		{ DSN('dsname') }	
		{ DSN('dsname(member)') }	
	[USER ({ number })
		{ number1:number2 }	
]

- FROM** This operand specifies the external sequential data set or partitioned data set member containing the UPRs from which the accounting information is taken. This data set must have been created by a COPY OUT command. If this operand is omitted, the input document library specified in the JCL is used.
- ddname** is the name on the DD, DLBL, or TLBL statement that identifies the FROM data set.
- DSN...** is valid in OS/VS2 only. It indicates that the FROM data set is dynamically allocated. If the specified data set name is in quotation marks, it is used as given. If not in quotation marks, it is qualified as described in Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics." COPY is the suffix qualifier.
- TO** This operand specifies the data set or partitioned data set member to which the accounting records are written. If this keyword is omitted, the default ddname is SMF for OS/VS2 and DSMACNT for VSE.
- ddname** is the name on the DD, DLBL, or TLBL statement identifying the TO data set.
- DSN...** is valid in OS/VS2 only. It indicates that the TO data set is dynamically allocated. If the specified data set name is in quotation marks, it is used as given. Otherwise it is qualified as described in Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics." ACCOUNT is the suffix qualifier.
- USER** This parameter specifies the library number(s) of the user(s) for which accounting information is generated. If this operand is omitted, accounting information is

generated for all library users defined in the document library (or whose profiles are included in the COPY OUT data set if FROM was specified). See dlib# under "Commonly Used Command Operands" in Chapter 2 for the characteristics of a library user number. USER without a number is invalid.

number is the library user number of a single user from whose profile accounting information is generated.

number1:number2 is the inclusive range of library user numbers from which accounting information is to be generated.

Notes:

1. If no FROM data set is designated, the input document library is assumed and input document library JCL statements are then required.
2. If dynamic allocation (available for OS/VS2 only) is not being used, JCL statements are required for the data set on which accounting records are written. The accounting data set must be:
 - a. A predefined sequential file or partitioned data set member on tape or disk (for OS/VS2)
 - b. A predefined VSAM entry-sequenced data set (for VSE).
3. For VSE, the accounting file is a VSAM entry-sequenced data set created using the Access Method Services DEFINE CLUSTER command with the following parameters:

NONINDEXED RECORDSIZE(128 128)

All other parameters are optional, but we recommend that you not specify UNIQUE or SHAREOPTIONS(3). If you specify SUBALLOCATION rather than UNIQUE, DLF can dynamically extend the accounting data set.

4. The records produced by the ACCOUNT command include:
 - a. Library user number
 - b. Time and date when the accounting record was written
 - c. Amount of direct-access storage occupied by the user's owned library
 - d. Amount of direct-access storage occupied by the user's documents in project and public libraries.

CAUTION:

A data set member is cleared when opened for output, even if the DLF command subsequently fails.

Examples

- This example scans the sequential input data set specified by the DD, DLBL, or TLBL statement named INDD, creates accounting records from all the UPRs in the data set, and writes these records to the data set specified by the DD or DLBL statement labeled OUTDD. The data set specified by INDD must have been created with the COPY OUT command.

```
AUTH 123/ADMPASS  
ACCOUNT FROM(INDD) TO(OUTDD)
```

- This example scans the directory data set specified by the DSMINDIR (for OS/VS2) or DSMIDIR (for VSE) JCL statement and creates accounting records from all the UPRs. These records are written to the data set specified by the DD statement named SMF for OS/VS2 or the DLBL statement named DSMACNT for VSE.

```
AUTH 123/ADMPASS  
ACCOUNT
```

- This example scans the directory data set specified by the DSMINDIR (for OS/VS2) or DSMIDIR (for VSE) JCL statement and creates accounting records from UPR 20 157 through 21 764, inclusive. These records are written to the data set specified by a DD statement (for OS/VS2) or DLBL statement (for VSE) labeled SUBSET.

```
AUTH 123/ADMPASS  
ACCOUNT USER(20157:21764) TO(SUBSET)
```

- This example defines an accounting data set for VSE.

```
// EXEC IDCAMS,SIZE=AUTO  
  DEFINE CLUSTER -  
    (NAME(DSMLIB.ACCOUNT) NONINDEXED -  
    VOLUMES(DSMVOL) CYLINDERS(2) RECORDSIZE(128 128))  
/*
```

ALTER

An administrator can execute seven ALTER commands:

- ALTER CLASS
- ALTER CLUSTER
- ALTER DOCUMENT
- ALTER MAP
- ALTER PROCESS
- ALTER SYSTEM
- ALTER USER.

Each ALTER command changes one of the various profiles in the directory data set. All of these commands except ALTER DOCUMENT change profiles created by previous, corresponding DEFINE DOCUMENT commands. ALTER DOCUMENT changes the document entry record (DER) of a document currently existing in the document library and is the only ALTER command that does not require administrator authority.

To understand the description of the ALTER commands that follow, the reader must have read and understood the material in the section "Commonly Used Command Operands" in Chapter 2.

ALTER CLASS

This command alters the cluster name associated with a class of documents by changing the class entry record (CER).

ALTER CLASS	name CLUSTER(name)
-------------	--------------------

name This operand specifies the name of the class. name must be defined by a previous DEFINE CLASS command.

CLUSTER(name) This operand specifies the name of the new cluster in which a document with this class is normally stored. The specified name must be defined by a previous DEFINE CLUSTER command.

Notes:

1. The output document library JCL statements are required.
2. Both name and CLUSTER(name) are required.
3. The command operands must be entered in the order shown above.

Example

This example changes to LONGER the cluster name associated with the class name of LETTER. All subsequent imports of documents with a class of LETTER and no specific cluster designation will be put into the cluster named LONGER.

```
AUTH 123/ADMPASS
ALTER CLASS LETTER CLUSTER(LONGER)
```

ALTER CLUSTER

This command changes a cluster profile record (CPR) and formats the cluster if space has been added.

ALTER CLUSTER	name [INDD(ddname)] [OUTDD(ddname)] [PASSWORD ({ VSAM-password }) { 'VSAM-password' }]
---------------	--

- name** This operand specifies the CPR to be changed. name must be defined by a previous DEFINE CLUSTER command.
- INDD(ddname)** This operand specifies the name of the DD (OS/VS2) or DLBI (VSE) statement that must subsequently be included in the JCL to define this cluster to DLF for input.
- If you do not specify the INDD operand, DLF will not change the INDD value in the CPR.
- OUTDD(ddname)** This operand specifies the name of the DD (OS/VS2) or DLBI (VSE) statement that must be included in the JCL to define this cluster to DLF for output.
- If you do not specify the OUTDD operand, DLF will not change the OUTDD value in the CPR for the cluster being modified.
- PASSWORD(VSAM-password)** This operand specifies the password in the Access Method Services DEFINE or ALTER statement for this cluster. This password is not used by DLF and need not be specified unless the document library is to be accessed directly by a non-Documents Library Facility program that requires the password for authorization or OPEN processing or both. VSAM passwords are 1-8 characters long. If they contain any special characters they must be enclosed in quotation marks.
- If you do not specify the PASSWORD operand, DLF will not change the VSAM-password value in the CPR for the cluster being modified.

Notes:

1. The ddnames for the default cluster (DSMPTLIB) cannot be changed. An error will result if you attempt to do so.
2. The output document library JCL statements are required.

Example

This example alters the CPR for the cluster LONGER. Hereafter, this cluster is identified to DLF by the DD (OS/VS2) or DLBL (VSE) JCL statement(s) named DSMIPUT (when the cluster is used as input) or DSMOPUT (when the cluster is used as output).

```
AUTH 123/ADMPASS  
ALTER CLUSTER LONGER INDD(DSMIPUT) OUTDD(DSMOPUT)
```

ALTER DOCUMENT

This command locks or unlocks a document, changes its attributes, or changes its name. If only one version of a document exists, all the operands of this command can be used to alter it. If more than one version exists, only the DATA, VERSION, and LOCK or UNLOCK operands can be used.

The ALTER DOCUMENT command does not require administrator authority. However, general users can alter only their own documents and the documents in their own libraries regardless of ownership. See Chapter 3, "General User Commands." An administrator can alter any document in the document library.

ALTER DOCUMENT	docid
	<div>[DATA(name)]</div> <div>[VERSION(number)]</div> <div><div>[NEWNAME ({ dname, }) { 'dname' }]</div><div>[NEWDATA(name)]</div><div>[NEWCLASS(name)]</div><div>[NEWSOURCE(name)]</div><div>[LOCK UNLOCK]</div></div>

- docid** This operand specifies the current document identification of the document to be altered.
- DATA(name)** This operand specifies the current data type of the document to be altered. If not specified, the data type default for the current authorized user is used. A zero (0) placed in the name field indicates that a null data type has been assigned to this document.
- VERSION(number)** This operand specifies the specific version of the document to be altered. If a document has versions, then VERSION(number) must be specified. If a document has more than one version, only its lock status can be altered by this command.
- NEWNAME(dname)** This operand specifies the document's new name and may or may not be entered as a quoted string. If NEWNAME is not specified, the document name is not changed.
- NEWDATA(name)** This operand specifies the new data type for the document. If a zero is placed in the name field, the current data name is nullified. If NEWDATA is not specified, the data type for the document is not changed.
- NEWCLASS(name)** This operand specifies a new class for this document. If a zero is placed in the name field, the current class name is nullified. If NEWCLASS

is not specified, the class for the document is not changed.

NEWSOURCE(name) This operand specifies a new source attribute for this document. If a zero is placed in the name field, the current source attribute in the document entry record is nullified. If **NEWSOURCE** is not specified, the source attribute of the document is not changed.

LOCK This keyword specifies that the document is to be locked against normal access. It can be accessed only through the **COPY**, **ARCHIVE**, and **RETRIEVE** commands and the **LOCK** option of the **LIST** command until it is unlocked by another **ALTER** command. Only one version of a document can be locked by each **ALTER** command.

If you omit the **LOCK/UNLOCK** operand pair from the **ALTER DOCUMENT** command, no change is made to the document's lock status.

UNLOCK This keyword specifies that this document is to be unlocked and be accessible again to authorized users. Only one version of a document with multiple versions can be unlocked by each **ALTER** command.

If you omit the **LOCK/UNLOCK** operand pair from the **ALTER DOCUMENT** command, no change is made to the document's lock status.

Notes:

1. The output document library JCL statements are required.
2. When a document or a version of a document is locked, **UNLOCK** is the only keyword that can be used to alter the document.
3. Documents with null data types (expressed as **DATA(0)**), can be copied into libraries whose owners do not have a default data type of null. In order for a user to conveniently access these documents when they have been copied in, he should change their data type to the default data type of his UPR. This is done with an **ALTER DOCUMENT** command as shown in the example below:

```
AUTH 123/ADMPASS
ALTER DOCUMENT DOCA DATA(0) NEWDATA(SCRIPT).
```

4. **ALTER DOCUMENT** cannot be used on documents that have been archived.

Examples

- This example locks version 10 of document **MULTIPLE** in library 456.

```
AUTH 123/ADMPASS
ALTER DOCUMENT 456 MULTIPLE VERSION(10) LOCK
```

- This example changes the name and class of document **VARIABLE**.

```
AUTH 123/ADMPASS
ALTER DOCUMENT VARIABLE NEWNAME(STABLE) -
NEWCLASS(MEMO)
```

ALTER MAP

This command alters a user mapping record (UMR) that associates an external user with a library user number.

ALTER MAP	HOST('identifier') ID('identifier') { [USER(number)] [NEWHOST('identifier')] [NEWID('identifier')] }
-----------	---

- HOST('identifier')** This operand specifies the current identification for the external system. The identifier field must be a quoted string. One or two characters are allowed.
- ID('identifier')** This operand specifies how the external user is now identified on the external system. The identifier field must be a quoted string of from 1 through 10 characters.
- USER(number)** This operand specifies a new library user number to which the external user is mapped. If this parameter is omitted, the user number is unchanged.
- NEWHOST('identifier')** This operand specifies the new external system identification. The identifier field must be a quoted string. One or two characters are allowed. If this operand is omitted, the host identifier is unchanged.
- NEWID('identifier')** This operand specifies how the external user is now identified on the external system. The identifier field must be a quoted string of from 1 through 10 characters. If this operand is omitted, the user identification on the external system is unchanged.

Notes:

1. The output document library JCL statements are required.
2. HOST and NEWHOST identifier fields can contain any information; they are not restricted to data that identifies a specific system.
3. Multiple external users can be mapped to a single library user number.
4. The altered map cannot be identical to an existing map.
5. As an administrator, you can alter your own user mapping record.

Example

This example sets all subsequent mapping of non-DLF user IDENT in the external system of HI to library user 1 234.

```
AUTH 123/ADMPASS  
ALTER MAP HOST('HI') ID('IDENT') USER(1234)
```

ALTER PROCESS

This command changes the name of a processor in an existing processor profile record (PPR). The PPR must be created by a previous DEFINE PROCESS command. A PPR is created to associate a processor with a pair of data type names. The processor can be automatically invoked with the DLF commands IMPORT, EXPORT, READ, and SCRIPT. These commands specify or imply that the document to be acted upon currently has a data type name identical to the first of the pair of PPR data names. The document is to have a data type name identical to the second of the pair of PPR data names after the action is complete. Such a processor often converts formatting controls contained in a document from those recognized by one formatter (such as ATMS formatter) to those recognized by another (such as SCRIPT formatter). The type of formatting controls that are contained in a document are indicated by its data type. For details concerning processors, see Chapter 8, "Using DLF to Call Processors."

ALTER PROCESS	INDATA(name) OUTDATA(name) [ENTRY(name)]
---------------	---

INDATA(name) This operand specifies the first of a pair of data type names in the PPR. It shows the input data type of documents to be handled by the processor.

OUTDATA(name) This operand specifies the second of a pair of data type names in the PPR. It shows the output data type of documents handled by the processor.

ENTRY(name) This term specifies the entry point name of the module doing the special processing for this combination of data type names. If this operand is omitted or if a zero is specified for (name), the processor field in the PPR is nullified and no special processing is done for this combination of data type names. The PPR, however, still exists. Any command specifying this combination of data types does execute but no special processing is done. (name) must be one to eight characters and is changed to uppercase. It can contain any alphameric characters and the @, #, and \$. The first character must not be numeric.

Notes:

1. The output document library JCL statements are required.
2. The following table shows the correspondence of INDATA and OUTDATA as specified in the DEFINE PROCESS or ALTER PROCESS command with data pairs specified on other DLF commands.

<u>Command</u>	<u>Input Data Type</u>	<u>Output Data Type</u>
DEFINE PROCESS	INDATA	OUTDATA
ALTER PROCESS	INDATA	OUTDATA
IMPORT	INDATA	DATA
EXPORT	DATA	OUTDATA
READ	DATA	OUTDATA
SCRIPT	DATA	implied(SCRIPT)

Example

This example gives a new module entry point name for the processor associated with the INDATA/OUTDATA pair, ATMS/SCRIPT.

```
AUTH 123/ADMPASS  
ALTER PROCESS INDATA(ATMS) OUTDATA(SCRIPT) ENTRY(DSMACAIP)
```

ALTER SYSTEM

This command formats any space that was added to the default 4K cluster by Access Method Services since the last DEFINE SYSTEM or ALTER SYSTEM command. It can also change the RACF resource name, the ENQ/LOCK RNAME, or the operator routing code number associated with this document library or all three.

ALTER SYSTEM	[LIBRARY(name)]
	[RACFNAME(name)]
	[ROUTE(number)]

LIBRARY(name) This operand specifies a new name to be used as the RNAME in the ENQ macro (OS/VS2) or as the LOCK name (VSE). (name) can be 1 to 12 alphanumeric characters long or the number 0 but cannot begin with a numeric character. Use 0 as the (name) to nullify the current library name. If LIBRARY is omitted, the library name is unchanged.

CAUTION:

LIBRARY(name) must be in effect for certain invocations of the document library. See Chapter 2 under "JCL Statements" for details.

RACFNAME(name) This operand specifies a new RACF resource class name. The name can be 4 to 8 alphanumeric characters, or the characters \$, #, and @. The first character must be alphabetic or one of the characters \$, #, or @. Use a zero (0) as the RACFNAME to nullify the current RACFNAME. If RACFNAME is omitted, the name is unchanged. The operand is valid only in OS/VS2.

ROUTE(number) This operand specifies a new OS operator ROUTE code as defined in the write-to-operator (WTO) macro. Messages and replies resulting from the OPERATOR keyword of the ARCHIVE and RETRIEVE commands are sent to and received from this OS operator. The number field entry must be in the range of 1 to 16. Specifying 0 will give the current default DLF routing code (3). See the OS/VS2 MVS Supervisor Services and Macro Instructions manual for information on the WTO macro. This operand is not valid in VSE. If this keyword is omitted, the ROUTE code remains unchanged.

Note: The output document library JCL statements are required.

Example

This example specifies a new RACF security name.

```
AUTH 123/ADMPASS
ALTER SYSTEM RACFNAME(NEWKEY)
```

ALTER USER

This command changes the description or characteristics of a user's library by modifying the user's UPR.

ALTER USER	number
	<div><div>[PASSWORD({/password }) {/password'}) NOPASS])</div><div>[SPACE(number)])</div><div>[CLASS(name)])</div><div>[DATA(name)])</div><div>[SOURCE(name)])</div><div>[PROJLIB ({ number }) { NONE }])</div><div>[LOCK UNLOCK])</div><div>[ADM NOADM])</div><div>[CONTROL NOCONTROL])</div></div>

number

This parameter specifies a library user number.

**PASSWORD
or
NOPASS**

This operand specifies the new or changed password for this user. If specified, it must be given with each subsequent AUTH command for this user. The characteristics of the user password are the same as those of a document password except that they are associated with a user instead of a document. If PASSWORD or NOPASS is not specified, the specified user's password is not changed. NOPASS specifies that the existing user password is to be deleted.

SPACE(number)

This operand specifies a new maximum amount of document library direct-access space assigned to this library user number. This maximum is specified as a number from 0 to 4 000 000 in 1 000-byte units. Thus, the largest number of bytes allowed is 4 000 000 000. If a zero is specified in the (number) field, the user cannot store documents in the document library. If SPACE(number) is not specified, the current maximum is unchanged.

CLASS(name)	This operand specifies a new name for the user's default class. A zero in the name field nullifies the class designation in the UPR. If CLASS(name) is not specified, the user's default class is unchanged.
DATA(name)	This operand specifies a new name for the user's default data type. A zero in the name field nullifies the data designation in the UPR. If DATA(name) is not specified, the user's default data type is unchanged.
SOURCE(name)	This operand specifies a new name for the default source of the user's data. A zero in the (name) field nullifies the source designation in the UPR. If SOURCE(name) is not specified, the user's default source designation is unchanged.
PROJLIB(number)	<p>This operand specifies the new project library number for this user. This project library must be defined in the document library. If this user already owns documents in a different project library, no change is made, and an error message results.</p> <p>If PROJLIB(NONE) is specified, the project library field in this UPR is nullified, and the user no longer has a project library assignment. If the user owns documents in the project library previously assigned, no changes are made to the UPR and an error message results. If PROJLIB is omitted, the user's project library assignment is not changed.</p>
LOCK	<p>This keyword specifies that this user is no longer authorized to use the document library. When this request has been executed, the user cannot access the document library until an ALTER USER command with UNLOCK is issued. This command does not prevent other users from accessing documents owned by the locked user if the documents have share status or reside in a project or public library.</p> <p>If you omit the LOCK/UNLOCK operand pair from the ALTER USER command, no change is made to the lock status of the user being modified.</p>
UNLOCK	<p>This keyword specifies that the user is to be reinstated as a valid user.</p> <p>If you omit the LOCK/UNLOCK operand pair from the ALTER USER command, no change is made to the lock status of the user being modified.</p>
ADM	<p>This operand gives administrator authority to the user.</p> <p>If you omit the ADM/NOADM operand pair from the ALTER USER command, no change is made to this user's authority level.</p>
NOADM	This operand removes administrator authority from the user.

If you omit the ADM/NOADM operand pair from the ALTER USER command, no change is made to this user's authority level.

CONTROL

This operand specifies that a formerly uncontrolled public or project library is now controlled. Other users who are authorized to access this project or public library can no longer store documents in it. Documents already stored in this library by other users are not affected and they can still be accessed by those authorized to do so. The owner of this library can continue to store documents in the library even though it is controlled. This operand is invalid for private libraries.

If you omit the CONTROL/NOCONTROL operand pair from the ALTER USER command, no change is made to the control status of the specified user's library.

NOCONTROL

This operand specifies that a formerly controlled public or project library is now uncontrolled. This operand is invalid for private libraries.

If you omit the CONTROL/NOCONTROL operand pair from the ALTER USER command, no change is made to the control status of the specified user's library.

Notes:

1. The output document library JCL statements are required.
2. An administrator cannot lock himself or remove his own administrator authority. Another user with administrator authority must execute this command. In this way, at least one user with administrator authority exists in the document library at all times.
3. Administrators can alter their own UPRs (except ADM and LOCK as noted above). However, the new characteristics will not be in effect until the next AUTH command for that administrator is issued.
4. The library type (public, project, or private) cannot be changed by the ALTER USER command.
5. If an error condition is detected in any operand, an error message results and no changes are made for the ALTER command.
6. The user can specify a variable in any operand that is identical to the corresponding variable in the current UPR. In that case, the variable will not change.

Examples

- This example removes the project library assignment for user 201 and changes the password to SETH. If user 201 owns any documents in the project library, the command is not executed, and an error message results.

```
AUTH 123/ADMPASS  
ALTER USER 201 PROJLIB(NONE) PASSWORD(/SETH)
```

- This example denies user 201 access to the document library.

```
AUTH 123/ADMPASS  
ALTER USER 201 LOCK
```

ARCHIVE ALL

This command moves documents from the document library to an external sequential data set. Included in the move is the text of the document and its document entry record (DER). A copy of the DER is retained in the directory data set.

When a document has been archived, the only commands that can access it are the RETRIEVE commands. The RETRIEVE commands move the document back to the document library from the external sequential data set. A number of commands, however, can act on the copy of the DER retained in the directory data set; these include COPY IN, COPY OUT, COPY LIBRARY, PURGE USER, PURGE DOCUMENT, and LIST DOCUMENT (with the ARCHIVE option). Locked documents can be archived. When documents are archived, the user profile record (UPR) reflects the increase in available DASD space.

This command archives all documents meeting the selection criteria specified in the command.

To understand the description of the ARCHIVE ALL command that follows, the reader must have read and understood the material in the section "Commonly Used Command Operands" in Chapter 2.

ARCHIVE ALL	<pre> { ddname } { DSN(dsname) } TO ({ DSN(dsname(member)) }) { DSN('dsname') } { DSN('dsname(member)') }</pre> <p>[DATA(name)]</p> <p>[USER ({ (number) }) { (number1:number2) } { (:number2) }]</p> <p>[DATED ({ date } { date1:date2 } { date1: } { :date2 }]</p> <p>[NOTE('44 characters') OPERATOR]</p>
-------------	---

TO

This required operand specifies the external sequential data set or partitioned data set member receiving the archived documents and DERs.

ddname is the name on the DD, DLBL, or TLBL JCL statement that identifies the TO data set.

DSN... is valid in OS/VS2 only. It shows that the TO data set is dynamically allocated. If the specified data set name is in quotation marks, it is used as given. Otherwise it is qualified as described in Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics." ARCHIVE is the suffix qualifier. If TO(DSN...) in any of its forms is used, neither NOTE nor OPERATOR can be specified.

DATA(name) This operand specifies that only documents with this data type are archived. The default is all data types that meet other selection criteria.

USER Only an administrator can use this operand. It specifies which users' documents are to be archived. If the USER parameter is not specified by an administrator, then all users' documents that meet other selection criteria will be archived.

number is the library user number of a user. The documents of this user that meet other selection criteria will be archived.

number1:number2 specifies an inclusive range of library user numbers. The documents of these users that meet other selection criteria are archived.

:number2 specifies an inclusive range of library user numbers from the first in the document library through the specified number. The documents of these users that meet other selection criteria are archived.

DATED This operand specifies that only documents stored in the document library on or between the dates specified are archived. If DATED is not specified, all documents that meet other selection criteria are archived regardless of the dates they were stored in the document library. Each date specified must be in the form mm/dd/yy where mm is the month (1 to 12), dd the day of month (1 to 31), and yy the last two digits of the year.

date specifies that only documents stored in the document library on a particular date are archived.

date1:date2 specifies that only documents stored in the document library within an inclusive range of dates are archived.

date1: specifies that only documents stored in the document library on or after the specified date are archived.

:date2 specifies that only documents stored in the document library on or before the specified date are archived.

NOTE('44 characters') This operand specifies that up to 44 characters of information in a quoted string can be placed in the DERNOTE field of the DER of each archived document. Typically this field keeps track of the location of the archived document. This information remains available in the document library because a copy of the DER is retained in the directory data set.

OPERATOR This operand specifies that a message is written from the archive program to the console operator requesting the volume serial number of the volume containing the sequential data set in which the documents are archived. This number is retained in the DERNOTE field of the DER of each archived document. OPERATOR and NOTE are mutually exclusive.

Notes:

1. The input and output document library JCL statements are required.
2. If you use TO(ddname), you can also enter the NOTE or OPERATOR keyword. If you use TO(DSN...) in any of its forms, you cannot enter either NOTE or OPERATOR. TO(DSN...) in any of its forms dynamically allocates the data set, and DLF places the resulting data set location information in the DERNOTE field (which is also used by NOTE and OPERATOR) of the DER of each document archived. TO(DSN...) is valid under OS/VS2 only.
3. For VSE, use the ENVIRONMENT command to specify the characteristics of the external data set if the default values are not applicable to you. See Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics," for further details.
4. Locked documents can be archived.

Examples

- This example archives all selected documents to a sequential data set, ARCD5. Only documents with a data type of ATMS stored in the document library between January 2, 1979 and January 2, 1980, inclusive, are archived.

```
AUTH 123/ADMPASS  
ARCHIVE ALL DATA(ATMS) DATED(01/02/79:01/02/80) TO(ARCD5)
```

- This example archives all documents from the libraries of users 1 000 to 1 500 to a sequential data set, ARCD5. Because the USER parameter is specified, the authorized user must be an administrator.

```
AUTH 123/ADMPASS  
ARCHIVE ALL USER(1000:1500) TO(ARCD5)
```

COPY

There are four COPY commands with administrative functions:

- COPY IN
- COPY LIBRARY
- COPY OUT
- COPY USER.

All COPY commands create duplicate document(s) and DER's.

There are two ways to copy documents:

1. Between one user's library and another user's library either within the same document library or in another document library
2. Between a user's library and an external sequential data set.

To understand the description of the COPY commands that follow, the reader must have read and understood the material in the section "Commonly Used Command Operands" in Chapter 2.

Note: The COPY commands can be used to copy documents from one cluster to another cluster that has the same CI size. The EXPORT/IMPORT commands must be used to move documents between clusters of different CI sizes.

Preparing to Execute a Copy Command

1. **Copying from a document library:** When you are copying from a document library, the input document library JCL statements must refer to the document library from which the documents are copied. The output JCL statements can refer to:
 - a. An existing document library, into which documents and profiles are written with or without the REPLACE option. This can be the same as the input document library.
 - b. A newly defined and formatted document library.
 - c. A newly defined external sequential data set on either tape or disk.
2. **Copying into a document library:** When you are copying into a document library, the output JCL statements must refer to that document library. The input JCL statements can refer to:
 - a. The same document library.
 - b. Another document library.
 - c. An external sequential data set residing on either tape or disk. This data set was created by the COPY OUT command and contains profiles and documents that make up either all or part of a document library.
3. **Copying from one document library to another:** When copying documents from one document library to another, the current library user must be authorized in both document libraries and have the same password in both. When the command requires the user to be an administrator, the current user must be an administrator in both document libraries affected by the command.

To copy from one document library to another, the administrator can copy the document to an external sequential data set and then from the external sequential data set to the other document library. The administrator can also copy the document directly to the other document library using the JCL statements, which specify both an input and an output library.

Administrator Capabilities

1. An administrator can copy any user's profile and documents and all attribute profiles, except the SPR and CPRs. An administrator is the only one who can copy a complete document library.
2. An administrator can add to an existing document library by copying UPRs and their associated documents and attribute profiles from an external sequential data set or from another document library. If the users already exist in the library, he can add only the documents from these outside sources.
3. An administrator can specify a range of user libraries to copy from.

COPY IN

COPY IN restores all or part of a document library (that was copied to a backup data set with the COPY OUT command) to the same or a different document library. Documents from a backup data set can be copied in by users other than the user who created the data set.

A general user can restore only his own documents into a document library where he is defined by the same library user number as that on the backup data set. A general user cannot restore the documents of other users. However, he can use a complete document library as backup even though he is automatically limited by the commands he can use.

An administrator can restore the documents of all users. Only an administrator can use this command to copy processor profile records (PPRs), class entry records (CERs), and user mapping records (UMRs). An administrator can copy user profile records (UPRs) found on the backup data set but not found in the document library to which they are copied.

An administrator can copy document entry records (DERs) that do not have an associated UPR in the document library to which they are copied. The documents and their DERs are copied without being located in a user's library, in anticipation that a UPR will be defined. DERs existing without a UPR are known as ghost DERs because they cannot be accessed.

An error condition can arise if the UPRs are never copied in from the sequential data set. In this case, DERs will exist in the document library without any user defined for them. When this occurs, listing (LIST DOCUMENT command) the documents of the user with the next lower library user number picks up those DERs and it is apparent that the error exists. The administrator can make a recovery, if desired, by defining the library user number (DEFINE USER command) and purging the user number (PURGE USER command).

COPY IN	[{ ddname { DSN(dsname) FROM ({ DSN(dsname(member)) { DSN('dsname') { DSN('dsname(member)') }]]]]
	[USER ({ number { number1:number2 })]
	[SELECT ({ dname { dname1:dname2 { 'dname' { 'dname1':'dname2' })]
	[DATA ({ name { name1:name2 })]
	[CLUSTER(name)]
	[REPLACE NOREPLACE] [NOATTRIBUTE]

FROM

This operand identifies the data set or member of a partitioned data set containing the backup data created by the COPY OUT command.

The FROM keyword is required in VSE. If FROM is not provided in OS/VS2, the data set name Uuserid.BACKUP.COPY is used, where Uuserid is U followed by the user number of the current user. This data set must be previously defined.

ddname is the name on the DD, DLBL, or TLBL statement used for input.

DSN... is the name of the previously defined sequential data set used for input. COPY is the suffix qualifier if the dsname is not in quotation marks. This subparameter is valid for dynamic allocation in OS/VS2 only. See Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics," for details.

USER

Only an administrator can use the USER operand. It specifies that a library user number and all documents owned by that user are selected from the input data set and copied into the document library. If this operand is omitted, all library user numbers and their documents are copied.

number specifies the library user number of the user to be copied.

number1:number2 specifies that an inclusive range of users and their documents are to be copied.

SELECT

This operand names a document (or documents) to be copied. The default is to copy all documents in the user range that meet other selection criteria. A password cannot be used when specifying the name of the document being copied. Document names enclosed in quotes can contain any character and will not be folded to uppercase.

dname specifies the document name of the document to be copied.

dname1:dname2 specifies that a range of document names is to be copied.

DATA

This operand specifies the data type(s) to be copied. The default copies all data types.

name specifies a single data type to be copied.

name1:name2 specifies that an inclusive alphabetical range of data types is copied.

CLUSTER(name) This operand specifies the name of a cluster copied. Only documents originally copied out from a cluster with the same name will be selected for copying to this cluster name. Both clusters must have the same control interval size or the copy does not occur.

If you do not specify the CLUSTER operand, the default is all documents that meet the other selection criteria, regardless of the cluster in which they are stored.

- REPLACE** This operand specifies that the incoming document replaces an existing document in the library to which it is copied.
- NOREPLACE** This operand specifies that the incoming document does not replace an existing document. This operand is the default.
- NOATTRIBUTE** This operand is relevant to administrators only. It specifies that only UPRs and DERs are copied in, but no other directory data set records. The default is that all directory data set records (with the exception of the SPRs and CPRs) are copied in.

Notes:

1. The output document library JCL statements are required.
2. In order to save and restore a full cluster, use the COPY OUT and COPY IN commands specifying only the CLUSTER operand.

Examples

- This example copies in all the documents previously copied out. The documents are copied from the external sequential data set identified by ddname BACKUP. Any documents already in the document library are replaced. Because the authorized user is an administrator, any directory data set records (except the SPR and CPRs) on the backup data set are also copied in.

```
AUTH 123/ADMPASS
COPY IN FROM(BACKUP) REPLACE
```

- This example copies in documents previously copied out. The documents are copied from the sequential data set identified by the ddname BACKUP. Only documents whose names fall alphabetically between AZUSA and ZEBRA, inclusive, will be copied in.

```
AUTH 123/ADMPASS
COPY IN FROM(BACKUP) SELECT(AZUSA:ZEBRA)
```

COPY LIBRARY

This command copies a complete document library (directory data set and source clusters except for the SPR and CPRs) from one set of VSAM data sets to another. It expects a complete document library made up of one VSAM KSDS and at least as many ESDSs as necessary to make the new document library match the clusters in the existing document library. The document library receiving the copies must have exactly the same format as the document library sending the copies. That is, the new ESDSs must have identical CI sizes and DD names, identified through DEFINE CLUSTER with the same cluster names as the clusters of the existing ESDSs. DD statements for the input and output directory data sets and all input and output clusters must be present. If any of these requirements are not met, execution of the command terminates.

This exactness is necessary so there will be no references to nonexistent clusters in the new library. Because this command is used for creating a new document library or a backup document library, no other workable alternative is available. Other forms of copying partial libraries, however, are available to users through COPY IN, COPY OUT, and COPY USER.

The output library may already contain data; whether it is to be replaced or not is determined by the REPLACE/NOREPLACE operands. If neither operand is given, NOREPLACE is the default. If a duplicate user record is encountered and the NOREPLACE operand is in effect, a message prints indicating that the user and all his documents will not be copied. If a duplicate DER record is encountered, it is a ghost DER (no corresponding UPR); a message prints and the document is not copied. This error occurs only when the system has halted in mid-process.

If a duplicate document is encountered in the new document library (that is, the library user number, document name, data type, and version are identical), the document is not replaced and a message is written unless the REPLACE operand is in effect.

Note that the SPR record and each cluster's record 0 will always remain intact because their time stamps cannot be altered. All clusters must be formatted prior to the COPY process.

COPY LIBRARY	<div>REPLACE NOREPLACE</div>
--------------	----------------------------------

REPLACE This operand specifies that duplicate documents are replaced in the document library that is receiving documents.

NOREPLACE This operand specifies that duplicate documents are not replaced in the document library being copied to. This is the default.

Notes:

1. The input and output document library JCL statements are required.
2. The receiving document library must have at least as much space as the sending document library.

Example

This example copies the entire input document library to another document library. Any equivalent documents in the output document library are replaced.

```
AUTH 123/ADMPASS  
COPY LIBRARY REPLACE
```

COPY OUT

This command provides a backup for all or part of a document library. An administrator can copy the documents and UPRs of a specific user, a range of users, or all users. An administrator can also copy directory data set records with the exception of the SPR and CPRs. The SPR and CPRs can never be copied out because COPY commands can be performed only on formatted document libraries. General users can copy any document they own (that is, in their own library, their assigned project library, or any public library).

The data set generated by the COPY OUT command is described in Appendix D.

Note: In VSE, if you have multiple COPY commands in a job step, you may want to put multiple files on a single tape (because you can specify only one tape unit in a step). Use NOREWIND to override the default tape options, REWIND and UNLOAD, in the ENVIRONMENT command. Then each COPY command will create a separate file on the tape in the order you process them.

COPY OUT	<div data-bbox="629 732 1186 880"> <pre> [{ ddname } { DSN(dsname) } TO ({ DSN(dsname(member)) }) { DSN('dsname') } { DSN('dsname(member)') }] </pre> </div> <div data-bbox="629 902 1186 976"> <pre> [USER ({ number }) { number1:number2 }] </pre> </div> <div data-bbox="629 998 1186 1125"> <pre> [SELECT ({ dname } { dname1:dname2 }) { 'dname' } { 'dname1':'dname2' }] </pre> </div> <div data-bbox="629 1146 1125 1221"> <pre> [DATA ({ name }) { name1:name2 }] </pre> </div> <div data-bbox="629 1242 857 1285"> <pre> [NOATTRIBUTE] </pre> </div> <div data-bbox="629 1306 890 1359"> <pre> [CLUSTER(name)] </pre> </div>
----------	--

TO This operand identifies the sequential data set or member of a partitioned data set receiving the data. The TO operand is required in VSE.

If TO is not provided in OS/VS2, the data set name Usernumber.BACKUP.COPY is used, where Usernumber is U followed by the user number of the current user. This data set must have been defined previously.

ddname is the name on the DD, DLBL, or TLBL statement to receive the data.

DSN... is the name of the previously defined sequential data set to receive the data. COPY is the suffix qualifier, if dsname is not in quotation marks. See Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics," for details. This subparameter is valid for dynamic allocation in OS/VS2 only.

USER

Only an administrator can use this operand. It specifies a library user whose documents are copied. All documents owned by each specified user in that user's own library, that user's project library and any public libraries are copied whether locked or unlocked. If this keyword is omitted by an administrator, all documents of all users in the document library are copied.

number specifies a single user whose documents are copied.

number1:number2 specifies an inclusive range of users whose documents are to be copied.

SELECT

This operand names a document to be copied. All versions are copied. If this keyword is omitted, all documents that meet other selection criteria are copied, including those locked and unlocked and the DERs for any archived documents. Only the document name can be used; neither the library number nor the password can be entered. Document names enclosed in quotes can contain any character and will not be folded to uppercase.

dname specifies the name of a single document copied.

dname1:dname2 specifies an inclusive alphabetical range of document names to be copied. All versions are copied, including those owned by the user in project or public libraries and any that are locked or archived.

DATA

This operand specifies a particular data type to be copied. The default is all data types.

name specifies that a particular data type be copied.

name1:name2 specifies that an inclusive, alphabetic range of data types be copied. The default is all data types.

NOATTRIBUTE

This operand is relevant to administrators only. It specifies that none of the directory data set records (PPRs, CERs, UMRs) is written to the backup data set. If an administrator does not specify this keyword, all directory data set records are copied (with the exception of the SPR and the CPRs).

CLUSTER(name)

This operand specifies the name of a cluster from which the copy comes. Only documents in this cluster are copied. To save and restore an entire cluster, use only this parameter. When you later issue the COPY IN command using this backup data set, both clusters must have the same control interval size or the subsequent COPY IN does not occur.

If you do not specify the CLUSTER operand, the default is all documents that meet the other selection criteria, regardless of the cluster in which they are stored.

Notes:

1. The input document library JCL statements are required.
2. All versions of each document are always written to the backup data set.

Examples

- This example creates a backup copy of the directory records (UPRs, PPRs, CERS, UMRs) and selected documents for all users in the document library. The documents are copied to the sequential data set identified by ddname BACKUP. Only documents with names that fall in the inclusive alphabetic range of ESSENTIAL through NONESSENTIAL are copied. These documents can be copied back into the document library with the COPY IN command.

```
AUTH 123/ADMPASS  
COPY OUT TO(BACKUP) SELECT(ESSENTIAL:NONESSENTIAL)
```

- This example creates a complete backup copy of the entire library. All profiles (except SPR and CPRs) and documents are backed up.

```
AUTH 123/ADMPASS  
COPY OUT TO(BACKUP)
```

- This example creates a backup copy of all the documents belonging to users in the inclusive numeric range of 123 through 456. The documents are copied to the external sequential data set identified by ddname BACKOUT. Only an administrator can specify a user range. These documents can be copied back into the document library with the COPY IN command.

```
AUTH 123/ADMPASS  
COPY OUT TO(BACKOUT) USER(123:456)
```

COPY USER

This command copies all the documents from one user's library to another user's library in the same or a different document library. For a project or public library, all documents in the library identified by the FROMUSER keyword are copied including those documents whose owner is not the owner of the library. For a private library, the COPY USER command copies only those documents in the private library identified by the FROMUSER keyword. Any documents owned by that user, but residing in a project or public library identified by another user number, are not copied. Archived documents are not copied. The password on any copied document is nullified in the copy.

Document copies will be owned by the user specified in the TOUSER(number) operand. If the user being copied to does not exist, a new UPR is created using the FROMUSER UPR as a model. Any password is removed from the new UPR.

All the copies are made in the same clusters as the original documents if the document libraries are the same. If the document libraries are different, the output library must have the same cluster names defined as the input library, and the CI sizes must be the same for the corresponding names. If the cluster names and sizes in the output library are not the same as those in the input library, any documents in clusters that do not match will not be copied, and a message will be issued for each affected document.

If you are copying within the same document library, the FROMUSER and TOUSER numbers must be different.

Only the administrator can use this command.

COPY USER	FROMUSER(number) TOUSER(number)
-----------	---------------------------------

FROMUSER(number) This operand specifies the library sending the copies. A zero in the (number) field is invalid.

TOUSER(number) This operand specifies the library receiving the copies. This library owns all the new copies. A zero in the (number) field is invalid.

Notes:

1. The input and output document library JCL statements are required.
2. All keywords are required.
3. There are no defaults.

Example

This example copies all of the documents in user 101's library to user 102's library.

```
AUTH 123/ADMPASS
COPY USER FROMUSER(101) TOUSER(102)
```

DEFINE

An administrator can execute six DEFINE commands:

- DEFINE CLASS
- DEFINE CLUSTER
- DEFINE MAP
- DEFINE PROCESS
- DEFINE SYSTEM
- DEFINE USER.

Only the administrator can use the define commands. The commands create new profile records in the directory data set.

To understand the description of the define commands that follow, the reader must have read and understood the material in the section "Commonly Used Command Operands" in Chapter 2.

DEFINE CLASS

This command creates a class entry record (CER), which associates a class of documents with a document library cluster.

DEFINE CLASS	name CLUSTER(name)
--------------	--------------------

name This operand specifies the name by which the class is known, such as letter, book, or memo.

CLUSTER(name) This operand specifies the name of the cluster into which a document with this class is to be stored if no specific cluster is specified when the document is imported. The cluster name must have been previously defined with a **DEFINE CLUSTER** command.

Notes:

1. The output document library JCL statements are required.
2. If you want to define a class name for the 4K default cluster, use **CLUSTER(DSMPTLIB)**.
3. Parameters must be specified in the order listed above.

Example

This example defines a class called **LETTER** and associates it with the cluster **SMALL**. **SMALL** becomes the default cluster in which documents imported without a specific cluster designation and with a class of **LETTER** are stored.

```
AUTH 123/ADMPASS
DEFINE CLASS LETTER CLUSTER(SMALL)
```

DEFINE CLUSTER

This command creates a new cluster profile record (CPR), which defines a VSAM ESDS cluster to the document library. The control intervals of the defined cluster are formatted for use as a source cluster in the document library. This command also reconnects purged clusters to the document library.

DEFINE CLUSTER	name INDD(ddname) OUTDD(ddname) [PASSWORD ({ VSAM-password }) ['VSAM-password']] [FORCE NOFORCE]
-----------------------	--

- name** This operand specifies the name by which the cluster will be known.
- INDD(ddname)** This operand specifies the DD (OS/VS2) or DLBL (VSE) statement that must subsequently be included in the JCL to define this cluster to DLF for input.
- OUTDD(ddname)** This operand specifies the DD (OS/VS2) or DLBL (VSE) statement that must be included in the JCL to define this cluster to DLF for output.
- PASSWORD(VSAM-PASSWORD)** This operand specifies the password in the AMS DEFINE or ALTER statement for the specified cluster. This password is not used by DLF and need not be specified unless the document library is to be accessed directly by a non-Documents Library Facility program which requires the password for authorization or OPEN processing or both. VSAM-passwords are 1 to 8 characters long. If they contain any special characters they must be enclosed in single quotation marks.
- If you do not specify the PASSWORD operand, DLF will not change the VSAM-password value in the CPR for the cluster being modified.
- FORCE** This operand specifies that if Document Entry Records (DERs) associated with the new cluster name already exist in the DLF output directory and you are not reconnecting a cluster that you disconnected with the PURGE CLUSTER command, then the DERs will be erased, the cluster will be formatted, and the CPR written to the output library.
- NOFORCE** This operand specifies that if DERs associated with the new cluster name already exist, the DEFINE CLUSTER command will terminate with an error message and the cluster will not be defined. If no such DERs exist, the cluster will be formatted and the CPR written to the DLF output directory. This is the default.

Notes:

1. The output document library JCL statements are required.
2. The control-interval size of the cluster is determined by DLF through a SHOWCB macro and is retained in the CPR.

3. The CPR for the 4K default cluster is defined by the DEFINE SYSTEM command, as if the following command were issued:

For OS/VS2 -

```
DEFINE CLUSTER DSMPTLIB INDD(DSMINLIB) OUTDD(DSMPTLIB)
```

For VSE -

```
DEFINE CLUSTER DSMPTLIB INDD(DSMILIB) OUTDD(DSMPLIB)
```

4. If document entry records (DERs) already exist in the DLF output directory for this cluster and you are not reconnecting a cluster that you disconnected with the PURGE CLUSTER command, then the cluster is not defined unless the FORCE operand is given.

For example, if you are reorganizing a single ESDS source cluster (see "Reorganizing an ESDS Source Cluster on page 211) and you did not remove the DERs associated with the cluster name before deleting and redefining the cluster with VSAM, the DEFINE CLUSTER command will detect DERs left in the DLF output directory. To delete these DERs and define a cluster with the given name, use the FORCE operand on the DEFINE CLUSTER command.

For more information on removing the DERs before defining a cluster, see the DOCUMENT operand of the PURGE CLUSTER command or the PURGE DOCUMENT command.

Example

This example creates a CPR for the cluster SMALL and associates it with the VSAM data sets identified by the DD or DLBI names DSMILB1 (when used as input) and DSMPLB1 (when used as output). If this cluster has not been formatted previously, this command formats it for use as a source cluster in the document library.

```
AUTH 123/ADMPASS
DEFINE CLUSTER SMALL INDD(DSMILB1) OUTDD(DSMPLB1)
```

DEFINE MAP

This command creates a user mapping record (UMR), which associates an external user with a library user number.

DEFINE MAP	HOST('identifier') ID('identifier') USER(number)
------------	--

HOST('identifier') This operand specifies the external host identifier. Either 1 or 2 characters can be used in the identifier field, and the field must be entered as a quoted string.

ID('identifier') This operand specifies how the external user is identified on the external host system. From 1 through 10 characters are used in the identifier field, and the field must be entered as a quoted string.

USER(number) This operand specifies the library user number to which this external user is mapped.

Notes:

1. The output document library JCL statements are required.
2. All keywords are required.
3. There are no defaults.
4. Multiple external users can be mapped to a single library user number.

Examples

- This example maps ATMS user 36 to library user number 20 000.
AUTH 123/ADMPASS
DEFINE MAP HOST('AT') ID('36') USER(20000)
- This example maps external user IDFROM on host system HI to library user number 10 000.
AUTH 123/ADMPASS
DEFINE MAP HOST('HI') ID('IDFROM') USER(10000)

DEFINE PROCESS

This command creates a processor profile record (PPR), which associates a processor with a pair of data type names. A processor can be automatically invoked when the DLF commands IMPORT, EXPORT, READ, and SCRIPT are executed. These commands specify or imply that the document acted on has a current data type name identical to the first of the pair of PPR data type names. After the action is complete, it will have a data type name identical to the second of the pair of PPR data type names. Such a processor commonly converts formatting controls contained in a document from those recognized by one formatter (such as ATMS formatter) to those recognized by another formatter (such as SCRIPT formatter). The type of formatting controls that are contained in a document usually are indicated by its data type. For details about processors, see Chapter 8, "Using DLF to Call Processors."

DEFINE PROCESS	INDATA(name) OUTDATA(name) [ENTRY(name)]
----------------	---

INDATA(name) This operand specifies the first of a pair of data type names in the PPR. It indicates the input data type of documents to be handled by the processor.

OUTDATA(name) This operand specifies the second of a pair of data type names in the PPR. It indicates the output data type of the documents handled by the processor.

ENTRY(name) This operand specifies the entry point name of the module to do the special processing for this combination of data type names. If this operand is omitted or if a zero is specified for name, the processor field in the PPR is null and no special processing is implied for commands specifying this combination of data type names. The name must be 1 to 8 characters in length, which DLF will fold to uppercase. It may contain any alphameric characters and the characters @, #, and \$. The first character must not be numeric.

Notes:

1. The output document library JCL statements are required.
2. The following table shows the correspondence of INDATA and OUTDATA as specified in the DEFINE PROCESS or ALTER PROCESS command with data type pairs specified on other DLF commands.

Command	Input Data Type	Output Data Type
DEFINE PROCESS	INDATA	OUTDATA
ALTER PROCESS	INDATA	OUTDATA
IMPORT	INDATA	DATA
EXPORT	DATA	OUTDATA
READ	DATA	OUTDATA
SCRIPT	DATA	implied(SCRIPT)

Example

This example defines the entry name of a module to be called if a DLF command is issued with the input data type of ATMS and the output data type of SCRIPT.

```
AUTH 123/ADMPASS  
DEFINE PROCESS INDATA(ATMS) OUTDATA(SCRIPT) ENTRY(DSMACAIP)
```

DEFINE SYSTEM

This command creates the system profile record (SPR) and establishes the document library by formatting the space for the default 4K cluster. It can also be used to specify certain global values for the Document Library Facility.

DEFINE SYSTEM	[LIBRARY(name)]
	[RACFNAME(name)]
	[ROUTE(number)]

LIBRARY(name)

This operand specifies a name to be used as the RNAME in the ENQ macro (OS/VS2) or as the LOCK name (VSE). (name) can be 1 to 12 alphanumeric characters but cannot begin with a numeric character. If this keyword is omitted, no LIBRARY enqueue name is associated with this document library.

CAUTION:

LIBRARY(name) must be in effect for certain invocations of the document library. See the caution in "JCL Statements" on page 13 for details.

RACFNAME(name)

This operand specifies the RACF security code name (RACF resource class name) used for RACF protection of library user numbers. (name) can be 4 to 8 alphanumeric characters or the characters \$, #, and @. The first character must be alphabetic or one of the characters \$, #, and @. If not specified, there is no RACF protection for library user numbers. This does not affect the user protection provided by DLF. This operand is valid only for OS/VS2.

ROUTE(number)

This operand specifies the number of the OS operator route code as defined in the write-to-operator (WTO) macro. Messages and replies resulting from the OPERATOR keyword of the ARCHIVE and RETRIEVE commands are sent to and received from the OS operator. The number specified must be in the range of 1 to 16. If not specified, 3 is used. See the applicable OS/VS2 Supervisor Services and Macro Instructions manual for information on the WTO macro. This operand is not valid in VSE.

Notes:

1. The DEFINE SYSTEM command is required only once for initialization of a document library. Only one SPR exists for each document library.
2. The output document library JCL statements must specify the new VSAM data sets, that is, the directory data set (KSDS) and the basic 4K cluster for the source data set (ESDS). Any other ESDS clusters must be defined with the DEFINE CLUSTER command.

3. The input document library JCL statements must not be included if they refer to these new VSAM data sets. If they are included, DLF initialization routines will not find an SPR in the directory data set (because a DEFINE SYSTEM command will not yet have been issued) and the job step ends.
4. The DEFINE SYSTEM command includes an implicit DEFINE CLUSTER for the default 4K cluster, that is:
 For OS/VS2 -

```
DEFINE CLUSTER DSMPTLIB INDD(DSMINLIB) OUTDD(DSMPTLIB)
```

 For VSE -

```
DEFINE CLUSTER DSMPTLIB INDD(DSMILIB) OUTDD(DSMPLIB)
```
5. The DEFINE SYSTEM command must be the first DLF command in the job step that establishes a new document library. This is the only case where the first DLF command is not an AUTH command.
6. The command following the DEFINE SYSTEM command in the job stream must be a DEFINE USER command that defines a user with administrator authority.
7. Because DLF formats the complete source data set, the normal VSAM message resulting from this command is:

```
IEC0701 104-203, jobname, stepname, DSMPTLIB,,,dsn,cat
```
8. The ALTER SYSTEM command can be used to format new space added to the basic 4K cluster or to change any of the system parameters defined in the DEFINE SYSTEM command.

Example

This example defines an SPR, formats the basic 4K cluster, and defines a user with administrator authority.

```
DEFINE SYSTEM
DEFINE USER 12104 PASSWORD (/SEAGULL) ADM
```


DEFINE USER

This command creates a user profile record (UPR), which defines a user and the user's library to the document library.

DEFINE USER	<p>number</p> <p>[MODEL(number)]</p> <p>[PASSWORD({/password }) {'password'} NOPASS]</p> <p>[SPACE ({ number }) { 512 }]</p> <p>[CLASS(name)]</p> <p>[DATA(name)]</p> <p>[SOURCE(name)]</p> <p>[LIBTYPE ({ PRIVATE } { PROJECT }) { PUBLIC }]</p> <p>[PROJLIB ({ number }) { NONE }]</p> <p>[ADM NOADM]</p> <p>[CONTROL NOCONTROL]</p>
-------------	--

number

This operand specifies the number of a new user and his library.

MODEL(number)

This operand specifies the library user number whose UPR is used as a model for creating this UPR. Any values other than PASSWORD, which are not explicitly specified in this command, are taken from this model UPR. If not specified, no UPR model is used.

PASSWORD
or
NOPASS

This operand specifies the password associated with this user. If specified, it must be given with each subsequent AUTH command for this user. The characteristics of the user password are the same as those of a document password except that it is associated with a user instead of a document. If PASSWORD is not specified, no password is associated with this user. NOPASS specifies that this user will not have a password. This is the default and therefore need not be specified.

SPACE(number)

This operand specifies the maximum amount of document library direct access space that can be assigned to this library user number. The number field specification can be 0 to 4 000 000, specified in 1 000 byte units. Thus, the largest allowable number of bytes is 4 000 000 000. If SPACE(0) is specified, the user cannot store documents in the document library. If neither SPACE nor MODEL is specified, the default for the number field is 512 (512 000 bytes).

CLASS(name)

This operand specifies the user's default class. This class is assumed for this user if he does not specify CLASS where allowed on subsequent commands.

A null class is assigned if:

- This operand and the MODEL operand are omitted.
- 0 (zero) is specified in the name field.
- The MODEL operand is specified and the user has a null class name.

DATA(name)

This operand specifies the user's default data type. This data type is assumed for this user if he does not specify DATA where allowed on subsequent commands.

A null data type is assigned if:

- This operand and the MODEL operand are omitted.
- 0 (zero) is specified in the name field.
- The MODEL operand is specified and the user has a null data type name.

SOURCE(name)

This operand specifies the default source attribute associated with this user's documents if a specific source is not specified on commands that allow it.

A null source is assigned if:

- This operand and the MODEL operand are omitted.
- 0 (zero) is specified in the name field.
- The MODEL operand is specified and the user has a null source name.

LIBTYPE

This operand identifies the library type owned by this user.

PRIVATE specifies that the library is to be private. This is the default if neither MODEL nor LIBTYPE is specified.

PROJECT specifies that the library is to be a project library.

PUBLIC specifies that the library is to be a public library.

PROJLIB	This operand identifies the project library assigned to a user. This library must be currently defined in the document library. The number field identifies the library user number of the project number. NONE specifies that this user does not have access to any project library. If neither PROJLIB nor MODEL is specified, NONE is the default.
ADM	This operand specifies that this user is an administrator.
NOADM	This operand specifies that this user is not an administrator. This is the default.
CONTROL	This operand specifies that if LIBTYPE is PROJECT or PUBLIC, it is controlled, meaning no user other than the library owner can store or delete documents in it. Users who are otherwise authorized, however, can access documents stored in this library. This operand is invalid for private libraries.
NOCONTROL	This operand specifies that if LIBTYPE is PROJECT or PUBLIC, it is uncontrolled. This is the default. This operand is invalid for private libraries.

Note: The output document library JCL statements are required.

Examples

- This example defines user 200, with all defaults taken. They include NOPASS, SPACE(512), LIBTYPE(PRIVATE), PROJLIB(NONE), NOADM, and NOCONTROL.

```

AUTH 123/ADMPASS
DEFINE USER 200

```
- This example defines user 300 with the password SECRET. This library is a controlled project library with a maximum allocation of 100 000 bytes.

```

AUTH 123/ADMPASS
DEFINE USER 300 PASSWORD(/SECRET) -
LIBTYPE(PROJECT) SPACE(100) CONTROL

```
- This example defines user 101 with information from the profile of user 10. The PROJLIB parameter of user 10 is overridden. This user has access to project library 200.

```

AUTH 123/ADMPASS
DEFINE USER 101 MODEL(10) PROJLIB(200)

```

LIST

This command lists information from various profiles in the directory data set.

These are the LIST commands with administrative functions:

- LIST DOCUMENT
- LIST MAP
- LIST SYSTEM
- LIST USER.

Note: See Appendix F, "Output of LIST Commands," for an example of output from LIST DOCUMENT and LIST USER.

To understand the description of the LIST commands that follow, the reader must have read and understood the material in the section "Commonly Used Command Parameters" in Chapter 2.

A list becomes more specific as the number of operands qualifying it increases.

LIST DOCUMENT

This command lists information from the document entry records (DERs) for the documents assigned to either the current library user or to the user(s) specified in the USER keyword.

This command can be used by any library user, but only an administrator can use the USER operand of this command to specify multiple library users whose documents are to be listed.

LIST DOCUMENT	<pre> FROMLIB ({ INPUT }) { OUTPUT } TO ({ ddname } { DSN(dsname) } { DSN(dsname(member)) } { DSN('dsname') } { DSN('dsname(member)') } USER ({ number } { number1:number2 } SELECT ({ dname } { dname1:dname2 } { 'dname' } { 'dname1':'dname2' } DATED ({ date } { date1: } { date1:date2 } { :date2 } CLASS(name) CLUSTER(name) DATA(name) SOURCE(name) ALL OWN SPACE NOSPACE LOCK ARCHIVE VERSION INFO </pre>
---------------	--

FROMLIB Use this operand to specify which document library's document entry records you want to list. Your options are:

INPUT Indicates that you want to list all of the DERs (that also meet the other selection criteria) in the input document library. This is the default.

OUTPUT Indicates that you want to list all of the DERs (that also meet the other selection criteria) in the output document library.

TO This operand identifies the external sequential data set or member of a partitioned data set receiving the data. If TO is omitted, the data is written to a sequential data set with the ddname of DSMLIST (for OS/VS2) or to SYSLST (for VSE).

ddname is the name on the DD, DLBL, or TLBL JCL statement that describes the data set to which the list data is being sent.

DSN... is valid in OS/VS2 only. It shows that the list data set is dynamically allocated. If the specified data set name is in quotation marks, it is used as given. Otherwise it is qualified as described in Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics." LIST is the suffix qualifier.

USER This operand specifies the user(s) whose libraries are searched for DERs meeting the other selection criteria for the list. The default is to search the current user's private library, project library, and any public library in which he owns documents.

number specifies a particular user's library to be searched for DERs meeting the other selection criteria for the list. Administrators can specify any library. General users can specify their own library, their project library, or any public library. If zero (0) is specified for number, the current user's project library is searched.

number1:number2 specifies a range of users' libraries to be searched for DERs meeting other selection criteria for the list. Only an administrator can use this form of the USER operand to list documents belonging to multiple library users. General users can use this operand to list the documents in their project library or in any public library.

SELECT This operand specifies that selection is by document name. If not specified, all documents meeting the other selection criteria are listed. Document names enclosed in quotes can contain any character and will not be folded to uppercase.

dname specifies that only the document(s) with this document name are to be selected for the list.

dname1:dname2 specifies that documents whose names fall in the inclusive range of the names specified are to be selected for the list.

DATED This operand specifies that only documents stored in the document library on a specific date or between an inclusive range of dates are to be selected for the list. The default is documents stored on any or all dates that meet the other selection criteria. Specify the date in mm/dd/yy format, where mm is the month (1-12), dd is the day of the month (1-31), and yy is the last two digits of the year.

date specifies the given date.

date1: specifies the given date and all subsequent dates.

date1:date2 specifies an inclusive range of dates.

:date2 specifies the given date and all preceding dates.

CLASS(name) This operand specifies that only documents with the given class are to be listed. The default is all classes.

CLUSTER(name) This operand specifies that only documents in the given cluster are to be listed. The default is all clusters.

DATA(name) This operand specifies that only documents with the given data type are listed. The default is all data types.

SOURCE(name) This operand specifies that only documents with the given source attributes are to be listed. The default is all source attributes.

ALL This operand indicates that all documents in the designated libraries that meet the other selected criteria are to be listed.

OWN This operand specifies that only documents owned by the current user in the designated user libraries are to be listed. This is the default.

SPACE This operand specifies that you want to generate the Space Usage report rather than the Standard report you would receive by default. If you specify SPACE, you can specify any of the current LIST DOCUMENT options, but the INFO option would be ignored. If you specify both the ARCHIVE option and the SPACE option, the archived documents are listed and show the amount of physical space that would be used if the documents were retrieved; but the summary page does not include information about documents that are archived. The summary page is not generated unless more than one NON-ARCHIVED document is listed.

NOSPACE Use this operand to indicate that you want the Standard report for documents. This is the default.

LOCK This operand indicates that locked documents are to be listed as well as unlocked ones. The default is to list only unlocked documents.

ARCHIVE This operand specifies that the following information about archived documents is to be listed: the date of their archiving, the time of day, and the contents of the DERNOTE field. The default is to list only documents that are not archived. If you also specify the SPACE option, archived documents are listed but not included in the summary statistics.

VERSION This operand specifies that all versions of the document(s) are to be listed. The default is to list only the latest version of the document(s).

INFO This operand specifies that the listing includes the information from the DERINFO field (specified as DOCINFO on the IMPORT command), if such information exists. The default is not to list the information in the DERINFO field. This operand is ignored if you specify the SPACE option.

Notes:

1. Input document library JCL statements are required:
 - If you specify FROMLIB(INPUT)
 - If you omit the optional FROMLIB operand (allowing the command to default to INPUT).
2. Output document library JCL statements are required if you specify FROMLIB(OUTPUT).
3. An administrator can list information from the DERs of any document in any user's library.
4. General users can list information from the DERs of the documents they own (that is, all documents in their private libraries, all documents in their project libraries, and all documents in any public library in which they own documents).
5. Specifying this command without any operands, lists information from the DERs of the latest (or only) version of all documents that are owned by the current library user and that are not locked or archived. This is true for documents in the user's own library, his project library, and any public libraries in which he owns documents.
6. There are two types of LIST DOCUMENT reports:
 - The Standard report - generated by default or by specifying NOSPACE on the LIST DOCUMENT command.
 - The Space Usage report - generated by specifying SPACE on the LIST DOCUMENT command.

Both reports display space in bytes and include: library number, document name, class and data type, version number, owner ID, and the name of the cluster in which the document is stored. Each report has a summary line indicating the number of documents listed.

The Standard report also includes the document source, control and archive indicators, creation date, file size, and number of records.

The additional information available in the Space Usage report shows the control interval SIZE of the cluster, the COUNT of control intervals a document is using, and the percentages with the number of bytes that are FILLED and VACANT. If there is more than one NON-ARCHIVED document listed, a summary page is also generated. It provides a count and other summary information about ONLY the NON-ARCHIVED documents listed. This includes the SMALLEST, AVERAGE, and LARGEST document sizes and totals for the number of CTL-INTRVLs and RECORDS. The summary page also lists a total number of TEXT BYTES that excludes the bytes used by DLF for control purposes. This is essentially a total for the field called FILE SIZE on a Standard LIST DOCUMENT report (excluding any archived documents). The last lines on the summary page are totals and overall percentages for the FILLED, VACANT, and TOTAL fields on the Space Usage report.

7. See Appendix F, "Output of LIST Commands" on page 253 for examples of the two reports and Chapter 7, "Administering the Document Library" on page 205 for a more detailed explanation of the space usage statistics from the LIST DOCUMENT and LIST CLUSTER commands.
8. Alphabetical order for listing is blank, \$, #, @, a-z, A-Z, and 0-9.
9. The listing can show unexpected documents from a library that is neither the user's private library, his project library, nor any public library. These are ghost DERs. They appear if the UPR corresponding to the DER is never copied by the COPY IN command. Ghost DERs are picked up when listing the documents of the user with the next lowest user number. They can be recovered by defining the user (DEFINE USER command) whose UPR was missing. As an administrator, you can access the problem documents to purge (PURGE DOCUMENT command) or use them.

Example

- This example lists the Standard report information for the DERs of the latest (or only) version of all documents in libraries 1 through 9 999 999 of the input document library which are not locked or archived. The current user has administrator authority.

```
AUTH 123/ADMPASS
LIST DOCUMENT USER (1:9999999) ALL
```

- This example requests the Space Usage report for LIST DOCUMENT and specifies that only documents in the cluster named DSMPTLIB are to be listed. Documents are listed from the input document library.

```
AUTH 123/ADMPASS
LIST DOCUMENT CLUSTER(DSMPTLIB) SPACE
```

LIST MAP

This command lists information from all user mapping records (UMRs) defined to the document library.

LIST MAP	[FROMLIB ({ INPUT }) { OUTPUT }]
----------	---

FROMLIB Use this operand to specify which document library's user mapping records you want to list. Your options are:

INPUT Indicates that you want to list all of the UMRs in the input document library. This is the default.

OUTPUT Indicates that you want to list all of the UMRs in the output document library.

Notes:

1. Input document library JCL statements are required:
 - If you specify FROMLIB(INPUT).
 - If you omit the optional FROMLIB operand (allowing the command to default to INPUT).
2. Output document library JCL statements are required if you specify FROMLIB(OUTPUT).

Example

This example lists information from all UMRs defined in the input document library.

```
AUTH 123/ADMPASS
LIST MAP
```

LIST SYSTEM

This command lists information from the system profile record (SPR). Only an administrator can use this command.

LIST SYSTEM	[FROMLIB ({ INPUT }) { OUTPUT }]
-------------	---

FROMLIB Use this operand to specify which document library's system profile record (SPR) you want to list. Your options are:

INPUT Indicates that you want to list the SPR of the input document library. This is the default.

OUTPUT Indicates that you want to list the SPR of the output document library.

Notes:

1. Input document library JCL statements are required:
 - If you specify FROMLIB(INPUT).
 - If you omit the optional FROMLIB operand (allowing the command to default to INPUT).
2. Output document library JCL statements are required if you specify FROMLIB(OUTPUT).
3. The OS/VS2 listing includes these SPR values:
 - System creation date
 - Library name
 - RACF name
 - OS/VS2 operator routing code.
4. The VSE listing shows SPR values for system creation date and library name.

Example

This example lists information from the output document library SPR:

```
AUTH 123/ADMPASS
LIST SYSTEM FROMLIB(OUTPUT)
```

LIST USER

This command lists information from the indicated user profile records (UPRs). Administrators can use this command to list information from the UPRs of any or all users defined to the document library. A general user can list information only from the UPRs of his own library, his project library, or any public library.

LIST USER	<div data-bbox="645 421 1037 549">{ number } { number1:number2 } { ALL }</div> <div data-bbox="645 570 1053 644">FROMLIB ({ INPUT }) { OUTPUT }</div>
-----------	---

If you do not specify either number or number1:number2, DLF defaults to the current user's own UPR.

number	Specifies a particular library user.
number1:number2	Specifies an inclusive range of library users. If an administrator specifies a range of user numbers, information is listed from all the UPRs within the inclusive range. If a general user specifies a range of user numbers, information will be listed from the UPRs within this range that define only the users' libraries he is authorized to access. These include the UPRs of his own library, his project library, and any public library.
ALL	Indicates that you want to list information about all DLF users.
FROMLIB	Use this operand to specify which document library's user profile records you want to list. Your options are: INPUT Indicates that you want to list the UPR(s) (that also meet other selection criteria) in the input document library. This is the default. OUTPUT Indicates that you want to list the UPR(s) (that also meet other selection criteria) in the output document library.

Notes:

1. Input document library JCL statements are required:
 - If you specify FROMLIB(INPUT).
 - If you omit the optional FROMLIB operand (allowing the command to default to INPUT).
2. Output document library JCL statements are required if you specify FROMLIB(OUTPUT).
3. A zero (0) specified as the only user number, refers to the current user's project library. Zero (0) specified as the first number of a range causes DLF to assume you want to search from the beginning (user #1) of the Document Library.

4. The LIST USER command lists space in units of 1 000 bytes. "SPACE IN USE" and "PUB/PRJ SPACE" are rounded upward to the nearest thousand bytes.
5. Refer to Appendix F, "Output of LIST Commands" on page 253 for an example of the output from the LIST USER command.

Example

This example lists the UPRs of users 100 to 200, inclusive, from the input document library.

```
AUTH 123/ADMPASS
LIST USER 100:200
```

PROTECT

This command enables an administrator to change the access status of any document in the document library. It also enables general users to change the accessibility of any documents they own and any documents within their own libraries (regardless of document ownership).

To understand the description of the PROTECT command that follows, the reader must have read and understood the material in the section "Commonly Used Command Operands" in Chapter 2.

PROTECT	docid [DATA(name)] [VERSION(number)] [PASSWORD({/password }) NOPASS {/'password'}] [SHARE NOSHARE]
----------------	--

docid	This operand identifies the document whose access status is changed. An administrator does not have to specify a document's password.
DATA(name)	This operand specifies the document's data type. The default is the data type in the current library user's UPR.
VERSION(number)	This operand indicates the version of the document whose access status is changed. The default is the latest version. If the version is not found, the document is not altered, and a warning message is issued.
PASSWORD or NOPASS	The /password or /'password' operand specifies a new password for the document. The default is to leave the password unchanged. NOPASS indicates that any existing document password is deleted.
SHARE	This operand specifies that the document is shareable. The default is the current share status of the document. Documents with share status can be accessed by any valid user of the document library if the passwords are known. If you omit the SHARE/NOSHARE operand pair from the PROTECT command, no change is made to the document's share status.
NOSHARE	This operand specifies that the document no longer has share status. If you omit the SHARE/NOSHARE operand pair from the PROTECT command, no change is made to the document's share status.

Notes:

1. The output document library JCL statements are required.
2. The PROTECT command cannot change the accessibility of locked or archived documents.
3. An administrator can change the document password or share status of any document in any library.
4. A general user can change only the password or share status of documents he owns. If this user is the owner of a project or public library, he can change the password or share status of any document in this library, regardless of ownership.
5. If SHARE is specified for documents in a project library, it extends document accessibility to those who are not members of the project.
6. SHARE and NOSHARE have no effect when specified for documents in a public library, because all documents in a public library are automatically shareable.
7. NOSHARE is the default for documents being placed in private or project libraries with the IMPORT command. NOSHARE status assigned to a document in a project library limits access to the project members.

Examples

- This example assigns the password HIDEIT to the document named ACCOUNT in library 100 and specifies that it is not shared.

```
AUTH 123/ADMPASS  
PROTECT 100 ACCOUNT PASSWORD(/HIDEIT) NOSHARE
```

- This example removes the password from the document RECORDS in library 111 and makes it accessible to all users of the document library.

```
AUTH 123/ADMPASS  
PROTECT 111 RECORDS NOPASS SHARE
```

PURGE

The PURGE command removes profile records and documents from the document library.

There are six PURGE commands with administrative functions:

- PURGE CLASS
- PURGE CLUSTER
- PURGE DOCUMENT
- PURGE MAP
- PURGE PROCESS
- PURGE USER.

All PURGE commands except PURGE DOCUMENT require administrator authority. PURGE DOCUMENT and PURGE USER will delete documents from the library. The other PURGE commands affect profiles only.

To understand the descriptions of the PURGE commands that follow, the reader must have read and understood the material in the section "Commonly Used Command Operands" in Chapter 2.

PURGE CLASS

This command deletes a class entry record (CER) from the directory data set.

PURGE CLASS	name
-------------	------

name This operand specifies the class name of the CER to be purged.

Notes:

1. The output document library JCL statements are required.
2. Purging a CER from the document library does not affect documents stored with the class name associated with that CER, because a CER need not exist for all document classes used in the document library.

Example

This example purges the CER associated with the class called LETTERS.

```
AUTH 123/ADMPASS
PURGE CLASS LETTERS
```

PURGE CLUSTER

This command deletes a cluster profile record (CPR) from the directory data set, and optionally, deletes all documents associated with the cluster from the library.

PURGE CLUSTER	name <div data-bbox="690 400 915 476" style="border: 1px solid black; padding: 2px; display: inline-block;"> DOCUMENT NODOCUMENT </div>
----------------------	--

- name** This operand identifies the CPR to be purged.
- DOCUMENT** This operand specifies that all documents are to be purged from the cluster given on the command, together with the deletion of the CPR from the directory.
- NODOCUMENT** This operand specifies that only the CPR is to be purged from the directory. The documents remain intact on the source cluster, allowing the cluster to be reconnected later with a **DEFINE CLUSTER** command. This is the default.

Notes:

1. The output document library JCL statements are required.
2. When the **NODOCUMENT** operand is given, this command disconnects the specified cluster by deleting the CPR.

Documents stored in the cluster are not lost when the **NODOCUMENT** operand is given. The cluster is disconnected from DLF when the CPR is deleted, but the cluster itself and its documents are not deleted. A **DEFINE CLUSTER** command can be used to reconnect the cluster.

The default 4K cluster, **DSMPTLIB**, cannot be purged.
3. When the **DOCUMENT** operand is given, the **PURGE DOCUMENT** command processor is called to delete all documents associated with the specified cluster and then the **PURGE CLUSTER** command processor deletes the CPR. In this case, the cluster cannot be reconnected.

If the **DOCUMENT** operand is given and all documents are not purged (this could happen if some are **LOCKED** or the **PURGE DOCUMENT** command fails for some other reason), the CPR will not be deleted and the cluster will not be purged.
4. Any Class Entry Records (CERs) that associate a class of documents with the cluster being purged should either be purged (with the **PURGE CLASS** command) or associated with another cluster (using the **ALTER CLASS** command). This change prevents you from storing documents in a cluster that is not appropriate when you import documents with the classes associated with any purged cluster. The classes can be redefined (with the **DEFINE CLASS** command) or altered (with **ALTER CLASS**) to again associate the class with a cluster that you have reconnected to the document library (with the **DEFINE CLUSTER** command).

Example

This example deletes the CPR called **LARGE**. Documents stored in this cluster are not erased; they can be accessed after the cluster is reconnected by a **DEFINE CLUSTER** command specifying the same VSAM data set.

```
AUTH 1000000/MILLION
PURGE CLUSTER LARGE
```

PURGE DOCUMENT

This command purges documents from a user's library. Any library user can use this command.

PURGE DOCUMENT	docid [DATA(name)] [VERSION ({ number } { number1:number2 } { ALL })]]
----------------	---

docid This operand identifies the document to be purged. An administrator can specify any library and need not specify the document password. General users can specify their own, their project, or any public library in which they own documents as the dlib# (part of the docid). If the document to be purged has a password, the general user must supply it.

DATA(name) This operand specifies the data type of the document to be purged. The default is the current authorized user's default data type located in his UPR.

VERSION This operand specifies the version(s) of the document to be purged. If this keyword is omitted and the document has versions, an error message results. If the keyword is specified and the document is without versions, an error message also results. The VERSION operand without a value is invalid for this command.

number indicates the specific version of the document to be purged.

number1:number2 indicates an inclusive range of versions of the document to be purged.

ALL specifies that all versions of the document are purged.

Notes:

1. The output document library JCL statements are required.
2. Administrators can purge any document in any library. General users can purge any document they own. Each can also purge any documents in the library they own, regardless of ownership.
3. Locked documents cannot be purged.
4. Archived documents can be purged. This process involves the purging of only the copies of the DERs retained in the directory data set. The documents themselves and their DERs still remain on the data set to which they have been archived. These documents can no longer be retrieved, however, because the directory data set no longer contains the information required to retrieve them.
5. Administrators need not specify the document password in order to purge a document. However, general users must specify the password of a document that has one.
6. Only an administrator or the owner of a controlled project or public library can purge documents from those libraries.

Example

This example purges version 2 of the document named HISTORY with a data type of SCRIPT from user 100's private library.

```
AUTH 123/ADMPASS  
PURGE DOCUMENT 100 HISTORY DATA(SCRIPT) VERSION(2)
```

PURGE MAP

This command deletes a user mapping record (UMR), which associates an external user to a library user number in DLF.

PURGE MAP	HOST ('identifier') ID('identifier')
-----------	--------------------------------------

HOST('identifier') This operand specifies the external host system and is entered as a quoted string. Either 1 or 2 characters are allowed.

ID('identifier') This operand specifies the external user identifier and is entered as a quoted string. From 1 through 10 characters are allowed.

Notes:

1. The output document library JCL statements are required.
2. Both operands are required.
3. There are no defaults.

Example

This example deletes the UMR for an ATMS user identified as (operator) 36:

```
AUTH 123/ADMPASS
PURGE MAP HOST('AT') ID('36')
```

PURGE PROCESS

This command deletes a processor profile record (PPR) from the directory data set.

PURGE PROCESS	INDATA(name) OUTDATA(name)
----------------------	-----------------------------------

INDATA(name) This operand is the first of a pair of data type names that identifies the PPR purged.

OUTDATA(name) This term is the second of a pair of data type names that identifies the PPR purged.

Notes:

1. The output document library JCL statements are required.
2. Both operands are required.
3. There are no defaults for this command.

Example

This example purges the PPR associated with the data type pair ATMS and SCRIPT.

```
AUTH 123/ADMPASS  
PURGE PROCESS INDATA(ATMS) OUTDATA(SCRIPT)
```

PURGE USER

This command purges a user from the document library.

The purged user's UPR and all documents in its associated library are purged (including those owned by other users). The purge includes any UMRs that are associated with this user's library, archived documents, or locked documents. The purged user's documents residing in a project or public library owned by another user are not purged. The ownership of these documents is changed to the project or public library-owner.

This command can affect only a single user at a time.

An administrator cannot purge himself from the document library. One administrator must always be defined to DLF.

PURGE USER	number
------------	--------

number This operand specifies the library user number purged.

Notes:

1. The output document library JCL statements are required.
2. An administrator cannot purge himself from the document library. To change administrators, a DEFINE USER command must first designate the replacement administrator. This person can then purge the original administrator.

Examples

- This example purges user 1 001 and all documents in library 1 001. Ownership of documents belonging to user 1 001 in his project library is changed to the owner of the project library. Ownership of documents belonging to user 1 001 in public libraries is changed to the respective public library owners.

```
AUTH 123/ADMPASS  
PURGE USER 1001
```

- This example purges user 123 456. Library 123 456 is a public library. All documents in the library will be purged, whether owned by user 123 456 or not. Ownership of documents belonging to user 123 456 residing in his project library is changed to the owner of the project library. Ownership of documents belonging to user 123 456 that reside in other public libraries is changed to the respective public library owners.

```
AUTH 123/ADMPASS  
PURGE USER 123456
```

RETRIEVE

The only RETRIEVE command with administrative functions is:

- RETRIEVE ALL.

This command is used for returning archived documents to the document library. Documents that were locked at the time they were archived can be recovered. However, they will remain locked after they are returned to the document library.

To understand the description of the RETRIEVE command that follows, the reader must have read and understood the material in the section "Commonly Used Command Operands" in Chapter 2.

RETRIEVE ALL

This command recovers all selected documents in the FROM data set (an external data set or partitioned data set member). All archived versions of the selected document are retrieved. For VSE, use the ENVIRONMENT command to specify the archive data set's characteristics if the default values for the installation are not applicable. See Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics" for further details.

RETRIEVE ALL	{ ddname }	
	{ DSN(dsname) }	
	FROM ({ DSN(dsname(member)) }) }
	{ DSN('dsname') }	
	{ DSN('dsname(member)') }	
	[DATA(name)]	
	[USER ({ number }	
	{ number1:number2 }	
	{ :number2 }	
	[DATED(date)]	

FROM

This required operand identifies the external sequential data set or partitioned data set member that contains the archived documents to be retrieved.

ddname is the name of the DD, DLBL, or TLBL JCL statement defining the data set from which the documents are retrieved.

DSN... is valid in OS/VS2 only. It shows that the FROM data set is dynamically allocated. If the specified data set name is in quotation marks, it is used as given. Otherwise it is qualified as described in Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics." ARCHIVE is the suffix qualifier.

DATA(name)

This operand specifies the data type of the documents retrieved. The default is all data types that meet other selection criteria.

USER

This operand requires administrator authority. It specifies which user's documents are retrieved. If the current user is an administrator and USER is not specified, all user's documents that meet

the other selection criteria are retrieved. If the current user is a general user, only the documents that he owns, and that meet the other criteria are retrieved.

number specifies the library user number of a single user. This user's documents that meet the other selection criteria are retrieved.

number1:number2 specifies an inclusive range of library user numbers. These users' documents that meet the other selection criteria are retrieved.

:number2 specifies an inclusive range of library user numbers from the first in the document library through the one specified. These users' documents that meet the other selection criteria are retrieved.

DATED(date) This operand specifies that only documents originally stored in the document library on this particular date are to be retrieved. If DATED is not specified, all documents meeting the other selection criteria are retrieved. (date) must be in the form mm/dd/yy where mm is the month (1 to 12), dd the day of the month (01 to 31), and yy the last two digits of the year.

Notes:

1. The output document library JCL statements are required.
2. The FROM keyword is required in all systems.
3. Only a specific date is allowed in the DATED operand, not a range of dates such as may be specified with the ARCHIVE ALL command.

Example

This example retrieves all documents (without regard to date stored) previously archived to the sequential data set defined by the DD (OS/VS2) or DLBL/TLBL (VSE) JCL statement named ARCDs.

```
AUTH 123/ADMPASS
RETRIEVE ALL FROM(ARCDs)
```


CHAPTER 6. SETTING UP THE DOCUMENT LIBRARY

Before you can use DLF, you must define and format the data sets controlled by DLF. Use Access Method Services (AMS) to define, on any VSAM-supported device, the VSAM data sets you need.

To prepare the document library for use, you must:

1. Define a VSAM master catalog (if you do not already have one). Such a catalog already exists in OS/VS2, but might not be defined in VSE. See the appropriate Access Method Services manual.
2. Optionally, define a VSAM user catalog, if you intend to define these data sets in a user catalog. See the appropriate Access Method Services manual.

Note: To have password protection for the document library VSAM data sets, you must password-protect the VSAM master catalog, or define your VSAM data sets in a password-protected user catalog.

3. Define the VSAM data sets either in the VSAM master catalog or in a user catalog.
4. Install DLF and any required processors.

Note: Please check your JCL to ensure that each of the DD names you include is correctly mapped to the appropriate physical data set. Use caution when copying JCL to prevent DD names from being associated with the wrong physical data sets.

5. Define the document library and format the basic 4K source data set using the DEFINE SYSTEM command.
6. Define the required profile records of the directory data set.

DEFINING THE VSAM DATA SETS

At least two VSAM data sets are required, one key-sequenced and one entry-sequenced. You can define more than one ESDS.

Use an AMS DEFINE CLUSTER command to create the two VSAM data sets. Figure 21 on page 196 shows the required and recommended parameters for each data set.

Data Set	Required Parameters	Recommended Parameters
Directory (Key Sequence)	INDEXED KEYS(30 0) RECORDSIZE(64 256) SHAREOPTIONS(2)	INDEX(CONTROLINTERVALSIZE(x)NAME(name)) DATA(CONTROLINTERVALSIZE(y)NAME(name)) IMBED REPLICATE CYLINDERS FREESPACE SPEED READPW ERASE
Source (Entry Sequence)	NONINDEXED CONTROLINTERVALSIZE(4096) RECORDSIZE(4089 4089) CYLINDERS(n 0)	SHAREOPTIONS(2) SPEED READPW ERASE DATA(NAME(name))

Figure 21. VSAM Parameters for the Document Library Data Sets. The CONTROLINTERVALSIZE values for the directory data set must be large enough so that $2x + y \geq 4096$. For the IBM 3340, the recommended values are $x=1536$ and $y=1024$; for other devices, the recommended values are $x=1024$ and $y=2048$.

Specify UNIQUE only if you do not plan to extend your data sets later. A UNIQUE data set does not require allocated space before the DEFINE CLUSTER command is executed. However, in VSE, UNIQUE requires you to specify (on the EXTENT statement) the starting location and number of tracks required.

If you want to suballocate your data sets, specify SUBALLOCATION. In this case, first allocate the VSAM space by executing a DEFINE SPACE command before the DEFINE CLUSTER command. Then additional volumes can be allocated after the original definition.

A data name should be specified in the cluster definition of the source data set. You can then add candidate volumes later.

There must be at least one source data set with a control interval size of 4 096 bytes. This is the 4K default cluster. Other source data sets can be defined for use with the library. The value specified for control interval size of these source data sets must be in the range between 512 and 16 384 bytes in multiples of 512 bytes through 8 192, and in multiples of 2 048 bytes thereafter. The record size should be 7 bytes less than the control interval size.

Note: In addition to the required and recommended parameters shown in the table above, you can include other VSAM options as required by your installation. For example, you might want to specify a name for both the INDEX and DATA sets of the directory cluster.

Estimating Space Requirements

Use the following information as a guideline when you plan the space requirements (CYLINDERS parameter) for the VSAM data sets.

Directory Data Set (key-sequenced): This data set contains all the profile records and a document entry record (DER) for each document in the source data set. The amount of space allocated for this data set depends on the estimated number of entries.

The items in the directory data set and the space required for them are:

<u>Space</u>	<u>Item</u>
47 bytes	System profile
38 bytes	Class entry
58 bytes	Cluster profile
241 bytes	Document entry
32 bytes	Library entry
33 bytes	User mapping
38 bytes	Processor profile
108 bytes each	User profile
Based on CI size	VSAM RDFs and CIDF in each CI
Based on CI size	Index component

The FREESPACE designation for the directory data set and the distribution of profile records in your document library can have a significant affect on space requirements. See the appropriate VSAM manuals for a description of VSAM FREESPACE usage and its interaction with the sequential insert strategy. The FREESPACE designation in Figure 22 and Figure 23 is arbitrary. A different FREESPACE allocation may be appropriate for your installation.

Source Data Set (entry-sequenced): Multiple VSAM source clusters can be used to store the contents of documents. The clusters are defined using the Access Method Services with different control interval sizes as described in Figure 21 on page 196. In this way, space can be efficiently used by storing documents of similar sizes into a cluster with a control interval size corresponding to the average size of the documents. The default cluster, DSMPTLIB, is defined and formatted by the DEFINE SYSTEM command. It must be defined with a control interval size equal to 4 096 bytes.

The DEFINE CLUSTER command is used to format the clusters other than the default cluster, DSMPTLIB. Documents can be assigned to a specific cluster by specifying the CLUSTER keyword in the IMPORT or COPY DOCUMENT command. A document can also be assigned to a specific cluster if its class is associated with a cluster name through a DEFINE CLASS command. The document's class can be assigned explicitly using the CLASS keyword on the command or implicitly by using the default class for the current DLF user. Different versions of a document can reside on different clusters.

To estimate the space required, estimate the total number of bytes required for all documents and divide by the number of bytes per cylinder for the DASD being used. Add an appropriate number of cylinders for expansion. Only one document can reside in a control interval of a cluster. If a document is larger than the control interval, it begins to fill the next control interval. If a document does not fill a control interval, the unused space remains empty. Figure 37 on page 254 shows examples of document space usage in clusters with different control interval sizes.

All of the data sets can be on the same direct-access volume or on separate volumes. No secondary space should be specified for an entry-sequenced data set (the second number in the CYLINDERS, RECORDS, or TRACKS parameter must be zero or omitted) because the DEFINE CLUSTER command (or DEFINE SYSTEM command for the default cluster, DSMPTLIB) formats the data set in its entirety. Specifying secondary space other than zero would immediately allocate all secondary extents.

Allocating several volumes for a source data set improves performance during library access. For a multivolume entry-sequenced data set, as soon as the space on the first volume is formatted, space on the next volume is formatted.

Figure 22 on page 199 gives an example of defining the OS/VS2 VSAM data sets for the document library. Figure 23 on page 200 gives an example of defining the VSE VSAM data sets.

AUTHORIZED PROGRAM FACILITY

In an OS/VS2 environment, DLF must be linked as an authorized program and reside in an authorized program library. This allows the library program to access the library clusters without requesting the VSAM password for these clusters from the CPU operator. Modules that call the library program as a subroutine must also be authorized and reside in an authorized library. Modules that are called by the library program must reside in an authorized library; but need not be linked as authorized.

Modules called by DLF include:

- ATMS-III conversion program (part of the DCF product)
- SCRIPT/VS (part of the DCF product)
- User exit routine (DSMSPUXI user-supplied or IBM skeleton routine)
- User-written processors called from DLF command processors.
- The DLF password hashing routine (DSMSPHSH).

INSTALLING DLF

The DLF load module is named DSMSPEXC. It must reside in an Authorized Program Facility load module library and be linked as an authorized program when used with OS/VS2. In VSE, the load module must reside in a core-image library. The installation documentation distributed with the program product describes the installation process.

```

//DEFINE JOB
//DEFLIBS EXEC PGM=IDCAMS,REGION=1000K
//STEP CAT DD DSN=VSAMCAT,DISP=SHR
//SYS PRINT DD SYSOUT=A
//DLFVOL DD UNIT=3330,VOL=SER=DLFPAK,DISP=OLD
//SYSIN DD *
    DEFINE CLUSTER (NAME (DIRECTRY) -
        FILE (DLFVOL) VOLUMES (DLFPAK) -
        CYLINDERS (15) RECORDSIZE (64 256) -
        FREESPACE (30 30) SHAREOPTIONS (2) -
        ERASE SPEED INDEXED KEYS (30 0) IMBED -
        READPW (RPASS)) -
        INDEX(NAME(DIRECTRY.INDEX) CONTROLINTERVALSIZE (1024)) -
        DATA(NAME(DIRECTRY.DATA) CONTROLINTERVALSIZE (2048))
    DEFINE CLUSTER (NAME (SOURCE) -
        FILE (DLFVOL) VOLUMES (DLFPAK) -
        CYLINDERS (60) RECORDSIZE (4089 4089) -
        SHAREOPTIONS (2) ERASE -
        SPEED NONINDEXED -
        READPW (RPASS) -
        CONTROLINTERVALSIZE (4096) ) -
        DATA (NAME (SOURCE.DATA))
    DEFINE CLUSTER (NAME (SOURCEA) -
        FILE (DLFVOL) VOLUMES (DLFPAK) -
        CYLINDERS (20) RECORDSIZE (2041 2041) -
        SHAREOPTIONS (2) ERASE -
        SPEED NONINDEXED -
        READPW (RPASS) -
        CONTROLINTERVALSIZE (2048) ) -
        DATA (NAME (SOURCEA.DATA))
/*

```

for OS/VS2

Figure 22. Examples of Defining Document Library Data Sets (OS/VS2)

```

// JOB DEFINE
// DLBL IJSYSUC,'DSMCAT',,VSAM
// DLBL DLFVOL,'DLFLIB',,VSAM
// EXTENT ,DLFPAK
// EXEC IDCAMS,SIZE=AUTO
//      DEFINE CLUSTER (NAME (DIRECTRY) -
//          FILE (DLFVOL) VOLUMES (DLFPAK) -
//          CYLINDERS (15) RECORDSIZE (64 256) -
//          FREESPACE (30 30) SHAREOPTIONS (2) -
//          ERASE SPEED INDEXED KEYS (30 0) IMBED -
//          READPW (RPASS)) -
//          INDEX(NAME(DIRECTRY.INDEX) CONTROLINTERVALSIZE (1024)) -
//          DATA(NAME(DIRECTRY.DATA) CONTROLINTERVALSIZE (2048))
//      DEFINE CLUSTER (NAME (SOURCE) -
//          FILE (DLFVOL) VOLUMES (DLFPAK) -
//          CYLINDERS (60) RECORDSIZE (4089 4089) -
//          SHAREOPTIONS (2) ERASE -
//          SPEED NONINDEXED -
//          READPW (RPASS) -
//          CONTROLINTERVALSIZE (4096) ) -
//          DATA (NAME (SOURCE.DATA))
//      DEFINE CLUSTER (NAME (SOURCEA) -
//          FILE (DLFVOL) VOLUMES (DLFPAK) -
//          CYLINDERS (20) RECORDSIZE (2041 2041) -
//          SHAREOPTIONS (2) ERASE -
//          SPEED NONINDEXED -
//          READPW (RPASS) -
//          CONTROLINTERVALSIZE (2048) ) -
//          DATA (NAME (SOURCEA.DATA))
/*
/8

```

for VSE

Figure 23. Example of Defining Document Library Data Sets (VSE)

INSTALLING PROCESSORS

Installing processors to be used with DLF follows a procedure similar to that for the program.

A processor can be any program that processes a document supplied to it by the DLF. The term processor as used in this DLF Guide refers to a non-DLF program that may be called by the library program to perform special processing for certain DLF commands (IMPORT, EXPORT, SCRIPT, READ). Any conversion routine, such as the IBM-supplied ATMS conversion program is a processor. (The installation documentation distributed with DCF describes the installation procedure for the ATMS conversion program.)

To install a processor:

1. Link the processor into an authorized load library in OS/VS2 or into a core-image library in VSE.
2. Define the processor profile using the DEFINE PROCESS command in the document library, specifying the name of this load module.

DEFINING REQUIRED PROFILES

To prepare the document library for use, you must:

1. Define the system profile record with the DEFINE SYSTEM command. This command:
 - Causes the basic 4K source data set to be formatted
 - Causes both the system profile record and the cluster profile record for the basic 4K cluster to be written to the directory data set
 - Sets the system characteristics you specified in the DEFINE SYSTEM command.

CAUTION:

The DISP=SHR parameter allocates the document library to concurrent jobs. It does not prevent an OPEN for update that coincides with an OPEN for access. To avoid not found or lost data situations that may occur, DLF enqueues (locks in VSE) the document library for UPDATE and SHARE depending on the command being processed. On jobs with DISP=SHR specified for the output library, the LIBRARY keyword must have been specified on a DEFINE SYSTEM or ALTER SYSTEM command before running the job.

2. In the same job step that defines the system profile, you must define a user with administrative authority with the DEFINE USER command. Other users can optionally be defined at this time.
3. Define processor profiles with the DEFINE PROCESS command.
4. Define the ATMS processor profile if the ATMS Conversion Program is to be used. As seen in Figure 24 on page 202, its load module name is DSMACAIP.

Figure 24 is an example of defining required profiles. Note the comments in the examples.

```

//DEFINE JOB
//DLFINIT EXEC PGM=DSMSPEXC,REGION=1000K,PARM='LIST'
//STEP CAT DD DSN=VSAMCAT,DISP=SHR
//SYS PRINT DD SYSOUT=A
//DSMLIST DD SYSOUT=A
//DSMPDIR DD DSN=DIRECTRY,DISP=OLD
//DSMPTLIB DD DSN=SOURCE,DISP=OLD
//DSMPLIB1 DD DSN=SOURCEA,DISP=OLD
//SYSIN DD *
/* NO AUTH CONTROL STATEMENT IS REQUIRED TO DEFINE*/ -
/* THE FIRST PROFILES IN THE LIBRARY. (SYSTEM/USER)*/ -
/* DEFINE SYSTEM AND DEFINE USER ADM MUST BE */ -
/* INCLUDED IN THE SAME JOB STEP */
DEFINE SYSTEM LIBRARY (NQNAME)
DEFINE USER 12104 PASSWORD(/SECRET) ADM
/* THE ATMS ATTRIBUTE PROFILE IS DEFINED TO */-
/* CONVERT ATMS DOCUMENTS FOR SCRIPT/VS PROCESSING */-
DEFINE PROCESS INDATA(ATMS) OUTDATA(SCRIPT) -
ENTRY (DSMACAIP)
DEFINE CLUSTER LARGE INDD(DSMILIB1) OUTDD(DSMPLIB1)
/*

```

for OS/VS2

```

// JOB DEFINE
/* STEP CAT FOR A VSAM USER CATALOG
// DLBL IJSYSUC,'DSMCAT',,VSAM
// DLBL DSMPDIR,'DIRECTRY',,VSAM
// EXTENT ,DSMVOL
// DLBL DSMPLIB,'SOURCE',,VSAM
// EXTENT ,DSMVOL
// DLBL DSMPLIB1,'SOURCEA',,VSAM
// EXTENT ,DSMVOL
// OPTION LISTX,NODUMP
// EXEC DSMSPEXC,SIZE=AUTO
/* NO AUTH CONTROL STATEMENT IS REQUIRED TO DEFINE*/ -
/* THE FIRST PROFILES IN THE LIBRARY. */ -
/* DEFINE SYSTEM AND DEFINE USER ADM MUST BE */ -
/* INCLUDED IN THE SAME JOB STEP. */
DEFINE SYSTEM LIBRARY (NQNAME)
DEFINE USER 12104 PASSWORD(/SECRET) ADM
/* THE ATMS ATTRIBUTE PROFILE IS DEFINED TO */-
/* CONVERT ATMS DOCUMENTS FOR SCRIPT/VS PROCESSING */-
DEFINE PROCESS INDATA(ATMS) OUTDATA(SCRIPT) -
ENTRY (DSMACAIP)
DEFINE CLUSTER LARGE INDD(DSMILIB1) OUTDD(DSMPLIB1)
/*
/&

```

for VSE

Figure 24. Examples of Defining Minimum Profiles

PROVIDING RACF SECURITY FOR DATA AND USERS

When DLF runs under MVS, you can, with the help of your installation's RACF Administrator, use the Resource Access Control Facility (RACF, Program Product 5740-XXH) to protect DLF user numbers and/or the VSAM clusters containing the DLF document libraries. However, DLF has its own security features which work with or without RACF.

DLF and RACF provide security in very different ways:

RACF can protect an entire DLF document library if both the DLF directory data set (VSAM key-sequenced data set) and the DLF source data set (VSAM entry-sequenced data set) are properly defined to RACF. If a DLF document library is RACF protected with a universal access of NONE, any user of MVS who is not specifically RACF permitted to that DLF document library, will be denied access to ALL documents and DLF profiles in that DLF document library even without the exercise of any of DLF's security features.

If DLF document libraries and associated data sets are not RACF protected, they are available to any user of MVS with the ability to use an application capable of accessing those resources. Appropriate use of DLF's security features can prevent unauthorized access of any DLF resource through DLF, but RACF, running in a host system that uses RACF as designed, can also protect them from access through any application.

DLF security features and RACF protection can also work with each other. Typically, RACF allows FULL access to an entire DLF document library by a limited group of MVS users, while the DLF security features allow you to vary the level of protection from document to document and from user to user.

Although RACF can only control access to ALL documents in a given DLF source data set (VSAM ESDS) of a document library by either ALL or NO members of a group of users, DLF security features can be used to control different types of access to SOME, ALL or NO documents within a DLF document library, by SOME, ALL, or NO members of a group of users. (For example, if a DLF document has a password and a DLF user issues a DLF command that requires the specification of the correct password for that document, the user cannot successfully execute that DLF command without knowing that password -- even if that user has full RACF access to the entire document library.)

Note: For RACF command syntax, consult a recent edition of the MVS Resource Access Control Facility (RACF) Command Language Reference.

Protecting DLF Libraries with RACF

Any non-VSAM, DLF data set can be protected by one RACF ADDSD command, and individual MVS users or groups of users defined to RACF can be permitted any one of four levels of RACF access (NONE, READ, UPDATE, ALTER) with a RACF PERMIT command.

RACF protecting VSAM data sets is more complicated since, as of RACF Release 5, a single RACF ADDSD command cannot protect an entire VSAM cluster and RACF access levels are limited to either NONE or CONTROL for VSAM data sets.

- Each DLF directory data set (VSAM key-sequenced data set) has three names associated with it that were defined by the NAME, DATA, and INDEX operands of the VSAM DEFINE CLUSTER command. Each of these three names must be protected with a RACF ADDSD command to properly protect the VSAM cluster as a whole. PERMIT is a RACF command to specify the users or groups to be allowed access by RACF, and it must also be issued for each of the three associated names.

- Each DLF source data set (VSAM entry-sequenced data set) has two names associated with it that were defined by the NAME and DATA operands of the VSAM DEFINE CLUSTER command. (There is no index.) Both of these must be protected with a RACF ADDSD command to properly protect the VSAM cluster as a whole. PERMIT is a RACF command to specify the users or groups to be allowed access by RACF, and it must also be issued for both of the associated names.

Examples of RACF commands to protect the DLF directory data set:

ADDSD	'DSMLIB.DIRECTRY'	UACC(NONE)	
PERMIT	'DSMLIB.DIRECTRY'	ID(racuser1)	ACCESS(CONTROL)
PERMIT	'DSMLIB.DIRECTRY'	ID(racgrp9)	ACCESS(CONTROL)
ADDSD	'DSMLIB.DIRECTRY.DATA'	UACC(NONE)	
PERMIT	'DSMLIB.DIRECTRY.DATA'	ID(racuser1)	ACCESS(CONTROL)
PERMIT	'DSMLIB.DIRECTRY.DATA'	ID(racgrp9)	ACCESS(CONTROL)
ADDSD	'DSMLIB.DIRECTRY.INDEX'	UACC(NONE)	
PERMIT	'DSMLIB.DIRECTRY.INDEX'	ID(racuser1)	ACCESS(CONTROL)
PERMIT	'DSMLIB.DIRECTRY.INDEX'	ID(racgrp9)	ACCESS(CONTROL)

Notice that there are both ADDSD and PERMIT commands for each of the three associated names. In a host system in which RACF was active, the nine commands above would accomplish the following for the DLF directory data set called 'DSMLIB.DIRECTRY':

1. RACF would prevent all MVS users (other than 'racuser1' and the members of 'racgrp9') from accessing the DLF directory data set through any means.
2. The MVS users in 'racgrp9' and the one called 'racuser1' would be allowed full access to all resources of the DLF directory data set -- provided that DLF itself did not restrict access.

Protecting DLF User Numbers with RACF

There are four steps required to use RACF protection for DLF user numbers:

1. First, a RACF Administrator or System Programmer with SPECIAL RACF authority must use the RACF ICHERCDE assembler macro to define a new class for DLF users to the RACF Class Descriptor Table (CDT) and assemble and link-edit the result to produce a new RACF load module. (See Appendix C of Resource Access Control Facility (RACF) Installation Reference Manual.)
2. The second step is for a RACF Administrator or System Programmer with SPECIAL RACF authority to issue the RACF ALTUSER command to give CLAUTH to a DLF Administrator's LOGON/USERID for the class that was defined to the CDT in step 1. That will allow this DLF Administrator to add and delete DLF user numbers in the class.
3. The third step is for a DLF Administrator to use the DLF command of DEFINE SYSTEM (or ALTER SYSTEM) with the new RACF class specified in the RACFNAME operand. This will cause the class-name to be entered in the DLF System Profile Record (SPR) so that whenever DLF is initiated, it will know to allow RACF to judge whether a DLF user is authorized to access this DLF document library.
4. The fourth step is for a DLF Administrator to use the RACF RDEFINE command to define the DLF user numbers to the RACF class as members of the class. (The RACF RALTER and RDELETE commands can change or delete the original definitions when necessary.)

CHAPTER 7. ADMINISTERING THE DOCUMENT LIBRARY

This chapter discusses the operations an administrator performs on the document library. It provides examples of several tasks. JCL procedures can be set up for normal operations.

Administration of the document library requires administrator authority. This chapter describes operations that enable an administrator to:

- Authorize new users
- Copy profiles and documents
- Ensure adequate space for the document library
- List library space usage
- Allocate additional space in the library
- Reorganize the document library
- Modify attribute profiles
- Define user maps for external systems
- Define classes of data.

AUTHORIZING NEW USERS

A user is authorized by your defining a user profile that specifies the type of library the user has and whether the user has access to a project library as well.

Consider, for example, a user with the following requirements:

- The user expects to save, at one time, no more than 50 documents averaging 15 000 bytes each.
- The user needs to be able to use project library 110 000.
- The user wants a password.

The following DEFINE USER command could be used to define such a user:

```
DEFINE USER 123456 PASSWORD(/SECRET) -  
SPACE(750) PROJLIB(110000)
```

The SPACE option specifies 750 000 bytes as the space limit for this user's library. The default operands are LIBTYPE(PRIVATE) and NOADM. Note that the project library (110 000) must be defined for the command to be accepted.

Note: Only one document can reside in a control interval of a cluster. If a document is larger than the control interval, the document begins to fill the next control interval. If a document does not fill a control interval, the unused space remains empty.

COPYING PROFILES AND DOCUMENTS

Existing profiles and documents can be copied into a document library from a copy tape that was created from another document library. The COPY IN and COPY OUT commands are used for this operation. Be careful when creating profiles in this manner. Under some conditions, unwanted profiles can be copied. Also, user documents can be copied when there is no user profile defined for that user number. This condition can occur if a user's documents in public and project libraries are copied into a document library in which profiles for the project and public libraries do not exist. This can happen only when the user doing the copying is an administrator. These documents are still copied, but with ghost entries. Because the UPR is missing, the documents are not accessible. However, a listing of document names, in collating sequence, belonging to the user number immediately preceding the missing user number will also include these document names. UPRs should be defined for libraries containing documents that have been copied in without associated profiles. The documents can then be purged or accessed as required.

Note: The COPY commands can be used to copy documents from one cluster to another cluster that has the same CI size. The EXPORT/IMPORT commands must be used to move documents between clusters of different CI sizes.

ENSURING ADEQUATE SPACE FOR THE DOCUMENT LIBRARY

The following options allow an administrator to ensure that there is adequate space in the document library:

- Library users can list current library space usage.
- Document owners can archive infrequently used documents.
- Document owners can back up their inactive or obsolete documents onto tape with a COPY OUT command and then delete the documents from the document library with the PURGE DOCUMENT command.
- An administrator can delete obsolete user profiles and their associated documents with the PURGE USER command.
- Administrators can allocate additional space in the source VSAM data sets. The ALTER CLUSTER command can then be used to format the additional space.

Listing Library Space Usage

You can use the LIST CLUSTER and LIST DOCUMENT commands with the SPACE operand to determine the physical space usage in the document library. Both commands provide an alternative Space Usage report for this purpose.

THE LIST CLUSTER SPACE USAGE REPORT: expands the Standard LIST CLUSTER report to include:

- Cluster size (TOTAL)
- Amount of the cluster currently in use (USED)

- Amount of storage available to store additional documents (AVAILABLE). The available space is divided into two types:
 - Space that has never been used (NEVER USED) - which is contiguous space at the end of the cluster.
 - Space freed by deleting or archiving documents (REUSABLE) - which is normally not contiguous. None of this space will be used for document storage until all space in the cluster has been used at least once - indicated by NEVER USED = 0.

All data in the LIST CLUSTER Space Usage report (except CTL INTRVL SIZE) is shown in thousands of bytes and, where appropriate, as percentages. CTL INTRVL SIZE is given in bytes.

The information in this report can be used to determine which clusters should be expanded by allocating additional space or which should be reorganized due to fragmentation.

THE LIST DOCUMENT SPACE USAGE REPORT: expands the Standard LIST DOCUMENT report to include:

- Control interval size of the cluster
- COUNT of control intervals a document is using
- TOTAL number of bytes in the control intervals being used
- Percentages with the number of bytes that are FILLED
- Percentages with the number of bytes that are VACANT.

If there is more than one NON-ARCHIVED document listed, a summary page is also generated. It provides a count and other summary information about ONLY the NON-ARCHIVED documents listed. This includes the SMALLEST, AVERAGE, and LARGEST document sizes and totals for the number of CTL-INTRVLs and RECORDS. The summary page also lists a total number of TEXT BYTES that excludes the bytes used by DLF for control purposes. This is essentially a total for the field called FILE SIZE on a Standard LIST DOCUMENT report (excluding any archived documents). The last lines on the summary page are totals and overall percentages for the FILLED, VACANT, and TOTAL fields on the Space Usage report.

All data for individually listed documents is reported in bytes and, where applicable, a percentage. All percentages are rounded with the exception of those values between 0% and 1% which are rounded to 1% and those values between 99% and 100% which are rounded to 99%.

On the Standard LIST DOCUMENT report, the file size of a document is reported in bytes, with the addition of a 2-byte length field used by DLF for each logical record. On the Space Usage LIST DOCUMENT report, the number of bytes listed as FILLED includes an additional 48 bytes per control interval occupied by the document. This is the space which DLF uses to manage the control interval.

One of the most significant fields on this report is that labeled VACANT. This field shows the number of bytes in the control interval that are not being used at all. For example, if 1 800 bytes are VACANT in a document occupying one control interval of 2 048 bytes, the space usage in the library could be improved if the document were moved to a cluster with a smaller control interval size.

The information provided in the LIST DOCUMENT Space Usage report can be used to determine which documents poorly use the particular control interval size of the cluster in which they are stored. It can also identify document owners who may be inappropriately storing certain documents in a given cluster.

Deleting Obsolete Documents

You can free direct-access storage space by deleting documents with the PURGE DOCUMENT command. Obsolete user profiles should also be deleted.

Deleting a User Profile

To delete a UPR, use the PURGE USER command as explained in Chapter 5, "Administrator Commands." Deleting a user profile record (UPR) causes deletion of all documents (regardless of ownership) in that user's library. Any documents to be retained can be transferred to another user library (using COPY DOCUMENT or COPY OUT commands) before deleting the user. Documents owned by the deleted user that reside in another library (that is, his project library or any public library) are not deleted, but ownership is transferred to the owner of the library in which they reside.

Allocating Additional Space in the Document Library

To acquire additional space for the document library, you can either extend the current VSAM data sets or define new larger ones, in which case, the current document library can be copied to the larger one with the COPY LIBRARY command.

To allocate additional space for a document library, perform the following steps:

1. Use the VSAM Access Method Services ALTER command (with the ADDVOLUMES parameter) to allocate an additional volume, or volumes, to either the key- or entry-sequenced data set, as explained in the appropriate Access Method Services manual. (See Figure 25 on page 209.)
2. If an entry-sequenced data set is extended, use the DLF ALTER CLUSTER command to format the additional space. The ALTER SYSTEM command also can be used to format additional space in the default 4K cluster (cluster name DSMPTLIB). The space is not available to DLF until it is formatted. If only the key-sequenced data set is extended, no formatting is necessary. (See Figure 25 on page 209.)


```

//EXTEND JOB
//STEP1 EXEC PGM=IDCAMS
//STEP1 DD DSN=VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=A
//DSMDD DD VOL=SER=DSM02,UNIT=3330,DISP=OLD
//SYSIN DD *
/* IF THE DATA FACILITY EXTENDED FUNCTION (5740-XYQ) */ -
/* IS INSTALLED, OMIT THE DEFINE SPACE COMMAND */
DEFINE SPACE (VOLUMES (DSM02) CYLINDERS (20)-
FILE (DSMDD)) CATALOG (VSAMCAT)
/* THE NAME 'SOURCE.DATA/SECURITY' MUST */ -
/* HAVE BEEN SPECIFIED WITH THE DATA KEYWORD */ -
/* WHEN THE CLUSTER WAS DEFINED */
ALTER SOURCE.DATA/SECURITY -
ADDVOLUMES (DSM02)
/*
//STEP2 EXEC PGM=DSMSPEXC,PARM='LIST'
//STEP2 DD DSN=VSAMCAT,DISP=SHR
//SYSPRINT DD SYSOUT=A
//DSMLIST DD SYSOUT=A
//DSMPTDIR DD DSN=DIRECTRY,DISP=OLD
//DSMPTLIB DD DSN=SOURCE,DISP=OLD
//DSMPLIB1 DD DSN=SOURCEA,DISP=OLD
//SYSIN DD *
AUTH 12104/SECRET
ALTER CLUSTER LARGE
/*

```

```

// JOB EXTEND
// DLBL IJSYSUC,'VSAMCAT',,VSAM
// DLBL DSMDD,'CONTINUATION',,VSAM
// EXTENT ,USRPK2,1,0,5130,380
// DLBL DSMPLIB,'SOURCEA',,VSAM
// EXTENT ,USRPK2
// EXEC IDCAMS,SIZE=AUTO
DEFINE SPACE (VOLUMES (USRPK2) CYLINDERS(20)-
FILE(DSMDD)) CATALOG(VSAMCAT)
/* THE NAME 'SOURCE.DATA/SECURITY' MUST */ -
/* HAVE BEEN SPECIFIED WITH THE DATA KEYWORD */ -
/* WHEN THE CLUSTER WAS DEFINED */
ALTER SOURCE.DATA/SECURITY -
ADDVOLUMES (USRPK2)
/*
// OPTION LISTX
// DLBL IJSYSUC,'VSAMCAT',,VSAM
// DLBL DSMPDIR,'DIRECTRY',,VSAM
// EXTENT ,USRPK1
// DLBL DSMPLIB,'SOURCE',,VSAM
* VSAM REQUIRES AN EXTENT STATEMENT FOR ALL
* VOLUMES OF A VSAM FILE.
// EXTENT ,USRPK1
// EXTENT ,USRPK2
// DLBL DSMPLIB,'SOURCEA',,VSAM
// EXTENT ,USRPK2
// EXEC DSMSPEXC,SIZE=AUTO
AUTH 12104/SECRET
ALTER CLUSTER LARGE
/*
/&

```

Figure 25. Examples of Allocating and Formatting Additional Library Space

Notes:

1. These examples assume that the VSAM data sets are suballocated data sets because you cannot extend UNIQUE data sets.
2. See Chapter 6, in "Estimating Space Requirements," for information on defining additional VSAM clusters for efficient use of space.

REORGANIZING THE DOCUMENT LIBRARY

The document library may require reorganization for one or more of the following reasons:

- To reduce library access time by reducing document fragmentation
- To transfer the document library to a different storage device
- To reorganize the directory data set.

As users delete and add documents, an increasing number of documents will no longer occupy contiguous control intervals in direct-access storage; that is, DLF's library space will become fragmented. No space is lost because DLF reuses all space that has been freed, but document access can take more time. The relative amount of REUSABLE space can be identified by using the LIST CLUSTER command with the SPACE operand. None of the reusable space will be used for document storage until all space in the cluster has been used at least once (NEVER USED space = 0). See "Listing Library Space Usage" on page 206 for more information. To minimize the effect of fragmented space on performance, the document library should be reorganized periodically.

To reorganize, follow these steps if disk space is limited:

1. Use the COPY OUT command to copy the entire contents of the document library to a sequential data set such as tape.
2. Use the Access Method Services DEFINE CLUSTER commands to define new VSAM data sets. (See Chapter 6, "Setting Up the Document Library"). You must first delete the existing ones with the VSAM DELETE CLUSTER command if you intend to use the same names for the new VSAM data sets and to put them on the same DASD volume as the existing ones.
3. In the next job step, specify the DEFINE SYSTEM and DEFINE CLUSTER commands to format the newly defined VSAM data sets and specify the DEFINE USER command to define a user with administrative authority.
4. Specify the AUTH command that refers to the newly defined administrator.
5. Specify the COPY IN command to restore the copy of the document library to the newly formatted space.

To reorganize, follow these steps if your disk space is not limited:

1. Use the Access Method Services DEFINE CLUSTER command to define new VSAM data sets. (See Chapter 6, "Setting Up the Document Library.") You must define the VSAM data sets with different names.
2. In the next job step, specify the DEFINE SYSTEM and DEFINE CLUSTER commands to format the newly defined VSAM data sets and specify the DEFINE USER command to define a user with administrator authority. The user must exist in the original library with administrator authority.

3. Specify the AUTH command that refers to the newly defined administrator.
4. Specify the COPY LIBRARY command to copy the entire contents of the document library to the newly formatted space.
5. Use the VSAM DELETE CLUSTER command to delete the original clusters. Then you can rename the new clusters to the original cluster names.

Reorganizing the Directory

If only the key-sequenced data set needs reorganizing, use the Access Method Services REPRO command. Reorganizing might be necessary after a large number of users have been defined. Without the reorganization, DLF's performance may suffer. (Information about the REPRO command can be found in the appropriate Access Method Services manual.)

Reorganizing an ESDS Source Cluster

If one or more ESDS source cluster(s) needs to be reorganized, and not the entire library, you can use the following steps to reorganize each cluster¹.

1. Use the DLF COPY OUT command to copy the contents of the cluster to a sequential data set. The CLUSTER operand can be used to limit the COPY OUT to the cluster being reorganized.
2. Use the DLF PURGE CLUSTER command with the DOCUMENT operand to purge every document in the cluster being reorganized, together with the CPR. This command ensures that no DERs remain in the DLF output directory for the cluster as well as deleting the CPR for the cluster.

An alternative is to use the DLF PURGE DOCUMENT command to purge each document from the cluster being reorganized, then issue the DLF PURGE CLUSTER command to purge the cluster being reorganized.

3. Use the Access Method Services DEFINE CLUSTER command to define a new VSAM entry-sequenced data set. The new data set must have the same control interval size as the original cluster.

When disk space is limited, or you intend to use the same name for the new VSAM data set and put it on the same DASD volume; you must first use the AMS DELETE CLUSTER command to remove the existing data set. Otherwise, the original cluster can be deleted, and the new cluster renamed to replace it, after the new cluster is generated and verified.

4. Use the DLF DEFINE CLUSTER command to format the new cluster.
5. Use the DLF COPY IN command with the CLUSTER operand to restore the contents of the cluster from the data set to which it was copied in step 1. The CLUSTER operand selects only those documents in the cluster being reorganized.

¹ You may also reorganize your entire document library as described under "Reorganizing the Document Library" on page 210. Reorganizing the entire library will reorganize the cluster(s) you particularly want to reorganize. It will also accomplish all else that is done in a complete library reorganization. You should determine which method best suits your requirements.

Note:

When reorganizing an ESDS source cluster, an ADMINISTRATOR needs to be aware of potential conflicts when using the Access Method Services (AMS) DELETE CLUSTER and DEFINE CLUSTER commands in conjunction with the DLF DEFINE CLUSTER command.

If an ESDS source cluster is deleted and redefined with AMS, the DLF output directory records related to that cluster remain in the DLF output directory. This means that the CPR and the DERs for all documents included in the cluster are still in the DLF output directory.

If the source cluster itself is deleted and redefined with the same name with AMS and its associated DLF directory records are **not** deleted, DLF will detect and prevent potential errors that might otherwise occur when the cluster is redefined to DLF with the same name by issuing the DEFINE CLUSTER command with the NOFORCE operand.

DLF will recognize residual directory records associated with a former source cluster when a new cluster with the same name is being defined to DLF with the DEFINE CLUSTER command with the NOFORCE operand and will terminate processing with an error message. If the CPR is still in the DLF output directory, the PURGE CLUSTER command can be issued to delete it. If DERs remain in the DLF output directory, the DEFINE CLUSTER command with the FORCE operand can be issued to delete all the existing DERs associated with the cluster being defined and then create a new CPR to define the cluster.

MODIFYING AN ATTRIBUTE PROFILE

You can change a profile to refer to a new load module with another entry name. Or, to cause no processing to occur when this processor is called, you can specify 0 as the entry name. See the examples below.

```
//ALTER JOB                                     for OS/VS2
//STEP EXEC PGM=DSMSPEXC,PARM='LIST'
//STEP CAT DD DSN=VSAMCAT,DISP=SHR
//SYS PRINT DD SYSOUT=A
//DSMLIST DD SYSOUT=A
//DSMPDIR DD DSN=DSMLIB.DIRECTRY,DISP=OLD
//DSMPDLIB DD DSN=DSMLIB.SOURCE,DISP=OLD
//SYSIN DD *
AUTH 12104/SECRET
ALTER PROCESS INDATA(ATMSIII) OUTDATA(SCRIPT) ENTRY(0)
ALTER PROCESS INDATA(ATMSIII) OUTDATA(SCRIPT) -
ENTRY(DSMACAIP)
/*

// JOB ALTER                                     for VSE
// OPTION LISTX,NODUMP
// DLBL DSMPDIR,'DSMLIB.DIRECTRY',,VSAM
// EXTENT ,DSMVOL
// DLBL DSMPDLIB,'DSMLIB.SOURCE',,VSAM
// EXTENT ,DSMVOL
// EXEC DSMSPEXC,SIZE=AUTO
AUTH 12104/SECRET
ALTER PROCESS INDATA(ATMSIII) OUTDATA(SCRIPT) -
ENTRY(0)
ALTER PROCESS INDATA(ATMSIII) OUTDATA(SCRIPT) -
ENTRY(DSMACAIP)
/*
/&
```

Figure 26. Examples of Altering an Attribute Profile

DEFINING USER MAPS FOR EXTERNAL SYSTEMS

An administrator can authorize a user outside the Document Library Facility as a valid library user. To accomplish this authorization use the mapping procedure explained in Chapter 5 under DEFINE MAP.

DEFINING CLASSES OF DATA

An administrator can assign a class associated with a cluster to documents as a way of using library space more efficiently. See Chapter 5 under DEFINE CLASS for an explanation of class and the syntax for defining it.

PART 4. ADDITIONAL USES OF THE DOCUMENT LIBRARY FACILITY

Part 4 of the Document Library Facility Guide provides information about additional capabilities of DLF through the following chapters:

- Chapter 8, "Using DLF to Call Processors" on page 215 - discusses DLF input to a processor, processor activities, how a processor returns to DLF, and the calling sequence between DLF a processor.
- Chapter 9, "Using DLF as a Subroutine" on page 221 - explains how to call DLF from an application program and how to format input data supplied by another program using the SCRIPT FROM(*) command.
- Chapter 10, "Writing User Exit Routines" on page 225 - describes the uses for exit routines, calling sequence between DLF and a user exit routine, flags, return codes, messages and the exits for specific commands.

CHAPTER 8. USING DLF TO CALL PROCESSORS

A processor is a user-written or IBM-supplied program that can be called to process documents during the processing of four DLF commands: IMPORT, EXPORT, READ, and SCRIPT. Processors are defined to DLF by the DEFINE PROCESS command.

When a processor is called by DLF, it is passed the next sequential record from the input document. The processor can perform conversion or modification as necessary and even generate SCRIPT/VS text-formatting controls or other special conversions that are acceptable.

A processor can be written to generate records from a source in addition to DLF and insert them where you have indicated. You do this by setting the suppress-input flag in the processor interface. DLF then continues calling the processor without reading any more input. For example, you could write a processor so that it inserts a figure at a certain point in the program. The figure would be obtained from records external to DLF.

Processors must be stored in an Authorized Program Facility load module library for OS/VS2 or in a core image library for VSE. A processor does not need to be reentrant, but it must be relocatable and serially reusable and meet Authorized Program Facility requirements if it is used in OS/VS2.

The Document Composition Facility has a program that can be used with DLF to convert ATMS-II and ATMS-III documents to documents that can be formatted by SCRIPT/VS.

INPUT TO PROCESSOR

When control is passed to a processor, one data record followed by two bytes containing X'FFFF' are placed in the communication buffer. This is repeated for each call of the processor until there are no more records or an error occurs. A flag is set to tell the processor to initialize its work area.

The data and source type names for the input document and the data and source types for the output desired from the processor are supplied by DLF. The attribute names can be the same or different from each other. The processor should check these fields in the parameter area and verify that the correct attribute names for its processing are specified. See the table under DEFINE PROCESS (notes) in Chapter 5 for additional information on data attribute pairs.

The IMPORT command places the INDATA and INSOURCE names in the input parameter fields and the names specified in the DATA and SOURCE keywords in the output parameter fields when a processor is called.

The EXPORT command places the document data type and source in the input parameter fields and the OUTDATA and OUTSOURCE names in the output parameter fields when a processor is called.

The READ command fills the input and output parameter fields in the same way as the EXPORT command.

The SCRIPT command fills the input parameter fields in the same way as the EXPORT command. The word SCRIPT is placed in the output data type and source type fields.

As long as the return code is 4 or less and the end-of-file flag from the processor is not on, DLF continues to call the processor with the next sequential record to be processed.

When an end-of-file condition occurs on the input document, the first two bytes of the communication buffer are set to X'FFFF' and the EOF-on-source flag is set on by DLF before calling the processor.

PROCESSOR ACTIVITIES

The processor must check the initialization flag and perform any initialization necessary. During the processing of the SCRIPT command, the initialization flag can be set on at any time; consequently, the processor must be able to start over.

The processor can modify any or all of the contents of the communication buffer. There is also a work area available. The space for these two areas is provided by DLF, and the length of these areas is specified in the parameter area.

The work area is preserved by DLF between calls to the processor. This allows the processor to save data, registers, or any other type of information required between calls.

The processor can return anything in the communication buffer as long as it is in SLE format. (See "The Communication Buffer" on page 219 for further information.)

HOW A PROCESSOR RETURNS TO DLF

When the processor has finished with an input record, it may return with none, one, or more than one record in the communication buffer to DLF. Each record to be returned must be placed in the communication buffer. The number X'FFFF' must follow the last record.

When the processor finds the EOF-on-source flag on, it should place any remaining records in the communication buffer. DLF continues to return to the processor until the EOF-from-processor flag is on. When there are no records remaining, the EOF-from-processor flag must be turned on and the number X'FFFF' must be put into the next available location in the communication buffer.

Each time the processor returns control to DLF, the return and reason codes and flag settings must be set, as appropriate, in the parameter area. A message can be printed for any return by turning on the print-processor-message flag and setting the message address field appropriately.

Some examples of return and reason codes that a processor can return are shown in Figure 27.

CALLING SEQUENCE BETWEEN DLF AND A PROCESSOR

REGISTER CONVENTIONS

When a processor is called by DLF, the following register conventions apply:

Register 1 points to a list of three addresses. These three addresses in turn point to three additional addresses:

- The address of a parameter area
- The address of a communication buffer
- The address of a work area.

Register 13 points to a 72-byte register save area.

Register 14 points to a return address in DLF.

<u>Return Code</u>	<u>Reason Code</u>	<u>Meaning</u>
0	0	Normal return - Communication buffer may contain records. Processor expects to be called again if EOF-from-processor flag not set.
4	0	Warning return - Communication buffer may contain records. Processor expects to be called again if EOF-from-processor flag not set.
8	0	EOF-from-processor - Communication buffer may contain records. The library program will not call the processor again for return codes greater than 4.
8	4	Wrong input provided to the processor.
8	8	Wrong output expected from processor.
12	0	Processor conversion error.

Figure 27. Processor Return and Reason Codes

Register 15 points to the entry point of the processor.

When a processor returns to DLF, these registers should contain the same information.

PARAMETER AREA

<u>Offset (Length)</u>	<u>Description</u>
0(2)	Flags
1... ..	EOF on source
.1... ..	EOF from processor
..1... ..	Initialize processor
...1... ..	Print processor message
.... 1... ..	processor parameter present
.... .1... ..	suppress input data
.... ..xx	(reserved)
xxxx xxxx	(reserved)
2(2)	Reserved
4(4)	Message buffer address
8(8)	Output data attribute name (EBCDIC)
16(8)	Input data attribute name (EBCDIC)
24(8)	Class attribute (EBCDIC)
32(8)	Input source attribute (EBCDIC)
40(8)	Output source attribute (EBCDIC)
48(64)	Processor parameters
112(2)	Communication buffer length (control interval size)
114(2)	Work area length (binary value of 4 096)
116(4)	Return code (binary)
120(4)	Reason code (binary)

The flags set by DLF are:

<u>Flag</u>	<u>Meaning</u>
EOF on Source	0 - Not end of input 1 - The last record has been passed to the processor
Initialize Processor	0 - Not an initialize call 1 - The processor must initialize
Processor Parameter Present	0 - Parameter is missing 1 - Parameter is present

The flags set by the processor are:

<u>Flag</u>	<u>Meaning</u>
EOF from Processor	0 - Not end of processing 1 - The last record has already been processed or the communication buffer may contain records
Print Processor Message	0 - No message 1 - A message is to be printed; the message buffer address field points to the message area. Communication buffer may contain records
Suppress Input Data	0 - Read another record from an external data set or I/O module before returning to processor 1 - Return to processor without reading another record from an external data set or I/O module

Note: The Suppress Input bit is valid for input functions only. It is not valid for output functions such as SCRIPT and EXPORT.

MESSAGES

The processor can request that a message be printed by DLF. To do this the processor must:

1. Set the print-processor-message bit.
2. Put the address of a 125-byte message area in the message-buffer-address field of the parameter area and return control to DLF.

The format of the message area is:

<u>Offset (Length)</u>	<u>Name</u>	<u>Description</u>
0(2)	MSGLENG	Low-order 12 bits contain the length of the text message plus 5 bytes. Maximum value is 125.
2(2)		Reserved; must be zero.
4(1)	MSGASA	ASA printer code; usually a blank.
5	MSGTEXT	Text of message (maximum of 120 characters).

THE COMMUNICATION BUFFER

Each record that is put into the communication buffer must be in the source line entry (SLE) format:

<u>Offset (Length)</u>	<u>Name</u>	<u>Description</u>
0(2)	SLELENG	The record length; first 4 bits are zero; low-order 12 bits give the length. Maximum length is 4 058.
2	SLETEXT	The record.

The length field (SLELENG) contains the count of the data in SLETEXT.

The number X'FFFF' in the length field (SLELENG) signals the end of the data records. When this occurs as the first two bytes in the communication buffer, the buffer is considered empty.

EXAMPLES

The following examples show commands that define processors to DLF and then use those processors.

```
AUTH 300/ADMPASS
DEFINE PROCESS INDATA(ATMS) OUTDATA(SCRIPT) -
ENTRY(DSMACAIP)
DEFINE PROCESS INDATA(SCRIPT3) OUTDATA(SCRIPT) -
ENTRY(CONVERT)
DEFINE PROCESS INDATA(F3800) OUTDATA(PRINT) -
ENTRY(PRINTED)
```

```
AUTH 3001/GENUSER
IMPORT ATMSDOC INDATA(ATMS) DATA(SCRIPT) -
INSOURCE(ATMS) FROM(OUTSIDE)
SCRIPT SCRIPT3DOC DATA(SCRIPT3)(NOPROF FILE)
EXPORT FORMATTED DATA(F3800) OUTDATA(PRINT) -
TO(SYSPRINT)
```

Notes:

1. The IMPORT, EXPORT, READ, and SCRIPT commands invoke implied processors if they are specified with input and output data types even though the PROCESS keyword is not specified.
2. The conversion routine, DSMACAIP, is called when ATMSDOC is imported. The input/output data types of INDATA(ATMS) and DATA(SCRIPT) invoke a processor that was defined with the same input/output data type combination and an entry name of DSMACAIP.

3. SCRIPT3DOC is formatted invoking a processor whose entry name is CONVERT. The SCRIPT command was invoked with an input data type of SCRIPT3. The output data type of the SCRIPT command is always SCRIPT. Therefore, this combination of input/output data types invokes a processor that was defined with the same input/output data type combination and an entry name of CONVERT.
4. A formatted file is exported to SYSPRINT, and it calls the processor that was defined with an input data type of F3800, an output data type of PRINT, and an entry name of PRINTED.

CHAPTER 9. USING DLF AS A SUBROUTINE

DLF can be called as a subroutine by a user-written application program. This program must have been linked as an authorized program and reside in an authorized library. In addition, a second application program can supply data to be processed by SCRIPT/VS.

If DLF is used as a subroutine, commands are passed as parameters. The following commands can also be used in addition to those used in normal DLF processing. (See Chapter 3 and Chapter 5 for discussion of commands.)

Note: When a user-written application ends its use of DLF, be sure to make a final call to DLF issuing a CLOSE or CLOSE ALL command. This allows DLF to close all of the data sets that are open and issue final messages.

READ - to read a document sequentially from a library.

CLOSE - to close any documents used.

SCRIPT FROM(x) - indicates that another application program is to be called to furnish data records that are to be processed by SCRIPT/VS rather than obtaining data from the document library. The name of the program to be called is in the parameter area.

CALLING DLF FROM AN APPLICATION PROGRAM

The registers must be set as follows on entry to DLF:

- Register 1 must point to a list of six addresses, each of which points to an element of a parameter list created by and located within the calling program.

Note: The expanded parameter list element pointed to by the fourth address in the list has room for DLF to return a reason code and a version number as well as the return code. The reason code is the DLF message number of the first error message that would have resulted had the request been processed by DLF in a batch environment. The version number is that of the version of the document successfully stored (set only for IMPORT requests).

- Register 13 must point to a 72-byte register save area.
- Register 14 contains the application program's return address.
- Register 15 contains the DSMSPEXC entry point address.

Note: At the completion of processing, DLF places a return code into the return code field of the parameter area and into Register 15.

- For VSE, the calling program must also set the first byte of the user area in the VSE communication area to the character C before passing control to DSMSPEXC. (See the VSE/Advanced Functions Macro User's Guide and VSE/Advanced Functions Macro Reference for detailed information.)

The list of addresses contains pointers to:

1. First parameter - An EBCDIC string containing the LIST or NOLIST (LISTX or NOLISTX for VSE) keywords. LIST and LISTX request that detailed information about the processed documents be printed on the output device DSMLIST for OS/VS2 and SYSLST for VSE. This is only examined if an ALTER, COPY,

DEFINE, IMPORT, or EXPORT command is specified as the third parameter in this list.

Note: This parameter is only used on the first call to DLF. On all subsequent calls in the same job step, it is only used (and required) as a placeholder.

2. Second parameter - A halfword containing the length in binary of the command to be processed.
3. Third parameter - An EBCDIC character string representing a library command.

Notes:

- a. Only one DLF command can be specified for each call DLF.
 - b. The commands are variable length.
 - c. The command is alphameric characters. The maximum length of text is 1 024 bytes.
 - d. The rules governing separators, comments, and abbreviations apply.
4. Fourth parameter - This parameter contains either one or three fullwords depending on the value passed to DLF in the first fullword:
 - If the value passed to DLF in the first (or only) fullword is not the 4-character string 10F3 (X'F1D6C6F3'), the parameter is one fullword that will contain the highest return code from DLF after the command has been processed. All nonzero return codes from DLF are in multiples of 4:

Return Code	Meaning
0	Normal return
4	Warning message or EOF for READ
8	Command not completed successfully
12	Command not completed successfully
16	Severe error, DLF is stopping

- If the value passed to DLF is the 4-character string 10F3 indicating that this is the first of three fullwords:
 - The first fullword is the highest return code from DLF as described above.
 - The second fullword will contain the error reason code. This is the message number of the first DLF message issued for this command.
 - The third fullword will contain the version number of the imported document if the command was IMPORT. It will not be set for any other DLF command.

Note: The character string 10F3 must be reinitialized for each call to DLF.

5. Fifth parameter - this parameter changes according to the command.
 - With CLOSE, there is no fifth parameter.
 - With READ, it specifies the halfword length of a data area whose address is in the sixth parameter.
 - With SCRIPT, if the third parameter contains the SCRIPT FROM(*) command, this parameter should contain the name of a load module from which DLF can get records.

6. Sixth parameter - The address of an area that will contain the data in SLE format brought in by a READ operation.

Note: This parameter is required only for the READ command.

Example of Use

The following example shows a call written in a high-level language that requires a data area for a READ command.

```
CALL DSMSPEXC(LISTPARM,COMDLLEN,COMMAND,RTNCODE,AREALEN,AREA)
```

The equivalent call in assembler language might look like this:

```
PLIST    BAL 1,LOADNGO
          DC  A(LISTPARM)
          DC  A(COMDLLEN)
          DC  A(COMMAND)
          DC  A(RTNCODE)
          DC  A(AREALEN)
          DC  A(AREA)
LOADNGO  DS  0H
          L   15,=V(DSMSPEXC)
          BALR 14,15
```

Notes:

1. The parameters AREALEN and AREA are only required by a READ command.
2. The address in RTNCODE points to a one or three fullword field as described in item 4 above. The three fullword list in assembler language may look like this:

```
RTNCODE   DS  F
          ORG RTNCODE
          DC  CL4'10F3'    1-OF-3
RSNCODE    DS  F
VERSION    DS  F
```

The constant, 10F3, must be reinitialized before each call to DLF.

SCRIPT FROM(*) - INPUT DATA FROM ANOTHER PROGRAM

If DLF is called as a subroutine and data is to be supplied by a second program for SCRIPT/VS processing, another interface with DLF is used. DLF loads the second program, an I/O load module whose name is the fifth parameter of the parameter list used by the first application program.

DLF uses the operating system's loader to load the I/O load module. The I/O load module must reside in an Authorized Program Facility load library in the OS/VS2 environment or in a core-image library in the VSE environment.

When DLF calls the I/O load module, the interface is as follows:

1. Register 1 points to a parameter list with these contents:
 - a. The address of a fullword that contains the communication buffer pointer
 - b. The address of a fullword that contains the communication buffer length
 - c. The address of a fullword that contains the return code
 - d. The address of a fullword that contains the reason code.
2. Register 13 points to a 72-byte register save area.

3. Register 14 points to a return address in DLF.
4. Register 15 points to the entry point of the application I/O module.

When the I/O module returns to DLF, the return codes required are:

<u>Return Code</u>	<u>Meaning</u>
0	Normal return
4	EOF from I/O module
8	I/O module unable to supply records

One record is to be placed in the communication buffer by the I/O load module for each call. Each record can vary in length, but must not exceed the maximum length of 4 058 bytes. The records in the communication buffer must be in the source line entry (SLE) format as described under the topic "The Communication Buffer" on page 219.

Example of Use

The following example shows a call written from an application program to DLF and specifies that another routine (named in the ENTNAME parameter) will supply records to be formatted by SCRIPT/VS.

CALL DSMSPEXC(LISTPARM,COMDLN,COMMAND,RTNCODE,ENTNAME)

The equivalent call in assembler language might look like this:

```

          BAL 1,LOADNGO
PLIST    DC A(LISTPARM)
          DC A(COMDLN)
          DC A(COMMAND)
          DC A(RTNCODE)
          DC A(ENTNAME)
LOADNGO  DS 0H
          L 15,=V(DSMSPEXC)
          BALR 14,15

```

CHAPTER 10. WRITING USER EXIT ROUTINES

You can write user exit routines as described below. The user exit module, DSMSPUXI, supplied by IBM should reside in an authorized program library.

The following commands activate user exits:

- ARCHIVE
- DEFINE MAP
- DEFINE USER
- ALTER MAP
- ALTER USER
- EXPORT
- IMPORT
- PURGE DOCUMENT
- PURGE MAP
- PURGE USER
- RETRIEVE.

USES FOR EXIT ROUTINES

Exit routines can access directory data set records; this enables them to:

- Halt the execution of a command
- Insert information about a user or document
- Keep track of certain activity and record it on an external data set.

TAKING AN EXIT

When an exit is taken, the following records are accessible:

- The user mapping record (UMR), which shows how an external user, such as an ATMS user, is mapped to a library user number. The UMR is shown in Figure 28 on page 226.
- The document entry record (DER), which is explained in Chapter 1, "Document Library Facility Concepts and Conventions." The DER is shown in Figure 29 on page 227.
- The user profile record (UPR), which is described in Chapter 1, "Document Library Facility Concepts and Conventions." The UPR is shown in Figure 30 on page 228.

Two fields can be changed by an exit routine: DERAPPL, which is found in the DER, and UPRAPPL, which is found in the UPR. In the DERAPPL, you might keep track of the number of times a document has been exported. In the UPRAPPL field, you might insert information about a user's profile.

Each time an exit is activated, one or the other of these fields can be changed, depending on the command being executed. The UMR cannot be changed. The section "Exits at Each Command" in Figure 31 on page 229 explains which field is available at the execution of each command.

Offset (Length)	Name	Description
0 (33)	UMR	User Mapping Record
0 (30)	UMRKEY	UMR key
0 (16)	UMRGKEY	UMR generic key
0 (3)	UMR3XFF	Three bytes X'FFFFFF'
3 (1)	UMRCODE	UMR code (X'E0')
4 (2)	UMRHOST	Host system identifier
6 (10)	UMREXID	External user id
16 (14)	*	Reserved
30 (3)	UMRLIBNO	Library user number

Figure 28. The Format of the User Mapping Record (UMR)

Offset	(Length)	Name	Description
0	(241)	DER	Document Entry Record
0	(30)	DERKEY	DER key
0	(30)	DERGKEY	DER generic key
0	(28)	DERNVKEY	DER key less the version field
0	(20)	DERNDKEY	DER key without data/version fields
0	(3)	DERLIBNO	Library number in which document is stored
3	(1)	DERCODE	DER code field (X'70')
4	(16)	DERMEMNM	Document name
20	(8)	DERDATA	Document data type
28	(2)	DERVERS	Document version number
30	(1)	DERTYPE	Type flags
	x... ..	*	Reserved
	.1.. ..	DERPUB	Public library
	..1.	DERPROJ	Project library
	...1	DERPRIV	Private library (on = controlled, off = uncontrolled)
 xxxx	*	Reserved
31	(1)	DERFLAGS	Flags
	1... ..	DEROPEN	Last control interval may not have chain pointer
	.x..	DERPP	Password - removed for confidentiality
	..1.	DERVP	Document has versions
	...1	DERLOCK	Document is locked
 1...	DERSHR	Document is shared
1..	DERARCH	Document is archived
1	DERTODSN	Document archived to dynamically allocated data set
1	DEROPER	Document is archived with DERNOTE filled in by OPERATOR
32	(2)	*	Reserved
34	(8)	*	Reserved for library program use
42	(8)	DERSWTS	Save/write timestamp
50	(8)	DERATS	Archive timestamp
58	(8)	DESRCE	Source
66	(8)	DERCLASS	Class
74	(8)	DERCLNAM	Cluster - name of cluster in which document is stored
82	(3)	*	Reserved
85	(3)	DEROWNR	User number of owner of document in public or project library
88	(2)	DERFLSIL	Text area size within CIs of this document
90	(6)	*	Reserved
96	(4)	DERCFLSI	Current size of document
100	(6)	DERSLR	Logical records
100	(4)	DERNLR	Number of logical records in the document
104	(2)	DERLNLR	Number of logical records in the last control interval
106	(2)	DERNCI	Number of control intervals used
108	(8)	DERRBAS	RBA pointers
108	(4)	DERFRBA	First control interval RBA
112	(4)	DERLRBA	Last control interval RBA
116	(75)	DERAPPL	Field for user exit information
116	(75)	DERNOTE	ARCHIVE information (overlays DERAPPL if document is archived)
191	(50)	DERINFO	Document information

Figure 29. The Format of the Document Entry Record (DER)

Offset (Length)	Name	Description
0 (108)	UPR	User Profile Record
0 (30)	UPRKEY	UPR key
0 (4)	UPRGKEY	UPR generic key
0 (3)	UPRLIBNO	User library number
3 (1)	UPRCODE	UPR code (X'40')
4 (26)	*	Reserved
30 (8)	UPRPSWD	User password (removed for confidentiality)
38 (1)	UPRTYPE	Library type flags
1... ..	UPRPUB	Public library
.1... ..	UPRPROJ	Project library
..1... ..	UPRPRIV	Private library
...1... ..	UPRPROJC	Controlled library (on = controlled off = uncontrolled)
.... XXXX	*	Reserved
39 (1)	UPRFLAGS	Flags
1... ..	UPRNLOG	User space usage may be incorrect
.x... ..	UPRPP	Password present flag
..1... ..	UPRLOCK	User locked from DLF
...1... ..	UPRSYSAD	User is system administrator
.... XXXX	*	Reserved
40 (8)	UPRDATA	Default data type
48 (1)	*	Reserved
49 (3)	UPRPJLIB	Project library number
52 (4)	*	Reserved
56 (8)	UPRCLASS	Default class
64 (8)	UPRSOURC	Default source
72 (4)	UPRDPCNT	Number of documents in all user's libraries
76 (4)	UPRDSLIM	DASD space limitation (1000s of bytes)
80 (4)	UPRDSLIB	Space in use in user's own library
84 (4)	UPRDPLIB	Space in use in project/public libraries
88 (20)	UPRAPPL	Field for user exit information

Figure 30. The Format of the User Profile Record (UPR)

CALLING SEQUENCE BETWEEN DLF AND A USER EXIT ROUTINE

When writing your exit routine, you should replace the load module DSMSPUXI, which IBM supplies as a sample routine, with your user exit load module. The user exit load module should reside in an authorized program library.

REGISTER CONVENTIONS

When an exit routine is called by DLF, the following register conventions apply:

- Register 1 points to a word that points to the address of a parameter area.
- Register 13 points to a 72-byte save area.
- Register 14 points to a return address in DLF.
- Register 15 points to the entry address of DSMSPUXI.

When an exit routine returns control to DLF, these registers should contain the same information.

Offset (Length)	Name	Description
0 (402)	EXI	User exit interface parameter list
0 (2)	EXIRETRN	User exit return code
2 (2)	EXIRESN	User exit reason code
4 (125)	EXIMSG	User message area
4 (2)	EXIMSGLN	Message unit length
6 (2)	*	Reserved
8 (1)	EXIMSGCL	Message carriage control character
9 (120)	EXIMSGTX	Message text
129 (1)	EXIFLAGS	Flags
	EXIMSGON	1... .. Print user message
	EXIDERIN	.1... .. DER is present
	EXIUPRIN	.1... .. UPR is present
	EXIUMRIN	...1... .. UMR is present
	EXIAPPLX 1... Application field changed by user exit
	*xxx (reserved)
130 (8)	EXICOMND	Command name (EBCDIC)
138 (8)	EXICMOBJ	Command name second word (EBCDIC)
146 (256)	EXIDER	Copy of the DER
146 (120)	EXIUPR	Copy of the UPR
266 (60)	EXIUMR	Copy of the UMR

Figure 31. The Parameter Area of the User Exit Routine

PARAMETER AREA

The parameter area of the supplied module, DSMSPUXI, is shown in Figure 31.

Note: Either the DER or the UPR is present, never both simultaneously. If an affiliated UMR exists, it always follows the UPR.

FLAGS

The flags set by DLF are:

Flag	Meaning
DER is present	0 - No DER in buffer 1 - DER is present
UPR is present	0 - No UPR in buffer 1 - UPR is present
UMR is present	0 - No UMR in buffer 1 - UMR is present

The flags set by the exit routine are:

Flag	Meaning
Application field changed	0 - No user information present 1 - User has entered information
Print user message	0 - No message to be printed 1 - print message

Note: Each time your routine returns control to DLF, flags must be set as appropriate.

RETURN CODES

Return codes are issued every time your routine returns control to DLF.

A return code of 16 or greater for any command halts the processing by DLF.

Each time the exit routine returns control to DLF, return codes must be set.

MESSAGES

Your routine can request that a message be printed by the DLF. To do this, you must:

1. Set the print message bit (EXIMSGON) on in the parameter area.
2. Assign text length and control character (length is always text plus five).
3. Return control to DLF.

The format for the message area is:

<u>Offset (Length)</u>	<u>Name</u>	<u>Description</u>
0(2)	E XIMSGLN	Length of message
2(2)	(unused)	Reserved
4(1)	E XIMSGCL	Carriage control character (ASA)
5(120)	E XIMSGTX	Text of message (maximum of 120 characters)

The following example shows a user routine that sets return and reason codes in a DEFINE MAP command. The output resulting from this command is shown at the end of the example.

<u>Code</u>		<u>Comments</u>
L	R1,0(R1)	Load register 1 with pointer to parameter list
USING	EXI,R1	Use register 1 as base for parameter list
LH	R3,=H'0'	Load register 3 with 0
STH	R3,EXIRETRN	Put in 0 return code
STH	R3,EXIRESN	Put in 0 reason code
OI	EXIFLAGS,EXIMSGON	Turn on message request bit
MVI	EXIMSGTX,C' '	Move a blank to technical message area
MVC	EXIMSGTX+1(119),EXIMSGTX	Blank out entire message field
MVI	EXIMSGCL,C' '	Blank carriage control character
MVC	EXIMSGLN(2),MSGLN	Move in message length
MVC	EXIMSGTX(20),MESSAGE	Move in message
.		
.		
MESSAGE	DC CL20'NEW USER MAP DEFINED'	Message for exit routine
MSGLN	DC H'25'	Length of message plus 5

Input

AUTH 4060
 DEFINE MAP HOST('MP') ID('NAMEID') USER(4061)

Output

AUTH 4060
 DSMEXC091I RETURN CODE = 0000.
 DEFINE MAP HOST('MP') ID('NAMEID') USER(4061)
 NEW USER MAP DEFINED

HOST ID	EXTERNAL USER ID	USER NUMBER
MP	NAMEID	4061

 DSMEXC091I RETURN CODE = 0000

EXITS AT EACH COMMAND

Below are descriptions of changes you can make when user exits are activated for specific commands.

ARCHIVE USER EXIT

This exit is taken just before the DER for the document is written in the directory data set. You can change the DERAPPL field. All changes become part of the new DER. If a return code of 8 or greater is received, no archive is performed. If a return code of 12 or greater is received, the command is halted.

DEFINE MAP AND ALTER MAP USER EXITS

The DEFINE MAP or ALTER MAP exits are taken after the new or altered UMR is constructed but before the directory data set is changed. Both the new/altered UMR and its associated UPR are accessible, and you can alter the UPRAPPL field.

DEFINE USER AND ALTER USER USER EXITS

The exits are taken prior to writing the new UPR to the directory data set. The new UPR is accessible, and UPRAPPL can be changed.

EXPORT USER EXIT

This exit is taken after the DER is read but before the document is exported. Your routine can read the DER and change the DERAPPL field only if the output document library JCL is specified and the input and output document libraries are the same. No export is performed if a return code greater than 4 is received.

IMPORT USER EXIT

The exit is taken when the DER is completed and after all control intervals are written to the ESDS, but before the new DER is written in the directory data set. If the return code is 8 or greater, the current document is not imported. The DERAPPL field can be changed.

PURGE DOCUMENT USER EXIT

This exit is taken before the document is purged. The DERAPPL field can be changed; however, no changes are effective if the command is executed. If the document has been archived, the DERNOTE field overlays the DERAPPL field, showing where the archived document is stored.

PURGE MAP USER EXIT

When this exit is taken, you may use information in the UMR and its associated UPR to change the UPRAPPL field. However, no changes will be effective if the command is executed. A return code of 8 or greater ends the PURGE MAP command without purging the UMR.

PURGE USER USER EXIT

This exit is taken before the user is purged and before purging or changing the ownership of any documents associated with that user. A copy of the user profile record (UPR) is made available at the exit; but no changes can be made to the UPR. A return code of 8 or greater terminates the PURGE USER command.

RETRIEVE USER EXIT

This exit is taken each time a DER is read from the archive data set. If you change the DER, it becomes part of the DER written in the directory data set. A return code of 8 or greater means the current document will not be retrieved. If the ALL option is in effect, the rest of the documents in the list will be examined individually, and the ones with a return code of less than 8 will be retrieved. A return code of 12 or greater terminates the RETRIEVE command. Documents retrieved before the return code of 12 is received will remain in the document library.

Notes:

1. Changing the DERAPPL or UPRAPPL field does not affect execution of the command. It is merely a way for you to insert up to 75 characters of information about the document or the user's library for your own use.
2. Control of a command's execution is handled in the message area of DSMSPUXI where you may establish return codes for halting execution or for requesting dumps or accounting information. Refer to the section "Return Codes," in this chapter, for more details.

APPENDIX A. CONVERTING FROM RELEASE 1 TO RELEASE 3

Although there should be no need for conversion from Release 2 to Release 3, those users who still have Release 1 will need to convert to Release 3. This process is described below.

CONVERTING A DOCUMENT LIBRARY

CONVERSION PROCESS

The conversion from Release 1 to Release 3 requires that certain steps be followed by both administrators and library users.

Administrators

Define the Release 3 library as described in Chapter 6, "Setting Up the Document Library." The basic steps are listed below:

1. Define the VSAM data sets using the AMS command.
2. Define the default 4K entry-sequenced data set (ESDS) and the directory data set (KSDS).
3. Define other ESDSs as necessary for the current level of activity. More can be defined at any time.
4. Use the Access Method Services command, DEFINE CLUSTER, to allocate the data sets.
5. Define the library itself with the library commands, DEFINE SYSTEM, DEFINE USER, and DEFINE CLUSTER.

General Users

When the Release 3 Library is established, you (as a general user) can move documents into it in the following way:

1. Use the EXPORT command to move documents from the Release 1 library to external data sets. The Release 1 DLF must be used.
2. Operating under control of the Release 3 program, use the IMPORT command to move documents from the external data sets to the Release 3 library. The IMPORT command in Release 3 is similar to that in Release 1. (See "IMPORT" on page 58 for current syntax and new options.)
3. This technique allows for the formatting of documents using SCRIPT/VS in both Release 1 and Release 3. The documents can be formatted from the external data sets with both releases. (See "SCRIPT" on page 94 for details.)

SEQUENCING DATA

Release 3 does not support the library control of sequence numbers of data records. Data editors are the correct environment in which to control sequence numbers. The numbers can still exist in the data stored in the library but are not treated differently than any other data.

DATE KEYWORD

The DATE function on COPY has been moved to the ARCHIVE ALL and RETRIEVE ALL commands. DATE is still valid during a LIST DOCUMENT function.

APPENDIX B. JCL STATEMENTS, DYNAMIC ALLOCATION, AND DATA SET CHARACTERISTICS

JCL REQUIREMENTS

DLF is invoked through the following job control language (JCL):

- For OS/VS2,

```
// EXEC PGM=DSMSPEXC[,PARM='{LIST|NOLIST}']
```

If the parameter field is not specified, NOLIST is assumed.

- For VSE,

```
// OPTION {LISTX|NOLISTX}  
// EXEC PGM=DSMSPEXC[,SIZE={AUTO|nK}]
```

If the OPTION card is not supplied, NOLISTX is assumed.

If Advanced Function Printer output from the SCRIPT command is being produced with GML in the document source:

- Specify SIZE as at least 768K.
- Allocate 2 048K to the partition in which the job is processing. A 2 048K allocation will accommodate VSE/VSAM for the document, the GML library, and the font library data for the IBM 4250 printer or IBM 3820 Page Printer data.

Figure 32 lists the ddnames (for OS/VS2) or DLBL and TLBL names (for VSE) used by DLF, describes their function, and lists the commands that make use of them.

In addition to the given names, choose ddnames for the data sets you allocate. That is, for every DEFINE CLUSTER you issue, you must create a corresponding data set name.

OS/VS2 ddname	VSE DLBL,TLBL	Description	Applies to:
DSMINDIR	DSMIDIR	The default VSAM KSDS that contains the profiles and directories that are to be read as input. It can be the same data set as the DSMPTDIR statement (for OS/VS2) or DSMPDIR statement (for VSE).	ACCOUNT AUTH COPY EXPORT LIST READ SCRIPT
DSMINLIB	DSMILIB	The default VSAM ESDS that contains the documents (in 4096 byte control intervals) that are to be read as input. It can point to the same data set as the DSMPTLIB statement (for OS/VS2) or DSMPLIB (for VSE).	ACCOUNT AUTH COPY EXPORT LIST READ SCRIPT

Figure 32 (Part 1 of 3). JCL Data Set Names

OS/VS2 ddname	VSE DLBL, TLBL	Description	Applies to:
DSMPDIR	DSMPDIR	The default VSAM KSDS that contains the profiles and directories that are to be listed, modified, added, or deleted.	ALTER ARCHIVE AUTH COPY DEFINE IMPORT LIST PASSWORD PROTECT PURGE RETRIEVE
DSMPTLIB	DSMPLIB	The default VSAM ESDS to which new documents are added or from which purged documents are removed. Its status may also be listed.	ALTER ARCHIVE AUTH COPY DEFINE IMPORT LIST PASSWORD PROTECT PURGE RETRIEVE
SYSIN	SYSRDR (Logical Unit)	The data set that contains the commands. It can be any data set and can be a member of a partitioned data set (for OS/VS2). The records must be 80-byte card-images and may be blocked in OS/VS2.	all commands
DSMLIST	SYSLIST (Logical Unit)	The required data set for listing commands, messages, and document names. It can also be used as a sequential output data set for various commands.	all commands
DSMTERMI	DSMITRM	The data set that contains data that will be used in place of terminal input. This is needed only if the SCRIPT/VS control words .TE or .RV are used.	SCRIPT
DSMTERMO	DSMOTRM	A data set that will be used in place of terminal output. It is required only if the SCRIPT/VS control word .TY is used.	SCRIPT
DSMUTTOC	DSMUTOC	A SCRIPT/VS-generated file into which the table of contents entries are written when SCRIPT/VS encounters the .PT (Put Table of Contents) or .H0 - .H6 (Head Level) control words.	SCRIPT
DSMUTMSG	DSMUMSG	The SCRIPT/VS-generated file into which messages are written when the MESSAGE(DELAY) option of the DCF SCRIPT command is specified.	SCRIPT
DSMUTCTF	DSMUCTF	The SCRIPT/VS-generated file into which STAIRS/VS CTF output is written when SCRIPT/VS encounters the CTF option.	SCRIPT
DSMUTWTF	DSMUWTF	A SCRIPT/VS-generated file into which input lines may be placed dynamically when SCRIPT/VS encounters the .WF (Write to File) control word.	SCRIPT

Figure 32 (Part 2 of 3). JCL Data Set Names

OS/VS2 ddname	VSE DLBL, TLBL	Description	Applies to:
anyname	anyname	A data set that can be referenced by the listed commands.	ACCOUNT ARCHIVE COPY EXPORT IMPORT LIST RETRIEVE SCRIPT

Figure 32 (Part 3 of 3). JCL Data Set Names

DATA SET NAMING FOR DYNAMIC ALLOCATION

When dynamic allocation is used in the OS/VS2 environment, data set names specified in the TO and FROM keywords are qualified in the following manner:

- If the name specified for dsname is coded in quotation marks, the name is considered fully qualified, and no further qualifier is required.
- If the name specified for dsname is not coded in quotation marks, the name is qualified as follows:
 - A prefix qualifier is added that starts with the letter U followed by the library number specified in the most recent AUTH. command.
 - A suffix qualifier is added that is an alphabetic name, dependent upon the DLF command. Refer to the description of the specific command for the suffix qualifier.

If the name specified for dsname is followed by a partitioned data set member name enclosed in parentheses, the member is read or written as a sequential data set. The member name and the parentheses must be within the quotation marks if quotation marks are used.

For example, the following illustrates how data set names are qualified if user 12 345 issued the LIST command with a TO(DSN(dsname)) operand:

<u>Specified dsname</u>	<u>Actual dsname</u>
A	'U12345.A.LIST'
A(B)	'U12345.A.LIST(B)'
DOC.LIST	'U12345.DOC.LIST.LIST'
'D001.BOOK.FILE'	'D001.BOOK.FILE'

Note: When dynamic allocation is used, the data set must be either cataloged or defined by JCL in the DLF step.

DEFAULT DATA SET CHARACTERISTICS

Figure 33 on page 240 and Figure 34 on page 241 show the default record length and block size for each of the data sets used by DLF and by SCRIPT/VS when called by DLF.

OS/VS2	Default ddname	RECFM	LRECL	BLKSIZE
ACCOUNT FROM		VB	16388*	16780*
ACCOUNT TO	SMF	V	84	88
ARCHIVE TO		VB	16388*	16780*
COPY FROM		VB	16388*	16780*
COPY TO		VB	16388*	16780*
EXPORT TO		FB*	80*	800*
IMPORT FROM		*	*	*
LIST DOCUMENT TO	DSMLIST other	VM FB	256 80	260 800
RETRIEVE FROM		VB	16388*	16780*
SCRIPT FROM		*	*	*
SCRIPT FILE	DSMLIST other	VM VB	256 256	260 260
Table of Contents	DSMUTTOC*	VB	256	260
Terminal Input	DSMTERMI*	VB	256	260
Terminal Output	DSMTERMO*	VB	256	260
Write to File	DSMUTWTF*	VB	256	260
Messages	DSMUTMSG*	VB	256	260
Command Input	SYSIN*	F*	80*	80*
Printer	DSMLIST*	VM*	256*	260*

Figure 33. OS/VS2 Default Data Set Characteristics. Values marked with an asterisk (*) are fixed and cannot be changed. Values marked with (#) indicate that LRECL format for IMPORT, EXPORT, and SCRIPT FROM can be fixed, variable, fixed block, or variable blocked as defined by the user. However for EXPORT, the LRECL must be long enough to hold the largest record. Notes:

1. The data created by the ACCOUNT command is described in Appendix E, "Format of the Accounting Record" in Figure 36 on page 251. The data sets used by the COPY command are described in Appendix D, "Format of Sequential Data Set for COPY OUT."
2. Default data set characteristics only apply to how an output data set is created when these characteristics are not specified in JCL and the data set is not already created. Input data sets (ACCOUNT FROM, COPY FROM, IMPORT FROM, RETRIEVE FROM, and SCRIPT FROM) have already been assigned these characteristics when created. They are included here only for completeness.
3. If DSMLIST is not used for SCRIPT FILE, output is directed to another ddname ("other" in the figure above). If no DCB information is provided, DCF defaults are RECFM=VB, LRECL=256, and BLKSIZE=260. You may find these defaults undesirable if the file is output to a preallocated data set on DASD or to SYSOUT. For the IBM 4250 printer, the DCB parameters must be: RECFM=VB, LRECL=2052, and BLKSIZE=2056. For the IBM 3800 Model 3 and the IBM 3820 Page Printer, the DCB parameters are: RECFM=VBM, LRECL=8205, and BLKSIZE=8209.
4. When RECFM=VBT for a 3380 dataset, the message IEC137I ("Track Overflow Reset for...") appears because 3380s do not support track overflow. Although DLF commands process correctly with VBT, you can suppress the message by specifying RECFM=VB (instead of the default, VBT) for 3380 datasets.

VSE	Default DLBL/TLBL	RECFM	LRECL	BLKSIZE	Logical Unit	Device
ACCOUNT FROM		VBT	16388*	16780*	SYS001	2400
ACCOUNT TO	DSMACNT	V	84	88		
ARCHIVE TO		VBT	16388*	16780*	SYS002	2400
COPY FROM		VBT	16388*	16780*	SYS002	2400
COPY TO		VBT	16388*	16780*	SYS003	2400
EXPORT TO		FB	80	800	SYS004	2400
IMPORT FROM		FB	80	800	SYS005	2400
LIST DOCUMENT TO	SYSLST other	FM* F	121* 121	121* 121	SYSLST SYS014	Printer 2400
RETRIEVE FROM		VBT	16388*	16780*	SYS002	2400
SCRIPT FROM		VB	256	2560	SYS006	2400
SCRIPT FILE	SYSLST other	FM* VB	121* 256	121* 2560	SYSLST SYS007	Printer 2400
Table of Contents	DSMUTOC*	VB	256	2560	SYS009	2400
Terminal Input	DSMITRM*	VB	256	2560	SYS010	2400
Terminal Output	DSMOTRM*	VB	256	2560	SYS011	2400
Write to File	DSMUWTF*	VB	256	2560	SYS012	2400
Messages	DSMUMSG*	VB	256	2560	SYS013	2400
Command Input	SYSRDR*	F*	80*	80*		Reader
Printer	SYSLST*	FM*	121*	121*		Printer

Figure 34. VSE Default Data Set Characteristics. Values marked with an asterisk (*) are fixed and cannot be changed. However, for EXPORT, the LRECL must be long enough to hold the largest record.

Notes:

1. The data created by the ACCOUNT command is described in Appendix E, "Format of the Accounting Record" in Figure 36 on page 251. The data sets used by the COPY command are described in Appendix D, "Format of Sequential Data Set for COPY OUT."
2. If you do not use SYSLST for SCRIPT FILE, output is directed to another DLBL or TLBL ("other" in the table above). If the device is 4250, you must change the DCB information using the ENVIRONMENT command and the FORMAT(VB), BLOCKSIZE(2056), and RECORDSIZE(2048) operands. If the device is an IBM 3820 Page Printer, use the ENVIRONMENT command and its operands to specify FORMAT(VB), RECORDSIZE(8201), and BLOCKSIZE(8205). See "ENVIRONMENT" on page 50 for an example.

APPENDIX C. DCF FORMATTER (SCRIPT/VS) AND DLF INTERFACE

This appendix describes the communications protocol with which SCRIPT/VS and DLF interact. SCRIPT/VS depends upon this interface for the macros and source files used during formatting. The utility work files listed in Figure 15 on page 115 are managed by SCRIPT/VS.

SCRIPT/VS FORMATTER REQUIREMENTS

When DLF calls the SCRIPT/VS formatter, it calls the module DSMESBEG with register 1 pointing to a parameter list.

REGISTER 1: The parameter list pointed to by register 1 contains a pointer to five fullword addresses. The first two addresses are present in all operating environments; the next two addresses are environmentally dependent; the last address is the same in all operating environments. The format of the parameter list is:

For all environments:

1st Word - Address of a SCRIPT/VS command
2nd Word - Address of DSMSPSIC module
5th Word - Address of SPGPRDCB

For OS/VS2 only:

3rd Word - Address of DSMSPQNM module
4th Word - Address of DSMSPDYN module

For VSE only:

3rd Word - Address of DSMSPDOS module
4th Word - Address of the field in the library program's
SPG table that contains the SCRIPT/VS
command number

The SPGPRDCB address points to a DCB for the output data set which must be available for the formatter to use - DSMLIST for OS/VS2 or SYSLST for VSE.

READ MODULE

The DLF module used by the SCRIPT/VS formatter to read data is called DSMSPSIC. When the formatter calls this module, register 1 must point to a parameter list. Figure 35 on page 244 contains the format of this parameter list.

The formatter supplies the information for fields 2 through 18. DLF gets the area to be used as a communication buffer and puts the address of this buffer in field 19. If a NOTE is done, the library program supplies the record number in field 20. If a POINT is done, the formatter supplies the number. The library program supplies the information in fields 21 through 26.

When DSMSPSIC is called to get the next record, it places the record in a communication buffer pointed to by the address in field 19 in Figure 35 on page 244. The format of the buffer is:

<u>Description</u>	<u>Length</u>	<u>Format</u>
Flags	2 bits	(reserved)
Record length	14 bits	binary
Text		characters

No.	Field	Length	Offset	Format	Description
1	SICFLG SICCLA SICCLS SICGNV SICPNT SICEOF SICNRD SICRAS	1	0	BIN	1..... Close all files .1..... Close 1 file ..1..... Get next record ...1.... Point to record1... Note this record1.. EOF of source1. Open no-read1 Release all dynamic space
2	SICFLG2 SICSVERN SICSDATA SICSZERO	1	1	BIN	1..... Saw version no. .1..... Saw data keywords ..1..... Saw zero library number ...XXXXX Unused or reserved
3	*	3	2		Unused or reserved
4	SICSRLIB	5	3	BIN	Search library number
5	SICGMLIB	4	8	BIN	GML library number
6	*	1	12		Unused reserved
7	SICLIBRN	3	13	BIN	SRC library number
8	SICDOCN	16	16	CHAR	Document name
9	SICVERS	2	32	BIN	Version number
10	*	2	34		Unused or reserved
11	SICDOCP	8	36	CHAR	Document password
12	SICFRDD	8	44	CHAR	From ddname
13	SICFRPDS	8	52	CHAR	From PDS member name
14	SICFRDSN	44	60	CHAR	From data set name
15	SICFRQAL	8	104	CHAR	DSN qualifier
16	SICSRCD	8	112	CHAR	Source data name
17	SICOUTDA	8	120	CHAR	Output data name
18	SICIPNAM	8	128	CHAR	Input processor name
19	SICCMREC	4	136	BIN	SLE record address
20	SICNPREC	4	140	BIN	Note or point record number
21	SICRTNCD	4	144	BIN	Return code
22	SICREACD	4	148	BIN	Reason code
23	*	4	152		Unused or reserved
24	SICPARM	64	156	CHAR	Processor parameters
25	SICINSOR	8	220	CHAR	Input source name
26	SICOUTSC	8	228	CHAR	Output source name

Figure 35. Read Module Parameter List

ADDITIONAL INFORMATION FOR OS/VS2

When dynamic allocation is used, two additional modules are also used, DSMSPQNM and DSMSPDYN. Both of these modules are contained in DLF.

DSMSPQNM: The DSMSPQNM module qualifies a dsname as Unnnnn.name. qualifier, and then, if desired by the caller, performs dynamic allocation by a call to DSMSPDYN. See Appendix B, "JCL Statements, Dynamic Allocation, and Data Set Characteristics," for additional information on data set name qualification.

When DSMSPQNM is called, register 1 must point to a list of five fullword addresses. They are:

1. The address of a 1-byte field

where:

Bit 0	0 - indicates qualify and allocate
	1 - indicates qualify only
Bit 1-7	(reserved)

2. The address of a 44-character data set name field containing the 8-character name to be qualified at entry and the qualified name at exit.
3. The address of an 8-character member name field if the data set to be allocated is a member of a partitioned data set.
4. The address of an 8-character field containing the qualifying name.
5. The address of an 8-character field into which the ddname is placed if dynamic allocation is completed successfully.

All character parameters are left justified in the fields and padded with blanks.

The qualified name consists of a standard OS/VS2 MVS data set name of the form Unnnnnnn.name.qualifier where nnnnnnn is the current authorized user number, name is the name passed as the second parameter and qualifier is the name passed as the fourth parameter. Any trailing blanks in the names are suppressed.

DSMSPDYN: When DSMSPDYN is called, register 1 points to a list of four fullword addresses that contain:

1. The address of a flag byte

where:

Bit 0-4	(reserved)
Bit 5	0 - Member name not specified
	1 - Member name specified
Bit 6	0 - Do not allocate
	1 - Allocate the data set
Bit 7	0 - Do not unallocate
	1 - Unallocate the data set

2. The address of a 44-character field containing the data set name to be allocated if the allocate flag is on.
3. The address of an 8-character field containing the name of the member to be allocated if the data set to be allocated is a member of a partitioned data set.
4. The address of an 8-character field containing:
 - The ddname allocated when this program returns to the caller after a successful allocation
 - The ddname to be unallocated for an unallocation request

All character fields in the parameters are left justified and padded with blanks.

ADDITIONAL INFORMATION FOR VSE

In a VSE environment, the action of the OS/VS2 QSAM I/O function must be simulated. This is done by DLF module DSMSPDOS. DLF initializes an area to look like an OS/VS2 data control block which is used for communication during OS/VS2 I/O operations.

When DSMSPDOS is called, register 1 must point to a list of three fullword addresses. They are:

FCODE The address of a one-byte request code.

TODCB The address of a fullword containing the address of a DCB.

TOREC The address of a full word containing the address of a buffer or the address of the SCRIPT/VS command number.

The significance of these fields during various I/O operations is discussed below.

OPEN: The formatter supplies the following information:

- The contents of the field pointed to by FCODE is X'F1'.
- The contents of the field pointed to by TODCB is the address of the OS/VS2 DCB with the ddname already filled in.
- The contents of the field pointed to by TOREC is the address of the SCRIPT/VS command number.

The command number and the DCB DDNAME field determine what type of DTF is built and opened. DSMSPDOS uses nine basic DTF expansions:

DTFDI	for SYSPCH
DTFDI	for SYSLST
DTFDI	for SYSRDR
DTFDI	for SYSIPT
DTFPR	for 3800
DTFMT	for undefined records on tape
DTFMT	for variable-length records on tape
DTFSD	for undefined records on disk
DTFSD	for spanned records on disk

Prior to opening a file, a skeleton DTF is moved to a GETVIS area where various fields are initialized, notably the CCW addresses, the I/O buffer addresses, and the block size. Modifications to the DTFs found in DSMSPDOS may cause unpredictable results because the DTFs are initialized from the ENVIRONMENT command table before being opened.

CLOSE: The formatter supplies the following information:

- The contents of the field pointed to by FCODE is X'F2'.
- The contents of the field pointed to by TODCB is the address of the OS/VS2 DCB.
- TOREC is not used.

GET: The formatter supplies the following information:

- The contents of the field pointed to by FCODE is X'F3'.
- The contents of the field pointed to by TODCB is the address of an OS/VS2 DCB.
- The contents of the field pointed to by TOREC is the address of the input buffer where the record is to be placed.

PUT: The formatter supplies the following information:

- The contents of the field pointed to by FCODE is X'F4'.
- The contents of the field pointed to by TODCB is the address of the OS/VS DCB.
- The contents of the field pointed to by TOREC is the address of the output buffer. The record is already prepared with an RDW (record description word).

APPENDIX D. FORMAT OF SEQUENTIAL DATA SET FOR COPY OUT

The data set used by the COPY OUT command can be on direct-access storage or magnetic tape. It can contain directories, profiles, and documents from a document library. The records are put out as blocked, variable-length, track-overflow records (RECFM=VBT for OS/VS2) or spanned records (for VSE) with a block size of 16 780 and a maximum logical record size of 16 388 bytes. Each logical record contains one of the following:

- A profile or directory entry from the key-sequenced data set
- A control interval from the entry-sequenced data set.

The sequence of the items is:

1. Profiles, if any, in the order PPRs, CERs (but only if NOATTR was not specified). The SPR and CPR(s) are never copied since COPY IN requires a fully defined and formatted document library.
2. User profiles with the associated data records - all documents in the user's ownership set - in ascending order by user number and name.
3. User mapping records (only if NOATTR was not specified).

The information with each user profile is made up of the following:

- The user profile record.
- The documents owned by the user in a project library if one has been specified. These are arranged in ascending order by name and data type, and in descending sequence by version. Each document has its directory entry followed by its control intervals.
- The documents in the user library associated with that user profile in ascending sequence by name and data type, and descending sequence by version number. Each document has its directory record followed by its control intervals.
- The documents from each public library (first used, second used, ..., last used) in ascending sequence by name and data type, and in descending sequence by version. Each document has its directory entry followed by its control intervals.

APPENDIX E. FORMAT OF THE ACCOUNTING RECORD

Figure 36 contains the format of the type 47 SMF record created by the ACCOUNT command. The System Measurement Facility (SMF) is a control program option of OS/VS2.

Offset	Field Size	Data Format	Contents
0 (0)	1	Binary	System indicator
1 (1)	1	Binary	Record type
2 (2)	4	Binary	Time in hundredths of a second (for OS/VS2) or seconds (for VSE) when the record was moved to accounting buffer.
6 (6)	4	Packed	Date record was moved to accounting buffer in the form 00yydddf (for OS/VS2) or 0yymmddf (for VSE), where f is sign.
10 (A)	4	EBCDIC	System identification ('DLF')
14 (E)	2	Binary	Subsystem id (X'0007' signifies DLF)
16 (10)	2	Binary	Reserved
18 (12)	2	Binary	Length of rest of record, not including this field.
20 (14)	1	Logon condition flags	Bit 0 User not logged off from last session Bit 1 Reserved Bit 2 Reserved Bit 3 Reserved Bit 4 Reserved Bit 5 Library program account record Bit 6 Reserved Bit 7 User locked
21 (15)	3	Binary	User identification
24 (18)	1	Binary	Library type code
25 (19)	3	Binary	Project library identification
28 (1C)	1	Binary	Privilege class
29 (1D)	4	Binary	DASD space used (in bytes) in this library
33 (21)	4	Binary	DASD space used (in bytes) in project and public libraries
37 (25)	7		Reserved

Figure 36. Format of the Accounting Record

APPENDIX F. OUTPUT OF LIST COMMANDS

Figure 37 on page 254 shows the output you will get by issuing a series of DLF LIST commands:

You identify yourself to DLF as general user 88 having a password of PSWD88:

AUTH 88/PSWD88

and request a listing of your status by issuing the command:

LIST USER 88

You then determine the status and space usage of your documents by issuing the commands:

LIST DOCUMENT
LIST DOCUMENT SPACE ARCHIVE

This allows you to see the documents in your private library and those you own in public libraries and your project library.

You can issue the LIST CLUSTER command with the SPACE operand to help determine where you can store new documents most effectively.

LIST CLUSTER SPACE

Notes:

1. The output of the LIST USER command shows the following:
 - a. User 88 has a private library with a password. The password is not shown in the listing for password security. User 88 is not locked nor is user 88's own library controlled. User 88 is assigned project library 1 000. User 88 may also store documents containing up to 512 000 bytes of text in the document library and has used 20% of this allotment - 1 000 bytes in user 88's own library (SPACE IN USE) and 103 000 bytes in the user's assigned project library plus various public libraries (PRJ/PUB SPACE).
 - b. Under the LIST USER command, SPACE IN USE is the space used by documents owned by the user in the user's own library - the sum of FILE SIZE for those documents. PUB/PRJ SPACE is the space used by the documents owned by that user in the user's project library and/or any public libraries in which that user owns documents. Both numbers are given in thousands of bytes, rounded upward to the nearest thousand bytes.

The LIST USER SPACE LIMIT is the maximum number of text bytes (in thousands) the user is allowed to own in the document library. Percent full (%%% FULL) is the percent of SPACE LIMIT used by documents owned by that user (SPACE IN USE plus PUB/PRJ SPACE). It is rounded down to the nearest percent (%).
2. The first LIST DOCUMENT command lists all of the documents owned by the current user (user 88) using the standard LIST DOCUMENT report. Because the command did not specify LOCK, ARCHIVE, or VERSION, no locked or archived documents and only the latest version of any versioned document is listed. This example shows user 88 has stored documents in his own library (library 88), in two public libraries (library 1 314 151 - the SCRIPT/VS required public library and library 9 999 999 - a general use public library), and in his project library (library 1 000).

DSMPIN004I DIRECTORY DATA SET DSMINDIR OPEN FOR INPUT.

DSMPIN007I DIRECTORY DATA SET DSMPTDIR OPEN FOR OUTPUT.

DSMPIN054I CLUSTER DSMPTLIB AVAILABLE FOR INPUT.

DSMPIN055I CLUSTER DSMPTLIB AVAILABLE FOR OUTPUT.

DSMPIN001I LIST PARAMETER IN EFFECT FOR THIS STEP.

AUTH 88/

DSMEXC091I RETURN CODE = 0. TIME = 16:58:19

LIST USER 88

USER...	PRIVILEGE	LIBRARY	CONTROLS...	PROJECT	CLASS...	SOURCE...	DATA....	SPACE..	SPACE..	PRJ/PUB	XXXX	HOST	EXIT.....
NUMBER	CLASS	TYPE	PHD	LCK	CTL	LIBRARY	ATTR	ATTR	ATTR	LIMIT	IN USE	SPACE	FULL INFORMATION
88	NONE	PRIV	YES	NO	NO	1000				512	1	103	20%
****	1 USER(S) LISTED. ****												

DSMEXC091I RETURN CODE = 0. TIME = 16:58:19

LIST DOCUMENT

LIBRARY	DOCUMENT.....	DOCUMENT	DATA....	DOCUMENT	CLUSTER	CONTROLS...	CREATION	VER-	ARC	OWNER..	FILE...	# OF.		
NUMBER	NAME	CLASS	TYPE	SOURCE	NAME	SHR	PHD	LCK	DATE	SION	HV	ID	SIZE	REC.
88	DOC 1				DSMPTLIB	NO	NO	NO	05/10/85		NO	88	397	45
88	VERSION				DSMPTLIB	NO	NO	NO	05/10/85	2	NO	88	89	3
1314151	ATMSREP				DSMPTLIB	PUB	NO	NO	05/10/85		NO	88	3768	240
9999999	BOOK1				MEMO	PUB	NO	NO	05/10/85		NO	88	28458	279
9999999	BOOK2				DSMPTLIB	PUB	NO	NO	05/10/85		NO	88	28458	279
9999999	BOOK3				BOOK	PUB	NO	NO	05/10/85		NO	88	28458	279
9999999	MEMO1				MEMO	PUB	NO	NO	05/10/85		NO	88	450	9
9999999	MEMO2				DSMPTLIB	PUB	NO	NO	05/10/85		NO	88	450	9
9999999	MEMO3				BOOK	PUB	NO	NO	05/10/85		NO	88	450	9
1000	REPORT				DSMPTLIB	NO	NO	NO	05/10/85		NO	88	12288	2048
****	10 DOCUMENT(S) LISTED. ****													

DSMEXC091I RETURN CODE = 0. TIME = 16:58:20

LIST DOCUMENT SPACE ARCHIVE

Figure 37 (Part 1 of 2). Example of output from LIST commands

LIBRARY NUMBER	DOCUMENT NAME	DATA TYPE	VER- SION	OWNER ID	DOCUMENT CLASS	CLUSTER NAME	CTL-INTRVL SIZE	# OF REC.	* * SPACE USAGE (BYTES-%) * *	* * VACANT
88	DOC 1			88	DSMPTLIB		4096	1	45	4096 446- 11% 3650- 89%
88	VERSION		2	88	DSMPTLIB		4096	1	3	4096 138- 3% 3958- 97%
1314151	ATMSREP			88	DSMPTLIB		4096	1	240	4096 3837- 94% 259- 6%
9999999	ARCHIVED			88	DSMPTLIB		4096	2*	80*	8192* 5058* 62% 3134* 38%
*** ARCHIVE INFORMATION = 05/10/85 16:58:08.31										
9999999	BOOK1			88	MEMO		512	70	279	35840 31888- 89% 3952- 11%
9999999	BOOK2			88	DSMPTLIB		4096	8	279	32768 28850- 88% 3918- 12%
9999999	BOOK3			88	BOOK		16384	2	279	32768 28556- 87% 4212- 13%
9999999	MEMO1			88	MEMO		512	1	9	512 499- 97% 13- 3%
9999999	MEMO2			88	DSMPTLIB		4096	1	9	4096 499- 12% 3597- 88%
9999999	MEMO3			88	BOOK		16384	1	9	16384 499- 3% 15885- 97%
1000	REPORT			88	DSMPTLIB		4096	4	2048	16384 12484- 76% 3900- 24%

***** 11 DOCUMENT(S) LISTED ***** -- SUMMARY NEXT PAGE --

**** SUMMARY REPORT OF SPACE USAGE IN 10 NON-ARCHIVED DOCUMENTS:

* * DOCUMENT SIZES =====> 138 = SMALLEST
 10770 = AVERAGE
 31888 = LARGEST

* * COUNT TOTALS =====> 90 = NUMBER OF CTL-INTRVL
 3200 = NUMBER OF RECORDS
 103286 = NUMBER OF TEXT BYTES

* * SPACE USAGE (BYTES-%) =====> 107696- 71% = FILLED
 43344- 29% = VACANT
 151040-100% = TOTAL

DSMEXC091I RETURN CODE = 0. TIME = 16:58:20

LIST CLUSTER SPACE

CLUSTER NAME	INPUT DD NAME	OUTPUT DD NAME	CTL INTRVL SIZE	* * * * S P A C E U S A G E (1000S OF BYTES-%) * * * *	TOTAL SPACE	USED	AVAILABLE	NEVER USED	REUSABLE
BOOK	DSMILIB2	DSMPLIB2	16384		393	49- 12%	344- 88%	344- 88%	0- 0%
DSMPTLIB	DSMINLIB	DSMPTLIB	4096		406	103- 25%	303- 75%	221- 55%	82- 20%
MEMO	DSMILIB1	DSMPLIB1	512		235	36- 15%	199- 85%	0- 0%	199- 85%

DSMEXC091I RETURN CODE = 0. TIME = 16:58:20

DSMEXC095I END OF LIBRARY PROGRAM. TIME = 16:58:21

DSMEXC099I HIGHEST RETURN CODE FROM LIBRARY PROGRAM = 0.

Figure 37 (Part 2 of 2). Example of output from LIST commands

3. Both LIST DOCUMENT reports list all document size characteristics in bytes. The document size (called FILE SIZE in the standard LIST DOCUMENT report) is the number of text bytes in the document plus a 2-byte length field for each logical text record.
4. The output of the second LIST DOCUMENT command shows the Space Usage report.

- a. This command specifies two additional operands. The SPACE operand causes the command to produce a different report - the LIST DOCUMENT Space Usage report. The ARCHIVE operand adds document ARCHIVED, the only archived document owned by user 88, to the documents that are also listed in the Standard report.
- b. The LIST DOCUMENT Space Usage report relates each document to the CLUSTER in which it is stored. This space is distributed over COUNT control intervals (CTL-INTRVLs), each of which is SIZE bytes long. The TOTAL SPACE USAGE, in bytes, for the document shows the total physical space in the control interval(s) occupied by the document.

FILLED space is the space used by the document's text (called FILE SIZE in the standard LIST DOCUMENT report) plus the space used by DLF for cluster management (49-bytes per control interval). VACANT space is used by neither the document nor DLF. It occurs because each control interval may contain all or part of one and only one document and because DLF does not split logical text records between control intervals. VACANT and FILLED are listed in bytes and as percentages of the TOTAL space used by the document.

- c. If there is more than one NON-ARCHIVED document listed, a summary page is also generated. It provides a count and other summary information about ONLY the NON-ARCHIVED documents listed. This includes the SMALLEST, AVERAGE, and LARGEST document sizes and totals for the number of CTL-INTRVLs and RECORDS. The summary page also lists a total number of TEXT BYTES that excludes the bytes used by DLF for control purposes. This is essentially a total for the field called FILE SIZE on a Standard LIST DOCUMENT report (excluding any archived documents). The last lines on the summary page are totals and overall percentages for the FILLED, VACANT, and TOTAL fields on the Space Usage report.
- d. All percentages are rounded with the exception of those values between 0% and 1% which are rounded to 1% and those values between 99% and 100% which are rounded to 99%.
- e. The LIST DOCUMENT Space Usage report in the example shows the comparative efficiency of storing documents in clusters of different control interval sizes. In the example, two documents of different sizes, BOOKn and MEMOn, are stored in three clusters of different control interval sizes.

The large document, BOOKn, is stored with similar VACANT space and physical space usage in each cluster. The document should be stored in the cluster using the fewest control intervals to decrease DLF's access time. That would be cluster BOOK in this example.

The small document, MEMOn, is stored with decreasing efficiency (increasing VACANT space) as the cluster's control interval SIZE increases since only one document can occupy a given control interval. It should be stored in a cluster which has a control interval SIZE slightly larger than the number of bytes FILLED by the document, which would be cluster MEMO in this example.

5. The output of the last command shows the Space Usage report for clusters:

- a. The LIST CLUSTER command with the SPACE operand shows which clusters have space available for document storage and how that space is distributed. The report obtains data from the cluster profile record (CPR) and from the free space header (FSH) found in the first control interval (CI) for each cluster in the document library.
- b. The LIST CLUSTER Space Usage report lists all data in thousands of bytes except CTL INTRVL SIZE, which is given in bytes. All percentages are rounded with the exception of those values between 0% and 1% which are rounded to 1% and those values between 99% and 100% which are rounded to 99%.

TOTAL SPACE gives the overall size of the cluster. This is divided into the space USED by documents and the space AVAILABLE for additional documents. The AVAILABLE space consists of contiguous space at the end of the cluster which was NEVER USED for document storage, plus REUSABLE space. The REUSABLE space was returned to the cluster after purging, archiving, or replacing documents and is probably fragmented. DLF will use all space in each cluster once before reusing any space (i.e., NEVER USED must be zero before REUSABLE space is reused).

- c. In this example, Cluster BOOK has 12% of its space currently used to store documents. All the remaining space has never been used and, therefore, is all contiguous space (i.e., all documents will be stored in consecutive control intervals and there will be no increase in document access time due to fragmentation).

Cluster DSMPTLIB has both space that has been NEVER USED and REUSABLE space available for document storage. Fragmentation is not a current problem because all space that was NEVER USED will be used first.

Cluster MEMO has 15% of its space currently used to store documents. All of the remaining space has been used at least once and, therefore, is probably fragmented (i.e., all documents whose sizes require more than one control interval will probably be stored in non-consecutive control intervals and there will probably be an increase in document access time due to that fragmentation). The fragmentation problem in cluster MEMO could be resolved using the procedures described in Chapter 7, "Administering the Document Library" on page 205. However, if most documents that will go in that cluster will be smaller than the control interval size (i.e., use only one control interval), this would not be necessary.

- d. All clusters in the example were allocated ten tracks of 3380 DASD space by Access Methods Services. The TOTAL SPACE for cluster MEMO with a control interval SIZE of 512 bytes is significantly less than the TOTAL SPACE for clusters DSMPTLIB and BOOK. This shows the potential variations in usable allocated space between clusters of different control interval sizes. Space allocation can be most effectively used by matching DASD type to control interval size based on knowledge of AMS space allocation. (See the appropriate VSAM/AMS manual for allocation details.)

APPENDIX G. RELATIONSHIP OF ATMS-III AND THE DOCUMENT LIBRARY FACILITY

The Document Library Facility (Release 3) may be used by ATMS-III users for their archive storage. ATMS-III documents that are archived are actually put into the document library with an IMPORT command. Documents to be archived are read by ATMS-III from its permanent storage and are submitted as data within a batch job through the system's internal reader. A batch job thus submitted by ATMS-III contains: 1) JCL required to invoke DLF; 2) an AUTH command identifying ATMS-III as a library user; and 3) one or more IMPORT commands each of which is followed by a document being archived from ATMS-III.

Because data submitted through the internal reader is limited to 80 byte records, ATMS-III breaks up the text units of its documents (text units are variable length up to 230 bytes) into segments prior to submission. DLF must then recreate the original units before storing the document into the document library. This is done by DLF in response to continuation controls inserted into the text unit segments by the ATMS-III program. After a complete text unit is constructed, DLF invokes the ATMS conversion processor (DSMACAIP) that is supplied by IBM as a part of the Document Composition Facility (5748-XX9). Options generated by ATMS-III on the IMPORT command cause the conversion processor to be invoked for each recreated ATMS-III unit. The conversion processor deletes any trailing blanks from the recreated unit and eliminates the New Line (X'15') character which delimits units in ATMS-III. This conversion processor may also convert ATMS-III controls to SCRIPT/VS macros and symbols (depending on the INDATA and DATA value in the IMPORT command) before returning to DLF, which stores the converted record.

The options used by ATMS-III on the generated IMPORT commands described above are discussed in detail in Appendix D of the Advanced Text Management System-III (ATMS-III) Program Reference Manual, SH20-2424. The functions of the conversion processor (DSMACAIP) are discussed in Document Composition Facility: SCRIPT/VS Text Programmer's Guide, SH35-0069.

The ATMS-III user need not be concerned with the IMPORT command or options used for archiving to the document library because these are generated by ATMS-III. Likewise, the library user need not be concerned with the IMPORT options required to invoke the conversion processor for this special processing of documents archived from ATMS-III since the proper IMPORT command is generated by ATMS-III as a part of a job stream invoking DLF.

The library user may, however, want to IMPORT ATMS documents that have been put out to a sequential data set by ATMS-III. See "Processing ATMS Documents" in Chapter 4 of this manual for examples of how these sequential ATMS files can be imported or scripted by a library user.

GLOSSARY OF TERMS AND ABBREVIATIONS

This glossary includes words and phrases that have special meanings in the Document Library Facility. The terms are defined as they are used in this book. If you do not find the term you are looking for, refer to the index or to the IBM Vocabulary For Data Processing, Telecommunications, and Office Systems, GC20-1699.

This glossary also includes definitions developed by the American National Standards Institute (ANSI) and the International Organization for Standardization (ISO), which are identified by an asterisk. This material is reproduced from the American National Dictionary for Information Processing, copyright 1977 by the Computer and Business Equipment Manufacturers Association, copies of which may be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018.

administrator. A user of DLF with the additional authority to perform certain functions, among which is the ability to issue the privileged commands associated with setting up and maintaining the document library.

ATMS-III. Advanced Text Management System. An interactive terminal system for the editing, storing, and formatting of text documents.

attributex. A characteristic of a unit of data such as length, value, or method of representation.

attribute name. A name or label given to a particular type of data contained in a document, such as class, data type, or source.

authorize. The AUTH command causes a user, identified by user number, to be authorized to perform certain functions using the library program.

batch environment. The environment in which noninteractive programs are executed.

byte. In System/3, System/360, and System/370, a sequence of eight adjacent binary digits that are operated upon as a unit and that constitute the smallest addressable unit in the system.

calling sequence. An arrangement of instructions, and in some cases of data, that is necessary to perform the transfer of control to another routine.

CER. See class entry record.

character. A symbol used in printing or data processing. For example, a letter of the alphabet, a numeral, a punctuation mark, or any other symbol that represents information.

class entry record (CER). A set of data that describes the category into which a certain group of documents falls, such as book, letter, or memo. This record also identifies the cluster in which this class of documents is stored.

cluster. In systems with VSAM, a group of related, cataloged, stored records identified by name such as NEWCLUST. In the document library, a cluster is made up of records contained in the control intervals of one VSAM entry-sequenced data set (ESDS).

cluster profile record (CPR). A set of data that describes the VSAM data sets of various control-interval sizes for the storage of documents.

command statement. A statement that is used to cause functions to be performed by a program.

control interval. A fixed-length area of direct-access storage in which VSAM stores records and distributes free space. Also, in a key-sequenced data set or file, the set of records pointed to by an entry in the sequence-set index record. It is the unit of information that VSAM transmits to or from direct-access storage. A control interval always comprises an integral number of physical records.

controlled library. A project or public library that is write-access controlled; that is, only the library owner or a user with administrator authority can delete documents from the library. Only the library owner can store documents in the library.

convention. A general agreement about how material will be presented.

conversion. The process of changing from one form of representation to another.

* American National Standard Definition

conversion processor. A computer program that processes a machine-readable document. The document can include formatting controls written in one formatter language, and the processor can produce a machine-readable document that includes formatting controls appropriate for another formatter language.

CPR. See cluster profile record.

current user. The library user who issued the last DLF AUTH command. DLF uses this user's user profile record (UPR) to determine which functions the current user may perform and what defaults to use for document attributes such as data and class.

data control block. A structured work area used by a problem program to communicate with an access method.

data definition name (ddname). The job control language (JCL) name of a data definition (DD) statement, which corresponds to a data control block that contains the same name. DLF uses the ddname to generically describe both the ddname on the OS/VS2 DD statement and the filename on the VSE DLBL and TLBL statements.

data definition (DD) statement. A job control statement that describes a data set associated with a particular job step.

data set. The major unit of data storage and retrieval in the operating system, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

data set name. The term or phrase used to identify a data set. See also **qualified name**.

data type. Primarily used with other attributes to make the document unique. Also indicates the type of formatting controls contained within a document. Controls are instructions in the text of the document. The instructions are interpreted by a text-formatting program (ATMS formatter or SCRIPT/VS, for example), and specifies how the document is to be formatted.

data space. A storage area defined in the volume table of contents of a direct-access volume for the exclusive use of VSAM to store files, indexes, and catalogs.

DCB. See data control block.

DD card. See data definition statement.

ddname. See data definition name.

debug. To detect, trace, and eliminate errors in computer programs.

default value. A value assumed by a computer program when a control word, command, or control statement with no parameters is processed or when a value is not specified for a parameter.

DER. See document entry record.

directory data set. A data set that contains all the profile records and document entry records for each document in the source data set. These records provide managerial information for the document library.

dname. See document name

document. A machine-readable collection of lines of text or images, usually called a source document. See also **source document**.

document entry record (DER). A set of data that describes a document stored in the library. The description includes the document's name, data type, password, share status, location in the source data set, data stored, version numbers if applicable, and to what class and cluster it is assigned.

document library. A set of at least two VSAM data sets with a particular internal structure used by the Document Library Facility. It is formatted and maintained by the library program and contains profiles and documents.

document name. An alphanumeric term that identifies a document.

dsname. See data set name.

dynamic allocation. For OS/VS2 MVS only, a facility that allows programs to access files without the need for DD statements in the job's JCL.

EBCDIC. Extended binary-coded decimal interchange code. A coded character set consisting of 8-bit coded characters.

ESDS. Entry-sequenced data set. A VSAM data set whose records are loaded without respect to their contents and whose relative byte addresses cannot change. Records are retrieved and stored by addressed access, and new records are added at the end of the data set.

export. To send out. As used in DLF, signifies that a document is to be copied from the library to an external data set (outside the DLF library).

file. A set of related records treated as a unit. Frequently used as a synonym for data set (particularly in the VSE environment).

formatter. (1) A computer program that prepares a source document to be printed. (2) That part of SCRIPT/VS that arranges input lines for a particular logical device type.

Generalized Markup Language (GML). A language that may be used to identify the parts of a source document without respect to particular processing.

general user. An authorized user of the library program with limited capabilities. A general user does not have administrator authority. General users control only those documents and libraries they own.

ghost DERS. Document entry records that do not have an associated user profile record (UPR) in the document library into which they are copied.

GML. See Generalized Markup Language.

hexadecimal. Pertaining to a number system based on 16, using the sixteen digits 0, 1, ..., 9, A, B, C, D, E, F. For example, 1B (hexadecimal) equals 27 (decimal). (See also EBCDIC.)

host system. A data processing system that is used to maximize the use of facilities in an operating environment.

import. To bring in. As used in DLF, signifies that a file on an external (outside the DLF library) data set is to be copied into the library as a document.

input device. A computer-controlled device used to enter information into a computer system (for example, a terminal used to create a document).

input line. A line or record, as entered into a source file, to be processed by a computer program.

interface. A shared boundary. An interface might be a portion of storage or registers accessed by two or more computer programs.

JCL. See job control language.

job control language (JCL). A language of control statements used to identify a computer job or describe its requirements to the operating system.

job control statement. A statement that provides an operating system with information about the job being run.

keyword. One of the predefined words of an artificial language.

KSDS. Key-sequenced data set. A VSAM data set whose records are loaded in key sequence and controlled by an index. Records are retrieved and stored by

keyed access or by addressed access, and new records are inserted in sequence by means of distributed free space. Relative byte addresses of records can change because of control interval or control area splits.

LER. Library entry record. LER's indicate the number of documents a user has placed in a public library owned by another library user. It also indicates the number of the library where the documents were placed.

library. (1) A collection of related files. (2) An area in the source (entry-sequenced) VSAM data set that is used for the storage of data, primarily documents.

library owner. A user identified by a library user number.

library user. An authorized user of the library program. A library user can be a general user or an administrator. DLF identifies each library user by that user's unique library user number.

load module. A program unit that is suitable for loading into processor storage for execution; it is usually the output of a linkage editor.

load module library. A partitioned data set that is used to store load modules.

machine-readable. Ability, by a machine, to acquire or interpret data from a storage device, from a data medium, or from another source.

macro. An instruction in a source language that is to be replaced by a defined sequence of instructions in the same source language.

operand. A predefined word that further qualifies a DLF command. Command operands can be keyword or variable (positional) parameters.

option. Information entered with a command to control the execution of the library program.

ownership set. All the documents owned by a particular library user regardless of which library the documents are stored in.

parameter. A variable that is given a constant value for a specified application, which may denote the application.

password. A unique string of characters that a program, computer operator, or user must supply to meet security requirements before gaining access to data.

PPR. See processor profile record.

private library. A library that is accessible only to its owner, and in some instances, administrators, except for those documents that have been specified as shared. This is a controlled library.

processor. In software, a computer program that performs functions such as compiling, assembling, translating for a specific programming language. In DLF, a non-DLF program that can be called by the library program to perform special processing for certain DLF commands.

processor profile record (PPR). A set of data that identifies a non-DLF program used to perform special processing for certain DLF commands. The record indicates the processed document's data type prior to processing (INDATA), its data type after processing (OUTDATA), and the entry point name of the processor's load module.

profile. A collection (user profile) of information that defines a user, the cluster in which a class of documents is stored (class entry record), processors (processor profiles), the document library (system profile), or a cluster (cluster profile).

project library. A library that is accessible to a specific group of users, known as project members, as specified in the user profiles for those users. This library can be controlled or uncontrolled.

public library. A library that is accessible to any authorized user of the document library. This library can be controlled or uncontrolled.

qualified name. A name explicitly accompanied by a specification of the class to which it belongs in a specified classification system.

qualifier. In a data set name, all names in a qualified name other than the rightmost, which is called the simple name.

read access. With respect to a document library, a library user's ability to read documents in the document library for which he knows the document's name, password, data type, and library of residence. A general user can read any document in his own library, his project library, or any public library, and any shared documents. An administrator can read any document in any user's library.

SCRIPT/VS. A document and text-processing program that executes in the ATMS, TSO foreground, and CMS environments.

sequential data set. A data set whose records are organized on the basis of their successive physical positions, such as on magnetic tape.

source document. A machine-readable collection of lines of text or characters that is used for input to a computer program.

space. A blank area separating words or lines. Represented in EBCDIC by the character identified as X'40'.

SPR. See system profile record.

subroutine. A sequenced set of statements that can be used in one or more computer programs and at one or more points in a computer program.

syntax. (1) The structure of expressions in a language. (2) The rules governing the structure of a language.

system profile record (SPR). A set of data that describes the document library itself.

terminal. A device, usually equipped with a keyboard and some kind of display, capable of sending and receiving information over a communication channel.

UMR. See user mapping record.

uncontrolled library. A project or public library in which any authorized user can store documents or delete documents he owns.

UPR. See user profile record.

user mapping record (UMR). A set of data that describes how an external user, such as an ATMS user, is known to the library program. The record identifies the external system the user is on, the user's identification on that system, and the library user number assigned.

user number. The 1 to 7 numeric characters that are used by the library program to identify a user. Associated with this user number are a user profile and a library.

user profile record (UPR). A set of data that describes the privileges, library type, space allocation, and project library of a particular user of the library program. It is identified by a user number.

version. In a document library, a variation on the original composition of a document.

work area. That portion of main storage used by DLF to temporarily hold data.

write access. With respect to a document library, a user's ability to store documents in his private and uncontrolled project libraries, and any uncontrolled public libraries.

INDEX

A

- abbreviations, command 17
- Access Method Services
 - use of 195
- ACCOUNT command
 - coding examples 131
 - description 129
 - format 129
 - notes about 130
 - smf record created by 251
- ALTER CLASS Command
 - coding example 133
 - description 133
 - format 133
 - note about 133
- ALTER CLUSTER command
 - coding example 135
 - description 134
 - format 134
 - note about 134
 - use for
 - in formatting additional space 208
- ALTER Commands
 - listing of 132
 - summary of 132
- ALTER DOCUMENT command
 - coding examples 30, 137
 - description 29, 136
 - examples of use
 - changing document names and attributes 123
 - format 29, 136
 - notes about 30, 137
- ALTER MAP command
 - coding example 139
 - description 138
 - format 138
 - notes about 138
- ALTER PROCESS command
 - coding example 141
 - description 140
 - format 140
 - note about 140
- ALTER SYSTEM command
 - coding example 142
 - description 142
 - format 142
 - note about 142
 - use for
 - to format the newly allocated space. 208
- ALTER USER command
 - coding examples 145
 - description 143
 - format 143
 - notes about 145
- application program
 - address list 221
 - calling DLF
 - address list 221
 - example 223
 - register settings 221
 - return codes 221
 - data supplied by 221
 - data to DLF
 - parameter list format 223
 - use of parameter list 223
 - example 224
 - requirement 198
 - in authorized library 198
- ARCHIVE ALL command
 - coding examples 33, 148
 - description 32, 146
 - format 32, 146
 - notes about 33, 148
- ARCHIVE command
 - examples of use
 - archiving and retrieving your documents and their versions 119
- ARCHIVE commands
 - listing of 31
 - summary of 31
- ARCHIVE DOCUMENT command
 - coding examples 35
 - description 34
 - format 34
 - notes about 35
- ARCHIVE NAMELIST command
 - coding examples 38
 - description 36
 - format 36
 - note about 36
- ATMS
 - conversion program
 - must reside in authorized library 198
 - used for 215
 - examples
 - processing ATMS documents 117
- attribute names
 - class
 - use for 197
- attribute processor
 - reason codes
 - sample list 217
 - return codes
 - sample list 217
- attribute profile
 - example of modifying 211
- AUTH command
 - coding examples 39
 - description 39
 - examples
 - backing and copying 107
 - converting ATMS documents 117
 - formatting with SCRIPT/VS 110, 111, 113
 - producing a copy 105
 - purging and listing 108
 - with IMPORT 101
 - format 39
 - notes about 39
 - use during reorganization 210
- authority
 - needed for 205
 - use of 205
- authorized library
 - required 198
 - use by user-written programs 221
 - use of 200
 - ATMS conversion program 198
 - DLF 198

library program 198
SCRIPT/VS 198

B

backing up documents
example 107

C

class entry record 27
definition 261
CLASS see Operands, Keyword Operands
attribute 10
definition 21
CLOSE command
description 40
format 40
notes about 40
use by user-written programs 221
CLUSTER see operands, keyword operands
definition 22
clusters
efficient use of 197
use for 197
coding examples
COPY IN command 153
Coding Samples
examples of coding commands
RETRIEVE DOCUMENT command 90
command abbreviations 17
control interval
definition 261
COPY command
examples
backing up documents 107
copying documents into another
document library 107
COPY commands
documents from a tape 206
listing of 41
summary of use 41
COPY DOCUMENT command
coding examples 45
description 42
examples of use
backing up your documents 107
copying versions of documents 121
format 43
notes about 45
use of
in freeing space 208
COPY IN command
coding example 153
coding examples 47
description 46, 151
format 46, 151
note about 47
notes about 153
COPY LIBRARY command
coding example 155
description 154
note about 154
use of
during adding more space 208
COPY OUT command
coding example 49
coding examples 158

data set for 249
description 48, 156
examples of use
backing up your documents 107
format 48, 156
notes about 49, 157
use for
to reorganize 210
use of
to free space 206

COPY USER command
coding example 159
description 159
format 159
notes about 159
use of
in freeing space 208
copying documents into another document
library
example 107
core-image library
required 198
use of 200
CYLINDERS parameter
restriction for entry-sequenced data
set 197
use of 196

D

DATA see operands, keyword operands
definition 21
data sets
default characteristics 237
DD statements
definition 13
requirements
table of predefined ddnames 13
ddnames
SCRIPT/VS utility names 115
table of predefined ddnames 13
defaults
data set characteristics 237
DEFINE CLASS command
coding examples 161
description 161
format 161
note about 161
DEFINE CLUSTER command
coding example 163
description 162
example
defining VSAM data sets 198
format 162
notes about 162
use of 195
DEFINE Commands
listing of 160
DEFINE MAP command
coding examples 164
description 164
format 164
notes about 164
DEFINE PROCESS command
coding example 166
description 165
format 165
notes about 165
purpose 215
using to define required
profiles 201

DEFINE SYSTEM command
 action regarding secondary
 extents 197
 coding example 168
 description 167
 notes about 167
 use during reorganization 210
 use of 201
 result 201
 what it does 197
 DEFINE USER command
 coding examples 171
 description 169
 format 169
 note about 171
 use 201
 result 201
 use during reorganization 210
 use of
 defining user 205
 deleting
 obsolete documents 208
 DER (document entry record)
 location 196
 directory data set
 contents 197, 201
 description 196
 reasons for reorganizing 211
 space requirements guidelines 197
 DLBL names
 SCRIPT/VS utility names 115
 table of predefined DLBL names 13
 DLBL statement
 requirements
 table of predefined ddnames 13
 DLF
 called from application program 221
 commands used by user-written
 programs 221
 data from application program 223
 examples of library user
 functions 101
 register conventions 216
 return codes
 from I/O load module 224
 with a processor
 saves work areas 216
 with application program
 return code placement 221
 with processors
 format of parameter area 217
 DLF commands
 examples of use 101
 dlib# see operands, variable parameter
 operands
 definition 19
 dname see operands, variable parameter
 operands
 definition 19
 docid see operands, variable parameter
 operands
 definition 19
 document
 copying from a tape 206
 default cluster for 197
 deleting
 when user profile deleted 208
 examples
 archiving and retrieving your
 documents and their versions 119
 backing up your documents 107
 changing document names and
 attributes 123
 copying into another document
 library 107
 formatting a document with
 SCRIPT/VS 110
 processing ATMS 117
 processing ATMS documents 116
 producing single copy of 105
 putting documents into a
 library 101
 storing in multiple clusters 197
 document entry record
 definition 262
 Document Library
 adding more space 208
 example 208
 steps for 208
 defining 195
 ensuring adequate space 206
 reorganizing
 reasons 210
 reorganizing if space is limited
 steps to 210
 reorganizing if space is not limited
 steps to 210
 structure 3
 VSAM master catalog
 required 195
 VSAM parameters for defining 196
 Document Library Facility
 capabilities 3
 definition 3
 install 195
 dpass see operands, variable parameter
 operands
 definition 20
 dpass, see operands, variable parameter
 operands
 definition 19
 DSMIDIR
 when required 13
 DSMILIB
 when required 13
 DSMINDIR
 when required 13
 DSMINLIB
 when required 13
 DSMPDIR
 when required 13
 DSMPLIB
 when required 13
 DSMPDIR
 when required 13
 DSMPTDIR
 when required 13
 DSMPTLIB
 when required 13
 dynamic allocation
 definition 262

E

entry-sequenced data set
 multi-volume
 space formatting 197
 ENVIRONMENT command
 coding examples 52
 description 50
 examples
 backing and copying 107
 converting ATMS documents 117
 formatting with SCRIPT/VS 110,
 111, 113
 producing a copy 105

- putting documents into a library 101
 - examples of use
 - archiving and retrieving your documents and their versions 119
 - formatting a document with SCRIPT/VS 110
 - putting documents into your library 102
 - format 50
 - notes about 52
- ESDS Source Cluster
 - reorganizing 211
- examples
 - application program CALL 224
 - AUTH command
 - backing up and copying 107
 - converting ATMS documents 117
 - formatting with SCRIPT/VS 110, 111, 113
 - listing document names 108
 - producing a copy 105
 - putting documents into a library 101
 - COPY command
 - backing up and copying 107
 - COPY LIBRARY command 155
 - COPY OUT command 158
 - COPY USER command 159
 - defining VSAM data sets 198
 - ENVIRONMENT command
 - backing up and copying 107
 - converting ATMS documents 117
 - formatting with SCRIPT/VS 110, 111, 113
 - producing a copy 105
 - putting documents into a library 101
 - examples of coding commands
 - ACCOUNT command 131
 - ALTER CLASS Command 133
 - ALTER CLUSTER command 135
 - ALTER DOCUMENT command 30, 137
 - ALTER MAP command 139
 - ALTER PROCESS command 141
 - ALTER SYSTEM command 142
 - ALTER USER command 145
 - AUTH command 39
 - COPY DOCUMENT command 45
 - COPY IN command 47
 - DEFINE CLASS command 161
 - DEFINE CLUSTER command 163
 - DEFINE MAP command 164
 - DEFINE PROCESS command 166
 - DEFINE SYSTEM command 168
 - DEFINE USER command 171
 - ENVIRONMENT command 52
 - EXPORT command 56
 - IMPORT command 65, 68
 - LIST CLUSTER command 70
 - LIST DOCUMENT command 75
 - LIST PROCESS command 76
 - LIST USER command 78
 - PASSWORD command 79
 - PROTECT command 81, 183
 - PURGE DOCUMENT command 83
 - PURGE USER command 191
 - READ command 86
 - RETRIEVE ALL command 88, 193
 - RETRIEVE DOCUMENT command 90
 - RETRIEVE NAMELIST command 93

- SCRIPT command 98
 - examples of command coding
 - COPY OUT Command 49
 - EXPORT command
 - producing a copy 105
 - IMPORT command
 - converting ATMS documents 117
 - putting documents into a library 101
 - LIST command
 - listing document names 108
 - LIST DOCUMENT command 177
 - LIST MAP command 178
 - LIST SYSTEM command 179
 - LIST USER command 181
 - PURGE CLASS command 185
 - PURGE CLUSTER command 186
 - PURGE command
 - listing document names 108
 - PURGE DOCUMENT command 188
 - PURGE MAP command 189
 - PURGE PROCESS command 190
 - SCRIPT command
 - converting ATMS documents 117
 - formatting with SCRIPT/VS 110, 111, 113
 - executing DLF 13
 - EXPORT command
 - coding examples 56
 - description 54
 - examples
 - producing a copy 105
 - examples of use
 - producing copies of a single document 105
 - format 54
 - notes about 56
 - use for
 - in converting release 1 library 235
 - use with processors 215

F

- formatting a document with SCRIPT/VS
 - example 110, 111, 113
 - IBM 3800 Model 1 111
 - IBM 3800 Model 3 111
 - IBM 3820 Page Printer 111
 - IBM 4250 printer 111
- formatting additional space
 - example of formatting extended library space 208
 - using ALTER CLUSTER for 208
 - using ALTER SYSTEM for 208
- FROM see operands, keyword operands
- definition 23

G

- ghost DERs 75, 151, 154, 177, 263
 - definition of 151
 - recovering from 177

I

IMPORT command
 coding example 68
 coding examples 65
 description 58
 example atms 116
 examples
 converting ATMS documents 117
 putting documents into a library 101
 examples of use
 importing data dynamically using 103
 processing ATMS documents 116
 putting documents into your own library 101
 format 59
 notes about 65
 use for
 in converting release 1 library 235
 use with attribute processor 215
 use with processors 215
installing
 library program 198
 processor 200
 processors 200

J

JCL 13, 16, 95, 237
 data set names 237
 requirements 237
 rules for 13
 sample statements 16
JCL and command syntax conventions 13
job control statements (JCS)
 see JCL

L

library
 how established 205
library conversion
 from release 1 to release 3
 steps for administrators 235
 steps for general users 235
library user
 capabilities 101
 examples of functions 101
LIST CLASS command
 description 68
 format 68
 notes about 68
LIST CLUSTER command
 coding example 70
 description 69
 format 69
LIST commands
 listing of 67, 172

summary of 67, 172
LIST DOCUMENT command
 coding examples 75, 177
 description 71, 173
 examples of use
 listing document names 108
 format 71, 173
 notes about 74, 176
 to locate documents with no profiles 206
LIST MAP command
 coding examples 178
 description 178
 format 178
 notes about 178
LIST parameter
 DLF as a batch job 13
 when calling DLF as subroutine 221
LIST PROCESS command
 coding example 76
 description 76
 format 76
 notes about 76
LIST SYSTEM command
 coding examples 179
 description 179
 format 179
 notes about 179
LIST USER command
 coding examples 78, 181
 description 77, 180
 format 77, 180
 notes about 180
listing document names
 example 108
LISTX parameter
 (see LIST parameter) 221
load module name
 for DLF 198
 specifying for processor 200

M

multi-volume
 entry-sequenced data set
 formatting 197

N

name see operands, variable parameter
operands
 definition 20
NOLIST parameter
 DLF as a batch job 13
 when calling DLF as subroutine 221
NOLISTX parameter
 (see NOLIST parameter) 221
null data types
 copying documents with 30, 137
number see operands, variable parameter
operands
 definition 20

0

operands

- commonly used 19
- keyword operands 19
 - CLASS 19
 - CLUSTER 19
 - DATA 19
 - FROM 19
 - SOURCE 19
 - TO 19
 - VERSION 19
- variable parameter operands
 - dlib# 19
 - dname 19
 - docid 19
 - dpass 19
 - name 19
 - number 19

OS/VS2

- Authorized Program Facility
 - required 198
- ddnames
 - SCRIPT/VS utility names 115

OS/VS2 MVS

- installation requirements 198
- VSAM master catalog 195

P

parameters

- for VSAM data sets 196

PASSWORD command

- coding examples 79
- description 79
- format 79
- notes about 79

performance

- improving 197

processing ATMS documents

processor

- ATMS processor 215
 - definition 200
 - description 215
 - example 200
 - examples of calling 219
 - format of communication buffer 219
 - format of message area 218
 - format of parameter area 217
 - functions 215
 - input names 215
 - input to 215
 - installation procedure 200
 - installing 200
 - message on return 216
 - modifies communication buffer 216
 - notes about 219
 - output names 215
 - processing sequence 215, 216
 - reason codes
 - on return 216
 - register conventions 216
 - requirements 215
 - return codes
 - continue processing 215
 - on return 216
 - return processing 216
 - saves information 216

stored

- in authorized library 215
- in core-image library 215
- to print messages 218
- use of communication buffer 216
- when called 215
- with a processor
 - supplies work areas 216

processors

- install 195
- producing copies of a single document
 - example 105

profile

- defining 200
- location 196
- specifying load module name 200

profiles

- copying 206
- unwanted copied 206

project library

- accessing documents 205
- no profiles for documents 206

PROTECT command

- coding example 81, 183
- description 80, 182
- format 80, 182
- notes about 81, 183

public library

- no profiles for documents 206

PURGE CLASS command

- coding example 185
- description 185
- format 185
- note about 185

PURGE CLUSTER command

- coding example 186
- description 186
- format 186
- notes about 186

PURGE commands

- listing of 82, 184
- summary of 184

PURGE DOCUMENT command

- coding example 83, 188
- description 82, 187
- examples of use

- purging documents and getting lists of document names 108

- format 82, 187

- note about 82, 187

- use of

- to free space 206-208

PURGE MAP command

- coding example 189
- description 189
- format 189
- notes about 189

PURGE PROCESS command

- description 190
- format 190
- notes about 190

PURGE USER command

- coding example 191
- description 191
- format 191
- notes about 191

- use of

- to free space 206

purging documents

- example 108

putting documents into a library

- example 101

R

RACF (see Resource Access Control Facility) 203

READ command

- coding example 86
- description 84
- example 223
- format 84
- notes about 85
- READ command 223
- use by user-written programs 221
- use with a processor 215
- use with processors 215

reorganizing directory data set

- reasons for doing 211

REPRO command

- use for reorganizing directory data set 211

Resource Access Control Facility

- changing RACF resource name 142
- providing RACF security 203
 - command examples 204
 - general information 203
 - protecting DLF user numbers 204
 - protecting libraries 203
- reference manuals v

RETRIEVE ALL command

- coding example 88, 193
- description 87, 192
- format 87, 192
- notes about 88, 193

RETRIEVE command

- examples of use
 - archiving and retrieving your documents and their versions 121

RETRIEVE commands

- listing of 87, 192
- summary of 87, 192

RETRIEVE DOCUMENT command

- coding example 90
- coding examples 90
- description, 89
- format 89
- notes about 90

RETRIEVE NAMELIST command

- description 91
- format 91
- notes about 91

S

SCRIPT command

- coding examples 98
- data supplied by application program 221
- description 94
- examples
 - converting ATMS documents 117
 - formatting a document 110, 111, 113
- examples of use
 - formatting a document with SCRIPT/VS 110
 - processing ATMS documents 116
- format 94
- notes about 97
- use by user-written programs 221
- use with a processor 215

use with processors 215

SCRIPT/VS

- control words
 - requiring utility data sets 115
- examples
 - formatting with 110, 111, 113
- installing
 - OS/VS requirements 198
 - VSE requirements 198
- load module name 198
- must reside in authorized library 198
- utility data set names 115

SCRIPT/VS control words

- requiring utility data sets 115

secondary extents

- restriction 197

see entry-sequenced data set

see source data set

setting up document library

- defining VSAM data sets 195

source data set

- DER
 - location 196
 - relation to documents 196
 - when formatted 201
- source name see operands, keyword operands
- operands
 - definition 21
- space in library
 - adding more
 - defining larger data sets 208
 - extending current data sets 208
 - reorganizing if not limited 210
 - reorganizing if space is limited 210
- space requirements
 - documents 5, 197
- space requirements guidelines
 - VSAM data sets 196
- space usage reports
 - LIST CLUSTER command 69
 - LIST DOCUMENT command 73, 175
 - notes about 70, 206, 207
- syntax conventions 13
- system profile
 - defining 201
 - location 196, 201
 - required 201
 - what it does 201

T

TO see operands, keyword operands

- definition 22

U

user exit routines

- calling sequence for 228, 229
- parameter area 229
- register conventions 228
- commands changed by
 - ALTER MAP 232
 - ALTER USER 232
 - DEFINE MAP 232
 - DEFINE USER 232
 - EXPORT 232
 - IMPORT 232

- PURGE DOCUMENT 232
- PURGE MAP 232
- RETRIEVE 232
- commands for 225
- example of
 - DEFINE MAP 230
- fields changed by 225
- flags
 - set by DLF 229
 - set by exit routine 229
- messages for 230
- module for 225
- notes about 233
- records accessed by
 - document entry record (DER) 225
 - user mapping record (UMR) 225
 - user profile record (UPR) 225
- return codes for 230
- uses for 225
- user mapping record
 - definition of 264
- user profile
 - copying 206
 - defining initially 201
 - deleting
 - steps to 208
 - unwanted profiles 208
 - example of defining 205
 - location 196
 - purpose 205
 - required 201
 - use 201
- user profile record
 - definition of 265
- user-written programs
 - requirements 221
 - use of
 - authorized library 221
 - library commands 221
- utility data sets
 - required for SCRIPT/VS 115

V

- version
 - definition of 265
- VERSION see operands, keyword operands
 - definition 22
- versions
 - examples of coding 119
 - residing on different clusters 197
- volume
 - one or more
 - allocated improves performance 197
 - data set allocation 197
- VSAM
 - password protection 195
- VSAM data sets
 - creating more space 208
 - define in a catalog 195
 - defining 195
 - required 195
 - entry-sequenced 195
 - examples
 - creating more space 208
 - defining data sets 199, 200
 - key-sequenced 195
 - location 197
 - parameters for 196
 - restriction 197
 - see also source data set, directory data set
 - see also VSAM master catalog
 - space requirements guidelines 196
 - to have password protection 195
- VSAM master catalog
 - defining 195
 - password protection 195
- VSAM user catalog
 - password protection 195
 - use of 195
- VSE
 - core-image library
 - required 198
 - DLBL names
 - SCRIPT/VS utility names 115
 - installation requirements 198
 - VSAM master catalog required 195

Order No. SH20-9165-5

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

You may use this form to communicate your comments about this publication, its organization, or subject matter with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or the IBM branch office serving your locality.

- | | Yes | No |
|---|--------------------------|--------------------------|
| • Does the publication meet your needs? | <input type="checkbox"/> | <input type="checkbox"/> |
| • Did you find the information: | | |
| Accurate? | <input type="checkbox"/> | <input type="checkbox"/> |
| Easy to read and understand? | <input type="checkbox"/> | <input type="checkbox"/> |
| Easy to retrieve? | <input type="checkbox"/> | <input type="checkbox"/> |
| Organized for convenient use? | <input type="checkbox"/> | <input type="checkbox"/> |
| Legible? | <input type="checkbox"/> | <input type="checkbox"/> |
| Complete? | <input type="checkbox"/> | <input type="checkbox"/> |
| Well illustrated? | <input type="checkbox"/> | <input type="checkbox"/> |
| Written for your technical level? | <input type="checkbox"/> | <input type="checkbox"/> |
| • How do you use this publication: | | |
| As an introduction to the subject? | <input type="checkbox"/> | |
| For advanced knowledge of the subject? | <input type="checkbox"/> | |
| To learn about operating procedures? | <input type="checkbox"/> | |
| As an instructor in class? | <input type="checkbox"/> | |
| As a student in class? | <input type="checkbox"/> | |
| As a reference manual? | <input type="checkbox"/> | |
| • What is your occupation? | | |

Comments:

If you would like a reply, please give your name and address.

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail them directly to the address on the back of the title page.)

Reader's Comment Form

Fold and tape

Please Do Not Staple

Fold and tape

Attention: Information Products Division
Department 580



NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40

ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Information Products Division
P. O. Box 1900
Department 580
Boulder, Colorado 80301



Fold and tape

Please Do Not Staple

Fold and tape

Document Library Facility Guide Printed in U.S.A. SH20-9165-5



IBM Technical Newsletter

This Newsletter No. SN32-8009

Date: April 22, 1987

Base Publication No. SH20-9165-5

File No. S370-20

Previous Newsletters None

Document Library Facility Guide

© Copyright IBM Corp. 1987

This Technical Newsletter provides replacement pages for the subject publication.

The effected pages are:

17, 18	Replace
161 - 164	Replace
185, 186	Replace
195, 196	Replace
211, 212	Replace
212.1, 212.2	Add

A change to the text is indicated by a TNL statement at the top of the effected page, and a vertical line to the left of the effected text.

Note: *Please file this cover letter at the back of the manual to provide a record of changes.*



Printed in USA

SH20-9165-05

