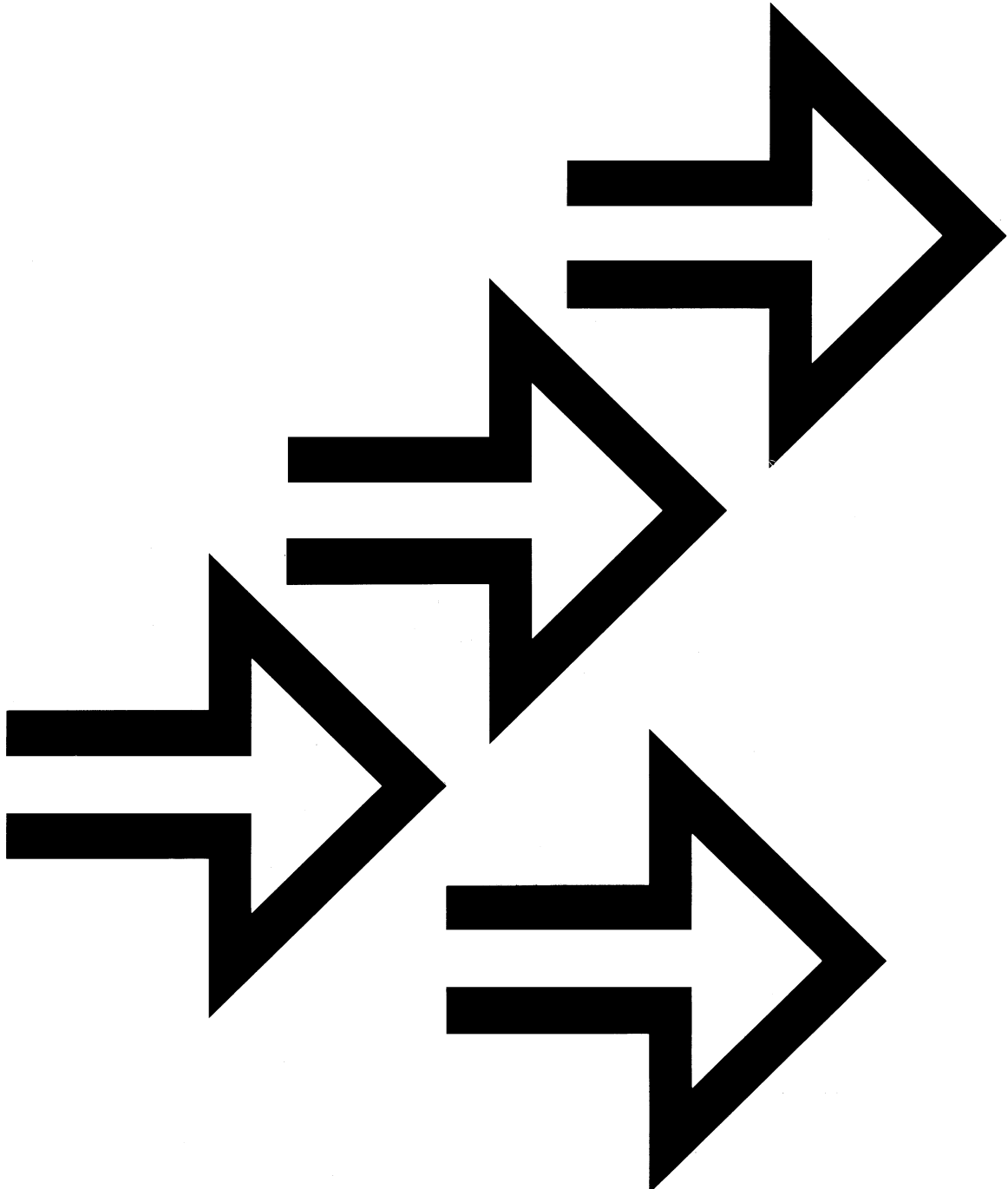




# Document Library Facility Diagnosis Reference

Program  
Product

Contains Restricted Materials of IBM  
Licensed Materials—Property of IBM  
© Copyright IBM Corp. 1985  
LY35-0072-1  
File No. S370-20



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.

## Second Edition (June 1985)

This book is a major revision of Document Library Facility Diagnosis Reference (LY35-0072-0), which is now obsolete. This edition applies to Release 3 of the Document Library Facility, Program Number 5748-XXE, and to any subsequent releases until otherwise indicated in new editions or technical newsletters.

This product contains restricted materials of IBM program number 5748-XXE.

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 1984, 1985

## **PREFACE**

### **PURPOSE**

This book provides information to help you isolate failures in the Document Library Facility (DLF) and communicate those failures to an IBM support center representative. It will help you use failure clues or symptoms to narrow the area of the program that must be examined and to associate problem symptoms with the functions performed by the Document Library Facility.

### **HOW TO USE THIS BOOK**

You can use the "Data Areas" and "Service Aids" sections in this book with the Document Library Facility Diagnosis Guide to help develop the keyword string.

If a search of the software support facility (SSF) and early warning system (EWS) does not produce a fix, the following steps can help you narrow the area to be examined and associate problem symptoms with program functions:

- Read the "Program Overview" to gain an understanding of DLF structure, control flow, principal control and data areas, and relationships of the DLF functions.
- Go to the section in the "Program Functional Description" that describes the function that you suspect was in operation at the time of the program failure.
- If you know what module was in control at the time of the failure, go to the "Module Directory" for a module description and its calling relationships. This information can help you isolate the failing module (referring to the "Program Functional Description" might be helpful).
- Refer to the Document Library Facility Diagnosis Guide to determine what information is needed to complete an authorized program analysis report (APAR).

### **PREREQUISITE KNOWLEDGE**

Prerequisite knowledge for using this book is a basic understanding of the Document Library Facility capabilities as described in:

- Document Library Facility Guide, SH20-9165, which is the user guide for the Document Library Facility.
- Document Library Facility Diagnosis Guide, SY35-0071, which describes the process of developing keywords to aid in the diagnosis of Document Library Facility program failures.

### **HOW THIS BOOK IS ORGANIZED**

- **Section I Program Overview** — This part of the book presents an overview of DLF, its purpose, and its functions.
- **Section II - Program Functional Description** — Describes the principal processing paths through each function and shows how the functions interact with each other.
- **Section III - Module Directory** — Shows the module-to-module communications within DLF.

- **Section IV - Data Areas** — Describes the data areas used for communication between functions and programs.
- **Section V - Service Aids** — Describes tools that collect or analyze diagnostic information for DLF.

#### **RELATED PUBLICATIONS**

- Document Library Facility Diagnosis Guide, SY35-0071, provides customers and IBM support representatives with information for diagnosing program products.
- Document Library Facility: Messages, SH35-0049, documents the messages DLF issues and suggests actions to be taken. This publication is available only in printed form.
- Document Composition Facility Diagnosis Guide, SY35-0067, describes Document Composition Facility diagnosis procedures.
- OS/VS2 MVS Supervisor Services and Macro Instructions, GC28-0683, contains routing codes that DLF uses when communicating with the console operator.
- Field Engineering Programming System General Information, G229-2228, describes how to complete APAR forms.

#### **RESTRICTED PUBLICATION**

- Document Composition Facility Diagnosis Reference, LY35-0068, contains information about the logic and processing of the Document Composition Facility.



## **PUBLICATION LIBRARY GUIDE FOR THE DOCUMENT LIBRARY FACILITY**

The following table is a library guide to the manuals for the Document Library Facility (DLF). The manuals are listed as they relate to user tasks.

| <b>User Tasks</b>                               | <b>Typical Audience</b> | <b>Recommended Books</b>   | <b>Brief Description</b>  |
|---|-------------------------|--|---|
| Using the document library                      | DLF library user        | DLF Guide (SH20-9165)<br>DLF Messages (SH35-0049)  | Describes DLF, its interfaces, the library commands, and library messages |
| Installing and maintaining DLF                  | Systems programmer      | DLF Program Directory<br>DLF Guide (SH20-9165)<br>DLF Diagnosis Guide (SY35-0071)<br>DLF Diagnosis Reference (LY35-0072)<br>DLF Messages (SH35-0049) | Gives information on error isolation and program tailoring                |
| Setting up and maintaining the document library | DLF administrator       | DLF Guide (SH20-9165)<br>DLF Messages (SH35-0049)  | Describes DLF, its interfaces, the library commands, and library messages |



## CONTENTS

|   |           |
|---|-----------|
| <b>Section I - Program Overview</b>                     | <b>1</b>  |
| General Description                                     | 1         |
| Purpose   | 1         |
| Functions   | 1         |
| General User Commands                                   | 1         |
| Library Administrator Commands                          | 3         |
| General Data Flow                                       | 4         |
| The Document Library                                    | 6         |
| Library Program Control Flow                            | 6         |
| Principal Control and Data Areas                        | 7         |
| External Interfaces                                     | 7         |
| Operating Environment                                   | 7         |
| Programming Systems                                     | 7         |
| Hardware Requirements                                   | 8         |
| Storage Requirements                                    | 8         |
| Environment Dependent Facilities                        | 9         |
| Structural Overview                                     | 10        |
| Document Library Structure                              | 10        |
| The Library Directory (KSDS) Data Set(s)                | 10        |
| The Library Source Entry Sequenced Data Sets            | 13        |
| Program Structure                                       | 14        |
| DLF Load Modules  | 14        |
| DLF Source Modules                                      | 15        |
| External Modules (not a part of the DLF product)        | 17        |
| Program Conventions                                     | 18        |
| Register Conventions                                    | 19        |
| Module Naming Conventions                               | 19        |
| Communication Between Modules of the Library Program    | 19        |
| Control Flow Within the Library Program                 | 19        |
| Principal Data and Control Areas of the Library Program | 21        |
| Internal DLF Interfaces                                 | 23        |
| External Interfaces                                     | 24        |
| Batch Invocation Interface                              | 26        |
| DLF Called as a Subroutine - Interface                  | 26        |
| Password Hashing Interface                              | 29        |
| User Exit Interface                                     | 31        |
| Special Processor Interface                             | 33        |
| DLF to DGF Interface                                    | 36        |
| DCF to DLF Interfaces                                   | 38        |
| SCRIPT Input Module Interface                           | 43        |
| External Data Sets Used by DLF                          | 45        |
| <b>Section II - Program Functional Description</b>      | <b>46</b> |
| DLF Initialization, Control, and Termination            | 46        |
| DLF Control and Termination - DSMSPXEC                  | 46        |
| DLF Initialization - DSMSPPIN                           | 47        |
| Command Validation                                      | 49        |
| Command Decode and Validation - DSMSPCMD                | 49        |
| Command Scan and Parse - DSMSPSCN                       | 49        |
| Command Syntax Graphs Part 1 — DSMSPSYX                 | 50        |
| Command Syntax Graphs Part 2 — DSMSPSY                  | 50        |
| Command Initialization - DSMSPINT                       | 51        |
| Command Processing                                      | 52        |
| ACCOUNT Command - DSMSPACT                              | 52        |
| ARCHIVE Commands (3 commands) - DSMSPARC                | 53        |
| AUTH Command - DSMSPAUT                                 | 54        |
| CLOSE Command - DSMSPCLO                                | 54        |
| COPY Commands (Controller) - DSMSPCPY                   | 55        |
| COPY DOCUMENT Command - DSMSPCPD                        | 55        |
| COPY IN Command - DSMSPCPI                              | 56        |
| COPY LIBRARY Command - DSMSPCPL                         | 56        |
| COPY OUT Command - DSMSPCPO                             | 57        |
| COPY USER Command - DSMSPCPU                            | 57        |
| DEFINE/ALTER Commands (Controller) - DSMSPDEF           | 58        |
| DEFINE/ALTER CLUSTER Command - DSMSPDAC                 | 59        |
| ALTER DOCUMENT Command - DSMSPDAD                       | 59        |
| DEFINE/ALTER PROCESS Command - DSMSPDAF                 | 60        |

|  |         |
|--|---------|
| DEFINE/ALTER CLASS Command - DSMSPDAL                | 60      |
| DEFINE/ALTER MAP Command - DSMSPDAM                  | 61      |
| DEFINE/ALTER SYSTEM Command - DSMSPDAS               | 62      |
| DEFINE/ALTER USER Command - DSMSPDAU                 | 62      |
| ENVIRONMENT Command (VSE only) - DSMSPDOS            | 63      |
| EXPORT Command - DSMSPEXP                            | 65      |
| IMPORT Command - DSMSPIMP                            | 66      |
| LIST Commands (Controller) - DSMSPLIS                | 67      |
| LIST CLASS Command - DSMSPLIL                        | 68      |
| LIST CLUSTER Command - DSMSPLIC                      | 68      |
| LIST DOCUMENT Command - DSMSPLID                     | 69      |
| LIST MAP Command - DSMSPLIM                          | 70      |
| LIST PROCESS Command - DSMSPLIF                      | 71      |
| LIST SYSTEM Command - DSMSPLIS (LISTSYS Subroutine)  | 71      |
| LIST USER Command - DSMSPLIU                         | 72      |
| PASSWORD Command - DSMSPPAS                          | 73      |
| PROTECT Command - DSMSPPRO                           | 73      |
| PURGE Commands (Controller) - DSMSPPUR               | 74      |
| PURGE CLASS Command - DSMSPPUL                       | 74      |
| PURGE CLUSTER Command - DSMSPPUC                     | 75      |
| PURGE DOCUMENT Command - DSMSPPUD                    | 75      |
| PURGE MAP Command - DSMSPPUM                         | 76      |
| PURGE PROCESS Command - DSMSPPUF                     | 77      |
| PURGE USER Command - DSMSPPUU                        | 77      |
| READ Command - DSMSPREP                              | 78      |
| RETRIEVE Commands (3 commands) - DSMSPRET            | 79      |
| Command Termination                                  | 80      |
| Command Termination - DSMSPTRM                       | 80      |
| Services   | 81      |
| External File Input Services - DSMSPDSI              | 81      |
| External File Output Services - DSMSPDSO             | 81      |
| Dynamic Allocation Services (OS/VS2 only) - DSMSPDYN | 82      |
| User Exit Controlling Services - DSMSPEXI            | 83      |
| Cluster Control Services - DSMSPGCL                  | 83      |
| Get Next Record Services - DSMSPGNX                  | 84      |
| Password Hashing Services - DSMSPHSH                 | 86      |
| Input Library Access Services - DSMSPILU             | 86      |
| Library Input Services - DSMSPLBI                    | 87      |
| Library Output Services - DSMSPLBO                   | 88      |
| Library Validation Services - DSMSPLSV               | 89      |
| Message Services - DSMSPSMG                          | 89      |
| Output Library I/O Services - DSMSPOLU               | 90      |
| Qualify Name Services (OS/VS2 only) - DSMSPQNM       | 91      |
| Read Concatenate Services - DSMSPRDC                 | 92      |
| Source Input Services - DSMSPSIC                     | 93      |
| SMF Output Services - DSMSPSMF                       | 94      |
| Source Input Open Services - DSMSPSOP                | 95      |
| Sequential Output Services - DSMSPTPO                | 96      |
| User Exit (Skeleton Supplied) - DSMSPUXI             | 96      |
| <br>Section III - Module Directory                   | <br>98  |
| <br>Section IV - Data Areas                          | <br>104 |
| Document Library Records                             | 105     |
| Directory Data Set (KSDS) Records                    | 105     |
| Class Entry Record (CER)                             | 106     |
| Cluster Profile Record (CPR)                         | 107     |
| Document Entry Record (DER)                          | 108     |
| Library Entry Record (LER)                           | 110     |
| Processor Profile Record (PPR)                       | 111     |
| System Profile Record (SPR)                          | 112     |
| User Mapping Record (UMR)                            | 113     |
| User Profile Record (UPR)                            | 114     |
| Source Data Set (ESDS) Records                       | 116     |
| Control Interval Trailer (CIT)                       | 117     |
| Free Space Header (FSH)                              | 119     |
| Free Space Record (FSR)                              | 121     |
| Source Line Entry (SLE)                              | 122     |
| System Control Blocks                                | 123     |
| Cluster Control Block (CLB)                          | 124     |
| List Processor Command Work Area (LST)               | 125     |
| Open File Name Table (OFN)                           | 128     |
| Standard Register Save Area Used By DLF (SAV)        | 131     |
| Library Global Table (SPG)                           | 133     |

|  |            |
|--|------------|
| Parameter Lists . . . . .                          | 150        |
| User Exit Interface (EXI) Parameter List . . . . . | 151        |
| Special Processor Parameter List (IPP) . . . . .   | 153        |
| Source Input Parameter List (SIC) . . . . .        | 154        |
| <b>Section V - Service Aids . . . . .</b>          | <b>157</b> |
| FE Patch Areas . . . . .                           | 157        |
| The Register-Save-Area Chain . . . . .             | 157        |
| Module Naming Convention . . . . .                 | 158        |

## **FIGURES**

|     |   |     |
|-----|---|-----|
| 1.  | Summary of the General User Commands . . . . .                  | 1   |
| 2.  | Summary of the Administrator Commands . . . . .                 | 3   |
| 3.  | General Data Flow . . . . .                                     | 5   |
| 4.  | Structure of the Document Library . . . . .                     | 11  |
| 5.  | Structure of Directory Record Keys . . . . .                    | 12  |
| 6.  | Control Flow Within the Library Program . . . . .               | 20  |
| 7.  | Principal Control and Data Areas . . . . .                      | 23  |
| 8.  | DLF External Interfaces . . . . .                               | 25  |
| 9.  | DLF Batch Invocation Interface . . . . .                        | 27  |
| 10. | DLF Called as a Subroutine - Interface . . . . .                | 28  |
| 11. | DLF Password Hashing Interface . . . . .                        | 30  |
| 12. | DLF User Exit Interface . . . . .                               | 32  |
| 13. | EXPORT and IMPORT Command Special Processor Interface . . . . . | 34  |
| 14. | READ and SCRIPT Command Special Processor Interface . . . . .   | 35  |
| 15. | DLF to DCF Interface . . . . .                                  | 37  |
| 16. | DCF to DLF Interface For Source Input Data . . . . .            | 39  |
| 17. | DCF to DLF Interface For Dynamic Allocation . . . . .           | 40  |
| 18. | DCF to DLF Interface For Qualify Name Services . . . . .        | 41  |
| 19. | DCF to DLF Interface for VSE I/O Services . . . . .             | 42  |
| 20. | SCRIPT Input Module - Interface . . . . .                       | 44  |
| 21. | table of called modules . . . . .                               | 98  |
| 22. | A Register Save Area . . . . .                                  | 157 |
| 23. | A Chain of Called Modules . . . . .                             | 158 |

## **SUMMARY OF AMENDMENTS**

### **SECOND EDITION**

Significant changes in this edition are:

- The addition of information pertinent to the advent of two Small Program Enhancements (SPEs):
  1. The first Small Program Enhancement applies 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.
- Changes required for the support of VSE/AF 2.1.0.





## **SECTION I - PROGRAM OVERVIEW**

This section provides a general description of the Document Library Facility, its operating environment, structure, and function.

### **GENERAL DESCRIPTION**

#### **PURPOSE**

The Document Library Facility (DLF) functions as a repository for data of many types and provides software facilities for the storage, retrieval, and management of documents owned by library users. If the Document Composition Facility (DCF) is also installed, documents can be formatted (SCRIPT) in the batch environments in which DLF runs (see "Operating Environment" on page 7).

#### **FUNCTIONS**

The library program performs the functions specified by user commands. Commands can be in the input stream (if DLF is invoked as a batch job) or passed by another program (if DLF is called as a subroutine).

#### **General User Commands**

The following figure summarizes the commands any library general user can issue to perform the library program functions described in Document Library Facility Guide.

| 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.  |

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

| Command                 | Function  |
|-------------------------|---|
| 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 or provides a Space Usage report.  |
| LIST DOCUMENT (Changed) | Lists the attributes of selected documents stored in the user's own library, project library, or any public library or provides a Space Usage report.   |
| 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 1 (Part 2 of 2). Summary of the General User Commands

## Library Administrator Commands

The following figure summarizes the library program commands that the library administrator uses to perform the functions described in in Document Library Facility Guide.

| 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.   |
| DEFINE MAP     | Maps an external user to a library user number.  |
| DEFINE PROCESS | Defines a processor to DLF.  |

Figure 2 (Part 1 of 2). Summary of the Administrator Commands

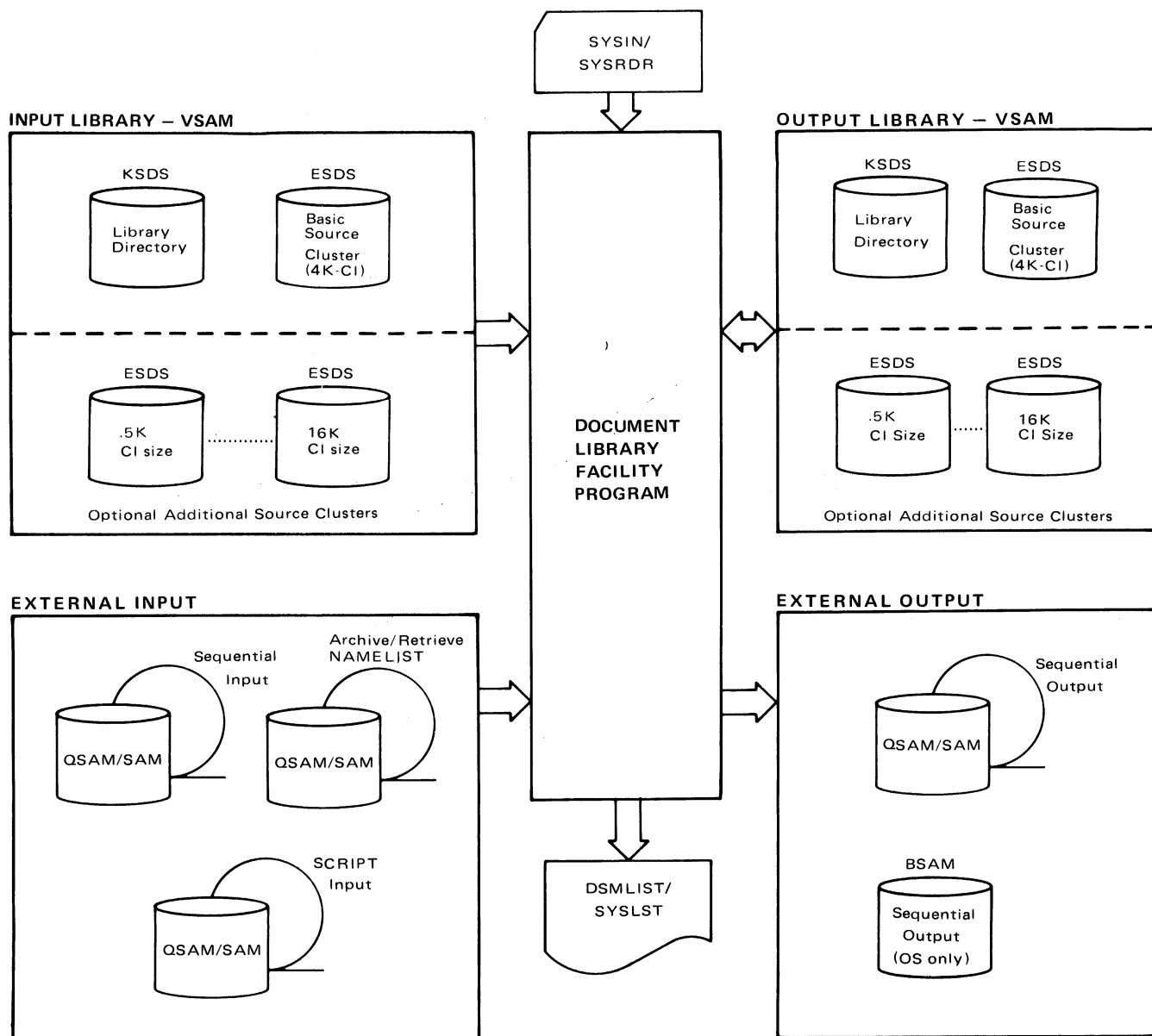
| Command                    | Function   |
|----------------------------|--|
| DEFINE SYSTEM              | Establishes the document library by formatting the space for the default 4K cluster, defining RACF protection, specifying the ENQ/LOCK RNAME, and specifying the number of the OS/VS2 operator routing code. |
| DEFINE USER                | Defines a new user and his library to DLF.   |
| LIST DOCUMENT<br>(Changed) | Lists the attributes of any or all of the documents in the document library or provides a Space Usage report.  |
| LIST MAP<br>(Changed)      | Lists all of the user mapping records (UMRs) that are defined in the directory data set.   |
| LIST SYSTEM<br>(New)       | Lists information from the system profile record (SPR).  |
| LIST USER<br>(Changed)     | Lists the user profile records (UPRs) of any or all library users.   |
| 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 2 (Part 2 of 2). Summary of the Administrator Commands

## GENERAL DATA FLOW

As illustrated in Figure 3, the primary components of the Document Library Facility are the library and the library program. The library contains text documents and other user data as well as information about the library itself. The library program provides facilities for the controlled storage and access of data in the library.

The library resides on VSAM data sets and consists of a directory data set and one or more source data sets. The directory data set (a VSAM key-sequenced data set (KSDS)) contains information about the library and its users, including pointers to users' documents in the source data sets. Source data sets are VSAM entry-sequenced data set (ESDS) clusters. At least one cluster, the basic 4K (CI size) cluster is required. Additional clusters of CI size ranging from 0.5K to 16K are optional.



**Figure 3. General Data Flow**

DLF can be invoked to use two separate physical libraries, one for input and one for output. The normal DLF invocation, however, specifies the same physical library for both input and output. Allowing DLF invocation with two separate physical libraries facilitates creation of new document libraries and/or the moving of users and their documents from one physical document library to another.

The library program imports documents to and exports documents from the library. Therefore, it has facilities to access external (non-library) data sets to read or write data. The

library program also accesses commands from the job input stream (SYSIN in the OS/VS2 MVS environment, SYSRDR in the VSE environment) and outputs messages and formatted data to a print data set (DSMLIST in the OS/VS2 MVS environment, SYSLST in the VSE environment).

The general data flow among the document library(s), the library program, the external data sets, and the system data sets is shown in Figure 3 on page 5.

## THE DOCUMENT LIBRARY

As stated before, the document library consists of a directory data set and one or more source clusters. The directory data set is a VSAM KSDS data set and the source clusters are VSAM ESDS data sets:

- The document library directory consists of profile and entry records that contain information about the library, the library users, and the documents belonging to each library user. There are eight different types of directory records. The position of a record in the directory is determined by its key (first 30 bytes of DLF directory records).
- The library source clusters are used to store users' documents. The basic cluster (4K CI size cluster) is required. Additional source clusters (with CI size ranging from .5K to 16K within VSAM constraints) are optional. CIs within each cluster are formatted when the cluster is defined (with the DEFINE CLUSTER command) or altered with additional space (with the ALTER CLUSTER command).

## LIBRARY PROGRAM CONTROL FLOW

The library program is a single load module named DSMSPEXC consisting of either 64 (in a VSE environment) or 65 (in an OS/VS2 MVS environment) link-edited DLF modules. DSMSPEXC is also the controlling module of the library program.

The library program can be invoked as a batch job or called as a subroutine by another program. In either case, the DSMSPEXC module gets control and calls DSMSPPIN (a module within the library program) to initialize DLF.

DSMSPEXC is then ready to control command processing. If DLF is invoked as a batch job, the commands will be read one at a time from the job input stream. If DLF is called as a subroutine, a single command is passed to DLF by the calling program.

When DSMSPEXC isolates a command, it controls the analysis, initialization, processing, and termination of that command. DSMSPEXC performs each step by calling another module from within the library program. DSMSPEXC then returns to the calling program for the next command or reads the next command from the job input stream. This command processing loop continues until there are no more commands to be processed or until the library program detects a terminating error. At that point DSMSPEXC returns to the host system (or to the calling program if DLF was called as a subroutine).

**Note:** When DLF is called as a subroutine, it can be called many times by the calling program during a DLF session (once for each command to be processed). DSMSPEXC sets flags on the first call so that it will not call DSMSPPIN for DLF initialization on subsequent calls. DSMSPEXC resets these flags when DLF terminates.

## PRINCIPAL CONTROL AND DATA AREAS

The SPG (DLF global table) is an area within the library program that contains frequently used information such as data areas, flags, control blocks, buffers, pointers to loaded programs, and pointers to dynamic areas.

Register 11 is set to the address of the SPG by DSMSPEXC when DLF is invoked (or called as a subroutine) and normally remains set during the DLF session. Modules of the library program obtain addressability to the SPG fields through the DSMGSPG mapping structure.

In addition to the SPG, there are many control areas, data areas, and buffers used by DLF modules. Many of these are declared in individual modules and make up a part of an internal interface between two DLF modules. Others are dynamically acquired areas that exist for a longer time span (some for the entire DLF session) and are pointed to by fields in the SPG.

## EXTERNAL INTERFACES

The library program has external interfaces to programs that can call or be called by the library program. External interfaces include:

- An interface to the host system (varying with the host environment) when DLF is invoked as a batch job.
- An interface to a program that calls DLF as a subroutine (varying with the host environment).
- An interface to the password hashing load module (DSMSPHSH) that is part of the DLF program product.
- An interface to User Exit Routines (DSMSPUXI). A skeleton DSMSPUXI is provided with the DLF program product.
- An interface to special processors that can be called during the processing of some commands (IMPORT, EXPORT, READ, and SCRIPT). Special processors are user-written or provided by other products (such as the ATMS to SCRIPT conversion processor - DSMACAIP - provided with the DCF program product).
- Interfaces to the DCF program product.
- An interface to a user-written I/O module that can be used to furnish the input data for DCF when DCF is used under DLF.

## OPERATING ENVIRONMENT

### Programming Systems

DLF Release 3 can be installed in the batch environments of the following virtual storage operating systems, provided VSAM and the Access Method Services are available:

- OS/VS2 MVS S.P. Rel. 1.3.0
- VSE AF 1.3.x or VSE AF 2.1.0 (Program Number 5747-CC1) with VSE/VSAM Release 2 (Program Number 5746-AM2).

The appearance in this publication of either "OS/VS2" or "MVS" refers to the OS/VS2 MVS system product.

The DCF Release 3 program product is required if formatting (SCRIPT) is to be done in the batch environment under DLF Release 3.

**Note:** These program products are designed to work with the specified operating system release levels and any subsequent releases and modifications unless otherwise stated.

VSAM is used for library storage and retrieval. SAM (VSE) or BSAM/QSAM (MVS) are used to read or write external data sets.

## Hardware Requirements

**Processors:** The Document Library Facility Release 3 operates on all System/370 Models 138 and above (including the 30xx series), and the 4331 and 4341 processors, as supported by OS/VS2 MVS and VSE. Floating-point hardware is required.

**Direct-Access Storage Devices:** The Document Library Facility uses VSAM to access data in its libraries. Direct-access devices supported by VSAM in the host operating system are supported by the Document Library Facility.

The Document Library Facility uses BSAM/QSAM to read or write to external data sets in the OS/VS2 environment. Direct-access devices supported by these access methods in the OS/VS2 environment are supported by the Document Library Facility.

The Document Library Facility uses SAM to read or write to external data sets in the VSE environment. Direct-access devices supported by SAM in the VSE environment are supported by the Document Library Facility. An ENVIRONMENT command is used in the VSE environment to communicate external data set characteristics to the Document Library Facility. DASD device types recognized on the ENVIRONMENT command are:

- DASD (generic)
- 2314
- 2319
- 3310
- 3330
- 3340
- 3350
- 3370
- 3375.

Other device types that are recognized on the ENVIRONMENT statement include:

- 2400
- TAPE (generic)
- 3800.

## Storage Requirements

All storage estimates shown are in addition to those required for the prerequisite System Control Program.

**Real Storage:** The Document Library Facility uses the paging facilities of the operating system to run in less real storage than the amount of virtual storage required. The factors determining how much real storage is required are numerous and dependent on the installation. Therefore, no precise statement of real storage requirements can be made. The system programmer responsible for installing the programs must establish the amount of real storage to be used.



**Virtual Storage:** The Document Library Facility requires 200K bytes of virtual storage. When the Document Composition Facility is used with the library, an additional 15K bytes of virtual storage are required in addition to the storage necessary for the Document Composition Facility.

**Direct-Access Storage:** The library load modules require 30 tracks of 3330 direct-access storage. Additional storage is required for any data conversion routines, the exact amount being determined by the individual program. The user must establish the amount of direct-access storage required for storing documents. (See "Estimating Space Requirements" in Document Library Facility Guide for more complete information about the space requirements for DLF.)

## **Environment Dependent Facilities**

Most of the DLF facilities are available in both of the environments in which DLF runs. The major exceptions are:

- Dynamic Allocation (and therefore the "DSN(xxx)" options that can be specified in the FROM and TO operands of some DLF commands) is available only in the OS/VS2 environment. (See the Appendix "JCL Statements, Dynamic Allocation, and Data Set Characteristics" in the Document Library Facility Guide for additional information on dynamic allocation.)
- Resource Access Control Facility - RACF (and therefore the RACFNAME option of the DEFINE and ALTER SYSTEM commands) is available only in the OS/VS2 environment. (See "Providing RACF Security" in Document Library Facility Guide for additional information on RACF.)
- Authorized Program Facility is available only in the OS/VS2 environment and MODESETS are issued by DLF only in the OS/VS2 environment. (See "Authorized Program Facility" in Document Library Facility Guide for additional information on the Authorized Program Facility.)
- Routing codes are available only in the OS/VS2 environment. They are used in communicating with the console operator when the OPERATOR parameter of the ARCHIVE/RETRIEVE is in effect. (See OS/VS2 MVS Supervisor Services and Macro Instructions for more information.)
- The ENVIRONMENT command is required and allowed only in the VSE environment. It is required in VSE so that DLF can identify the device type and data set characteristics of external data sets. In OS/VS2, identifying device types and data set characteristics is accomplished by reading the Job File Control Block (JFCB) and using a DCB exit to examine the data set characteristics filled in by OPEN from the JCL and data set label information.

Some functions that DLF provides for all environments are accomplished through different facilities, for example:

- I/O to external data sets is done by the DSMSPDOS module in the VSE environment. However, this task is performed by several modules in the OS/VS2 environment. The DSMSPDOS module is not provided with the DLF product for the OS/VS2 environment.
- The LIBRARY name as specified on the DEFINE SYSTEM or ALTER SYSTEM commands is used to serialize library resources during command processing. The ENQ/DEQ macros are used in OS/VS2 environments to enqueue/dequeue share or exclusive (depending on the command). The LOCK/UNLOCK macros (in conjunction with DTLs - define the lock - in the SPG) are used in the VSE environment to lock/unlock share or exclusive (depending on the command).

- Macros that provide essentially the same function in the OS/VS2 and VSE environments might have different names and calling sequences. (For example, GETMAIN/FREEMAIN for OS/VS2, GETVIS/FREEVIS for VSE, LOAD for OS/VS2, CDLOAD for VSE.)

## **STRUCTURAL OVERVIEW**

This section contains discussions about:

- Structure of the library data sets
- Structure and conventions of the DLF program
- Control flow through the library program
- Principal data and control areas
- Internal and external interfaces
- External files (data sets).

## **DOCUMENT LIBRARY STRUCTURE**

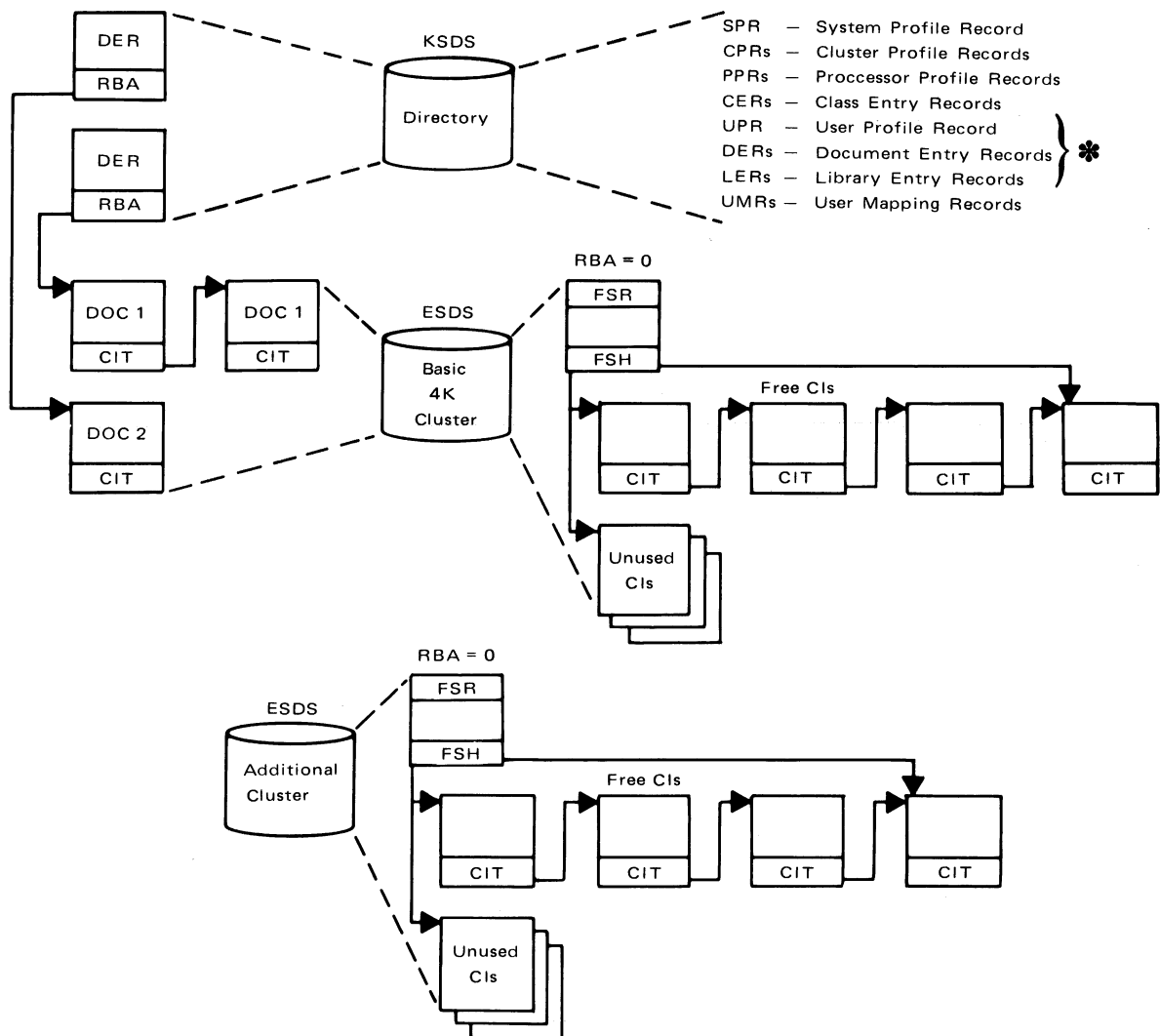
The document library consists of a directory data set and one or more source clusters as shown in Figure 4. The directory data set is a VSAM key-sequenced data set and the source clusters are VSAM entry-sequenced data sets.

### **The Library Directory (KSDS) Data Set(s)**

The document library directory data set consists of profile and entry records containing information about the library, library users, and documents belonging to each library user. There are eight types of directory records. The position of a record in the directory is determined by its key (first 30 bytes of DLF directory records).

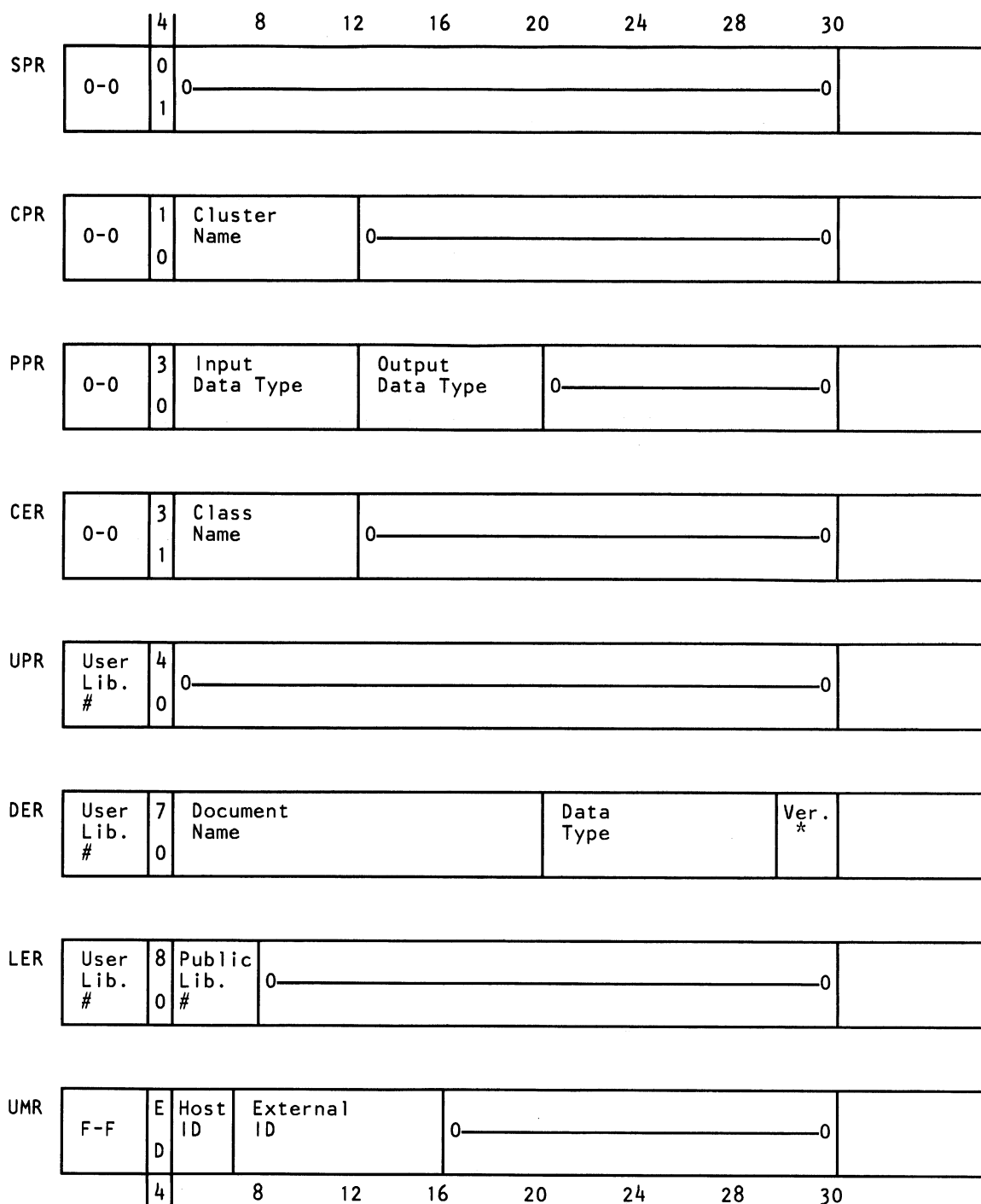
Figure 5 shows the structure of the directory record keys. The entire record structure of each directory record is shown in "Section IV - Data Areas" on page 104. Directory record types are:

- System Profile Record (SPR). There is only one SPR and it falls at the beginning of the directory and contains information global to the DLF system. The SPR is generated by the library program in response to a DEFINE SYSTEM command during the first DLF session against the library.
- Cluster Profile Records (CPRs). There is a CPR for the basic 4K source cluster as well as for each additional source cluster defined (DEFINE CLUSTER command) for this document library. CPRs fall in alphabetical sequence by cluster name following the SPR.
- Processor Profile Records (PPRs). There is one PPR for each special processor defined (DEFINE PROCESS command) to be associated with a data type pair. They fall in alphabetical sequence by indata/outdata data types following the CPRs.
- Class Entry Records (CERs). There is one CER for each class of documents defined (DEFINE CLASS command) to the document library. They fall in alphabetical sequence by class name following the PPRs. A CER associates a class name with a specific source cluster.
- The next three types of directory records are grouped. That is, the group (consisting of at least the UPR and, optionally, one or more DERs and/or one or more LERs) is repeated for each authorized user of the document library.



**Figure 4. Structure of the Document Library**

- **User Profile Records (UPRs).** There is a UPR for each user who has been defined (DEFINE USER command) to the document library. UPRs and their associated DERs and LERs (if any) fall in sequence by library number (equivalent to the user number assigned when the user is defined).
- **Document Entry Records (DERs).** There is a DER for each document in the document library. Each describes a document which is stored in a source ESDS cluster and contains a pointer to the starting point of that document. DERs fall in sequence by document name, data type, and version number (in complement form) following the UPR of the user in whose library they are stored.



\* - In complement form.

**Figure 5. Structure of Directory Record Keys**

- Library Entry Records (LERs). One or more LERs may be associated with a library user. One LER is created for each public library (other than a public library which happens to be that users own library) in which that user

owns documents. The LERs fall in sequence by public library number following the DERs which follow the UPR of the user to which the LERs apply.

- User Mapping Records (UMRs). There is a UMR for each mapping (DEFINE MAP command) of an external user to a DLF library user number. The UMRs fall in sequence by host-id/External-id following the last set of UPR/DERs/LERs.

### The Library Source Entry Sequenced Data Sets

The library source data sets (clusters) are used to store users' documents. The basic cluster (4K control interval size) is required. Additional source clusters (with control interval size ranging from .5K to 16K within VSAM constraints) are optional. Control intervals within each cluster are formatted when the cluster is defined (DEFINE CLUSTER command) or altered with additional space (ALTER CLUSTER command). This formatting consists of creating an FSR and FSH in the first control interval and creating a skeleton control interval trailer (CIT) for each control interval of the cluster other than the first. The control intervals are structured as follows:

- First control interval (RBA 0) in the cluster
  - Free Space Record (FSR). Located at the beginning of the control interval and contains the cluster time stamp.
  - Free Space Header (FSH). Located at the end of the control interval and contains pointers to the unused and free chains of control intervals within the cluster. Unused control intervals are those that have not yet been used for document storage. Free control intervals are those that were previously used for document storage but that are no longer in use.
- All control intervals other than the first control interval in the cluster contain a control interval trailer. The control interval trailers are located at the end of the control interval and may contain a pointer to the next control interval on a control interval chain. A control interval chain can be a free control interval chain or a document control interval chain depending upon whether the control interval is currently on the free control interval chain or being used for document storage.

**Note:** Control intervals in the unused chain are not chained together as are the control intervals on the free chain. Chaining is not necessary because the unused control intervals are physically sequential at the end of the cluster since they have not yet been used. Even unused control intervals have skeleton control interval trailers (created when the source cluster is formatted). Skeleton control interval trailers contain only those fields required by VSAM at the end of each control interval. The required fields are the last seven bytes of the control interval trailer. The last seven bytes of the FSH also contain the required VSAM control information.

**Note:** DLF uses control interval reads and writes in handling all input and output to the source entry-sequenced data sets. This means that DLF blocks and unblocks records in the control intervals.

## PROGRAM STRUCTURE

Three types of modules make up the structure of the Document Library Facility: load modules, source modules, and external modules.

DLF load modules include the library program, the password hashing module, and a third module that contains user exit routines.

Source modules used by DLF perform control, command validation, command initialization, command and subcommand processing, command termination, and service functions.

External load modules (those not part of the DLF product) include those called by DLF to format documents or accommodate user-created processors or programs that call DLF.

Each type of module is discussed in detail below.

### DLF Load Modules

#### DSMSPEXC The Library Program.

The term library program, used here and in other DLF publications, refers to the main load module of DLF.

The library program is a single load module, DSMSPEXC, that consists of either 64 (in the VSE environment) or 65 (in the OS/VS2 environment) link-edited DLF modules. The source module that controls the library program is within this module and is also named DSMSPEXC.

The size of this load module is approximately 120,000 decimal bytes.

#### DSMSPHSH DLF Password Hashing Module.

This load module consists of a single source program (DSMSPHSH). It is a separate load module, allowing it to be loaded separately and called by other programs and products (ATMS-III, for example) that use the document library.

The size of the load module is approximately 400 decimal bytes.

#### DSMSPUXI User Exit Module.

The DSMSPUXI load module (as distributed) consists of a single source program (DSMSPUXI) which is distributed as a skeleton (simply returns to calling routine) module with DLF. Users can link-edit their own exit routines as DSMSPUXI, replacing the distributed skeleton DSMSPUXI. See the Document Library Facility Guide for additional information on User Exits.

The size of this skeleton load module is approximately 50 decimal bytes.

## **DLF Source Modules**

The DLF source modules listed below are arranged according to the functional category to which they logically belong. You can find a detailed description of each module in "Section II - Program Functional Description" on page 46.

### **1. DLF Initialization, Control, and Termination**

The following source modules perform the initialization, control, and termination functions of the DLF library program:

- DSMSPEXC — DLF Processing Controller and Terminator
- DSMSPPIN — DLF Processing Initiator

### **2. Command Validation**

The following source modules perform command syntax checking and validation:

- DSMSPCMD — Command Decode
- DSMSPSCN — Command Scan and Parse
- DSMSPSYX — Command Syntax Graphs - Part 1
- DSMSPSYX — Command Syntax Graphs - Part 2

### **3. Command Initialization**

The following source module initializes processing of each command:

- DSMSPINT — Command Initialization

### **4. Command and Sub-Command Processors**

The following source modules process the DLF Commands. The DSMSPDEF, DSMSPPUR, DSMSP LIS, and DSMSPCPY do not actually process commands but control which of the sub-command processors are to be called to process a specific DEFINE, PURGE, LIST, or COPY command respectively.

- DSMSPACT—ACCOUNT Command
- DSMSPARC—ARCHIVE Command
- DSMSPAUT—AUTHORIZATION Command (AUTH)
- DSMSPCLO—CLOSE Command
- DSMSPCPD—COPY DOCUMENT Command
- DSMSPCPI—COPY IN Command
- DSMSPCPL—COPY LIBRARY Command
- DSMSPCPO—COPY OUT Command
- DSMSPCPU—COPY USER Command
- DSMSPCPY—COPY Commands (Controller)
- DSMSPDAC—DEFINE/ALTER CLUSTER Command
- DSMSPDAD—ALTER DOCUMENT Command
- DSMSPDAF—DEFINE/ALTER PROCESSOR Command
- DSMSPDAL—DEFINE/ALTER CLASS Command
- DSMSPDAM—DEFINE/ALTER USER MAP Command

- DSMSPDAS—DEFINE/ALTER SYSTEM Command
- DSMSPDAU—DEFINE/ALTER USER Command
- DSMSPDEF—DEFINE/ALTER Commands (Controller)
- DSMSPDOS—ENVIRONMENT Command (VSE only—  
Also provides I/O services)
- DSMSPEXP—EXPORT Command
- DSMSPIMP—IMPORT Command
- DSMSPLIC—LIST CLUSTER Command
- DSMSPLID—LIST DOCUMENT Command
- DSMSPLIF—LIST PROCESSOR Command
- DSMSPLIL—LIST CLASS Command
- DSMSPLIM—LIST USER MAP Command
- DSMSPLIS—LIST Commands (Controller) and LIST SYSTEM  
Command
- DSMSPLIU—LIST USER Command
- DSMSPPAS—PASSWORD Command
- DSMSPPRO—PROTECT Command
- DSMSPPUC—PURGE CLUSTER Command
- DSMSPPUD—PURGE DOCUMENT Command
- DSMSPPUF—PURGE PROCESSOR Command
- DSMSPPUL—PURGE CLASS Command
- DSMSPPUM—PURGE USER MAP Command
- DSMSPPUR—PURGE Commands (Controller)
- DSMSPPUU—PURGE USER Command
- DSMSPREP—READ Command
- DSMSPRET—RETRIEVE Command

#### 5. Command Termination

The following source module terminates processing of each command:

- DSMSPTRM—Command Termination

#### 6. Services

The following source modules provide services for DLF processing:

- DSMSPDSI—External Data Set Input Services
- DSMSPDSO—External Data Set Output Services
- DSMSPDYN—Dynamic Allocation Services (OS/VS2 only)
- DSMSPEXI—User Exit Controlling Services
- DSMSPGCL—Cluster Control Services
- DSMSPGNX—Get Next Record Services



- DSMSPHSH—Password Hashing Services
- DSMSPILU—Input Library I/O Services
- DSMSPLBI—Library Input Services
- DSMSPLBO—Library Output Services
- DSMSPLSV—Space Validation Services
- DSMSPSIC—Source Input Services
- DSMSPSMF—SMF Output Services
- DSMSPSOP—Source Open Services
- DSMSPTPO—Sequential Output Services
- DSMSPUXI—User Exit—Dummy Supplied

#### 7. Global Data Areas

- DSMSPPSG—DLF Global Data Area (generates SPG initial values)

### External Modules (not a part of the DLF product)

The following modules are not a part of the DLF product but can be called from DLF:

1. DSMACAIP—ATMS to SCRIPT/VS Conversion Processor (part of DCF)

This load module can be called as a special processor by DLF during the processing of IMPORT, EXPORT, READ, or SCRIPT commands. The actual DLF modules within the library program which call DSMACAIP are DSMSPSDI (IMPORT processing), DSMSPSDO (EXPORT processing), or DSMSPSIC (READ or SCRIPT processing). DLF modules do not actually know they are calling DSMACAIP, they just load and call the special processor specified or implied in the command.

See Document Library Facility Guide for additional information on special processors. See "External Interfaces" on page 24 in this manual for additional information on the interface to special processors.

2. DSMLOS30 (OS/VS2) DSMDLS30 (VSE)—SCRIPT/VS Command Processor (part of DCF)

Depending on the operating system, the DSMSPEXC module of the library program calls either DSMLOS30 or DSMDLS30 to process a SCRIPT command. DSMSPEXC then:

- Passes the address of the SCRIPT command
- Passes the address of DLF service modules that DCF needs to process SCRIPT commands.

DSMLOS30 (or DSMDLS30) module is loaded when the first SCRIPT command of a DLF session is to be processed. The names "DSMLOS30" and "DSMDLS30" are hard-coded in DLF.

See "External Interfaces" on page 24 for additional information about the interface(s) between DLF and DCF.

3. **DSMSPUXI** — User-Created Exit Code (overlays the skeleton module provided with DLF)

The DSMSPUXI load module is called by DLF command processors (via DSMSPEXI which functions as a user exit controller). DSMSPUXI is called during the processing of certain commands at points where a user wants to analyze (and perhaps modify) certain data in the DER, UPR, and/or UMR.

See Document Library Facility Guide for a complete discussion on all the DLF command processor exits and what is available for the user to modify at that exit. See "External Interfaces" on page 24 in this section for additional information on the interface to the user exit routines (DSMSPUXI).

**Note:** These user written routines replace the skeleton module supplied with DLF as a part of the product. The name of the load module containing these routines must be DSMSPUXI since that is what is loaded (by DSMSPPIN during DLF initialization).

4. **XXXXXXXX** — User-Created Special Processors

Special processors can be called during processing of the IMPORT, EXPORT, READ, or SCRIPT commands. They get control each time a line (SLE/record) has been read. They can modify, delete, or add lines. See comments under DSMACAIP above and "External Interfaces" on page 24 for more information.

5. **XXXXXXXX** — User-Created Calling Program (calls DLF as subroutine)

User-written programs can call DLF to process commands. In addition to the DLF commands available when DLF is invoked as a batch job, the READ and CLOSE commands can also be issued when DLF is called as a subroutine. This program passes DLF one command at a time using the interface procedures described in "External Interfaces" on page 24.

(See Document Library Facility Guide for detailed information about user-created programs that call DLF as a subroutine.)

6. **XXXXXXXX**—User Created I/O module to Supply Input for SCRIPT.

When DLF is called as a subroutine to process a SCRIPT command, the calling program can specify the name of a user written I/O module that is to supply input data to be formatted (SCRIPT). This I/O module is loaded and called by the source input subsystem of DLF when DCF calls that subsystem to acquire input data.

See "External Interfaces" on page 24 for additional information about the interface between the DLF source input subsystem and a user written I/O module.

## PROGRAM CONVENTIONS

The following section describes under separate headings the conventions DLF uses to set registers, name modules, and communicate between the modules of the the library program.

## Register Conventions

The Document Library Facility uses these register conventions:

|               |   |  |
|---------------|---|--|
| Register 0    | - | Parameter register                     |
|               | - | Reason Code on return (where defined)  |
|               | - | Message number                         |
| Register 1    | - | Parameter register                     |
| Registers 2-9 | - | Work registers                         |
| Register 10   | - | Commonly a second code base register   |
| Register 11   | - | Base register for SPG                  |
| Register 12   | - | First code base register               |
| Register 13   | - | Address of Register Save Area          |
| Register 14   | - | Return address                         |
| Register 15   | - | Entry point address                    |
|               | - | Return code on return (where defined). |

## Module Naming Conventions

The program modules of the Document Library Facility (including its load modules) begin with the letters DSMSP and have a three-character descriptive suffix that uniquely identifies each module. DLF macros begin with DSMG and have a three-character suffix.

## Communication Between Modules of the Library Program

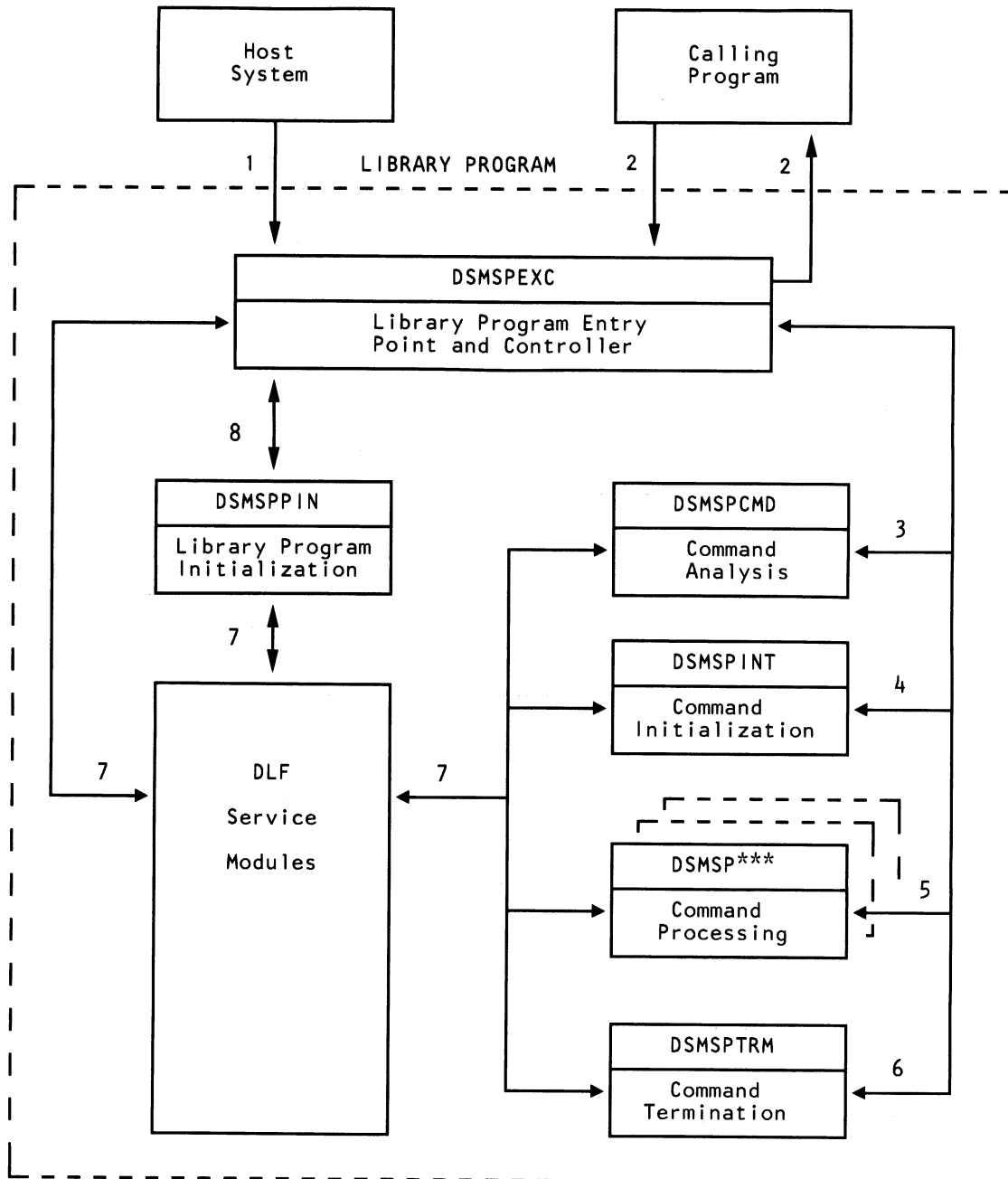
The library global table (SPG), a data and control area within the program, is used extensively for communication among the modules of the library program. In addition to the SPG, other control blocks and parameter lists are often used to pass information from one module to another. "Principal Data and Control Areas of the Library Program" on page 21, "External Interfaces" on page 24, and "Section IV - Data Areas" on page 104 offer detailed descriptions of these other areas and how DLF uses them.

## CONTROL FLOW WITHIN THE LIBRARY PROGRAM

The library program is a single load module named DSMSPEXC that consists of either 64 (VSE environment) or 65 (OS/VS2 environment) link-edited DLF modules. The controlling module of the library program is also named DSMSPEXC.

The library program can be invoked as a batch job (1 in Figure 6) or it can be called as a subroutine (2 in Figure 6) by another program. In either case, the DSMSPEXC module gets control and calls the DSMSPPIN module (8 in Figure 6) to perform DLF initialization. Initialization includes opening library and system data sets. System data sets include the input stream data set (SYSIN for OS/VS2 environments and SYSRDR for VSE environments) and the output/message data set (DSMLIST for OS/VS2 environments and SYSLST for VSE environments). When DLF initialization is complete, DSMSPPIN returns to DSMSPEXC.

DSMSPEXC is now ready to control the processing of commands. If DLF was invoked as a batch job, then the commands will be read one at a time from the job input stream. If DLF was called as a subroutine, then a single command will have been passed to DLF by the calling program. In either case, DSMSPEXC moves the command to a buffer in the SPG table (a global data and control table within the library program) where it will be accessible by other library program modules. Once DSMSPEXC isolates a



Note: DSMSP\*\*\* represents any one of the DLF Command Processors.

Figure 6. Control Flow Within the Library Program

command, it controls the analysis, initialization, processing, and termination of that command. Each step is accomplished by calling another module within the library program.

DSMSPEXC calls DSMSPCMD (3 in Figure 6) to control command parsing and analysis. On return from DSMSPCMD, various fields

in the SPG (DLF Global Table) are set to the parameter values in the command. Other SPG flag fields are set to indicate which operands and parameters were seen in the command.

DSMSPEXC then calls DSMSPINT (4 in Figure 6) to initialize processing for this command. Initialization includes opening any external data sets (as specified in the command) required to process the command. DSMSPINT returns to DSMSPEXC when initialization is complete. DSMSPEXC then calls the appropriate command processor or command processor controller (5 in Figure 6) to handle the processing requested by the command.

**Note:** A command processor handles the command directly. A command processor controller does preliminary processing common to a group of commands (such as the LIST, DEFINE, COPY, ALTER, or PURGE command groups) and then calls the appropriate command processor to complete processing.

The command processor or command processor controller returns to DSMSPEXC when processing is complete.

DSMSPEXC then calls DSMSPTRM (6 in Figure 6) to end processing for this command. This includes closing any external data sets opened by DSMSPINT while initializing the command. DSMSPTRM then returns to DSMSPEXC.

DSMSPEXC returns to the calling program (2 in Figure 6) for the next command (if DLF was called as a subroutine) or reads the next command from the job input stream (if DLF was invoked as a batch job). This command processing loop continues until DLF detects a terminating error or until there are no more commands to be processed. End of file on input stream (if DLF is invoked as a batch job) or a CLOSE ALL command (if DLF was called as a subroutine) signals that there are no more commands to be processed.

When there are no more commands to be processed, DSMSPEXC closes the library and system data sets and returns to the host environment (if invoked as a batch job) or to the calling program (if called as a subroutine).

All DLF modules that perform control, analysis, initialization, command processing, or termination functions make use of DLF service modules that are a part of the library program (7 in Figure 6) and perform commonly required services such as library and external I/O.

**Note:** DLF can be called many times by the calling program (once for each command to be processed) during a DLF session. DSMSPEXC sets flags on the first call so that it will not call DSMSPIN for DLF initialization on subsequent calls. DSMSPEXC resets these flags when a CLOSE ALL command terminates the session.

## PRINCIPAL DATA AND CONTROL AREAS OF THE LIBRARY PROGRAM

The library global table (SPG) is an area within the library program that contains data areas, flags, control blocks, buffers, pointers to loaded programs, pointers to dynamic areas, and other data and control information frequently needed for DLF processing. Constituting the DSMSPSPG module, the library global table is link-edited as part of the library program.

When DLF is invoked (or called as a subroutine), register 11 is set by DSMSPEXC to the address of the SPG and normally remains set for the entire DLF session. Some DLF modules (DSMSPDYN, DSMSPQNM, DSMSPDOS, and DSMSPSIC, for example) might get control from the Document Composition Facility (DCF) when DLF calls DCF to process a SCRIPT command. These modules must reset register 11 to the address of the SPG when DCF calls them for services. Register 11 does not point to the SPG when a user exit is called.

In addition to the SPG, there are many control areas, data areas, buffers, and parameter lists which are used by DLF modules. Many of these are declared in individual modules and comprise a part of an internal interface between two DLF modules. Other areas, however, are dynamically acquired, active over a longer time span (some for the entire DLF session), and pointed to by fields in the SPG. Figure 7 on page 23 shows the principal control and data areas that are based on pointers in the SPG.

The SPGCLIFP and the SPGCLOFP fields of the SPG point to the first Cluster Control Block (CLB) on the input and output CLB chains respectively. Each CLB on the input chain controls an ESDS cluster that is open for input. Each CLB on the output chain controls an ESDS cluster that is open for output. The DSMSPGCL module (part of the library program) builds these CLBs when the clusters are opened. The basic 4K cluster (input and/or output) is opened at DLF initialization when DSMSPGCL is called by DSMSPPIN. Other clusters can be opened by DSMSPGCL when it is called by another library program module to activate a cluster which is not currently open.

The SPGCLBIA field of the SPG points to the input CLB that controls the ESDS cluster currently active for input. The SPGCLBOA field of the SPG points to the output CLB which controls the ESDS cluster which is currently active for output. These SPG fields are set by DSMSPGCL when it is called by another DLF module to activate a cluster. Activating a cluster can include opening the cluster if it is not already open.

Each CLB resides in a separately acquired (via GETMAIN or GETVIS) dynamic storage area. The areas are acquired by DSMSPGCL when a new CLB is built.

Figure 7 also shows the Open File Name (OFN) table pointed to by the SPGSICSP field in the SPG. The OFN resides in a separately acquired dynamic storage area acquired by DSMSPINT when that module is initializing for the first SCRIPT or READ command of a DLF session. The area is freed by DSMSPEXC at DLF termination.

The OFN table is used to control the access of data from up to 16 sources concurrently during SCRIPT command processing. It is also used when DLF is called as a subroutine and the calling program uses repetitive READ commands to access data from up to 16 documents concurrently.

The OFN consists of two arrays, each having 16 elements. The nth element in the first array is associated with the nth element in the second array and together these elements are considered an OFN entry.

The first array element of an active OFN entry (its source is open) contains a pointer to a dynamic storage area used during data access from the source identified in the second array element of the entry. This dynamic area is acquired when the entry is activated and contains subareas that are pointed to by fields of the second array element of the OFN entry. Subareas include a communication area, a work area, a source buffer, a DCB area, and/or a document entry record (DER) area. The size of these areas depends upon the source from which data will be obtained. For the READ command, the source is always a document stored in the document library. For the SCRIPT command, the source can be a document in the document library, an external data set, or (if DLF called as a subroutine) an I/O module specified by the calling program.

Figure 7 also shows the SPG16K — a 16K buffer area within the SPG that is pointed to by the SPG16KB field. The SPG016KB field points to another 16K buffer area acquired as dynamic storage by DSMSPPIN at DLF initialization.

The source module components of the library program interface and communicate with each other through parameter lists and data

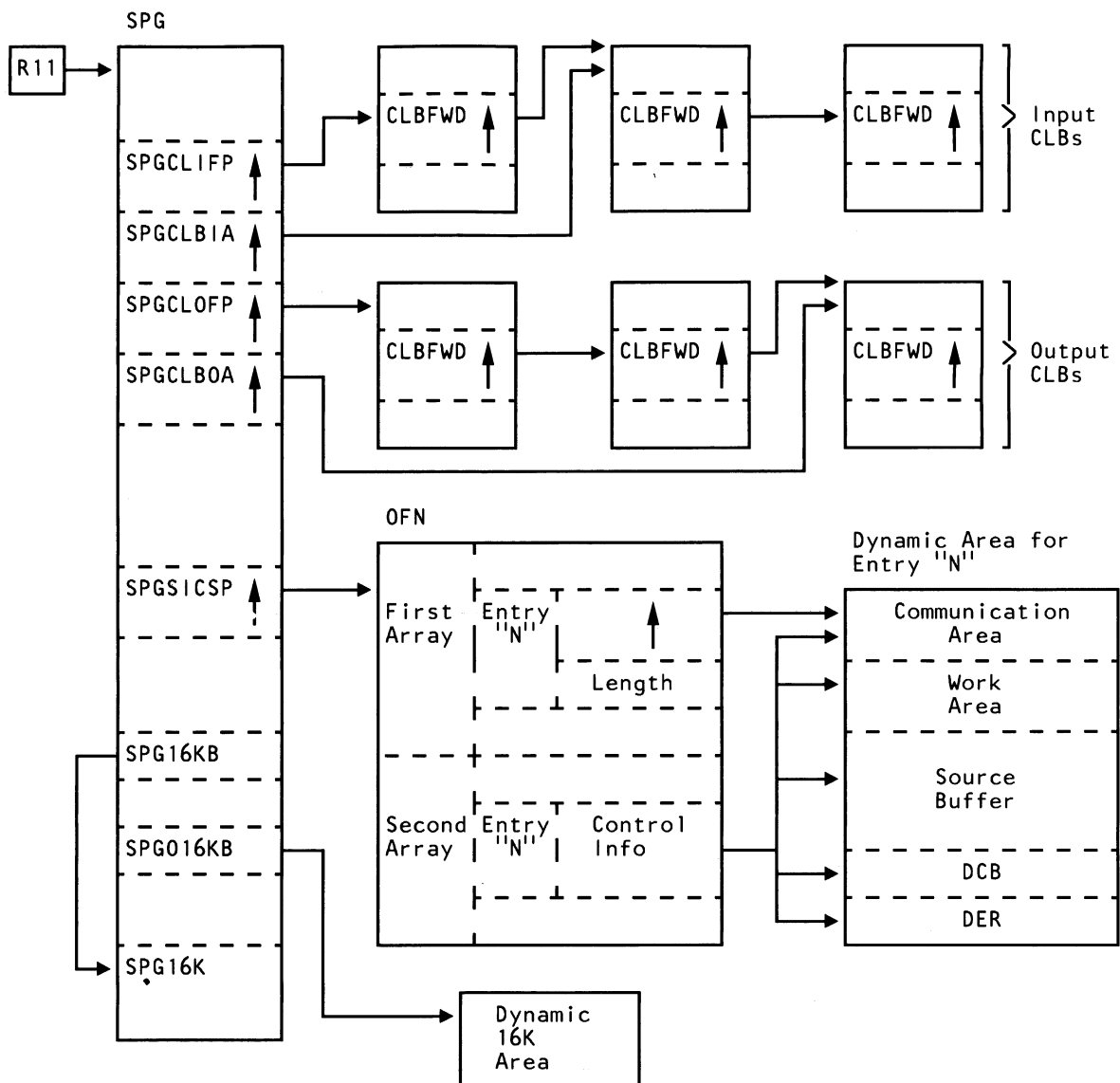


Figure 7. Principal Control and Data Areas

and control areas. The major areas are described in detail in "Section IV - Data Areas" on page 104.

#### INTERNAL DLF INTERFACES

DLF uses the SPG extensively for communication between the modules of the library program. Other data areas and parameter lists (the Cluster Control Block — CLB, the Open File Name table — OFN, and the list processor common work area — LST) are sometimes required. These areas are described in "Section IV - Data Areas" on page 104. Their use by a particular function is described in "Section II - Program Functional Description" on page 46.

## EXTERNAL INTERFACES

The external interfaces shown in Figure 8 on page 25 are first discussed briefly to give an overview of the external interfaces. A detailed discussion of each of these interfaces follows this overview.

The library program can be invoked as a batch job (1 in Figure 8) or as a subroutine (2 in Figure 8) by another program. If DLF is invoked as a batch job, then it reads commands from the job input stream. If DLF is called as a subroutine, then the calling program passes the commands to DLF one at a time (one for each call) along with other controlling information.

DSMSPHSH is a load module that is part of the DLF product. It is link-edited as a separate module so that it can be used by other application systems (such as ATMS-III — 8 in Figure 8) as well as by the modules of the library program. Library program modules which need to hash passwords branch (BALR) to DSMSPHSH (7 in Figure 8). The entry point of DSMSPHSH is saved at SPGSHAD when DSMSPHSH is loaded by DSMSPPIN at DLF initialization.

DSMSPUXI is a load module containing user exit routines. A skeleton DSMSPUXI module is distributed with the DLF product. DSMSPPIN loads DSMSPUXI at DLF initialization time and its entry point is saved in the SPG at SPGUXIAD. DLF command processing modules that provide user exits invoke DSMSPUXI via DSMSPEXI (the user exit controller within the library program) which in turn branches (BALR — 6 in Figure 8) to the entry point address in SPGUXIAD.

Some commands (IMPORT, EXPORT, SCRIPT and READ) might request the invocation of special processors during command processing. Special processors of this type (5 in Figure 8) get control when:

- A source line entry (SLE — a data record in DLF format) is read from a document in the library (for EXPORT, SCRIPT and READ commands)
- A data record is read from the external source data set (for IMPORT and SCRIPT commands)
- A data record is returned to DLF by a user-written I/O module (for SCRIPT commands only when DLF is called as a subroutine).

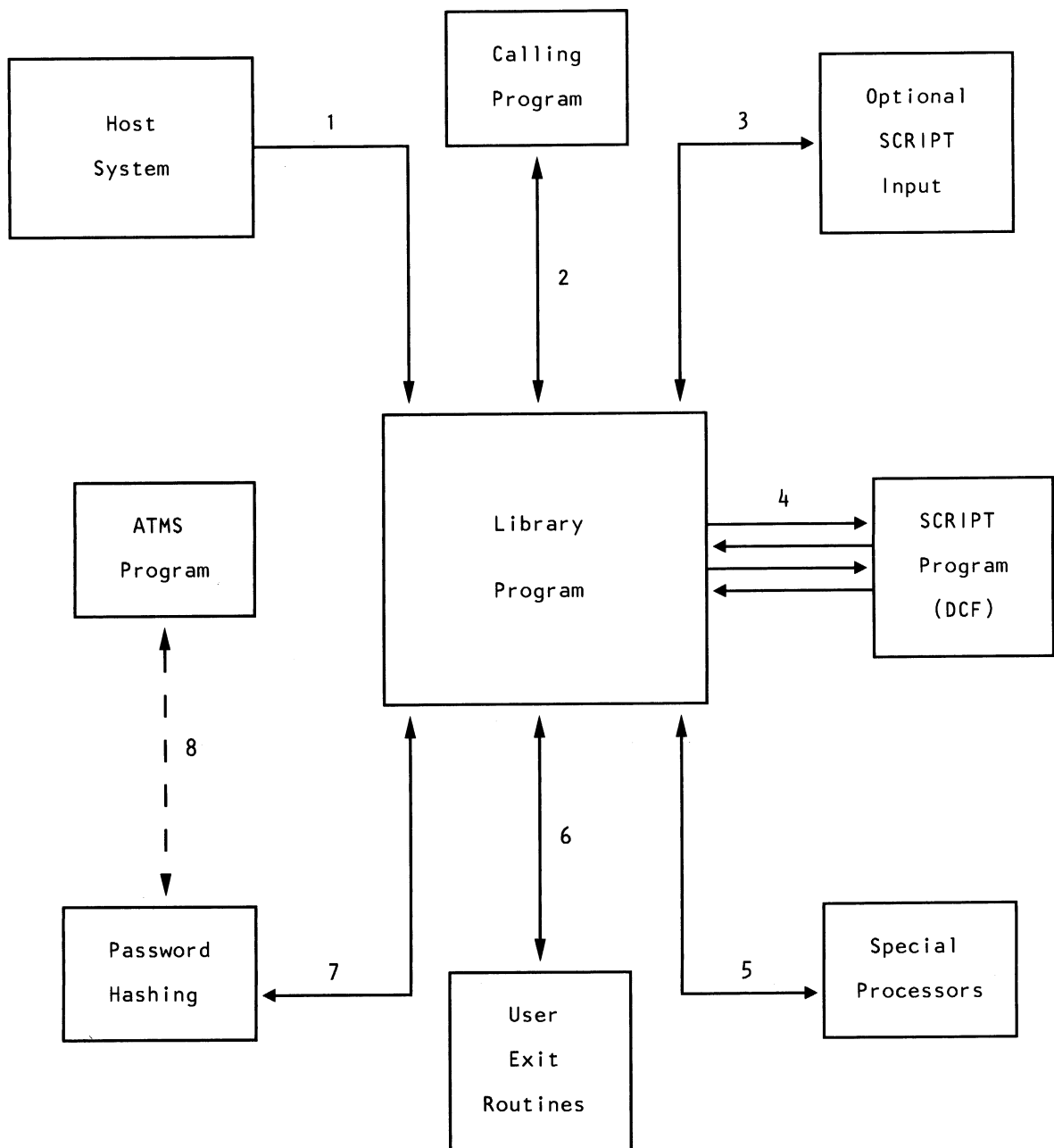
Special processors can add, delete, or modify data during command processing. No special processors are provided as a part of the DLF product. However, DSMACAIP (the ATMS to SCRIPT conversion processor) is provided as part of the Document Composition Facility program product. You can also specify special processors provided by other products or those you write yourself. The name of the processor is either specified or implied by information in the command being processed.

For DLF to process SCRIPT commands, the Document Composition Facility must also be installed. There is a double interface between DLF and DCF (4 in Figure 8) that allows:

- DLF to call DCF to syntax check, analyze, and process a SCRIPT command
- DCF to call DLF for dynamic allocation (OS/VS2 only) and I/O services while processing a SCRIPT command.

Called as a subroutine, DLF can process SCRIPT commands that require data input from an I/O module (3 in Figure 8). When DCF calls DLF to provide input data, instead of reading data from the document library or an external data set, DLF calls the I/O module whose entry point name is specified by the calling program. The module provides the data which DLF then passes back to DCF for processing by SCRIPT/VS.





**Figure 8. DLF External Interfaces**

---

## Batch Invocation Interface

As shown in Figure 9 on page 27, when the library program gets control it must first determine whether DLF has been invoked as a batch job or called as a subroutine. How the DSMSPEXC module of the library program does this depends on the host environment:

- OS/VS2 environments. The library program checks the high order bit of the fullword field pointed to by register 1. If this bit is 1, DLF has been invoked as a batch job.
- VSE environments. The user area of the VSE communication region is checked (its addressability acquired by the COMRG macro). If this field does not contain a "C," DLF has been invoked as a batch job.

After determining that the library program has been invoked as a batch job, DSMSPEXC sets on the SPGEXEQD flag in the SPG in order to have a ready indicator for modules that must know how DLF got control.

DSMSPEXC then calls DSMSPPIN to initialize DLF. One of the initialization tasks DSMSPPIN performs is determining whether or not the LIST option is in effect. How DSMSPPIN does this depends on the host environment:

- OS/VS2 environments. Register 1 points to the PARM field specified on the EXEC statement. The first two bytes of the PARM field contain the length of the PARM. The remainder of the field is the PARM itself and for DLF should be LIST, NOLIST or hex zeroes (null). If it is LIST, the SPGLIST flag in the SPG is set on to have a ready indicator for those modules that must know if the LIST option is in effect.
- VSE environments. The Communication Region (addressable through the COMRG macro) contains a COMRLSTX flag byte that, when on, indicates that LISTX was specified on the OPTIONS JCS. If the COMRLSTX flag is on, SPGLIST is also set on to indicate the LISTX specification.

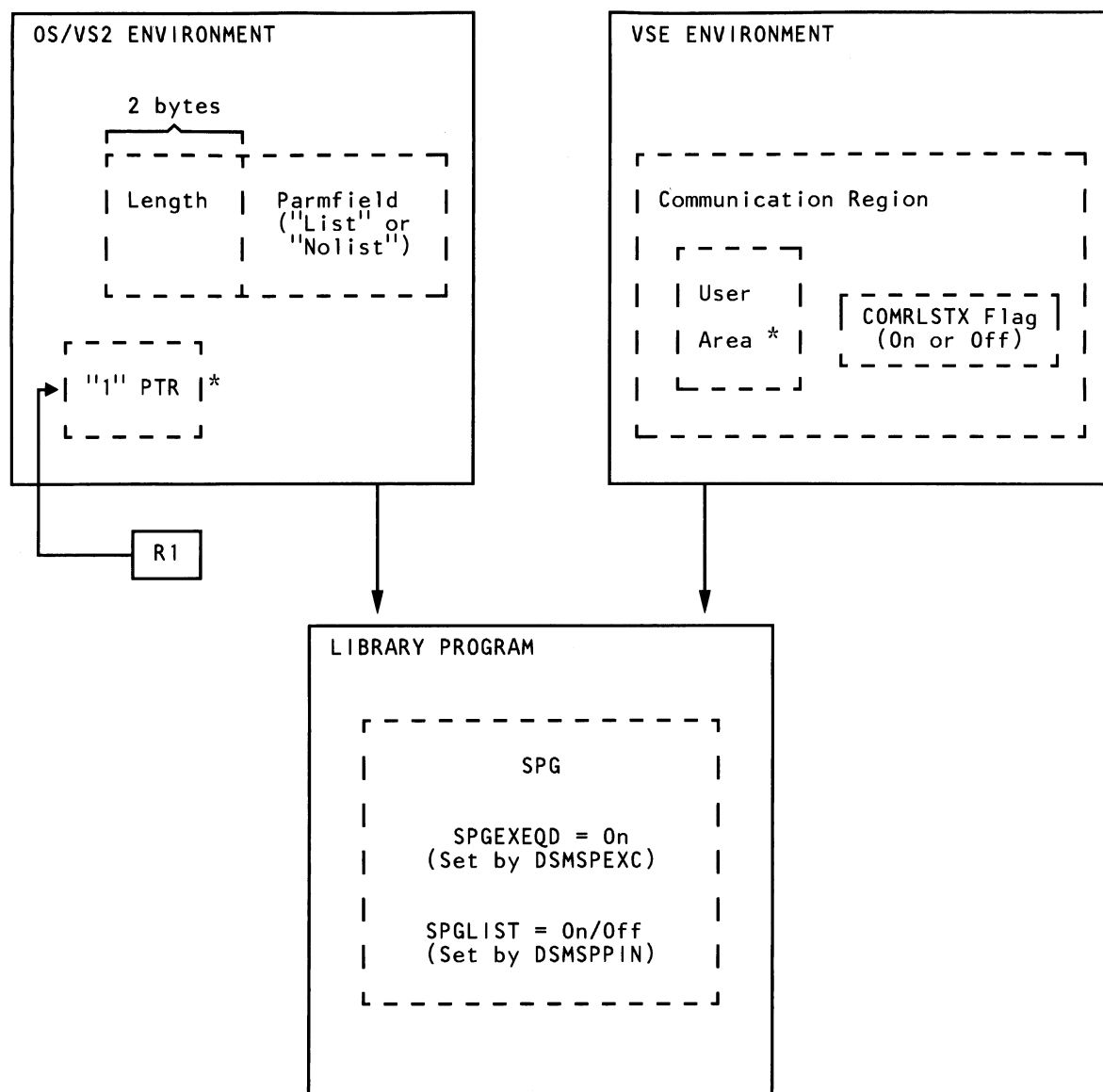
## DLF Called as a Subroutine - Interface

As shown in Figure 10 on page 28, when the library program gets control it must first determine whether DLF was invoked as a batch job or was called as a subroutine. How the DSMSPEXC module of the library program does this depends on the host environment:

- MVS environments. DSMSPEXC checks the high order bit of the fullword field pointed to by register 1. If this bit is "0", then DLF was called as a subroutine.
- VSE environments. DSMSPEXC checks the user area of the VSE Communication region (addressable through the COMRG macro). If this field contains a "C," then DLF was called as a subroutine.

When it has been determined that the library program was called as a subroutine, DSMSPEXC sets the SPGEXEQD flag off in the SPG in order to have a ready indicator for those modules that must know how DLF got control. DSMSPEXC then saves the addresses passed (pointed to by register 1) to DLF by the calling program in the SPGCALED fields. There is room for six addresses in the SPGCALED fields. DSMSPEXC also saves the contents of register 1 in SPGCALR1.

DSMSPEXC then calls DSMSPPIN to initialize DLF. Among the initialization functions performed by DSMSPPIN, is to determine whether or not the LIST option is in effect. To do this, DSMSPPIN checks the field pointed to by the first address passed by the calling program. This field is expected to contain



**\*Notes:**

1. In the OS/VS2 environment, if there is a "1" in the high order bit of the full word pointed to by register one, DLF was called as a batch job.
2. In the VSE environment, if the user area in the communication region does not contain a C, DLF was called as a batch job.

**Figure 9. DLF Batch Invocation Interface**

either LIST or NOLIST. If neither is specified, NOLIST is assumed. If LIST is specified, then the SPGLIST flag in the SPG is set on to have a ready indicator for those modules that must know if the LIST option is in effect.

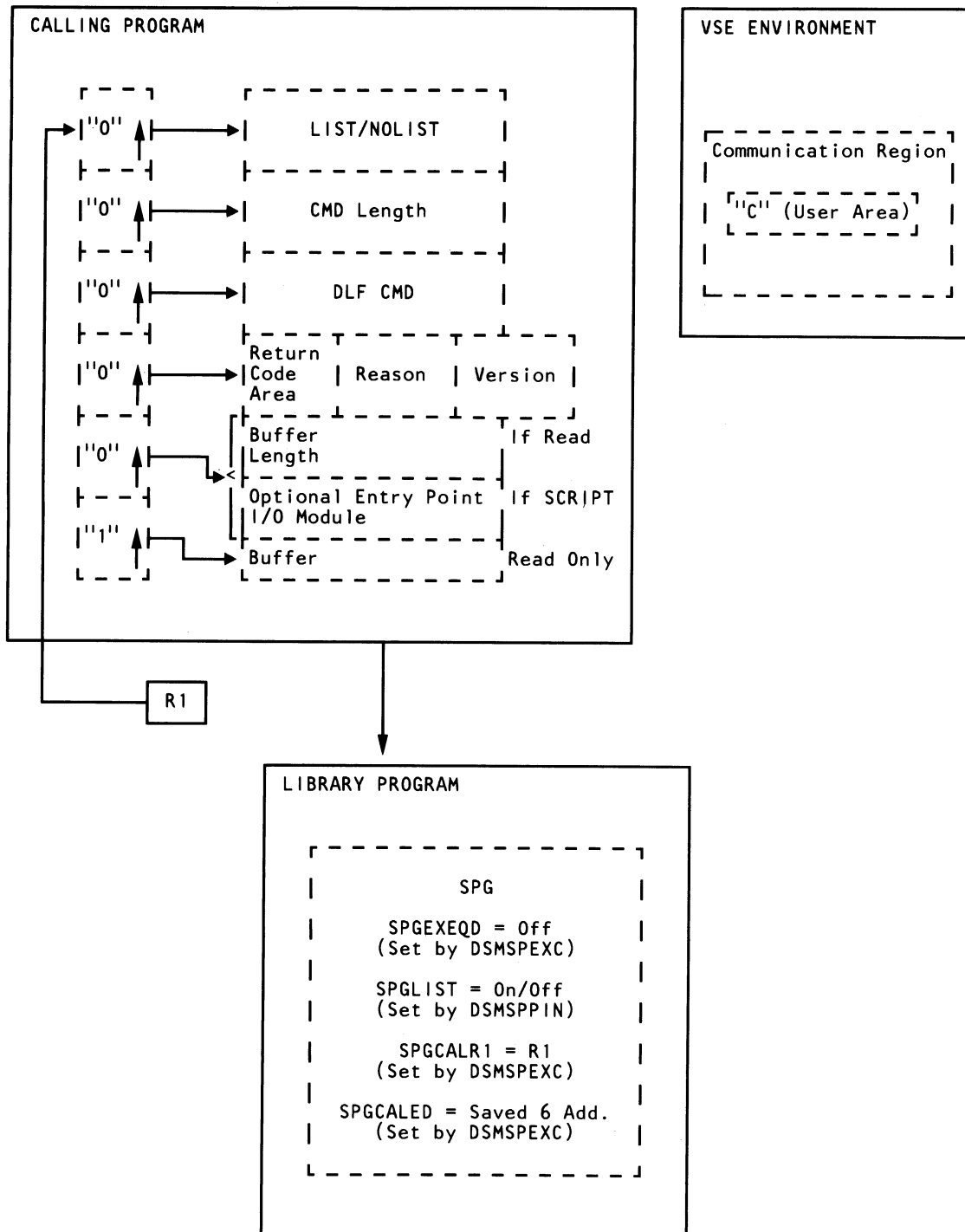


Figure 10. DLF Called as a Subroutine - Interface

Any command that is allowed when DLF is invoked as a batch job is valid when DLF is called as a subroutine. In addition, the READ and CLOSE commands (not valid if DLF is invoked as a batch job) are allowed.

One command is passed (pointed to by the third address passed by the calling program) and processed each time DLF is called. The

first time DLF is called (or the first time after a CLOSE ALL command), DSMSPEXC calls DSMSPPIN for DLF initialization prior to processing the first command. Subsequent calls process one command. The command length is specified in the 2-byte field pointed to by the second address passed and must not exceed 1020 bytes.

DLF returns a return code (highest for the command) in the field pointed to by the fourth address passed by the calling program each time it returns to the calling program.

If the calling program set the field pointed to by the fourth address to "10F3" prior to calling DLF, then DLF expects three fullword fields pointed to by the fourth address:

1. The return code field as described above
2. A field in which the reason code (message number of first error message for the command - if any) is returned to the caller
3. A field in which the version number of a versioned document just imported (IMPORT command only) is returned to the caller.

If the reason code information and/or the version information is desired, the calling program must set the first field to "10F3" prior to each call of DLF.

If the command is a READ command, the calling program is requesting that the first/next record (properly called Source Line Entry - SLE - see "Section IV - Data Areas" on page 104 for a discussion of SLEs) of a document in the DLF library be returned to it in the buffer pointed to by the sixth passed address. The length of this buffer is in the halfword field pointed to by the fifth address.

If the command is a SCRIPT command, then the calling program is requesting that DLF call DCF to process a SCRIPT command. Optionally, the calling program might specify that data to be formatted is to be furnished by an I/O module (user written) instead of coming from the DLF library or from an external data set. If this option is elected, the fifth address passed by the calling program will point to an eight-byte field containing the entry point name of the I/O module. When DLF is subsequently called by DCF for input services, it will call this I/O module to furnish the input rather than reading the data from a specified external data set or a document in the DLF library.

**Note:** The fifth address is required only for the READ command and is ignored for all commands other than READ or SCRIPT. The sixth address is ignored for all commands other than READ.

See "Section IV - Data Areas" on page 104 for a detailed description of each parameter pointed to by the addresses passed by the calling program.

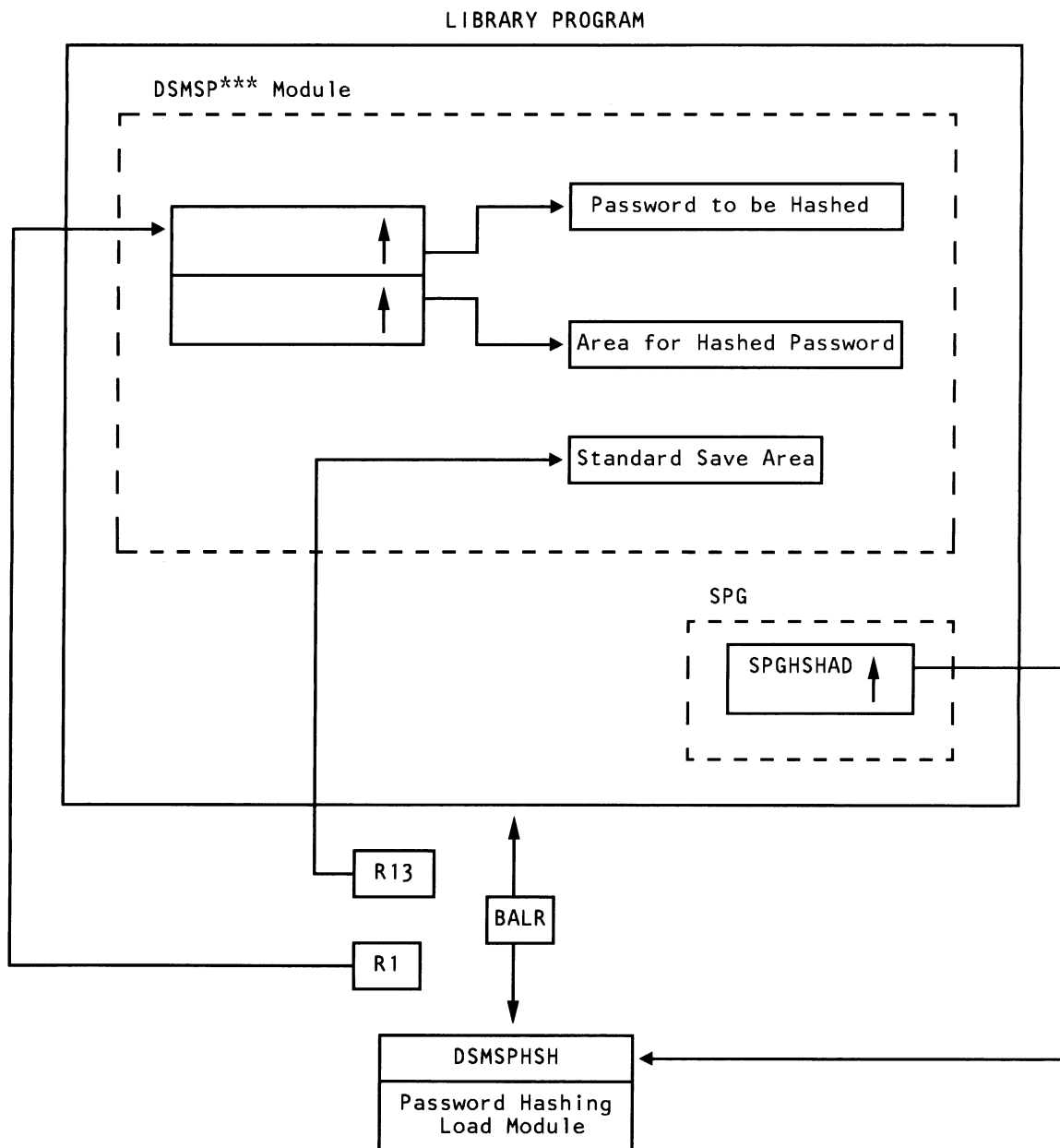
## **Password Hashing Interface**

DSMSPHSH, a part of the DLF program product, is a separate load module (not a part of the library program load module) and is illustrated in Figure 11 on page 30.

DSMSPHSH was made a separate load module so it can be used by other products (e.g., ATMS-III) that access documents in the library.

DSMSPPIN loads DSMSPHSH during DLF initialization. The entry point address is stored in SPGSHAD by DSMSPPIN for subsequent use by other library program modules.

There are four library program modules (DSMSPARC, DSMSPINT, DSMSPRET, and DSMSPSOP) that branch (BALR) to DSMSPHSH to hash



Note: DSMSP\*\*\* may be DSMSPINT, DSMSPARC, DSMSPRET, or DSMSPSOP

Figure 11. DLF Password Hashing Interface

either a user password or a document password. These passwords are stored in hashed form for additional security.

When DSMSPHSH is called (by branch — BALR — to an address stored in SPGSHAD), register 1 must point to a list of two addresses that point to required parameters. The first address points to an eight-byte field containing the password to be hashed and the second address points to an eight-byte field to which the hashed password is to be returned.

Other programs (e.g., ATMS-III) that use DSMSPHSR must load and/or call the module and match the interface as described above.

## User Exit Interface

The interface to DSMSPUXI (a load module containing user exit routines) is illustrated in Figure 12 on page 32. A skeleton DSMSPUXI module is distributed with the DLF product. DSMSPPIN loads DSMSPUXI at DLF initialization and its entry point is saved in the SPG at SPGUXIAD.

DLF command processing modules that provide user exits invoke DSMSPUXI through DSMSPEXI (the user exit controller within the library program), which branches (BALR) to the entry point address saved in SPGUXIAD.

There are nine library program modules that call DSMSPEXI to branch to DSMSPUXI for user exit processing:

- DSMSPARC
- DSMSPDAM
- DSMSPDAU
- DSMSPEXP
- DSMSPIMP
- DSMSPPAD
- DSMSPPUM
- DSMSPPUU
- DSMSPRET.

DSMSPEXI is the user exit controller and masks system-sensitive data (i.e., the SPG) from the user exit and also provides output message services on return from the user exit if a message is requested.

When DSMSPEXI is called, register 1 must point to a fullword field containing the address of the EXI parameter list (see "Section IV - Data Areas" on page 104 for a description). Calling programs use DSMGEXI to generate the structure of this parameter list.

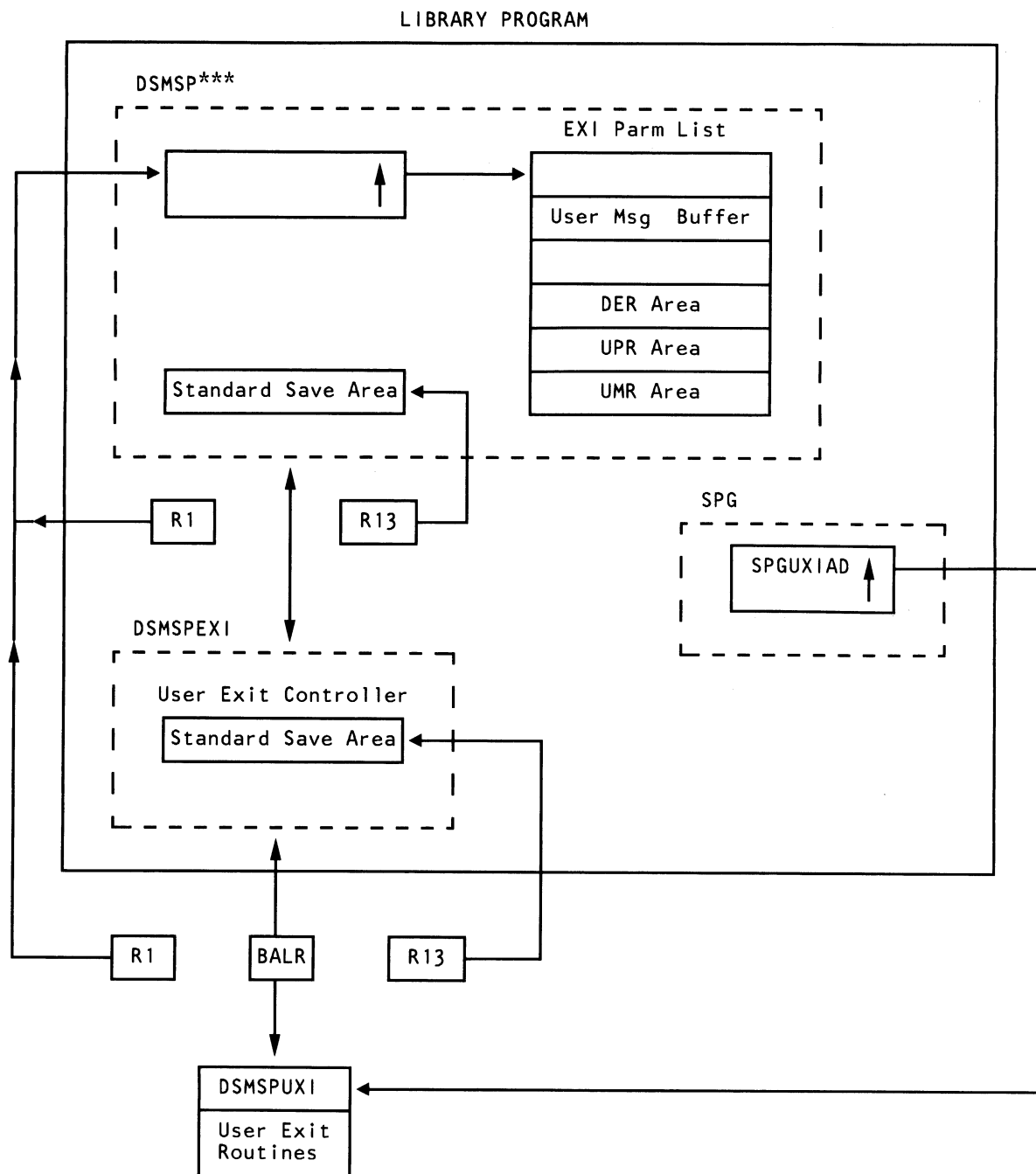
DSMSPEXI saves register 11 (pointer to the SPG) in its data area and clears register 11 prior to calling the user exit routines. DSMSPEXI also blanks out any passwords in the DER or UPR (within the EXI parameter list) prior to calling the exit routines.

DSMSPEXI then branches (BALR) to DSMSPUXI with register 1 set as when DSMSPEXI was entered. The entry point address of DSMSPUXI is located in SPGUXIAD.

The Document Library Facility Guide contains a chapter about user exit routines that describes:

- Each command processing module exit
- What is available to the user exit at that point
- What can be modified for that exit.

The Document Library Facility Guide also explains how the user exit can specify a return code and furnish a message for output through DLF facilities.



Note: DSMSPEX1 may be DSMSPARC, DSMSPDAM, DSMSPPAU, DSMSPEXP, DSMSPPIM, DSMSPPUD, DSMSPPUM, DSMSPPUU, or DSMSPPRET

Figure 12. DLF User Exit Interface



## Special Processor Interface

Special processors can be called by DLF during processing of certain commands (IMPORT, EXPORT, READ and SCRIPT). If special processors are specified or implied in the command, they get control for each line of data (SLE/record) handled by DLF during command processing.

Special processors are not part of the DLF program product. Another product can furnish a special processor (e.g., the ATMS to SCRIPT conversion processor, DSMACAIP, that is part of the Document Composition Facility) or it can be user-written.

Special processors can be specified explicitly by their entry point name within the IMPORT, EXPORT, READ, or SCRIPT command. They can also be implied if an INDATA/OUTDATA (document data type) pair as specified or implied in the command is associated with a processor profile record (See "Document Library Structure" on page 10.) In this case, the processor profile record (PPR) can specify the entry point name of a special processor to be used during command processing.

**Note:** Data type is a document characteristic that often indicates the type of formatting controls contained within that document. If a document's data type is to change during processing, then a conversion of some type is implied. That is, a document being imported having an INDATA type of ATMS and an OUTDATA (DATA as specified on the IMPORT command) type of SCRIPT implies that during the import we are converting ATMS controls to controls that SCRIPT will recognize. That's exactly what happens if we define a PPR with the data pair of ATMS and SCRIPT and associate that pair with the DSMACAIP (part of the DCF product) special processor.

When a special processor gets control, register 1 must point to a list of three addresses; each of which points to another address. These second level addresses point as follows:

- Address 1 — pointer to a parameter list (IPP parameter list - see "Section IV - Data Areas" on page 104)
- Address 2 — pointer to a communication buffer (4K or CI size)
- Address 3 — pointer to a 4K work area

The special processor always sees this interface regardless of which command (IMPORT, EXPORT, READ, or SCRIPT) is being processed. The set up of this interface by DLF modules, however, differs significantly depending upon the type of command being processed.

Figure 13 shows how DSMSPDSI (external I/O service module used by DSMSPIMP during IMPORT processing) or DSMSPDSO (external I/O service module used by DSMSPEXP during EXPORT processing) sets up the address pointers for a special processor. The module branches (BALR) to the entry point address of the special processor each time it acquires another data-record/SLE. When DSMSPIMP or DSMSPEXP loads the special processor, the the entry point address of that special processor is stored in the SPGIPENT field of the loading module.

Figure 14 shows how DSMSPGNX (the get next module of the Source Input subsystem of DLF) sets up the address pointers for a special processor. Remember that only the parameter list is declared within the module. The communication buffer and the work area are both in a dynamic storage area associated with an OFN (see "Section IV - Data Areas" on page 104) entry that controls the source currently being used to provide input for READ or SCRIPT command processing. The OFN entry also includes a field (OFNIPENT) containing the address of the special processor (loaded by DSMSPSOP) when the source is opened.

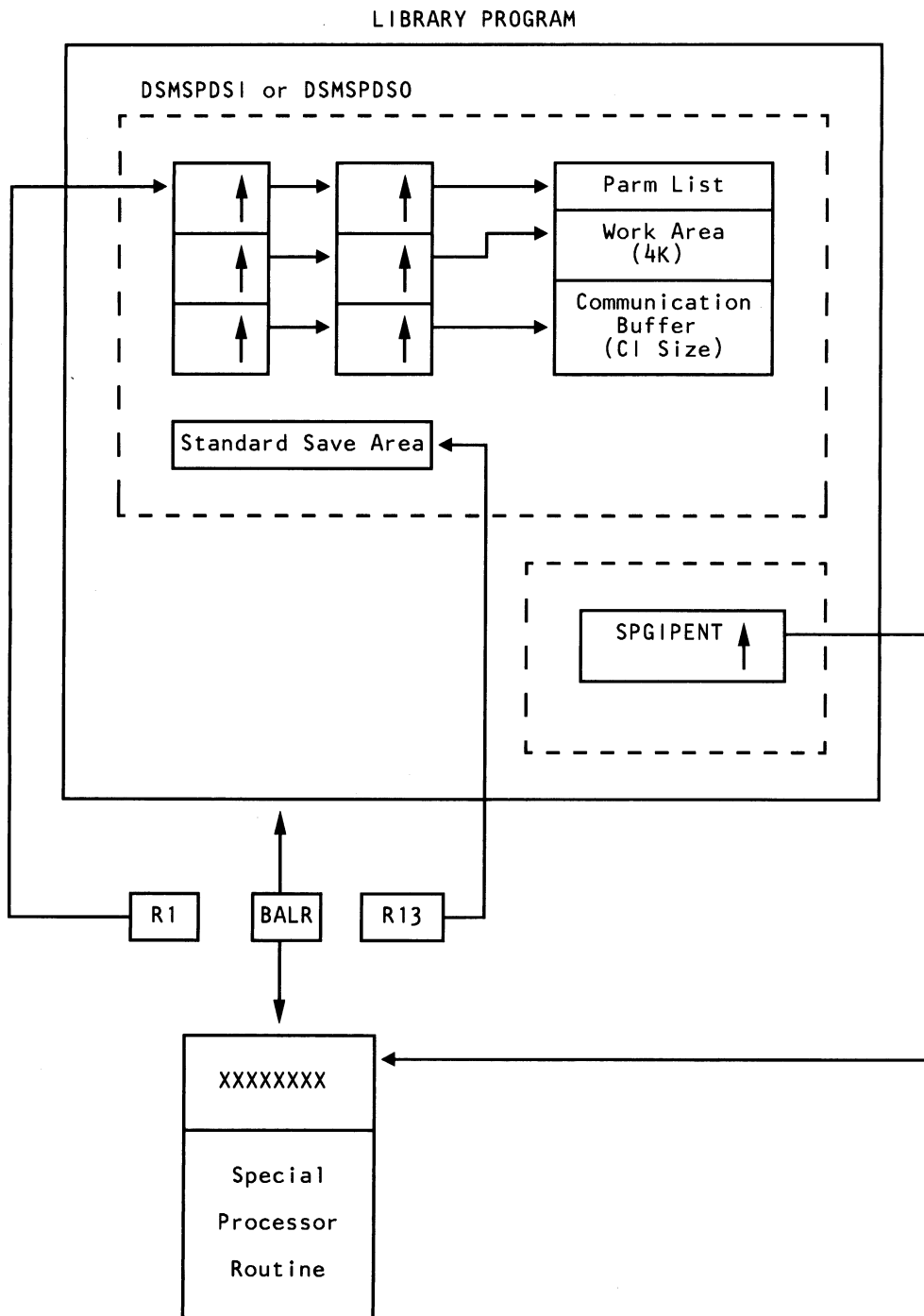


Figure 13. EXPORT and IMPORT Command Special Processor Interface

**Note:** A source is opened when the DSMSPSIC (source input controller) cannot find the requested source in the OFN table. Source is requested by a the source input controller parameter list when DSMSPSIC is called either by DCF or by the READ/CLOSE command processor modules of the library program. (See "Section

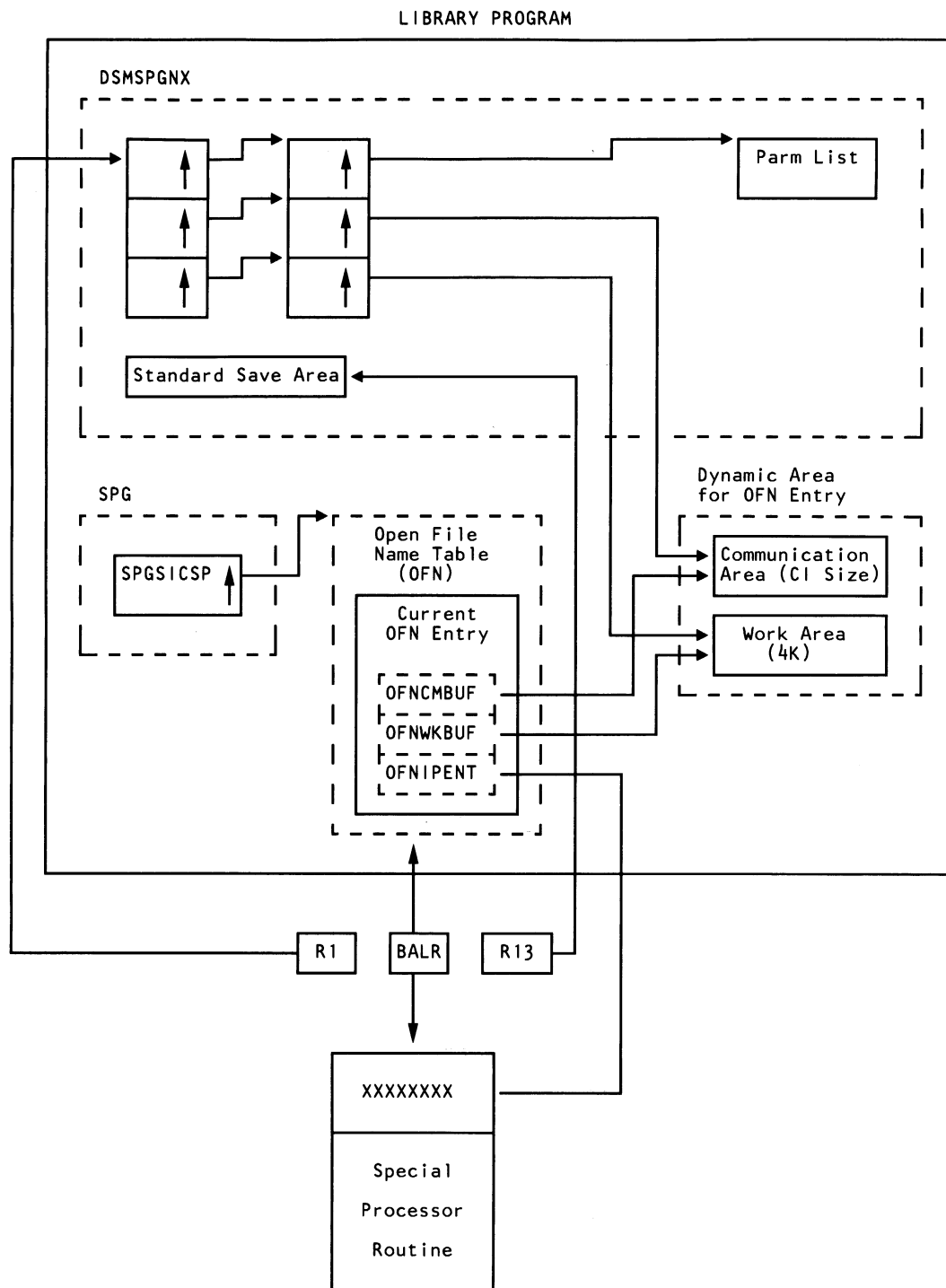


Figure 14. READ and SCRIPT Command Special Processor Interface

IV - Data Areas" on page 104 for more information about the source input controller parameter list.)

The first of the second level addresses points to the special processor parameter list (also called IPP). "Section IV - Data Areas" on page 104 describes the structure of this parameter list.

The Document Library Facility Guide explains how to use special processors. It details the input passed to the processor, return protocol, register conventions, and other useful information about writing a special processor.

## DLF to DCF Interface

Figure 15 on page 37 illustrates the interface used when DLF calls DCF to process a SCRIPT command.

The first time a SCRIPT command is processed for a DLF session, the DSMSPEXC module:

- Loads the DCF (SCRIPT/VS) load module (DSML0S30 — OS/VS2 or DSMDLS30 — VSE)
- Places the load module address in SCRENTPT (a field within DSMSPEXC)
- Acquires a dynamic storage area (1024 bytes long) in which to pass SCRIPT commands to DCF
- Stores the address of this buffer in SCRBUFFER (a field within DSMSPEXC).

Then, for the first and each subsequent SCRIPT command to be processed in the session, DSMSPEXC moves the command to the buffer pointed to by SCRBUFFER and branches (BALR) to either DSML0S30 (OS/VS2) or DSMDLS30 (VSE) with register 1 pointing to a list of five addresses:

- Address 1 — Pointer to the SCRIPT command in the buffer.
- Address 2 — Address of the entry point of the DLF module DSMSPSIC (source input controller) which is called by DCF for input data during SCRIPT command processing.
- Address 3 — Setting depends on host environment:
  - OS/VS2 — Set to the entry point of the DLF module DSMSPQNM (qualify data set name for dynamic allocation) which DCF might call to qualify a data set name of an external data set to be used by DCF during SCRIPT command processing.
  - VSE — Set to the entry point of the DLF module DSMSPDOS (I/O module) which DCF can call for I/O services to an external data set.

Address 4 — Setting depends on host environment:

- OS/VS2 — Set to the entry point of the DLF module DSMSPDYN (dynamic allocation of a data set) which DCF might call to dynamically allocate an external data set which is to be used by DCF during SCRIPT command processing.
- VSE — Set to the address of SPGCOMM in the SPG. SPGCOMM contains the command code for the command currently being processed (in this case the command code for the SCRIPT command).
- Address 5 — Address of the SPGPRDCB field in the SPG. This is to provide DCF with the address of the DCB which controls the print data set DSMLIST (OS/VS2) or SYSLST (VSE).

The Document Library Facility Guide contains an appendix describing the use of the SCRIPT/VS Formatter with the Library

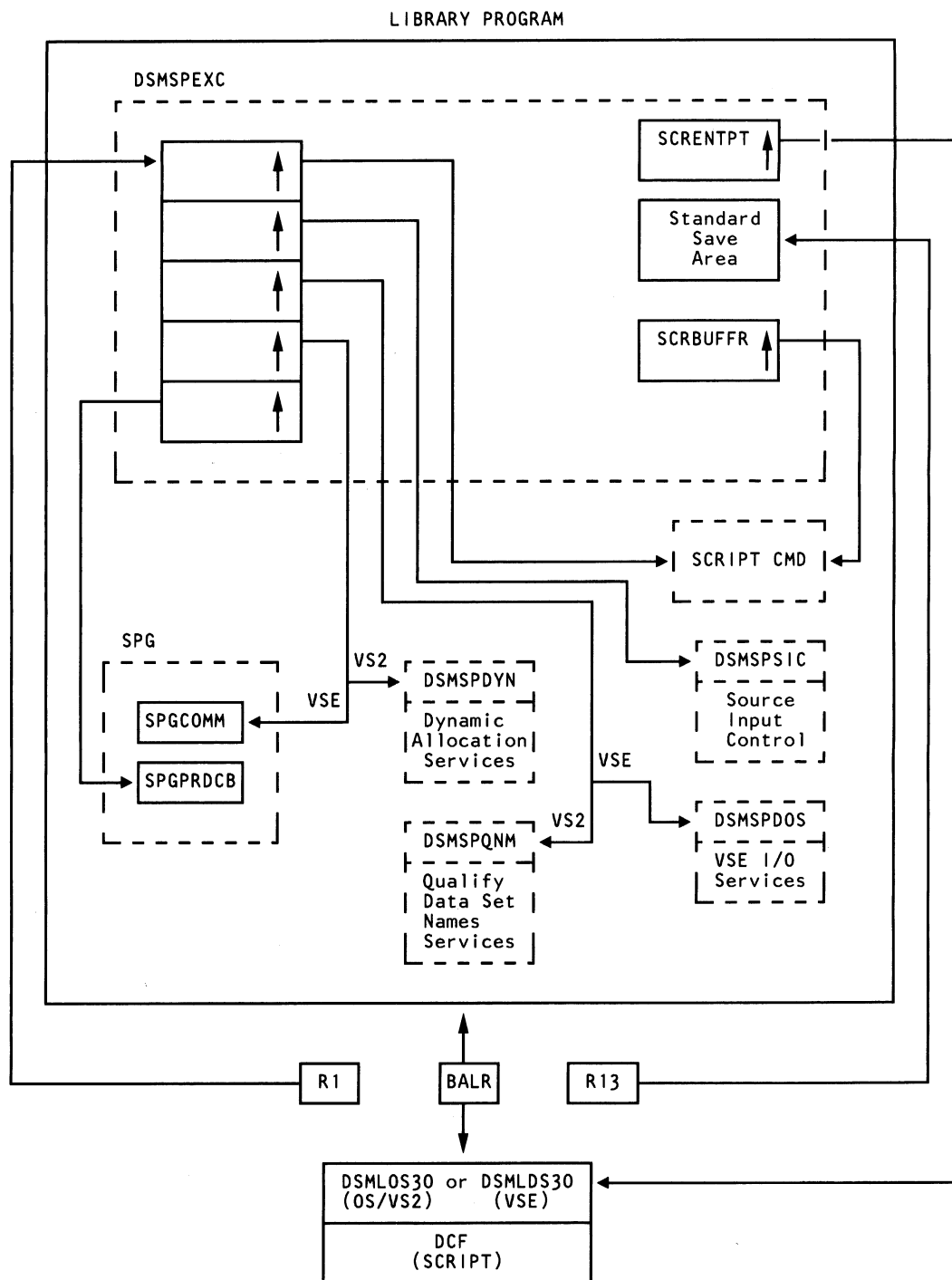


Figure 15. DLF to DCF Interface

Program. The appendix details various interactions between DLF and DCF during SCRIPT command processing.

## DCF to DLF Interfaces

The following section describes the various interfaces DCF uses to request DLF services during SCRIPT command processing.

### Source Input Interface

Figure 16 on page 39 illustrates the interface used when DCF calls DSMSPSIC (Source Input Control module) for input data. DCF branches (BALR) to the address of DSMSPSIC (entry point was passed to DCF when called by DLF) with register 1 pointing to an address field containing the address of a Source Input Control (SIC) parameter list. This parameter list is described in detail in "Section IV - Data Areas" on page 104. DCF will set fields in this parameter list to specify to DSMSPSIC the function (read, note, point, or close) to be performed and the source (DLF document, external data set, or I/O module) that is to provide the input data.

DSMSPSIC and other modules of the source input subsystem (DSMSPSOP and DSMSPGNX) interface to provide the requested function. If the read function is specified, the accessed data is pointed to by a field in the SIC parameter list on return to DCF.

### Dynamic Allocation Interface

Figure 17 illustrates the interface used when, in the OS/VS2 environment only, DCF calls DSMSPDYN (Dynamic Allocation module) to dynamically allocate or unallocate an external data set. DCF branches (BALR) to the address of DSMSPDYN (entry point was passed to DCF when called by DLF) with register 1 pointing to a list of four addresses:

- Address 1 — A flag byte field.
  - If bit 4 is on, return type of data set organization.
  - If bit 5 is on, the member name is present.
  - If bit 6 is on, DSMSPDYN must ALLOCATE an external data set.
  - If bit 7 is on, DSMSPDYN must UNALLOCATE an external data set.
- Address 2 — 44-byte field containing a fully qualified data set name.
- Address 3 — 8-byte field containing PDS member name (if flag byte — bit 5 is on).
- Address 4 — 8-byte field for ddname. The ddname will be returned by DSMSPDYN to DCF in this field if the request is for ALLOCATION. If the request is UNALLOCATE, DSMSPDYN expects this field to contain the ddname of the data set previously dynamically allocated.
- Address 5 — 2-byte field for type of data set organization. If requested, type of data set organization is returned in bit 4 of the flag byte field.

**Note:** Dynamic allocation is available only in the OS/VS2 environment.

### Qualify Data Set Name Interface

Figure 18 on page 41 illustrates the interface used when DCF calls DSMSPQNM (Qualify data set name module) to qualify a data set name. The data set name is qualified by concatenation of a prefix and a suffix according to the rules specified in the Document Library Facility Guide.

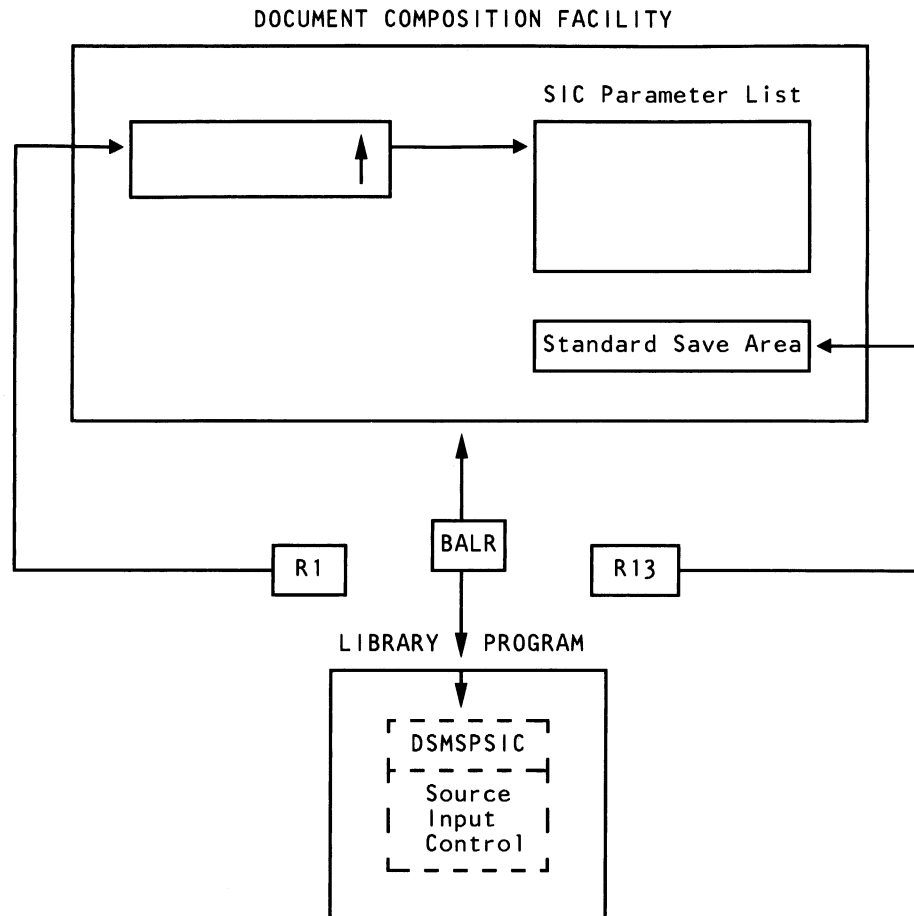


Figure 16. DCF to DLF Interface For Source Input Data

DCF branches (BALR) to the address of DSMSPQNM (entry point was passed to DCF when called by DLF) with register 1 pointing to a list of five addresses which point to fields set as follows:

- Address 1 — A flag byte field (bit 0 ON - means qualify the name only, do not call DSMSPDYN to dynamically allocate; bit 1 ON - means return 2-byte type of data set organization pointed to by the sixth address).
- Address 2 — 44-byte field containing the unqualified data set name which is to be qualified. The qualified data set name will be returned in this field.
- Address 3 — 8-byte field which can contain a PDS member name.
- Address 4 — 8-byte field which contains the suffix to be used to qualify this data set name (the prefix is constructed from the current DLF users library number).
- Address 5 — 8-byte field for ddname. The ddname will be returned to DCF in this field if dynamic allocation is done (flag bit is not on). DSMSPQNM will call DSMSPDYN to dynamically allocate the qualified data set name and DSMSPDYN will return the ddname to DSMSPQNM after dynamic allocation.

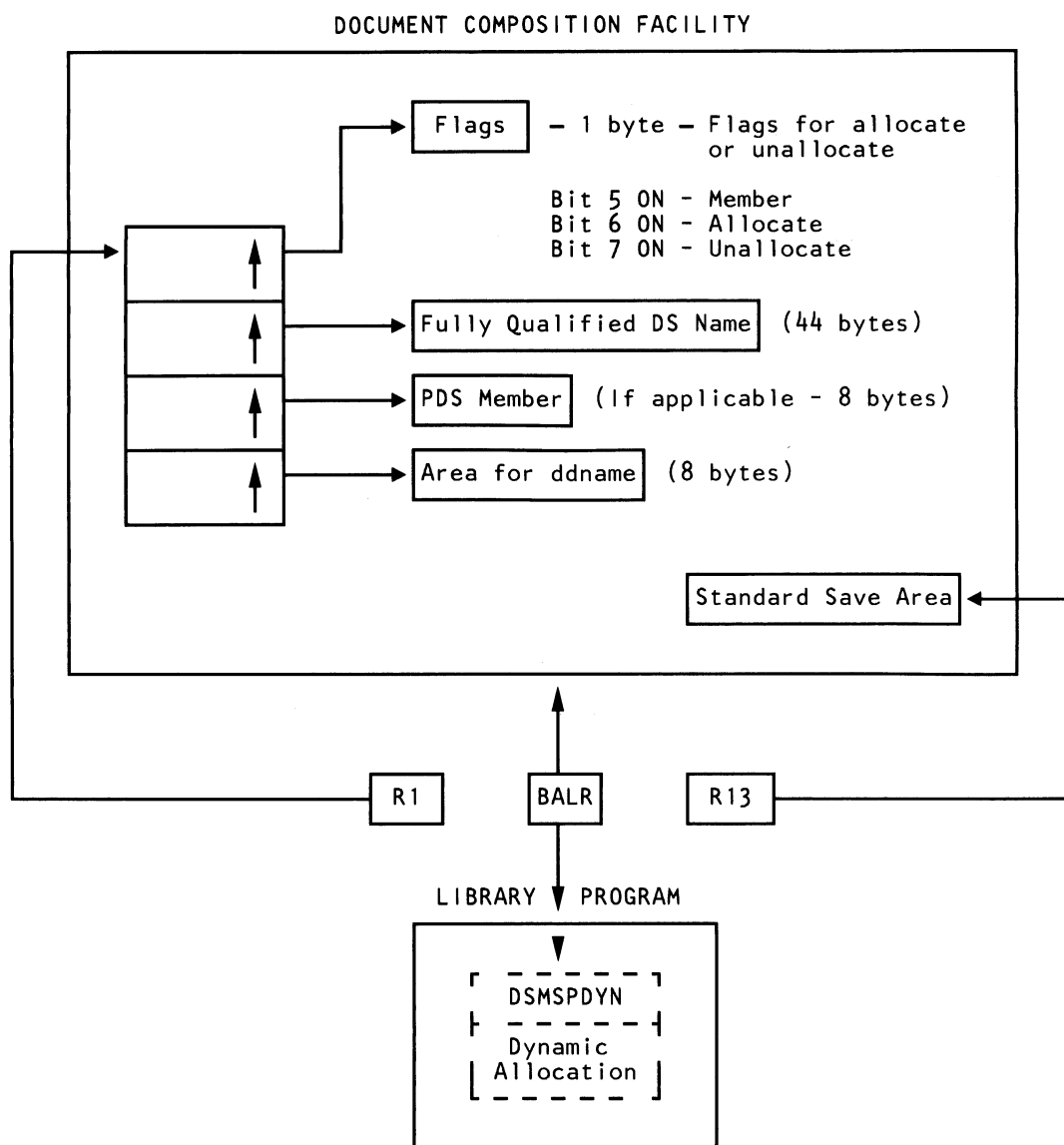


Figure 17. DCF to DLF Interface For Dynamic Allocation

**Note:** Dynamic allocation is available only in the OS/VS2 environment.

- Address 6 — 2-byte field to indicate the type of data set organization. This data will be returned if Bit 1 of the flag byte is ON. DSMSPDYN will return this data to DSMSPQNM when dynamic allocation is complete.

#### VSE I/O Services Interface

Figure 19 illustrates the interface used when DCF calls DSMSPDOS (VSE I/O module in DLF) for I/O services to an external (non-library) data set. DCF branches (BALR) to the address of DSMSPDOS (entry point was passed to DCF when called by DLF) with register 1 pointing to a list of three addresses:



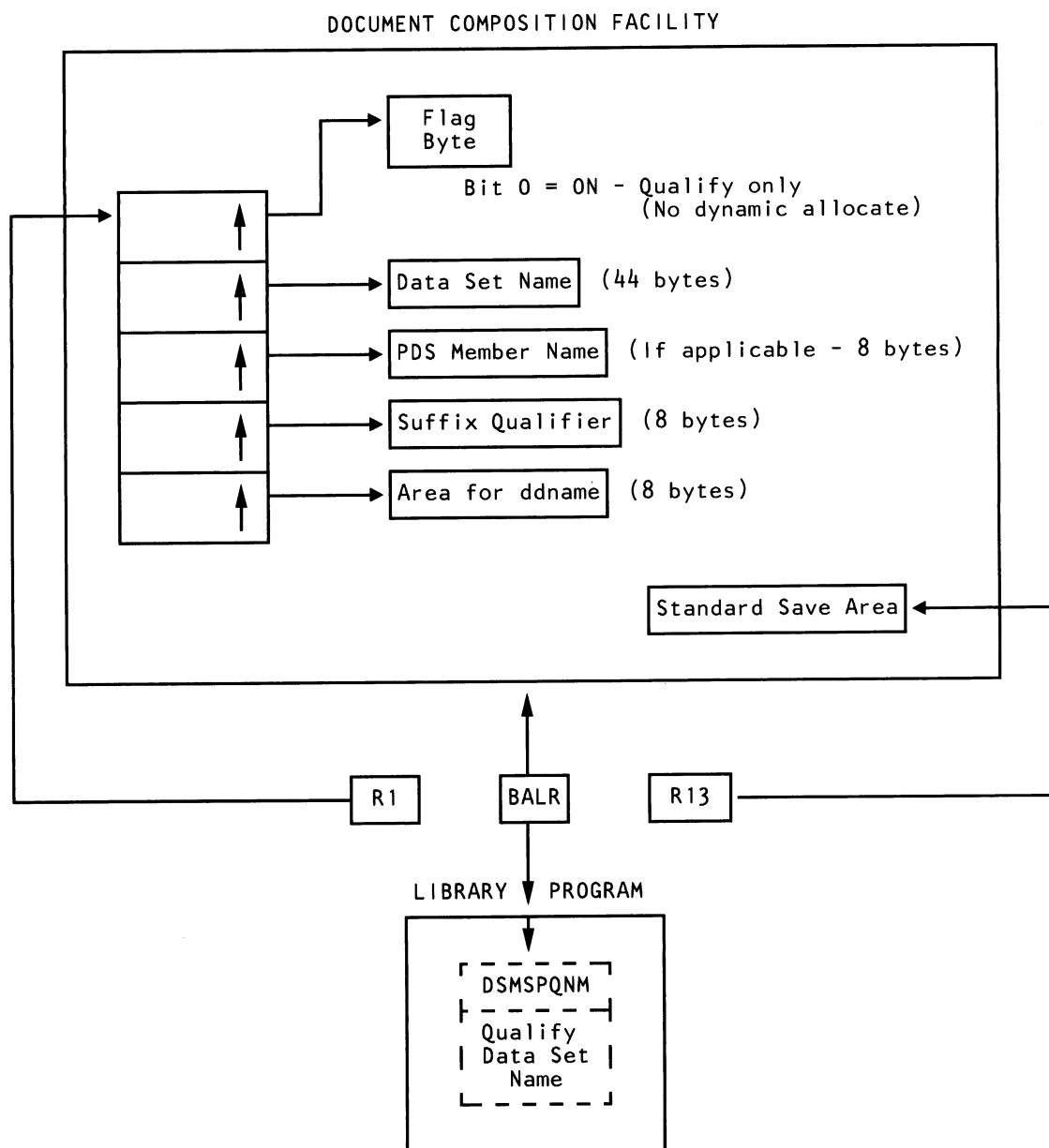
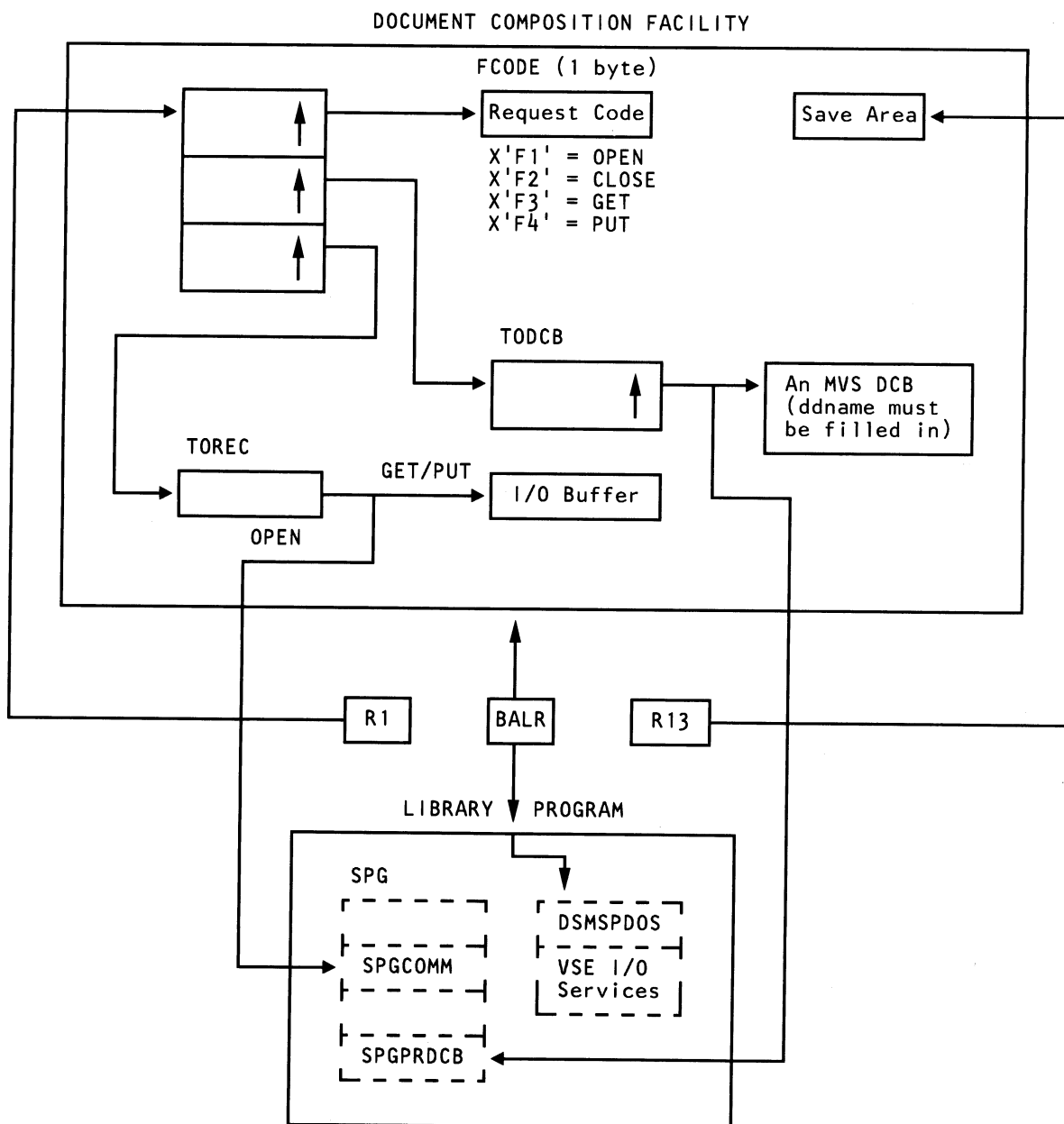


Figure 18. DCF to DLF Interface For Qualify Name Services

- Address 1 — points to FCODE (1 byte request code field) which is set with one of the following:
  - X'F1' = OPEN
  - X'F2' = CLOSE
  - X'F3' = GET
  - X'F4' = PUT
- Address 2 — points to TODCB (fullword pointer field) which contains the address of an OS DCB. OS DCBs are used in both the OS/VS2 and VSE environments as a common interface for



**Figure 19. DCF to DLF Interface for VSE I/O Services**

- **Address 3** — points to TOREC (fullword pointer field) which contains an address. The address set in TOREC depends on the function to be performed. For GET and PUT requests, TOREC points to an I/O buffer in the calling program's storage area. For OPEN requests, TOREC points to SPGCOMM in the DLF SPG (this address was passed to DCF when called by

DLF and contains the command code of the command currently being processed - in this case a SCRIPT command). TOREC is not required, and is therefore ignored for close requests.

**Note:** This interface is used in the VSE environment only. Document Library Facility Guide contains an appendix describing the use of the SCRIPT/VS Formatter with the Library Program. The appendix details the various interactions between DLF and DCF during SCRIPT command processing.

## SCRIPT Input Module Interface

Figure 20 on page 44 illustrates the interface used when DLF calls the I/O module (specified by the calling program when DLF is called as a subroutine) that is to supply the input data to be formatted (SCRIPT).

**Note:** The description below refers to the OFN and other areas that are discussed in more detail under the "Special Processor Interface" on page 33 for READ and SCRIPT commands. A review of that discussion and Figure 14 on page 35 will make the following discussion easier to follow.

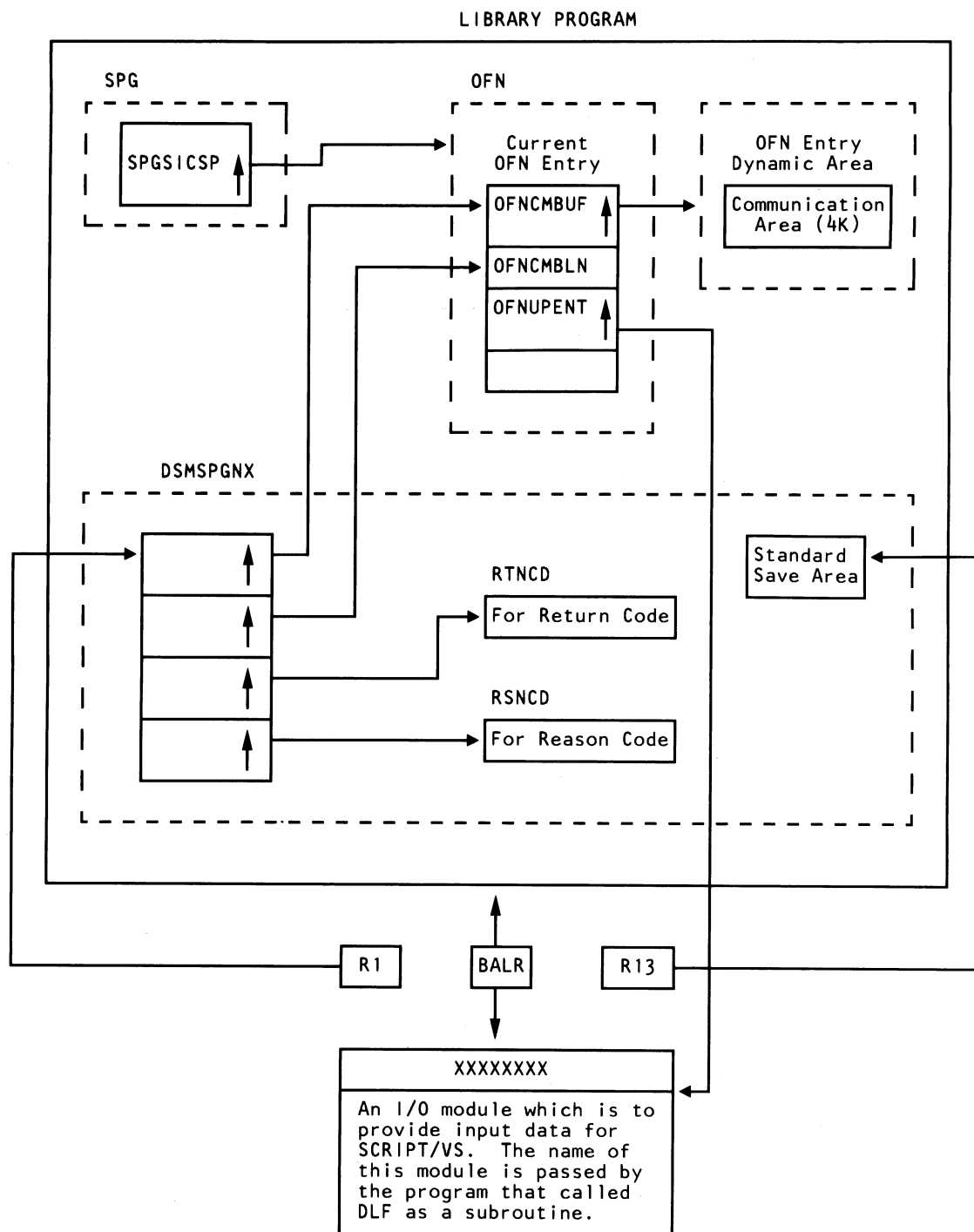
DLF provides the facilities for a calling program to specify that the data to be formatted during SCRIPT command processing is to come from an I/O module that the calling program specifies. The calling program specifies the entry name of this I/O module in a field pointed to by the fifth address of the list of addresses pointed to by register 1 when DLF is called.

The I/O module is loaded by DSMSPSOP when the source is opened (the first time DSMSPSIC is called by DCF to provide input data for a SCRIPT command where FROM(\*) was specified as the input source). The address of the module is placed in the OFNUPENT field of the OFN entry for this source.

When DCF calls DSMSPSIC for input data during SCRIPT command processing and DSMSPSIC finds that the requested source (in this case the I/O module) has been opened, then DSMSPSIC calls DSMSPGNX to get the input data. DSMSPGNX examines the OFN entry for the source, finds that an I/O module is to be called for the input and then calls the I/O module (branch to entry point address saved in OFNUPENT) with register 1 set to a list of four addresses:

- Address 1 — pointer to OFNCMBUF in the OFN entry. OFNCMBUF contains the address of the communication area that is within the dynamic storage area for this OFN entry.
- Address 2 — pointer to OFNCMBLN in the OFN entry. OFNCMBLN contains the length of the communication buffer.
- Address 3 — pointer to a fullword field where the I/O module is to return a return code.
- Address 4 — pointer to a fullword field where the I/O module is to return a reason code.

The Document Library Facility Guide describes this interface under "SCRIPT FROM(\*) - INPUT DATA FROM ANOTHER PROGRAM" in the chapter on "USING THE LIBRARY PROGRAM AS A SUBROUTINE". This description includes an example and also specifies the return codes DLF expects.



**Figure 20. SCRIPT Input Module - Interface**

## EXTERNAL DATA SETS USED BY DLF

DLF accesses external files via SAM in the VSE environment and via BSAM/QSAM in the OS/VS2 environment.

DCBs are used in both the OS/VS2 and VSE environments to control external data sets. In the OS/VS2 environment, these DCBs are also used to access the external data sets. In the VSE environment, the information in the DCB is used by the DSMSPDOS module (I/O module for external data sets in the VSE environment) to build and/or modify a proper VSE DTF before an actual open or I/O operation.

Most of the DCBs for external (non-library) data sets are coded in the SPG as shown below:

- SPGPRDCB — Controls DMSLIST (OS/VS2) or SYSLST (VSE)
- SPGRDDCB — Controls non-library (external) sequential input
- SPGNMDCB — Controls NAMELIST data set (ARCHIVE & RETRIEVE commands)
- SPGWBD CB — Controls non-library BSAM output (OS/VS2)
- SPGWQDCB — Controls non-library QSAM (OS/VS2) or SAM (VSE) output.

DCBs that are not coded in the SPG are:

- INDCB — coded in DSMSPEXC and controls SYSIN (OS/VS2) or SYSRDR (VSE).
- DCBMODEL — this model DCB is coded in DSMSPSOP and is used only as a model to build DCBs for external data sets which are to be used for SCRIPT input. The model is copied to the dynamic storage area for each OFN entry which controls an external data set. DSMSPSOP then modifies the copy of the DCB model using fields from the SIC parameter list. DCF sets up the SIC parameter list before calling the DLF module DSMSPSIC to get the first/next record of an external data set.

**Note:** See "External Interfaces" on page 24 for additional information regarding DLF and DCF interfaces.

## **SECTION II - PROGRAM FUNCTIONAL DESCRIPTION**

This section describes the principal processing paths through each function. These functions are:

- Initialization, Control, and Termination
- Command Validation
- Command Initialization
- Command Processing
- Command Termination
- Services
- Global Data Areas.

### **DLF INITIALIZATION, CONTROL, AND TERMINATION**

The initialization, control, and termination functions of the Document Library Facility are performed by two modules, DSMSPEXC and DSMSPPIN.

#### **DLF Control and Termination - DSMSPEXC**

**INPUT:** DSMSPEXC determines how it was called by analyzing input parameters as described in "External Interfaces" on page 24.

Command segments are read from the input stream and concatenated until a complete command is available.

**DESCRIPTION:** DSMSPEXC is the processing controller and terminator of DLF. This module always gets control when DLF is invoked (whether as a batch job or as a called subroutine) and maintains control throughout the duration of the DLF job. Other DLF modules are called by DSMSPEXC for DLF functions and services.

When DLF is invoked as a batch job, DSMSPEXC gains control to perform initialization functions and then calls DSMSPPIN to:

- Analyze JCL input parameters for the LIST/NOLIST option
- Open the system and library data sets
- Perform other DLF initialization processing.

DSMSPEXC then goes into a command processing loop which:

- Reads a command
- Calls DSMSPCMD to analyze the command
- Calls DSMSPPIN to open any additional data sets required for the command
- Calls the required command processor
- Calls DSMSPTRM to close any data sets that are unique to the command.

This command processing loop continues until the end of the input data stream, or until a terminating error (return code 16) occurs, at which point DSMSPEXC closes the system and library data sets and relinquishes control.

DSMSPEXC also gets control when DLF is called as a subroutine. In this case, DSMSPEXC will return to the caller after each call request (command) has been processed until it receives a CLOSE ALL request. The system and library data sets will then be closed and control relinquished to the calling program.

See "Control Flow Within the Library Program" on page 19 and "External Interfaces" on page 24 in Section I for additional information on the control and termination functions of DSMSPEXC.

**OUTPUT:** DSMSPPMSG is called to issue messages if errors are detected during processing. "EXC" will appear as the 4th through 6th characters of the message ID of messages initiated by DSMSPEXC.

#### **DLF Initialization - DSMSPPIN**

**INPUT:** When DSMSPPIN gets control, SPGEXEQD is set on by DSMSPEXC if DLF was invoked as a batch job, or is set off if DLF was called as a subroutine.

**DESCRIPTION:** DSMSPEXC calls DSMSPPIN at the beginning of a DLF session to:

- Open the system data sets (SYSIN and DSMLIST for OS/VS2, SYSRDR and SYSLST for VSE)
- Load the hashing (DSMSPHSH) and user exit (DSMSPUXI) modules
- Obtain I/O buffers
- Open the input and/or output library data sets specified in the JCL
- Determine whether the LIST option is in effect for this DLF session by checking for the:
  - JCL LIST parameter (if running under OS/VS2 as a batch job)
  - LISTX option in the Communication Region (if running under VSE as a batch job)
  - Parameter list passed by the calling program (if DLF is called as a subroutine).

DSMSPPIN calls DSMSPGCL to open the basic 4K clusters for both input and output. DSMSPGCL also does validity checking to make sure that the requested basic 4K clusters do, in fact, belong to the directory data set with which the JCL associates them.

**Note:** Because the calling program provides the input commands, the SYSIN/SYSRDR data set will not be opened if DLF was called as a subroutine.

System data sets (SYSIN and DSMLIST) are opened within DSMSPPIN in the OS/VS2 environment. In the VSE environment, system data sets (SYSRDR and SYSLST) are opened by a call to DSMSPDOS.

See "Control Flow Within the Library Program" on page 19 and "External Interfaces" on page 24 for additional information on the control and termination functions of DSMSPPIN.

#### **OUTPUT**

##### **Normal Completion**

- System data sets (DSMLIST AND SYSIN for OS/VS2, SYSLST and SYSRDR for VSE) are open.
- Input and/or output Directory and basic 4K clusters which were specified in the JCL are open.

- Messages are issued to DSMLIST (OS/VS2) or SYSLST (VSE) for library data sets successfully opened.
- SPGLIST is set to indicate LIST or NOLIST processing and a message is issued indicating which option is in effect for this DLF run.
- Register 15 is set to zero.

#### **Error Exits**

- When any real error is detected during DSMSPPIN processing, Register 15 is set to a non-zero return code; causing the remainder of DSMSPPIN processing to be bypassed. When a non-zero return code is passed back to the calling program (DSMSPEXC), DLF processing terminates.
- Error messages are issued to DSMLIST (OS/VS2) or SYSLST (VSE) indicating the reason for the termination.
- Some system and/or library data sets might be open, and must therefore be closed by the DSMSPEXC termination routines.

DSMSPPMSG is called to issue messages if errors are detected during processing. "PIN" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPPIN.



## COMMAND VALIDATION

The command syntax checking and command validation functions are performed by the four modules described below.

### Command Decode and Validation - DSMSPCMD

**INPUT:** When DSMSPCMD gets control, a complete command will have been read and placed in the SPG16K buffer of the SPG.

**DESCRIPTION:** DSMSPCMD is the command scan controller for DLF and is called by DSMSPEXC when any command has been read and is ready to be validated. This module also gets control during ARCHIVE NAMELIST or RETRIEVE NAMELIST processing (from DSMSPARC or DSMSPRET respectively) to scan, tokenize, and validate a namelist entry.

For each command (except SCRIPT) DSMSPCMD calls DSMSPSCN first for initialization and then to tokenize and analyze the command. DSMSPCMD identifies but does not validate SCRIPT commands. Instead, SCRIPT commands are passed by DSMSPEXC to DCF for parsing, validation, and processing.

DSMSPSCN uses the syntax graphs (DSMSPSYX and DSMSPSY) and designated subroutines within DSMSPCMD to identify and validate command operands before placing them in SPG fields.

**OUTPUT:** Command variables are placed in fields within the SPG and flags are set to indicate which operands appear on the command.

DSMSPMSG is called to issue error messages if any are detected during command processing. Because of the special save area procedures that relate modules DSMSPCMD and DSMSPSCN, "SCN" will appear as the 4th through 6th characters of the message ID of messages initiated by DSMSPCMD.

### Command Scan and Parse - DSMSPSCN

**INPUT:** Register 1 points to a parameter list containing:

- The address of the scan work area (SPG16K + 4K)
- The address of the current DLF command (SPG16K buffer)
- The length of the current DLF command
- The address of the beginning of the syntax graph (label CMDVERB in DSMSPSYX).

When DSMSPCMD calls DSMSPSCN for the first time, DSMSPCMD passes the scan work area address, and indicates initialization by setting the next three parameters to zero. In a second call to DSMSPSCN (for parsing), DSMSPCMD passes all parameters.

**DESCRIPTION:** DSMSPSCN performs the tokenizing and analysis of all DLF commands except the SCRIPT command. It is called twice by DSMSPCMD; first for initialization and again for command scanning.

The command scanning process has two parts: tokenizing and analysis.

In the first part, DSMSPSCN scans the command one character at a time to build a string of tokens. Each token is a valid segment of a DLF command such as a subscripted word ( 'xxx(' ), a colon ( ':' ), or a quoted string ( 'ssss' ). Separator characters are ignored. Each token consists of the token type, token length, and token text (if not obvious from the token type). The token string ends either when the complete command is tokenized and a command-end token is added, or when the first

error (e.g., an invalid character) occurs and an error token is added to the end of the token string.

In the second part of scanning, analysis, DSMSPSCN uses the syntax graph contained in modules DSMSPSYX and DSMSPSY to determine if the tokens form a valid DLF command. As it identifies valid tokens, DSMSPSCN sets variables and flags in SPGPARG. During analysis, DSMSPSCN matches the type, length, and content of the current token in the token string against each of a group of valid tokens in the syntax graph. When the two match, a designated subroutine (in DSMSPCMD) is called to perform additional testing, identify the variable, and place the token contents in SPGPARG. If no errors are found, processing proceeds with the next token at the location in the syntax graph identified by the matching valid token. Correct completion of the token string matching process or an error token ends command analysis.

**OUTPUT:** DSMSPCMD's first call to DSMSPSCN initializes the scan work area. The second call produces a token string in the scan work area and the internal form of the current command in SPGPARG. Variable parameter operands and the variable subparameters of keywords are saved just as they were input. Bits in SPGBOL indicate the presence or absence of operands and specified conditions. Indicators in SPGCOMM and SPGSUBC, respectively, identify the first and second words of the command.

DSMSPSCN does not issue error messages directly. Errors that occur within DSMSPSCN are issued by DSMSPCMD with "SCN" as the 4th through 6th characters of the message ID. There are two groups of errors that DSMSPSCN recognizes:

- Tokenizing errors (invalid characters, misplaced left parentheses, invalid operands, and incomplete quoted strings or comments)
- Analysis errors (duplicate and/or invalid operands, unknown commands, and missing required parameters).

During the analysis phase, when DSMSPSCN locates the first error (regardless of type), it returns to DSMSPCMD. DSMSPCMD calls DSMSPMSG to issue the necessary error messages.

#### **Command Syntax Graphs Part 1 — DSMSPSYX**

**INPUT:** Not applicable — no executed code.

**DESCRIPTION:** DSMSPSYX contains the first segment of the syntax graph for DLF commands. It is used in conjunction with DSMSPSY (the second half of the syntax graph), DSMSPSCN (the parser), and DSMSPCMD (command validation) to parse each DLF command.

DSMSPSYX contains only the first syntax graph segment. There is no executed code.

**OUTPUT:** Not applicable — no executed code.

#### **Command Syntax Graphs Part 2 — DSMSPSY**

**INPUT:** Not applicable — no executed code.

**DESCRIPTION:** DSMSPSY contains the second segment of the syntax graph for DLF commands. It is used in conjunction with DSMSPSYX (the first half of the syntax graph), DSMSPSCN (the parser), and DSMSPCMD (command validation) to parse each DLF command.

DSMSPSY contains only the second syntax graph segment. There is no executed code.

**OUTPUT:** Not applicable — no executed code.

## COMMAND INITIALIZATION - DSMSPINT

**INPUT:** When DSMSPINT gets control, a complete command has been read by DSMSPEXC and validated by DSMSPCMD. DSMSPCMD has placed operands of the command into fields in the SPG.

The following directory records from the input library might be required during command initialization:

- SPR — to get OPERATOR routing information during ARCHIVE or RETRIEVE command initialization
- The DER of the document being retrieved during RETRIEVE DOCUMENT initialization — to get DERNOTE information (that might indicate the external data set or the volume on which the document is archived).

**DESCRIPTION:** DSMSPINT performs command initialization processing. This module gets control from DSMSPEXC (the Processing Controller of DLF) after a command has been analyzed and is ready to be processed. No other modules call DSMSPINT.

This module opens any data sets required to process a DLF command, including any non-library data sets (external data sets) that are required during the transfer of data to or from the library as specified by some DLF commands.

In the VSE environment, DSMSPINT calls DSMSPDOS to open the external data set.

In the OS/VS2 environment, the external data set is opened within DSMSPINT. Before the open, and if necessary, DSMSPQNM and DSMSPDYN are called to qualify and dynamically allocate the data set, respectively.

DSMSPINT also calls DSMSPHSH to hash any passwords (user or document) that are keyed in the command, insures that the input and/or output document library is open for those commands which require them, and (for the READ and SCRIPT commands) acquires dynamic storage for the open file name table (OFN).

During initialization for some commands, directory records must be accessed from the input library. DSMSPINT calls DSMSPILU to read these records.

**OUTPUT:** External data sets (if required by the command) are opened, any passwords in the command are hashed, and, if this is the first READ or SCRIPT command of the session, a dynamic storage area is acquired for the OFN table. DSMSPMSG is called to issue messages if errors are detected during command processing. "INT" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPINT.

## COMMAND PROCESSING

The source modules described in this section process the DLF commands and thereby provide the primary functions of DLF. However, each command function not only requires one of the command processors, but also requires command validation, command initialization, and command termination. Each additional function is discussed in a separate section.

The DSMSPDEF, DSMSPPUR, DSMSPDIS, and DSMSPCPY modules decide which of the sub-command processors must be called to process a specific DEFINE, PURGE, LIST, or COPY command. Each of the modules described here provides the primary function for at least one DLF command.

### ACCOUNT Command - DSMSPACT

**INPUT:** Operands of the ACCOUNT command are placed into fields in the SPG. DSMSPACT analyzes these fields to determine how the ACCOUNT command is to be processed.

DSMSPACT reads each of the UPRs associated with the users in the range specified in the ACCOUNT command. The UPRs are read (by DSMSPDIS) either from the input document library (if FROM was not specified in the ACCOUNT command) or from the external data set specified in the FROM operand of the ACCOUNT command. Such an external data set must have been created by the COPY OUT command prior to issuing the ACCOUNT command.

**DESCRIPTION:** DSMSPACT calls DSMSPACT when an ACCOUNT command has been read by DSMSPACT, validated by DSMSPCMD, initialized by DSMSPINT and is ready for processing.

DSMSPACT controls the accumulation, formatting, and output of the account information which it obtains from the User Profile Records (UPRs) of those users within the range specified in the ACCOUNT command.

If DSMSPACT finds UPRNLOG=ON in a selected UPR, the space utilization fields for the user associated with that UPR might not have been properly updated the last time a command capable of changing these fields was issued. DSMSPACT recalculates the amount of space in use by this user through the DERREAD procedure. The DERREAD procedure:

- Reads each DER (via DSMSPDIS) in the library associated with the UPR and accumulates the space used.
- Accumulates the space used by DERs associated with documents this user owns in public libraries (pointed to by LERs) and in his project library.

The accumulated space usage totals replace those in the UPR buffer before DSMSPSMF is called to format the type-47 accounting information record for this user.

**OUTPUT:** The account information is written to the data set specified in the TO operand of the ACCOUNT command. If TO is not specified, the data set described by the JCL statement labeled SMF (OS/VSE) or DSMACNT (VSE) is used. The actual creation of the type-47 SMF records and the actual writing to the output data set is done by DSMSPSMF which is called by DSMSPACT to handle each selected UPR.

DSMSPMSG is called to issue messages if errors are detected during command processing. "ACT" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPACT.

## ARCHIVE Commands (3 commands) - DSMSPARC

**INPUT:** Operands of the ARCHIVE command are placed into fields in the SPG. DSMSPARC analyzes these fields to determine how this ARCHIVE command is to be processed. There are three ARCHIVE commands, ARCHIVE DOCUMENT, ARCHIVE NAMELIST, and ARCHIVE ALL.

Depending on the specific ARCHIVE command, one or more documents in the output DLF library are read (via the DSMSPOLU service module) during processing.

The ARCHIVE NAMELIST command requires DSMSPARC to archive documents that are named in a list. This list can be in an external data set or in the input stream. Namelist records are validated by DSMSPCMD and are read from the namelist by DSMSPRDC.

**DESCRIPTION:** DSMSPARC is called by DSMSPEXC when an ARCHIVE command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT and is ready for processing.

Just before each document's updated DER is written to the directory data set, DSMSPARC calls DSMSPEXI (user exit controller) to invoke the user exit. The user exit might elect not to archive the document or documents.

One or more documents are archived from the output library to an external data set specified in the command. A single document is archived with the ARCHIVE DOCUMENT command. A group of documents named in a list are archived with the ARCHIVE NAMELIST command. All documents which meet certain selection criteria (i.e., user number range, data type, date range) are archived with the ARCHIVE ALL command.

DSMSPARC insures that the current user is authorized to archive the specified/implied documents from the specified or implied library. DSMSPARC uses DSMSPOLU to read the records of the document(s) from the output library. It uses DSMSPTPO to convert the records to variable format and write them to the external data set.

If an ARCHIVE NAMELIST command is being processed, DSMSPARC uses DSMSPRDC to read records (selection information) from the namelist. DSMSPARC uses DSMSPCMD to tokenize each namelist entry.

After the control intervals and DER of a document have been successfully written to the specified or implied external data set, the control intervals which make up the document in the output library are returned to the free chain for re-use. The DER of the document is modified to show that the document is archived, the date it was archived, and its location information. The modified DER is written back to the output library to retain information about the newly archived document. DSMSPARC uses DSMSPOLU for all output library I/O.

**OUTPUT:** DSMSPARC uses DSMSPTPO to convert the records to be archived to variable format and write them to the external data set specified or implied in the command. DSMSPARC uses DSMSPOLU to write the updated DERs to the output library.

DSMSPMSG is called to issue messages if errors are detected during command processing. "ARC" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPARC.

## **AUTH Command - DSMSPAUT**

**INPUT:** Operands of the AUTH command are placed into fields in the SPG. DSMSPAUT analyzes these fields to determine how this AUTH command is to be processed. The UPR of the user specified in the AUTH command is accessed from the input and/or output library.

**DESCRIPTION:** DSMSPAUT is called by DSMSPEXC when an AUTH command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT and is ready for processing.

DSMSPAUT uses DSMSPILU and DSMSPOLU to access the UPR of the specified user in the input or output library, respectively. DSMSPAUT checks the validity of the user specified in the AUTH command and determines that user's authority to access the input and/or output DLF library. It also determines and saves authorization data, user characteristics, and user defaults in the current user fields of the SPG. These fields will be used during the processing of following commands (until another AUTH command is received).

**Note:** In an OS/VS2 environment, user authorization can include RACF checking (via RACHECK macro) if a RACF class name was specified in the DEFINE SYSTEM command for the input and/or output DLF library.

**OUTPUT:** The current user fields of the SPG are set with authorization data, user characteristics, and user defaults of the user specified in the AUTH command. This data is obtained from the UPR (input and/or output library) of that user.

DSMSPMSG is called to issue messages if errors are detected during command processing. "AUT" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPAUT.

## **CLOSE Command - DSMSPCLO**

**INPUT:** Operands of the CLOSE command are placed into fields in the SPG. DSMSPCLO analyzes these fields to determine how the CLOSE command is to be processed.

**DESCRIPTION:** DSMSPCLO is called by DSMSPEXC when a CLOSE command has been received from a program calling DLF as a subroutine, validated by DSMSPCMD, initialized by DSMSPINT and is ready for processing.

DSMSPCLO determines if a CLOSE ALL command or a CLOSE command for a specific file has been received, then sets up the SIC parameter list accordingly before calling DSMSPSIC to close a all files or a specific file.

The file(s) to be closed are any that were opened by a previous READ command. The READ and CLOSE command can only be issued by a program which calls DLF as a subroutine. The file(s) to be closed might be actual external data sets or DLF documents. (See "Services" on page 81 for additional information.)

**OUTPUT:** The specified files will have been closed by DSMSPSIC.

DSMSPMSG is called to issue messages if errors are detected during command processing. "CLO" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPCLO.

## **COPY Commands (Controller) - DSMSPCPY**

**INPUT:** Operands of a COPY command are placed into fields in the SPG. DSMSPCPY then analyzes these fields to determine which COPY command has been read so it can do preliminary processing and call the correct command processor.

**DESCRIPTION:** DSMSPCPY is called by DSMSPEXC when a COPY command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT, and is ready for processing.

DSMSPCPY is the selection and control module for the DLF COPY commands. For all COPY commands except COPY OUT, DSMSPCPY ensures the presence of the DLF output library. It also tests for the presence and accuracy of some of the parameters on the individual COPY commands and performs initial verification that the current user is authorized to perform the copy.

DSMSPCPY calls the appropriate command processor depending on the COPY command being processed:

- COPY DOCUMENT (DSMSPCPD)
- COPY IN (DSMSPCPI)
- COPY LIBRARY (DSMSPCPL)
- COPY OUT (DSMSPCPO)
- COPY USER (DSMSPCPU).

**OUTPUT:** See output from the various COPY command functions.

DSMSPMSG is called to issue messages if errors are detected during command processing. "CPY" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPCPY.

## **COPY DOCUMENT Command - DSMSPCPD**

**INPUT:** Operands of the COPY DOCUMENT command are placed into fields in the SPG. DSMSPCPD analyzes these fields to determine how this COPY DOCUMENT command is to be processed.

Directory records of both the input and output library or libraries (they both can be the same physical library) are input to the COPY DOCUMENT function. Also, the control intervals of the document to be copied are read from the input library.

**DESCRIPTION:** DSMSPCPD is called by DSMSPCPY when a COPY DOCUMENT command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT, passed preliminary screening by DSMSPCPY, and is ready for processing.

DSMSPCPD copies documents from one DLF library to another, between user libraries of the same DLF library, or within the same user library.

DSMSPCPD verifies that the current user is authorized to do the specified copy and then controls the copy. DSMSPILU is used to read directory records and document control intervals from the input library, DSMSPOLU is used to read directory records from the output library, and DSMSPLB0 is used to write the document (DER and control intervals) to the specified/implied user library.

**OUTPUT:** The document and its DER are written to the specified/implied user library in the DLF output library (which can be physically the same as in the DLF input library). The UPR of the user to which the document is copied is updated.

DSMSPMSG is called to issue messages if errors are detected during command processing. "CPD" will appear as the 4th through

6th characters of the message ID of messages initiated from DSMSPCPD.

#### **COPY IN Command - DSMSPCPI**

**INPUT:** Operands of the COPY IN command are placed into fields in the SPG. DSMSPCPI analyzes these fields to determine how the COPY IN command is to be processed.

An external sequential data set that was created by a previous COPY OUT command must be used as input to this function.

**DESCRIPTION:** DSMSPCPI is called by DSMSPCPY when a COPY IN command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT, passed preliminary screening by DSMSPCPY, and is ready for processing.

DSMSPCPI processes a COPY IN command by sequentially reading the COPY OUT data set and writing into the DLF output library those directory records and documents that are specified or implied by the command.

DSMSPOLU is called to read directory records from the output library to verify that the copy can be accomplished. DSMSPCPI reads the COPY OUT data set directly in OS/VS2 environments; but calls DSMSPDOS to read this data set in VSE environments. DSMSPB0 is called to write documents and their corresponding DERS to the output library and to update the UPR of the user library into which they are copied. DSMSPOLU is called to write other directory records (selected from the COPY OUT data set) to the output library.

**OUTPUT:** Selected documents and directory records (as specified or implied by the COPY IN command) will have been copied from the COPY OUT data set to the output library.

DSMSPMSG is called to issue messages if errors are detected during command processing. "CPI" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPCPI.

#### **COPY LIBRARY Command - DSMSPCPL**

**INPUT:** Operands of the COPY LIBRARY command are placed into fields in the SPG. DSMSPCPL analyzes these fields to determine how the COPY LIBRARY command is to be processed.

Directory records of both the input and output library are input to the COPY LIBRARY function. Also, the control intervals of the document to be copied are read from the input library.

**DESCRIPTION:** DSMSPCPL is called by DSMSPCPY when a COPY LIBRARY command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT, passed preliminary screening by DSMSPCPY, and is ready for processing.

DSMSPCPL processes a COPY LIBRARY command by sequentially reading the DLF input library and writing into the DLF output library all documents and directory records (other than the SPR and CPRs). The control intervals of a document are copied after their identifying DERS are copied. Existing users and documents in the output library are replaced only if so specified on the command.

DSMSPOLU is called to read directory records from the output library to verify that the copy can be accomplished. DSMSPILU is called to read directory records and document control intervals from the input library. DSMSPB0 is called to write each document and its associated DER to the output library. DSMSPOLU is called to write UPRs, CERs, PPRs, and UMRs to the output library.



**OUTPUT:** All documents and directory records (other than the SPR and CPRs) found in the input library are copied to the output library unless they already existed in the output library and the REPLACE option was not specified by the COPY LIBRARY command.

DSMSPMSG is called to issue messages if errors are detected during command processing. "CPL" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPCPL.

#### **COPY OUT Command - DSMSPCPO**

**INPUT:** Operands of the COPY OUT command are placed into fields in the SPG. DSMSPCPO analyzes these fields to determine how the COPY OUT command is to be processed.

Directory records and document control intervals are read from the input library.

**DESCRIPTION:** DSMSPCPO is called by DSMSPCPY when a COPY OUT command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT, passed preliminary screening by DSMSPCPY, and is ready for processing.

DSMSPCPO processes a COPY OUT command by copying selected documents and directory records (as specified/IMPLIED by the COPY OUT command) from the input library to a specified or IMPLIED external data set. DSMSPCPO reads the input library sequentially to determine the order in which records are to be copied. SPRs and CPRs are not copied and CERs, PPRs, and UMRs are copied only if an administrator issues the COPY OUT command.

DSMSPILU is used to read directory records and document control intervals from the input library. DSMSPTP0 is used to write directory records and document control intervals to the external data set specified or IMPLIED by the COPY OUT command.

**OUTPUT:** Documents and directory records (not including the SPR and CPRs) that meet selection criteria are copied to the external data set specified or IMPLIED in the COPY OUT command.

DSMSPMSG is called to issue messages if errors are detected during command processing. "CPO" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPCPO.

#### **COPY USER Command - DSMSPCPU**

**INPUT:** Operands of the COPY USER command are placed into fields in the SPG. DSMSPCPU analyzes these fields to determine how the COPY USER command is to be processed.

Directory records of both the input and output library (they can both be the same physical library) are input to the COPY USER function. Also, the control intervals of the document to be copied are read from the input library.

**DESCRIPTION:** DSMSPCPU is called by DSMSPCPY when a COPY USER command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT, passed preliminary screening by DSMSPCPY, and is ready for processing. DSMSPCPU copies the documents of a user from the input library to another or the same user in the output library (the output library can physically be the same as the input library). If the specified user does not exist in the output library, a UPR is created for that user.

DSMSPCPU verifies that the current user is authorized to do the specified copy, then controls the copy. DSMSPILU is used to read directory records and document control intervals from the input library. DSMSPOLU is used to read directory records from

the output library, and DSMSPB0 is used to write documents (DERs and control intervals) to the specified user library in the output library.

**OUTPUT:** The documents and DERs are written to the specified user library in the output library (which can be physically the same as in the DLF input library). The UPR of the user to which the documents are copied is updated. If the UPR did not exist in the output library, one is created.

DSMSPMSG is called to issue messages if errors are detected during command processing. "CPU" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPCPU.

#### **DEFINE/ALTER Commands (Controller) - DSMSPDEF**

**INPUT:** Operands of a DEFINE/ALTER command are placed into fields in the SPG. DSMSPDEF analyzes these fields to determine which DEFINE/ALTER command has been read so it can do preliminary processing and call the correct command processor.

**DESCRIPTION:** DSMSPDEF is called by DSMSPexc when a DEFINE/ALTER command has been read by DSMSPexc, validated by DSMSPCMD, initialized by DSMSPINT, and is ready for processing.

DSMSPDEF is the selection and control module for the DLF DEFINE/ALTER commands. It tests for the presence and accuracy of some of the parameters on the individual DEFINE/ALTER commands and performs initial verification that makes sure the current user is authorized to perform the define or alter function.

DSMSPDEF calls the appropriate command processor depending on the DEFINE/ALTER command being processed:

- DEFINE/ALTER CLUSTER - DSMSPDAC
- ALTER DOCUMENT - DSMSPDAD
- DEFINE/ALTER PROCESS - DSMSPDAF
- DEFINE/ALTER CLASS - DSMSPDAL
- DEFINE/ALTER MAP - DSMSPDAM
- DEFINE/ALTER SYSTEM - DSMSPDAS
- DEFINE/ALTER USER - DSMSPDAU.

**OUTPUT:** See output from the various DEFINE/ALTER command functions.

DSMSPMSG is called to issue messages if errors are detected during command processing. "DEF" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPDEF.

## DEFINE/ALTER CLUSTER Command - DSMSPDAC

**INPUT:** Operands of the DEFINE CLUSTER or ALTER CLUSTER command are placed into fields in the SPG. DSMSPDAC analyzes these fields to determine how the DEFINE or ALTER CLUSTER command is to be processed.

The existing CPR of the cluster is read from the output library for an ALTER CLUSTER command.

**DESCRIPTION:** DSMSPDAC is called by DSMSPDEF when a DEFINE CLUSTER or ALTER CLUSTER command has been read by DSMSPDXC, validated by DSMSPCMD, initialized by DSMSPINT, passed preliminary screening by DSMSPDEF, and is ready for processing. DSMSPDAC can also be called from DSMSPDAS to format the basic 4K cluster during DEFINE SYSTEM or ALTER SYSTEM command processing.

If the CPR exists (valid only for the ALTER CLUSTER command) it is modified as indicated and written to the output library. If the CPR does not exist (valid only for DEFINE CLUSTER command) a CPR is constructed as indicated by the command and written to the output library. The new data set (DEFINE) or additional space (ALTER) is formatted and a free space header (FSH) constructed. The CPR is reread and updated with the correct control interval size. The cluster definition data will be written if the LIST option is active for this DLF session.

DSMSPDAC calls DSMSPOLU to read and write the specified CPR (output library). DSMSPDIS is called to write cluster definition data when the LIST option is specified (via job step JCL) for the current DLF session. (See descriptions of DLF modules DSMSPDXC and DSMSPPIN under "DLF Initialization, Control, and Termination" on page 46 for additional information about the LIST option.)

**OUTPUT:** Either a new CPR (DEFINE CLUSTER) or a modified CPR (ALTER CLUSTER) will have been written to the output library and the complete cluster or its additional space will be formatted.

DSMSPMSG is called to issue messages if errors are detected during command processing. "DAC" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPDAC.

## ALTER DOCUMENT Command - DSMSPDAD

**INPUT:** Operands of the ALTER DOCUMENT command are placed into fields in the SPG. DSMSPDAD analyzes these fields to determine how the ALTER DOCUMENT command is to be processed.

The output library is accessed for the DER of the document to be modified.

**DESCRIPTION:** DSMSPDEF calls DSMSPDAD when an ALTER DOCUMENT command has been read by DSMSPDXC, validated by DSMSPCMD, initialized by DSMSPINT, passed preliminary screening by DSMSPDEF, and is ready for processing.

DSMSPDAD processes the ALTER DOCUMENT command by determining whether or not:

- The user is authorized to alter this document
- The altered form of the DER already exists
- The original DER exists.

Version characteristics, single or multiple and the requested version, are tested next, followed by password, and parameter compatibility tests.

DSMSPOLU is used to read and write the DER of the document being altered.

**OUTPUT:** The specified document is altered as indicated by the ALTER DOCUMENT command. The altered DER information is listed if the LIST option is active (via job step JCL) for this DLF session. (See descriptions of DLF modules DSMSPEXC and DSMSPPIN under "DLF Initialization, Control, and Termination" on page 46 for additional information about the LIST option.)

DSMSPMSG is called to issue messages if errors are detected during command processing. "DAD" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPDAD.

#### **DEFINE/ALTER PROCESS Command - DSMSPDAF**

**INPUT:** Operands of the DEFINE PROCESS or ALTER PROCESS command are placed into fields in the SPG. DSMSPDAF analyzes these fields to determine how the DEFINE or ALTER PROCESS command is to be processed.

The existing PPR for the processor is read from the output library for an ALTER PROCESS command.

**DESCRIPTION:** DSMSPDAF is called by DSMSPDEF when a DEFINE PROCESS or ALTER PROCESS command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT, passed preliminary screening by DSMSPDEF, and is ready for processing.

If the PPR exists (valid only for ALTER PROCESS command) it is modified as indicated and written to the output library. If the PPR does not exist (valid only for DEFINE PROCESS command) a PPR is constructed as indicated by the command and written to the output library. The processor definition data will be listed if the LIST option is active (via job step JCL) for this DLF session. (See descriptions of DLF modules DSMSPEXC and DSMSPPIN under "DLF Initialization, Control, and Termination" on page 46 for additional information about the LIST option.)

DSMSPDAF calls DSMSPOLU to read and write the specified PPR (output library). DSMSPLIS is called to write processor definition data when the LIST option is specified (via job step JCL) for the current DLF session. (See descriptions of DLF modules DSMSPEXC and DSMSPPIN under "DLF Initialization, Control, and Termination" on page 46 for additional information about the LIST option.)

**OUTPUT:** Either a new PPR (DEFINE PROCESS) or a modified PPR (ALTER PROCESS) will have been written to the output library.

DSMSPMSG is called to issue messages if errors are detected during command processing. "DAF" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPDAF.

#### **DEFINE/ALTER CLASS Command - DSMSPDAL**

**INPUT:** Operands of the DEFINE CLASS or ALTER CLASS command are placed into fields in the SPG. DSMSPDAL analyzes these fields to determine how the DEFINE or ALTER CLASS command is to be processed.

The existing CER for the class is read from the output library for an ALTER CLASS command.

**DESCRIPTION:** DSMSPDAL is called by DSMSPDEF when a DEFINE CLASS or ALTER CLASS command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT, passed preliminary screening by DSMSPDEF, and is ready for processing.

If the CER exists (valid only for ALTER CLASS command) it is modified as indicated and written to the output library. If the CER does not exist (valid only for DEFINE CLASS command) a CER is constructed as indicated by the command and written to the

output library. The class definition data will be listed if the LIST option is active (via job step JCL) for this DLF session. (See descriptions of DLF modules DSMSPEXC and DSMSPPIN under "DLF Initialization, Control, and Termination" on page 46 for additional information about the LIST option.)

DSMSPDAL calls DSMSPOLU to read and write the specified CER (output library). DSMSPLIS is called to write class definition data when the LIST option is specified (via job step JCL) for the current DLF session. (See descriptions of DLF modules DSMSPEXC and DSMSPPIN under "DLF Initialization, Control, and Termination" on page 46 for additional information about the LIST option.)

**OUTPUT:** Either a new CER (DEFINE CLASS) or a modified CER (ALTER CLASS) will have been written to the output library.

DSMSPMMSG is called to issue messages if errors are detected during command processing. "DAL" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPDAL.

#### **DEFINE/ALTER MAP Command - DSMSPDAM**

**INPUT:** Operands of the DEFINE MAP or ALTER MAP command are placed into fields in the SPG. DSMSPDAM analyzes these fields to determine how the DEFINE or ALTER MAP command is to be processed.

The existing UMR for the map is read from the output library for an ALTER MAP command.

**DESCRIPTION:** DSMSPDAM is called by DSMSPDEF when a DEFINE MAP or ALTER MAP command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT, passed preliminary screening by DSMSPDEF, and is ready for processing.

If the UMR exists (valid only for ALTER MAP command), it is modified as indicated by the command and written to the output library. If the UMR does not exist (valid only for DEFINE MAP command), a UMR is constructed in the manner indicated by the command and written to the output library. The map definition data is listed if the LIST option is active (via job step JCL) for this DLF session. (See descriptions of DLF modules DSMSPEXC and DSMSPPIN under "DLF Initialization, Control, and Termination" on page 46 for additional information about the LIST option.)

DSMSPDAM calls DSMSPOLU to read and write the specified UMR (output library). DSMSPLIS is called to write map definition data when the LIST option is specified (via job step JCL) for the current DLF session. (See descriptions of DLF modules DSMSPEXC and DSMSPPIN under "DLF Initialization, Control, and Termination" on page 46 for additional information about the LIST option.)

A user exit is available for both commands that allows scanning of both the new UMR and the UPR associated with it. The user exit can only change the UPRAPPL field, causing the updated UPR to be written.

**OUTPUT:** Either a new UMR (DEFINE MAP) or a modified UMR (ALTER MAP) is written to the output library. The user exit might cause the UPR to be modified and written to the output library.

DSMSPMMSG is called to issue messages if errors are detected during command processing. "DAM" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPDAM.

## **DEFINE/ALTER SYSTEM Command - DSMSPDAS**

**INPUT:** Operands of the DEFINE SYSTEM or ALTER SYSTEM command are placed into fields in the SPG. DSMSPDAS analyzes these fields to determine how the DEFINE or ALTER SYSTEM command is to be processed.

The existing SPR for the system is read from the output library for an ALTER SYSTEM command.

**DESCRIPTION:** DSMSPDAS is called by DSMSPDEF when a DEFINE SYSTEM or ALTER SYSTEM command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT, passed preliminary screening by DSMSPDEF, and is ready for processing.

If the SPR exists (valid only for ALTER SYSTEM command), it is modified as indicated by the command and written to the output library. If the SPR does not exist (valid only for DEFINE SYSTEM command at the initial session of DLF against this document library), the SPR is constructed as indicated by the command and written to the output library.

DSMSPDAS calls DSMSPOLU to read and write the specified SPR (output library). DSMSPDAC is called to define or reformat the default 4K cluster and create or modify the CPR for this cluster as required.

**OUTPUT:** Either a new SPR (DEFINE SYSTEM) or a modified SPR (ALTER SYSTEM) is written to the output library. If the system was defined (DEFINE SYSTEM) or additional space added (ALTER SYSTEM), the default 4K cluster is formatted and its CPR created or modified by call to DSMSPDAC.

DSMSPMSG is called to issue messages if errors are detected during command processing. "DAS" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPDAS.

## **DEFINE/ALTER USER Command - DSMSPDAU**

**INPUT:** Operands of the DEFINE USER or ALTER USER command are placed into fields in the SPG. DSMSPDAU analyzes these fields to determine how the DEFINE or ALTER USER command is to be processed.

The existing UPR for the user is read from the output library for an ALTER USER command.

**DESCRIPTION:** DSMSPDAU is called by DSMSPDEF when a DEFINE USER or ALTER USER command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT, passed preliminary screening by DSMSPDEF, and is ready for processing.

If the UPR exists (valid only for ALTER USER command), it is modified as indicated by the command and written to the output library. If the UPR does not exist (valid only for DEFINE USER command), a UPR is constructed as indicated by the command and written to the output library. The user definition data will be listed if the LIST option is active for this DLF session.

DSMSPDAU calls DSMSPOLU to read and write the specified UPR (output library). DSMSPDIS is called to write user definition data when the LIST option is specified (via job step JCL) for the current DLF session. (See descriptions of DLF modules DSMSPEXC and DSMSPPIN under "DLF Initialization, Control, and Termination" on page 46 for additional information about the LIST option.)

A user exit is available for both commands that allows scanning of the new (or for ALTER USER the original) UPR. The user exit might only change the UPRAPPL field, causing the updated UPR to be written.

**OUTPUT:** Either a new UPR (DEFINE USER) or a modified UPR (ALTER USER) is written to the output library. The user exit might also cause the UPR to be modified and written to the output library.

DSMSPMSG is called to issue messages if errors are detected during command processing. "DAU" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPDAU.

#### **ENVIRONMENT Command (VSE only) - DSMSPDOS**

**INPUT:** Operands of an ENVIRONMENT command are placed into fields in the SPG. The fields are then analyzed by DSMSPDOS to determine processing.

There are also three input parameters used by DSMSPDOS when it is called either to process an ENVIRONMENT command or function as a VSE I/O service module: FCODE, TODCB, and TOREC.

Each input parameter is detailed below.

- **FCODE** — an integer indicating the reason for call:
  - 0 = process the ENVIRONMENT command
  - 1 = open a file
  - 2 = close a file
  - 3 = get a record
  - 4 = put a record
  - 5 = open CONSOLE file
  - 6 = write a message to console
  - 7 = write a message to console and read reply
- **TODCB** — to OS/VS DCB - for FCODE = 1, 2, 3, or 4 only
- **TOREC** — pointer to buffer if FCODE = 3, 4, 6, or 7. If FCODE = 1, TOREC is a pointer to the current command number (SPGCOMM). If FCODE = 0, 2, or 5, the TOREC parameter is not used.

**DESCRIPTION:** DSMSPDOS is called to perform three functions:

- Simulate OS/VS QSAM support in a VSE environment
- Handle write to operator (WTO) and write to operator with reply (WTOR) functions in a VSE environment
- Process the parameters of the ENVIRONMENT command to set up the necessary information for the QSAM support function.

Processing is a matter of checking FCODE for 0 - 7 and calling the associated subprocedure:

- FCODE = 0            ENVCMD
- FCODE = 1            DOSOPEN
- FCODE = 2            DOSCLOSE
- FCODE = 3            DOSGET
- FCODE = 4            DOSPUT
- FCODE = 5 - 7        CONSOLE.

ENVCMDB is called to process the parameters of the ENVIRONMENT command. It places the results into an array called ENVPARMS where each entry is identified by its unique DLBI name. There are enough entries for 32 different DLBI names. If this number is exceeded a return is made to the calling program with a return code of 16 in register 15.

DOSOPEN is called to open the file indicated by the calling program. This is accomplished by determining the type of file being opened and calling one of eight subprocedures:

- OPENIPT — to open SYSIPT
- OPENLST — to open SYSLST
- OPENIMP — to open the IMPORT file
- OPENEXP — to open the EXPORT file
- OPENLIS — to open the 'List To' file
- OPENCPPI — to open the copy-input file or retrieve file
- OPENCPPO — to open the copy-output file or archive file
- OPENWRK — to open one of the DCF work files.

DOSCLOSE is called to close a file upon completion of processing. Cleanup is accomplished and the designated file closed. If necessary, the buffer area is freed. The FILINFO block is set to not in use. Return is with register 15 equal to 0, unless FREEVIS was unsuccessful.

DOSGET is used to get one logical record and place it in the caller's buffer. All input record formats are processed. Fixed length records are passed as fixed; variable, spanned, and undefined length records are mapped to variable format.

Deblocking is accomplished by using the variables RECSTART and BLKEND in the FILINFO block. If on entry RECSTART = BLKEND:

- The current block is empty
- A get macro is issued
- Data is moved
- RECSTART is advanced.

If the input file is spanned, additional gets might be needed to collect the entire record.

On entry, DCBADR points to the DCB and TOREC points to the buffer unless DSMSPDSI is the calling module. If DSMSPDSI is the caller, buffer W3 in the GETVIS area is used. IOREG, IOSIZE, and the MVCL registers (TOADR, TOLN, FRADR, and FRLEN) are locally restricted. If RECSTART = BLKEND then a new block is read from the input file. This is accomplished by the INCLUDE code GENGET. GENGET executes a VSE get macro as follows:

1. If the file is assigned ignore, a branch is made to the endfile exit. Otherwise, the get is issued.
2. IOREG and RECSIZE registers are set with the location and length of the physical record (if the file is fixed length then RECSIZE has already been set because VSE does not set RECSIZE in this case).
3. If the blocksize is over 4k, as in the case of copy files, there is only one I/O area and the get does not set IOREG. Instead, IOREG is also preset with the buffer address. RECSTART and BLKEND in the FILINFO block are set based on the RECSIZE and IOREG settings.



When the get macro completes, a logical record is moved to the caller's buffer. If spanned records are being read, and another physical record is needed, GENGET is again used to get the physical record. If the records are not fixed length records, when the complete record is moved to the buffer an RDW is built at the beginning of the buffer. DOSGET returns to the main procedure with return code of zero.

If a get is issued and no data exists, the end-of-file exit ENDFILE is activated by the VSE logic module. The GENGET include code branches here rather than issuing a get against a file assigned ignore EXITADR is set to the caller's EODAD address and GENGET returns to DSMSPDOS with the return code of zero. The EOF indicator is also set for DOSCLOSE.

DOSPUT is used to collect one logical record and send it to a file. If the output file is assigned 'ignore', DOSPUT returns immediately to DSMSPDOS with a return code of zero. If spanned format is used, full VSE LIOCS support avoids end-of-volume problems. For all other formats (F, FB, V, VB, and U), blocking is handled as required and the physical block is output as undefined format.

FILINFO fields (BLKSTART, RECSTART, and BLKEND) control the blocking process. If a blocked record to be put will exceed remaining space in the physical record, a put is issued to output the present buffer and obtain a new one. A new record is then moved to the buffer and RECSTART is updated to the first byte beyond the record just moved.

Upon entry, the DCBADR points to DCB and TOREC points to the record in the caller's buffer (the RDW, if not fixed format). Registers (IOREG, IOSIZE, TOADR, TOLEN, FRADR, and FRLEN) are locally RESTRICTED for use in this procedure. The DTF is checked to see if file is ignored and if so, DOSPUT returns to DSMSPDOS with register 15=0. DSMSPDOS tests for spanned format and if the format is spanned, DSMSPDOS issues a put with the buffer as the work area. Return is then made to the main routine. Otherwise, one record is moved and blocked. If the buffer is full, a put is issued and return is made to the main routine.

CONSOLE is called to:

- Issue a message to the console (WTO) using a PUT to console DTF
- Issue a message to the console and read the reply (WTOR) using a PUTR to console DTF.

Console DTF does not have to be opened.

OUTPUT: The exit will be to either the point of invocation or, in the case of end-of-file in FCODE = 3 operation, to the caller's EODAD exit. Return Code = 0 if successful. If GET the record has been returned as requested. Otherwise, the requested service is accomplished and the DCB passed is updated to reflect the result.

#### EXPORT Command - DSMSPEXP

INPUT: Operands of the EXPORT command are placed into fields in the SPG. DSMSPEXP analyzes these fields to determine how the EXPORT command is to be processed.

The document to be exported is read from the DLF input library.

DESCRIPTION: DSMSPEXP is called by DSMSPEXC when an EXPORT command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT and is ready for processing.

DSMSPEXP controls the processing of the EXPORT command. A document in the input DLF library is read, converted to output

format, and written to the data set specified or implied by the EXPORT command. DSMSPEXP uses DSMSPLBI to read the document from the input library and DSMSPDSO to deblock, format, and write control intervals to the external data set.

Prior to the export, DSMSPEXP insures that the current user is authorized to export the specified document from the specified or implied library. Also, if a special processor is implied by the EXPORT command (OUTDATA specified), DSMSPLU is used to read the PPR from the input library to get the entry point name of the processor. If a special processor was specified or implied by the EXPORT command, the processor is loaded and its entry point placed in the SPGIPENT field of the SPG. DSMSPDSO calls the processor prior to the output of each record.

The user exit (DSMSPUXI) is called (via the user exit controller - DSMSPEXI) after authorization checks and setup but before exporting the document. The user exit can modify the DERAPPL field of the DER or it can elect to stop the EXPORT by returning a return code of eight (8) or higher. If the UPR is modified by the user exit, DSMSPOLU is called to write the modified UPR to the output document library. See Document Library Facility Guide for additional information on user exits.

**OUTPUT:** The specified document is exported to the specified or implied external data set.

DSMSPMSG is called to issue messages if errors are detected during command processing. "EXP" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPEXP.

#### **IMPORT Command - DSMSPIMP**

**INPUT:** Operands of the IMPORT command are placed into fields in the SPG. DSMSPIMP analyzes these fields to determine how the IMPORT command is to be processed.

The document to be imported is read from the external data set specified or implied by the IMPORT command.

Directory records are read from the DLF output library.

**DESCRIPTION:** DSMSPIMP is called by DSMSPEXC when an IMPORT command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT and is ready for processing.

DSMSPIMP controls the processing of the IMPORT command. Data from an external data set specified or implied by the IMPORT command is read, formatted into control intervals and stored as a document in the DLF output library. DSMSPIMP uses DSMSPDSI to read the external data set and construct control intervals. It uses DSMSPLB0 to update the UPR of the user who is to own the document being imported, insert the new DER, and write each control interval of the document to the output DLF library. It uses DSMSPOLU to read and write various output library directory records.

Prior to the import, DSMSPIMP insures that the current user is authorized to import the specified document into the specified or implied library. Also, if a special processor is implied by the IMPORT command (INDATA specified), DSMSPOLU is used to read the PPR from the output library to get the entry point name of the processor. If a special processor was specified or implied by the IMPORT command, the processor is loaded and its entry point placed in the SPGIPENT field of the SPG. DSMSPDSI calls the processor for each record read from the external data set.

The user exit (DSMSPUXI) is called (via the user exit controller - DSMSPEXI) after all control intervals are written to the ESDS but before the new DER is written to the directory of the output library. The user exit can modify the DERAPPL field of the DER or it can elect to stop the IMPORT by returning a return code of

eight (8) or higher. See the Document Library Facility Guide for additional information on user exits.

**OUTPUT:** The specified document will be imported to the specified or implied user library.

DSMSPMSG is called to issue messages if errors are detected during command processing. "IMP" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPIMP.

#### **LIST Commands (Controller) - DSMSPLIS**

**INPUT:** There is no parameter list for calling DSMSPLIS; all required information is communicated to DSMSPLIS via SPG fields.

Operands of a LIST command are placed into fields in the SPG. The subcommand field (SPGSUBC) in the SPG contains a subcommand code that governs the selection of the proper LIST command processor. If DSMSPLIS is called from DSMSPEXC, SPGCOMM contains the command code for the LIST commands (CMDLIS - 11) and fields in the SPGPARGMA contains selection criteria (if required) to be used by the selected LIST command processor. If DSMSPLIS is called from a DEFINE/ALTER command processor, SPGCOMM is zero and the information to be listed by the selected LIST processor is placed in SPGDIBUF by the calling program.

**DESCRIPTION:** DSMSPLIS is called by DSMSPEXC when a LIST command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT, and is ready for processing. DSMSPLIS can also be called by the DEFINE/ALTER command processors to list information about a directory record just defined or altered. Calls to DSMSPLIS by the DEFINE/ALTER command processors are made only if the LIST option is specified (by job step JCL) for the current DLF session (see descriptions of the DLF modules DSMSPEXC and DSMSPPIN for additional information regarding the LIST option).

DSMSPLIS is the selection and control module for the DLF LIST functions and the command processor for the LIST SYSTEM command. It ensures that the current user is authorized to access the input or output document library to be listed and sets up the list processor common work area (LST) prior to calling one of the following list command processors:

1. LIST CLASS (DSMSPLIL)
2. LIST CLUSTER (DSMSPLIC)
3. LIST DOCUMENT (DSMSPLID)
4. LIST MAP (DSMSPLIM)
5. LIST PROCESS (DSMSPLIF)
6. LIST SYSTEM (DSMSPLIS - LISTSYS subroutine)
7. LIST USER (DSMSPLIU)

The List Processor Common Work Area (LST) is described in "Section IV - Data Areas."

**OUTPUT:** The List Processor Common Work Area LST is based at a displacement of 4K into the SPG16K buffer and is initialized by DSMSPLIS prior to calling a command processor. Information from the system profile record (SPR) will be listed for a LIST SYSTEM command.

DSMSPMSG is called to issue messages if errors are detected during command processing. "LIS" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPLIS.

## LIST CLASS Command - DSMSPLIL

**INPUT:** There is no parameter list for calling DSMSPLIL; all required information is communicated to DSMSPLIL via SPG fields.

On entry to DSMSPLIL, DSMSPLIS has already initialized the List Processor Common Work Area (LST). This means that all flag bytes and return and reason code fields have been cleared. The LST is located at a displacement of 4K into the SPG16K buffer of the SPG.

If DSMSPDAL calls DSMSPLIL (via DSMSPLIS), SPGCOMM is zero and the Class Entry Record (CER) to be listed is in the SPGDIBUF field of the SPG.

If DSMSPLIL is called to process the LIST CLASS command, SPGCOMM contains the command code for LIST commands (CMDLIS - 11) and DSMSPLIL reads the CERs to be listed from the requested directory data set (via DSMSPLU for the input document library and DSMPOLU for the output document library).

**DESCRIPTION:** DSMSPLIL lists data from Class Entry Records (CERs).

DSMSPLIL is called by DSMSPLIS (the controlling module for LIST command processors) when a LIST CLASS command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT, passed preliminary screening by DSMSPLIS, and is ready for processing. DSMSPLIL can also be called (via DSMSPLIS) by the DEFINE/ALTER CLASS command processor (DSMSPDAL) to list data from a CER just defined or altered. DSMSPDAL calls DSMSPLIL (via DSMSPLIS) to list CER data only if the LIST option is specified (via job step JCL) for the current DLF session. (See descriptions of DLF modules DSMSPEXC and DSMSPPIN under "DLF Initialization, Control, and Termination" on page 46 for additional information about the LIST option.)

**OUTPUT:** There are two exits from DSMSPLIL. The first is after listing information from the single CER in SPGDIBUF (as requested by DSMSPDAL via DSMSPLIS). The second is after listing information from all CERs requested by a LIST CLASS command.

DSMSPMMSG is called to issue messages if errors are detected during command processing. "LIL" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPLIL.

## LIST CLUSTER Command - DSMSPLIC

**INPUT:** There is no parameter list for calling DSMSPLIC; all required information is communicated to DSMSPLIC via SPG fields.

On entry to DSMSPLIC, DSMSPLIS has already initialized the List Processor Common Work Area (LST). This means that all flag bytes and return and reason code fields have been cleared. The LST is located at a displacement of 4K into the SPG16K buffer of the SPG.

If DSMSPLIC is called from DSMSPDAL (via DSMSPLIS), SPGCOMM is zero and the Cluster Profile Record (CPR) to be listed is in the SPGDIBUF field of the SPG.

If DSMSPLIC is called to process the LIST CLUSTER command, SPGCOMM contains the command code for LIST commands (CMDLIS - 11) and DSMSPLIC calls DSMSPLU (input library) or DSMPOLU (output library) to read the CPRs to be listed from the input or output document library directory data sets.

When the SPACE option of the SPACE/NOSPACE operand pair is specified on the LIST CLUSTER command, DSMSPLIC will also call DSMSPGLC to activate each cluster for which there is a CPR. This call ensures that all available CLB are active and the free

space header (FSH) in each CLB is current. The FROMLIB operand determines whether the cluster is activated for input or output.

**DESCRIPTION:** DSMSPLIC lists data from Cluster Profile Records (CPRs). It also lists additional data from the Free Space Header (FSH) of the Cluster Control Block (CLB) for each cluster identified by a CPR when the SPACE option of the SPACE/NOSPACE operand pair is specified on the LIST CLUSTER command.

DSMSPLIC is called by DSMSPLIS (the controlling module for LIST command processors) when a LIST CLUSTER command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT, passed preliminary screening by DSMSPLIS, and is ready for processing. DSMSPLIC might also be called (via DSMSPLIS) by the DEFINE/ALTER CLUSTER command processor (DSMSPDAC) to list data from the CPR just defined or altered. DSMSPDAC calls DSMSPLIC (via DSMSPLIS) to list CPR data only if the LIST option is specified (via job step JCL) for the current DLF session. (See descriptions of the DLF modules DSMSPEXC and DSMSPPIN for additional information regarding the LIST option.)

**OUTPUT:** There are two exits from DSMSPLIC. The first is after listing information from the single CPR in SPGDIBUF (as requested by DSMSPDAC via DSMSPLIS). The second is after listing information about all clusters (as requested by a LIST CLUSTER command).

The SPACE option of the SPACE/NOSPACE operand pair causes DLF to use the data in the CPR(s) plus the data from the FSH in the CLB for each cluster identified by a CPR to produce an alternate report showing the space usage in each cluster. If SPACE is not specified on the LIST CLUSTER command, the default is NOSPACE and DLF produces the standard cluster report using only the data in all CPRs.

DSMSPMMSG is called to issue messages if errors are detected during command processing. "LIC" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPLIC.

#### **LIST DOCUMENT Command - DSMSPLID**

**INPUT:** There is no parameter list for calling DSMSPLID; all required information is communicated to DSMSPLID via SPG fields.

On entry to DSMSPLID, DSMSPLIS has already initialized the List Processor Common Work Area (LST). This means that all flag bytes and return and reason code fields have been cleared. The LST is located at a displacement of 4K into the SPG16K buffer of the SPG.

If DSMSPLID is called from DSMSPDAD (via DSMSPLIS), SPGCOMM is zero and the Document Entry Record (DER) to be listed is in the SPGDIBUF field of the SPG.

If DSMSPLID is called to process the LIST DOCUMENT command, SPGCOMM contains the command code for LIST commands (CMDLIS - 11) and DSMSPLID reads the DERs to be listed from the requested library directory data set (via DSMSPLBI for the input document library or DSMSPOLU for the output document library). The DERs that are listed depend on the current contents of other SPG fields set either when the current user issued the AUTH command or when the LIST DOCUMENT command was syntax-checked and validated.

**DESCRIPTION:** DSMSPLID lists data from Document Entry Records (DERs).

DSMSPLIS (the controlling module for LIST command processors) calls DSMSPLID when a LIST DOCUMENT command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT, passed preliminary screening by DSMSPLIS, and is ready for processing. DSMSPLID can also be called (via DSMSPLIS) by the ALTER DOCUMENT

command processor (DSMSPDAD) to list data from the DER just altered. DSMSPDAD calls DSMSPLID (via DSMSPLIS) to list DER data only if the LIST option is specified (via job step JCL) for the current DLF session. (See the discussion of the DSMSPEXC and DSMSPPIN modules under "DLF Initialization, Control, and Termination" on page 46 for additional information regarding the LIST option.)

When DSMSPLID is called to process the LIST DOCUMENT command, the Document Entry Records (DERs) it lists are determined by the options specified in the command and whether the current user is an administrator or a general user.

**OUTPUT:** There is only one exit from DSMSPLID. It is either after listing information from a single DER in SPGDIBUF (as requested by DSMSPDAD via DSMSPLIS) or after listing information from selected DERs (as requested by a LIST DOCUMENT command).

The SPACE option of the SPACE/NOSPACE operand pair causes DLF to use the data in the DER(s) to produce an alternate report showing the effectiveness of space usage by each document and includes two summary lines. If SPACE is not specified on the LIST DOCUMENT command, the default is NOSPACE and DLF produces the Standard document report.

DSMSPMSG is called to issue messages if errors are detected during command processing. "LID" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPLID.

#### **LIST MAP Command - DSMSPLIM**

**INPUT:** There is no parameter list for calling DSMSPLIM; all required information is communicated to DSMSPLIM via SPG fields.

On entry to DSMSPLIM, DSMSPLIS has already initialized the List Processor Common Work Area (LST). This means that all flag bytes and return and reason fields have been cleared. The LST is located at a displacement of 4K into the SPG16K buffer of the SPG.

If DSMSPLIM is called from DSMSPDAM (via DSMSPLIS), SPGCOMM is zero and the User Mapping Record (UMR) to be listed is in the SPGDIBUF field of the SPG.

If DSMSPLIM is called to process the LIST MAP command, SPGCOMM contains the command code for LIST commands (CMDLIS - 11) and DSMSPLIM reads the UMRs to be listed from the requested library directory data set (via DSMSPLIU for the input document library and DSMSPOLU for the output document library).

**DESCRIPTION:** DSMSPLIM lists data from User Mapping Records (UMRs).

DSMSPLIM is called by DSMSPLIS (the controlling module for LIST command processors) when a LIST MAP command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT, passed preliminary screening by DSMSPLIS, and is ready for processing. DSMSPLIM can also be called (via DSMSPLIS) by the DEFINE/ALTER MAP command processor (DSMSPDAM) to list data from the UMR just defined or altered. DSMSPDAM calls DSMSPLIM (via DSMSPLIS) to list UMR data only if the LIST option is specified (via job step JCL) for the current DLF session. (See descriptions of DLF modules DSMSPEXC and DSMSPPIN under "DLF Initialization, Control, and Termination" on page 46 for additional information about the LIST option.)

**OUTPUT:** There are two exits from DSMSPLIM. The first is after listing the information from the single UMR in SPGDIBUF (as requested by DSMSPDAM via DSMSPLIS). The second is after listing information from all UMRs requested by a LIST MAP command.

DSMSPMSG is called to issue messages if errors are detected during command processing. "LIM" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPLIM.

#### **LIST PROCESS Command - DSMSPLIF**

**INPUT:** There is no parameter list for calling DSMSPLIF; all required information is communicated to DSMSPLIF via SPG fields.

On entry to DSMSPLIF, DSMSPLIS has already initialized the List Processor Common Work Area (LST). This means that all flag bytes and return and reason fields have been cleared. The LST is located at a displacement of 4K into the SPG16K buffer of the SPG.

If DSMSPLIF is called from DSMSPDAF (via DSMSPLIS), SPGCOMM is zero and the Processor Profile Record (PPR) to be listed is in the SPGDIBUF field of the SPG.

If DSMSPLIF is called to process the LIST PROCESS command, SPGCOMM contains the command code for LIST commands (CMDLIS - 11) and DSMSPLIF reads the PPRs to be listed from the requested library directory data set (via DSMSPILU for the input document library and DSMSPOLU for the output document library).

**DESCRIPTION:** DSMSPLIF lists data from Processor Profile Records (PPRs).

DSMSPLIF is called by DSMSPLIS (the controlling module for LIST command processors) when a LIST PROCESS command has been read by DSMSPDXC, validated by DSMSPCMD, initialized by DSMSPINT, passed preliminary screening by DSMSPLIS, and is ready for processing. DSMSPLIF can also be called (via DSMSPLIS) by the DEFINE/ALTER PROCESS command processor (DSMSPDAF) to list data from the PPR just defined or altered. DSMSPDAF calls DSMSPLIF (via DSMSPLIS) to list PPR data only if the LIST option is specified (via job step JCL) for the current DLF session. (See descriptions of DLF modules DSMSPDXC and DSMSPPIN under "DLF Initialization, Control, and Termination" on page 46 for additional information about the LIST option.)

**OUTPUT:** There are two exits from DSMSPLIF. The first is after listing information from the single PPR in SPGDIBUF (as requested by DSMSPDAF via DSMSPLIS). The second is after listing information from all PPRs requested by a LIST PROCESS command.

DSMSPMSG is called to issue messages if errors are detected during command processing. "LIF" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPLIF.

#### **LIST SYSTEM Command - DSMSPLIS (LISTSYS Subroutine)**

**INPUT:** There is no parameter list for calling LISTSYS; all required information is communicated to DSMSPLIS via SPG fields.

On entry to the LISTSYS subroutine, DSMSPLIS has already initialized the List System Common Work Area (LST). This means that all flag bytes and return and reason fields have been cleared. The LST is located at a displacement of 4K into the SPG16K buffer of the SPG.

The LISTSYS subroutine of DSMSPLIS is only called to process the LIST SYSTEM command. SPGCOMM contains the command code for LIST commands (CMDLIS - 11) and LISTSYS reads the SPR to be listed from the DLF input or output document library (via DSMSPILU for the input document library or DSMSPOLU for the output document library).

**DESCRIPTION:** The LISTSYS subroutine lists data from a System Profile Record (SPR).

LISTSYS is a subroutine called by DSMSPLIS (the controlling module for LIST command processors) when a LIST SYSTEM command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT, passed preliminary screening by DSMSPLIS, and is ready for processing.

**OUTPUT:** There is one exit from LISTSYS. It occurs after listing information from the SPR requested by a LIST SYSTEM command.

DSMSPMSG is called to issue messages if errors are detected during command processing. "LIS" will appear as the 4th through 6th characters of the message ID of messages initiated from LISTSYS.

#### LIST USER Command - DSMSPLIU

**INPUT:** There is no parameter list for calling DSMSPLIU; all required information is communicated to DSMSPLIU via SPG fields.

On entry to DSMSPLIU, DSMSPLIS has already initialized the List Processor Common Work Area (LST). This means that all flag bytes and return and reason fields have been cleared. The LST is located at a displacement of 4K into the SPG16K buffer of the SPG.

If DSMSPLIU is called from DSMSPDAU (via DSMSPLIS), SPGCOMM is zero and the User Profile Record (UPR) to be listed will be in the SPGDIBUF field of the SPG.

If DSMSPLIU is called to process the LIST USER command, SPGCOMM contains the command code for LIST commands (CMDLIS - 11) and DSMSPLIU reads the UPRs to be listed from the requested library directory data set (via DSMSPLIU for the input document library and by DSMSPOLU for the output document library). The UPRs listed depend on the current contents of other SPG fields set either when the current user issued the AUTH command or when the LIST USER command was validated.

**DESCRIPTION:** DSMSPLIU lists data from User Profile Records (UPRs).

DSMSPLIS (the controlling module for LIST command processors) calls DSMSPLIU when a LIST USER command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT, passed preliminary screening by DSMSPLIS, and is ready for processing. DSMSPLIU can also be called (via DSMSPLIS) by the DEFINE/ALTER USER command processor (DSMSPDAU) to list data from the UPR just defined or altered. DSMSPDAU calls DSMSPLIU (via DSMSPLIS) to list UPR data only if the LIST option is specified (via job step JCL) for the current DLF session. (See descriptions of DLF modules DSMSPEXC and DSMSPPIN under "DLF Initialization, Control, and Termination" on page 46 for additional information regarding the LIST option.)

When DSMSPLIU is called to process the LIST USER command, the User Profile Records (UPRs) it lists are determined by:

- The library user number(s), if any, specified in the command
- Whether the current user is an administrator or a general user.

General users and administrators can specify a range of users in the LIST USER command. If the current user is an administrator, the command lists all user profiles in the specified inclusive range. However, if the current user is a general user, only the user profiles of his own library, his project library, or any public library within the specified/implied range will be listed.



**OUTPUT:** There are two exits from DSMSPLIU. The first is after listing information from the single UPR in SPGDIBUF (as requested by DSMSPDAU via DSMSPLIS). The second is after listing information from selected UPRs requested by a LIST USER command.

DSMSPMSG is called to issue messages if errors are detected during command processing. "LIU" will be the 4th through the 6th characters of the message ID of messages initiated from DSMSPLIU.

#### **PASSWORD Command - DSMSPPAS**

**INPUT:** Operands of the PASSWORD command are placed into fields in the SPG. DSMSPPAS analyzes these fields to determine how the PASSWORD command is to be processed. The UPR of the current user is accessed from the output library.

**DESCRIPTION:** DSMSPPAS is called by DSMSPEXC when a PASSWORD command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT and is ready for processing.

DSMSPPAS reads the UPR of the current user from the output library and modifies (or removes if NOPASS is specified) the password in that UPR as specified in the PASSWORD command that is being processed. The UPR is then written back to the output library.

DSMSPPAS calls DSMSPOLU to access the UPR of the current user in the output library.

**Note:** The password specified in the PASSWORD command is hashed by DSMSPHSH when called by the DLF command initialization routine (DSMSPINT) before DSMSPPAS gets control. The hashed password will be in the SPGPASS1 field in the SPG.

**OUTPUT:** The password of the current user is modified or eliminated as specified in the PASSWORD command being processed. The password is changed in the UPR of the current user in the output library (not the input library).

DSMSPMSG is called to issue messages if errors are detected during command processing. "PAS" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPPAS.

#### **PROTECT Command - DSMSPPRO**

**INPUT:** Operands of the PROTECT command are placed into fields in the SPG. DSMSPPRO analyzes these fields to determine how the PROTECT command is to be processed. The DER of the document specified in the PROTECT command is accessed from the output library.

**DESCRIPTION:** DSMSPPRO is called by DSMSPEXC when a PROTECT command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT and is ready for processing.

DSMSPPRO reads the DER of the document specified in the PROTECT command and modifies the password or the share status of that document.

**Note:** The password(s) specified in the PROTECT command are hashed by DSMSPHSH when called by the DLF command initialization routines (DSMSPINT) prior to DSMSPPRO getting control. The hashed passwords are in the SPGPASS1 and SPGPASS2 fields of the SPG (SPGPASS1 contains the old password, if any, and SPGPASS2 contains the new password, if any).

**OUTPUT:** The password and/or share status of the specified document is modified as specified in the PROTECT command being processed.

DSMSPPMSG is called to issue messages if errors are detected during command processing. "PRO" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPPRO.

#### **PURGE Commands (Controller) - DSMSPPUR**

**INPUT:** There is no parameter list for calling DSMSPPUR; all required information is communicated to DSMSPPUR via SPG fields.

Operands of a PURGE command are placed into fields in the SPG. The subcommand field (SPGSUBC) in the SPG contains a code that indicates the proper PURGE command processor. The fields in SPGPARGA contain selection criteria (if required) to be used by a particular PURGE command processor.

**DESCRIPTION:** DSMSPPUR is the selection and control module for the DLF PURGE functions.

DSMSPPUR is called by DSMSPEXC when a PURGE command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPCMD, and is ready for processing.

DSMSPPUR determines whether the proper conditions are set for the command, ensures that the output library is open, and ensures that the current user is authorized to issue the command. For valid requests, DSMSPPUR calls the appropriate processor:

- PURGE CLASS - DSMSPPUL
- PURGE CLUSTER - DSMSPPUC
- PURGE DOCUMENT - DSMSPPUD
- PURGE MAP - DSMSPPUM
- PURGE PROCESS - DSMSPPUF
- PURGE USER - DSMSPPUU.

**OUTPUT:** See output from the various PURGE command functions.

DSMSPPMSG is called to issue messages if errors are detected during command processing. "PUR" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPPUR.

#### **PURGE CLASS Command - DSMSPPUL**

**INPUT:** Operands of the PURGE CLASS command are placed into fields in the SPG. DSMSPPUL analyzes these fields to determine how the PURGE CLASS command is to be processed.

The existing CER of the class is read from the output library.

**DESCRIPTION:** DSMSPPUL processes the PURGE CLASS command.

DSMSPPUL is called by DSMSPPUR when a PURGE CLASS command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPCMD, passed preliminary screening by DSMSPPUR, and is ready for processing.

DSMSPPUL reads the CER to insure its existence and erases the CER.

DSMSPPUL calls DSMSPOLU to read and erase the CER in the output library.

**OUTPUT:** The CER of the specified class is purged from the output library.

DSMSPPMSG is called to issue messages if errors are detected during command processing. "PUL" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPPUL.

#### **PURGE CLUSTER Command - DSMSPPUC**

**INPUT:** Operands of the PURGE CLUSTER command are placed into fields in the SPG. DSMSPPUC analyzes these fields to determine how the PURGE CLUSTER command is to be processed.

The existing CPR of the cluster is read from the output library.

**DESCRIPTION:** DSMSPPUC processes the PURGE CLUSTER command.

DSMSPPUC is called by DSMSPPUR when a PURGE CLUSTER command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT, passed preliminary screening by DSMSPPUR, and is ready for processing.

DSMSPPUC processes the PURGE CLUSTER command by ensuring that the cluster being purged is the active output cluster (and not the default 4K cluster, which cannot be purged). If necessary, DSMSPPUC calls DSMSPGCL to activate the specified cluster. The CPR is read to, make sure it exists, and erased. The purged cluster is closed for input and output. The cluster name of the active output CLB, the CLB of the purged cluster, is erased.

DSMSPPUC calls DSMSPOLU to read and erase the CPR in the output library.

**OUTPUT:** The CPR of the specified cluster is purged from the output library and the cluster name erased from the CLB for that cluster.

DSMSPPMSG is called to issue messages if errors are detected during command processing. "PUC" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPPUC.

#### **PURGE DOCUMENT Command - DSMSPPUD**

**INPUT:** Operands of the PURGE DOCUMENT command are placed into fields in the SPG. DSMSPPUD analyzes these fields to determine how the PURGE DOCUMENT command is to be processed.

The existing DER (more than one DER if the document has multiple versions), the library owner's UPR, the document owner's UPR, and the LER (if the document is stored in a public library not owned by the document owner) can be read from the output library.

Which directory records are read and how they are processed depends on the current content of other SPG fields that were set either when the current user issued the AUTH command or when the PURGE DOCUMENT command was checked for syntax errors and validated.

**DESCRIPTION:** DSMSPPUD processes the PURGE DOCUMENT command.

DSMSPPUR calls DSMSPPUD when a PURGE DOCUMENT command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT, passed preliminary screening by DSMSPPUR, and is ready for processing.

DSMSPPUD purges a single document or multiple versions of a single document from the document library.

DSMSPPUD processes the PURGE DOCUMENT command by determining if the current user has the authority to perform the purge and that command parameters are correct. For each unlocked version, DSMSPPUD calls the user exit, erases the DER, and frees the

document space in a source data set. When all versions have been successfully purged, it updates the space allocation field in the UPR of the document owner and, if the document was purged from a public library not owned by the owner of the document, updates the LER in the document owners library associated with that public library.

DSMSPPUD calls DSMSPOLU to read, write and erase directory records in the output library, calls DSMSPLSV to update the document owners space allocation, calls DSMSPEXI to call the user exit, and calls DSMSPGCL to ensure the cluster on which the document is stored is active and open for output.

#### **OUTPUT:**

- The DERs of all purged versions of the document are erased from the directory.
- The space previously occupied by the document is placed on the free space queue for the cluster from which the document was purged.
- The space allocation field of the document owner's UPR is updated.
- If the document was purged from a public library, and was owned by a user other than the owner of that public library, the LER associated with that public library in the document owner's library has its document count decremented. The LER is erased if the count was zero after decrementing.
- DSMSPPMSG is called to issue messages if errors are detected during command processing. "PUD" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPPUD.

#### **PURGE MAP Command - DSMSPPUM**

**INPUT:** Operands of the PURGE MAP command are placed into fields in the SPG. DSMSPPUM analyzes these fields to determine how the PURGE MAP command is to be processed.

The existing UMR and its associated UPR are read from the output library.

**DESCRIPTION:** DSMSPPUM processes the PURGE MAP command.

DSMSPPUM is called by DSMSPPUR when a PURGE MAP command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT, passed preliminary screening by DSMSPPUR, and is ready for processing.

DSMSPPUM processes the PURGE MAP command by reading both the UMR to be purged and that UMR's associated UPR. The UMR is purged and DSMSPPMSG is called to issue a message indicating that the purge has taken place.

A user exit is available for the command allowing scanning of the UMR and the associated UPR. The user exit might only change the UPRAPPL field, causing an updated UPR to be written.

DSMSPPUM calls DSMSPOLU to read, write and erase directory records in the output library and calls DSMSPEXI to call the user exit.

**OUTPUT:** The UMR specified in the PURGE MAP command is erased from the output library. The UPRAPPL field in the UPR associated with the purged UMR might be changed by the user exit.

DSMSPPMSG is called to issue messages if errors are detected during command processing. "PUM" will appear as the 4th through

6th characters of the message ID of messages initiated from DSMSPPUM.

#### **PURGE PROCESS Command - DSMSPPUF**

**INPUT:** Operands of the PURGE PROCESS command are placed into fields in the SPG. DSMSPPUF analyzes these fields to determine how the PURGE PROCESS command is to be processed.

The existing PPR of the processor is read from the output library.

**DESCRIPTION:** DSMSPPUF processes the PURGE PROCESS command.

DSMSPPUF is called by DSMSPPUR when a PURGE PROCESS command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT, passed preliminary screening by DSMSPPUR, and is ready for processing.

DSMSPPUF reads the PPR of the processor to be purged - making sure it exists. The PPR is purged and DSMSPPMSG is called to issue a message indicating that the purge has taken place.

DSMSPPUF calls DSMSPOLU to read and erase the PPR in the output library.

**OUTPUT:** The PPR of the specified processor will have been purged from the output library.

DSMSPPMSG is called to issue messages if errors are detected during command processing. "PUF" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPPUF.

#### **PURGE USER Command - DSMSPPUU**

**INPUT:** Operands of the PURGE USER command are placed into fields in the SPG. DSMSPPUU analyzes these fields to determine how it must process this PURGE USER command.

Existing directory records (UPRs, UMRs, DERs, and LERs) associated with the user to be purged are read from the output library.

**DESCRIPTION:** DSMSPPUU is the command processor for the PURGE USER command.

DSMSPPUR calls DSMSPPUU when a PURGE USER command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT, passed preliminary screening by DSMSPPUR, and is ready for processing.

DSMSPPUU makes sure the current user is:

- Authorized to purge the user specified in the command
- Not accidentally trying to erase his own UPR, UMRs, and documents...

The user exit for PURGE USER is called. All UMRs for the purged user are located and erased by searching the full list of UMRs in the output directory. All documents in the purged user's own library are purged. Space corrections for other owners are made when the purged user owns either a public or project library in which they own documents. The documents owned by the purged user in any public or in his project library are transferred to the ownership of the library in which they reside. Finally, the UPR of the purged user is erased.

**DSMSPPUU calls:**

- DSMSPOLU to read, write and erase directory records in the output library
- DSMSPEXI to call the user exit
- DSMSPGCL to ensure that the cluster containing the documents to be purged is active.

**Note:** DSMSPPUU may also call DSMSPLSV to update the space usage fields of the UPRs of users who had documents stored in the purged user's library (the purged user's library must have been a public or project library in this case).

**OUTPUT:**

- The UPR for the purged user will be erased.
- All UMRs associating an external user with the purged user are erased.
- The DER's for all documents in the purged user's own library are erased, and their source data set space placed on the appropriate free space queue.
- All LER's in the purged users library are purged.
- All documents owned by the purged user in other public libraries or his own project library are transferred to the ownership of the library in which they reside.
- The space usage fields of the UPRs of users who owned documents in the purged users public or project library are updated. The LERs of users owning documents in a purged public library are erased.
- DSMSPMMSG is called to issue messages if errors are detected during command processing. "PUU" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPPUU.

#### **READ Command - DSMSPREP**

**INPUT:** Operands of the READ command are placed into fields in the SPG. DSMSPREP analyzes these fields to determine how the READ command is to be processed.

**DESCRIPTION:** DSMSPREP is called by DSMSPEXC when a READ command has been received (from a program which is using DLF as a subroutine), validated by DSMSPCMD, initialized by DSMSPINT and is ready for processing.

DSMSPREP sets up the calling sequence to call DSMSPSIC to READ a record (SLE) from a specific DLF document.

DSMSPREP sets up the SIC parameter list according to the operands specified on the READ command. On return from DSMSPSIC, DSMSPREP moves the record (SLE) read by DSMSPSIC to the calling program's buffer. The address of this buffer is passed to DLF by the calling program. (See "External Interfaces" on page 24 for additional information.)

**OUTPUT:** DSMSPSIC reads the first/next record (SLE) from the specified DLF document and DSMSPREP moves the record into the buffer of the calling program.

DSMSPMMSG is called to issue messages if errors are detected during command processing. "RED" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPREP.

## **RETRIEVE Commands (3 commands) - DSMSPRET**

**INPUT:** Operands of the RETRIEVE command are placed into fields in the SPG. DSMSPRET analyzes these fields to determine how it must process this RETRIEVE command. There are three retrieve commands, RETRIEVE DOCUMENT, RETRIEVE NAMELIST, and RETRIEVE ALL.

Depending on the specific RETRIEVE command, one or more documents are read from the archive data set specified in the RETRIEVE command.

The RETRIEVE NAMELIST command requires DSMSPRET to retrieve documents that are named in a list. The list can be in an external data set or in the input stream. Records from the namelist, validated by DSMSPCMD, are read by DSMSPRDC.

**DESCRIPTION:** DSMSPRET is called by DSMSPEXC when a RETRIEVE command has been read by DSMSPEXC, validated by DSMSPCMD, initialized by DSMSPINT and is ready for processing.

DSMSPRET controls the processing of the RETRIEVE command. One or more archived documents are retrieved from an external archive data set back into the output DLF library.

DSMSPRET insures that the current user is authorized to retrieve the specified document into the specified or implied library. It reads records of the retrieved documents from the archive data set directly via GET macros in the OS/VS2 environment and uses DSMSPDOS to read the records of the document(s) from the archive data set in the VSE environment. DSMSPOLU is used for I/O to the DLF output library.

There are three retrieve commands, RETRIEVE DOCUMENT, RETRIEVE NAMELIST, and RETRIEVE ALL. A single document is retrieved with the RETRIEVE DOCUMENT command. A group of documents named in a list are retrieved with the RETRIEVE NAMELIST command. All documents that meet certain selection criteria (i.e., user number range, data type, specified date) are retrieved with the RETRIEVE ALL command.

The RETRIEVE NAMELIST command requires DSMSPRET to retrieve documents that are named in a list. This list can be in an external data set or in the input stream. DSMSPRDC is used to read records from the namelist. DSMSPRET uses DSMSPCMD to tokenize each namelist entry.

**Note:** After the control intervals and DER of a document have been retrieved from the specified/implied external data set into the DLF output library, the DER of the document is modified to reflect the fact that the document is no longer archived and is written back to the output library. The UPR of the user who owns the document is updated to reflect the proper DASD space use for that user in the DLF library.

After authorization checks and setup, but before actually retrieving the document, DSMSPRET calls DSMSPEXI (the user exit controller) to invoke the user exit. The user exit might elect not to retrieve the document.

**OUTPUT:** The specified documents are retrieved from the specified/implied external data set or sets and their DERs modified as indicated above. The UPRs of the owners of the retrieved documents are updated to reflect the change in DASD space usage resulting from the RETRIEVE command.

DSMSPMMSG is called to issue messages if errors are detected during command processing. "RET" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPRET.

## COMMAND TERMINATION

The command termination function is performed by one module and is described below.

### Command Termination - DSMSPTRM

**INPUT:** DSMSPTRM determines the command it is to terminate and the processes it is to perform by fields in the SPG that were set during command validation, initialization, and processing.

**DESCRIPTION:** DSMSPTRM performs command termination processing, gaining control from DSMSPEXC (the processing controller of DLF) after a command has been processed (or an error encountered during processing). No other modules call DSMSPTRM.

The primary function of this module is to close any data sets specifically required by this command that were opened by DSMSPINT during command initialization.

DSMSPTRM also terminates (via VSAM ENDREQ) any outstanding requests against the input or output library, dequeues (OS/VS2) or unlocks (VSE) the input and/or output library directory (if an enqueue/lock is outstanding against them), and, if terminating a SCRIPT command, calls DSMSPSIC to close any data sets and/or DLF documents that were used for SCRIPT command processing and have not yet been closed.

**Note:** If SPGSSTAR=ON (data was being read from the input stream for the command being terminated), then DSMSPTRM resets the EOD address in the SYSIN/SYSRDR DCB (located at INDCB label in DSMSPEXC and pointed to by the SPGIDCBP field in the SPG) to the address of DATAEND, that is also within DSMSPEXC. The EOD address has to be reset because the read/concatenate routines (DSMSPRDC) use the INDCB and change the EOD address when reading data from the input stream.

#### OUTPUT:

- External data sets opened at command initialization (by DSMSPINT) are closed.
- Any requests outstanding against the input and/or output library directory are terminated via the ENDREQ macro.
- Any enqueue (OS/VS2) or lock (VSE) outstanding against the input and/or output library are dequeued (OS/VS2) or unlocked (VSE).
- If terminating a SCRIPT command, any DLF documents or external data sets used during SCRIPT processing that are still open are closed and their work space cleared by a call to DSMSPSIC.
- If the terminated command was reading from the input stream (SPGSSTAR=ON), the EOD address of the INDCB (hard-coded in DSMSPEXC) is reset to the address of the DATAEND routine in DSMSPEXC.
- DSMSPPMSG is called to issue messages if errors are detected during processing. "TRM" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPTRM.



## SERVICES

The source modules discussed in the following section provide essential services for:

- Modules that perform DLF initialization, control, and termination
- Modules that perform command validation, initialization, processing, and termination functions.

### External File Input Services - DSMSPDSI

**INPUT:** DSMSPDSI reads data to be imported from an external data set.

When DSMSPDSI gets control, various fields in the SPG are set that indicate to DSMSPDSI the required processing. These fields are set either when the IMPORT command is validated or by DSMSPIMP.

**DESCRIPTION:** DSMSPDSI provides external file input services for DSMSPIMP (the IMPORT command processor).

DSMSPDSI reads records from an external data set, converts records to DLF format (SLEs), and blocks these SLEs within VSAM control intervals (CIs).

DSMSPIMP calls DSMSPDSI during IMPORT command processing each time DSMSPIMP is ready for another control interval. No other modules call DSMSPDSI. DSMSPDSI uses DSMSPRDC to get data from the input stream (if FROM(\*) is specified on the IMPORT command). If input is to come from an external sequential data set instead of the input stream, DSMSPDSI uses DSMSPDOS to get data from the data set in a VSE environment or issues the GET macro directly to get data from the data set in an OS/VS2 environment.

If a special processor is specified (PROCESS operand of IMPORT command) or implied (INDATA operand on IMPORT command), DSMSPDSI calls the processor each time a record (SLE) is ready to be blocked into a control interval. The special processor can add, delete and/or modify records. It can also be used to supply all the records of the document being imported. See "External Interfaces" on page 24 for information on the interface between DSMSPDSI and a special processor.

**OUTPUT:** The SPG16K buffer in the SPG contains a formatted control interval containing records read from the external file.

DSMSPMSG is called to issue messages if errors are detected during command processing. "DSI" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPDSI.

### External File Output Services - DSMSPDSO

**INPUT:** When DSMSPDSO gets control, various fields in the SPG are set that indicate to DSMSPDSO the processing required. These fields are set either when the EXPORT command is validated or by DSMSPEXP.

The SPG16K buffer in the SPG contains a control interval that is to be deblocked and output to the external data set.

**DESCRIPTION:** DSMSPDSO provides external file output services for DSMSPEXP (the EXPORT command processor).

DSMSPDSO converts data from a control interval of a DLF document into external data format and controls the output of that data.

DSMSPDSO is called by DSMSPEXP during EXPORT command processing each time DSMSPEXP is ready to export another control interval. No other modules call DSMSPDSO.

If TO(DSMLIST) is specified on the EXPORT command in an OS/VS2 environment or TO(SYSLST) is specified on the EXPORT command in a VSE environment, then DSMSPDSO uses DSMSPMSG to print or list data from the document. If DSMLIST is not specified in an OS/VS2 environment, DSMSPDSO uses the PUT macro (or WRITE/CHECK macros if BSAM OPEN) to output data from the document to the external data set. If SYSLST is not specified in a VSE environment, then DSMSPDSO calls DSMSPDOS to output data to the external data set.

If a special processor is specified (PROCESS operand of EXPORT command) or implied (specification of OUTDATA operand on EXPORT command), then that processor is called by DSMSPDSO each time a record (SLE) is deblocked from a control interval. The special processor can add, delete and/or modify records. See "External Interfaces" on page 24 for information on the interface between DSMSPDSO and a special processor.

**OUTPUT:** The data in the control interval passed to DSMSPDSO by DSMSPEXP is deblocked and output to the external file.

DSMSPMSG is called to issue messages if errors are detected during processing. "DSO" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPDSO.

#### **Dynamic Allocation Services (OS/VS2 only) - DSMSPDYN**

**INPUT:** Register 1 points to an address list as described under "DLF to DCF Interface" on page 36. The parameters pointed to by these addresses identify the data set to be dynamically allocated or unallocated.

**DESCRIPTION:** DSMSPDYN provides dynamic allocation services for DLF when operating in an OS/VS2 environment.

DSMSPDYN is called by four DLF modules (DSMSPGNX, DSMSPQNM, DSMSPSIC, and DSMSPSOP) to allocate/deallocate external (non-library) data sets. It is called to allocate a data set using its fully qualified data set name or to unallocate a data set previously allocated.

**Note:** DSMSPDYN can also be called from modules of the DCF program product to dynamically allocate/unallocate data sets for SCRIPT/VS.

See "DCF to DLF Interfaces" on page 38 for additional information.

**OUTPUT:** The specified data set is dynamically allocated or unallocated as requested. If the data set was allocated, then the ddname for the data set is returned in the field pointed to by the fourth address of the address list pointed to by register 1 on entry to DSMSPDYN. The type of data set organization is returned in the fifth address of the address list pointed to by register 1 on entry to DSMSPDYN.

DSMSPMSG is called to issue messages if errors are detected during processing. "DYN" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPDYN.

## User Exit Controlling Services - DSMSPEXI

**INPUT:** DLF application programs that call DSMSPEXI for a user exit set Register 1 to point to an address field that points to the EXI parameter list as described under "External Interfaces" on page 24. This parameter list is used for communication between the program taking the user exit, DSMSPEXI - the user exit controller, and DSMSPUXI - the user exit itself.

**DESCRIPTION:** DSMSPEXI is the controller for user exits. This module gets control from any program making a user exit.

DSMSPEXI clears necessary security fields in the records to be passed from the calling program and then passes the records to the user exit program (DSMSPUXI). A skeleton DSMSPUXI is furnished by IBM and can be replaced by a user-written program if a user requires exit processing. The skeleton DSMSPUXI simply returns to DSMSPEXI with no action.

DLF application programs that call DSMSPEXI for a user exit set register 1 to point to an address field that points to the EXI parameter list. This parameter list can contain a Document Entry Record (DER) or a User Profile Record (UPR), but not both. If it contains a UPR, a User Mapping Record (UMR) can also be present. DSMSPEXI passes this parameter list on to DSMSPUXI after clearing passwords from the DER or UPR.

On return from DSMSPUXI, DSMSPEXI outputs error messages and sets return codes as requested by the user exit. Return is to the program calling DSMSPEXI.

Additional information on user exits can be found under "External Interfaces" on page 24. Also, the Document Library Facility Guide contains an entire chapter on user exit routines.

**OUTPUT:** DSMSPEXI calls DSMSPMMSG to issue error messages requested by the user exit. The UPR and UMR or DER in the EXI parameter list might be modified by the user exit routines.

## Cluster Control Services - DSMSPGCL

**INPUT:** When DSMSPGCL gains control, various fields in the SPG are set that indicate to DSMSPGCL the processing required. These fields are set by the program that calls DSMSPGCL. The input and output chains of CLBs (pointed to by the SPGCLIFP and SPGCLOFP fields of the SPG) are used.

**DESCRIPTION:** DSMSPGCL performs these three I/O services for library source (VSAM ESDS) data sets:

1. Activates an ESDS source cluster for input and/or output to allow document text to be accessed, stored, or purged
2. Ensures the presence of an updated header record in the first control interval (CI 0) of the cluster
3. Closes the cluster when DLF stops processing, or when the cluster is defined, altered, or purged.

DSMSPGCL opens library source (VSAM ESDS) data sets and generates cluster control blocks (CLBs) that are used by DLF when accessing these data sets. CLBs are chained from the SPG. There are two CLB chains, one for clusters open for input, and one for clusters open for output. The CLB chains are push down chains, meaning that the last CLB entered on a chain is pointed to by the chain head pointer for that chain. The newest CLB on the chain points to the next newest and so on. The input CLB chain head is at SPGCLIFP and the output CLB chain head is at SPGCLOFP. See "Internal DLF Interfaces" on page 23 for additional information.

DSMSPGCL can be called to activate a cluster for input, for output, or to activate both an input and an output cluster with

the same call. If called to activate a cluster for input, DSMSPGCL searches the input CLB chain for the cluster to see if it is already open for input and, if so, activates the cluster and refreshes the FSH of the CLB from CI 0, and makes it the currently active input cluster.

If called to activate a cluster for output, DSMSPGCL searches the output CLB chain for the cluster to see if it is already open for output and, if so, activates the cluster, refreshes the FSH of the CLB from CI 0, and makes it the currently active output cluster. If called to activate both an input and an output cluster, the input cluster is activated first.

If the requested cluster is not found on the CLB chain searched, DSMSPGCL creates a CLB for the cluster, places it on the proper CLB chain, opens the cluster and sets its CLB as the currently active input or output cluster depending on whether the request was for input or output.

DSMSPGCL also does certain validation checking on a newly opened cluster, making sure that CISIZE is the same as that specified in the cluster's CPR and that the timestamp in the clusters first control interval matches the timestamp of the directory (input or output) associated with the cluster.

DSMSPGCL assists DLF in controlling free and unused space on the ESDS source data sets via the Free Space Header (FSH) in CI 0. A current copy of the FSH for each open cluster is kept in the cluster's CLB. The output CLB FSH is updated each time control intervals are taken from the unused or free chain of that cluster to store, copy, or retrieve a document. It is also updated when control intervals are returned to the free control interval chain (as when a document is archived or purged) and when DLF terminates. At the same time the FSH in CI 0 is updated by DSMSPGCL to keep the source cluster current in case of abnormal termination of DLF.

DSMSPGCL reads CI 0 each time it is called to activate an output cluster for three reasons:

1. To make sure the FSH in the CLB is correct prior to a possible document text update
2. To make sure that the same DLF source cluster can be accessed by multiple cluster names and/or DD statements
3. To allow two batch jobs to run against the same library clusters simultaneously.

DSMSPGCL reads CI 0 each time it is called to activate an input cluster to ensure the LIST CLUSTER command processes the current cluster status.

DSMSPGCL closes all clusters when DLF stops processing, or closes a single cluster when that cluster is defined, altered, or purged by a DLF command. To make sure values are current, CI 0 is normally written prior to closing the cluster.

**OUTPUT:** DSMSPMSG is called to issue messages if errors are detected during processing. "GCL" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPGCL.

#### Get Next Record Services - DSMSPGNX

**INPUT:** DSMSPSIC calls DSMSPGNX with register 1 pointing to a list of three addresses:

- The first address points to the SICPLPTR (containing the address of the SIC parameter list in the program that called DSMSPSIC).

- The second address points to FOUNDX, a full word field (declared in DSMSPSIC) containing a number from 1 to 16 that corresponds to the OFN entry number associated with the document, data set, or I/O module to be opened, read from, or have a point operation run against.
- The third address points to INITREAD, a one-byte field declared in DSMSPSIC and used by DSMSPGNX to control when it is to read from the beginning of the document/data-set.

**DESCRIPTION:** DSMSPGNX performs get record or get SLE services for SCRIPT and READ command processing. This module gets control from DSMSPSIC to:

- Read a record from an external data set (SCRIPT command processing only)
- Read an SLE from a DLF document (SCRIPT or READ command processing)
- Get a record from an I/O module (SCRIPT command processing only)
- Re-initialize reading from the beginning of a document or data set (point operation for SCRIPT command processing only).

SCRIPT command processing is controlled by DCF, which uses DLF for source input requests (DLF documents, external data sets, or I/O module data) when DCF operates in a batch environment. Programs calling DLF as a subroutine can read DLF documents one line (SLE) at a time using the READ command and can close (free-up control areas for the document) the document being read when processing is complete. These functions are accomplished by three DLF modules and a number of data areas called source input control. DSMSPGNX is a part of the source input control subsystem of DLF as are DSMSPSIC and DSMSPSOP. DSMSPSIC is the source input controller and gains control either from DLF modules or from the DCF program product (See "External Interfaces" on page 24 for additional information on the interfaces between DCF and DLF.) DSMSPSIC calls DSMSPGNX to get the first/next record of a specified data set or document. If the call is for the first record, DSMSPGNX calls DSMSPSOP to open the data set or document (in the case of documents - open consists only of setting up the dynamic control area and reading the DER).

Special processors (not a part of the DLF program product) can be called during DSMSPGNX processing. If a processor is specified (or implied by the INDATA/OUTDATA type pairs) in the READ or SCRIPT command, then DSMSPGNX calls the processor (BALR to entry point of the loaded program) after each read of a record or SLE. The processor can modify or delete any record/SLE passed to it and the results are reflected in the SCRIPT or READ output. The processor can also insert records/SLEs before or add records/SLEs after the record/SLE passed to it. See "External Interfaces" on page 24 for information on the interface between DSMSPGNX and a special processor.

**OUTPUT:** I/O buffer - DSMSPGNX reads control intervals or physical blocks into the source buffer located within the dynamic storage area for the OFN entry. This area is acquired by DSMSPSOP when the OFN entry is opened (prior to the first SLE/record read). One SLE/record at a time is moved from the source buffer to the communication area that is also located in the dynamic storage area for the OFN entry.

On return from DSMSPGNX for a get next operation, the SICCMREC field of the SIC parameter list points to the SLE/record that moved to the communication area.

**Note:** The dynamic storage area for an OFN entry also contains a work area that can be used by any processor called from DSMSPGNX.

DSMSPMSG is called to issue messages if errors are detected during processing. "GNX" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPGNX.

## **Password Hashing Services - DSMSPHSH**

**INPUT:** When DSMSPHSH is called (by BALR to the address stored in SPGSHSHAD), register 1 must point to a list of two addresses that point to required parameters. The first address points to an eight-byte field containing the password to be hashed and the second address points to an eight-byte field in which the hashed password is to be returned. See "External Interfaces" on page 24 for information on the interface between DSMSPHSH and its callers.

**DESCRIPTION:** DSMSPHSH provides password hashing services for DLF and can also be called by the direct read modules of ATMS-III. DSMSPHSH is a separate load module and is loaded by DSMSPPIN at DLF initialization. The entry point address is placed in the SPGSHSHAD field of the SPG by DSMSPPIN. DSMSPHSH is invoked by a calling routine with a BALR using the address in SPGSHSHAD.

There are four library program modules (DSMSPARC, DSMSPINT, DSMSPRET, and DSMSPSOP) that BALR to DSMSPHSH to hash either a user password or a document password. These passwords are stored in the library in hashed form for additional security.

DSMSPHSH hashes an eight-character password into a 64-bit string that is returned in the caller's parameter list.

**OUTPUT:** The hashed password is returned in the 8-byte parameter field pointed to by the second of two passed addresses as noted under Input.

## **Input Library Access Services - DSMSPILU**

**INPUT:** There is no parameter list used in calling DSMSPILU. When DSMSPILU gets control, various fields in the SPG are set that indicate to DSMSPILU the required processing. The SPG fields set by the calling program include:

- SPGILUI - Processing flags as follows:
  - SPGIESDS - ON = ESDS to be accessed, OFF = KSDS to be accessed
  - SPGIKEQ - ON = Search Key equal (KSDS only)
  - SPGIFKS - ON = Full Key search (KSDS only)
  - SPGIDIR - ON = Direct search (KSDS only), OFF = Sequential
- SPGIACSS - Will Contain "R" (read)
- SPGIKEYL - Key length
- SPGIARG - Pointer to search argument (KSDS only)
- SPGIAREA - Pointer to buffer (KSDS only - SPG16K used for ESDS).

**DESCRIPTION:** DSMSPIU provides access services against the input library (both the directory (KSDS) and source (ESDS) clusters).

DSMSPIU is called by DLF application modules (see "Section III - Module Directory" on page 98 for a complete list of the modules that call DSMSPIU) to access directory (KSDS) records or source (ESDS) cluster control intervals. One directory or source record is accessed each time DSMSPIU is called.

Calling programs store information in the ILU interface fields of the SPG that tell DSMSPIU which record to access.

**OUTPUT:** If the KSDS was accessed, the requested record is in the buffer pointed to by SPGIAREA (see Input above). If the ESDS was accessed, the record is in the SPGL6K buffer.

#### **Library Input Services - DSMSPLBI**

**INPUT:** There is no parameter list used in calling DSMSPLBI. When DSMSPLBI gets control, various fields in the SPG are set that indicate the processing DSMSPLBI must perform. Fields set by the calling program include:

- SPGLBII - Processing flags as follows:
  - SPGREQ - ON = Search Key equal
  - SPGRGE - ON = Search Key equal or greater
  - SPGGR - ON = Generic Search Key
  - SPGRS - ON = Read Sequential
  - SPGRFI - ON = Read data set control interval
  - SPGDR - ON = Search Direct
  - SPGRPL2 - ON = Use SPGIRPL2 instead of SPGIRPL1
- SPGLBIG - Length of generic search key
- SPGDIBUF - Search argument and input buffer (KSDS only - ESDS records are returned in the SPGL6K buffer).

**DESCRIPTION:** DSMSPLBI provides access services against the DLF input library (both the directory and source (ESDS) clusters) and/or against a backup (COPY OUT format) or archive data set.

DSMSPLBI is called by DLF application modules (see "Section III - Module Directory" on page 98 for a complete list of the modules that call DSMSPLBI) to access directory (KSDS) records or source (ESDS) cluster control intervals either from the DLF input library or from a backup/archive data set. One directory or source record is accessed each time DSMSPLBI is called.

Calling programs store information in the LBI interface fields of the SPG to tell DSMSPLBI which record to access.

DSMSPLBI reads records directly from the input library. In the OS/VS2 environment, DSMSPLBI also reads records from the external data set (backup or archive data set) directly. In the VSE environment, DSMSPDOS is called to access the external data set.

**OUTPUT:** If a directory record was accessed, the requested record is in the SPGDIBUF buffer (See Input above). If a control interval was accessed, The record is in the SPGL6K buffer.

## Library Output Services - DSMSPLB0

**INPUT:** There is no parameter list used in calling DSMSPLB0. When DSMSPLB0 gets control, various fields in the SPG are set that indicate the processing required of DSMSPLB0. Fields set by the calling program include:

- SPGLBOI - Processing flags as follows:
  - SPGUPRRD - ON = prepare document owner's UPR for update
  - SPGUPRWT - ON = update document owner's UPR
  - SPGCIANY - ON = write a control interval record
  - SPGCIBEG - ON = prepare to write first control interval record after setting up and testing DER
  - SPGCIEND - On = write last control interval record, DER, and LER, if required.
- SPGDOBUF - Output buffer containing the directory record to process.

**Note:** Control intervals to be written are passed by the calling program in the SPGL6K buffer in the SPG.

**DESCRIPTION:** DSMSPLB0 provides I/O services against the DLF output library (both the directory and source (ESDS) clusters).

DSMSPLB0 is called to store a document in an output library source cluster. Each document consists of a document entry record and one or more control intervals that contain the text of the document. The document entry record and control intervals can be for a new document or a copy of or replacement for an existing document. DSMSPLB0 updates the document owner's user profile record with space usage and document count information, then creates or updates a library entry record when the document is stored in a public library owned by a user other than the document owner. How DSMSPLB0 performs for any one call is governed by the input supplied by the calling program. The COPY and IMPORT command processors call DSMSPLB0 in different ways to perform separate functions.

Calling programs store information in the LBO interface fields of the SPG that tell DSMSPLB0 which record(s) to access or write.

Five DLF command processors call DSMSPLB0:

- DSMSPIMP - IMPORT command processor
- DSMSPCPD - COPY DOCUMENT command processor
- DSMSPCPI - COPY IN command processor
- DSMSPCPL - COPY LIBRARY command processor
- DSMSPCPU - COPY USER command processor.

**OUTPUT:** Records are written to the DLF output library as indicated under **Description** above.

DSMSPMMSG is called to issue messages if errors are detected during processing. "LBO" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPLB0.



## Library Validation Services - DSMSPLSV

**INPUT:** The SPGDOBUF buffer in the SPG contains the UPR for which space usage information is to be calculated.

**DESCRIPTION:** DSMSPLSV provides library space validation services for various DLF command processors. It is called to calculate the library space usage and document count of a user if the UPRNLOG flag in that user's UPR is on. UPRNLOG=ON indicates that the space usage might have changed for the user but is not yet reflected in the UPR. DSMSPLSV calculates this space by reading all the DERs associated with the UPR (that is, DERs that immediately follow the UPR and have the same library number in the key) and accumulating space usage for those owned by this user. It also accumulates space usage from the DERs belonging to this user that are contained in the user's project library or any public library. All documents owned by the user are also counted.

See "Section III - Module Directory" on page 98 for specific modules that call DSMSPLSV.

**Note:** If the count of documents a user owns in a public library is different from the count specified in the LER for that public library (in the LERCOUNT field), then the LER with updated LERCOUNT is written back to the output library. If there are no documents owned by the user in the public library, then the LER is erased.

**OUTPUT:** The UPRDSLIB and UPRDPLIB fields in the UPR in SPGDOBUF are recalculated (by reading all the DERs belonging to this user and accumulating the space used for each document).

DSMSPMSG is called to issue messages if errors are detected during processing. "LSV" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPLSV.

## Message Services - DSMSPMSG

**INPUT:** There is no parameter list for DSMSPMSG. Parameters are passed in register 0, register 1, and the SPG. The type of message processing is determined by the value in register 0:

- Register 0  $\geq 0$  — Issue a message from the message table in DSMSPMSG (canned messages)
  - Register 0 = The message number of the requested message
  - SPGMSGD = Any required fill-in fields for the requested message. DSMSPMSG takes the data from these 4 fields and places it in the proper position in the message, converting numeric data to EBCDIC and removing leading zeros, and using the proper field length depending on the requested message.
- Register 0 = -1 (SPG#PRNT) — Issue a user-generated message (special message)
  - Register 0 = -1
  - Register 1 = Address of the user message that includes any required output control data
- Register 0 = -2 (SPG#LIST) — Print unblocked text records from one control interval (CI print)
  - Register 0 = -2
  - SPG16K = The control interval to be printed
- Register 0 = -3 (SPG#DHEX) — Print a formatted hex dump
  - Register 0 = -3
  - Register 1 = Data length
  - SPG16K = The data to be dumped

**DESCRIPTION:** DSMSPMSG provides library message services and is used by most library program modules. (See "Section III - Module Directory" on page 98 for a list of the modules that call DSMSPMSG.)

DSMSPMSG passes messages and data from DLF to DSMLIST (OS/VS2) or SYSLST (VSE). It handles 4 types of output:

- Predefined informational and error messages (canned messages)
- One-line messages set up by the calling routine (special messages)
- All the text records from one control interval, unblocked (CI print)
- Formatted hex dumps (hex dump).

DSMSPMSG handles the output in the OS/VS2 environment. In the VSE environment, DSMSPDOS is called to output each record.

## OUTPUT

### Normal Completion

- The specified message or other output is passed to DSMLIST (OS/VS2) or SYSLST (VSE).
- The reason (register 0) and return (register 15) codes are always 0.

### Error Exits

- END DSMSPMSG

The calling routine requested a DLF message that does not exist (unused message number 0-146), or a message number higher than 146. The message, "ERROR IN LIBRARY PROGRAM", is printed. DSMSPMSG returns to the calling program with return and reason codes = 0 — the same return and reason codes as a successful completion.

- Control Interval Print (register 0 = -2) makes an emergency exit to DSMSPEXC entry point DSMSPFIN when it is requested to print a control interval having an invalid number of records (CITNLR <= 0 usually indicating a null control interval). Just prior to the exit, DSMSPMSG will print the message "ERROR IN LIBRARY PROGRAM".

## Output Library I/O Services - DSMSPOLU

**INPUT:** There is no parameter list used in calling DSMSPOLU. When DSMSPOLU gets control, fields in the SPG are set to indicate the processing required of DSMSPOLU. Fields set by the calling program include:

- SPGOLUI - Processing flags as follows:
  - SPGOKSDS - ON = KSDS to be accessed
  - SPGOESDS - ON = ESDS to be accessed
  - SPGOKEQ - ON = Search Key equal (KSDS only)
  - SPGOFKS - ON = Full Key search (KSDS only)
  - SPGODIR - ON = Direct search (KSDS only)
  - SPGOINS - ON = Insert processing request (KSDS only)
- SPGOACSS - Access Request Code (R=Read, W=Write, E=Eraser)
- SPGOKEYL - Key length
- SPGORECL - Record length for insert (KSDS only)

- **SPGOARG** - Pointer to search argument (KSDS only)
- **SPGOAREA** - Pointer to buffer (KSDS & ESDS)
- **SPGOARGX** - The RBA of the control interval to be accessed if DSMSPOLU is called to read or write a control interval from the source (ESDS) data set.

**DESCRIPTION:** DSMSPOLU provides I/O services against the output library (both the directory (KSDS) and source (ESDS) clusters).

DLF application modules call DSMSPOLU (see "Section III - Module Directory" on page 98 for a complete list of the modules that call DSMSPOLU) to read, write, or erase directory (KSDS) records or to read or write source (ESDS) records. One directory or source record is read, written, or erased each time DSMSPOLU is called.

Calling programs store information in the OLU interface fields of the SPG to tell DSMSPOLU which record to read, write, or erase.

**OUTPUT:** The requested record is read, written, or erased. If a read was requested, the record is in the buffer pointed to by SGOAREA (see Input above).

#### **Qualify Name Services (OS/VS2 only) - DSMSPQNM**

**INPUT:** On entry to DSMSPQNM, register 1 points to a list of five addresses. These addresses point to parameter fields in the calling program and are set as:

- **First address** — points to a 1-byte field containing request flags. This one-byte field is referenced as QNMFLAGS in DSMSPQNM.
  - Bit 0 — Request is to qualify data set only if this bit (referenced as ONLYQAL) is on and no dynamic allocation will occur
  - Bit 1 — Requests that type of data set organization be returned if this bit is on
  - Bits 2-7 — Not used.
- **Second address** — points to a 44-byte field containing a data set name that might not be fully qualified at entry (see fourth address below). The field is referenced as DSNAME in DSMSPQNM.
- **Third address** — points to an 8-byte field containing a member name if a partitioned data set is to be qualified and/or dynamically allocated. The field is referenced as MEMNAME in DSMSPQNM.
- **Fourth address** — points to an 8-byte field containing the qualifier name if the name pointed to by the second address is not already fully qualified. The field is referenced as QUALNAME in DSMSPQNM.
- **Fifth address** — points to an 8-byte field in which the DDNAME is to be returned to the calling program after dynamic allocation. The field is referenced as DDNAME in DSMSPQNM.
- **Sixth address** — Points to a 2-byte field in which the type of data set organization is to be returned.

See "DCF to DLF Interfaces" on page 38 for additional information.

**DESCRIPTION:** DSMSPQNM provides services (in the OS/VS2 environment only) to qualify a data set name and optionally to call DSMSPDYN for dynamic allocation of the data set.

DSMSPINT calls DSMSPQNM to fully qualify an external data set name and, optionally, to call DSMSPDYN to dynamically allocate that data set. This is done before DSMSPINT opens the external data set during command initialization.

DSMSPQNM is also called by DCF modules to qualify data set names and/or to dynamically allocate data sets required for SCRIPT/VS processing. The entry point address of DSMSPQNM is passed to DSMESBEG in a parameter list when that program is called by DSMSPXEC to do SCRIPT command processing. See "DCF to DLF Interfaces" on page 38 for additional information.

**OUTPUT:** The specified data set is fully qualified and, optionally, the data set is dynamically allocated.

The fully qualified data set name is in the field pointed to by the second address of a list of addresses pointed to by register 1 (same list received by DSMSPQNM at entry). If dynamic allocation was requested, the DDNAME of the allocated data set is in the field pointed to by the fifth of the list of address pointed to by register 1. If the type of data set organization was requested, the type is returned in the field pointed to by the sixth address of the address list pointed to by register 1.

DSMSPMSG is called to issue messages if errors are detected during processing. "QNM" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPQNM.

#### Read Concatenate Services - DSMSPRDC

**INPUT:** Register 1 points to

- A parameter list containing the address of the DCB of the external data set to be read from
- Flags that indicate first-call and end-of-data conditions.

**DESCRIPTION:** DSMSPRDC uses DLF data continuation rules to create a concatenated data record from one or more input records. DSMSPRDC's caller passes the address of the DCB to be used. This DCB is already open for input and controls the data set containing the data to be read. For each successful call to DSMSPRDC, a complete record (with one or more concatenated parts) is returned to the calling program. When DSMSPRDC reaches end-of-file or end-of-data, it indicates this condition to the calling program by setting on the EOF flag in the RDC parameter list.

DSMSPARC and DSMSPRET call DSMSPRDC to read records of a namelist during ARCHIVE or RETRIEVE command processing. The namelist might be in the input stream or in an external data set. If the namelist is in the input stream, DSMSPARC or DSMSPRET sets the DCB pointer to point to INDCB. INDCB is located in DSMSPXEC and pointed to by the SPGIDCBP field of the SPG. If the namelist is in an external data set, DSMSPARC or DSMSPRET sets the DCB pointer to point to SPGNMDCB. SPGNMDCB is in the SPG.

DSMSPRDC is also called by DSMSPDSI to read data from the input stream during IMPORT command processing if FROM(\*) was specified. In this case, DSMSPDSI sets the DCB pointer to point to INDCB, located within DSMSPXEC and pointed to by the SPGIDCBP field of the SPG.

DSMSPRDC calls DSMSPDOS for the actual I/O in a VSE environment, but issues GET macros directly within DSMSPRDC in an OS/VS2 environment.

**OUTPUT:** DSMSPRDC reads (and concatenates, if required) a record from the input source controlled by the DCB passed to DSMSPRDC by the calling program. This record is in the RDCBUFFER buffer within DSMSPRDC and is pointed to by the address in RDCRECRD

(pointer field whose address was passed by the calling program and set by DSMSPRDC).

DSMSPMSG is called to issue messages if errors are detected during processing. "RDC" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPRDC.

#### Source Input Services - DSMSPSIC

**INPUT:** DSMSPSIC is called with register 1 pointing to the address of the SIC parameter list located in the module that called DSMSPSIC. See "Section IV - Data Areas" on page 104 for details of this parameter list.

**DESCRIPTION:** DSMSPSIC performs source input control for SCRIPT and READ command processing. This module gets control from:

- DSMSPRED (for READ command processing)
- DSMSPCLO (for CLOSE command processing)
- DSMLOS30 — OS/VS2 or DSMDLS30 — VSE (for source input for SCRIPT command processing)
- DSMSPTRM (to close documents/data-sets after a SCRIPT operation)
- DSMSPEXC (at DLF termination to free dynamic storage areas used in source input control - including the Open File Name (OFN) table).

The primary function of this module is to provide source input control for SCRIPT and READ operations. SCRIPT commands are processed by DCF, which uses DSMSPSIC for source input requests (either DLF documents or external data sets) when operating in a batch environment. Programs that call DLF as a subroutine can read DLF documents one record (SLE) at a time using the READ command and can close (free-up control areas for the document) the document being read when processing is complete. DSMSPSIC controls all these functions.

DSMSPGNX and DSMSPSOP are also part of the source input control subsystem. DSMSPSIC calls DSMSPGNX to get the first/next record of a specified data set or document. If this call is made to get the first record, DSMSPGNX in turn calls DSMSPSOP to open the data set or document (in the case of documents - open consists only of setting up the dynamic control area and reading the DER).

DSMSPSIC calls the following modules:

- DSMSPDOS (VSE environments only) — Called to close any external data sets used for SCRIPT command processing. In OS/VS2 environments, these data sets are closed within DSMSPSIC.
- DSMSPDYN (OS/VS2 only) — Called to unallocate dynamically allocated data sets (after close).
- DSMSPGNX — Called to get the first/next record from a data set or DLF document.

**Note:** DSMSPSIC can be called to reposition to a previous record (point function).

- DSMSPMSG — Called to issue error messages.

## OUTPUT

- If this is a read request, the SICCMREC field in the SIC parameter list contains the address of the SLE/record that was read.
- If this is a note request, SICNPREC in the SIC parameter list contains the relative record number of the last record/SLE read.
- If this is a point request, the record/SLE preceding the requested point record/SLE is read and next read will cause the point record to be read.
- If this is a CLOSE or CLOSE ALL request, then the external data set(s) (if any) in the OFN are closed and, in OS/VS2 only, dynamically unallocated if they were previously dynamically allocated. Dynamic storage used to control access of the document or external data set is released.
- If DSMSPSIC was called for a release storage request, then the OFN as well as all of the dynamic storage areas pointed to by OFN entries are released.

Explicit error messages are output (via DSMSPMSG) should errors be detected during processing. "SIC" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPSIC.

## SMF Output Services - DSMSPSMF

**INPUT:** There is no parameter list for calling DSMSPSMF, all required information is communicated to DSMSPSMF via SPG fields.

SPGDOBUF contains the UPR from which accounting information is to be taken.

**DESCRIPTION:** DSMSPSMF provides SMF output services for the ACCOUNT command processor (DSMSPACT). DSMSPACT calls DSMSPSMF once for each UPR selected in response to an ACCOUNT command. When DSMSPSMF gets control the UPR is in the SPGDOBUF buffer of the SPG. DSMSPSMF builds and outputs a type-47 SMF record from information in the UPR.

The account information is written to the data set specified in the TO operand of the ACCOUNT command. If TO is not specified, the data set described by the JCL statement labeled SMF (OS/VS2) or DSMACNT (VSE) is used. The account data set is a sequential OS data set in OS/VS2 environments and is an entry sequenced VSAM data set in VSE environments.

DSMSPSMF is called by DSMSPACT each time a UPR is found that matches the selection criteria for the ACCOUNT command processing.

**OUTPUT:** A type-47 accounting record is output (to the account data set) for the UPR that DSMSPACT passed to DSMSPSMF in the SPGDOBUF buffer of the SPG.

DSMSPMSG is called to issue messages if errors are detected during processing. "SMF" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPSMF.

## Source Input Open Services - DSMSPSOP

**INPUT:** DSMSPSOP is called by DSMSPGNX with register 1 pointing to a pair of addresses. The first address points to the SICPLPTR (continuing the address of the SIC parameter list in the program that called DSMSPSIC). The second address points to FOUNDX, a full word field (declared in DSMSPSIC) containing a number from 1 to 16 that corresponds to the OFN entry number associated with the document, data set, or I/O module to be opened.

See "Section IV - Data Areas" on page 104 for additional information on the SIC parameter list.

**DESCRIPTION:** DSMSPSOP performs open services for SCRIPT and READ command processing. This module gets control from DSMSPGNX to open an external data set (SCRIPT command processing only), open a DLF document (SCRIPT or READ command processing), or open an I/O module (valid only if processing a SCRIPT command and DLF has been called as a subroutine).

SCRIPT command processing is controlled by DCF (which uses DLF for source input requests, DLF documents, external data sets, or I/O module data) when operating in a batch environment. Programs that call DLF as a subroutine can read DLF documents one line (SLE) at a time using the READ command and can close (free-up control areas for the document) the document being read when processing is complete. These functions are accomplished by a source input control (SIC) subsystem that consists of three DLF modules and a number of data areas.

DSMSPSOP is part of the source input control subsystem of DLF, as are DSMSPSIC and DSMSPGNX. DSMSPSIC is the source input controller and gains control either from DLF modules or from the DCF program product. DSMSPSIC calls DSMSPGNX to get the first/next record of a specified data set or document. If the call is made to get the first record, DSMSPGNX calls DSMSPSOP to open the document, data set or I/O module.

**Note:** In the case of documents - the open operation consists of setting up the dynamic control area and reading the DER. In the case of I/O modules, the open operation consists of loading the module and setting up the dynamic control area.

Special processors (not part of the DLF program product) can be loaded during DSMSPSOP processing. If a processor is specified (or implied by the INDATA/OUTDATA pair of data types) in the READ or SCRIPT command, then DSMSPSOP loads the processor and retains its entry point in the OFN entry to which it pertains. The processor can modify or delete any record/SLE passed to it and the results are reflected in the SCRIPT or READ output. The processor can also insert records/SLEs before or add records/SLEs after the record/SLE passed to it.

DSMSPSOP calls the following modules:

- DSMSPDOS (VSE environments only) — to open or close an external data set to be used for SCRIPT command processing. In OS/VS2 environments, the close and open of external data sets is accomplished within DSMSPSOP.
- DSMSPLBI — to read a DER of a DLF document from a VSAM KSDS data set (DLF directory).
- DSMSPHSH — to hash a document password.
- DSMSPILU — to read a Processor Profile Record (PPR).
- DSMSPDYN (OS/VS2 only) — to dynamically allocate external data sets, if required.

- **DSMSPGCL** — to open and/or activate a DLF ESDS cluster for input, providing access to a document on that cluster.
- **DSMSPMSG** — to issue informational and/or error messages.

**OUTPUT:** The document, external data set, or I/O module is opened and an active OFN entry created for it. This entry contains control information and pointers to a dynamic storage area acquired by DSMSPSOP. This dynamic storage area contains the communication area, a work area (for processors), a source buffer, and space for a DCB and a DER.

The external data set, if required, is dynamically allocated (VS2 only), if required, then opened.

The I/O module, if required, is loaded.

The DER of the DLF document, if required, is read into the dynamic storage area.

DSMSPMSG is called to issue messages if errors are detected during processing. "SOP" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPSOP.

### **Sequential Output Services - DSMSPTP0**

**INPUT:** Directory records are passed to DSMSPTP0 in the SPGDOBUF buffer and control intervals are passed in the SPG16K.

**DESCRIPTION:** DSMSPTP0 delivers directory records and control intervals to an external sequential data set (one record for each call). Directory records are passed to DSMSPTP0 by the caller in the SPGDOBUF. Control intervals are passed to DSMSPTP0 in the SPG16K buffer.

During ARCHIVE command processing, DSMSPARC calls DSMSPTP0 each time DSMSPARC is ready to archive a DER or the control interval of a document. It is also called by DSMSPCPO during COPY OUT command processing each time DSMSPCPO is ready to output a directory record or the control interval of a document. No other modules call DSMSPTP0.

In an OS/VS2 environment, DSMSPTP0 uses the PUT macro against SPGWQDCB (QSAM DCB) to deliver records to the external data set. In a VSE environment, DSMSPTP0 calls DSMSPDOS to output the record to the external data set.

**OUTPUT:** The record passed to DSMSPTP0 is written to the external data set.

DSMSPMSG is called to issue messages if errors are detected during processing. "TP0" will appear as the 4th through 6th characters of the message ID of messages initiated from DSMSPTP0.

### **User Exit (Skeleton Supplied) - DSMSPUXI**

**INPUT:** register 1 points to a full word address of the EXI parameter list. See "Section IV - Data Areas" on page 104 for a detailed description of information in the parameter list.

**DESCRIPTION:** The DSMSPUXI module supplied by IBM with the DLF product is a skeleton module that sets up a pointer to the user exit parameter list and return to the calling program (DSMSPEXI) with a return code of zero.

DSMSPUXI is a separate load module that is loaded by DSMSPPIN at DLF initialization. The entry point address is saved by DSMSPPIN in the SPGUXIAD field of the SPG.



If user exit processing is desired, you can replace this module with your own code.

See "External Interfaces" on page 24 for additional information on user exits.

**OUTPUT:** The supplied skeleton module returns to the calling program.

See "External Interfaces" on page 24 for additional information on writing your own user exits.

### SECTION III - MODULE DIRECTORY

The following table lists each DLF module in alphabetic order, the last three characters of modules that call the module, a short description of the module, and the last three characters of modules the module calls.

| MODULE   | CALLED BY            | DESCRIPTION   | CALLS                                       |
|----------|----------------------|---|---|
| DSMSPACT | EXC                  | ACCOUNT command processor   | LBI, MSG, SMF                               |
| DSMSPARC | EXC                  | ARCHIVE Command processor   | CMD, EXI, GCL, HSH, LSV, MSG, OLU, RDC, TPO |
| DSMSPAUT | EXC                  | AUTH Command processor  | ILU, MSG, OLU                               |
| DSMSPCLO | EXC                  | CLOSE Command processor   | MSG, SIC                                    |
| DSMSPCMD | ARC, EXC<br>RET, SCN | Command scan controller for converting a DLF command or an ARCHIVE/RETRIEVE NAMELIST entry to a set of parameters in the DLF global table | MSG, SIC                                    |
| DSMSPCPD | CPY                  | COPY DOCUMENT command processor   | GCL, ILU, LBO, MSG, OLU                     |
| DSMSPCPI | CPY                  | COPY IN command processor   | DOS, GCL, LBO, MSG, OLU                     |
| DSMSPCPL | CPY                  | COPY LIBRARY command processor  | GCL, ILU, LBO, MSG, OLU                     |
| DSMSPCPO | CPY                  | COPY OUT command processor  | GCL, ILU, MSG, TPO                          |
| DSMSPCPU | CPY                  | COPY USER command processor   | GCL, ILU, LBO, MSG, OLU                     |
| DSMSPCPY | EXC                  | COPY command processor control module   | CPD, CPI, CPL, CPO, CPU, GCL, MSG           |
| DSMSPDAC | DAS, DEF             | DEFINE/ALTER CLUSTER command processor  | GCL, LIS, MSG, OLU                          |
| DSMSPDAD | DEF                  | ALTER DOCUMENT command processor  | LIS, MSG, OLU                               |
| DSMSPDAF | DEF                  | DEFINE/ALTER PROCESS command processor  | LIS, MSG, OLU                               |
| DSMSPDAL | DEF                  | DEFINE/ALTER CLASS command processor  | LIS, MSG, OLU                               |

Figure 21 (Part 1 of 6). table of called modules

| MODULE   | CALLED BY  | DESCRIPTION  | CALLS  |
|----------|--|--|--|
| DSMSPDAM | DEF  | DEFINE/ALTER MAP command processor   | EXI, LIS, MSG, OLU   |
| DSMSPDAS | DEF  | DEFINE/ALTER SYSTEM command processor  | DAC, GCL, MSG, OLU   |
| DSMSPDAU | DEF  | DEFINE/ALTER USER command processor  | EXI, LIS, MSG, OLU   |
| DSMSPDEF | EXC  | DEFINE/ALTER command processor control module  | DAC, DAD, DAF, DAL, DAM, DAS, DAU, MSG   |
| DSMSPDOS | CPI, DSI, DSO, EXC, GNX, INT, LBI, LID, MSG, PIN, RDC, RET, SIC, SOP, TPO, TRM | VSE I/O support  | none   |
| DSMSPDSI | IMP  | Data Set Input: reads batch system data sets and converts records to required format                       | DOS, MSG, RDC  |
| DSMSPDSO | EXP  | Data Set Output: converts records to batch formats and writes s them to batch data sets                    | DOS, MSG   |
| DSMSPDYN | GNX, QNM, SIC, SOP   | Dynamic Allocation Services: allocates and unallocates data sets by data set names (Available only in MVS) | MSG  |
| DSMSPEXC | Host Operating System  | Main executor module   | ACT, ARC, AUT, CLO, CMD, CPY, DEF, DOS, EXP, GCL, IMP, INT, LIS, MSG, PAS, PIN, PRO, PUR, RED, RET, SIC, TRM |
| DSMSPEXI | ARC, DAM, DAU, EXP, IMP, PUD, PUM, PUU, RET                                    | Common exit routine for user exits   | MSG, UXI   |
| DSMSPEXP | EXC  | EXPORT command processor   | DSO, EXI, GCL, ILU, LBI, MSG, OLU  |

Figure 21 (Part 2 of 6). table of called modules

| MODULE   | CALLED BY   | DESCRIPTION   | CALLS   |
|----------|---|---|---|
| DSMSPGCL | ARC, CPD,<br>CPI, CPL,<br>CPO, CPU,<br>CPY, DAC,<br>DAS, EXC,<br>EXP, GNX,<br>IMP, LIC,<br>LBO, PIN,<br>PUC, PUD,<br>PUU, RET,<br>SOP | Cluster Services: make an ESDS cluster available for use, update the Free Space Header (FSH) in Control Interval 0 (CI 0), or close an ESDS cluster | MSG   |
| DSMSPGNX | SIC, SOP  | Get Next Record subroutine: obtains the next source record from the communications buffer or from the source document                               | DOS, DYN,<br>GCL, LBI,<br>MSG, SOP                      |
| DSMSPHSH | ARC, INT,<br>RET, SOP   | Hash Routine: hashes an 8-byte character string into a 64-bit numerical value   | None  |
| DSMSPILU | AUT, CPD,<br>CPL, CPO,<br>CPU, EXP,<br>INT, LIC,<br>LIF, LIL,<br>LIM, LIS,<br>LIU, SOP  | Access Input Library: used to read from the input document library  | None  |
| DSMSPIMP | EXC   | IMPORT command processor  | DSI, EXI,<br>GCL, LBO,<br>MSG, OLU                      |
| DSMSPINT | EXC   | Command Initialization: ensures that data sets required by the command being processed are opened   | DOS, HSH,<br>ILU, MSG,<br>QNM                           |
| DSMSPLBI | ACT, EXP,<br>GNX, LID,<br>SOP   | Library Tape Input: retrieves records from a backup tape or from the input library  | DOS   |
| DSMSPLBO | CPD, CPI,<br>CPL, CPU,<br>IMP   | Library Output: inserts or updates records in the output library  | GCL, LSV,<br>MSG  |
| DSMSPLIC | LIS   | LIST CLUSTER command processor  | ILU, GCL,<br>MSG, OLU                                   |
| DSMSPLID | LIS   | LIST DOCUMENT command processor   | DOS, LBI,<br>MSG, OLU                                   |
| DSMSPLIF | LIS   | LIST PROCESS command processor  | ILU, MSG,<br>OLU  |
| DSMSPLIL | LIS   | LIST CLASS command processor  | ILU, MSG,<br>OLU  |
| DSMSPLIM | LIS   | LIST MAP command processor  | ILU, MSG,<br>OLU  |
| DSMSPLIS | DAC, DAD,<br>DAF, DAL,<br>DAM, DAU,<br>EXC  | List command processor control module and LIST SYSTEM command processor   | ILU, LIC,<br>LID, LIF,<br>LIL, LIM,<br>LIU, MSG,<br>OLU |

Figure 21 (Part 3 of 6). table of called modules

| MODULE   | CALLED BY   | DESCRIPTION  | CALLS         |
|----------|---|--|---------------|
| DSMSPLIU | LIS   | LIST USER command processor  | ILU, MSG, OLU |
| DSMSPLSV | ARC, LBO, PUD, PUU, RET   | Validate DASD library space used: determines the amount of space used by documents stored for one library user | None          |
| DSMSPMSG | ACT, ARC, AUT, CLO, CMD, CPD, CPI, CPL, CPO, CPU, CPY, DAC, DAD, DAF, DAL, DAM, DAS, DAU, DEF, DSI, DSO, DYN, EXC, EXI, EXP, GCL, GNX, IMP, INT, LBO, LIC, LID, LIF, LIL, LIM, LIS, LIU, PAS, PIN, PRO, PUC, PUD, PUF, PUL, PUM, PUR, PUU, QNM, RED, RET, SIC, SOP, TPO | Message Writer: contains message text and issues message   | DOS           |
| DSMSPOLU | ARC, AUT, CPD, CPI, CPL, CPU, DAC, DAD, DAF, DAL, DAM, DAS, DAU, EXP, LIC, LID, LIF, LIL, LIM, LIS, LIU, PAS, IMP, PUC, PRO, PUF, PUD, PUM, PUL, RET, PUU   | Access output library: used to read, insert, or modify a record in the output document library                 | None          |
| DSMSPPAS | EXC   | PASSWORD command processor   | MSG, OLU      |
| DSMSPPIN | EXC   | Program Initializer: opens DSMLIST, SYSIN, and required VSAM data sets, and acquires main storage              | DOS, GCL, MSG |
| DSMSPPRO | EXC   | PROTECT command processor  | MSG, OLU      |
| DSMSPPUC | PUR   | PURGE CLUSTER command processor  | GCL, MSG, OLU |

Figure 21 (Part 4 of 6). table of called modules

| MODULE    | CALLED BY          | DESCRIPTION  | CALLS                                       |
|-----------|--------------------|--|---|
| DSMSPPUD  | PUR                | PURGE DOCUMENT command processor   | EXI, GCL, LSV, MSG, OLU                     |
| DSMSPPUF  | PUR                | PURGE PROCESS command processor  | MSG, OLU                                    |
| DSMSPPUL  | PUR                | PURGE CLASS command processor  | MSG, OLU                                    |
| DSMSPPUM  | PUR                | PURGE MAP command processor  | EXI, MSG, OLU                               |
| DSMSPPUR  | EXC                | PURGE command processor control module   | MSG, PUC, PUD, PUF, PUL, PUM, PUU           |
| DSMSPPUU  | PUR                | PURGE USER command processor   | EXI, GCL, LSV, MSG, OLU                     |
| DSMSPQNM  | INT                | Qualify data set name: qualifies the data set name with the library identification (Available only in MVS)                   | DYN, MSG                                    |
| DSMSPRDC  | ARC, DSI, RET      | Reads and concatenates data for IMPORT FROM(*) from SYSIN or for ARCHIVE/RETRIEVE NAMELIST from SYSIN or a data set          | DOS   |
| DSMSPRED  | EXC                | READ command processor   | MSG, SIC                                    |
| DSMSPRET  | EXC                | RETRIEVE command processor   | CMD, DOS, EXI, GCL, HSH, LSV, MSG, OLU, RDC |
| DSMSPSCN  | CMD                | Command scan to divide a command line into tokens and encode it according to the syntax graph in DSMSPSYX and DSMSPSY Y      | CMD   |
| DSMSPSIC  | CLO, EXC, RED, TRM | Source Input Control: opens and closes source documents and passes source records to the calling module                      | DOS, DYN, GNX, MSG                          |
| DSMSPSMF  | ACT                | Creates the accounting records from the UPRs passed by the ACCOUNT command   | None  |
| DSMSPSOP  | GNX                | Source Input Control Open Procedure: opens an OS data set or sets up the parameters to read a document from a source library | DOS, DYN, GCL, HSH, ILU, LBI, MSG           |
| DSMSPSPG  |                    | SPG CSECT  | None  |
| DSMSPSYX  |                    | Syntax graph for command interpretation - Part 1   | None  |
| DSMSPSY Y |                    | Syntax graph for command interpretation - Part 2   | None  |

Figure 21 (Part 5 of 6). table of called modules

| MODULE   | CALLED BY | DESCRIPTION   | CALLS    |
|----------|-----------|---|----------|
| DSMSPTPO | ARC, CPO  | Sequential Output Routine: creates a backup tape for the COPY OUT command or an archive tape for an ARCHIVE command | DOS, MSG |
| DSMSPTRM | EXC       | Command Termination: closes data sets and clears work fields used by the command                                    | DOS, SIC |
| DSMSPUXI | EXI       | User exit dummy module  | None     |

Figure 21 (Part 6 of 6). table of called modules

#### **SECTION IV - DATA AREAS**

A detailed description of each DLF data area is included in this section. You will find general information about the data areas and how they relate to each other under "Structural Overview" on page 10.



## DOCUMENT LIBRARY RECORDS

### DIRECTORY DATA SET (KSDS) RECORDS

The document library directory consists of profile and entry records containing information about the library, library users, and documents belonging to each library user. The position of a record within the directory is determined by its key (the first 30 bytes of a DLF directory record - see Figure 5 on page 12 for the layout of directory record keys).

The eight directory record types are:

- Class Entry Record (CER)
- Cluster Profile Record (CPR)
- Document Entry Record (DER)
- Library Entry Record (LER)
- Processor Profile Record (PPR)
- System Profile Record (SPR)
- User Mapping Record (UMR)
- User Profile Record (UPR).

The remainder of this section is devoted to showing the structure of each of these directory record types.

## Class Entry Record (CER)

The class entry record (CER) is a DLF directory data set record that indicates the DLF source data set (cluster) in which a specific class of documents are stored.

The DLF commands that use class entry records include:

- DEFINE CLASS
- ALTER CLASS
- PURGE CLASS
- LIST CLASS.

## CER

| OFFSETS | TYPE           | LENGTH | NAME     | DESCRIPTION             |
|---------|----------------|--------|----------|-------------------------|
| 0       | (0) STRUCTURE  | 38     | CER      |                         |
| 0       | (0) CHARACTER  | 30     | CERKEY   |                         |
| 0       | (0) CHARACTER  | 12     | CERGKEY  | GENERIC KEY             |
| 0       | (0) UNSIGNED   | 3      | CER3X00  | 3 BYTE BINARY ZEROS     |
| 3       | (3) CHARACTER  | 1      | CERCODE  | CER CODE '31'X          |
| 4       | (4) CHARACTER  | 8      | CERNAME  | CLASS NAME              |
| 12      | (C) CHARACTER  | 18     |          | UNUSED RESERVED (ZEROS) |
| 30      | (1E) CHARACTER | 8      | CERCLNAM | DEFAULT CLUSTER NAME    |

## CROSS-REFERENCE

|          |    |         |   |
|----------|----|---------|---|
| CER      | 0  | CERKEY  | 0 |
| CERCLNAM | 1E | CERNAME | 4 |
| CERCODE  | 3  | CER3X00 | 0 |
| CERGKEY  | 0  |         |   |

## Cluster Profile Record (CPR)

The cluster profile record (CPR) is a DLF directory data set record that indicates the characteristics of a particular document library source data set (VSAM ESDS).

The DLF commands that use cluster profile records include:

- ALTER CLUSTER
- DEFINE CLUSTER
- DEFINE/ALTER SYSTEM
- LIST CLUSTER
- PURGE CLUSTER.

### CPR

| OFFSETS | TYPE           | LENGTH | NAME     | DESCRIPTION             |
|---------|----------------|--------|----------|-------------------------|
| 0       | (0) STRUCTURE  | 58     | CPR      |                         |
| 0       | (0) CHARACTER  | 30     | CPRKEY   |                         |
| 0       | (0) CHARACTER  | 12     | CPRGKEY  |                         |
| 0       | (0) UNSIGNED   | 3      | CPR3X00  | 3 BYTES HEX ZERO        |
| 3       | (3) CHARACTER  | 1      | CPRCODE  | CPR CODE '10'X          |
| 4       | (4) CHARACTER  | 8      | CPRNAME  | NAME OF THE CLUSTER     |
| 12      | (C) CHARACTER  | 18     |          | UNUSED RESERVED (ZEROS) |
| 30      | (1E) CHARACTER | 8      | CPRDDNI  | INPUT CLUSTER DDNAME    |
| 38      | (26) CHARACTER | 8      | CPRDDNO  | OUTPUT CLUSTER DDNAME   |
| 46      | (2E) CHARACTER | 8      | CPRPSWD  | CLUSTER PASSWORD        |
| 54      | (36) BITSTRING | 1      | CPRFLAGS | FLAG BYTE               |
|         | 1.iii iiii     |        | CPRFORM  | CLUSTER FORMATTED FLAG  |
| 55      | (37) CHARACTER | 1      |          | UNUSED RESERVED         |
|         |                |        |          | UNUSED RESERVED         |
| 56      | (38) UNSIGNED  | 2      | CPRCISI  | CONTROL INTERVAL SIZE   |

### CROSS-REFERENCE

|          |    |         |    |    |
|----------|----|---------|----|----|
| CPR      | 0  | CPRFORM | 36 | 80 |
| CPRCISI  | 38 | CPRGKEY | 0  |    |
| CPRCODE  | 3  | CPRKEY  | 0  |    |
| CPRDDNI  | 1E | CPRNAME | 4  |    |
| CPRDDNO  | 26 | CPRPSWD | 2E |    |
| CPRFLAGS | 36 | CPR3X00 | 0  |    |

## Document Entry Record (DER)

The document entry record (DER) is a DLF directory data set record that describes a document stored under DLF. The description includes document name, password, share status, location within a source data set (ESDS), date stored, whether or not the document has versions, and to which class and cluster the document belongs.

DLF commands that use document entry records include:

- ALTER DOCUMENT
- ARCHIVE Commands
- COPY Commands
- EXPORT
- IMPORT
- LIST DOCUMENT
- PROTECT
- PURGE DOCUMENT
- PURGE USER
- RETRIEVE Commands.

### DER

| OFFSETS | TYPE           | LENGTH | NAME     | DESCRIPTION                           |
|---------|----------------|--------|----------|---------------------------------------|
| 0       | (0) STRUCTURE  | 241    | DER      |                                       |
| 0       | (0) CHARACTER  | 30     | DERKEY   | FILE LOCATOR KEY                      |
| 0       | (0) CHARACTER  | 30     | DERGKEY  | GENERIC KEY                           |
| 0       | (0) CHARACTER  | 28     | DERNVKEY | KEY LESS THE VERSION FIELD            |
| 0       | (0) CHARACTER  | 20     | DERNDKEY | KEY LESS DATATYPE & VERSION           |
| 0       | (0) UNSIGNED   | 3      | DERLIBNO | LIBRARY NO.                           |
| 3       | (3) CHARACTER  | 1      | DERCODE  | DER TYPE CODE. '70'X                  |
| 4       | (4) CHARACTER  | 16     | DERMEMNM | MEMBER NAME                           |
| 20      | (14) CHARACTER | 8      | DERDATA  | DATA TYPE                             |
| 28      | (1C) UNSIGNED  | 2      | DERVERS  | VERSION NUMBER                        |
| 30      | (1E) BITSTRING | 1      | DERTYPE  | TYPE                                  |
|         | 1... ..        |        |          | UNUSED                                |
|         | .1... ..       |        | DERPUB   | PUBLIC LIBRARY                        |
|         | ..1. ....      |        | DERPROJ  | PROJECT LIBRARY                       |
|         | ...1 ....      |        | DERPRIV  | PRIVATE LIBRARY ON CONTROLLED OFF NON |
|         | ..... 1111     |        |          | CONTROLLED                            |
| 31      | (1F) BITSTRING | 1      | DERFLAGS | UNUSED RESERVED                       |
|         | 1... ..        |        | DEROPEN  | FLAG BYTE                             |
|         | .1... ..       |        | DERPP    | LAST CI MAY NOT HAVE ZERO CH POINTER  |
|         | ..1. ....      |        | DERVP    | PASSWORD PRESENT                      |
|         | ...1 ....      |        | DERLOCK  | DOCUMENT IS VERSIONED                 |
|         | .... 1...      |        | DERSHR   | DOCUMENT IS LOCKED                    |
|         | .... .1..      |        | DERARCH  | SHARE CHARACTERISTIC                  |
|         | .... ..1.      |        | DERTODSN | DOCUMENT IS ARCHIVED                  |
|         | .... ...1      |        | DEROPER  | DOC ARCHIVED TO DYN ALLOC D.S.        |
|         |                |        |          | DERNOTE FILLED IN BY OPERATOR         |

| OFFSETS | TYPE           | LENGTH | NAME     | DESCRIPTION                 |
|---------|----------------|--------|----------|-----------------------------|
| 32      | (20) UNSIGNED  | 2      |          | UNUSED RESERVED             |
| 34      | (22) CHARACTER | 8      | DERPSWD  | FILE PASSWORD               |
| 42      | (2A) CHARACTER | 8      | DERSWTS  | SAVE/WRITE TIMESTAMP        |
| 50      | (32) CHARACTER | 8      | DERATS   | ARCHIVE TIMESTAMP           |
| 58      | (3A) CHARACTER | 8      | DESRCE   | SOURCE ATTRIBUTE            |
| 66      | (42) CHARACTER | 8      | DERCLASS | CLASS ATTRIBUTE             |
| 74      | (4A) CHARACTER | 8      | DERCLNAM | CLUSTER NAME                |
| 82      | (52) CHARACTER | 3      |          | UNUSED RESERVED             |
| 85      | (55) UNSIGNED  | 3      | DEROWNR  | OWNER NUMBER                |
| 88      | (58) UNSIGNED  | 2      | DERFLSIL | SIZE LIMIT (IN 1000'S)      |
| 90      | (5A) CHARACTER | 6      |          | UNUSED RESERVED             |
| 96      | (60) UNSIGNED  | 4      | DERCFLSI | CURRENT FILE SIZE           |
| 100     | (64) CHARACTER | 6      | DESLR    |                             |
| 100     | (64) UNSIGNED  | 4      | DERNLR   | NUMBER OF LOGICAL RECORDS   |
| 104     | (68) SIGNED    | 2      | DERLNLR  | LAST CI NO. OF LOG RECORDS  |
| 106     | (6A) UNSIGNED  | 2      | DERNCI   | NUMBER OF CONTROL INTERVALS |
| 108     | (6C) CHARACTER | 8      | DERRBAS  | RBA POINTERS                |
| 108     | (6C) UNSIGNED  | 4      | DERFRBA  | FIRST CONTROL INTERVAL RBA  |
| 112     | (70) UNSIGNED  | 4      | DERLRBA  | LAST CONTROL INTERVAL RBA   |
| 116     | (74) CHARACTER | 75     | DERAPPL  | USER EXIT INFORMATION       |
| 116     | (74) CHARACTER | 75     | DERNOTE  | ARCHIVE INFORMATION OVERLAY |
| 191     | (BF) CHARACTER | 50     | DERINFO  | DOCUMENT INFORMATION        |

#### CROSS-REFERENCE

|          |    |    |          |    |    |
|----------|----|----|----------|----|----|
| DER      | 0  |    | DERNDKEY | 0  |    |
| DERAPPL  | 74 |    | DERNLR   | 64 |    |
| DERARCH  | 1F | 04 | DERNOTE  | 74 |    |
| DERATS   | 32 |    | DERNVKEY | 0  |    |
| DERCFLSI | 60 |    | DEROPEN  | 1F | 80 |
| DERCLASS | 42 |    | DEROPER  | 1F | 01 |
| DERCLNAM | 4A |    | DEROWNR  | 55 |    |
| DERCODE  | 3  |    | DERPP    | 1F | 40 |
| DERDATA  | 14 |    | DERPRIV  | 1E | 10 |
| DERFLAGS | 1F |    | DERPROJ  | 1E | 20 |
| DERFLSIL | 58 |    | DERPSWD  | 22 |    |
| DERFRBA  | 6C |    | DERPUB   | 1E | 40 |
| DERGKEY  | 0  |    | DERRBAS  | 6C |    |
| DERINFO  | BF |    | DESHR    | 1F | 08 |
| DERKEY   | 0  |    | DESLR    | 64 |    |
| DERLIBNO | 0  |    | DESRCE   | 3A |    |
| DERLNLR  | 68 | 10 | DESWTS   | 2A |    |
| DERLOCK  | 1F |    | DERTODSN | 1F | 02 |
| DERLRBA  | 70 |    | DERTYPE  | 1E |    |
| DERMEMNM | 4  |    | DERVERS  | 1C |    |
| DERNCI   | 6A |    | DERVP    | 1F | 20 |

## Library Entry Record (LER)

The library entry record (LER) is a DLF directory data set record that indicates the public library in which a user has stored documents.

The commands that use LERs include:

- COPY Commands
- IMPORT
- LIST DOCUMENT
- PURGE DOCUMENT
- PURGE USER.

### LER

| OFFSETS | TYPE          | LENGTH | NAME     | DESCRIPTION                             |
|---------|---------------|--------|----------|---|
| 0       | (0) STRUCTURE | 32     | LER      | LIBRARY ENTRY RECORD                    |
| 0       | (0) CHARACTER | 30     | LERKEY   | RECORD KEY                              |
| 0       | (0) CHARACTER | 7      | LERGKEY  | GENERIC KEY                             |
| 0       | (0) UNSIGNED  | 3      | LEROWNR  | USERID OF OWNER OF FILE                 |
| 3       | (3) CHARACTER | 1      | LERCODE  | LER CODE = '80'X                        |
| 4       | (4) UNSIGNED  | 3      | LERLIBNO | PUBLIC LIBRARY NO.                      |
| 7       | (7) CHARACTER | 23     |          | UNUSED RESERVED (ZERO)                  |
| 30      | (1E) UNSIGNED | 2      | LERCOUNT | COUNT OF FILES IN LIBRARY FOR THIS USER |

### CROSS-REFERENCE

|          |    |          |   |
|----------|----|----------|---|
| LER      | 0  | LERKEY   | 0 |
| LERCODE  | 3  | LERLIBNO | 4 |
| LERCOUNT | 1E | LEROWNR  | 0 |
| LERGKEY  | 0  |          |   |

## Processor Profile Record (PPR)

The processor profile record (PPR) is a DLF directory data set record that specifies a special processor that is associated with a data type pair. A data type pair can be specified or implied on certain DLF commands. When a data type pair (INDATA and OUTDATA) specified in a command match the pair in a PPR, the processor specified in the PPR is called with each data record of a document while that command is processing.

The commands that use processor profile records include:

- ALTER PROCESSOR
- DEFINE PROCESSOR
- LIST PROCESSOR
- PURGE PROCESSOR
- IMPORT
- EXPORT
- SCRIPT
- READ.

### PPR

| OFFSETS | TYPE           | LENGTH | NAME     | DESCRIPTION          |
|---------|----------------|--------|----------|----------------------|
| 0       | (0) STRUCTURE  | 38     | PPR      |                      |
| 0       | (0) CHARACTER  | 30     | PPRKEY   | PROFILE RECORD KEY   |
| 0       | (0) CHARACTER  | 20     | PPRGKEY  | GENERIC KEY          |
| 0       | (0) UNSIGNED   | 3      | PPR3X00  | ZERO CODE 3X'00'     |
| 3       | (3) CHARACTER  | 1      | PPRCODE  | PPR CODE '30'X       |
| 4       | (4) CHARACTER  | 8      | PPRIDATA | INPUT DATA TYPE      |
| 12      | (C) CHARACTER  | 8      | PPRODATA | OUTPUT DATA TYPE     |
| 20      | (14) CHARACTER | 10     |          | UNUSED RESERVED      |
| 30      | (1E) CHARACTER | 8      | PPRENTY  | PROCESSOR ENTRY NAME |

### CROSS-REFERENCE

|         |    |          |   |
|---------|----|----------|---|
| PPR     | 0  | PPRIDATA | 4 |
| PPRCODE | 3  | PPRKEY   | 0 |
| PPRENTY | 1E | PPRODATA | C |
| PPRGKEY | 0  | PPR3X00  | 0 |

## System Profile Record (SPR)

The system profile record (SPR) is a DLF directory data set record containing global information that identifies a single document library. As such, there is one system profile record for each document library.

The DLF commands that use system profile records include:

- ALTER SYSTEM
- DEFINE SYSTEM.

### SPR

| OFFSETS | TYPE           | LENGTH | NAME     | DESCRIPTION              |
|---------|----------------|--------|----------|--------------------------|
| 0       | (0) STRUCTURE  | 61     | SPR      |                          |
| 0       | (0) CHARACTER  | 30     | SPRKEY   | SYSTEM PROFILE REC KEY   |
| 0       | (0) CHARACTER  | 6      | SPRGKEY  | GENERIC KEY              |
| 0       | (0) CHARACTER  | 6      | SPR6X00  | ALL ZERO FIELD           |
| 0       | (0) CHARACTER  | 3      |          | UNUSED RESERVED '00'X    |
| 3       | (3) CHARACTER  | 1      | SPRCODE  | SPR CODE '01'X           |
| 4       | (4) CHARACTER  | 2      |          | UNUSED RESERVED '00'X    |
| 6       | (6) CHARACTER  | 24     |          | UNUSED RESERVED          |
| 30      | (1E) CHARACTER | 8      | SPRDSTS  | DEFINE SYSTEM TIME STAMP |
| 38      | (26) CHARACTER | 12     | SPRLIBN  | LIBRARY NAME             |
| 50      | (32) UNSIGNED  | 1      | SPRACFCL | RACF CLASS NAME LENGTH   |
| 51      | (33) CHARACTER | 8      | SPRACFCN | RACF CLASS NAME          |
| 59      | (3B) UNSIGNED  | 1      | SPROPRTC | OPERATOR ROUTING CODE    |
| 60      | (3C) BITSTRING | 1      | SPRFLAG  | FLAG BYTE                |
|         | 1... ..        |        | SPRACFL  | RACF IS PRESENT FLAG     |
|         | .1... ..       |        | SPRENQF  | ENQUE REQUIRED FLAG      |
|         | ..11 1111      |        |          | UNUSED RESERVED          |

### CROSS-REFERENCE

|          |    |          |    |
|----------|----|----------|----|
| SPR      | 0  | SPRFLAG  | 3C |
| SPRACFCL | 32 | SPRGKEY  | 0  |
| SPRACFCN | 33 | SPRKEY   | 0  |
| SPRACFL  | 3C | SPRLIBN  | 26 |
| SPRCODE  | 3  | SPROPRTC | 3B |
| SPRDSTS  | 1E | SPR6X00  | 0  |
| SPRENQF  | 3C |          |    |



## User Mapping Record (UMR)

The user mapping record (UMR) is a DLF directory data set record that describes how an external user, such as an ATMS-III user, is known to DLF. The record identifies the user's external system, the user's identification on that system, and the library user number assigned to that user in DLF.

The DLF commands that use user mapping records include:

- ALTER MAP
- DEFINE MAP
- LIST MAP
- PURGE MAP.

### UMR

| OFFSETS | TYPE | LENGTH | NAME | DESCRIPTION |
|---------|------|--------|------|-------------|
|---------|------|--------|------|-------------|

|    |      |           |    |          |                         |
|----|------|-----------|----|----------|-------------------------|
| 0  | (0)  | STRUCTURE | 33 | UMR      |                         |
| 0  | (0)  | CHARACTER | 30 | UMRKEY   | USER MAPPING RECORD KEY |
| 0  | (0)  | CHARACTER | 16 | UMRXKEY  | DEF/PUR/ALT GENERIC KEY |
| 0  | (0)  | CHARACTER | 16 | UMRGKEY  | GENERIC KEY             |
| 0  | (0)  | CHARACTER | 3  | UMR3XFF  | THREE BYTES 'FF'X       |
| 3  | (3)  | CHARACTER | 1  | UMRCODE  | UMR CODE. 'E0'X         |
| 4  | (4)  | CHARACTER | 2  | UMRHOST  | HOST IDENTIFIER         |
| 6  | (6)  | CHARACTER | 10 | UMREXID  | EXTERNAL USER ID        |
| 16 | (10) | CHARACTER | 14 |          | UNUSED RESERVED         |
| 30 | (1E) | UNSIGNED  | 3  | UMRLIBNO | LIBRARY NUMBER          |

### CROSS-REFERENCE

|         |   |          |    |
|---------|---|----------|----|
| UMR     | 0 | UMRKEY   | 0  |
| UMRCODE | 3 | UMRLIBNO | 1E |
| UMREXID | 6 | UMRXKEY  | 0  |
| UMRGKEY | 0 | UMR3XFF  | 0  |
| UMRHOST | 4 |          |    |

## User Profile Record (UPR)

The user profile record (UPR) is a DLF directory data set record that describes the privileges, library type, space allocation, and project library of a particular DLF user. It is identified by a user number, generally referred to as the library number or libno.

The DLF commands that use user profile records include:

- AUTH
- ALTER USER
- COPY Commands
- DEFINE USER
- LIST USER
- PASSWORD
- PURGE USER.

## UPR

| OFFSETS | TYPE          | LENGTH | NAME     | DESCRIPTION                  |
|---------|---------------|--------|----------|------------------------------|
| 0       | (0) STRUCTURE | 108    | UPR      |                              |
| 0       | (0) CHARACTER | 72     | UPRMODEL | UPR FIELDS WHICH DEFINE USER |
| 0       | (0) CHARACTER | 30     | UPRKEY   | USER PROFILE RECORD KEY      |
| 0       | (0) CHARACTER | 4      | UPRGKEY  | GENERIC KEY                  |
| 0       | (0) UNSIGNED  | 3      | UPRLIBNO |                              |

### LIBRARY NUMBER

|    |                |    |          |                         |
|----|----------------|----|----------|-------------------------|
| 3  | (3) CHARACTER  | 1  | UPRCODE  | PROFILE CODE = '40'X    |
| 4  | (4) CHARACTER  | 26 |          | UNUSED RESERVED (ZERO)  |
| 30 | (1E) CHARACTER | 8  | UPRPSWD  | USER PASSWORD           |
| 38 | (26) BITSTRING | 1  | UPRTYPE  | TYPE                    |
|    | 1111 .....     |    | UPRLTYPE | LIBRARY TYPE FLAGS      |
|    | 1... .....     |    | UPRPUB   | PUBLIC LIBRARY          |
|    | .1... .....    |    | UPRPROJ  | PROJECT LIBRARY         |
|    | ..1. ....      |    | UPRPRIV  | PRIVATE LIBRARY         |
|    | ...1 .....     |    | UPRPROJC | CONTROLLED LIBRARY FLAG |

ON IS CONTROLLED, OFF UNCONTROLLED

| OFFSETS | TYPE                        | LENGTH | NAME  | DESCRIPTION  |
|---------|-----------------------------|--------|---|--|
| 39      | .... 1111<br>(27) BITSTRING | 1      | UPRSTYPE<br>UPRFLAGS<br>UPRNLOG<br>UPRPP<br>UPRLOCK<br>UPRSYSAD | RESERVED<br>FLAG BYTE<br>USER NOT LOGGED OFF<br>PASSWORD PRESENT<br>USER LOCKED FROM SYSTEM<br>SYSTEM ADMINISTRATOR<br>UNUSED RESERVED |
| 40      | (28) CHARACTER              | 8      | UPRDATA   | DEFAULT DATA ATTRIBUTE   |
| 48      | (30) CHARACTER              | 1      |   | UNUSED RESERVED  |
| 49      | (31) UNSIGNED               | 3      | UPRPJLIB  | PROJECT LIBRARY NUMBER   |
| 52      | (34) CHARACTER              | 4      |   | UNUSED RESERVED  |
| 56      | (38) CHARACTER              | 8      | UPRCLASS  | DEFAULT CLASS ATTRIBUTE  |
| 64      | (40) CHARACTER              | 8      | UPRSOURC  | DEFAULT SOURCE ATTRIBUTE   |
| 72      | (48) UNSIGNED               | 4      | UPRDPCNT  | USERS DOCUMENT COUNT   |
| 76      | (4C) CHARACTER              | 12     | UPRACCNT  | UPR FIELDS FOR USER ACCOUNTING   |
| 76      | (4C) SIGNED                 | 4      | UPRDSLIM  | DASD SPACE LIMITATION (1000'S)   |
| 80      | (50) UNSIGNED               | 4      | UPRDSLIB  | DASD SPACE IN CURRENT LIBRARY  |
| 84      | (54) UNSIGNED               | 4      | UPRDPLIB  | DASD SPACE IN PROJECT/PUBLIC LIBRARIES   |
| 88      | (58) CHARACTER              | 20     | UPRAPPL   | HOST EXIT INFORMATION  |
| 108     | (6C) CHARACTER              | 0      | UPREND  | END OF UPR   |

#### CROSS-REFERENCE

|          |    |          |    |    |
|----------|----|----------|----|----|
| UPR      | 0  | UPRLOCK  | 27 | 20 |
| UPRACCNT | 4C | UPRLTYPE | 26 | F0 |
| UPRAPPL  | 58 | UPRMODEL | 0  |    |
| UPRCLASS | 38 | UPRNLOG  | 27 | 80 |
| UPRCODE  | 3  | UPRPJLIB | 31 |    |
| UPRDATA  | 28 | UPRPP    | 27 | 40 |
| UPRDPCNT | 48 | UPRPRIV  | 26 | 20 |
| UPRDPLIB | 54 | UPRPROJ  | 26 | 40 |
| UPRDSLIB | 50 | UPRPROJC | 26 | 10 |
| UPRDSLIM | 4C | UPRPSWD  | 1E |    |
| UPREND   | 6C | UPRPUB   | 26 | 80 |
| UPRFLAGS | 27 | UPRSOURC | 40 |    |
| UPRGKEY  | 0  | UPRSTYPE | 26 | 0F |
| UPRKEY   | 0  | UPRSYSAD | 27 | 10 |
| UPRLIBNO | 0  | UPRTYPE  | 26 |    |

## **SOURCE DATA SET (ESDS) RECORDS**

The library source data set records are used to store documents and control information. The five types of source data set records used by DLF include:

- Control Intervals (CI)
- Control Interval Trailers (CIT)
- Free Space Headers (FSH)
- Free Space Records (FSR)
- Source Line Entries (SLE).

The remainder of this section is devoted to illustrating the structure of library source data set records. See "Document Library Structure" on page 10 for more information about these data sets.

## Control Interval Trailer (CIT)

The control interval trailer (CIT) is a set of control fields at the end of each control interval (other than the first) of each DLF library ESDS cluster. The first control interval of each library ESDS cluster contains a free space header (FSH) rather than a CIT. Each control interval (other than the first) of a cluster being defined (DEFINE CLUSTER command) to DLF is formatted with a skeleton CIT. Also, each new control interval of a cluster being altered (ALTER CLUSTER) after space has been added to it is formatted with a skeleton CIT. This is done by the DEFINE/ALTER CLUSTER module, DSMSPDAC. The skeleton CIT contains only those fields required by VSAM at the end of each control interval - the last seven bytes of the control interval trailer (starting with label CITVSAM).

DLF uses the control interval trailer to chain sets of control intervals together. For example, all control intervals containing a particular document's data are chained together - with the chain head in the directory entry record (DER) for that document. Also, all control intervals that were used for document storage but are now free (through a purge or archive operation), are chained together with the chain head in the free space header (FSH) for the cluster.

DLF modules that modify the control interval trailer of a control interval are, generally, modules that acquire control intervals from the unused or free chain for document storage, and those that return control intervals to the free chain after a document has been archived or purged.

Modules that use rather than modify the control interval trailer of a control interval are, generally, those that must sequentially retrieve the data of a document (for example, EXPORT or SCRIPT) without modifying the document in the DLF library.

## CIT

| OFFSETS | TYPE           | LENGTH | NAME     | DESCRIPTION                              |
|---------|----------------|--------|----------|--|
| 0       | (0) STRUCTURE  | 48     | CIT      |  |
| 0       | (0) UNSIGNED   | 3      | CITLIBNO | LIBRARY NUMBER                           |
| 3       | (3) CHARACTER  | 1      |          | UNUSED RESERVED                          |
| 4       | (4) CHARACTER  | 16     | CITMBRA  | MEMBER NAME                              |
| 20      | (14) CHARACTER | 8      | CITDATA  | CONTENT ATTRIBUTE                        |
| 28      | (1C) SIGNED    | 2      | CITVERS  | VERSION NUMBER (COMPLEMENT)              |
| 30      | (1E) SIGNED    | 2      | CITNLR   | NUMBER OF LOGICAL RECORDS                |
| 32      | (20) UNSIGNED  | 4      | CITNRBA  | NEXT CONTROL INTERVAL RBA                |
| 32      | (20) ADDRESS   | 4      | CITNEXT  | NEXT 4K BLOCK ADDRESS                    |
| 36      | (24) SIGNED    | 2      | CITDLEN  | DATA LENGTH                              |
| 36      | (24) ADDRESS   | 2      | CITCLDSP | DISP IN BLOCK TO NEXT LINE               |
| 38      | (26) CHARACTER | 3      |          | RESERVED                                 |
| 41      | (29) CHARACTER | 7      | CITVSAM  | VSAM CONTROL INFORMATION                 |
| 41      | (29) UNSIGNED  | 3      | CITVRDF  | VSAM RDF                                 |
| 41      | (29) BITSTRING | 1      | CITRDF   | VSAM RDF FLAG BYTE                       |
| 42      | (2A) SIGNED    | 2      | CITRECL  | VSAM RECORD LENGTH                       |
| 44      | (2C) UNSIGNED  | 4      | CITVCIDF | VSAM CIDF                                |
| 44      | (2C) ADDRESS   | 2      | CITCIFSD | CONTROL INTERVAL FREE SPACE DISPLACEMENT |
| 46      | (2E) SIGNED    | 2      | CITCIFSL | CONTROL INTERVAL FREE SPACE LENGTH       |

**CROSS-REFERENCE**

|          |    |
|----------|----|
| CIT      | 0  |
| CITCIFSD | 2C |
| CITCIFSL | 2E |
| CITCLDSP | 24 |
| CITDATA  | 14 |
| CITDLEN  | 24 |
| CITLIBNO | 0  |
| CITMBRA  | 4  |
| CITNEXT  | 20 |

|          |    |
|----------|----|
| CITNLR   | 1E |
| CITNRBA  | 20 |
| CITRDFF  | 29 |
| CITRECL  | 2A |
| CITVCIDF | 2C |
| CITVERS  | 1C |
| CITVRDF  | 29 |
| CITVSAM  | 29 |

## Free Space Header (FSH)

A free space header (FSH) is created by the DLF module DSMSPDAC for each DLF library ESDS cluster at the time that cluster is formatted. The free space header is placed at the end of the first control interval of the cluster. The free space header and the free space record occupy the same control interval and are created at the same time.

The free space header contains information about the location and amount of unused and free (previously used for document storage and returned to the free chain) control intervals within the cluster. It must be updated each time unused or free control intervals are allocated for document storage and each time control intervals previously used to store documents are returned to the free chain.

The free space header of a cluster is read into the CLB whenever that cluster is required for processing a DLF command. It is updated as command processing proceeds and is written back out as a part of the first control interval of the cluster when command processing is complete.

## FSH

| OFFSETS | TYPE                      | LENGTH | NAME                 | DESCRIPTION  |
|---------|---------------------------|--------|----------------------|--|
| 0       | (0) STRUCTURE             | 32     | FSH                  | FREE SPACE HEADER  |
| 0       | (0) CHARACTER             | 8      | FSHXCDS              |  |
| 0       | (0) CHARACTER             | 4      | FSHXL RBA            |  |
| 0       | (0) CHARACTER             | 1      | FSHXID               | CLUSTER IDENTIFIER   |
| 1       | (1) CHARACTER             | 3      |                      | UNUSED RESERVED  |
| 4       | (4) UNSIGNED              | 4      | FSHXRBA              | NEXT UNUSED RBA AT END OF DATA SET   |
| 8       | (8) UNSIGNED              | 4      | FSHHRBA              | HIGHEST RBA+1 IN DATA SET  |
| 12      | (C) UNSIGNED              | 4      | FSHFRBA              | FIRST FREE CONTROL INTERVAL  |
| 16      | (10) CHARACTER            | 8      | FSHLCDS              | FREE SPACE PTRS  |
| 16      | (10) UNSIGNED             | 4      | FSHLRBA              | LAST FREE CONTROL INTERVAL   |
| 20      | (14) UNSIGNED             | 4      | FSHNCI               | NUMBER OF FREE CONTROL INTERVALS   |
| 24      | (18) BITSTRING<br>1... .. | 1      | FSHFLAGS<br>FSHINVLD | FREE SPACE HEADER FLAGS<br>INDICATES FSHLRBA AND FSHNCI MAY NOT BE<br>VALID SET AT SYSTEM INITIALIZATION AND<br>RESET SYSTEM TERMINATION |
|         | .111 1111                 |        |                      | RESERVED   |
| 25      | (19) CHARACTER            | 7      | FSHVSAM              | VSAM INDICATORS  |
| 25      | (19) UNSIGNED             | 3      | FSHVRDF              | VSAM RDF   |
| 25      | (19) BITSTRING            | 1      | FSHRDFF              | VSAM RDF FLAG BYTE   |
| 26      | (1A) UNSIGNED             | 2      | FSHRECL              | VSAM RECORD LENGTH   |
| 28      | (1C) UNSIGNED             | 4      | FSHVCIDF             | VSAM CIDF  |
| 28      | (1C) ADDRESS              | 2      | FSHCIFSD             | C. I. FREE SPACE DISPLACEMENT  |
| 30      | (1E) UNSIGNED             | 2      | FSHCIFSL             | C. I. FREE SPACE LENGTH  |

**CROSS-REFERENCE**

|          |    |    |          |    |
|----------|----|----|----------|----|
| FSH      | 0  |    | FSHRDFF  | 19 |
| FSHCIFSD | 1C |    | FSHRECL  | 1A |
| FSHCIFSL | 1E |    | FSHVCIDF | 1C |
| FSHFLAGS | 18 |    | FSHVRDF  | 19 |
| FSHFRBA  | C  |    | FSHVSAM  | 19 |
| FSHRBA   | 8  |    | FSHXCDS  | 0  |
| FSHINVLD | 18 | 80 | FSHXID   | 0  |
| FSHLCDS  | 10 |    | FSHXLRBA | 0  |
| FSHLRBA  | 10 |    | FSHXRBA  | 4  |
| FSHNCI   | 14 |    |          |    |



### Free Space Record (FSR)

A free space record (FSR) is created by the DLF module DSMSPDAC for each library ESDS cluster at the time that cluster is formatted. The FSR is placed at the beginning of the first control interval of the cluster.

The free space header contains only one field: a timestamp indicating the date and time the cluster was defined or altered.

**Note:** The free space header and the free space record are created at the same time.

#### FSR

| <u>OFFSETS</u> | <u>TYPE</u> | <u>LENGTH</u> | <u>NAME</u> | <u>DESCRIPTION</u> |
|----------------|-------------|---------------|-------------|--------------------|
|----------------|-------------|---------------|-------------|--------------------|

|   |               |   |          |                          |
|---|---------------|---|----------|--------------------------|
| 0 | (0) STRUCTURE | 8 | FSR      | FREE SPACE HEADER RECORD |
| 0 | (0) CHARACTER | 8 | FSRTSTMP | DEFINE SYSTEM TIMESTAMP  |

#### CROSS-REFERENCE

|     |   |          |   |
|-----|---|----------|---|
| FSR | 0 | FSRTSTMP | 0 |
|-----|---|----------|---|

## Source Line Entry (SLE)

DLF stores documents in the control intervals of the library source clusters in the form of one or more source line entry (SLE) records (basically just a line of text preceded by a two-byte length field). A source line entry cannot span control intervals and is therefore limited in length to control interval size minus the length of the control interval trailer (CIT).

If the high order bit of the source line entry length field (first two bytes of the source line entry) is set to "1", this source line entry is the last source line entry in the control interval.

### SLE

| OFFSETS | TYPE          | LENGTH | NAME     | DESCRIPTION                  |
|---------|---------------|--------|----------|------------------------------|
| 0       | (0) STRUCTURE | 2      | SLE      |                              |
| 0       | (0) UNSIGNED  | 2      | SLELENG  | LOW-ORDER 15 BITS: TEXT LENG |
| 0       | (0) BITSTRING | 1      | SLEFLAGS | FLAG BIT IN SLELENG          |
|         | 1... ..       |        | SLEE0B   | END OF BLOCK INDICATOR       |
|         | .111 1111     |        |          | NEVER TO BE USED BY ANYONE   |
| 2       | (2) CHARACTER | 0      | SLETEXT  | TEXT OF SOURCE LINE          |

### CROSS-REFERENCE

|          |   |    |         |   |
|----------|---|----|---------|---|
| SLE      | 0 |    | SLELENG | 0 |
| SLEE0B   | 0 | 80 | SLETEXT | 2 |
| SLEFLAGS | 0 |    |         |   |

## SYSTEM CONTROL BLOCKS

The principal control blocks used for DLF processing are:

- Cluster Control Block (CLB)
- List Processor Command Work Area (LST)
- Open File Name Table (OFN)
- Standard Register Save Area (SAV)
- Library Global Table (SPG).

The largest and most often used of these control blocks is the library global table (SPG). The SPG is an area within the library program that contains frequently used information such as data areas, flags, control blocks, buffers, pointers to loaded programs, and pointers to dynamic areas.

In addition to the SPG, there are control blocks that are declared in the individual modules and make up a part of the internal interface between two DLF modules. Others are dynamically acquired areas that exist for a longer period of time (some for the entire DLF session) and are pointed to by fields in the SPG.

The section that follows is devoted to illustrating the structure of each of these control blocks.

## Cluster Control Block (CLB)

A cluster control block is created at processing time for each ESDS cluster that is open for input or output. Each cluster control block is contained in a separate storage area. Input cluster control blocks are chained from the SPG field SPGCLIFP and output cluster control blocks are chained from the SPG field, SPGCLOFP.

Each cluster control block contains information (i.e., cluster name, flags, ACB, and free space header) required to control an ESDS library cluster. The VSAM RPLs used to access these clusters are located in the SPG.

### CLB

| OFFSETS | TYPE           | LENGTH | NAME     | DESCRIPTION                |
|---------|----------------|--------|----------|----------------------------|
| 0       | (0) STRUCTURE  | 224    | CLB      |                            |
| 0       | (0) CHARACTER  | 8      | CLBNAME  | CLUSTER NAME               |
| 8       | (8) ADDRESS    | 4      | CLBFWD   | FORWARD CHAIN POINTER      |
| 12      | (C) ADDRESS    | 4      |          | UNUSED RESERVED            |
| 16      | (10) BITSTRING | 1      | CLBFLAG  | CLUSTER FLAG FIELD         |
|         | 1... ..        |        | CLBINDS  | CLUSTER IS OPEN FOR INPUT  |
|         | .1... ..       |        | CLBOUTDS | CLUSTER IS OPEN FOR OUTPUT |
|         | ..1... ..      |        | CLBUNFMT | UNFORMATTED DATA SET       |
|         | ...1 1111      |        |          | UNUSED RESERVED            |
| 17      | (11) CHARACTER | 1      |          | UNUSED RESERVED            |
| 18      | (12) UNSIGNED  | 2      | CLBCIS   | CONTROL INTERVAL SIZE      |
| 20      | (14) CHARACTER | 4      |          | UNUSED RESERVED            |
| 24      | (18) CHARACTER | 8      | CLBSTAMP | LIBRARY TIME STAMP         |
| 32      | (20) CHARACTER | 80     | CLBIACB  | INPUT ACCESS CONTROL BLK   |
| 112     | (70) CHARACTER | 80     | CLBOACB  | OUTPUT ACCESS CONTROL BLK  |
| 192     | (C0) CHARACTER | 32     | CLBFSH   | FREE SPACE HEADER          |

### CROSS-REFERENCE

|         |    |          |    |    |
|---------|----|----------|----|----|
| CLB     | 0  | CLBINDS  | 10 | 80 |
| CLBCIS  | 12 | CLBNAME  | 0  |    |
| CLBFLAG | 10 | CLBOACB  | 70 |    |
| CLBFSH  | C0 | CLBOUTDS | 10 | 40 |
| CLBFWD  | 8  | CLBSTAMP | 18 |    |
| CLBIACB | 20 | CLBUNFMT | 10 | 20 |

## List Processor Command Work Area (LST)

The List Processor Common Work Area is used by all of the list processor modules of DLF.

This area is used by list processor modules as a work area and to retain information that must be saved for the duration of list processing. This information includes the type of list being performed, the names and characteristics of the entities to be listed, and the status of list processing.

| This work area is always stored in the SPG at location SPG16K plus 4K.

### LST

| OFFSETS | TYPE           | LENGTH | NAME     | DESCRIPTION              |
|---------|----------------|--------|----------|--------------------------|
| 0       | (0) STRUCTURE  | 646    | LST      | WORK AREA IN 16K BUFFER  |
| 0       | (0) CHARACTER  | 8      | LSTPCK   | CONVERSION PACK DEC      |
| 8       | (8) CHARACTER  | 8      | LSTUP    | CONVERSION UNPACK AREA   |
| 16      | (10) CHARACTER | 16     | LSTSTMM  | BEGINNING MEMBER NAME    |
| 32      | (20) CHARACTER | 16     | LSTENDM  | ENDING MEMBER NAME       |
| 48      | (30) CHARACTER | 16     | LSTPRDOC | PREVIOUS DOCUMENT NAME   |
| 64      | (40) CHARACTER | 8      | LSTSTDAT | BEGINNING DATE           |
| 64      | (40) SIGNED    | 4      | LSTSHDFP | HI ORDER = '4ENNNNNN'X   |
| 68      | (44) SIGNED    | 4      | LSTSLDFP | LOW ORDER = MICRO SECS   |
| 72      | (48) CHARACTER | 8      | LSTENDAT | ENDING DATE              |
| 72      | (48) SIGNED    | 4      | LSTEHDFP | HIORDER = '4ENNNNNN'X    |
| 76      | (4C) SIGNED    | 4      | LSTELDFP | LOW ORDER = MICROSECS    |
| 80      | (50) CHARACTER | 8      | LSTDATCA | DATA CONTENT ATTRIBUTE   |
| 88      | (58) CHARACTER | 8      | LSTCLSCA | CLASS CONTENT ATTRIBUTE  |
| 96      | (60) CHARACTER | 8      | LSTSRCCA | SOURCE CONTENT ATTRIBUTE |
| 104     | (68) SIGNED    | 4      | LSTRETC  | RETURN CODE              |
| 108     | (6C) SIGNED    | 4      | LSTRSNC  | REASON CODE              |
| 112     | (70) SIGNED    | 4      | LSTSTLB  | BEGINNING LIBRARY NUM    |
| 116     | (74) SIGNED    | 4      | LSTNDLB  | ENDING LIBRARY NUM       |
| 120     | (78) SIGNED    | 4      | LSTRCT   | COUNT OF RECORDS PROC    |
| 124     | (7C) SIGNED    | 2      | LSTLC    | COUNT OF LINES PRINTED   |
| 126     | (7E) SIGNED    | 2      | LSTPC    | COUNT OF PAGES PRINTED   |
| 128     | (80) ADDRESS   | 4      | LSTPHDP1 | PRINT HEADER 1 POINTER   |
| 132     | (84) ADDRESS   | 4      | LSTPHDP2 | PRINT HEADER 2 POINTER   |

| OFFSETS | TYPE             | LENGTH | NAME     | DESCRIPTION              |
|---------|------------------|--------|----------|--------------------------|
| 136     | (88) SIGNED      | 2      | LSTPHDL1 | PRINT HEADER 1 LENGTH    |
| 138     | (8A) SIGNED      | 2      | LSTPHDL2 | PRINT HEADER 2 LENGTH    |
| 140     | (8C) CHARACTER   | 14     | LSTSKEY  | LIBRARY KEY SAVE AREA    |
| 154     | (9A) UNSIGNED    | 2      | LSTCTLFL | CONTROL FLAG FIELD       |
|         | 1... ..          |        | LSTPROC  | PROCESS KEYWORD SPEC     |
|         | .1... ..         |        | LSTUSER  | USER SPECIFIED           |
|         | ..1... ..        |        | LSTLIB   | LIBRARIES SPECIFIED      |
|         | ...1... ..       |        | LSTALL   | ALL KEYWORD SPECIFIED    |
|         | ....1... ..      |        | LSTOWN   | OWN KEYWORD SPEC/DEFAULT |
|         | .... .1... ..    |        | LSTPLIB  | PROJECT LIB SPECIFIED    |
|         | .... ..1... ..   |        | LSTLER   | LER BEING PROCESSED      |
|         | .... ....1... .. |        | LSTEND   | PROCESSING CONTROL FLG   |
|         | 1... ..          |        | LSTCLASS | CLASS SPECIFIED          |
|         | .1... ..         |        | LSTDATA  | DATA SPECIFIED           |
|         | ..1... ..        |        | LSTSOURC | SOURCE SPECIFIED         |
|         | ....1... ..      |        | LSTLOCK  | LOCK SPECIFIED           |
|         | .... .1... ..    |        | LSTARCHV | ARCHIVE SPECIFIED        |
|         | .... ..1... ..   |        | LSTVERSN | VERSION SPECIFIED        |
|         | .... ...1... ..  |        | LSTINFO  | INFO SPECIFIED           |
| 156     | (9C) CHARACTER   | 130    | LSTPRB   | SPMSG PRINT BUFFER       |
| 156     | (9C) CHARACTER   | 5      | LSTHDR   | SPMSG PRINT CONTROLS     |
| 156     | (9C) SIGNED      | 2      | LSTLEN   | SPMSG TEXT LENGTH        |
| 158     | (9E) CHARACTER   | 2      | LSTCODE  | SPMSG MESSAGE CODE       |
| 160     | (A0) CHARACTER   | 1      | LSTCCC   | SPMSG CHAR CONTROL CHAR  |
| 161     | (A1) CHARACTER   | 125    | LSTTXT   | SPMSG PRINT TEXT         |
| 286     | (11E) CHARACTER  | 130    | LSTHD1B  | HEADER LINE 1 BUFFER     |
| 286     | (11E) CHARACTER  | 5      | LSTHD1A  | DSMPMSG PARAMETERS       |
| 286     | (11E) SIGNED     | 2      | LSTHD1L  | LINE LENGTH              |
| 288     | (120) CHARACTER  | 2      | LSTHD1C  |                          |
| 290     | (122) CHARACTER  | 1      | LSTHD1CC | CARRIAGE CONTROL CHAR    |
| 291     | (123) CHARACTER  | 125    | LSTHD1T  | HEADER TEXT AREA         |
| 416     | (1A0) CHARACTER  | 130    | LSTHD2B  | HEADER LINE 2 BUFFER     |
| 416     | (1A0) CHARACTER  | 5      | LSTHD2A  | DSMPMSG PARAMETERS       |
| 416     | (1A0) SIGNED     | 2      | LSTHD2L  | LINE LENGTH              |
| 418     | (1A2) CHARACTER  | 2      | LSTHD2C  |                          |
| 420     | (1A4) CHARACTER  | 1      | LSTHD2CC | CARRIAGE CONTROL CHAR    |
| 421     | (1A5) CHARACTER  | 125    | LSTHD2T  | HEADER TEXT AREA         |
| 546     | (222) CHARACTER  | 100    | LSTWORK  | LIST WORK AREA           |
| 646     | (286) CHARACTER  | 0      |          | END OF DSMLST            |

#### CROSS-REFERENCE

|          |     |    |          |     |    |
|----------|-----|----|----------|-----|----|
| LST      | 0   |    | LSTHD1T  | 123 |    |
| LSTALL   | 9A  | 10 | LSTHD2A  | 1A0 |    |
| LSTARCHV | 9B  | 08 | LSTHD2B  | 1A0 |    |
| LSTCCC   | A0  |    | LSTHD2C  | 1A2 |    |
| LSTCLASS | 9B  | 80 | LSTHD2CC | 1A4 |    |
| LSTCLSCA | 58  |    | LSTHD2L  | 1A0 |    |
| LSTCODE  | 9E  |    | LSTHD2T  | 1A5 |    |
| LSTCTLFL | 9A  |    | LSTINFO  | 9B  | 02 |
| LSTDATA  | 9B  | 40 | LSTLC    | 7C  |    |
| LSTDATCA | 50  |    | LSTLEN   | 9C  |    |
| LSTEHDFF | 48  |    | LSTLER   | 9A  | 02 |
| LSTELDFP | 4C  |    | LSTLIB   | 9A  | 20 |
| LSTEND   | 9A  | 01 | LSTLOCK  | 9B  | 10 |
| LSTENDAT | 48  |    | LSTNDLB  | 74  |    |
| LSTENDM  | 20  |    | LSTOWN   | 9A  | 08 |
| LSTHDR   | 9C  |    | LSTPC    | 7E  |    |
| LSTHD1A  | 11E |    | LSTPCK   | 0   |    |
| LSTHD1B  | 11E |    | LSTPHDL1 | 88  |    |
| LSTHD1C  | 120 |    | LSTPHDL2 | 8A  |    |
| LSTHD1CC | 122 |    | LSTPHDP1 | 80  |    |
| LSTHD1L  | 11E |    | LSTPHDP2 | 84  |    |

|          |    |    |          |     |    |
|----------|----|----|----------|-----|----|
| LSTPLIB  | 9A | 04 | LSTSOURC | 9B  | 20 |
| LSTPRB   | 9C |    | LSTSRCCA | 60  |    |
| LSTPRDOC | 30 |    | LSTSTDAT | 40  |    |
| LSTPROC  | 9A | 80 | LSTSTLB  | 70  |    |
| LSTRCT   | 78 |    | LSTSTMM  | 10  |    |
| LSTRETC  | 68 |    | LSTTXT   | A1  |    |
| LSTRSNC  | 6C |    | LSTUP    | 8   |    |
| LSTSHDFP | 40 |    | LSTUSER  | 9A  | 40 |
| LSTSKEY  | 8C |    | LSTVERSN | 9B  | 04 |
| LSTSLDFP | 44 |    | LSTWORK  | 222 |    |

## Open File Name Table (OFN)

The Open File Name (OFN) table contains information about each file that is open for input during a READ or SCRIPT operation. In this case, a file can be either a DLF document or an external data set.

**Note:** The fields from OFN part 1 (ending at offset 132) through the end of the structure are repeated for each entry (16 times).

Dynamic GETMAIN/GETVIS space is acquired for the OFN by DSMSPINT (command initialization module) the first time a SCRIPT or READ command is to be processed. A pointer to this dynamic area is placed in the SPGSICSP field of the SPG. The header part of the OFN table is cleared and the characters "NT" are placed in the OFNID field (beginning the OFN table).

### OFNTABLE

| OFFSETS | TYPE          | LENGTH | NAME     | DESCRIPTION            |
|---------|---------------|--------|----------|------------------------|
| 0       | (0) STRUCTURE | 4420   | OFNTABLE | OPEN FILE NAME TABLE   |
| 0       | (0) CHARACTER | 132    | OFNHDR   | HEADER GROUP           |
| 0       | (0) CHARACTER | 2      | OFNID    | TABLE ID = NT          |
| 2       | (2) SIGNED    | 2      | OFNMAX   | NUMBER OF ENTRIES      |
| 4       | (4) ADDRESS   | 4      | OFNBLK1  | 1ND BLOCK OF DYN STORE |
| 8       | (8) ADDRESS   | 4      |          | LENGTH OF DYN STORAGE  |
| 12      | (C) ADDRESS   | 4      | OFNBLK2  | 2ND BLOCK              |
| 16      | (10) ADDRESS  | 4      |          |                        |
| 20      | (14) ADDRESS  | 4      | OFNBLK3  | 3RD BLOCK              |
| 24      | (18) ADDRESS  | 4      |          |                        |
| 28      | (1C) ADDRESS  | 4      | OFNBLK4  | 4TH BLOCK              |
| 32      | (20) ADDRESS  | 4      |          |                        |
| 36      | (24) ADDRESS  | 4      | OFNBLK5  | 5TH BLOCK              |
| 40      | (28) ADDRESS  | 4      |          |                        |
| 44      | (2C) ADDRESS  | 4      | OFNBLK6  | 6TH BLOCK              |
| 48      | (30) ADDRESS  | 4      |          |                        |
| 52      | (34) ADDRESS  | 4      | OFNBLK7  | 7TH BLOCK              |
| 56      | (38) ADDRESS  | 4      |          |                        |
| 60      | (3C) ADDRESS  | 4      | OFNBLK8  | 8TH BLOCK              |
| 64      | (40) ADDRESS  | 4      |          |                        |
| 68      | (44) ADDRESS  | 4      | OFNBLK9  | 9TH BLOCK              |
| 72      | (48) ADDRESS  | 4      |          |                        |



| OFFSETS |           | TYPE      | LENGTH | NAME     | DESCRIPTION            |
|---------|-----------|-----------|--------|----------|------------------------|
| 76      | (4C)      | ADDRESS   | 4      | OFNBLK10 | 10TH BLOCK             |
| 80      | (50)      | ADDRESS   | 4      |          |                        |
| 84      | (54)      | ADDRESS   | 4      | OFNBLK11 | 11TH BLOCK             |
| 88      | (58)      | ADDRESS   | 4      |          |                        |
| 92      | (5C)      | ADDRESS   | 4      | OFNBLK12 | 12TH BLOCK             |
| 96      | (60)      | ADDRESS   | 4      |          |                        |
| 100     | (64)      | ADDRESS   | 4      | OFNBLK13 | 13TH BLOCK             |
| 104     | (68)      | ADDRESS   | 4      |          |                        |
| 108     | (6C)      | ADDRESS   | 4      | OFNBLK14 | 14TH BLOCK             |
| 112     | (70)      | ADDRESS   | 4      |          |                        |
| 116     | (74)      | ADDRESS   | 4      | OFNBLK15 | 15TH BLOCK             |
| 120     | (78)      | ADDRESS   | 4      |          |                        |
| 124     | (7C)      | ADDRESS   | 4      | OFNBLK16 | 16TH BLOCK             |
| 128     | (80)      | ADDRESS   | 4      |          |                        |
| 132     | (84)      | CHARACTER | 4288   | OFNENTRY | INDEXED ENTRY TABLE    |
| 132     | (84)      | CHARACTER | 156    | OFNPART1 | FIRST HALF OF ENTRY    |
| 132     | (84)      | BITSTRING | 2      | OFNDSFLG | DATA SET INFO FLAGS    |
|         | 1... ..   |           |        | OFNINUSE | ENTRY IN USE FLAG      |
|         | .1.. ..   |           |        | OFNOPNSC | SUCCESSFUL OPEN        |
|         | ..1. .... |           |        | OFNOPNLY | OPEN ONLY REQUEST      |
|         | ...1 1111 |           |        |          | RESERVED               |
|         | 1... ..   |           |        | OFNTYPS  | SAM DATA SET           |
|         | .1.. ..   |           |        | OFNTYPP  | PAM DATA SET           |
|         | ..1. .... |           |        | OFNTYPC  | DLF DOCUMENT (FILE)    |
|         | ...1 .... |           |        | OFNTYPA  | APPLICATION INPUT DATA |
|         | .... 1... |           |        | OFNDYN   | DYNAMICALLY ALLOCATED  |
|         | .... .1.. |           |        | OFNEOF   | EOF ON LAST READ       |
|         | .... ..1  |           |        |          | RESERVED               |
|         | .... ...1 |           |        | OFNSEQ8  | NUMBERS IN 1ST 8 BYTES |
| 134     | (86)      | SIGNED    | 2      | OFNRECS  | SAM REC SIZE           |
| 136     | (88)      | UNSIGNED  | 4      | OFNLIBRN | LIBRARY NUMBER         |
| 140     | (8C)      | CHARACTER | 8      | OFNDATA  | DATA TYPE ATTRIBUTE    |
| 148     | (94)      | UNSIGNED  | 2      | OFNVERS  | VERSION NUMBER         |
| 150     | (96)      | CHARACTER | 16     | OFNDOCN  | DOCUMENT NAME          |
| 166     | (A6)      | CHARACTER | 8      | OFNDOCP  | DOCUMENT PASSWORD      |
| 174     | (AE)      | CHARACTER | 8      | OFNFRDD  | FROM DDNAME            |
| 182     | (B6)      | CHARACTER | 8      | OFNFRPDS | FROM PDS MEMBER NAME   |
| 190     | (BE)      | CHARACTER | 44     | OFNFRDSN | FROM BATCH FILE DSN    |
| 234     | (EA)      | CHARACTER | 8      | OFNSRCDA | SOURCE DATA ATTRIBUTE  |
| 242     | (F2)      | CHARACTER | 8      | OFNOUTDA | OUTPUT DATA ATTRIBUTE  |
| 250     | (FA)      | SIGNED    | 2      | OFNSRBLN | SOURCE READ BUF LENGTH |
| 252     | (FC)      | ADDRESS   | 4      | OFNSRBUF | SOURCE READ BUFFER     |
| 256     | (100)     | SIGNED    | 4      | OFNSROFS | CURRENT SOURCE REC PTR |
| 260     | (104)     | ADDRESS   | 4      | OFNCMREC | CURRENT SLE IN COM BUF |
| 264     | (108)     | ADDRESS   | 4      | OFNCMBUF | COM BUFFER ADDRESS     |
| 268     | (10C)     | SIGNED    | 2      | OFNCMBLN | COM BUFFER LENGTH      |
| 270     | (10E)     | SIGNED    | 2      | OFNWKBLN | IP WORK BUFFER LENGTH  |
| 272     | (110)     | ADDRESS   | 4      | OFNWKBUF | IP WORK BUFFER         |
| 276     | (114)     | SIGNED    | 4      | OFNRECCT | RELATIVE RECORD COUNT  |

| OFFSETS   | TYPE      | LENGTH | NAME     | DESCRIPTION             |
|-----------|-----------|--------|----------|-------------------------|
| 280 (118) | ADDRESS   | 4      | OFNDCBPT | DCB ADDRESS             |
| 284 (11C) | ADDRESS   | 4      | OFNDERPT | DER RECORD SAVE AREA    |
| 288 (120) | CHARACTER | 112    | OFNPART2 | SECOND HALF OF ENTRY    |
| 288 (120) | CHARACTER | 8      | OFNIPNAM | INPUT PROCESSOR NAME    |
| 296 (128) | ADDRESS   | 4      | OFNIPENT | INPUT PROCESSOR ENTRY   |
| 300 (12C) | ADDRESS   | 4      | OFNUPENT | USER PGM RETURN ENTRY   |
| 304 (130) | CHARACTER | 8      | OFNUPNAM | APPL EXIT PGM NAME      |
| 312 (138) | CHARACTER | 8      | OFNCLASS | CLASS NAME FOR DOCUMENT |
| 320 (140) | CHARACTER | 8      | OFNISRCE | INSOURCE NAME           |
| 328 (148) | CHARACTER | 8      | OFNOSRCE | OUTSOURCE NAME          |
| 336 (150) | CHARACTER | 64     | OFNPARM  | PARM FOR PROCESSOR      |

#### CROSS-REFERENCE

|          |     |          |     |    |
|----------|-----|----------|-----|----|
| OFNBLK1  | 4   | OFNHDR   | 0   |    |
| OFNBLK10 | 4C  | OFNID    | 0   |    |
| OFNBLK11 | 54  | OFNINUSE | 84  | 80 |
| OFNBLK12 | 5C  | OFNIPENT | 128 |    |
| OFNBLK13 | 64  | OFNIPNAM | 120 |    |
| OFNBLK14 | 6C  | OFNISRCE | 140 |    |
| OFNBLK15 | 74  | OFNLIBRN | 88  |    |
| OFNBLK16 | 7C  | OFNMAX   | 2   |    |
| OFNBLK2  | C   | OFNOPNLY | 84  | 20 |
| OFNBLK3  | 14  | OFNOPNSC | 84  | 40 |
| OFNBLK4  | 1C  | OFNOSRCE | 148 |    |
| OFNBLK5  | 24  | OFNOUTDA | F2  |    |
| OFNBLK6  | 2C  | OFNPARM  | 150 |    |
| OFNBLK7  | 34  | OFNPART1 | 84  |    |
| OFNBLK8  | 3C  | OFNPART2 | 120 |    |
| OFNBLK9  | 44  | OFNRECCT | 114 |    |
| OFNCLASS | 138 | OFNRECS  | 86  |    |
| OFNCMBLN | 10C | OFNSEQ8  | 85  | 01 |
| OFNCMBUF | 108 | OFNSRBLN | FA  |    |
| OFNCMREC | 104 | OFNSRBUF | FC  |    |
| OFNDATA  | 8C  | OFNSRCDA | EA  |    |
| OFNDCBPT | 118 | OFNSROFS | 100 |    |
| OFNDERPT | 11C | OFNTABLE | 0   |    |
| OFNDOCN  | 96  | OFNTYPA  | 85  | 10 |
| OFNDOCP  | A6  | OFNTYPC  | 85  | 20 |
| OFNDSFLG | 84  | OFNTYPP  | 85  | 40 |
| OFNDYN   | 85  | OFNTYPS  | 85  | 80 |
| OFNENTRY | 84  | OFNUPENT | 12C |    |
| OFNEOF   | 85  | OFNUPNAM | 130 |    |
| OFNFRDD  | AE  | OFNVERS  | 94  |    |
| OFNFRDSN | BE  | OFNWKBLN | 10E |    |
| OFNFRPDS | B6  | OFNWKBUF | 110 |    |

## Standard Register Save Area Used By DLF (SAV)

All DLF modules with executable code provide a register save area that is used to save its registers when it calls another routine.

### SAV

| OFFSETS | TYPE  | LENGTH | NAME                             | DESCRIPTION   |
|---------|---|--------|----------------------------------|---|
| 0       | (0) STRUCTURE                                     | 76     | SAV                              |   |
| 0       | (0) CHARACTER                                     | 12     | SAVCNTRL                         | CONTROL AREA  |
| 0       | (0) BITSTRING<br>1... ..<br>.1... ..<br>..11 1111 | 1      | SAVFLAGS<br>SAVINUSE<br>SAVINVLD | CONTROL FLAGS:<br>1=SAVE AREA IN USE<br>1=INVALID SAVE AREA<br>RESERVED |
| 1       | (1) CHARACTER                                     | 3      | SAVMODNM                         | MODULE NAME OF OWNER  |
| 4       | (4) ADDRESS                                       | 4      | SAVPREV                          | ADDRESS OF PREVIOUS SAVE AREA   |
| 8       | (8) ADDRESS                                       | 4      | SAVNEXT                          | ADDRESS OF NEXT SAVE AREA   |
| 12      | (C) CHARACTER                                     | 64     | SAVREGS                          | SAVED REGISTERS:  |
| 12      | (C) ADDRESS                                       | 4      | SAVR14                           | REGISTER 14   |
| 16      | (10) ADDRESS                                      | 4      | SAVR15                           | REGISTER 15   |
| 20      | (14) ADDRESS                                      | 4      | SAVR0                            | REGISTER 0  |
| 24      | (18) ADDRESS                                      | 4      | SAVR1                            | REGISTER 1  |
| 28      | (1C) ADDRESS                                      | 4      | SAVR2                            | REGISTER 2  |
| 32      | (20) ADDRESS                                      | 4      | SAVR3                            | REGISTER 3  |
| 36      | (24) ADDRESS                                      | 4      | SAVR4                            | REGISTER 4  |
| 40      | (28) ADDRESS                                      | 4      | SAVR5                            | REGISTER 5  |
| 44      | (2C) ADDRESS                                      | 4      | SAVR6                            | REGISTER 6  |
| 48      | (30) ADDRESS                                      | 4      | SAVR7                            | REGISTER 7  |
| 52      | (34) ADDRESS                                      | 4      | SAVR8                            | REGISTER 8  |
| 56      | (38) ADDRESS                                      | 4      | SAVR9                            | REGISTER 9  |
| 60      | (3C) ADDRESS                                      | 4      | SAVR10                           | REGISTER 10   |
| 64      | (40) ADDRESS                                      | 4      | SAVR11                           | REGISTER 11   |
| 68      | (44) ADDRESS                                      | 4      | SAVR12                           | REGISTER 12   |
| 72      | (48) ADDRESS                                      | 4      | SAVR13                           | REGISTER 13   |
| 76      | (4C) CHARACTER                                    | 0      | SAVDSA                           | MODULE DYNAMIC STORAGE AREA   |

**CROSS-REFERENCE**

|          |    |    |        |    |
|----------|----|----|--------|----|
| SAV      | 0  |    | SAVR11 | 40 |
| SAVCNTRL | 0  |    | SAVR12 | 44 |
| SAVDSA   | 4C |    | SAVR13 | 48 |
| SAVFLAGS | 0  |    | SAVR14 | C  |
| SAVINUSE | 0  | 80 | SAVR15 | 10 |
| SAVINVLD | 0  | 40 | SAVR2  | 1C |
| SAVMODNM | 1  |    | SAVR3  | 20 |
| SAVNEXT  | 8  |    | SAVR4  | 24 |
| SAVPREV  | 4  |    | SAVR5  | 28 |
| SAVREGS  | C  |    | SAVR6  | 2C |
| SAVR0    | 14 |    | SAVR7  | 30 |
| SAVR1    | 18 |    | SAVR8  | 34 |
| SAVR10   | 3C |    | SAVR9  | 38 |

## Library Global Table (SPG)

The Library Global Table (SPG) is a major data area used for communication between modules. Register 11 points to SPG throughout processing, except when any non-DLF module has control.

The following table shows the format of the SPG data area.

### SPG

| OFFSETS | TYPE           | LENGTH | NAME     | DESCRIPTION                |
|---------|----------------|--------|----------|----------------------------|
| 0       | (0) STRUCTURE  | 20480  | SPG      |                            |
| 0       | (0) CHARACTER  | 8      | SPGISTMP | INPUT LIBRARY TIMESTAMP    |
| 8       | (8) CHARACTER  | 8      | SPGTSTMP | OUTPUT LIBRARY TIMESTAMP   |
| 16      | (10) BITSTRING | 1      | SPGOPENA | VSAM OPEN FLAGS            |
|         | 1... ..        |        | SPGDI    | INPUT DIRECTORY OPEN       |
|         | .1... ..       |        | SPGDO    | OUTPUT DIRECTORY OPEN      |
|         | ...1... ..     |        | SPGPRSTN | OUTPUT LIBRARY UNFORMATTED |
|         | ...1 1111      |        |          | RESERVED                   |
| 17      | (11) BITSTRING | 1      | SPGOPEND | OS OPEN FLAGS              |
|         | 1... ..        |        | SPGSIN   | SYSIN DATA SET OPEN        |
|         | .1... ..       |        | SPGSPR   | DSMLIST DATA SET OPEN      |
|         | ...1... ..     |        | SPGRD    | INPUT OS DATA SET OPEN     |
|         | ...1... ..     |        | SPGWQ    | OUTPUT QSAM DATA SET OPEN  |
|         | .... 1... ..   |        | SPGWB    | OUTPUT BSAM DATA SET OPEN  |
|         | .... .1... ..  |        | SPGNM    | INPUT NAMELIST OS DATA SET |
|         | .... .1... ..  |        | SPGSYEOB | SYSIN EODAD FLAG SEEN      |
|         | .... .1... ..  |        | SPGCLOG  | CMD LOG DATA SET OPEN      |
| 18      | (12) CHARACTER | 1      | SPGOPRTC | OPERATOR ROUTING CODE      |
| 19      | (13) CHARACTER | 1      |          | RESERVED                   |
| 20      | (14) ADDRESS   | 4      | SPGIDCBP | PTR TO SYSIN DCB IN SPEXC  |

### DSMSPMSG PARAMETER AREA

|    |                |    |          |                           |
|----|----------------|----|----------|---------------------------|
| 24 | (18) CHARACTER | 32 | SPGMSGD  | CANNED MSG FILL-IN FIELDS |
| 24 | (18) UNSIGNED  | 3  | SPGMSGLB | LIBRARY NO.(BINARY)       |
| 24 | (18) UNSIGNED  | 3  | SPGMSGRS | VSAM REASON CODE          |
| 27 | (1B) CHARACTER | 16 | SPGMSGFL | DOCUMENT NAME (EBCDIC)    |
| 27 | (1B) CHARACTER | 10 | SPGMSGEX | MAP EXTERNAL ID (EBCDIC)  |
| 37 | (25) CHARACTER | 2  | SPGMSGHO | MAP HOST ID (EBCDIC)      |
| 43 | (2B) CHARACTER | 8  | SPGMSGDD | DDNAME (EBCDIC)           |
| 43 | (2B) CHARACTER | 8  | SPGMSGOP | VSAM OPERATION NAME       |
| 51 | (33) CHARACTER | 3  |          | UNUSED RESERVED           |
| 54 | (36) UNSIGNED  | 2  | SPGMSGRC | RETURN CODE (BINARY)      |
| 54 | (36) UNSIGNED  | 2  | SPGMSGVV | VERSION NUMBER            |

### AUTHORIZATION DATA FOR CURRENT USER

| OFFSETS | TYPE           | LENGTH | NAME     | DESCRIPTION                         |
|---------|----------------|--------|----------|-------------------------------------|
| 56      | (38) UNSIGNED  | 3      | SPGUSID  | AUTH USERID (BINARY) (CURRENT USER) |
| 59      | (3B) BITSTRING | 1      | SPGAUTHF | FLAGS TO BE USED BY AUTH CMD        |
|         | 1... ..        |        | SPGRACFI | RACF PROTECTION INPUT LIB           |
|         | .1... ..       |        | SPGRACFO | RACF PROTECTION OUTPUT LIB          |
|         | ..11 1111      |        |          | RESERVED                            |
| 60      | (3C) ADDRESS   | 4      | SPGHSHAD | HASHING PROCEDURE ENTRY POINT       |
| 64      | (40) ADDRESS   | 4      | SPGUXIAD | DSMSPUXI ENTRY PNT ADDR             |
| 68      | (44) UNSIGNED  | 3      | SPGPROJI | PROJECT LIBRARY NUMBER INPUT LIB    |
| 71      | (47) BITSTRING | 1      | SPGPRIVI | PRIVILEGE FLAG INPUT LIBRARY        |
|         | 1... ..        |        | SPGEXSTI | USER EXISTS IN INPUT LIB            |
|         | .1... ..       |        |          | RESERVED                            |
|         | ..1... ..      |        | SPGSPRVI | SYSTEM ADMIN PRIVILEGES             |
|         | ...1 ..        |        | SPGPRJCI | CONTROLLED PROJECT LIBRARY          |
|         | .... 1111      |        |          | RESERVED                            |
| 72      | (48) UNSIGNED  | 3      | SPGPROJO | PROJECT LIBRARY NUMBER OUTPUT LIB   |
| 75      | (4B) BITSTRING | 1      | SPGPRIVO | PRIVILEGE FLAG OUTPUT LIBRARY       |
|         | 1... ..        |        | SPGEXSTO | USER EXISTS IN OUTPUT LIB           |
|         | .1... ..       |        |          | RESERVED                            |
|         | ..1... ..      |        | SPGSPRVO | SYSTEM ADMIN PRIVILEGES             |
|         | ...1 ..        |        | SPGPRJCO | CONTROLLED PROJECT LIBRARY          |
|         | .... 1111      |        |          | RESERVED                            |

#### LIBRARY ENQUEUE/LOCK DATA

|    |                |    |          |                         |
|----|----------------|----|----------|-------------------------|
| 76 | (4C) BITSTRING | 1  | SPGENQFL | ENQUE/LOCK FLAG BYTE    |
|    | 1... ..        |    | SPGENQX  | EXCLUSIVE ENQ/LOCK FLAG |
|    | .1... ..       |    | SPGENQS  | SHARED ENQUE/LOCK FLAG  |
|    | ..11 1111      |    |          | UNUSED RESERVED         |
| 77 | (4D) CHARACTER | 7  |          | UNUSED RESERVED         |
| 84 | (54) CHARACTER | 12 | SPGORNAM | OUT LIB. ENQ-LOCK NAME  |
| 96 | (60) CHARACTER | 12 | SPGIRNAM | IN LIB. ENQ-LOCK NAME   |

#### RACF VARIABLE AREA

|     |                |   |          |                          |
|-----|----------------|---|----------|--------------------------|
| 108 | (6C) UNSIGNED  | 1 |          | RESERVED                 |
| 109 | (6D) UNSIGNED  | 1 | SPGIRACL | INPUT LIB RACF NAME LEN  |
| 110 | (6E) CHARACTER | 8 | SPGIRACN | INPUT LIBRARY RACF NAME  |
| 118 | (76) UNSIGNED  | 1 |          | RESERVED                 |
| 119 | (77) UNSIGNED  | 1 | SPGORACL | OUTPUT LIB RACF NAME LEN |
| 120 | (78) CHARACTER | 8 | SPGORACN | OUTPUT LIBRARY RACF NAME |

#### USER DEFAULTS INPUT LIBRARY ATTRIBUTE DEFAULTS

|     |                |   |         |                             |
|-----|----------------|---|---------|-----------------------------|
| 128 | (80) CHARACTER | 8 | SPGIDFD | IN LIB DEF DATA ATTR NAME   |
| 136 | (88) CHARACTER | 8 | SPGIDFC | IN LIB DEF CLASS ATTR NAME  |
| 144 | (90) CHARACTER | 8 | SPGIDFS | IN LIB DEF SOURCE ATTR NAME |
| 152 | (98) CHARACTER | 8 | SPGIDFL | IN LIB DEF CLUSTER NAME     |

#### OUTPUT LIBRARY ATTRIBUTE DEFAULTS

| OFFSETS | TYPE           | LENGTH | NAME    | DESCRIPTION                  |
|---------|----------------|--------|---------|------------------------------|
| 160     | (A0) CHARACTER | 8      | SPGODFD | OUT LIB DEF DATA ATTR NAME   |
| 168     | (A8) CHARACTER | 8      | SPGODFC | OUT LIB DEF CLASS ATTR NAME  |
| 176     | (B0) CHARACTER | 8      | SPGODFS | OUT LIB DEF SOURCE ATTR NAME |
| 184     | (B8) CHARACTER | 8      | SPGODFL | OUT LIB DEF CLUSTER NAME     |

FLAGS USED FOR DSMSPLBI,DSMSPDSI INTERFACE

|     |                |     |          |   |
|-----|----------------|-----|----------|---|
| 192 | (C0) BITSTRING | 2   | SPGLBII  | DSMSPLBI CONTROL FLAGS                        |
|     | 1... ..        |     | SPGREQ   | READ KEY EQUAL                                |
|     | .1.. ..        |     | SPGRGE   | READ KEY GREATER/EQUAL                        |
|     | ..1. ....      |     | SPGGR    | READ GENERIC                                  |
|     | ...1 ....      |     | SPGRS    | READ SEQUENTIAL(DSMSPLBI) 1ST CALL (DSMSPDSI) |
|     | .... 1...      |     | SPGRU    | READ FOR UPDATE                               |
|     | .... .1..      |     |          | UNUSED RESERVED                               |
|     | .... ..1.      |     | SPGRFI   | READ FILE C.I.                                |
|     | .... ...1      |     | SPGDR    | READ DIRECT                                   |
|     | 1... ..        |     | SPGRPL2  | USE SPGIRPL2                                  |
|     | .1... ..       |     |          | UNUSED RESERVED                               |
|     | ..11 1111      |     |          | RESERVED                                      |
| 194 | (C2) UNSIGNED  | 2   | SPGLBIG  | GENERIC KEY SEARCH LTH                        |
| 196 | (C4) CHARACTER | 256 | SPGDIBUF | DIRECTORY INPUT BUFFER                        |

|     |               |   |          |                             |
|-----|---------------|---|----------|-----------------------------|
| 452 | (1C4) ADDRESS | 4 | SPGIPENT | INPUT PROCESSOR ENTRY POINT |
|-----|---------------|---|----------|-----------------------------|

FLAGS USED FOR DSMSPDSO, DSMSPLBO, DSMSPTPO INTERFACE  
THERE ARE 5 POSSIBLE TYPES OF CALLS TO DSMSPLBO TO STORE  
A SINGLE DOCUMENT IN THE DOCUMENT LIBRARY. STORAGE  
OF EACH DOCUMENT REQUIRES AT LEAST CALLS 1 & 5 PLUS ALL,  
OR A REASONABLE COMBINATION, OF CALLS 2-4 DEPENDING  
ON DOCUMENT LENGTH. THE CALLS, IN SEQUENCE, ARE:  
1.SPGUPRRD=ON & SPGDOBUF CONTAINS THE DEROWNR'S UPR  
2.SPGCIANY=ON, SPGCIBEG=ON, SPGDOBUF CONTAINS THE DER  
OF THE DOCUMENT BEING STORED, & SPG16K CONTAINS THE  
FIRST CI OF THE DOCUMENT BEING STORED  
3.SPGCIANY=ON & SPG16K CONTAINS A CI, ANY CI EXCEPT  
THE FIRST OR LAST CI  
4.SPGCIANY=ON, SPGCIEND=ON, & SPG16K CONTAINS THE LAST  
CI OF THE DOCUMENT BEING STORED  
5.SPGUPRWT=ON  
SEE THE PROLOGUE OF MODULE DSMSPLBO FOR THE DETAILS OF  
THE COMPLETE CALLING SEQUENCE.  
DSMSPDSO AND DSMSPTPO USE FLAGS SPGWD AND SPGWFI FOR  
WRITING TO AN EXTERNAL FILE.

| OFFSETS | TYPE  | LENGTH | NAME   | DESCRIPTION   |
|---------|---|--------|--|---|
| 456     | (1C8) BITSTRING<br>1... ..<br>.1.. ....<br>..1. ....<br>...1 ....<br>.... 1..<br>.... .1..<br>.... ..1. | 1      | SPGLBOI<br>SPGWD<br><br>SPGWF1<br>SPGUPRRD<br><br>SPGUPRWT<br><br>SPGCIANY<br>SPGCIBEG<br><br>SPGCIEND | DSMSPLBO CONTROL FLAGS<br>WRITE DIRECTORY RECORD TPO AND DSO<br>COMMUNICATION<br>WRITE FILE CI TPO AND DSO COMMUNICATION<br>DO UPR SETUP ON THE FIRST CALL TO LBO<br>FOR THIS DOCUMENT UPR OF DEROWNR PASSED<br>IN SPG SPGDOBUF<br>DO CLEANUP AND UPDATE UPR ON FINAL CALL<br>TO LBO FOR THIS DOCUMENT<br>WRITE A CI ON THIS CALL<br>TEST THE DER AND WRITE THE FIRST CI ON<br>THIS CALL<br>WRITE THE LAST CI, THE DER, AND THE LER<br>ON THIS CALL<br>RESERVED<br>RESERVED |
| 457     | (1C9) CHARACTER<br>.....1   | 3      |  |   |
| 460     | (1CC) CHARACTER   | 260    | SPGDOVBT   | VARIABLE BLOCKED TAPE BUFFER USED BY<br>DSMSPTPO  |
| 460     | (1CC) UNSIGNED  | 4      | SPGDORDW   | RECORD DESCR WORD   |
| 460     | (1CC) SIGNED  | 2      | SPGDRLEN   | RECORD DATA LENGTH  |
| 462     | (1CE) UNSIGNED  | 2      | SPGDR00  | RESERVED  |
| 464     | (1D0) CHARACTER   | 256    | SPGDOBUF   | DIRECTORY OUTPUT BUFFER   |

#### FLAGS USED FOR DSMSPOLU INTERFACE

|     |   |   |   |   |
|-----|---|---|---|---|
| 720 | (2D0) BITSTRING<br>1... ..<br>.1.. ....<br>..1. ....<br>...1 ....<br>.... 1..<br>.... .1..<br>.... ..11 | 1 | SPGOLUI<br>SPGOKSDS<br>SPGOESDS<br>SPGOKEQ<br>SPGOFKS<br>SPGODIR<br>SPGOINS | DSMSPOLU CONTROL FLAGS<br>ACCESS DIRECTORY DATA SET<br>ACCESS ESDS DATA SET<br>RPL KEQ IF ON, KGE IF OFF<br>RPL FKS IF ON, GEN IF OFF<br>RPL DIR IF ON, SEQ IF OFF<br>INSERT PROCESSING REQUEST<br>RESERVED |
| 721 | (2D1) CHARACTER   | 1 | SPGOACSS  | ACCESS MODE 'E' ERASE 'R' READ 'W'<br>WRITE   |
| 722 | (2D2) BITSTRING   | 2 | SPGOKEYL  | GENERIC KEY LENGTH  |
| 724 | (2D4) UNSIGNED  | 2 | SPGORECL  | RECORD LENGTH FOR INSERT PROC   |
| 726 | (2D6) CHARACTER   | 2 |   | UNUSED RESERVED   |
| 728 | (2D8) ADDRESS   | 4 | SPGOARG   | POINTER TO VSAM SEARCH ARG  |
| 732 | (2DC) ADDRESS   | 4 | SPGOAREA  | POINTER TO I/O BUFFER   |

#### FLAGS USED FOR DSMSPILU INTERFACE

|     |  |   |   |  |
|-----|--|---|---|--|
| 736 | (2E0) BITSTRING<br>1... ..<br>.1.. ....<br>..1. ....<br>...1 ....<br>.... 1..<br>.... .111 | 1 | SPGILUI<br>SPGIESDS<br><br>SPGIKEQ<br>SPGIFKS<br>SPGIDIR<br>SPGIUPD | DSMSPILU CONTROL FLAGS<br>ACCESS INPUT ESDS DATA SET IF ESDS IS<br>NOT INDICATED, THEN ACCESS THE<br>DIRECTORY<br>RPL KEQ IF ON, KGE IF OFF<br>RPL FKS IF ON, GEN IF OFF<br>RPL DIR IF ON, SEQ IF OFF<br>UPDATE IF ON, NOUPDATE IF OFF<br>RESERVED |
| 737 | (2E1) CHARACTER  | 1 | SPGIACSS  | ACCESS MODE 'E' ERASE 'R' READ 'W'<br>WRITE  |
| 738 | (2E2) BITSTRING  | 2 | SPGIKEYL  | GENERIC KEY LENGTH   |
| 740 | (2E4) ADDRESS  | 4 | SPGIARG   | POINTER TO VSAM SEARCH ARG   |
| 744 | (2E8) ADDRESS  | 4 | SPGIAREA  | POINTER TO I/O BUFFER  |



| OFFSETS | TYPE | LENGTH | NAME | DESCRIPTION |
|---------|------|--------|------|-------------|
|---------|------|--------|------|-------------|

|                                      |
|--------------------------------------|
| CLUSTER CONTROL INTERFACE (DSMSPGCL) |
|--------------------------------------|

|     |       |           |   |           |                                |
|-----|-------|-----------|---|-----------|--------------------------------|
| 748 | (2EC) | CHARACTER | 8 | SPGCLINM  | NAME OF INPUT CLUSTER TO OPEN  |
| 756 | (2F4) | CHARACTER | 8 | SPGCLONM  | NAME OF OUTPUT CLUSTER TO OPEN |
| 764 | (2FC) | BITSTRING | 1 | SPGCLFLG  | CLUSTER REQUEST FLAGS          |
|     |       | 1... ..   |   | SPGICLST  | INPUT CLUSTER REQUEST          |
|     |       | .1.. ..   |   | SPGOCCLST | OUTPUT CLUSTER REQUEST         |
|     |       | ..1. .... |   | SPGOCCLBI | INITIAL OUT REQ FROM PIN       |
|     |       | ...1 .... |   | SPGWFHSH  | WRITE FSH TO CI=0 AP>          |
|     |       | .... 1... |   | SPGDACSP  | INDICATOR USED BY DAC          |
|     |       | .... .1.. |   | SPGCCL0   | CLOSE BOTH IF SAME             |
|     |       | .... ..1. |   | SPGICLO   | CLOSE INPUT CLB                |
|     |       | .... ...1 |   | SPGOCLO   | CLOSE OUTPUT CLB               |
| 765 | (2FD) | CHARACTER | 3 |           | UNUSED RESERVED                |
| 768 | (300) | ADDRESS   | 4 | SPGCLIFP  | 1ST CLB ON INPUT CHAIN         |
| 772 | (304) | ADDRESS   | 4 | SPGCLOFP  | 1ST CLB ON OUTPUT CHAIN        |
| 776 | (308) | ADDRESS   | 4 | SPGCLBIA  | ACTIVE INPUT CLB POINTER       |
| 780 | (30C) | ADDRESS   | 4 | SPGCLBOA  | ACTIVE OUTPUT CLB POINTER      |

|                            |
|----------------------------|
| SUBROUTINE CALL PARAMETERS |
|----------------------------|

|     |       |           |    |          |  |
|-----|-------|-----------|----|----------|--|
| 784 | (310) | ADDRESS   | 4  | SPGCALR1 | CALLERS REGISTER 1                             |
| 788 | (314) | CHARACTER | 24 | SPGCALED | PARAMETERS SET IF DLF CALLED AS SUBR           |
| 788 | (314) | ADDRESS   | 4  | SPGCALL1 | PTR TO 'LIST/NOLIST'                           |
|     |       | 1... ..   |    | SPGEXEQD | FLAG OFF IF CALLED, FLAG ON IF EXECUTED        |
| 792 | (318) | ADDRESS   | 4  | SPGCALL2 | PTR TO COMMAND LENGTH                          |
| 796 | (31C) | ADDRESS   | 4  | SPGCALL3 | PTR TO COMMAND                                 |
| 800 | (320) | ADDRESS   | 4  | SPGCALL4 | PTR TO RETURN CODE                             |
| 804 | (324) | ADDRESS   | 4  | SPGCALL5 | PTR TO READ BUFFER LEN PTR TO FORMAT ENTRY ADR |
| 808 | (328) | ADDRESS   | 4  | SPGCALL6 | PTR TO READ BUFFER                             |

|   |
|---|
| INFO RETURNED TO CALLER WHEN DLF CALLED AS A SUBROUTINE |
|---|

|     |       |        |   |         |                     |
|-----|-------|--------|---|---------|---------------------|
| 812 | (32C) | SIGNED | 4 | SPGRETM | RSN CODE FOR CALLER |
| 816 | (330) | SIGNED | 4 | SPGRETV | VERS # FOR CALLER   |

|                   |
|-------------------|
| COMMAND WORK AREA |
|-------------------|

|     |       |           |   |         |                      |
|-----|-------|-----------|---|---------|----------------------|
| 820 | (334) | BITSTRING | 1 | SPGEXEC | EXEC STATEMENT PARMS |
|     |       | 1... ..   |   | SPGLIST | LIST PARAMETER       |
| 821 | (335) | UNSIGNED  | 3 |         | RESERVED             |
|     |       |           |   |         | RESERVED             |
| 824 | (338) | SIGNED    | 2 | SPGCOMM | COMMAND CODE         |
| 826 | (33A) | SIGNED    | 2 | SPGSUBC | SUBCOMMAND CODE      |

| OFFSETS | TYPE | LENGTH | NAME | DESCRIPTION |
|---------|------|--------|------|-------------|
|---------|------|--------|------|-------------|

|  |  |  |  |  |
|--|--|--|--|--|
| LOCATION OF DYNAMIC STORAGE GOTTEN FOR DCBS AND READ BUFFERS |  |  |  |  |
|--|--|--|--|--|

|     |       |         |   |          |                        |
|-----|-------|---------|---|----------|------------------------|
| 828 | (33C) | ADDRESS | 4 | SPGSICSP | PTR TO GETMAIN STORAGE |
|-----|-------|---------|---|----------|------------------------|

|   |  |  |  |  |  |
|---|--|--|--|--|--|
| COMMAND PARAMETER AREA.<br>VALUES SET BY DSMSPCMD,<br>REFERENCED AS NOTED |  |  |  |  |  |
|---|--|--|--|--|--|

|     |       |           |     |          |  |
|-----|-------|-----------|-----|----------|--|
| 832 | (340) | CHARACTER | 694 | SPGPARGA |  |
|-----|-------|-----------|-----|----------|--|

|  |  |  |  |  |  |
|--|--|--|--|--|--|
| TIME DIFFERENCE, IN MICROSECONDS, BETWEEN LOCAL TIME AND<br>GREENWICH. VALUE IS NEGATIVE IF WEST OF GREENWICH. |  |  |  |  |  |
|--|--|--|--|--|--|

|     |       |           |   |          |  |
|-----|-------|-----------|---|----------|--|
| 832 | (340) | CHARACTER | 8 | SPGTMDIF |  |
|-----|-------|-----------|---|----------|--|

|   |  |  |  |  |  |
|---|--|--|--|--|--|
| DATE OR DATE-1 IN ARCHIVE ALL, LIST DOCUMENT, AND<br>RETRIEVE ALL |  |  |  |  |  |
|---|--|--|--|--|--|

|     |       |           |   |         |                       |
|-----|-------|-----------|---|---------|-----------------------|
| 840 | (348) | CHARACTER | 8 | SPGDAT1 | FLOATING POINT DATE-1 |
|-----|-------|-----------|---|---------|-----------------------|

|     |       |           |   |  |  |
|-----|-------|-----------|---|--|--|
| 840 | (348) | CHARACTER | 4 |  |  |
|-----|-------|-----------|---|--|--|

|     |       |        |   |          |                  |
|-----|-------|--------|---|----------|------------------|
| 844 | (34C) | SIGNED | 4 | SPGDATF1 | FIXED POINT DAYS |
|-----|-------|--------|---|----------|------------------|

|  |  |  |  |  |  |
|--|--|--|--|--|--|
| DATE-2 IN LIST DOCUMENT AND RETRIEVE ALL |  |  |  |  |  |
|--|--|--|--|--|--|

|     |       |           |   |         |                       |
|-----|-------|-----------|---|---------|-----------------------|
| 848 | (350) | CHARACTER | 8 | SPGDAT2 | FLOATING POINT DATE-2 |
|-----|-------|-----------|---|---------|-----------------------|

|     |       |           |   |  |  |
|-----|-------|-----------|---|--|--|
| 848 | (350) | CHARACTER | 4 |  |  |
|-----|-------|-----------|---|--|--|

|     |       |        |   |          |                  |
|-----|-------|--------|---|----------|------------------|
| 852 | (354) | SIGNED | 4 | SPGDATF2 | FIXED POINT DAYS |
|-----|-------|--------|---|----------|------------------|

|  |  |  |  |  |  |
|--|--|--|--|--|--|
| FROM(DNAME) IN ACCOUNT, COPY IN/OUT, IMPORT,<br>AND RETRIEVE |  |  |  |  |  |
|--|--|--|--|--|--|

|     |       |           |   |          |  |
|-----|-------|-----------|---|----------|--|
| 856 | (358) | CHARACTER | 8 | SPGFRMDD |  |
|-----|-------|-----------|---|----------|--|

|   |  |  |  |  |  |
|---|--|--|--|--|--|
| TO(DNAME) IN ACCOUNT, ARCHIVE, COPY IN/OUT,<br>EXPORT, AND RETRIEVE |  |  |  |  |  |
|---|--|--|--|--|--|

|     |       |           |   |         |  |
|-----|-------|-----------|---|---------|--|
| 864 | (360) | CHARACTER | 8 | SPGTODD |  |
|-----|-------|-----------|---|---------|--|

|  |  |  |  |  |  |
|--|--|--|--|--|--|
| FROM(DSN(DSNAME)) IN ACCOUNT, COPY IN/OUT, IMPORT,<br>AND RETRIEVE |  |  |  |  |  |
|--|--|--|--|--|--|

|     |       |           |    |          |  |
|-----|-------|-----------|----|----------|--|
| 872 | (368) | CHARACTER | 44 | SPGFRDSN |  |
|-----|-------|-----------|----|----------|--|

| OFFSETS | TYPE | LENGTH | NAME | DESCRIPTION |
|---------|------|--------|------|-------------|
|---------|------|--------|------|-------------|

FROM(DSN(...(MEMBERNAME))) IF DATA SET IS A PDS  
IN ACCOUNT, COPY IN/OUT, IMPORT,  
AND RETRIEVE

916 (394) CHARACTER 8 SPGFRMEM

TO(DSN(DSNAME)) IN ACCOUNT, ARCHIVE, COPY IN/OUT,  
EXPORT, AND LIST DOCUMENT

924 (39C) CHARACTER 44 SPGTODSN

TO(DSN(...(MEMBERNAME))) IF TO DATA SET IS A PDS  
IN ACCOUNT, ARCHIVE, COPY IN/OUT,  
EXPORT, AND LIST DOCUMENT

968 (3C8) CHARACTER 8 SPGTOMEM

NAMelist (DSN('DSNAME')) IN ARCHIVE AND RETRIEVE

976 (3D0) CHARACTER 8 UNUSED RESERVED

984 (3D8) CHARACTER 44 SPGDSNM2 SECOND DATA SET NAME

1028 (404) CHARACTER 48 UNUSED RESERVED

NAMelist (DSN('...(MEMBERNAME)')) - OR - (DDNAME)  
IN ARCHIVE AND RETRIEVE

1076 (434) CHARACTER 8 SPGDSME2 SECOND MEMBER NAME

1084 (43C) CHARACTER 8 UNUSED RESERVED

INDD(AAAAAAAAA) IN ALTER/DEFINE CLUSTER  
INDATA(AAAAAAAAA) IN ALTER/DEFINE/PURGE PROCESS AND IMPORT  
DATANAME2 IN COPY IN/OUT DATA(DATANAME1:DATANAME2)

1092 (444) CHARACTER 8 SPGNAME1

OUTDD(AAAAAAAAA) IN ALTER/DEFINE CLUSTER  
OUTDATA(AAAAAAAAA) IN ALTER/DEFINE/PURGE PROCESS+EXPORT + READ  
NEWDATA(AAAAAAAAA) IN ALTER DOCUMENT

1100 (44C) CHARACTER 8 SPGNAME2

PASSWORD IN AUTH  
MEM1-PASS IN COPY  
FILE-PASS IN EXPORT, IMPORT  
PASSWD(A) IN DEF/ALT USER

1108 (454) CHARACTER 8 SPGPASS1

| OFFSETS | TYPE | LENGTH | NAME | DESCRIPTION |
|---------|------|--------|------|-------------|
|---------|------|--------|------|-------------|

MEMNEW-PASS IN COPY

1116 (45C) CHARACTER 8 SPGPASS2

'USERNUM' IN AUTH, DEF/ALT USER  
'USERNUM1' IN COPY, ACCOUNT

1124 (464) UNSIGNED 1 RESERVED  
1125 (465) UNSIGNED 3 SPGUSR1

'USERNUM2' IN ACCOUNT, COPY  
MODEL(USERNUM) IN DEF/ALT USER  
FOR(USER NUM) IN IMPORT

1128 (468) UNSIGNED 1 RESERVED  
1129 (469) UNSIGNED 3 SPGUSR2

'NEWNUM' IN COPY  
PROJLIB(USERNUM) IN DEF/ALT USER

1132 (46C) UNSIGNED 1 RESERVED  
1133 (46D) UNSIGNED 3 SPGUSR3

SPACE(N) IN DEF/ALT USER

1136 (470) SIGNED 4 SPGSPACN MAX LIBRARY SPACE IN 1000'S OF BYTES

LIBTYPE( ) IN DEFINE/ALTER USER

1140 (474) CHARACTER 4 SPGLIBTC

1140 (474) CHARACTER 1 SPGLIBTY 'B'=PUB, 'O'=PROJ, 'I'=PRIV  
1141 (475) CHARACTER 3 RESERVED

FILENAME IN IMPORT, EXPORT  
'MEM1' IN COPY

1144 (478) CHARACTER 19 SPGFLNM1

1144 (478) UNSIGNED 3 SPGLIB1  
1147 (47B) CHARACTER 16 SPGMEMN1  
1163 (48B) CHARACTER 1 RESERVED

'MEM2' IN COPY, 'MEMNEW' IN COPY

1164 (48C) CHARACTER 19 SPGFLNM2

1164 (48C) UNSIGNED 3 SPGLIB2  
1167 (48F) CHARACTER 16 SPGMEMN2  
1183 (49F) CHARACTER 1 RESERVED

| OFFSETS | TYPE | LENGTH | NAME | DESCRIPTION |
|---------|------|--------|------|-------------|
|---------|------|--------|------|-------------|

DATA(AAAAAAAA) IN ALL COMMANDS USING DATA OPERAND  
 DATANAME1 IN DATA(DATANAME1;DATANAME2) IN COPY IN/OUT  
 ENTRY(AAAAAAAA) IN ALTER/DEFINE PROCESS

-----  
 1184 (4A0) CHARACTER 8 SPGDATA1

INDATA IN IMPORT COMMAND  
 OUTDATA IN EXPORT COMMAND

-----  
 1192 (4A8) CHARACTER 8 SPGAUXD

CLASS(AAAAAAAA) IN ALL COMMANDS USING CLASS OPERAND  
 NEWCLASS(AAAAAAAA) IN ALTER DOCUMENT COMMAND

-----  
 1200 (4B0) CHARACTER 8 SPGCLASS

CLUSTER(AAAAAAAA) IN ALL COMMANDS USING CLUSTER OPERAND

-----  
 1208 (4B8) CHARACTER 8 SPGCLUST

SOURCE(AAAAAAAA) IN ALL COMMANDS USING SOURCE(OPERAND)  
 NEWSOURCE(AAAAAAAA) IN ALTER DOCUMENT COMMAND  
 INSOURCE(AAAAAAAA) IN IMPORT COMMAND  
 OUTSOURCE(AAAAAAAA) IN EXPORT COMMAND

-----  
 1216 (4C0) CHARACTER 8 SPGSOURC

-----  
 1224 (4C8) CHARACTER 8 SPGINSRC

HOST('IDENTIFIER') OPERAND NAMEHOLDER

-----  
 1232 (4D0) CHARACTER 2 SPGAHOST

NEWHOST('IDENTIFIER') OPERAND NAMEHOLDER

-----  
 1234 (4D2) CHARACTER 2 SPGNHOST

ID('IDENTIFIER') OPERAND NAMEHOLDER

-----  
 1236 (4D4) CHARACTER 10 SPGAEXID

NEWID('IDENTIFIER') OPERAND NAMEHOLDER

-----  
 1246 (4DE) CHARACTER 10 SPGNEWID

| OFFSETS | TYPE | LENGTH | NAME | DESCRIPTION |
|---------|------|--------|------|-------------|
|---------|------|--------|------|-------------|

NOTE(' 44 CHARACTER IDENTIFIER ') NAMEHOLDER

1256 (4E8) CHARACTER 44 SPGANOTE

DOCINFO('50 CHARACTER IDENTIFIER ') NAMEHOLDER

1300 (514) CHARACTER 50 SPGDOCIF

APPLINFO('32 CHARACTER IDENTIFIER ') NAMEHOLDER

1350 (546) CHARACTER 75 SPGAPPIF  
1425 (591) CHARACTER 1

PROCESSOR PARAMETER INFORMATION PASSED TO USER PROCESSORS

1426 (592) CHARACTER 64 SPGPARM

TYPE, UNIT, RECORDSIZE, BLOCKSIZE, FORMAT FROM ENVIRONMENT  
COMMAND.

1490 (5D2) CHARACTER 10 SPGENV  
1490 (5D2) UNSIGNED 2 SPGDTTYPE

1492 (5D4) UNSIGNED 2 SPGDUNIT  
1494 (5D6) UNSIGNED 2 SPGRSIZE

1496 (5D8) UNSIGNED 2 SPGBSIZE  
1498 (5DA) CHARACTER 2 SPGFRMAT

VERSION NUMBER1 AND VERSION NUMBER2 FOR VERSIONING  
NEWVERSION NUMBER1 AND NEWVERSION NUMBER2 FOR COPY NEWVERSIONS

1500 (5DC) CHARACTER 6 SPGVRSNS

1500 (5DC) UNSIGNED 2 SPGVERS1  
1502 (5DE) UNSIGNED 2 SPGVERS2

1504 (5E0) UNSIGNED 2 SPGNVRS1  
1506 (5E2) UNSIGNED 2 WAS SPGCISZE

YES|NO PARAMETERS AND VALUE-FOUND BITS

1508 (5E4) BITSTRING 18 SPGB00L

REPLACE|NOREPLACE IN COPY, IMPORT  
1 --- 0 INITIAL= 0

1... .. SPGREPL

LOCK|UNLOCK IN ALTER/LIST DOCUMENT AND ALTER USER COMMANDS  
1 --- 0 INITIAL= 0

| OFFSETS | TYPE | LENGTH | NAME | DESCRIPTION |
|---------|------|--------|------|-------------|
|---------|------|--------|------|-------------|

|            |  |  |         |  |
|------------|--|--|---------|--|
| .1... .... |  |  | SPGLOCK |  |
|------------|--|--|---------|--|

|  |  |  |            |  |
|--|--|--|------------|--|
| ADM   NOADM (PRIVILEGE) IN DEF/ALT USER<br>1-----0 |  |  | INITIAL= 0 |  |
|--|--|--|------------|--|

|           |  |  |         |  |
|-----------|--|--|---------|--|
| ..1. .... |  |  | SPGUADM |  |
|-----------|--|--|---------|--|

|   |  |  |  |  |
|---|--|--|--|--|
| COPY-FLAG: 1 IF ':MEM2' GIVEN<br>INITIAL= 0 |  |  |  |  |
|---|--|--|--|--|

|           |  |  |          |  |
|-----------|--|--|----------|--|
| ...1 .... |  |  | SPGCPYM2 |  |
|-----------|--|--|----------|--|

|   |  |  |  |  |
|---|--|--|--|--|
| ONE OF THE FOLLOWING BITS IS SET TO 1 WHENEVER A<br>VALUE IS PLACED IN A NON-BOOLEAN PARAMETER ABOVE.<br>A ZERO INDICATES THAT THE CORRESPONDING PARAMETER<br>WAS NOT SEEN IN SCANNING THE COMMAND. |  |  |  |  |
|---|--|--|--|--|

|            |          |                            |
|------------|----------|----------------------------|
| .... 1...  | SPGSAWFR | SAW 'FROM'                 |
| .... .1..  | SPGSAWTO | SAW 'TO'                   |
| .... ..1.  | SPGSAWN1 | VALUE IN SPGNAME1          |
| .... ...1  | SPGSAWN2 | VALUE IN SPGNAME2          |
| 1... ....  | SPGSAWU1 | VALUE IN SPGUSR1           |
| .1... .... | SPGSAWU2 | VALUE IN SPGUSR2           |
| ..1. ....  | SPGSAWU3 | VALUE IN SPGUSR3           |
| ...1 ....  | SPGSAWL1 | VALUE IN SPGLIB1           |
| .... 1...  | SPGSAWM1 | VALUE IN SPGMEMN1          |
| .... .1..  | SPGSAWL2 | VALUE IN SPGLIB2           |
| .... ..1.  | SPGSAWM2 | VALUE IN SPGMEMN2          |
| .... ...1  | SPGSAWP1 | VALUE IN SPGPASS1          |
| 1... ....  | SPGSAWP2 | VALUE IN SPGPASS2          |
| .1... .... | SPGSAWLT | SAW 'LIBTYPE()'            |
| ..1. ....  | SPGALTP1 | SAW 'PASSWD' 'NOPASSWD'    |
| ...1 ....  | SPGALTS1 | SAW 'SHARE NOSHARE'        |
| .... 1...  | SPGALTLK | SAW 'LOCK' 'UNLOCK'        |
| .... .1..  | SPGALTAD | SAW 'ADM' 'NOADM'          |
| .... ..1.  | SPGALTPJ | SAW 'PROJLIB'              |
| .... ...1  | SPGALTSP | SAW 'SUPRESS' 'NOSUPPRESS' |
| 1... ....  | SPGALTCL | SAW 'CONTROL' 'NOCONTROL'  |
| .1... .... | SPGALTRP | SAW 'REPLACE' 'NOREPLACE'  |
| ..1. ....  | SPGSAWSP | SAW 'SPACE(NNNNN)'         |
| ...1. .... | SPGSAWD1 | SAW DATE-1                 |
| .... 1...  | SPGSAWD2 | SAW DATE-1 AND DATE-2      |
| .... .1..  | SPGSAWD3 | SAW DATE-2 ONLY            |
| .... ..1.  | SPGSAWD4 | SAW DATED(DATE-1:) ONLY    |
| .... ...1  | SPGSAWSH | SAW SHARE                  |
| 1... ....  | SPGCLALL | SAW CLOSE ALL              |
| .1... .... | SPGSAWTD | SAW TO(DSN                 |
| ..1. ....  | SPGSAWFD | SAW FROM(DSN               |
| ...1 ....  | SPGSAWID | SAW ID                     |
| .... 1...  | SPGSAWON | SAW OWN                    |
| .... .1..  | SPGLABEL | SAW LABEL                  |

| OFFSETS | TYPE            | LENGTH | NAME     | DESCRIPTION                            |
|---------|-----------------|--------|----------|--|
| ....    | ..1.            |        | SPG2400  | SAW 2400                               |
| ....    | ...1            |        | SPGRWIND | SAW REWIND                             |
| 1...    | ....            |        | SPGUNLOD | SAW UNLOAD                             |
| ..1.    | ....            |        | SPGSAWFT | SAW FORMAT                             |
| ..1.    | ....            |        | SPGSAWTP | SAW TYPE                               |
| ...1    | ....            |        | SPGSAWUN | SAW UNIT                               |
| ....    | 1...            |        | SPGSAWBK | SAW BLOCKSIZE                          |
| ....    | ..1.            |        | SPGSAWRC | SAW RECORDSIZE                         |
| ....    | ...1            |        | SPGSAWDL | SAW DLBL   TLBL   FILENAME             |
| 1...    | ...1            |        | SPGFDMEM | SAW FROM DSN MEMBER NAME               |
| 1...    | ....            |        | SPGTDMEM | SAW TO DSN MEMBER NAME                 |
| ..1.    | ....            |        | SPGQALNM | TO DSN QUALIFICATION REQUIRE           |
| ..1.    | ....            |        | SPGSAWOA | SAW OWN ALTERNATE                      |
| ...1    | ....            |        | SPGFRQAL | FROM DSN QUALIF REQUIRED               |
| ....    | 1...            |        | SPGSCLS  | VALUE IN SPGCLASS                      |
| ....    | ..1.            |        | SPGSCLS  | VALUE IN SPGCLUST                      |
| ....    | ...1            |        | SPGSDATA | VALUE IN SPGDATA1                      |
| ....    | ...1            |        | SPGSAUXD | VALUE IN SPGAUXD                       |
| 1...    | ....            |        | SPGSSORC | VALUE IN SPGSOURC                      |
| ..1.    | ....            |        | SPGSHOST | VALUE IN SPGAHOST                      |
| ..1.    | ....            |        | SPGSEXID | VALUE IN SPGAEXID                      |
| ...1    | ....            |        | SPGSNHST | VALUE IN SPGNHOST                      |
| ....    | 1...            |        | SPGSNWID | VALUE IN SPGNEWID                      |
| ....    | ..1.            |        | SPGSNOTE | VALUE IN SPGANOTE                      |
| ....    | ...1            |        | SPGSDOCI | VALUE IN SPGDOCIF                      |
| ....    | ...1            |        | SPGSAPPL | VALUE IN SPGAPPIF                      |
| 1...    | ....            |        | SPGSPARM | VALUE IN SPGPARM                       |
| ..1.    | ....            |        | SPGALTIO | SAW INPUT OUTPUT                       |
| ..1.    | ....            |        | SPGINOUT | 0 - INPUT   1 - OUTPUT                 |
| ...1    | ....            |        | SPGSLSPC | SAW SPACE NOSPACE                      |
| ....    | 1...            |        | SPGLSPAC | 0 - NOSPACE   1 - SPACE                |
| ....    | ..1.            |        | SPGPROCR | PROCESSOR LOADED (NOT SYNTAX VARIABLE) |
| ....    | ...1            |        | SPGSVRN1 | SAW VERSION-1                          |
| ....    | ...1            |        | SPGSVRN2 | SAW VERSION-2                          |
| 1...    | ....            |        | SPGSVRNA | SAW 'VERSION(ALL)'                     |
| ..1.    | ....            |        | SPGSVERS | SAW 'VERSION' ONLY                     |
| ..1.    | ....            |        | SPGSNVR1 | SAW NEWVERSION-1                       |
| ...1    | ....            |        | SPGSNVR3 | SAW 'NEWVERSION' ONLY                  |
| ....    | 1...            |        | SPGSCONL | SAW 'CONTROL' 'NOCONTROL'              |
| ....    | ..1.            |        | SPGSREPL | SAW 'REPLACE' 'NOREPLACE'              |
| ....    | ...1            |        | SPGSWALL | SAW 'ALL'                              |
| ....    | ...1            |        | SPGSSTAR | SAW 'DD(*)' OR 'FROM(*)'               |
| 1...    | ....            |        | SPGSOPER | SAW 'OPERATOR'                         |
| ..1.    | ....            |        | SPGSHX70 | SAW HEX X'70'                          |
| ..1.    | ....            |        | SPGSATTR | SAW 'NOATTRIBUTE'                      |
| ...1    | ....            |        | SPGSSUPP | SAW 'SUPPRESS' 'NOSUPPRESS'            |
| ....    | 1...            |        | SPGSARCH | SAW 'ARCHIVE'                          |
| ....    | ..1.            |        | SPGSINFO | SAW 'INFO'                             |
| ....    | ...1            |        |          | UNUSED RESERVED                        |
| ....    | ...1            |        | SPGISSRC | SAW INSOURCE PARAMETER                 |
| 1...    | ....            |        |          | UNUSED RESERVED                        |
| ..1.    | ....            |        | SPGALTP2 | SAW NOPASS KEYWORD                     |
| ..1.    | ....            |        | SPGSDSN2 | SAW SECOND DATA SET NAME               |
| ...1    | ....            |        | SPGSDSM2 | SAW SECOND DS MEMBER NAME              |
| ....    | 1111            |        |          |  |
| 1111    | 1111            |        |          |  |
| 1111    | 1111            |        |          |  |
| 1111    | 1111            |        |          |  |
| 1111    | 1111            |        |          |  |
| 1111    | 1111            |        |          |  |
| 1111    | 1111            |        |          |  |
| 1526    | (5F6) CHARACTER | 2      |          | RESERVED<br>UNUSED RESERVED            |

ADDRESS OF 16K FILE CONTROL INTERVAL BUFFER

|      |               |   |         |                   |
|------|---------------|---|---------|-------------------|
| 1528 | (5F8) ADDRESS | 4 | SPG16KB | ADDR(SPG16K)      |
| 1532 | (5FC) ADDRESS | 4 | SPGFDRB | ADDR(SPG16K) + 4K |

DCB EXIT LISTS



| OFFSETS | TYPE | LENGTH | NAME | DESCRIPTION |
|---------|------|--------|------|-------------|
|---------|------|--------|------|-------------|

|      |       |           |   |          |                          |
|------|-------|-----------|---|----------|--------------------------|
| 1536 | (600) | CHARACTER | 8 | SPGEXLST | DCB EXIT LIST            |
| 1536 | (600) | CHARACTER | 1 | SPGJFCB  | JFCB CODE, (X'07')       |
| 1537 | (601) | ADDRESS   | 3 | SPGJFCBA | JFCB AREA ADDRESS        |
| 1540 | (604) | CHARACTER | 1 | SPGDCBE  | EXIT RTN CODE (X'85')    |
| 1541 | (605) | ADDRESS   | 3 | SPGDCBEA | DCB EXIT RTN ADDRESS     |
| 1544 | (608) | UNSIGNED  | 8 | SPGEOD   | DCB EODAD BRANCH CODE    |
| 1552 | (610) | ADDRESS   | 4 | SPGEODAD | ADDR OF EODAD SUBROUTINE |
| 1556 | (614) | CHARACTER | 4 |          | UNUSED RESERVED          |

#### DIRECTORY AND ACCOUNTING ACB'S FOR THE VSAM DATA SETS

|      |       |           |    |          |                      |
|------|-------|-----------|----|----------|----------------------|
| 1560 | (618) | CHARACTER | 80 | SPGIDACB | INPUT DIRECTORY ACB  |
| 1640 | (668) | CHARACTER | 80 | SPGODACB | OUTPUT DIRECTORY ACB |
| 1720 | (6B8) | CHARACTER | 80 |          | UNUSED-RESERVED      |

#### MODEL ESDS ACBS

|      |       |           |    |          |                       |
|------|-------|-----------|----|----------|-----------------------|
| 1800 | (708) | CHARACTER | 80 | SPGMIACB | MODEL INPUT ESDS ACB  |
| 1880 | (758) | CHARACTER | 80 | SPGMOACB | MODEL OUTPUT ESDS ACB |

| OFFSETS | TYPE | LENGTH | NAME | DESCRIPTION |
|---------|------|--------|------|-------------|
|---------|------|--------|------|-------------|

DCB'S FOR OS DATA SETS

|      |       |           |    |           |                     |
|------|-------|-----------|----|-----------|---------------------|
| 1960 | (7A8) | CHARACTER | 96 | SPGPRDCB  | DSMLIST DATA SET    |
| 2056 | (808) | CHARACTER | 96 | SPGRDDCB  | SEQUENTIAL DS INPUT |
| 2152 | (868) | CHARACTER | 96 | SPGNMDCB  | NAME LIST INPUT DS  |
| 2248 | (8C8) | CHARACTER | 88 | SPGWBD CB | SEQ DS OUTPUT BSAM  |
| 2336 | (920) | CHARACTER | 96 | SPGWQDCB  | SEQ DS OUTPUT QSAM  |

RPL'S FOR VSAM DATA SETS

|      |       |           |    |          |                       |
|------|-------|-----------|----|----------|-----------------------|
| 2432 | (980) | CHARACTER | 80 | SPGIRPL1 | INPUT DIRECTORY RPL 1 |
| 2512 | (9D0) | CHARACTER | 80 | SPGIRPL2 | INPUT DIRECTORY RPL 2 |
| 2592 | (A20) | CHARACTER | 80 | SPGIRPL3 | INPUT DIR RPL3        |
| 2672 | (A70) | CHARACTER | 80 | SPGIRPLX | 4K,16K DS INPUT RPL   |
| 2752 | (AC0) | CHARACTER | 80 | SPGORPL1 | OUTPUT DIR RPL 1      |
| 2832 | (B10) | CHARACTER | 80 | SPGORPL2 | OUTPUT DIR RPL 2      |
| 2912 | (B60) | CHARACTER | 80 | SPGORPL3 | OUTPUT DIR RPL3       |
| 2992 | (BB0) | CHARACTER | 80 | SPGORPLX | 4K,16K DS OUTPUT RPL  |
| 3072 | (C00) | CHARACTER | 80 |          | UNUSED-RESERVED       |

RBA ARG FLDS FOR SPGIRPLX AND SPGORPLX RPL'S

|      |       |          |   |          |                      |
|------|-------|----------|---|----------|----------------------|
| 3152 | (C50) | UNSIGNED | 4 | SPGIARGX | RBA ARG FLD:SPGIRPLX |
| 3156 | (C54) | UNSIGNED | 4 | SPGOARGX | RBA ARG FLD:SPGORPLX |

UNUSED - RESERVED AREA

|      |       |           |    |  |                  |
|------|-------|-----------|----|--|------------------|
| 3160 | (C58) | CHARACTER | 48 |  | UNUSED RESERVED) |
|------|-------|-----------|----|--|------------------|

ADDR OF WORKING 16K BUFFER FOR DLF OUTPUT LIB ESDS DATASETS

|      |       |         |   |          |  |
|------|-------|---------|---|----------|--|
| 3208 | (C88) | ADDRESS | 4 | SPG016KB | ADDR(WORK 16K BUFFER FOR OUTPUT ESDS DATASETS) |
|------|-------|---------|---|----------|--|

THE LENGTH OF FOLLOWING RESERVED FIELDS MUST BE SUCH THAT  
SPG16K FALLS AT X'1000' INTO THE SPG

| OFFSETS  | TYPE             | LENGTH | NAME     | DESCRIPTION                                      |
|--|------------------|--------|----------|--|
| 3212   | (C8C) CHARACTER  | 880    |          | RESERVED   |
| 3212   | (C8C) CHARACTER  | 380    |          | RESERVED BY DEV                                  |
| 3592   | (E08) CHARACTER  | 500    | SPGPATCH | F.E. PATCH AREA                                  |
| <div>16K BUFFER AREA. IN GENERAL - USED FOR INPUT CI BUFFER<br/>NOTE: BUFFER MUST BE AT END OF SPG TABLE</div> |                  |        |          |  |
| 4092   | (FFC) CHARACTER  | 16388  | SPGFVBT  | VARIABLE BLOCKED FILE BUFFER                     |
| 4092   | (FFC) UNSIGNED   | 4      | SPGFRDW  | RECORD DESCR WORD                                |
| 4092   | (FFC) SIGNED     | 2      | SPGFRLEN | RECORD DATA LENGTH                               |
| 4094   | (FFE) UNSIGNED   | 2      | SPGFR00  | RESERVED   |
| 4096   | (1000) CHARACTER | 16384  | SPG16K   |  |
| 4096   | (1000) UNSIGNED  | 16384  | DUMMY    | DUMMY ARRAY TO GET 16K SET ASIDE BY THE COMPILER |
| <div>END OF SPG TABLE</div>  |                  |        |          |  |
| 20480  | (5000) CHARACTER | 0      | SPGEND   | END OF SPG TABLE                                 |
| 20480  | (5000) CHARACTER | 0      | SPGLEN   | LENGTH OF SPG TABLE                              |

#### CROSS-REFERENCE

|          |      |          |      |    |
|----------|------|----------|------|----|
| DUMMY    | 1000 | SPGCLOFP | 304  |    |
| SPG      | 0    | SPGCLOG  | 11   | 01 |
| SPGAEXID | 4D4  | SPGCLONM | 2F4  |    |
| SPGAHOST | 4D0  | SPGCLUST | 4B8  |    |
| SPGALTAD | 5E6  | SPGCOMM  | 338  |    |
| SPGALTCL | 5E7  | SPGCPYM2 | 5E4  | 10 |
| SPGALTIO | 5EC  | SPGDACSP | 2FC  | 08 |
| SPGALTLK | 5E6  | SPGDATA1 | 4A0  |    |
| SPGALTPJ | 5E6  | SPGDATF1 | 34C  |    |
| SPGALTP1 | 5E6  | SPGDATF2 | 354  |    |
| SPGALTP2 | 5EF  | SPGDAT1  | 348  |    |
| SPGALTRP | 5E7  | SPGDAT2  | 350  |    |
| SPGALTSP | 5E6  | SPGDCBE  | 604  |    |
| SPGALTS1 | 5E6  | SPGDCBEA | 605  |    |
| SPGANOTE | 4E8  | SPGDI    | 10   | 80 |
| SPGAPPIF | 546  | SPGDIBUF | C4   |    |
| SPGAUTHF | 3B   | SPGDO    | 10   | 40 |
| SPGAUXD  | 4A8  | SPGDOBUF | 1D0  |    |
| SPGB00L  | 5E4  | SPGDOCIF | 514  |    |
| SPGBSIZE | 5D8  | SPGDORDW | 1CC  |    |
| SPGCALED | 314  | SPGD0VBT | 1CC  |    |
| SPGCALL1 | 314  | SPGDR    | C0   | 01 |
| SPGCALL2 | 318  | SPGDRLEN | 1CC  |    |
| SPGCALL3 | 31C  | SPGDROO  | 1CE  |    |
| SPGCALL4 | 320  | SPGDSME2 | 434  |    |
| SPGCALL5 | 324  | SPGDSNM2 | 3D8  |    |
| SPGCALL6 | 328  | SPGDTYPE | 5D2  |    |
| SPGCALR1 | 310  | SPGDUNIT | 5D4  |    |
| SPGCCLO  | 2FC  | SPGEND   | 5000 |    |
| SPGCIANY | 1C8  | SPGENQFL | 4C   |    |
| SPGCI8EG | 1C8  | SPGENQS  | 4C   | 40 |
| SPGCIEND | 1C8  | SPGENQX  | 4C   | 80 |
| SPGCLALL | 5E8  | SPGENV   | 5D2  |    |
| SPGCLASS | 4B0  | SPGEOD   | 608  |    |
| SPGCLBIA | 308  | SPGEODAD | 610  |    |
| SPGCLBOA | 30C  | SPGEXEC  | 334  |    |
| SPGCLFLG | 2FC  | SPGEXEQD | 314  | 80 |
| SPGCLIFP | 300  | SPGEXLST | 600  |    |
| SPGCLINM | 2EC  | SPGEXSTI | 47   | 80 |

|          |      |    |           |     |    |
|----------|------|----|-----------|-----|----|
| SPGEXSTO | 4B   | 80 | SPGNHOST  | 4D2 |    |
| SPGFDMEM | 5E9  | 01 | SPGNM     | 11  | 04 |
| SPGFDRB  | 5FC  |    | SPGNMDCB  | 868 |    |
| SPGFLNM1 | 478  |    | SPGNVRS1  | 5E0 |    |
| SPGFLNM2 | 48C  |    | SPGOACSS  | 2D1 |    |
| SPGFRDSN | 368  |    | SPGOAREA  | 2DC |    |
| SPGFRDW  | FFC  |    | SPGOARG   | 2D8 |    |
| SPGFRLN  | FFC  |    | SPGOARGX  | C54 |    |
| SPGFRMAT | 5DA  |    | SPGOCLEBI | 2FC | 20 |
| SPGFRMDD | 358  |    | SPGOCLO   | 2FC | 01 |
| SPGFRMEM | 394  |    | SPGOCLEST | 2FC | 40 |
| SPGFRQAL | 5EA  | 10 | SPGODACB  | 668 |    |
| SPGFR00  | FFE  |    | SPGODFC   | A8  |    |
| SPGFVBT  | FFC  |    | SPGODFD   | A0  |    |
| SPGGR    | C0   | 20 | SPGODFL   | B8  |    |
| SPGSHAD  | 3C   |    | SPGODFS   | B0  |    |
| SPGIACSS | 2E1  |    | SPGODIR   | 2D0 | 08 |
| SPGIAREA | 2E8  |    | SPGOESDS  | 2D0 | 40 |
| SPGIARG  | 2E4  |    | SPGOFKS   | 2D0 | 10 |
| SPGIARGX | C50  |    | SPGOINS   | 2D0 | 04 |
| SPGICLO  | 2FC  | 02 | SPGOKEQ   | 2D0 | 20 |
| SPGICLST | 2FC  | 80 | SPGOKEYL  | 2D2 |    |
| SPGIDACB | 618  |    | SPGOKSDS  | 2D0 | 80 |
| SPGIDCBP | 14   |    | SPGOLUI   | 2D0 |    |
| SPGIDFC  | 88   |    | SPGOPENA  | 10  |    |
| SPGIDFD  | 80   |    | SPGOPEND  | 11  |    |
| SPGIDFL  | 98   |    | SPGOPRTC  | 12  |    |
| SPGIDFS  | 90   |    | SPGORACL  | 77  |    |
| SPGIDIR  | 2E0  | 10 | SPGORACN  | 78  |    |
| SPGIESDS | 2E0  | 80 | SPGORECL  | 2D4 |    |
| SPGIFKS  | 2E0  | 20 | SPGORNAM  | 54  |    |
| SPGIKEQ  | 2E0  | 40 | SPGORPLX  | BB0 |    |
| SPGIKEYL | 2E2  |    | SPGORPL1  | AC0 |    |
| SPGILUI  | 2E0  |    | SPGORPL2  | B10 |    |
| SPGINOUT | 5EC  | 20 | SPGORPL3  | B60 |    |
| SPGINSRC | 4C8  |    | SPG016KB  | C88 |    |
| SPGIPENT | 1C4  |    | SPGPARM   | 592 |    |
| SPGIRACL | 6D   |    | SPGPARMA  | 340 |    |
| SPGIRACN | 6E   |    | SPGPASS1  | 454 |    |
| SPGIRNAM | 60   |    | SPGPASS2  | 45C |    |
| SPGIRPLX | A70  |    | SPGPATCH  | E08 |    |
| SPGIRPL1 | 980  |    | SPGPRDCB  | 7A8 |    |
| SPGIRPL2 | 9D0  |    | SPGPRIVI  | 47  |    |
| SPGIRPL3 | A20  |    | SPGPRIVO  | 4B  |    |
| SPGISTMP | 0    |    | SPGPRJCI  | 47  | 10 |
| SPGIUPD  | 2E0  | 08 | SPGPRJCO  | 4B  | 10 |
| SPGJFCB  | 600  |    | SPGPROCR  | 5EC | 04 |
| SPGJFCBA | 601  |    | SPGPROJI  | 44  |    |
| SPGLABEL | 5E8  | 04 | SPGPROJO  | 48  |    |
| SPGLBIG  | C2   |    | SPGPRSTN  | 10  | 20 |
| SPGLBII  | C0   |    | SPGQALNM  | 5EA | 40 |
| SPGLBOI  | 1C8  |    | SPGRACFI  | 3B  | 80 |
| SPGLEN   | 5000 |    | SPGRACFO  | 3B  | 40 |
| SPGLIBTC | 474  |    | SPGRD     | 11  | 20 |
| SPGLIBTY | 474  |    | SPGRDDCB  | 808 |    |
| SPGLIB1  | 478  |    | SPGREPL   | 5E4 | 80 |
| SPGLIB2  | 48C  |    | SPGREQ    | C0  | 80 |
| SPGLIST  | 334  | 80 | SPGRETM   | 32C |    |
| SPGLOCK  | 5E4  | 40 | SPGRETV   | 330 |    |
| SPGLSPAC | 5EC  | 08 | SPGRFI    | C0  | 02 |
| SPGMEMN1 | 47B  |    | SPGRGE    | C0  | 40 |
| SPGMEMN2 | 48F  |    | SPGRPL2   | C1  | 80 |
| SPGMIACB | 708  |    | SPGRS     | C0  | 10 |
| SPGMOACB | 758  |    | SPGRSIZE  | 5D6 |    |
| SPGMSGD  | 18   |    | SPGRU     | C0  | 08 |
| SPGMSGDD | 2B   |    | SPGRWIND  | 5E8 | 01 |
| SPGMSGEX | 1B   |    | SPGSAPPL  | 5EB | 01 |
| SPGMSGFL | 1B   |    | SPGSARCH  | 5EE | 08 |
| SPGMSGHO | 25   |    | SPGSATTR  | 5EE | 20 |
| SPGMSGLB | 18   |    | SPGSAUXD  | 5EA | 01 |
| SPGMSGOP | 2B   |    | SPGSAWBK  | 5E9 | 08 |
| SPGMSGRC | 36   |    | SPGSAWDL  | 5E9 | 02 |
| SPGMSGRS | 18   |    | SPGSAWD1  | 5E7 | 10 |
| SPGMSGVV | 36   |    | SPGSAWD2  | 5E7 | 08 |
| SPGNAME1 | 444  |    | SPGSAWD3  | 5E7 | 04 |
| SPGNAME2 | 44C  |    | SPGSAWD4  | 5E7 | 02 |
| SPGNEWID | 4DE  |    | SPGSAWFD  | 5E8 | 20 |

|           |     |    |           |      |    |
|-----------|-----|----|-----------|------|----|
| SPGSAWFR  | 5E4 | 08 | SPGSOURC  | 4C0  |    |
| SPGSAWFT  | 5E9 | 40 | SPGSPACN  | 470  |    |
| SPGSAWID  | 5E8 | 10 | SPGSPARM  | 5EC  | 80 |
| SPGSAWLT  | 5E6 | 40 | SPGSPR    | 11   | 40 |
| SPGSAWL1  | 5E5 | 10 | SPGSPRVI  | 47   | 20 |
| SPGSAWL2  | 5E5 | 04 | SPGSPRVO  | 4B   | 20 |
| SPGSAWM1  | 5E5 | 08 | SPGSREPL  | 5ED  | 04 |
| SPGSAWM2  | 5E5 | 02 | SPGSSORC  | 5EB  | 80 |
| SPGSAWN1  | 5E4 | 02 | SPGSSTAR  | 5ED  | 01 |
| SPGSAWN2  | 5E4 | 01 | SPGSSUPP  | 5EE  | 10 |
| SPGSAWOA  | 5EA | 20 | SPGSUBC   | 33A  |    |
| SPGSAWON  | 5E8 | 08 | SPGSVERS  | 5ED  | 40 |
| SPGSAWP1  | 5E5 | 01 | SPGSVRNA  | 5ED  | 80 |
| SPGSAWP2  | 5E6 | 80 | SPGSVRN1  | 5EC  | 02 |
| SPGSAWRC  | 5E9 | 04 | SPGSVRN2  | 5EC  | 01 |
| SPGSAWSH  | 5E7 | 01 | SPGSWALL  | 5ED  | 02 |
| SPGSAWSP  | 5E7 | 20 | SPGSYEOB  | 11   | 02 |
| SPGSAWTD  | 5E8 | 40 | SPGTDMM   | 5EA  | 80 |
| SPGSAWTO  | 5E4 | 04 | SPGTDMDIF | 340  |    |
| SPGSAWTP  | 5E9 | 20 | SPGTODD   | 360  |    |
| SPGSAWUN  | 5E9 | 10 | SPGTODSN  | 39C  |    |
| SPGSAWU1  | 5E5 | 80 | SPGTOMEM  | 3C8  |    |
| SPGSAWU2  | 5E5 | 40 | SPGTSTMP  | 8    |    |
| SPGSAWU3  | 5E5 | 20 | SPGUADM   | 5E4  | 20 |
| SPGSCLESS | 5EA | 08 | SPGUNLOD  | 5E9  | 80 |
| SPGSCLEST | 5EA | 04 | SPGUPRRD  | 1C8  | 20 |
| SPGSCONL  | 5ED | 08 | SPGUPRWT  | 1C8  | 10 |
| SPGSDATA  | 5EA | 02 | SPGUSID   | 38   |    |
| SPGSDOCI  | 5EB | 02 | SPGUSR1   | 465  |    |
| SPGSDSM2  | 5EF | 10 | SPGUSR2   | 469  |    |
| SPGSDSN2  | 5EF | 20 | SPGUSR3   | 46D  |    |
| SPGSEXID  | 5EB | 20 | SPGUXIAD  | 40   |    |
| SPGSHOST  | 5EB | 40 | SPGVERS1  | 5DC  |    |
| SPGSHX70  | 5EE | 40 | SPGVERS2  | 5DE  |    |
| SPGSICSP  | 33C |    | SPGVRSNS  | 5DC  |    |
| SPGSIN    | 11  | 80 | SPGWB     | 11   | 08 |
| SPGSINFO  | 5EE | 04 | SPGWBDDB  | 8C8  |    |
| SPGSISRC  | 5EE | 01 | SPGWD     | 1C8  | 80 |
| SPGSLSPC  | 5EC | 10 | SPGWFI    | 1C8  | 40 |
| SPGSNHST  | 5EB | 10 | SPGWFSH   | 2FC  | 10 |
| SPGSNOTE  | 5EB | 04 | SPGWQ     | 11   | 10 |
| SPGSNVR1  | 5ED | 20 | SPGWQDCB  | 920  |    |
| SPGSNVR3  | 5ED | 10 | SPG16K    | 1000 |    |
| SPGSNWID  | 5EB | 08 | SPG16KB   | 5F8  |    |
| SPGSOPER  | 5EE | 80 | SPG2400   | 5E8  | 02 |

## PARAMETER LISTS

The primary parameter lists used in DLF processing for communication between various DLF modules are:

- User Exit Interface Parameter List (EXI)
- Special Processor Parameter List (IPP)
- Source Input Parameter List (SIC).

The remainder of this section is devoted to illustrating the structure of each of these parameter lists.

## User Exit Interface (EXI) Parameter List

This interface parameter list contains fields used for communication between DLF modules and the user exit routines (DSMSPUXI). Various application programs exit to the user routines (through the User Exit Controller - DSMSPEXI) at points during processing where a user may want to check or modify data. The modules that call DSMSPEXI for this purpose are:

- DSMSPARC — ARCHIVE Commands
- DSMSPDAM — DEFINE/ALTER MAP
- DSMSPDAU — DEFINE/ALTER USER
- DSMSPEXP — EXPORT
- DSMSPIMP — IMPORT
- DSMSPUD — PURGE DOCUMENT
- DSMSPUM — PURGE MAP
- DSMSPUU — PURGE USER
- DSMSPRET — RETRIEVE Commands.

The modules listed above make various records available to and communicate with the user exit by way of the interface parameter list.

### EXI

| OFFSETS | TYPE            | LENGTH | NAME     | DESCRIPTION                  |
|---------|-----------------|--------|----------|------------------------------|
| 0       | (0) STRUCTURE   | 402    | EXI      |                              |
| 0       | (0) SIGNED      | 2      | EXIRETRN | USER EXIT RETURN CODE        |
| 2       | (2) SIGNED      | 2      | EXIRESN  | USER EXIT REASON CODE        |
| 4       | (4) CHARACTER   | 125    | EXIMSG   | USER MESSAGE AREA            |
| 4       | (4) UNSIGNED    | 2      | EXIMSGLN | MESSAGE UNIT LENGTH          |
| 6       | (6) UNSIGNED    | 2      |          | UNUSED RESERVED              |
| 8       | (8) CHARACTER   | 1      | EXIMSGCL | MESSAGE CONTROL CHARACTER    |
| 9       | (9) CHARACTER   | 120    | EXIMSGTX | MESSAGE TEXT                 |
| 129     | (81) BITSTRING  | 1      | EXIFLAGS | DSMSPEXI INTERFACE FLAGS     |
|         | 1... ..         |        | EXIMSGON | MESSAGE REQUEST BIT. 1 = YES |
|         | .1.. ....       |        | EXIDERIN | A DER IS IN THE INTERFACE    |
|         | ..1. ....       |        | EXIUPRIN | A UPR IS IN THE INTERFACE    |
|         | ...1 ....       |        | EXIUMRIN | A UMR IS IN THE INTERFACE    |
|         | .... 1...       |        | EXIAPPLX | APPLICATION DATA CHANGED     |
|         | .... .111       |        |          | UNUSED RESERVED              |
| 130     | (82) CHARACTER  | 8      | EXICOMND | EBCDIC COMMAND NAME          |
| 138     | (8A) CHARACTER  | 8      | EXICMOBJ | EBCDIC COMMAND OBJECT        |
| 146     | (92) CHARACTER  | 256    | EXIDER   | COPY OF THE DER              |
| 146     | (92) CHARACTER  | 120    | EXIUPR   | COPY OF THE UPR              |
| 266     | (10A) CHARACTER | 60     | EXIUMR   | COPY OF THE UMR              |

**CROSS-REFERENCE**

|          |    |    |          |     |    |
|----------|----|----|----------|-----|----|
| EXI      | 0  |    | EXIMSGLN | 4   |    |
| EXIAPPLX | 81 | 08 | EXIMSGON | 81  | 80 |
| EXICMOBJ | 8A |    | EXIMSGTX | 9   |    |
| EXICOMND | 82 |    | EXIRESON | 2   |    |
| EXIDER   | 92 |    | EXIRETRN | 0   |    |
| EXIDERIN | 81 | 40 | EXIUMR   | 10A |    |
| EXIFLAGS | 81 |    | EXIUMRIN | 81  | 10 |
| EXIMSG   | 4  |    | EXIUPR   | 92  |    |
| EXIMSGCL | 8  |    | EXIUPRIN | 81  | 20 |



## Special Processor Parameter List (IPP)

This parameter list is used as an interface between DLF modules and special processors not provided by the DLF product (i.e., DSMACAIP which is packaged with DCF but runs as a processor under DLF). The special processor parameter list is used in three DLF modules: DSMSPDSI, DSMSPDSO, and DSMSPGNX.

### IPP

| OFFSETS | TYPE           | LENGTH | NAME      | DESCRIPTION                 |
|---------|----------------|--------|-----------|-----------------------------|
| 0       | (0) STRUCTURE  | 124    | IPP       |                             |
| 0       | (0) BITSTRING  | 2      | IPPFLG    | FLAG BITS                   |
|         | 1... ..        |        | IPPEOFS   | EOF ON SOURCE               |
|         | .1.. ..        |        | IPPEOFP   | END OF PROCESSING FROM IP   |
|         | ..1. ....      |        | IPPINIT   | IP MUST INITIALIZE          |
|         | ...1 ....      |        | IPPMMSGR  | IP REQUESTS PRINT MESSAGE   |
|         | .... 1...      |        | IPPPARF   | PROCESSOR PARMS PRESENT     |
|         | .... .1..      |        | IPPNOIP   | SUPPRESS INPUT TO PROCESSOR |
|         | .... ..11      |        |           |                             |
|         | 1111 1111      |        |           | RESERVED                    |
| 2       | (2) SIGNED     | 2      |           | RESERVED                    |
| 4       | (4) ADDRESS    | 4      | IPPMMSGAD | IP MESSAGE ADDRESS PTR      |
| 8       | (8) CHARACTER  | 8      | IPPDATOA  | OUTPUT DATA ATTRIBUTE       |
| 16      | (10) CHARACTER | 8      | IPPDATIA  | INPUT DATA ATTRIBUTE        |
| 24      | (18) CHARACTER | 8      | IPPCCLASS | CLASS ATTRIBUTE             |
| 32      | (20) CHARACTER | 8      | IPPISRCE  | INPUT SOURCE ATTRIBUTE      |
| 40      | (28) CHARACTER | 8      | IPPOSRC   | OUTPUT SOURCE ATTRIBUTE     |
| 48      | (30) CHARACTER | 64     | IPPPARM   | PROCESSOR PARAMETERS        |
| 112     | (70) SIGNED    | 2      | IPPCMBLN  | COMMUNICATION BUFFER LENGTH |
| 114     | (72) SIGNED    | 2      | IPPWKBLN  | WORK BUFFER LENGTH          |
| 116     | (74) SIGNED    | 4      | IPPRTNCD  | RETURN CODE                 |
| 120     | (78) SIGNED    | 4      | IPPRSCD   | REASON CODE                 |

### CROSS-REFERENCE

|           |    |           |    |    |
|-----------|----|-----------|----|----|
| IPP       | 0  | IPPMMSGAD | 4  |    |
| IPPCCLASS | 18 | IPPMMSGR  | 0  | 10 |
| IPPCMBLN  | 70 | IPPNOIP   | 0  | 04 |
| IPPDATIA  | 10 | IPPOSRC   | 28 |    |
| IPPDATOA  | 8  | IPPPARF   | 0  | 08 |
| IPPEOFP   | 0  | IPPPARM   | 30 |    |
| IPPEOFS   | 0  | IPPRSCD   | 78 |    |
| IPPFLG    | 0  | IPPRTNCD  | 74 |    |
| IPPINIT   | 0  | IPPWKBLN  | 72 |    |
| IPPISRCE  | 20 |           |    |    |

## Source Input Parameter List (SIC)

In general, the source input parameter list (SIC) is used to pass information to and from the DLF source input module (DSMSPSIC). DSMSPSIC, however, may pass the address of this parameter list to DSMSPGNX that in turn may pass it to DSMSPSOP. Therefore, the information in the parameter list is available not only to DSMSPSIC and the modules which call it, but also to the modules that provide sub-services to DSMSPSIC. The actual parameter list is always located within the module that calls DSMSPSIC.

The DLF modules that call DSMSPSIC are:

- DSMSPCLO (to close source input files after READ commands)
- DSMSPEXC (to close source input files at DLF termination)
- DSMSPRED (to read (or open via DSMSPSOP) source input records)
- DSMSPTRM (to close source input files when the command ends).

## NOTES

1. When the Document Composition Facility is also installed and runs under DLF, it is passed the address of DSMSPSIC and uses the SIC parameter list for communicating its source input requirements.
2. DSMSPSIC passes the address of the SIC parameter list to DSMSPGNX (get next record sub-service) as a call argument and DSMSPGNX passes the address to DSMSPSOP (source input open sub-services) as a call argument.

The format of the SIC parameter list is shown below.

## SICPARML

| OFFSETS   | TYPE           | LENGTH | NAME     | DESCRIPTION         |
|-----------|----------------|--------|----------|---------------------|
| 0         | (0) STRUCTURE  | 236    | SICPARML | SIC PARAMETER LIST  |
| 0         | (0) BITSTRING  | 1      | SICFLG   | FLAG BYTE           |
| 1...      | ....           |        | SICCLA   | CLOSE ALL FILES     |
| .1...     | ....           |        | SICCLS   | CLOSE 1 FILE        |
| ..1...    | ....           |        | SICGNX   | GET NEXT RECORD     |
| ...1...   | ....           |        | SICPNT   | POINT TO RECORD     |
| ....1...  | ....           |        | SICNTE   | NOTE THIS RECORD    |
| ....1..   | ....           |        | SICEOF   | EOF ON SOURCE       |
| ......1   | ....           |        | SICNRD   | OPEN, NO READ       |
| .......1  | ....           |        | SICRAS   | REL ALL DYN SPACE   |
| 1         | (1) BITSTRING  | 1      | SICFLG2  | SECOND FLAG BYTE    |
| 1...      | ....           |        | SICSVERN | SAW VERSION KEYWORD |
| .1...     | ....           |        | SICSDATA | SAW DATA KEYWORD    |
| ..1...    | ....           |        | SICSZERO | SAW ZERO LIBNUM     |
| ...1 1111 | ....           |        |          | NOT USED            |
| 2         | (2) CHARACTER  | 3      |          | RESERVED            |
| 5         | (5) UNSIGNED   | 3      | SICSRLIB | SEARCH LIB NUMBER   |
| 8         | (8) SIGNED     | 4      | SICGMLIB | GML LIBRARY NUMBER  |
| 12        | (C) CHARACTER  | 1      |          | UNUSED RESERVED     |
| 13        | (D) UNSIGNED   | 3      | SICLIBRN | SRC LIB NUMBER      |
| 16        | (10) CHARACTER | 16     | SICDOCN  | DOCUMENT NAME       |
| 32        | (20) UNSIGNED  | 2      | SICVERS  | VERSION NUMBER      |
| 34        | (22) CHARACTER | 2      |          | RESERVED            |
| 36        | (24) CHARACTER | 8      | SICDOCP  | DOCUMENT PASSWORD   |

| OFFSETS | TYPE           | LENGTH | NAME     | DESCRIPTION         |
|---------|----------------|--------|----------|---------------------|
| 44      | (2C) CHARACTER | 8      | SICFRDD  | FROM DDNAME         |
| 52      | (34) CHARACTER | 8      | SICFRPDS | FROM PDS MEM NAME   |
| 60      | (3C) CHARACTER | 44     | SICFRDSN | FROM BATCH FILE DSN |
| 104     | (68) CHARACTER | 8      | SICFRQAL | DSN QUALIFIER       |
| 112     | (70) CHARACTER | 8      | SICSRCD  | INPUT DATA TYPE     |
| 120     | (78) CHARACTER | 8      | SICOUTDA | OUTPUT DATA TYPE    |
| 128     | (80) CHARACTER | 8      | SICIPNAM | INPUT PROCESSOR NAM |
| 136     | (88) ADDRESS   | 4      | SICCMREC | SLE RECORD ADDRESS  |
| 140     | (8C) SIGNED    | 4      | SICNPREC | NOTE/POINT REC NUM  |
| 144     | (90) SIGNED    | 4      | SICRTNCD | ERROR RETURN CODE   |
| 148     | (94) SIGNED    | 4      | SICREACD | ERROR REASON CODE   |
| 152     | (98) UNSIGNED  | 4      |          | RESERVED            |
| 156     | (9C) CHARACTER | 64     | SICPARM  | PROCESSOR PARAMETER |
| 220     | (DC) CHARACTER | 8      | SICINSOR | INSOURCE PARAMETER  |
| 228     | (E4) CHARACTER | 8      | SICOUTSC | OUTSOURCE PARAMETER |

#### CROSS-REFERENCE

|          |    |    |          |    |    |
|----------|----|----|----------|----|----|
| SICCLA   | 0  | 80 | SICNPREC | 8C |    |
| SICCLS   | 0  | 40 | SICNRD   | 0  | 02 |
| SICCMREC | 88 |    | SICNTE   | 0  | 08 |
| SICDOCN  | 10 |    | SICOUTDA | 78 |    |
| SICDOCP  | 24 |    | SICOUTSC | E4 |    |
| SICEOF   | 0  | 04 | SICPARM  | 9C |    |
| SICFLG   | 0  |    | SICPARML | 0  |    |
| SICFLG2  | 1  |    | SICPNT   | 0  | 10 |
| SICFRDD  | 2C |    | SICRAS   | 0  | 01 |
| SICFRDSN | 3C |    | SICREACD | 94 |    |
| SICFRPDS | 34 |    | SICRTNCD | 90 |    |
| SICFRQAL | 68 |    | SICSDATA | 1  | 40 |
| SICGMLIB | 8  |    | SICSRCD  | 70 |    |
| SICGNX   | 0  | 20 | SICSRLIB | 5  |    |
| SICINSOR | DC |    | SICSVERN | 1  | 80 |
| SICIPNAM | 80 |    | SICSZERO | 1  | 20 |
| SICLIBRN | D  |    | SICVERS  | 20 |    |



## **SECTION V - SERVICE AIDS**

### **FE PATCH AREAS**

There is a patch area in each module except DSMSPSYX and DSMSPSY. Patch areas vary in size from 50 bytes to 5% of the module size, whichever is greater.

An area of approximately 1200 bytes exists between the SPG016KB and SPGFVBT labels, in the DSMSPSPG module, that may also be used for FE patches.

### **THE REGISTER-SAVE-AREA CHAIN**

As the program executes, each entry to a new module is recorded in save areas within the calling module and the called module. One save area is diagrammed in Figure 22. For the layout of register save areas, see "Standard Register Save Area Used By DLF (SAV)" on page 131.

---

| Displacement from<br>Beginning of Save<br>Area (decimal) | Length<br>(decimal) | Description   |
|--|---------------------|---|
| 0  | 1                   | (Not used)  |
| 1  | 3                   | Name of this module   |
| 4  | 4                   | Backward pointer — points<br>to save area in the<br>calling module    |
| 8  | 4                   | Forward pointer — points<br>to save area in the next<br>called module |
| 12   | 60                  | Save area for registers<br>14 through 12                              |

Figure 22. A Register Save Area

---

When calling another module, the calling module passes the address of its save area in register 13. The called module records the call in two save areas, as follows:

1. In the save area of the calling module, using register 13, the called module stores:
  - The address of its own save area (the forward pointer — SAVNEXT)
  - Registers 14 through 12 (SAVREGS).
2. In its own save area, the called module stores:
  - The address of the save area of the calling module (the backward pointer — SAVPREV)
  - Its own name (SAVMODNM).

Register 14, as stored, contains the return address into the calling module. Register 15 contains the entry address of the called module.

By requesting a dump while the Document Library Facility is executing and tracing back up the register-save-area chain, you can see where the current execution is imbedded within the sequence of calling modules. Figure 23 shows a schematic diagram of a chain of called modules.

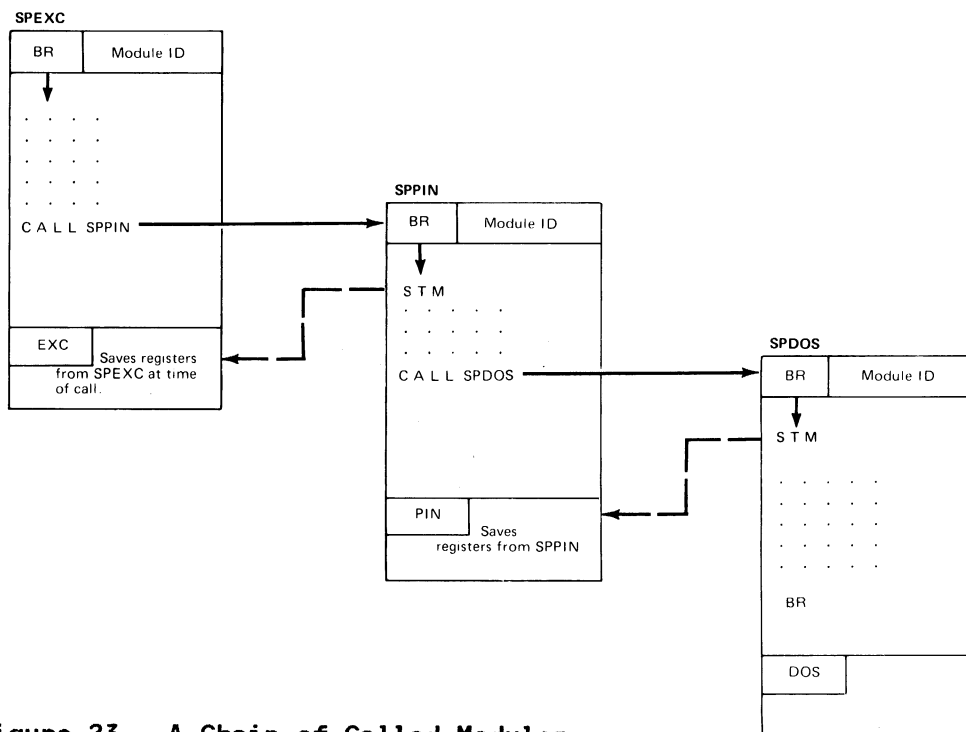


Figure 23. A Chain of Called Modules

#### MODULE NAMING CONVENTION

Every module in the Library Program has an 8-character name that begins with DSMSF. The last three characters of this name uniquely identify the module. See "Section III - Module Directory" on page 98 for names and descriptions of Document Library Facility program modules.

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

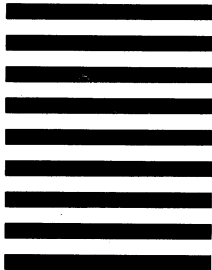
Attention: Information Development  
Department 580



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  
Box 1900  
Department 580  
Boulder, Colorado 80301



Document Library Facility Diagnosis Reference LY35-0072-1

Fold and tape

Please Do Not Staple

Fold and tape







**File Number S370-20**



Printed in the United States of America  
on recycled paper containing 10%  
recovered post-consumer fiber.

**Document Library Facility  
Diagnosis Reference**

**LY35-0072-1**

LY35-0072-01

