

sgi.



# Host Software Installation and Operations Guide

Release 1.8 of the  
StorHouse Host Software

Publication Number  
007-6316-001  
November 19, 2013

StorHouse®



© 2013 Silicon Graphics International Corp. All Rights Reserved; provided portions may be copyright in third parties, as indicated elsewhere herein. No permission is granted to copy, distribute, or create derivative works from the contents of this electronic documentation in any manner, in whole or in part, without the prior written permission of SGI.

Publication Number 007-6316-001

#### LIMITED RIGHTS LEGEND

The software described in this document is "commercial computer software" provided with restricted rights (except as to included open/free source) as specified in the FAR 52.227-19 and/or the DFAR 227.7202, or successive sections. Use beyond license provisions is a violation of worldwide intellectual property laws, treaties and conventions. This document is provided with limited rights as defined in 52.227-14.

#### TRADEMARKS AND ATTRIBUTIONS

SGI, SGI InfiniteStorage, the SGI logo, Supportfolio, SGI Trusted Edge, and SGI StorHouse are trademarks or registered trademarks of Silicon Graphics International Corp. or its subsidiaries in the United States and other countries. All other trademarks mentioned herein are the property of their respective owners.

# Contents

<b>Welcome .....</b>	<b>xi</b>
Intended Audience .....	xi
Contents of the Guide .....	xi
Related Documentation .....	xii
Notational Conventions .....	xiv
<b>Chapter 1: StorHouse Host Software Overview .....</b>	<b>1-1</b>
The StorHouse Subsystem and Callable Interface .....	1-1
The Interactive Interface .....	1-2
The File Transfer Interface .....	1-2
The CICS Interface .....	1-3
<b>Chapter 2: StorHouse Host Software Functions and Installation Re- quirements .....</b>	<b>2-1</b>
StorHouse Host Software Functions .....	2-1
StorHouse Host Software Installation Requirements .....	2-2
<b>Chapter 3: Configuring StorHouse Connections .....</b>	<b>3-1</b>
Direct Connect MVSCP/IOCP and HCD Definitions .....	3-1
MVSCP/IOCP Examples .....	3-2
Parallel Channel Connection of 224 I/O Devices .....	3-2
ESCON Channel Connection of 224 I/O Devices .....	3-2



ESCON Channel Connection with Two CHPIDs, One ESCON Director, Two Logical Control Unit Images, and 112 I/O Devices per Control Unit Image ..... 3-3

EMIF ESCON Channel Connection Shared by Two LPARs, One ESCON Director, Two Logical Control Unit Images, and 112 I/O Devices per Control Unit Image .. 3-3

Hardware Configuration Definition Example ..... 3-4

Missing Interrupt Handler (MIH) Considerations ..... 3-4

IP Connectivity ..... 3-4

**Chapter 4: Host Subsystem Installation (LSM1800) ..... 4-1**

Step 1: Load File 7 ..... 4-2

Step 2: Set Up SMP/E Procedure ..... 4-2

Step 3: Allocate Required Datasets ..... 4-2

Step 4: Initialize SMP/E CSI ..... 4-3

Step 5: Execute SMP/E RECEIVE ..... 4-4

Step 6: Execute SMP/E APPLY ..... 4-4

Step 7: Verify the SMP Steps ..... 4-4

Step 8: Execute SMP/E ACCEPT ..... 4-5

Step 9: Load the C Runtime Library ..... 4-5

Step 10: Build Configuration File ..... 4-5

Step 11: Update SYS1.PARMLIB ..... 4-5

Step 12: Optionally Update SYS1.PARMLIB ..... 4-6

Step 13: Copy StorHouse Start Procedure to PROCLIB ..... 4-6

Step 14: Starting the StorHouse Subsystem ..... 4-6

Installation Validation ..... 4-6

**Chapter 5: TSO Interface Installation (LSM1810) ..... 5-1**

Step 1: Execute SMP/E RECEIVE ..... 5-2

Step 2: Execute SMP/E APPLY ..... 5-2

Step 3: Verify the Installation ..... 5-2

Step 4: Execute SMP/E ACCEPT ..... 5-3

<b>Chapter 6: File Transfer Interface Installation (LSM1820)</b> .....	<b>6-1</b>
Step 1: Execute SMP/E RECEIVE .....	6-2
Step 2: Execute SMP/E APPLY .....	6-2
Step 3: Verify the Installation .....	6-2
Step 4: Execute SMP/E ACCEPT .....	6-3
<b>Chapter 7: CICS Interface Installation (LSM1830)</b> .....	<b>7-1</b>
Step 1: Execute SMP/E RECEIVE .....	7-2
Step 2: Execute SMP/E APPLY .....	7-2
Step 3: Verify the Installation .....	7-2
Step 4: Execute SMP/E ACCEPT .....	7-2
Step 5: CICS Preparation .....	7-3
SMP/E Load Libraries .....	7-3
Defining the CICS Interface Programs .....	7-3
Updating the DFHRPL DD Concatenation .....	7-4
LSMS DD Statement .....	7-4
<b>Chapter 8: Host Software PTF/APAR Installation</b> .....	<b>8-1</b>
Step 1: Execute the SMP/E RECEIVE .....	8-1
Step 2: Execute the SMP/E APPLY .....	8-2
Step 3: Verify the PTFs .....	8-2
Step 4: Execute the SMP/E ACCEPT .....	8-2
<b>Chapter 9: Starting and Shutting Down the StorHouse Subsystem</b> .....	<b>9-1</b>
Starting the StorHouse Subsystem .....	9-1
Shutdown Processing .....	9-1
Normal Shutdown Processing .....	9-2
Quick Shutdown Processing .....	9-2
Forced Shutdown Processing .....	9-2
Active StorHouse Subsystems .....	9-2
Abnormal StorHouse Subsystem Termination .....	9-3
StorHouse Subsystem and Callable Interface Functions .....	9-3

<b>Chapter 10: Executing the Interactive Interface .....</b>	<b>10-1</b>
Invoking the Interactive Interface .....	10-1
Authorizing the SM1 Command .....	10-1
SM1 Command Optional Parameters .....	10-2
<b>Chapter 11: StorHouse Subsystem Operator Commands .....</b>	<b>11-1</b>
Command Recognition .....	11-1
List of Operator Commands .....	11-2
<b>Chapter 12: Configuring the StorHouse Host Software .....</b>	<b>12-1</b>
StorHouse Identification .....	12-2
Internal Identification .....	12-2
External Identification .....	12-2
Operator Console Support Services Identification .....	12-2
StorHouse Identification Parameter Rules .....	12-3
Keywords .....	12-4
Sample Configuration File .....	12-11
<b>Chapter 13: StorHouse Operator Message Facility .....</b>	<b>13-1</b>
The Message Monitoring Subtask .....	13-1
Operator Messages .....	13-1
Subsystem Configuration Parameters and Operator Commands .....	13-2
Operator Message Routing .....	13-2
Other Operator Configuration Parameters and Commands .....	13-3
<b>Chapter 14: Installation Exits .....</b>	<b>14-1</b>
Control of Access to Supervisor Services .....	14-1
The Application Program Linked Module .....	14-2
Account and File Access Control .....	14-2
Default Subsystem Name Determination .....	14-3
Session Connect Account/Password Validation .....	14-3
OPEN Security Exit (LSMUXSEO) .....	14-3
Function Filter Exit (LSMUXSEF) .....	14-3



Interactive Interface Signon Prompt Control .....	14-4
Changing the Installation Exits .....	14-4
<b>Chapter 15: StorHouse Subsystem SMF Recording .....</b>	<b>15-1</b>
SMF Record Types .....	15-1
SMF Record Information .....	15-2
Contents of Specific SMF Records .....	15-3
Session Record .....	15-3
Data Transfer Record .....	15-3
Global System Records .....	15-4
Interval Recording .....	15-4
SMF Record Format .....	15-4
<b>Appendix A: Sample JCL and Program Library .....</b>	<b>A-1</b>
<b>Appendix B: Direct Connect Error Codes .....</b>	<b>B-1</b>
Callable Interface Function Codes .....	B-1
FCN Status and Network-Specific Codes .....	B-1
Messages .....	B-2
<b>Index .....</b>	<b>Index-1</b>





# Figures

Figure 0-1: Unloading File 7 .....	4-2
Figure 0-2: Executing SMP/E RECEIVE .....	4-4
Figure 0-3: Executing SMP/E APPLY.....	4-4
Figure 0-4: Executing SMP/E ACCEPT.....	4-5
Figure 0-5: Executing SMP/E RECEIVE .....	5-2
Figure 0-6: Executing SMP/E APPLY.....	5-2
Figure 0-7: Executing SMP/E ACCEPT.....	5-3
Figure 0-8: Executing SMP/E RECEIVE .....	6-2
Figure 0-9: Executing SMP/E APPLY.....	6-2
Figure 0-10: Executing SMP/E ACCEPT.....	6-3
Figure 0-11: Executing SMP/E RECEIVE .....	7-2
Figure 0-12: Executing SMP/E APPLY.....	7-2
Figure 0-13: Executing SMP/E ACCEPT.....	7-3
Figure 0-14: Executing SMP/E RECEIVE .....	8-1
Figure 0-15: Executing SMP/E APPLY.....	8-2
Figure 0-16: Executing SMP/E ACCEPT.....	8-2
Figure 0-17: Sample Configuration File.....	12-11
Figure 0-18: Changing the Installation Exits.....	14-4



# Tables

Table 4-1: LSM1800 Distribution Tape Files .....	4-1
Table 4-2: Dataset Block Allocation .....	4-3
Table 5-1: LSM1810 Distribution Tape Files .....	5-1
Table 6-1: LSM1820 Distribution Tape Files .....	6-1
Table 7-1: LSM1830 Distribution Tape Files .....	7-1
Table 8-1: Standard SGI Tape Format .....	8-1
Table B-1: Network Specific Codes .....	B-2

# Welcome

The *Host Software Installation and Operations Guide* explains how to install the StorHouse® host software on an IBM platform. The StorHouse host software enables users to access and manipulate the StorHouse system. It includes the StorHouse Callable Interface, the StorHouse Interactive Interface, the File Transfer Interface, and optionally, the CICS Interface.

## Intended Audience

The *Host Software Installation and Operations Guide* is intended for system administrators and system programmers who are responsible for installing the StorHouse host software. By following the instructions in this guide, they should require little or no assistance from SGI to install the StorHouse host software.

## Contents of the Guide

The Guide contains 15 chapters and 2 appendices:

- Chapter 1, “StorHouse Host Software Overview,” describes StorHouse host software and its interfaces with the StorHouse system.
- Chapter 2, “StorHouse Host Software Functions and Installation Requirements,” describes the distribution of StorHouse host software and installation requirements.
- Chapter 3, “Configuring StorHouse Connections,” describes how to configure ESCON and IP connections to StorHouse.
- Chapter 4, “Host Subsystem Installation (LSM1800),” describes how to install the host Subsystem.

- Chapter 5, “TSO Interface Installation (LSM1810),” details how to install the TSO Interactive Interface.
- Chapter 6, “File Transfer Interface Installation (LSM1820),” explains how to install the interface that allows users to transfer files between the host and StorHouse.
- Chapter 7, “CICS Interface Installation (LSM1830),” describes how to install the CICS Interface that allows file transfers between the host and StorHouse via CICS Command Level application programs.
- Chapter 8, “Host Software PTF/APAR Installation,” describes how to install corrections to the SGI host software.
- Chapter 9, “Starting and Shutting Down the StorHouse Subsystem,” shows how to execute the Subsystem.
- Chapter 10, “Executing the Interactive Interface,” explains how to use the Interactive Interface.
- Chapter 11, “StorHouse Subsystem Operator Commands,” details the commands the operator may use when the StorHouse Subsystem is running.
- Chapter 12, “Configuring the StorHouse Host Software,” describes the StorHouse configuration parameters and how to set them.
- Chapter 13, “StorHouse Operator Message Facility,” describes the operation of the message facility in regard to operator messages.
- Chapter 14, “Installation Exits,” describes the user exit modules.
- Chapter 15, “StorHouse Subsystem SMF Recording,” describes the three types of SMF records supported by the StorHouse host software.
- Appendix A, “Sample JCL and Program Library,” lists the members of the dataset SAMPLES (sample JCL and program library) to be used when installing the host software.
- Appendix B, “Direct Connect Error Codes,” lists and describes the error codes returned from the Direct Connect feature software.

## Related Documentation

For more information about StorHouse, consult the following documents:

- *Callable Interface Programmer's Guide*, publication number 900013, is a reference for programmers who write applications that use the Callable Interface. It



explains the functions of the Callable Interface and contains a sample COBOL program.

- *Messages and Codes Manual*, publication number 900032, lists all StorHouse system and host messages, gives the meaning of each message, and indicates any actions to take as a result of the messages.
- *Command Language Reference Manual*, publication number 900005, explains all available commands in the StorHouse Command Language plus related StorHouse concepts.



## Notational Conventions

This book uses the following conventions for illustrating command formats, presenting examples, and identifying special terms:

Convention	Meaning
Angle brackets (< >)	Enclose optional entries
Braces ( { } )	Enclose descriptive terms or a choice of entries
Courier font	Code
Ellipses (...)	A repetition of the preceding material
<i>Italics</i>	New terms and emphasized text
lower case Helvetica font	User entries
UPPER CASE	System responses and StorHouse terms

---

# StorHouse Host Software Overview

The StorHouse host software allows users to communicate with StorHouse from an interactive session or from a user application program running on a host computer. The host software consists of the following components:

- The StorHouse Subsystem and Callable Interface
- The Interactive Interface
- The File Transfer Interface
- The CICS Interface.

## The StorHouse Subsystem and Callable Interface

The StorHouse Subsystem is a standard MVS, OS/390, or z/OS Subsystem that transfers data, control information, and status between user application programs and StorHouse. The Subsystem communicates with StorHouse either through a direct channel connection supported by EXCP-level I/O or through TCP/IP. The IP connection is supported through Open Edition sockets.

The *Callable Interface* provides a link between StorHouse and user application programs, which may be written in Assembler or in a high-level language such as COBOL, PL/I, or C. The Callable Interface functions in a batch or TSO environment. LSMCALL is a subroutine link-edited with the user program. It dynamically loads the other Callable Interface modules into the user address space and uses cross-memory services to communicate with the StorHouse Subsystem.

The StorHouse Subsystem must be active before any service can be requested from StorHouse. If it is not active, LSMCALL will return a value in the user's return code field to indicate that the Subsystem is inactive.

## The Interactive Interface

The *Interactive Interface* is a SGI-provided application program that allows users to enter StorHouse Command Language requests and to receive responses interactively. The Interactive Interface operates as a TSO Command Processor and uses the StorHouse Subsystem and Callable Interface to communicate with StorHouse.

The Interactive Interface performs the following major functions:

- Interacts with users through standard TSO terminal interface facilities. The interface is line oriented and receives requests and issues responses through the GETLINE and PUTLINE SVCs.
- Sends user requests to StorHouse by way of the StorHouse Subsystem. The Interactive Interface uses the Callable Interface just as any user application would. The Interactive Interface does not process user requests locally except for some terminal editing.
- Invokes the File Transfer Interface to PUT/GET files as requested by users. If a user requests a file transfer operation, StorHouse notifies the Interactive Interface and provides the information needed to invoke the File Transfer Interface. Once the transfer operation is complete, StorHouse sends the Interactive Interface an ending status, which is returned to the user.

The Interactive Interface runs in problem program mode and key and requires APF authorization if File Transfer Interface operations using DFDSS will be performed.

## The File Transfer Interface

The *File Transfer Interface* provides the services required to use the StorHouse Command Language PUT and GET commands. This Interface is used by the Interactive Interface; it is not intended for use by user-written applications. This Interface consists of routines that build control statements to invoke the IBM utility DFDSS and routines that are DFDSS exits to perform necessary StorHouse I/O functions.

The File Transfer Interface is required only for file transfer operations within the Interactive Interface and is not required if the Interactive Interface will be used only as an administrative tool. As noted above, some DFDSS functions require authorization; the Interactive Interface and the File Transfer Interface must be APF authorized.



## The CICS Interface

The *CICS Interface* provides a Callable Interface to StorHouse for CICS Command Level application programs written in any CICS-supported programming language (COBOL, COBOL II, PL/I, Assembler, or C).

The CICS Interface functions like the Callable Interface provided for Batch and TSO. CICS application programs use LSMCICS instead of LSMCALL to invoke StorHouse services. The LSMCICS subroutine is link-edited with the user program and communicates with the other components in the interface.

The StorHouse Subsystem considers the CICS address space to be a Multi-User Single Address Space (MUSAS). A unique 16-byte MUSAS character string is assigned to each CICS StorHouse transaction. This identifier can be displayed using various Subsystem operator commands and is used to provide resource cleanup for abnormally-terminated transactions.

The CICS Interface supports:

- Pseudo-conversational and conversational CICS transactions in a single CICS region or across multiple CICS regions linked via MRO or ISC.
- CICS releases from 1.7 to 4.1 to CICS/TS 2.2.



---

# StorHouse Host Software Functions and Installation Requirements

The StorHouse host software is installed with IBM's System Maintenance Program Extended (SMP/E). The installation process is designed to use a separate Consolidated Software Index (CSI), and SGI recommends that you follow this rule if possible. Currently, there are no known conflicts with IBM software names, but some of the StorHouse host software names do begin with the letter I. This can cause confusion if the software is installed in a CSI containing IBM software.

## StorHouse Host Software Functions

The StorHouse host software is distributed as separate functions, each on a separate tape. These functions should be installed individually. The Function Management IDs (FMIDs) are in the format LSMvrff, where:

- LSM – constant
- v – software version
- r – software release
- ff – function id.

The StorHouse host software functions are:

- The StorHouse Subsystem and Callable Interface, which has an ff of 00
- The Interactive Interface, which has an ff of 10
- The File Transfer Interface, which has an ff of 20
- The CICS Interface, which has an ff of 30.

Function 00 is a base function. It is an unconditional prerequisite for the other functions and must be installed before them. Installed, in this case, means that the SMP/E RECEIVE and APPLY steps have been completed.

## StorHouse Host Software Installation Requirements

To install and run the StorHouse host software, a site must have the following:

- An IBM™ mainframe computer host (any model) running MVS/ESA™, OS/390, or z/OS.
- A 3480 magnetic tape drive
- A 3380 or 3390 magnetic disk with 24 free cylinders
- A functioning StorHouse Storage Processor (SP)
- Magnetic tapes containing the StorHouse host software and installation control statements
- An authorized library for subsystem execution
- Available disk space equal to approximately 24 3380 cylinders
- 7 KB in CSA or ECSA
- SMP/E
- DFDSS version 2 release 2 (or later) if the optional File Transfer Interface (FMID LSM1820) is included
- StorHouse connectivity through Direct Connect or TCP/IP.

---

# Configuring StorHouse Connections

StorHouse supports ESCON connections to an MVS/ESA, OS/390, or z/OS Host and IP connections to an OS/390 or z/OS Host. This chapter explains how to configure each to work with StorHouse.

## Direct Connect MVSCP/IOCP and HCD Definitions

StorHouse supports parallel and ESCON channel attachments to an MVS/ESA, OS/390, or z/OS Host and appears as a channel-to-channel adapter device. StorHouse is defined as a device type 3088 when connected to a parallel channel, and as a device type SCTC for an ESCON connection. Each StorHouse channel interface can support a maximum of 230 devices (256 for UltraSPARC processors). StorHouse can be equipped with a maximum of six parallel, ESCON, or a combination of parallel and ESCON channel interfaces.

Each StorHouse channel interface can provide up to 16 control unit images, thus supporting connection to multiple ESCON channels through an ESCON Director (ESCD), or to shared ESCON channels on EMIF-capable processors connected directly or through an ESCON director. However, you must define a logical control unit image (using the CUADD parameter on the CNTLUNIT specification for IOCP or Logical Address on the Define Control Unit panel for HCD) for each non-EMIF ESCON channel or for each LPAR sharing an EMIF ESCON channel. Each logical control unit must have a unique I/O device number range. For EMIF, the I/O device number range must be dedicated to that LPAR (using the PARTITION parameter on the IODEVICE specification for IOCP or the Define Device Candidate List from the Define I/O Device panel for HCD). In other words, there must be a unique StorHouse logical control unit with associated I/O devices for each logical path from the MVS/ESA, OS/390, or z/OS Host(s).

The LSMS Subsystem supports MVS/SP 5.1 four-digit device numbers.

## MVSCP/IOCP Examples

Examples of MVSCP/IOCP are presented in this section for parallel and ESCON channel connections of 224 I/O devices.

### Parallel Channel Connection of 224 I/O Devices

Because a 3088 control unit will support either 32 or 64 I/O devices, four CNTLUNIT definitions are needed to define the 224 StorHouse devices. However, the LSMS Subsystem and StorHouse will treat the 224 devices as one contiguous range. In the LSMS Subsystem initialization file, the parameter SM\_NETPARM specifies the first device number in the range. In this example, the number is 600.

```
CHPID PATH=26,TYPE=BL
CNTLUNIT UNIT=3088,PROTOCL=S,SHARED=N,
        PATH=(26),CUNUMBR=600,UNITADD=((00,64))
IODEVICE UNIT=CTC,FEATURE=370,OFFLINE=YES,
        CUNUMBR=600,ADDRESS=(600,64)
CNTLUNIT UNIT=3088,PROTOCL=S,SHARED=N,
        PATH=(26),CUNUMBR=640,UNITADD=((40,64))
IODEVICE UNIT=CTC,FEATURE=370,OFFLINE=YES,
        CUNUMBR=640,ADDRESS=(640,64)
CNTLUNIT UNIT=3088,PROTOCL=S,SHARED=N,
        PATH=(26),CUNUMBR=680,UNITADD=((80,64))
IODEVICE UNIT=CTC,FEATURE=370,OFFLINE=YES,
        CUNUMBR=680,ADDRESS=(680,64)
CNTLUNIT UNIT=3088,PROTOCL=S,SHARED=N,
        PATH=(26),CUNUMBR=6C0,UNITADD=((C0,32))
IODEVICE UNIT=CTC,FEATURE=370,OFFLINE=YES,
        CUNUMBR=6C0,ADDRESS=(6C0,32)
```

### ESCON Channel Connection of 224 I/O Devices

**Note** For ESCON SCTC, the UNITADD parameter for both the CNTLUNIT and IODEVICE macros *must* be 00.

```
CHPID PATH=26,TYPE=CNC
CNTLUNIT UNIT=SCTC,PATH=(26),CUNUMBR=600,UNITADD=((00,224))
IODEVICE UNIT=SCTC,OFFLINE=YES,CUNUMBR=600,
        ADDRESS=(600,224),UNITADD=00
```

### ESCON Channel Connection with Two CHPIDs, One ESCON Director, Two Logical Control Unit Images, and 112 I/O Devices per Control Unit Image

```
CHPID PATH=25,TYPE=CNC,SWITCH=00
```

```
CHPID PATH=26,TYPE=CNC,SWITCH=00
```

```
*Define Logical Control Unit Image 0, CUADD=0
```

```
CNTLUNIT CUNUMBR=500,PATH=(25),LINK=C5,CUADD=0,  
UNITADD=((00,112)),UNIT=SCTC
```

```
IODEVICE UNIT=SCTC,CUNUMBR=500,ADDRESS=(500,112),  
OFFLINE=YES,UNITADD=00
```

```
*Define Logical Control Unit Image 1, CUADD=1
```

```
CNTLUNIT CUNUMBR=600,PATH=(26),LINK=C5,CUADD=1,  
UNITADD=((00,112)),UNIT=SCTC
```

```
IODEVICE UNIT=SCTC,CUNUMBR=600,ADDRESS=(600,112),  
OFFLINE=YES,UNITADD=00
```

### EMIF ESCON Channel Connection Shared by Two LPARs, One ESCON Director, Two Logical Control Unit Images, and 112 I/O Devices per Control Unit Image

```
CHPID PATH=25,TYPE=CNC,SWITCH=00,SHARED
```

```
* Define Logical Control Unit Image 1,CUADD=1,for LPAR1
```

```
CNTLUNIT CUNUMBR=500,PATH=(25),LINK=C5,CUADD=1,  
UNITADD=((00,112)),UNIT=SCTC
```

```
* Explicit I/O device definitions for LPAR1
```

```
IODEVICE UNIT=SCTC,CUNUMBR=500,ADDRESS=(500,112),PART=LPAR1,  
OFFLINE=YES,UNITADD=00
```

```
* Define Logical Control Unit Image 2,CUADD=2,for LPAR2
```

```
CNTLUNIT CUNUMBR=600,PATH=(25),LINK=C5,CUADD=2,  
UNITADD=((00,112)),UNIT=SCTC
```

```
* Explicit I/O device definitions for LPAR2
```

```
IODEVICE UNIT=SCTC,CUNUMBR=600,ADDRESS=(600,112),PART=LPAR2,  
OFFLINE=YES,UNITADD=00
```

## Hardware Configuration Definition Example

The MVSCP/IOCP definitions presented in the sections entitled “Parallel Channel Connection of 224 I/O Devices” and “ESCON Channel Connection of 224 I/O Devices” on page 3-2 translate directly into HCD panel definitions. When defining ESCON SCTC control units and I/O devices, the value entered in the “Unit Address” field on the “Control Unit Definition” and/or on the “Device Definition” panels must be “00” if the last two digits of the control unit number or first device number are not “00”.

Because 3088, CTC, and SCTC are standard IBM-defined devices, no special UIM is required.

## Missing Interrupt Handler (MIH) Considerations

The StorHouse devices must be exempt from MIH analysis. This is accomplished by providing an IECIOSxx member in SYS1.PARMLIB. The following line must be present in this member:

```
MIH DEV=(FFFF-LLLL) , TIME=00:00
```

“FFFF-LLLL” specifies the first and last StorHouse device numbers defined.

This IECIOSxx member is activated through the IOS=xx line in the IEASYSxx SYS1.PARMLIB member or by issuing the MVS operator command SET IOS=xx.

## IP Connectivity

StorHouse can be connected to an “MVS” (OS/390 or z/OS) host system using a LAN connection rather than a channel (bus/tag or ESCON) connection. To use an IP connection with StorHouse, the following must be in place:

- StorHouse requires Open MVS (OMVS) for socket-level support.
- The user-id that is associated with the LSM started task must have RACF OMVS access defined. The UID does not need to be zero. An example of adding OMVS access for an existing user-id follows:

```
ALTUSER userid +
OMVS(HOME('/') UID(3030372) PROGRAM('/bin/sh')) +
WORKATTR(WANAME('SM1 User') WAACCNT(@SYS1))
```

The plus sign (+) is a statement continuation indicator. You can split the command at any space, but the last character in any line (except the last line) must be a +.



- TCPIP must be active on the MVS system, which requires that UNIX System Services (USS) is configured for full-function Kernel Services. Installation, configuration, and operation of TCPIP and USS are beyond the scope of this manual.

A LAN connection is indicated in the subsystem configuration dataset (see Chapter 12) by specifying `NETWORK_TYPE = TCP`. The `SM_LINKNAME` provides the port number, and the `SM_HOSTID` parameter provides either the system name or the IP address for the StorHouse server.

If the StorHouse server is identified by its system name, either a Resolver must be available to contact a name server, or the name must be in the `/etc/hosts` file that resides in the root HFS. If a Resolver is used, SGI recommends that the TCPIP software be as current as possible. If the StorHouse system cannot be contacted through the available name services, then the system name may need to be put in `/etc/hosts` or the IP address must be coded directly in the configuration dataset.

# 3

## Configuring StorHouse Connections

---

IP Connectivity

## Host Subsystem Installation (LSM1800)

The distribution tape for LSM1800 is labeled SM1800. Table 4-1 lists the files on the tape.

**Table 4-1: LSM1800 Distribution Tape Files**

File	Dataset Name	Contents
1	SMPNCS	SMP/E modification control statements
2	LSM1800.F1	LSM1800 JCLIN
3	LSM1800.F2	LSM1800 loadlib (LSMLOAD)
4	LSM1800.F3	LSM1800 maclib (LSMMAC)
5	LSM1800.F4	LSM1800 source (LSMSRC)
6	CRUNTIME	C code runtime library
7	SAMPLES	Sample JCL and program library

Files 1 through 5 are used by the SMP/E installation procedures. File 6 is the C code runtime library that is required by the Subsystem, and file 7 contains sample JCL that can be used to complete the installation process. Appendix C in this document lists all members in the dataset SAMPLES.

## Step 1: Load File 7

Use the sample JCL in Figure 4-1 to load file 7, the SAMPLES dataset. Supply a valid JOB card and substitute appropriate values for the DSN, VOLSER, and UNIT parameters.

```
//LSMLOAD JOB . . . . <===
//UNLOAD EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=SAMPLES,VOL=SER=SM1800,UNIT=unit, <===
           LABEL=(7,SL),DISP=OLD
//SYSUT2 DD DSN=dsn, <===
           UNIT=unit, <===
           VOL=SER=volser, <===
           SPACE=(TRK,(15,1,12)),DISP=(NEW,CATLG,DELETE)
//SYSUT3 DD DSN=&&SYSUT3,SPACE=(TRK,(1,1)),UNIT=unit <===
//SYSUT4 DD DSN=&&SYSUT4,SPACE=(TRK,(1,1)),UNIT=unit <===
//SYSIN DD DUMMY
```

Figure 4-1: Unloading File 7

## Step 2: Set Up SMP/E Procedure

Member LSMSMPE in the SAMPLES dataset is an SMP/E JCL procedure that can be used for the install process. LSMSMPE may be copied to a system PROCLIB at this time or may be copied into executable JCL and used as an in-stream procedure. This procedure should be customized by replacing the substitution parameters with appropriate values before proceeding with the installation process.

## Step 3: Allocate Required Datasets

Edit member LSMALLOC in the SAMPLES dataset to specify the appropriate SMP/E and LSM dataset high-level indexes, as well as the SMP/E and LSM dataset units and volumes. Submit the job when all updates have been completed. This step must be completed successfully before proceeding with the next step. The datasets to be allocated are:

- ALSMLOAD – SMP/E dataset that will contain the host software distribution modules.
- LSMLOAD – SMP/E dataset that will contain executable code.
- LSMMAC – SMP/E dataset that will contain assembler language macros.
- LSMSRC – SMP/E dataset that will contain source for the default user exits.

- SMPPTS – Required SMP/E dataset.
- SMPMTS – Required SMP/E dataset.
- SMPSTS – Required SMP/E dataset.
- SMPSCDS – Required SMP/E dataset.
- SMPCSI – Required SMP/E VSAM dataset.

ALSMLOAD, LSMMAC, and LSMSRC are SMP/E distribution zone libraries. LSMLOAD, SMPMTS, and SMPSTS are SMP/E target zone libraries.

All datasets, except the CSI, are allocated by blocks in the amounts listed in Table 4-2. Make sure the requested space is available on the volumes specified in LSMALLOC before running this step.

**Table 4-2: Dataset Block Allocation**

Dataset	Blocks Required	Blocksize
ALSMLOAD	150	23,476
LSMLOAD	300	23,476
LSMMAC	105	6,160
LSMSRC	105	6,160
SMPMTS	105	6,160
SMPPTS	105	6,160
SMPSCDS	105	6,160
SMPSTS	105	6,160

Because the installation uses SMP/E, a CSI must also be allocated. This is a VSAM dataset that stores the management information needed by SMP/E. The CSI requires 9 cylinders of 3380 (or equivalent) disk space.

## Step 4: Initialize SMP/E CSI

The SMP/E CSI must now be initialized. Member LSMUCLIN in the SAMPLES dataset contains the JCL and UCLIN necessary to accomplish this. Edit this member to specify the correct CSI cluster name and ZONE names, TLIB prefix, and the LSM dataset high-level index. When all updates have been completed, submit the job.

The UCLIN specifies ASMA90 as the assembler. MVS/SP installations that do not have the high-level Assembler installed should change all occurrences of ASMA90 to IEV90 in LSMUCLIN.

Member LSMDDDEF in the SAMPLES dataset contains SMP/E DDDEF statements for all zones in the CSI. Edit this member to specify the CSI cluster and zone names, the TLIB unit and volume, and the correct high-level LSM dataset name prefix. Although running this job is optional, it will facilitate use of the ISPF SMP/E Dialogs. ISPF SMP/E Dialogs may be used for RECEIVE, APPLY, and ACCEPT commands.

## Step 5: Execute SMP/E RECEIVE

The software must be loaded using the SMP/E RECEIVE command shown in Figure 4-2. An appropriate unit must be specified for tape devices. The use of the SOURCEID operand is optional.

```

/LLSMRECV JOB ....
//LSMSMPE EXEC LSMSMPE
//SMPCNTL DD *
  SET BOUNDARY(GLOBAL) .
  RECEIVE SELECT (LSM1800) SOURCEID(source id name). <===
//SMPPTFIN DD DSN=SMPMCS,VOL=SER=SM1800,UNIT=unit, <===
//          LABEL=(1,SL),DISP=OLD
//SMPHOLD DD DUMMY

```

Figure 4-2: Executing SMP/E RECEIVE

## Step 6: Execute SMP/E APPLY

The APPLY process copies the macros and source members to the SMPMTS and SMPSTS libraries and creates the executable load modules in the LSMLOAD library. Use the job stream in Figure 4-3 to execute the SMP/E APPLY after updating the target zone name.

```

//LSMAPPLY JOB ...
//LSMSMPE EXEC LSMSMPE
//SMPCNTL DD *
  SET BOUNDARY(target zone name). <===
  APPLY SELECT(LSM1800) .

```

Figure 4-3: Executing SMP/E APPLY

## Step 7: Verify the SMP Steps

All SMP/E return codes should be zero. Any non-zero values must be investigated. If needed, call your service representative for assistance.

## Step 8: Execute SMP/E ACCEPT

The ACCEPT updates the distribution libraries and indicates that the software is completely installed as far as SMP/E is concerned. Use the job stream in Figure 4-4 to execute the SMP/E ACCEPT after updating the distribution zone name.

```
//LSMACCEPT JOB ....  
//LSMSMPE EXEC LSMSMPE  
//SMPCNTL DD *  
  SET BOUNDARY(distribution zone name). <===  
  ACCEPT SELECT(LSM1800).
```

Figure 4-4: Executing SMP/E ACCEPT

## Step 9: Load the C Runtime Library

Much of the StorHouse host software is written in C and requires the C runtime library to be available. Use member CRUNTIME in the SAMPLES dataset. Edit CRUNTIME to select the appropriate LSM high-level index and tape unit. Note that this load uses IEBCOPY, not SMP/E.

## Step 10: Build Configuration File

The SMCONFIG member of the SAMPLES dataset is a sample StorHouse Subsystem initialization file. Refer to Chapter 12, “Configuring the StorHouse Host Software.” This dataset supplies parameter information required for operation of the StorHouse Subsystem; it is read when the subsystem is started. These parameters should be reviewed prior to first starting the Subsystem. If you save the changes under another name or in a different library, the SMLSMS procedure will need to be updated (Step 13).

## Step 11: Update SYS1.PARMLIB

The LSMLOAD library must be authorized for the Subsystem to execute properly. Add the full name of the LSMLOAD library created by the LSMALLOC procedure executed in step 3 to the active IEAAPFxx member or PROCxx members in SYS1.PARMLIB. Make sure the correct volume serial number is also specified. The Subsystem will fail to initialize if this is not completed properly. If this is a new authorized library, you may need to IPL the system.

## Step 12: Optionally Update SYS1.PARMLIB

Putting the StorHouse Subsystem name in IEASSNxx results in the creation of the necessary MVS control blocks to support the StorHouse Subsystem at IPL time. Since the StorHouse Subsystem builds all necessary control blocks during initialization, it is unnecessary to define the requested Subsystem in the active IEASSNxx member of SYS1.PARMLIB. If you do define the name in IEASSNxx (for example, because of a site documentation standard), the Subsystem will use the pre-existing control blocks when it is started.

## Step 13: Copy StorHouse Start Procedure to PROCLIB

The procedure SMLSMS should be copied from the SAMPLES dataset to a system PROCLIB. If the configuration dataset was saved under a new name or placed in a different dataset, the SMLSMS procedure will need to be updated before trying to start the subsystem.

## Step 14: Starting the StorHouse Subsystem

Before starting the StorHouse Subsystem, you may want to refer to Chapter 9, “Starting and Shutting Down the StorHouse Subsystem,” to gain a more complete understanding of how it works. The LSMLOAD dataset must be authorized or the Subsystem will not initialize. Make sure the dataset name and volume serial number have been added to the active IEAAPFxx member of SYS1.PARMLIB.

## Installation Validation

The most straightforward validation of the StorHouse installation is through the use of the Interactive Interface, which is installed next. (For further information on the Interactive Interface, refer to Chapter 5, “TSO Interface Installation (LSM1810),” for installation procedures and Chapter 10, “Executing the Interactive Interface,” for a general discussion. The Interactive Interface is layered on the base product and exercises almost all of the base product code. Additionally, failures from this interface will produce diagnostic messages at the terminal.

The base product can be validated by running the program in the SMSAMP member of the SAMPLES dataset. This member contains the required JCL. Commentary in the member explains the necessary customization.



## TSO Interface Installation (LSM1810)

The distribution tape for LSM1810 is labeled SM1810. Table 5-1 lists the files on the tape.

**Table 5-1: LSM1810 Distribution Tape Files**

File	Dataset Name	Contents
1	SMPMCS	SMP/E modification control statements
2	LSM1810.F1	LSM1810 JCLIN
3	LSM1810.F2	LSM1810 loadlib (LSMLOAD)
4	LSM1810.F3	LSM1810 source (LSMSRC)

Files 1 through 4 are used by the SMP/E installation procedures. LSM1810 should be installed using the same SMP/E datasets that were used for the installation of the base component, LSM1800. The same target and distribution zones should also be used since some of the modules loaded during the installation of the base component are used during the installation of LSM1810. The LSMSMPE procedure from the SAMPLES library is also required.

**Note** ISPF SMP/E Dialogs may be used for RECEIVE, APPLY, and ACCEPT commands.

## Step 1: Execute SMP/E RECEIVE

The software must be loaded using the SMP/E RECEIVE command in Figure 5-1. An appropriate unit must be specified for tape devices. The use of the SOURCEID operand is optional.

```
//LSMRECV JOB ....
//LSMSMPE EXEC LSMSMPE
//SMPCNTL DD *
  SET BOUNDARY(GLOBAL) .
  RECEIVE SELECT(LSM1810) SOURCEID(source id name). <===
//SMFTFIN DD DSN=SMPMCS,VOL=SER=SM1800,UNIT=unit, <===
//          LABEL=(1,SL),DISP=OLD
//SMPHOLD DD DUMMY
```

Figure 5-1: Executing SMP/E RECEIVE

## Step 2: Execute SMP/E APPLY

The APPLY process creates the executable load modules in the LSMLOAD library. Use the job stream in Figure 5-2 to execute the SMP/E APPLY after updating the target zone name.

```
//LSMAPPLY JOB ....
//LSMSMPE EXEC LSMSMPE
//SMPCNTL DD *
  SET BOUNDARY(target zone name). <===
  APPLY SELECT(LSM1810) .
```

Figure 5-2: Executing SMP/E APPLY

## Step 3: Verify the Installation

All SMP/E return codes should be 4 or 0. Any values greater than 4 must be investigated. If needed, call your service representative for assistance. The return code 4 should be in link-edit operations, and the warning from the link-edit processing steps should be 2651.

## Step 4: Execute SMP/E ACCEPT

The ACCEPT updates the distribution libraries and indicates that the software is completely installed as far as SMP/E is concerned. Use the following job stream in Figure 5-3 to execute the SMP/E ACCEPT after updating the distribution zone name.

```
//LSMACEPT JOB ....  
//LSMSMPE EXEC LSMSMPE  
//SMPCNTL DD *  
  SET BOUNDARY(distribution zone name). <===  
  ACCEPT SELECT(LSM1810).
```

**Figure 5-3: Executing SMP/E ACCEPT**

**5**

**TSO Interface Installation (LSM1810)**

Step 4: Execute SMP/E ACCEPT

---

## File Transfer Interface Installation (LSM1820)

The distribution tape for LSM1820 is labeled SM1820. Table 6-1 lists the files on the tape.

**Table 6-1: LSM1820 Distribution Tape Files**

File	Dataset Name	Contents
1	SMPNCS	SMP/E modification control statements
2	LSM1820.F1	LSM1820 JCLIN
3	LSM1820.F2	LSM1820 loadlib (LSMLOAD)

Files 1 through 3 are used by the SMP/E installation procedures. LSM1820 should be installed using the same SMP/E datasets that were used for the installation of the base component, LSM1800. The same target and distribution zones should also be used since some of the modules loaded during the installation of the base component are used during the installation of LSM1820. The LSMSMPE procedure from the SAMPLES library is also required.

**Note** ISPF SMP/E Dialogs may be used for RECEIVE, APPLY, and ACCEPT commands.

## Step 1: Execute SMP/E RECEIVE

The software must be loaded using the SMP/E RECEIVE command in Figure 6-1. An appropriate unit must be specified for tape devices. The use of the SOURCEID operand is optional.

```
//LSMRECV JOB ....
//LSMSMPE EXEC LSMSMPE
//SMPCNTL DD *
  SET BOUNDARY(GLOBAL) .
  RECEIVE SELECT(LSM1820)SOURCEID(source id name).      <===
//SMFTFIN DD DSN=SMPMCS,VOL=SER=SM1820,UNIT=unit,      <===
//          LABEL=(1,SL),DISP=OLD
//SMPHOLD DD DUMMY
```

Figure 6-1: Executing SMP/E RECEIVE

## Step 2: Execute SMP/E APPLY

The APPLY process creates the executable load modules in the LSMLOAD library. Use the job stream in Figure 6-2 to execute the SMP/E APPLY after updating the target zone name.

```
//LSMAPPLY JOB ....
//LSMSMPE EXEC LSMSMPE
//SMPCNTL DD *
  SET BOUNDARY(target zone name).                        <===
  APPLY SELECT(LSM1820) .
```

Figure 6-2: Executing SMP/E APPLY

## Step 3: Verify the Installation

All SMP/E return codes should be zero. Any non-zero values must be investigated. If needed, call your service representative for assistance.

## Step 4: Execute SMP/E ACCEPT

The ACCEPT updates the distribution libraries and indicates that the software is completely installed as far as SMP/E is concerned. Use the job stream in Figure 6-3 to execute the SMP/E ACCEPT after updating the distribution zone name.

```
//LSMACEPT JOB ....  
//LSMSMPE EXEC LSMSMPE  
//SMPCNTL DD *  
  SET BOUNDARY(distribution zone name). <===  
  ACCEPT SELECT(LSM1820).
```

**Figure 6-3: Executing SMP/E ACCEPT**





## CICS Interface Installation (LSM1830)

The distribution tape for LSM1830 is labeled SM1830. Table 7-1 lists the files on the tape.

**Table 7-1: LSM1830 Distribution Tape Files**

File	Dataset Name	Contents
1	SMPMCS	SMP/E modification control statements
2	LSM1810.F1	LSM1830 JCLIN
3	LSM1810.F2	LSM1830 loadlib (LSMLOAD)

Files 1 through 3 are used by the SMP/E installation procedures. Install LSM1830 using the same SMP/E datasets used in the installation of the base component, LSM1800. The same target and distribution zones should also be used since some of the modules loaded during LSM1800 installation are used in LSM1830 installation. The LSMSMPE procedure from the SAMPLES library is also required.

**Note** ISPF SMP/E Dialogs may be used for RECEIVE, APPLY, and ACCEPT commands.

## Step 1: Execute SMP/E RECEIVE

The software must be loaded using the SMP/E RECEIVE command shown in Figure 7-1. An appropriate unit must be specified for tape devices. The use of the SOURCEID operand is optional.

```
//LSMRECV JOB ....
//LSMSMPE EXEC LSMSMPE
//SMPCNTL DD *
  SET BOUNDARY(GLOBAL) .
  RECEIVE SELECT(LSM1830) SOURCEID(source id name). <===
//SMFTFIN DD DSN=SMPMCS,VOL=SER=SM1830,UNIT=unit, <===
//          LABEL=(1,SL),DISP=OLD
//SMPHOLD DD DUMMY
```

Figure 7-1: Executing SMP/E RECEIVE

## Step 2: Execute SMP/E APPLY

The APPLY process creates the executable load modules in the LSMLOAD library. After updating the target zone name, use the job stream in Figure 7-2 to execute the SMP/E APPLY.

```
//LSMAPPLY JOB ....
//LSMSMPE EXEC LSMSMPE
//SMPCNTL DD *
  SET BOUNDARY(target zone name). <===
  APPLY SELECT(LSM1830) .
```

Figure 7-2: Executing SMP/E APPLY

## Step 3: Verify the Installation

All SMP/E return codes should be zero. Any non-zero values must be investigated. If needed, call your service representative for assistance.

## Step 4: Execute SMP/E ACCEPT

The ACCEPT updates the distribution libraries and indicates that SMP/E has completed its install.

After updating the distribution zone name, use the job stream in Figure 7-3 to execute the SMP/E ACCEPT.

```
//LSMACEPT JOB ....  
//LSMSMPE EXEC LSMSMPE  
//SMPCTL DD *  
  SET BOUNDARY(distribution zone name). <====  
  ACCEPT SELECT(LSM1830).
```

Figure 7-3: Executing SMP/E ACCEPT

## Step 5: CICS Preparation

The following paragraphs describe CICS preparation. They include a discussion of the SMP/E load libraries, a description of how to define CICS interface programs, a description of how to update the DFHRPL DD concatenation, and a discussion of the LSMS DD statement.

### SMP/E Load Libraries

During the SMP/E APPLY process, all load modules are linked into the LSMLOAD target zone library. During SMP/E ACCEPT processing, all load modules are copied into the ALSMLOAD distributed zone library.

### Defining the CICS Interface Programs

The CICS Interface was designed to exploit CICS Distributed Transaction Processing facilities. It consists of the LSMC transaction and five programs, LSMCFLH, LSMCICS, LSMLSMC, LSMLOAD, and LSMUXSSN. LSMLOAD and LSMUXSSN are components provided in the Host Subsystem component FMID LSM1800. LSMCICS is the interface stub link-edited with a user's transaction program and does not require a resource definition.

When a user program CALLS the LSMCICS stub, LSMCICS executes a CICS LINK to program LSMCFLH. When first establishing a StorHouse session, (using the CONNECT function), LSMCFLH starts server transaction LSMC which invokes program LSMLSMC. The LSMC transaction remains active for the life of the user's StorHouse session (CONNECT to DISCONNECT) and processes the user's requests from LSMCFLH to the Host Subsystem and StorHouse. The user's application can span multiple CICS pseudo-conversational tasks across multiple CICS regions and systems during the StorHouse session.

The LSMC transaction is defined as local with programs LSMLSMC, LSMCFLH, LSMLOAD, and LSMUXSSN in the CICS region that executes on the CPU that is

connected to StorHouse and runs the Host Subsystem. User transactions that access StorHouse can then be defined in this CICS region. User transactions can also be defined in any CICS region connected via MRO or ISC in which LSMC is defined as a *remote* transaction and a program definition for LSMCFLH is created.

For example, the following CICS regions are set up:

- CICS SMOR – StorHouse owning region
- CICS AOR1 – Application owning region 1
- CICS AOR2 – Application owning region 2.

The RDO statements for CICS SMOR are as follows:

```
DEFINE TRANSACTION(LSMC)    GROUP(SMORSMIF)    PROGRAM(LSMLSMC)
                           TASKDATALOC(ANY)    TASKDATAKEY(CICS)

DEFINE PROGRAM(LSMLSMC)    GROUP(SMORSMIF)    LANGUAGE(ASSEMBLER)
                           DATALOCATION(ANY)    EXECCKEY(CICS)
DEFINE PROGRAM(LSMCFLH)    GROUP(SMORSMIF)    LANGUAGE(ASSEMBLER)
                           DATALOCATION(ANY)
DEFINE PROGRAM(LSMLOAD)    GROUP(SMORSMIF)    LANGUAGE(ASSEMBLER)
                           ATALOCATION(ANY)    EXECCKEY(CICS)
DEFINE PROGRAM(LSMUXSSN)    GROUP(SMORSMIF)    LANGUAGE(ASSEMBLER)
                           DATALOCATION(ANY)    EXECCKEY(CICS)
```

The RDO statements for CICS AOR1 and AOR2 are as follows:

```
DEFINE TRANSACTION(LSMC)    REMOTESYSTEM(SMOR)    GROUP(RMTSMIF)
DEFINE PROGRAM(LSMCFLH)    LANGUAGE(ASSEMBLER)    GROUP(RMTSMIF)
```

These definitions will allow user transactions to be defined on the AOR or SMOR regions and access StorHouse. Additionally, in pseudo-conversation mode, the different transactions involved in the user's applications can span across the three regions during the StorHouse session. The only requirement is that the CICS terminal ID, EIBTRMID, remains the same.

Sample RDO and PPT/PCT definitions are included in the SAMPLES dataset in members CICSEDA, CICSPPPT, and CICSPPCT.

## Updating the DFHRPL DD Concatenation

Update the DFHRPL DD concatenation to include the LSMLOAD load library.

## LSMS DD Statement

An optional //LSMSssnm DD DUMMY statement may be defined in the CICS startup JCL. The ssnm is a 4-character default StorHouse Subsystem name. If an application

program does not specify a Subsystem name in the parameter list for the CONNECT function, program LSMUXSSN searches the TIOT DD entries for a DD that begins with LSMS. The next four characters of the DDNAME are used as the requested subsystem name.

# 7

## CICS Interface Installation (LSM1830)

---

Step 5: CICS Preparation

## Host Software PTF/APAR Installation

Corrections to SGI host software are distributed in standard IBM PTF Relfile tape format and are maintained using SMP/E. The tape is standard label and the volume serial number is always LSMPTF. The format of a standard SGI tape is shown in Table 8-1.

**Table 8-1: Standard SGI Tape Format**

File	Dataset Name	Contents
1	SMPMCS	PTF MCS statements (sequential)
2-n	Standard relfile name	Relfiles containing new modules

ISPF SMP/E Dialogs may be used for RECEIVE, APPLY, and ACCEPT commands.

### Step 1: Execute the SMP/E RECEIVE

Use the sample JCL in Figure 8-1 to receive the PTFs and exception SYSMOD data. Member LSMPTF in the SAMPLES JCL dataset contains this JCL. You must use the same CSI used to install the SGI host software that is being corrected. The LSMSMPE cataloged procedure was set up when the host software was initially installed.

```
//PRFRECJV JOB ...
//LSMSMPE EXEC LSMSMPE
//SMPCNTL DD *
  SET BOUNDARY (GLOBAL).
  RECEIVE SOURCEID (sourceid).
//SMPPTFIN DD DSN=SMPMCS,VOL=(,RETAIN,SER=LSMPTF),
//           UNIT=unit,                                     <===
//           LABEL=(1,SL),DISP=SHR
//SMPHOLD  DD DUMMY
```

**Figure 8-1: Executing SMP/E RECEIVE**

SOURCEID is recommended (multiple PTF/CUM tape consideration).

## Step 2: Execute the SMP/E APPLY

Use the sample JCL in Figure 8-2 to APPLY the PTFs. You must use the same CSI used to install the SGI host software that is being corrected. The LSMSMPE cataloged procedure was set up when the host software was initially installed.

```
//PTFAPPLY JOB ....  
//LSMSMPE EXEC LSMSMPE  
//SMPCNTL DD *  
  SET BOUNDARY(target zone name). <===  
  APPLY SOURCEID(sourceid) COMPRESS(ALL). <===
```

**Figure 8-2: Executing SMP/E APPLY**

You must set the target zone to the one used during the install of the SGI host software.

## Step 3: Verify the PTFs

Run any tests required to verify that the PTFs produce the required results. This should be done before accepting the PTFs so they can be rejected, if necessary.

## Step 4: Execute the SMP/E ACCEPT

Use the sample JCL in Figure 8-3 to ACCEPT the PTFs. You must use the same CSI used to install the SGI host software that is being corrected. The LSMSMPE cataloged procedure was set up when the host software was initially installed.

```
//PTFACCT JOB ....  
//LSMSMPE EXEC LSMSMPE  
//SMPCNTL DD *  
  SET BOUNDARY(distribution zone name). <===  
  ACCEPT SOURCEID(sourceid) COMPRESS(ALL). <===
```

**Figure 8-3: Executing SMP/E ACCEPT**

You must set the distribution zone to one used during the install of the SGI host software.



---

# Starting and Shutting Down the StorHouse Subsystem

The StorHouse Subsystem is a standard MVS Subsystem used to transport requests and responses between user applications and StorHouse. The StorHouse Subsystem must be active before any StorHouse services can be requested.

## Starting the StorHouse Subsystem

The StorHouse Subsystem is started by executing the program LSMLSMS from an authorized library. LSMLSMS accepts one parameter, the name of the StorHouse Subsystem (maximum of four characters) to be started. A default StorHouse Subsystem name of LSMS will be used if no parameter is specified. It is not required that the requested Subsystem already be defined in the active IEASSNxx member of SYS1.PARMLIB since LSMLSMS will dynamically build all needed control blocks during initialization.

The Subsystem should always be run as a started task since it establishes a cross-memory environment, and MVS will terminate the address space when the step completes. If the Subsystem is run as a batch job, the initiator will be terminated when the Subsystem is shut down.

## Shutdown Processing

The StorHouse Subsystem may be shut down using either the MVS STOP (P) command or the operator command, SHUTDOWN, described in Chapter 11, “StorHouse Subsystem Operator Commands.”

There are three types of shutdown requests that can be made, depending on the status of the StorHouse Subsystem: normal, quick, or forced. These types are explained in the following sections.

## Normal Shutdown Processing

Normal shutdown processing occurs when task processing is allowed to complete normally. The StorHouse Subsystem will not terminate until all sessions are completed. New sessions may be started during this time. A normal shutdown request is made by entering the SHUTDOWN operator command with no operands or by entering a single MVS STOP (P) command.

## Quick Shutdown Processing

Quick shutdown processing occurs when any active session must be ended as soon as possible and no new sessions should be allowed to start. Each active subtask will be posted for immediate termination but will be allowed to terminate on its own. A quick shutdown request is made by entering the SHUTDOWN operator command with the QUICK operand or by entering a second MVS STOP command.

## Forced Shutdown Processing

Forced shutdown processing occurs when all active sessions must be ended immediately and no new sessions should be allowed to start. Each active task is ABENDED by the dispatcher and detached immediately. A forced shutdown request is made by entering the SHUTDOWN operator command with the FORCE operand or by entering a third MVS STOP command. A forced shutdown request will not be accepted until a quick shutdown has been attempted.

## Active StorHouse Subsystems

Several StorHouse Subsystems may be active at one time, but LSMLSMS will not allow more than one Subsystem with the same name to be active at any one time. LSMLSMS determines that the requested Subsystem is active by locating the StorHouse Subsystem extension of the SSCVT and checking an active indicator in that extension. If the active indicator is found to be on, the current initialization effort will be terminated. Whenever the associated Subsystem is shut down, the active flag will be turned off so it can be restarted at a later time.

## Abnormal StorHouse Subsystem Termination

In the event that the StorHouse Subsystem should abnormally terminate or be canceled, every effort is made to turn the active flag off, but this may fail under certain circumstances. If this should happen, the Subsystem will not restart until the system is re-IPLed or the utility LSMCLEAR is executed. LSMCLEAR verifies that every active StorHouse Subsystem in the system is associated with a valid MVS Address Space Control Block (ASCB) and will turn off the active indicator for each Subsystem where this is not true.

If initialization should fail for any reason, an MVS WTO will be issued explaining the problem. Also, if the StorHouse Subsystem should ABEND, a LOGREC record will be written, and an SVC dump will be taken.

## StorHouse Subsystem and Callable Interface Functions

The StorHouse Subsystem and Callable Interface perform the following functions:

- The Callable Interface accepts the request from the user application. The application makes a call to the entry point LSMCALL or LSMCICS and passes parameters for a request.
- The Callable Interface transforms the request into an order. The Interface validates the parameters and builds an order with the information passed by the application.
- The Callable Interface places the order on the StorHouse Subsystem's active queue for processing. The Interface issues a space switch Program Call (PC) to a routine in the user's address space that places the order on the Subsystem's active queue. The routine receives control in full cross-memory mode, supervisor state, and key 0. After placing the order on the queue, the routine returns control to the application in the state in which LSMCALL or LSMCICS was entered.
- The Subsystem sends the request to StorHouse for processing. The main control program (dispatcher) removes orders from the active queue on a first-come, first-serve basis. It passes each order to a subtask that processes that type of order. The subtasks also obtain information from the user's address space by direct calls to a cross-memory move routine. The dispatcher and all subtasks run in the Subsystem's address space in problem program mode and key. The dispatcher may enter supervisor state and key 0 to perform special functions, such as obtaining locks for queue processing.

- The subtasks communicate with StorHouse using the connection specified in the configuration table (see Chapter 12). The connection uses allocation of devices and EXCP-level I/O or OE sockets and communication over LAN (or WAN) networks.
- The Subsystem returns a response for each user application request. To return responses, the Subsystem schedules SRBs to the user's address space and uses space switch PC routines to move data to the user's data areas.

---

## Executing the Interactive Interface

The Interactive Interface is a TSO command processor that allows users to enter StorHouse Command Language requests and receive responses interactively.

### Invoking the Interactive Interface

The command name is SM1. SM1 can be entered as a standard TSO command as long as the LSMLOAD library is one of the user's STEPLIB datasets or is in the system LINKLIST.

If the LSMLOAD library is not available as a STEPLIB dataset or is not in the system LINKLIST, then the TSO CALL command must be used.

### Authorizing the SM1 Command

The SM1 command does not require authorization unless GET and/or PUT commands using DFDSS are used. GET and PUT also require the File Transfer Interface. The command can be authorized by adding the name SM1 to the table IKJEFTE2, which contains names of authorized TSO command processors. If the TSO CALL command is used, then the IKJEFTE8 table must also be updated. Refer to the IBM publication *OS/VS2 MVS System Programming Library: TSO* for details on how to update these tables.

Alternatively, the IKJTSOxx member of PARMLIB can be updated.

- In the AUTHCMD NAMES List, add the entry:

```
SM1 SM    +
```

- In the AUTHPGM NAMES List, add these entries:

```
SM1      +
LSM1T    +
LSM1U    +
ADRDSSU  +
```

- Add the same entries to AUTHTSF NAMES:

```
SM1      +
LSM1T    +
LSM1U    +
ADRDSSU  +
```

## SM1 Command Optional Parameters

Normally, when first invoked, the SM1 command prompts for a StorHouse account ID and the account password. Two optional parameters may be requested if the keyword PROMPT is also entered on the command line.

The optional parameters are the name of the StorHouse system with which the session should be started and the name of the StorHouse Subsystem to be used. If the PROMPT keyword is omitted, the defaults as specified in the configuration dataset are used. The Subsystem name must match the parameter passed to LSMLSMS. The name of the StorHouse system must match the SM\_NAME of one of the SM\_ID entries in the configuration dataset. (See Chapter 12, “Configuring the StorHouse Host Software.”) A user exit is available to control the parameters for which the Interactive Interface will prompt. (See Chapter 14, “Installation Exits.”)

# StorHouse Subsystem Operator Commands

The StorHouse Subsystem provides operator commands that may be entered at an MVS system console. These commands include the ability to:

- Display status of the Subsystem, including the trace and SMF recording facilities
- Display status of specific task or order
- Display status of Direct Connect sessions
- Re-read the Configuration file
- Stop/cancel tasks
- Control tracing (enable, disable, print)
- Shut down the Subsystem
- Start StorHouse operator message sessions.

This chapter discusses command recognition and lists the available StorHouse Subsystem operator commands.

## Command Recognition

The appropriate Subsystem recognizes a command in one of two ways. The command can be preceded by a command character as specified in the configuration dataset when the Subsystem was started, or it can be preceded with the name of the desired Subsystem as specified in the parameter field when the Subsystem was started.

The following examples specify the command to shut down the StorHouse Subsystem called LSMS. The first uses the character '%' as the command character; the second uses the Subsystem name LSMS.

- %SHUTDOWN
- LSMS SHUTDOWN

All responses are issued through the WTO SVC using the route codes specified in the configuration file.

## List of Operator Commands

StorHouse Subsystem operator commands follow:

CANCEL,TASK, task# CANCEL,TASK,task #,DUMP	<p>Cancels a specific task. This command causes the dispatcher to ABEND the specified task with a system ABEND code 222. If ,DUMP is specified, the ABEND code is 122 and a dump is produced. All processing is terminated without a chance of an orderly shutdown of currently active processing. The dispatcher then issues a message indicating that a task has ABENDED.</p>
DISPLAY,LINKS DISPLAY,LINKS, ALL	<p>Displays the status of all Direct Connect Links from the subsystem to the StorHouse systems. If ,ALL is specified, the units assigned to all links are listed. Otherwise, only the units assigned to major control links are listed.</p>
DISPLAY, ORDER(S) DISPLAY,ORDER, jobname	<p>Displays the overall order processing status, which includes the following general information:</p> <ul style="list-style-type: none"> <li>• Total number of orders placed on the active queue</li> <li>• Total number of orders taken from the active queue</li> <li>• Number of orders that were obtained dynamically (GETMAINed).</li> </ul> <p>It also displays the following information for each order either on the active queue or currently being processed or for the requested job:</p> <ul style="list-style-type: none"> <li>• Processing task number</li> <li>• Related user job name</li> <li>• Type of task</li> <li>• Type of queue holding the order</li> <li>• Order flags.</li> </ul>
DISPLAY,SMF	<p>Displays the SMF recording status, which includes the following general information:</p> <ul style="list-style-type: none"> <li>• Whether SMF recording is being performed</li> <li>• The selected SMF record-type number</li> <li>• The SMF record subtypes that are being collected.</li> </ul>
DISPLAY,STATUS	<p>Displays the Subsystem status, which includes the following information:</p> <ul style="list-style-type: none"> <li>• Subsystem initialization date and time</li> <li>• Dispatcher status</li> <li>• Shutdown processing level, if any.</li> </ul>
DISPLAY,TASK(S) DISPLAY,TASK, jobname DISPLAY,TASK,task number	<p>Displays the overall task processing status, which includes the following information:</p> <ul style="list-style-type: none"> <li>• Total number of tasks processed</li> <li>• Current number of active tasks</li> <li>• Highest number of concurrent task reached</li> <li>• Maximum number of concurrent tasks allowed</li> <li>• Number of times maximum tasks was reached.</li> </ul>



It also displays the following information for each currently active task or for the requested task:

- Task number
- Related user job name
- Type of task
- Task status
- The task number of any associated task. (An ID [data transfer] task is related to an IC [command link] task.)
- MUSAS identification string (for example, a CICS transaction.)

DISPLAY,TRACE	Displays the trace status, which includes the following general information: <ul style="list-style-type: none"> <li>• Whether tracing is currently being performed</li> <li>• The status of the Trace Writer task</li> <li>• The number of entries in a trace table</li> <li>• The SYSOUT class for printing trace data.</li> </ul>
RECONFIG	Re-reads the configuration file. If there are errors in the file, they will appear as if the software were reading the configuration file during start-up. RECONFIG will not bring down the StorHouse host software because of configuration file errors.
SDUMP	Dumps the StorHouse Subsystem. This command causes an SVC dump to be taken to a SYS1.DUMPxx dataset.
SHUTDOWN	Shuts down the StorHouse Subsystem normally. This command causes the StorHouse Subsystem to enter normal shutdown processing, as described in Chapter 9, “Starting and Shutting Down the StorHouse Subsystem.”
SHUTDOWN, FORCE	Forces the StorHouse Subsystem to shut down. This command causes the StorHouse Subsystem to enter forced shutdown processing, as described in Chapter 9, “Starting and Shutting Down the StorHouse Subsystem.”
SHUTDOWN, FORCE,DUMP	Forces the StorHouse Subsystem to shut down with a dump. This command is processed the same as the command immediately above. In addition, it produces a dump for each task that is terminated.
SHUTDOWN, QUICK	Shuts down the StorHouse Subsystem quickly. This command causes the StorHouse Subsystem to enter quick shutdown processing, as described in Chapter 9, “Starting and Shutting Down the StorHouse Subsystem.”
SMOPER,START, smname	Starts StorHouse operator message tasks. This command initiates a subsystem StorHouse operator subtask to establish communications with the StorHouse system identified by smname. The smname parameter must be a valid StorHouse identifier in the configuration file.
STOP,TASK,task#	Stops a specific task. This command posts the specified StorHouse Subsystem subtask for immediate termination. This should allow for an orderly shutdown of any processing currently active for the subtask. If the subtask fails to terminate within one

minute, the dispatcher issues a message indicating that a task has failed to terminate when requested.

**TRACE,OFF** Controls trace deactivation. This command deactivates tracing. Active tasks will continue to trace to an in-memory table, but the table will not be written when the task ends.

**TRACE,ON** Controls trace activation. This command activates tracing and optionally sets the number of trace table entries to the value of the “entries” parameter. This command  
**TRACE,ON,entries** also starts a previously inactive Trace Writer task.

**TRACEWRT,task#** Prints a task’s trace table immediately. This command schedules the Trace Writer to print the specified task’s trace table immediately. The task’s trace table is not altered in any way, and normal tracing activity for the task continues.

---

## Configuring the StorHouse Host Software

The StorHouse host software is installed with a default set of configuration parameters hard coded into the StorHouse program. To change the parameters to values that reflect your site's needs, a dataset containing parameter specifications must be built, and the name of the dataset must be supplied in the subsystem start PROC.

When the StorHouse host software is activated, it uses the values in the dataset named by the SMCONFIG DD statement in the StorHouse procedure to override the default parameter values. A sample configuration dataset is supplied in the SMCONFIG member of the SAMPLES dataset.

The configuration dataset consists of statements that contain configuration parameter values. The statements must have the following form:

KEYWORD = VALUE

KEYWORD is a configuration parameter identifier and VALUE is the new value assigned to the parameter. Keywords can be abbreviated to the shortest string that uniquely identifies the keyword.

There can be more than one statement per line. Statements must be separated by a comma, a semicolon, or a space. An exclamation point (or a vertical bar) is recognized as the start of a comment field; the rest of the line is ignored. KEYWORD and VALUE are translated to uppercase unless they are enclosed in quotes (""). Also, if a VALUE contains any statement delimiter, the VALUE must be enclosed in quotes ("").

An integer value in a configuration file cannot contain commas. A decimal point can be specified if the maximum/minimum/default values are defined as decimal numbers.

## StorHouse Identification

Multiple StorHouse systems can be connected to a single host. To allow a user to specify a particular StorHouse system, the host software maintains a list of StorHouse identifiers. These identifiers are specified with the configuration keyword `SM_NAME`. The keywords `SM_NETTYPE`, `SM_NETPARM`, `SM_HOSTID`, and `SM_LINKNAME` associate connectivity parameters with an `SM_NAME`.

### Internal Identification

The value associated with the `SM_NAME` keyword provides an internal name for a specific StorHouse system. A host application must supply this value to establish a session with the defined StorHouse system. When the `SM_NAME` keyword is encountered in a configuration dataset, another entry is added to the list of StorHouse systems that can be referenced by the host. The value associated with `SM_NAME` becomes the internal identifying name for that StorHouse system.

### External Identification

External identification of a StorHouse system is supplied through four configuration keywords: `SM_NETTYPE`, `SM_NETPARM`, `SM_HOSTID`, and `SM_LINKNAME`. These keywords are specified following every `SM_NAME` entry in the configuration dataset. `SM_HOSTID` and `SM_LINKNAME` are required. The other parameters assume default values if not specified.

### Operator Console Support Services Identification

In addition to the internal and external identification of a StorHouse system, other StorHouse-specific information may be required. The StorHouse operator console support services require session (account/password) and startup strategy parameter values for each StorHouse system whose console messages will be displayed on the MVS host console. The keywords `SM_OPER_ACCT`, `SM_OPER_PASSWORD`, and `SM_OPER_START` allow specification of these parameter values. Like identification keywords, console services keywords must follow the `SM_NAME` entry. However, since `SM_OPER_ACCT`, `SM_OPER_PASSWORD`, and `SM_OPER_START` have default values, their specification is optional.

## StorHouse Identification Parameter Rules

The hard coded configuration parameters contain an entry for a default StorHouse system. For an installation with a single StorHouse system and default network naming, specifying another entry is unnecessary since the StorHouse system is identified by default. The following rules apply when specifying StorHouse identification parameters:

- SM\_NAME, SM\_HOSTID, and SM\_LINKNAME must be specified, and each group of identifiers must begin with SM\_NAME.
- SM\_NETTYPE is optional and does not have to be specified if the network interface is Direct Connect.
- SM\_NETPARAM is required for Direct Connect configurations. See the SM\_NETTYPE description for the type of network being used.
- The three StorHouse console support services keywords SM\_OPER\_ACCT, SM\_OPER\_PASSWORD, and SM\_OPER\_START are optional. These keywords may be specified following SM\_NAME. If omitted, the StorHouse entry is given the default values associated with these keywords.
- When more than one SM\_NAME is specified in the configuration file, the first identifies the default StorHouse system to be accessed if the user specifies a blank SM\_NAME or an SM\_NAME that is not equal to any SM\_NAME value. (The *Callable Interface Programmer's Guide* explains how to specify an SM\_NAME with the CONNECT function.)
- When no SM\_NAME entries are supplied in the configuration dataset, the default StorHouse List entry is defined as follows:
  - SM\_NAME = DEFSM1
  - SM\_NETTYPE = DIRECT
  - SM\_NETPARAM = C300
  - SM\_HOSTID = SM\_SP
  - SM\_LINKNAME = CMD\_LINK
  - SM\_OPER\_ACCT = SERVICE
  - SM\_OPER\_PASSWORD = SERVICE
  - SM\_OPER\_START = AUTOMATIC

## Keywords

The keywords used for IBM MVS systems are listed in the following sections.

### COMMAND\_CHARACTER

Specifies the character that the StorHouse Subsystem will use to recognize operator commands. The use of the command character is described in Chapter 11, “StorHouse Subsystem Operator Commands.” Characters that may be specified as command characters are:

% : . & + > < @ \$ ? \* / - ¢ \_

The following characters are also valid for COMMAND\_CHARACTER but must be enclosed in quotes (for example, “!”):

| space / ( = ! “

- Value Type: Character String
- Default Value: Null
- Maximum String Length: 1

### CONFIG\_FILENAME

Specifies a DD name for the next external configuration dataset to be processed. After processing the parameters in the current configuration dataset, the system reads the parameters from the next CONFIG\_FILENAME. If no CONFIG\_FILENAME is specified or the system cannot open the next configuration DD name, parameter processing ends. If specified, a DD statement with the specified name must be added to the SM1LSMS procedure. This parameter allows several configuration files to be chained together. The maximum number of chained files allowed is eight.

- Value Type: Character String
- Default Value: SMCONFIG
- Maximum String Length: 64

### DEF\_XM\_BUFFER

Defines the size of the buffer used to move user data records between the user’s address space and the StorHouse Subsystem. If the user program requests a buffer of zero, this buffer size is used. The value must be coordinated with MAX\_XM\_BUFFER.

- Value Type: Integer
- Default Value: 16000
- Minimum Value: 0
- Maximum Value: 16000000

### EXTRA\_FRAME\_BUFFERS

Limits the amount of storage used for buffers for a single data transfer operation. Each buffer is large enough to contain one frame, where a frame is the standard unit transferred between the host and StorHouse. A frame is 31,744 bytes long. A minimum of two frame buffers is required for any transfer; additional memory will be allocated to frame buffers or to host access method use based on this parameter.

- Value Type: Integer
- Default Value: 0

- Minimum Value: 0
  - Maximum Value: 255
- FILE\_SYSTEM\_TYPE** Sets the File System Type (FST) used for a PUT operation if the user specifies neither the /ASCII nor the /BINARY option in a command, and the particular dataset does not force a specific FST. The default value produces the best transfer performance, but the StorHouse files can be retrieved only by an IBM MVS host.
- NATIVE is an alias for DF\_DSS.
- Value Type: Keyword
  - Default Value: DF\_DSS
  - Alternate Values: ASCII, BINARY, NATIVE
- MAX\_SESSIONS** Sets the maximum number of StorHouse Subsystem tasks which can run at any one time. The StorHouse Interactive Interface signon and the commands PUT and GET, as well as the StorHouse Callable Interface Commands CONNECT and OPEN, will cause a new StorHouse Subsystem task to be started.
- If the number of sessions requested by users exceeds the value specified for MAX\_SESSIONS, the requests are queued. If the MAX\_SESSIONS value is too large, the subsystem may run out of virtual memory and user session requests will fail. An 8000K subsystem REGION supports 36 sessions of any type or 48 sessions if half the sessions are command links (CONNECT requests).
- Value Type: Integer
  - Default Value: 64
  - Minimum Value: 1
  - Maximum Value: 10000
- MAX\_XM\_BUFFER** Defines the size of the buffer used to move user data records between the user's address space and the StorHouse Subsystem. The value limits the buffer size that can be requested by the user. Large values increase data throughput by limiting cross-memory request overhead, but virtual storage must be available in the StorHouse Subsystem address space and the user's address space for each active sequential data transfer operation. If a user requests a larger buffer, this value is used; no error message results. Note that the Interactive Interface requests a 128,000-byte buffer for DFSS transfers.
- Value Type: Integer
  - Default Value: 128000
  - Minimum Value: 0
  - Maximum Value: 1600000
- MODE** Sets the normal (PRODUCTION), debug (DEBUG), or test (TEST) mode for the interface process. DEBUG and TEST should be set only when required for software error tracking by SGI customer support personnel.

	<ul style="list-style-type: none"> <li>• Value Type: Keyword</li> <li>• Default Value: PRODUCTION</li> <li>• Alternate Values: TEST, DEBUG</li> </ul>
NET_CONN_RETRY_COUNT	<p>Specifies the number of times connection to StorHouse will be retried if the connection attempt fails with a “no such OFFER name” error. Systems with many users concurrently active to StorHouse should specify at least the default value.</p> <ul style="list-style-type: none"> <li>• Value Type: Integer</li> <li>• Default Value: 12</li> <li>• Minimum Value: 0</li> <li>• Maximum Value: 32767</li> </ul>
NETWORK_TYPE	<p>Sets the Network type used in command and data transfer operations. This keyword is a synonym for SM_NETTYPE, which is explained later in this chapter.</p> <ul style="list-style-type: none"> <li>• Value Type: Keyword</li> <li>• Default Value: DIRECT_CONNECT</li> <li>• Alternate Values: TCP</li> </ul>
ORDER_QUEUE_FACTOR	<p>Specifies the value to be multiplied by MAX_SESSIONS to determine the initial number of orders allocated for the StorHouse Subsystem. Each order is 1,280 bytes in length. In the event that more orders are required, the StorHouse Subsystem will dynamically allocate more orders. The Subsystem will send a message to the console informing the operator that this has been done. This factor allows tuning to avoid large numbers of dynamic allocations.</p> <ul style="list-style-type: none"> <li>• Value Type: Decimal Number</li> <li>• Default Value: 1.0</li> <li>• Minimum Value: 1.0</li> <li>• Maximum Value: 100.0</li> </ul>
ROUTE_CODE	<p>Sets the routing code(s) assigned to StorHouse Subsystem messages. The syntax of the ROUTE_CODE configuration parameter is identical to the syntax of the ROUTCDE parameter of the MVS Write To Operator (WTO) macro instruction. The WTO macro instruction is described in the IBM manual, <i>OS/VS2 MVS Supervisor Services and Macro Instructions Manual</i>.</p> <ul style="list-style-type: none"> <li>• Value Type: Integer Values enclosed in parenthesis and separated by commas</li> <li>• Default Value: 1</li> <li>• Maximum Value: 16</li> </ul>
ROUTE_SMOPER_MSGS	<p>Sets the routing code(s) assigned to StorHouse operator console messages when they are routed to an MVS console. The ROUTE_SMOPER_MSGS configuration parameter syntax is identical to the syntax of the ROUTCDE parameter of the MVS Write To Operator (WTO) macro instruction. The WTO macro instruction is described in the IBM manual, <i>OS/VS2 MVS Supervisor Services and Macro Instructions Manual</i>.</p>



- Value Type: Integer Values enclosed in parentheses and separated by commas
  - Default Value: 1
  - Maximum Value: 16
- SM\_HOSTID** Specifies the Network Host Name for the StorHouse Storage Processor (SP). This information is used to identify the Command Link to the network. For direct connections, specify `SM_SP` as the value for this parameter. For TCP/IP, specify either the system name of the StorHouse server or its IP address (using dotted decimal notation, e.g., 10.1.2.3).
- Value Type: Character String
  - Default Value: None
  - Maximum String Length: 64
- SM\_LINKNAME** Specifies the Network Link (OFFER) Name for the Command Link to the SP. Unless your installation has customized the StorHouse configuration, specify `CMD_LINK` as the value for this parameter for direct connect or `1200` for tcp.
- Value Type: Character String
  - Default Value: None
  - Maximum String Length: 8
- SM\_NAME** Allows the StorHouse system administrator to configure a maximum of 32 StorHouse systems to a single host. `SM_NAME`, followed by `SM_HOSTID` and `SM_LINKNAME`, constitute a set of configuration parameters that are required for each StorHouse system connected to a host.
- By specifying the StorHouse system name at `SIGNON (CONNECT)` time, a user can specify which StorHouse system is to be used.
- The `SM_OPER_ACCT`, `SM_OPER_PASSWORD`, and `SM_OPER_START` keywords may be used to specify StorHouse operator console support information. `SM_NETTYPE` and `SM_NETPARM` may be used to specify optional network configurations.
- Value Type: Character String
  - Default Value: None
  - Maximum String Length: 6
- SM\_NETPARM** Specifies network parameter information required by specific types of networks. See the `SM_NETTYPE` definition that follows. The maximum string length allows for future networks.
- Value Type: Character String
  - Default Value: Blank
  - Maximum String Length: 16

**SM\_NETTYPE** Specifies the type of network to be used for the command and data links to a particular StorHouse system. The value `DIRECT_CONNECT` (the default) specifies that StorHouse is directly connected through a block multiplexor or ESCON channel using SGI's Direct Connect feature. The value `TCP` specifies a socket interface through an IP network.

The meaning of the `SM_NETPARM` parameter depends on the value of `SM_NETTYPE`. For Direct Connect, `SM_NETPARM` specifies the device number (`ccuu`) of the unit used for allocation of sessions between the host subsystem and StorHouse. The value must be three or four characters, the device number in hex. The value must be exactly the lowest-numbered unit configured for the StorHouse interface board. This number is provided by the installation to SGI during installation planning. `SM_NETPARM` should always be specified for directly connected StorHouse systems because the default value (`C300`) is not likely to match the system configuration.

For TCP, `SM_NETPARM` is ignored.

- Value Type: Keyword
- Default Value: `DIRECT_CONNECT`
- Alternate Values: `TCP`

**SM\_OPER\_ACCT** Specifies the StorHouse account identifier to be used by StorHouse operator console support services. The account ID and the password supplied by `SM_OPER_PASSWORD` are used to establish a session with StorHouse, through which StorHouse operator messages are routed to MVS consoles. The consoles to which these messages are routed are controlled by the `ROUTE_SMOPER_MSGS` configuration keyword.

- Value Type: Character String
- Default Value: `SERVICE`
- Maximum String Length: 12

SM_OPER_PASSWORD	<p>Specifies the StorHouse password associated with the account ID, which is specified by SM_OPER_ACCT. See the SM_OPER_ACCT description.</p> <ul style="list-style-type: none"><li>• Value Type: Character String</li><li>• Default Value: SERVICE</li><li>• Maximum String Length: 32</li></ul>
SM_OPER_START	<p>Specifies the strategy for starting the StorHouse operator console support services session with StorHouse. If the default value is used, operator messages from StorHouse can be retrieved only if the MVS console operator enters a command to start a session with StorHouse.</p> <p>The value AUTOMATIC specifies that a StorHouse operator console message session should always exist. The session is started when the StorHouse Subsystem is started. If StorHouse is down, an MVS console message indicates this condition. When StorHouse is restored to service, the operator console message session is also restored.</p> <p>The value IF_USED specifies that a StorHouse operator console message session should only be established if a user actually performs some operation on StorHouse. Once established, the message session continues until StorHouse is either shut down or until the operator issues a command to stop the session.</p> <ul style="list-style-type: none"><li>• Value Type: Keyword</li><li>• Default Value: MANUAL</li><li>• Alternate Values: AUTOMATIC, IF_USED</li></ul>
SMF_INTERVAL	<p>Specifies the interval (in minutes) for writing interval SMF records. SMF recording must be enabled by specifying a record-type number (SMF_RECORD_ID) and by enabling specific SMF record subtypes (SMF_SUBTYPES). Interval recording is a future capability. Currently, the keyword/value is parsed but no interval SMF records are written.</p> <ul style="list-style-type: none"><li>• Value Type: Integer</li><li>• Default Value: 0</li><li>• Minimum Value: 0</li><li>• Maximum Value: 1440</li></ul>
SMF_RECORD_ID	<p>Specifies the record-type number for all SMF records written by the StorHouse Subsystem. Specifying a value of zero (0) disables StorHouse SMF recording. Values 1 - 127 are reserved for MVS system SMF records. Note that even if SMF_RECORD_ID is given a non-zero value, individual record subtypes (see SMF_SUBTYPES) must also be enabled to generate SMF records.</p> <ul style="list-style-type: none"><li>• Value Type: Integer</li><li>• Default Value: 0</li><li>• Minimum Value: 0 or 128</li><li>• Maximum Value: 255</li></ul>

- SMF\_SUBTYPES** Specifies the StorHouse SMF record subtypes that will be written. Subtypes are defined in Chapter 15, “StorHouse Subsystem SMF Recording.” In the current release, valid values are 1 - 4. Subtypes 5 - 16 are reserved for future releases. SMF\_SUBTYPES may be specified multiple times. For each specification, subtype numbers are added to the list of enabled subtypes. Specifying NONE clears the list. Specifying the same subtype number more than once results in an error.
- Value Type may be one of the following:
    - An integer
    - A set of integers enclosed in parentheses and separated by commas
    - The character string NONE
  - Default Value: NONE
  - Minimum Value: 1
  - Maximum Value: 16
- THIS\_HOST\_ID** Provides an identifier for the particular host system.
- Value Type: Character String
  - Default Value: DEFAULT
  - Maximum String Length: 8
- TRACE\_SYSOUT** Specifies the SYSOUT class to which trace output is written. The value must be an alphabetic (A-Z) or a digit (0-9).
- Value Type: Character string
  - Default Value: A
  - Maximum String Length: 1
- TRACE\_TABLE\_LENGTH** Sets the number of entries in the in-memory trace table. This is a circular table that retains the most recent trace entries.
- This is a debug feature and is normally left at the default value (0). If a trace is required for problem resolution, the value should be at least 400. Trace table entries are about 128 bytes each. Sufficient virtual storage must be available for the table size specified.
- Value Type: Integer
  - Default Value: 0
  - Minimum Value: 0
  - Maximum Value: 32767

## Sample Configuration File

Figure 12-1 shows a sample configuration file.

```
THIS_HOST_ID = PROD13
COMMAND_CHARACTER = ?
ROUTE_CODE = (2,11)
DEFXM_BUFFER = 65536
MAX_SM_BUFFER = 131072
SMF_RECORD_ID = 247
SMF_SUBTYPES = (1,2,3)
!
SM_NAME = COMREP
SM_NETTYPE = DIRECT
SM_NETPARAM = 4C0
SM_HOSTID = SM_SP
SM_LINKNAME = CMD_LINK
SM_OPER_START = IF_USED
!
SM_NAME = OLBILL
SM_NETTYPE = DIRECT
SM_NETPARAM = 2800
SM_HOSTID = SM_SP
SM_LINKNAME = CMD_LINK
SM_OPER_START = AUTOMATIC
!
SM_NAME = sths02
SM_NETTYPE = TCP
SM_HOSTID = 10.11.12.13
SM_LINKNAME = 1200
```

**Figure 12-1: Sample Configuration File**



---

# StorHouse Operator Message Facility

Operator messages generated by StorHouse are displayed on a VDT/keyboard device that is connected directly to StorHouse. Because this display may be remote from the host computer operations area, a StorHouse Subsystem facility is provided to copy StorHouse operator messages to the system console. With this facility, the operator can view all outstanding StorHouse operator messages and provide responses to those messages requiring replies.

## The Message Monitoring Subtask

The message facility operates by starting a message monitoring subtask in the subsystem. The subtask establishes a session with StorHouse and uses a privileged StorHouse command to declare that session a receiver of operator messages. The subtask then polls StorHouse for operator messages until the subtask is explicitly stopped by an system operator request or until StorHouse is shut down.

## Operator Messages

Operator messages that the monitoring subtask receives from StorHouse are written to the system console using standard WTO/WTOR system MACROs. The messages are formatted as follows:

```
XIO170I subsys-id SM-name SM-message-id text
```

These messages are non-deletable (the messages “stick” to the screen without scrolling off if the console is in the RD mode). The StorHouse request may be answered from the StorHouse console or from any StorHouse host console. When the message request is satisfied, the message is deleted by the monitoring subtask.

If the monitoring subtask detects a problem in StorHouse, the following message is displayed:

```
XIO169I subsystem-id SM-name STORAGE MACHINE OPERATOR MESSAGE  
SESSION TERMINATED
```

This message is deleted when a message monitoring session can be re-established with StorHouse.

## Subsystem Configuration Parameters and Operator Commands

StorHouse Subsystem configuration parameters and operator commands control StorHouse operator message monitoring. They allow the installation to select system console route codes for StorHouse operator messages and to determine startup/reset strategies for the monitoring sessions for specific StorHouse systems. Commands to the subsystem allow the system operator to explicitly start and stop monitoring sessions.

### Operator Message Routing

StorHouse operator message routing is controlled by two subsystem configuration parameters: `ROUTE_CODE` and `ROUTE_SMOOPER_MSGS`. The `ROUTE_CODE`, which also controls the routing of messages from the StorHouse Subsystem itself, supplies the console routing for messages associated with serious StorHouse failures. `ROUTE_SMOOPER_MSGS` supplies the console routing for all other (normal) StorHouse operator messages.

The message monitoring subtasks startup/restart can be controlled independently for each StorHouse system. The configuration parameter `SM_OPER_START` indicates when an operator message monitoring subtask is started by the following values:

- `AUTOMATIC` – indicates that an operator message monitoring subtask is started as soon as the subsystem is started.
- `IF_USED` – indicates that an operator message monitoring subtask is started when a StorHouse session is established by an application program.
- `MANUAL` – indicates that an operator message monitoring subtask is started by explicit operator request.



## Other Operator Configuration Parameters and Commands

Other configuration parameters used by the message monitoring subtask are `SM_OPER_ACCT` and `SM_OPER_PASSWORD`. These parameters are described in Chapter 12, “Configuring the StorHouse Host Software.”

The `SMOPER` command allows the system operator to explicitly control startup of StorHouse operator message monitoring subtasks. The `SMOPER` command is described in Chapter 11, “StorHouse Subsystem Operator Commands.”



---

## Installation Exits

The StorHouse host software provides exit points that allow installation control of Callable Interface use of supervisor service MACROs and user access to StorHouse accounts and files. Installation-supplied exit modules are loaded by the Callable Interface and invoked with standard branch-and-link calls. Default exit routines, supplied in source form, are assembled and linked as part of the normal install process.

The source code for the supplied exit modules is in the LSMSRC dataset. The MACROs required for their assembly are in the LSMMAC dataset. The members of the LSMSRC and LSMMAC datasets are stored during SMP ACCEPT processing. Prior to ACCEPT, these members will be in SMPSTS and SMPMTS.

The sample exit modules include documentation of parameters and specific entry/exit conditions.

## Control of Access to Supervisor Services

The batch/TSO Callable Interface use of supervisor services MACROs from the user application address space is performed through user exits. This allows the Interface to be tailored for use in different environments. The default exits for the Callable Interface (LSMUX---) use standard assembler MACRO calls. These MACROs are listed as follows:

- |          |  |
|----------|--|
| LSMUXCHK | Used for task synchronization and is called whenever the Interface must wait for the occurrence of an external event. The default LSMUXCHK (FMID LSM1800) implements the WAIT Macro. |
| LSMUXFM  | Used for memory management and is called to release dynamically-acquired memory. The default LSMUXFM (FMID LSM1800) implements the FREEMAIN Macro.                                   |

- LSMUXGM      Used for memory management and is called to acquire memory dynamically. The default exit LSMUXGM (FMID LSM1800) uses the RC form of the GETMAIN Macro.
- LSMUXLOA     Called to bring a load module into memory. The default exit LSMUXLOA (FMID LSM1800) implements the LOAD Macro.

## The Application Program Linked Module

The user application program interfaces with the StorHouse Subsystem through standard, high-level language subroutine (or function) calls to module LSMCALL in batch or TSO environments and to module LSMCICS in the CICS environment. LSMCALL or LSMCICS is link-edited with the user program. To minimize the impact of StorHouse Subsystem software updates and maintenance, these routines perform minimal processing of user calls. They serve primarily to pass control to LSMLOAD, the main interface module, which is dynamically loaded into the user's address space. (LSMCICS issues a CICS LINK to module LSMCFLH. LSMCFLH passes control to LSMLOAD, which is brought into storage using a CICS LOAD command.)

Both LSMCALL and LSMCICS are distributed in source code format. Because LSMCALL accesses the operating system services LOAD, GETMAIN, and FREEMAIN, it must be assembled and linked at the host installation. SMP/E performs this assembly and link automatically during APPLY processing.

SMP/E cannot assemble and link LSMCICS automatically, however, because this module requires CICS Assembler Command Level translator preprocessing. Therefore, LSMCICS is also distributed in load module format and requires reassembly only if the host installation's CICS release is incompatible with CICS Release 1.7.0, Level 8804. Refer to Chapter 7, CICS Interface Installation (LSM1830),” for complete information.

## Account and File Access Control

The Callable Interface has security/access exits to allow installation control of:

- Default Subsystem name determination
- Session connect account/password validation
- File OPEN parameter validation
- Interactive File Access

In addition, there is an exit in the Interactive Interface to allow control of the signon prompts. This exit allows the installation to specify which prompts are to be issued, to supply information for any of the prompts, and to control signon retry.

## Default Subsystem Name Determination

LSMUXSSN is called whenever the Callable Interface or the CICS Interface requires the default name for the StorHouse Subsystem. This exit is not invoked when a specific Subsystem name has already been supplied with the CONNECT request.

The default exit searches for a DDname beginning with LSMS. If one is found, the value returned is the last four characters of that DDname. Otherwise, the name LSMS is returned. The FMID for this exit is LSM1800.

## Session Connect Account/Password Validation

LSMUXSEC is called during the processing of a CONNECT call. This exit can validate the account, password, StorHouse name, and StorHouse Subsystem name information supplied by the application program and can substitute new values for those fields.

The default exit is a no-operation. The FMID for this exit is LSM1800.

## OPEN Security Exit (LSMUXSEO)

LSMUXSEO is called during processing of an OPEN, OPEN-SEQ, or OPEN-VRAM function and during interactive processing of file-referencing commands. LSMUXSEO can be used to verify or modify parameters such as MODE, FILE NAME, PASSWORDS, GROUP, FILE SET ID, and VOLUME SET ID.

LSMUXSEO can return an error message to the calling program or to the interactive user. This error message can be customized by your system programming staff.

The exit has a 2-byte parameter field that indicates the execution mode (either callable or interactive) and the function type.

The LSMSRC library of the LSM1800 FMID contains the source code for the LSMUXSEO exit routine. The prologue comments section of the LSMUXSEO source code contains specific information about the execution modes and functions types in the parameter field.

## Function Filter Exit (LSMUXSEF)

LSMUXSEF is called when a user invokes SM-CMD-INTF or any pass-through function. This exit can restrict the use of certain StorHouse functions.

A 1-byte parameter containing the StorHouse function type is passed to LSMUXSEF. A function type of S indicates execution of a StorHouse command/response function. A function type of P indicates execution of a pass-through function.

LSMUXSEF can determine whether a user is restricted from calling a specific function. If the user is restricted, LSMUXSEF passes a return code of 4 to the user. A return code of 4 indicates that the function called by the user did not complete successfully.

The LSMSRC library of the LSM1800 FMID contains the source code for the LSMUXSEF exit routine. The prologue comments section of the LSMUXSEF source code contains specific information about the execution modes and functions types in the parameter field.

The FMID for the FUNCTION FILTER exit is LSM1800.

## Interactive Interface Signon Prompt Control

LSMUXITP is called before the Interactive Interface prompts the user for signon information (account, password, StorHouse name, and Subsystem name). This exit can provide the signon information directly and can override prompting.

The default exit is a no-operation. The FMID for this exit is LSM1810.

## Changing the Installation Exits

The SGI supplied versions of the exits can be replaced using SMP/E USERMODS. Use the job stream in Figure 14-1 to receive your USERMOD after coding your site's version of the installation exit:

```
//LSMRECV JOB
//LSMSMPE EXEC LSMSMPE
//SYSIN DD *
  SET BOUNDARY (GLOBAL).
  RECEIVE SELECT(your USERMOD id).           <====
//SMPTFIN DD *
++USERMOD(your USERMOD id).                 <====
++VER (Z038)
      FMID(the exit's FMID).                 <====
++SRT(exit name)                             <====
      TXLIB(USEREXIT).
//SMPHOLD DD DUMMY
```

**Figure 14-1: Changing the Installation Exits**

After the USERMOD has been received, it can be applied and accepted using the LSMSMPE procedure. The DD statement USEREXIT must be added to the APPLY and ACCEPT jobs and must point to the PDS containing the new version of the exit. You do not need to specify the member name because it will be taken from the ++SRC statement.

---

# StorHouse Subsystem SMF Recording

This chapter discusses StorHouse Subsystem SMF recording. It describes SMF record types, SMF record information, the contents of specific SMF records, interval recording, and the SMF record format.

## SMF Record Types

The StorHouse Subsystem supports the generation of four types of SMF records: session, data transfer operation, global subsystem statistics, and StorHouse operator subtask.

- *Session* records (SMF\_SUBTYPES value of 1) record the number of user requests. These records are written after an interactive user signs off from StorHouse (SIGNOFF command), after a Callable Interface user disconnects from StorHouse (DISCONNECT function call), or after any command-link disconnect is signaled from the network.
- *Data transfer operation* records (SMF\_SUBTYPES value of 2) record file identification information, a count of each request type, and mode/method strings. These records are written after the data transfer is closed by the user or after data-link disconnect is signaled from the network.
- *Global subsystem statistics* records (SMF\_SUBTYPES of 3) detail the number of performed operations and maximum use levels. These records are written when the subsystem is shut down normally.
- *StorHouse operator subtask* records (SMF\_SUBTYPES of 4) contain the information to identify the target StorHouse and commands processed.

The different record subtypes are identified by a record subtype number. The SMF\_SUBTYPES configuration parameter allows specification of the subtypes to

be generated. The default value for this parameter (NONE) turns off all StorHouse Subsystem SMF recording.

To enable SMF recording, two things must be specified: an integer between 128 and 255 for the value of `SMF_RECORD_ID` and a valid subtype number(s) for the value of `SMF_SUBTYPES`. Specifying zero (0) for `SMF_RECORD_ID` or specifying NONE for `SMF_SUBTYPES` disables SMF recording. Refer to Chapter 12, “Configuring the StorHouse Host Software,” for more information about `SMF_RECORD_ID` and `SMF_SUBTYPES`.

## SMF Record Information

All SGI SMF record subtypes contain the following basic information:

- System indicator
- Record-type number (`SMF_RECORD_ID` value)
- Time and date the record was written
- System identification (SID parameter that identifies the computer system complex from which the record was written)
- Job name associated with the address space that generated the request to the StorHouse Subsystem (the user id for a TSO session)
- StorHouse session start time and date
- Begin time and date of the action for which the SMF record is written
- Record subtype identification
- Reason code (why the record was written: end of operation or time interval expired)
- Version identifier (changed whenever the format of the record is changed by software updates)
- Status code associated with operation end
- Subsystem name.



The session, data transfer, and StorHouse Operator subtask records also contain the following information:

- SM\_NAME parameter value (from the CONNECT function)
- StorHouse user identification (account-id) for the session
- Task-table number for the session (32-bit number used by the Subsystem to identify the session; can be used to group all records for a session uniquely)
- Task-table number of the task that generated the SMF record
- MUSAS ID (for example, a CICS transaction).

## Contents of Specific SMF Records

The contents of session, data transfer, and global system SMF records are explained in the following sections.

### Session Record

In addition to the basic information set, the session record contains counts of requests made to the StorHouse Subsystem. These requests are counted in the following categories:

- File OPEN requests (including pass-through OPEN)
- Pass-through requests, not including OPEN (the count of operations performed by a user through the Interactive Interface)
- StorHouse-Command-Interface requests.

### Data Transfer Record

In addition to the basic information set, the data transfer record contains:

- StorHouse file identification: file name, group name, file set name, and volume set name
- Mode value (forced to READ or WRITE for non-VRAM™ transfers)
- Method value (forced to NONVRAM for non-VRAM transfers)

- Counts for read, write, delete, and update operations.

For VRAM single-record operations, the counts are record counts. For sequential operations, data may be blocked when it is moved from the user address space to the Subsystem. In this case, the counts are block counts.

## Global System Records

The Subsystem information record contains:

- Subsystem start time
- Number of tasks started
- Maximum number of active tasks
- Total number of processed orders (function requests from the user and internal intra-task communication)
- Dispatcher CPU and WAIT time.

## Interval Recording

Interval recording is a future capability. Interval recording allows an installation to specify a time interval for forcing SMF record writes. An SMF interval record is written for any operation that does not end within the specified interval. These records are specifically marked as interval records, because accounting programs generally must ignore interval records unless the end-of-operation record is missing. The counters used to build end-of-operation records are not reset by writing an interval record.

The configuration parameter `SMF_INTERVAL` specifies the interval length, in minutes. For current releases this parameter should remain at its default value of zero to indicate that no interval records will be generated.

## SMF Record Format

An assembly language DSECT for all SGI-generated SMF records is included in the software distribution, as member `LSMSMFR` in the `LSMMAC` dataset (file 4 of the base product distribution tape).

## Sample JCL and Program Library

The LSM1800 distribution dataset SAMPLES contains the sample JCL, program, and control statement members. SAMPLES contains the following members:

- CICSCEDA – RDO CEDA definitions for the CICS Interface
- CICSPT – CICS Sample Program Table
- CICSPT – CICS Sample Processing Program Table
- CRUNTIME – JCL used to load the C runtime library
- DCTEST – JCL to run a validation test for a directly connected StorHouse system
- LSMALLOC – JCL used to allocate datasets required for the installation
- LSMCLEAR – JCL for the Subsystem Communication Vector Table Clear program
- LSMDDDEF – SMP/E CSI DDDEF UCLIN
- LSMPTF – Sample JCL to receive an SGI PTF
- LSMFMFUT – Sample program to format LSMS SMF records
- LSMFMPE – SMP/E execution procedure
- LSMUCLIN – JCL and SMP/E UCLIN for CSI initialization
- OVCOMM – A command to activate OVSAM at IPL
- OVLOAD – A procedure to load OVSAM modules from the distribution tape
- OVPTF – A jobstream to apply OVSAM PTFs

- OVSAMIVP – A job stream that performs a quick test to see if OVSAM is properly installed
- OVSTART – A procedure that activates OVSAM
- OVSTAT – A job stream to check the status of OVSAM in the system
- OVSTOP – A job stream to deactivate OVSAM
- OVSTRT – A job stream to activate OVSAM for testing
- RESTRICT – Restricted Rights Notice
- SAMPCOB – Sample COBOL Callable Interface program
- SAMPSECX – Sample security exit for CONNECT
- SAMPSEFX – Sample function filter security exit
- SAMPSEOX – Sample security exit for OPEN
- SHOWFILE – Sample assembler program for the LSMFI utility
- SMCONFIG – Sample configuration parameter dataset
- SMLSMS – Subsystem execution procedure
- SMSAMP – Sample assembler Callable Interface program; can be used both as an assembler example and as an Installation Validation Program for the base product.

---

## Direct Connect Error Codes

When an error is returned from the Direct Connect software, there are three status codes available to assist diagnosis of the cause of the failure. The first code is the return code from the Callable Interface function. The second is the FCN status, a value created by the session-level network interface layer. The third code is the network specific code from connection support routines.

### Callable Interface Function Codes

The return codes from the Callable Interface functions are documented in the *Messages and Codes Manual* in the StorHouse User Document Set. These codes are usually very generic. They only indicate non-recoverable failure of the communications link between the mainframe and StorHouse.

### FCN Status and Network-Specific Codes

To extract the FCN status and network-specific codes, the application program must retrieve the error text messages associated with the return code. The Callable Interface EMSG function must be called to obtain these text messages. In the case of the TSO Interactive Interface, these messages are retrieved and displayed on the user's terminal.

The messages returned by EMSG indicate the FCN status value by their message identifier and text. If the network specific code is relevant, it is also included.

# Messages

These messages are returned by EMSG for a Direct Connect failure. Except where noted, the code dddd is not relevant.

**XIG123 Unexpected Disconnect of Network Link, Codes=0xnxxx, dddd**

**Explanation:** If nxxx is 421D, the DCB for the interface device could not be opened.

**User Response:** Check for MVS messages in the SYSLOG that further explain this failure.

**Explanation:** If nxxx is 421E, an error has occurred during the allocation processing for the interface device.

**User Response:** The code dddd is the network-specific code that further defines the failure. These codes are listed and described in Table B-1. Although the error description may assist in determining the source of the problem, you may need to call your SGI customer support representative for additional assistance.

**Table B-1: Network Specific Codes**

Code	Error Description
0104	The StorHouse software is down. Restart the StorHouse software.
0112	Units are not available for an LSSCP. This error can be caused by running multiple StorHouse subsystems, allocating all units to one of the subsystems, and then attempting to run additional StorHouse sessions from the other subsystem.
0116	Maximum number of LSSCPs running. At most 8 subsystems can be supported by one StorHouse system.
0120	LSSCP unit I/O error.
0124	Invalid address in SSCP message. Verify that the logical address and physical address match in the IOCP.
0132	Invalid address in SSCP message. If you have just upgraded StorHouse software, there may be a prerequisite host software upgrade.
0168	LSSCP device OPEN failure. Check SYSLOG for IOS messages.
0172	LSSCP device DYNALLOC failure. Check SYSLOG for messages documenting the DYNALLOC return codes.
0176	PSSCP read I/O error. See 0180.
0180	PSSCP write I/O error. Check subsystem configuration for correct SM_NETPARM.
0184	PSSCP device OPEN failure. Check SYSLOG for IOS messages.

Table B-1: Network Specific Codes (continued)

Code	Error Description
0188	PSSCP device DYNALLOC failure. Check SYSLOG for messages documenting the DYNALLOC return codes. Check the subsystem configuration dataset to ensure that the value specified for SM_NETPARM is correct (cuu address of lowest number device on the StorHouse interface).
0192	ENQ failed attempting to begin PSSCP allocation process.
0212	Same as 0112 but occurred during an LSSCP restart.
0216	Same as 0116 but occurred during an LSSCP restart.
0220	LSSCP unit I/O error.
0232	Invalid revision level in SSCP message.
0268-0292	Same as 0168-0192, respectively, but occurred during an LSSCP restart operation. Investigate the original error that caused the restart attempt.
1004	The StorHouse software is down. This should be corrected by restarting the StorHouse software. The reason for the StorHouse failure should be investigated.
1008	The StorHouse timed out while processing an open request.
1032	Invalid revision level in SSCP message.
1036	Shutdown of the StorHouse subsystem has been requested or forced by other errors.
1040	StorHouse requested subsystem release of a device but the device is in use.
1056	Link device DYNALLOC failure. Check SYSLOG for messages documenting the DYNALLOC return code. If you run with the StorHouse interface devices off-line, verify that the PATHs to all of these devices are logically on-line.
2002	Link devices EXCP for read operation bypassed because the SSCP has indicated that StorHouse is down.
2063	Invalid record length from read operation (zero bytes were received from StorHouse).
20xx	I/O error from read operation. xx is the IOS POST code in decimal. Convert it to hex before looking up the IOS codes.
2101	Same as 2002 but the operation was a write.
2163	Invalid record length from a write operation. StorHouse did not accept all the data written.
21xx	Same as 20xx but the operation is a write.
22xx	Same as 20xx but the operation is write end-of-file.
2301	Link device EXCP for write operation bypassed because the subsystem SSCP indicated that StorHouse is down.
2302	Same as 23301 but operation is a read.
xx32	StorHouse received invalid function code xx from the subsystem.

**XIG124I Network Session Limit Exceeded**

**Explanation:** An internal logic error in the Direct Connect facility. Indicates that the SSCP task request queue is empty and the link task is unable to communicate a request to the SSCP task.

**User Response:** Call your SGI customer support representative for assistance.

**XIG125I Network Function Logical Error, Codes=0xnxxx, dddd**

**Explanation:** An internal logic error in the Direct Connect facility.

- If nxxx is 4222, FCN caller specified a record length of zero or a record length too large for the type of connection.
- If nxxx is 4223, an internal processing sequence error occurred. Either the LCB is already open and the FCN caller is requesting open, close is requested but the LCB is not open, read/write is requested but the LCB is not open, or check is requested but the LCB is not open or is not busy.
- If nxxx is 4224, virtual memory for an LCB cannot be allocated.

**User Response:**

- If nxxx is 4222 or 4223, call your SGI customer support representative for assistance.
- If nxxx is 4224, try increasing the StorHouse Subsystem region size. If this does not correct the problem, contact your SGI customer support representative.



# Index

## A

access exits 14-2  
 active StorHouse subsystems 9-2  
 allocating  
   required datasets for host subsystem 4-2  
 application program linked module 14-2  
 Assembler programming language 1-3  
 authorizing the SM1 command 10-1  
 AUTOMATIC value for SM\_OPER\_START keyword  
   13-2

## B

building the configuration file 4-5

## C

C programming language 1-1, 1-3  
 Callable Interface  
   description 1-1  
   function codes 17-1  
   functions 9-3  
 CANCEL,TASK,task# command 11-2  
 changing the installation exits 14-4  
 CICS Distributed Transaction Processing 7-3

CICS Interface Installation  
   executing SMP/E ACCEPT command 7-2  
   executing SMP/E APPLY command 7-2  
   executing SMP/E RECEIVE command 7-2  
   preparation  
     defining the CICS interface programs 7-3  
     LSMS DD statement 7-4  
     SMP/E load libraries 7-3  
     updating the DFHRPL DD concatenation 7-4  
     verifying the SMP/E installation 7-2  
 CICS Interface, description 1-3  
 CICSCEDA member in SAMPLES dataset 16-1  
 CICSPT member in SAMPLES dataset 16-1  
 CICSPT member in SAMPLES dataset 16-1  
 COBOL II programming language 1-3  
 COBOL programming language 1-1, 1-3  
 COMMAND\_CHARACTER keyword 12-4  
 commands  
   MVS STOP (P) 9-2  
   MVS WTO 9-3  
   SHUTDOWN 9-2  
   SHUTDOWN /FORCE 9-2  
   SM1 10-1  
   SMOPER 13-3  
 commands, operator  
   CANCEL,TASK,task# 11-2  
   DISPLAY,LINKS 11-2  
   DISPLAY,ORDER(S) 11-2  
   DISPLAY,SMF 11-2  
   DISPLAY,STATUS 11-2  
   DISPLAY,TASK(S) 11-2  
   DISPLAY,TRACE 11-3  
   RECONFIG 11-3  
   SDUMP 11-3  
   SHUTDOWN 11-3  
   SHUTDOWN,FORCE 11-3

SHUTDOWN,FORCE,DUMP 11-3  
 SHUTDOWN,QUICK 11-3  
 SMOPER,START,smname 11-3  
 STOP,TASK,task# 11-3  
 TRACE,OFF 11-4  
 TRACE,ON 11-4  
 TRACEWRT,task# 11-4

components of host software 1-1

CONFIG\_FILENAME keyword 12-4

copying the StorHouse start procedure to PROCLIB 4-6

CRUNTIME member in SAMPLES dataset 16-1

## D

data transfer operation records 15-1, 15-3

datasets

LSMMAC 14-1

LSMSRC 14-1

DCTEST member in SAMPLES dataset 16-1

DEF\_XM\_BUFFER keyword 12-4

default subsystem name 14-3

Direct Connect

definitions for MVSCP/IOCP and HCD 3-1

error codes 17-1

SM\_NETPARAM identification parameter 12-3

StorHouse Subsystem and Callable Interface  
 functions 9-3

StorHouse Subsystem operator commands 11-2

DISPLAY,LINKS command 11-2

DISPLAY,ORDER(S) command 11-2

DISPLAY,SMF command 11-2

DISPLAY,STATUS command 11-2

DISPLAY,TASK(S) command 11-2

DISPLAY,TRACE command 11-3

## E

EMSG function for Callable Interface 17-1

ESCON channel connection of 224 I/O devices 3-2

examples

ESCON channel connection of 224 I/O devices 3-2

MVSCP/IOCP 3-2

parallel channel connection of 224 I/O devices 3-2

EXCP-level I/O 1-1

executing

Interactive Interface 10-1

SMP/E ACCEPT command

CICS Interface installation 7-2

File Transfer Interface installation 6-3

Host software PTF/APAR installation 8-2

Host Subsystem installation 4-5

TSO Interface installation 5-3

SMP/E APPLY command

CICS Interface installation 7-2

File Transfer Interface installation 6-2

Host software PTF/APAR installation 8-2

Host Subsystem installation 4-4

TSO Interface installation 5-2

SMP/E RECEIVE command

CICS Interface installation 7-2

File Transfer Interface installation 6-2

Host software PTF/APAR installation 8-1

Host Subsystem installation 4-4

TSO Interface installation 5-2

exits

LSMUXITP 14-4

LSMUXSEC 14-3

LSMUXSEF 14-3

LSMUXSEO 14-3

LSMUXSSN 14-3

external identification for StorHouse 12-2

EXTRA\_FRAME\_BUFFERS keyword 12-4

## F

FCN status and network-specific codes 17-1

File Transfer Interface installation

executing SMP/E ACCEPT command 6-3

executing SMP/E APPLY command 6-2

executing SMP/E RECEIVE command 6-2

verifying the installation 6-2

File Transfer Interface, description 1-2

FILE\_SYSTEM\_TYPE keyword 12-5

FMID Host Subsystem component 7-3

formats

- configuration dataset statements 12-1
- operator messages 13-1
- SMF records 15-4

function filter exit 14-3

Function Management ID format 2-1

functions for StorHouse host software 2-1

## G

global subsystem statistics records 15-1

global system records 15-4

## H

host software components 1-1

Host Software PTF/APAR installation

- executing SMP/E ACCEPT command 8-2
- executing SMP/E APPLY command 8-2
- executing SMP/E RECEIVE command 8-1
- verifying the installation 8-2

Host Subsystem installation

- allocating required datasets 4-2
- building configuration files 4-5
- copying StorHouse start procedure to PROCLIB 4-6
- executing SMP/E ACCEPT command 4-5
- executing SMP/E APPLY command 4-4
- executing SMP/E RECEIVE command 4-4
- initialize SMP/E CSI 4-3
- loading file 7 4-2
- loading the C runtime library 4-5
- optionally updating SYS1.PARMLIB 4-6
- setting up SMP/E procedure 4-2
- starting StorHouse subsystem 4-6
- updating SYS1.PARMLIB 4-5
- validating the installation 4-6
- verifying the SMP steps 4-4

## I

identifying operator console support services for

StorHouse 12-2

identifying the StorHouse system 12-2

IF\_USED value for SM\_OPER\_START keyword 13-2

initializing the SMP/E CSI 4-3

installation exits, general

- access control to supervisor services 14-1
- account control 14-2
- application program linked module 14-2
- changing 14-4
- file access control 14-2

installation exits, specific

- determining default Subsystem name (LSMUXSSN) 14-3
- function filter (LSMUXSEF) 14-3
- Interactive Interface signon prompt control (LSMUXITP) 14-4
- OPEN security (LSMUXSEO) 14-3
- session connect account/password validation (LSMUXSEC) 14-3

installation requirements 2-2

installing

- CICS Interface installation 7-1
- File Transfer Interface (LSM1810) 5-1
- File Transfer Interface (LSM1820) 6-1
- Host software PTFs and APARs 8-1
- Host Subsystem (LSM1800) 4-1

Interactive Interface

- authorizing SM1 command 10-1
- description 1-2
- executing 10-1
- functions 1-2
- invoking 10-1
- SM1 command optional parameters 10-2

interfaces

- Callable 1-1
- CICS 1-3
- File Transfer 1-2
- Interactive 1-2

internal identification for StorHouse 12-2

interval recording 15-4

invoking the Interactive Interface 10-1

IP connection

- requirements 3-4

SM\_NETPARM identification parameter 12-8

## K

keywords for IBM MVS systems

COMMAND\_CHARACTER 12-4  
 CONFIG\_FILENAME 12-4  
 DEF\_XM\_BUFFER 12-4  
 EXTRA\_FRAME\_BUFFERS 12-4  
 FILE\_SYSTEM\_TYPE 12-5  
 MAX\_SESSIONS 12-5  
 MAX\_XM\_BUFFER 12-5  
 MODE 12-5  
 NET\_CONN\_RETRY\_COUNT 12-6  
 NETWORK\_TYPE 12-6  
 ORDER\_QUEUE\_FACTOR 12-6  
 ROUTE\_CODE 12-6, 13-2  
 ROUTE\_SMOPER\_MSGS 12-6, 13-2  
 SM\_HOSTID 12-7  
 SM\_INTERVAL 12-9  
 SM\_LINKNAME 12-7  
 SM\_NAME 12-7  
 SM\_NETPARM 12-7  
 SM\_NETTYPE 12-8  
 SM\_OPER\_ACCT 12-8  
 SM\_OPER\_PASSWORD 12-9  
 SM\_OPER\_START 12-9, 13-2  
 SMF\_RECORD\_ID 12-9  
 SMF\_SUBTYPES 12-10  
 THIS\_HOST\_ID 12-10  
 TRACE\_SYSOUT 12-10  
 TRACE\_TABLE\_LENGTH 12-10

## L

loading

C runtime library 4-5  
 file 7 4-2

LSM1800 distribution tape 4-1  
 LSM1810 distribution tape 5-1  
 LSM1820 distribution tape 6-1  
 LSM1830 distribution tape 7-1  
 LSMALLOC member in SAMPLES dataset 16-1  
 LSMCALL module 1-1, 14-2

LSMCFLH module 14-2  
 LSMCICS module 14-2  
 LSMCICS subroutine 1-3  
 LSMCLEAR utility 9-3  
 LSMCLER member in SAMPLES dataset 16-1  
 LSMDDDEF member in SAMPLES dataset 16-1  
 LSMLOAD module 14-2  
 LSMLSMS program 9-1  
 LSMMAC dataset 14-1  
 LSMPTF member in SAMPLES dataset 16-1  
 LSMSMFUT member in SAMPLES dataset 16-1  
 LSMSMPE member in SAMPLES dataset 16-1  
 LSMSRC dataset 14-1  
 LSMSRC library 14-3, 14-4  
 LSMUCLIN member in SAMPLES dataset 16-1  
 LSMUXCHK macro 14-1  
 LSMUXFM macro 14-1  
 LSMUXGM macro 14-2  
 LSMUXITP exit 14-4  
 LSMUXLOA macro 14-2  
 LSMUXSEC exit 14-3  
 LSMUXSEF exit 14-3  
 LSMUXSEO exit 14-3  
 LSMUXSSN exit 14-3

## M

macros

LSMUXCHK 14-1  
 LSMUXFM 14-1  
 LSMUXGM 14-2  
 LSMUXLOA 14-2

MANUAL value for SM\_OPER\_START keyword 13-2  
 MAX\_SESSIONS keyword 12-5  
 MAX\_XM\_BUFFER keyword 12-5

message monitoring subtask 13-1  
 messages (Direct Connect error) 17-2  
 MODE keyword 12-5

#### modules

- LSMCALL 14-2
- LSMCFLH 14-2
- LSMCICS 14-2
- LSMLOAD 14-2

MVS missing interrupt handler (MIH) considerations 3-4  
 MVS STOP (P) command 9-2  
 MVS WTO command 9-3

## N

NET\_CONN\_RETRY\_COUNT keyword 12-6  
 NETWORK\_TYPE keyword 12-6

## O

OPEN security exit 14-3  
 operator commands. *See* commands, operator.  
 operator console support services for StorHouse, identifying 12-2  
 operator messages
 

- facility for StorHouse 13-1
- format 13-1
- routing 13-2

 ORDER\_QUEUE\_FACTOR keyword 12-6  
 OUPTF member in SAMPLES dataset 16-1  
 OVCOMM member in SAMPLES dataset 16-1  
 OVLOAD member in SAMPLES dataset 16-1  
 OVSAMIVP member in SAMPLES dataset 16-2  
 OVSTART member in SAMPLES dataset 16-2  
 OVSTAT member in SAMPLES dataset 16-2  
 OVSTOP member in SAMPLES dataset 16-2  
 OVSTRT member in SAMPLES dataset 16-2

## P

P function type 14-3  
 parallel channel connection of 224 I/O devices 3-2  
 PL/I programming language 1-1, 1-3  
 PROMPT keyword for SM1 command 10-2

## R

RDO statements for CICS regions 7-4  
 RECONFIG command 11-3  
 RESTRICT member in SAMPLES dataset 16-2  
 ROUTE\_CODE keyword 12-6, 13-2  
 ROUTE\_SMOPER\_MSGS keyword 12-6, 13-2  
 rules for specifying StorHouse identification parameters 12-3

## S

S function type 14-3  
 SAMPCOB member in SAMPLES dataset 16-2  
 sample StorHouse host software configuration file 12-11  
 SAMPLES dataset members
 

- CICSCEDA 16-1
- CICSPCT 16-1
- CICSPPT 16-1
- CRUNTIME 16-1
- DCTEST 16-1
- LSMALLOC 16-1
- LSMCLEAR 16-1
- LSMDDDEF 16-1
- LSMPTF 16-1
- LSMSMFUT 16-1
- LSMSMPE 16-1
- LSMUCLIN 16-1
- NETDEFS 16-1
- OUPTF 16-1
- OVCOMM 16-1
- OVLOAD 16-1
- OVSAMIVP 16-2
- OVSTART 16-2

- OVSTAT 16-2
- OVSTOP 16-2
- OVSTRT 16-2
- RESTRICT 16-2
- SAMPCOB 16-2
- SAMPSECX 16-2
- SAMPSEFX 16-2
- SAMPSEOX 16-2
- SHOWFILE 16-2
- SMCONFIG 16-2
- SMLSMS 16-2
- SMSAMP 16-2
- SAMPSECX member in SAMPLES dataset 16-2
- SAMPSEFX member in SAMPLES dataset 16-2
- SAMPSEOX member in SAMPLES dataset 16-2
- SDUMP command 11-3
- security exits 14-2
- session connect account and password 14-3
- session records 15-1, 15-3
- setting up the SMP/E procedure 4-2
- SHOWFILE member in SAMPLES dataset 16-2
- SHUTDOWN /FORCE command 9-2
- SHUTDOWN /QUICK command 9-2
- SHUTDOWN command 9-2, 11-3
- SHUTDOWN,FORCE command 11-3
- SHUTDOWN,FORCE,DUMP command 11-3
- SHUTDOWN,QUICK command 11-3
- shutting down the StorHouse subsystem
  - abnormal termination 9-3
  - forced shutdown 9-2
  - normal shutdown 9-2
  - quick shutdown 9-2
- SM\_HOSTID keyword 12-7
- SM\_LINKNAME keyword 12-7
- SM\_NAME keyword 12-7
- SM\_NETPARM keyword 12-7
- SM\_NETTYPE keyword 12-8
- SM\_OPER\_ACCT keyword 12-8
- SM\_OPER\_PASSWORD keyword 12-9
- SM\_OPER\_START keyword 12-9, 13-2
- SM1 command 10-1
  - authorizing 10-1
  - optional parameters 10-2
  - PROMPT keyword 10-2
- SMCONFIG member in SAMPLES dataset 16-2
- SMF records
  - basic information 15-2
  - data transfer 15-3
  - data transfer operation 15-1
  - format 15-4
  - global subsystem statistics 15-1
  - global system 15-4
  - session 15-1, 15-3
  - StorHouse operator subtask 15-1
- SMF\_INTERVAL keyword 12-9
- SMF\_RECORD\_ID keyword 12-9
- SMF\_SUBTYPES keyword 12-10
- SMLSMS member in SAMPLES dataset 16-2
- SMOPER command 13-3
- SMOPER,START,smname command 11-3
- SMSAMP member in SAMPLES dataset 16-2
- starting the StorHouse Subsystem 4-6, 9-1
- STOP,TASK,task# command 11-3
- StorHouse
  - identification 12-2
  - operator message facility 13-1
  - operator message routing 13-2
  - operator subtask records 15-1
- StorHouse host software
  - Callable Interface functions 9-3
  - configuration dataset 12-1
  - functions 2-1
  - installation requirements 2-2
  - keywords for IBM MVS systems 12-4
  - operator console support services identification 12-2
  - sample configuration file 12-11
  - StorHouse identification 12-2
  - StorHouse identification parameter rules 12-3
- StorHouse Subsystem

- command recognition 11-1
- description 1-1
- functions 9-3
- operator commands 11-1

Supervisor Services macros. *See* macros.

## T

THIS\_HOST\_ID keyword 12-10

TRACE,OFF command 11-4

TRACE,ON command 11-4

TRACE\_SYSOUT keyword 12-10

TRACE\_TABLE\_LENGTH keyword 12-10

TRACEWRT,task# command 11-4

TSO Interface installation

- executing SMP/E ACCEPT command 5-3
- executing SMP/E APPLY command 5-2
- executing SMP/E RECEIVE command 5-2
- verifying the installation 5-2

## U

updating

- SYS1.PARMLIB library 4-5

USERMODS call 14-4

utility, LSMCLEAR 9-3

## V

validating the session connect account and password 14-3

verifying

- PTFs 8-2
- SMP installation 4-4
- SMP/E installation
  - CICS Interface 7-2
  - File Transfer Interface 6-2
  - TSO Interface 5-2

