

HIPLEX MSCF V6.0

BS2000 Processor Networks

Comments... Suggestions... Corrections...

The User Documentation Department would like to know your opinion on this manual. Your feedback helps us to optimize our documentation to suit your individual needs.

Feel free to send us your comments by e-mail to:

manuals@ts.fujitsu.com

Certified documentation according to DIN EN ISO 9001:2008

To ensure a consistently high quality standard and user-friendliness, this documentation was created to meet the regulations of a quality management system which complies with the requirements of the standard DIN EN ISO 9001:2008.

cognitas. Gesellschaft für Technik-Dokumentation mbH

www.cognitas.de

Copyright and Trademarks

Copyright © Fujitsu Technology Solutions GmbH 2010.

All rights reserved.

Delivery subject to availability; right of technical modifications reserved.

All hardware and software names used are trademarks of their respective manufacturers.

Contents

1	Preface	11
1.1	Target group	14
1.2	Summary of the contents	14
1.3	Changes since the last version of the manual	16
1.4	Notational conventions	17
2	Basic information about the MSCF network	19
2.1	Hardware configuration	19
2.2	HIPLEX MSCF communication system	21
2.3	HIPLEX MSCF function network	24
3	Connection types and network types	27
3.1	LCS network	29
3.2	CCS network	30
3.3	Shared pubset network	31
3.4	XCS network	34
4	Applications and basic functions	41
4.1	General overview	41
4.2	Synchronization and serialization	43
4.3	Time in the network	46

4.4	Parallel HIPLEX	48
4.4.1	Network entry and exit by a processor	49
4.4.2	Configuration of the global storage	50
4.4.3	Use of global storage by DAB	52
4.4.4	Use of the global storage by GSVOL	54
4.4.5	Example of startup of parallel HIPLEX (two processors)	55
4.4.6	Error handling in the GSMAN subsystem	56
4.4.7	Reconfiguration of DAB cache areas in parallel HIPLEX	57
4.4.8	Reconfiguration of GS volumes in parallel HIPLEX	59
4.5	DMS functions in the network	60
4.5.1	Pubsets as file containers	60
4.5.2	Data management system and shared pubsets	64
4.5.3	Shared file system	65
4.5.4	Shared PLAM	67
4.6	Management of the memory hierarchy with HSMS	68
4.7	System-managed pubsets as switchover units	69
4.8	Support by HIPLEX AF	71
4.8.1	Live Monitor of HIPLEX AF	71
4.8.2	Standby mode with HIPLEX AF	73
4.9	Job assignment and monitoring	74
4.9.1	Job variables	75
4.9.2	Job initiation and termination	78
4.9.2.1	Access permission in the MSCF network (participant mode)	82
4.9.2.2	Reference to jobs in the MSCF network	85
4.9.3	Job checking	86
4.9.3.1	Conditional job control	86
4.9.3.2	Job and program monitoring	89
4.9.3.3	Job variables for monitoring the import and export of pubsets	92
4.9.4	Monitoring the network status	93
4.10	Job management and job handling with AVAS	96
4.10.1	Job distribution with AVAS in HIPLEX	98
4.10.2	AVAS environment in HIPLEX	99
4.11	HIPLEX MSCF and RFA (Remote File Access)	100
4.12	HIPLEX MSCF and SPD (Shareable Private Disk)	103

5	Installation and configuration	109
5.1	Installation of HIPLEX MSCF	110
5.2	Parameterization of MSCF	111
5.2.1	DSSM subsystem catalog	111
5.2.2	Processor identification	112
5.2.3	BCAM dependencies	113
5.2.4	MSCF configuration file	115
5.2.5	Global control parameters	119
5.3	Generation of the LCS network	129
5.4	Generation of the CCS network	130
5.5	Generation of the shared pubset network	131
5.5.1	Entering the shared pubsets in the MRSCAT	131
5.5.2	Creating a pubset as a shared pubset	132
5.5.3	Parameters for defining the shared pubset master	133
5.5.4	General information about shared pubset mode	135
5.5.5	Special operating modes	137
5.6	Generation of the XCS network	140
5.7	Parameterization of NSM	141
5.7.1	Configuration file	141
5.7.2	GS partition	141

6	Administration of MSCF networks	143
6.1	Starting HIPLEX MSCF	144
6.2	Establishing connections	147
6.3	Clearing down connections	148
6.4	Terminating HIPLEX MSCF	149
6.5	Displaying MSCF connections	150
6.6	Shared pubset network	151
6.6.1	Setting up a shared pubset network	151
6.6.2	Clearing down a shared pubset network	155
6.6.3	Retrieving information on shared pubsets	156
6.6.4	Operating states of a shared pubset	157
6.7	XCS network	158
6.7.1	Entry/re-entry to the XCS network	158
6.7.2	Leaving the XCS network	160
6.7.3	Handling an XCS pubset	161
6.7.4	Displaying the XCS configuration	161
7	Error recovery	163
7.1	Email notification about critical states	163
7.2	Analyzing the system status	164
7.3	Server task bottleneck	166
7.4	Communication bottleneck	167
7.5	Failure of BCAM connections	168
7.6	Loss of connection in an LCS/CCS network	168
7.7	Connection clear-down/loss in a shared pubset network	169
7.8	Loss of connection in an XCS network	171
7.9	Failure of an LCS/CCS partner	172
7.10	Failure of a processor in a shared pubset network	172
7.11	Termination of the master change import	174
7.12	Manual removal of a crashed sharer	174
7.13	Failure of an XCS sharer	175

7.14	Time elapsed until crash recovery	177
7.15	Operation during crash recovery	178
7.16	Erroneous partner crash detection	179
7.17	Rejection of a partner's entry to the network	180
7.18	Input/output errors when accessing the watchdog file	181
7.19	Failure of the watchdog file	182
7.20	Disk monitoring blocks	183
7.21	Cluster Recovery Lock (CRL)	183
7.22	Abnormal termination of HIPLEX MSCF	184
7.23	Disconnection of a processor from the XCS network	184
7.24	Abnormal termination of an MSCF system task	185
7.25	Termination of MSCF by a registered function	185
7.26	System termination by MSCF	186
7.27	Timeout of reconfiguration	186
7.28	Hanging shared pubset occupation	187
7.29	Hanging XCS occupation	187
7.30	Error recovery for GS caches in parallel HIPLEX mode	187
8	Performance and optimization	189
8.1	Software Monitor openSM2	189
8.1.1	Prerequisites for implementing SM2 in the MSCF network	190
8.1.2	Users in the MSCF network	190
8.1.3	Screen output	191
8.1.4	Standard SM2 output file	191
8.1.5	Restrictions	191
8.1.6	Monitoring programs and reports	192
8.2	Performance aspects for shared pubsets	193
8.3	Performant File Access (PFA) for shared pubsets	194
8.4	Emulation of shared pubsets in global storage (GSVOL)	200

9	Commands	201
	MODIFY-MSCF-CONNECTION	
	Modify partner-specific monitoring parameters	203
	MODIFY-MSCF-ENVIRONMENT	
	Modify HIPLEX MSCF environment	206
	MODIFY-NSM-ENVIRONMENT	
	Modify NSM configuration parameters	212
	RELEASE-CLUSTER-RECOVERY-LOCK	
	Release the Cluster Recovery Lock (CRL) manually	214
	SET-MSCF-ENVIRONMENT	
	Define MSCF environment	216
	SET-NSM-ENVIRONMENT	
	Define NSM configuration parameters	222
	SET-XCS-PUBSET	
	Declare shared pubset as the XCS pubset	225
	SHOW-MSCF-CONFIGURATION	
	Show information on HIPLEX MSCF configuration	227
	SHOW-NSM-CONFIGURATION	
	Show information on state of NSM and its partners	277
	SHOW-SHARED-PUBSET	
	Request information on shared pubsets	285
	SHOW-XCS-OCCUPATION	
	Show occupation information for TU tasks using XCS	290
	START-MSCF-CONNECTION	
	Dynamically establish MSCF connection to partner system	293
	START-SUBSYSTEM	
	Activate subsystem	299
	STOP-MSCF-CONNECTION	
	Clear down MSCF connection to partner system	303
	STOP-SUBSYSTEM	
	Deactivate subsystem	305
10	Macros	309
	DPSHSHP	
	Output information on current shared pubset configuration	310
	MCSINFO	
	Output information on current MSCF configuration	317

11	Appendix	333
11.1	MSCF interface in /ENTER-JOB and /ENTER-PROCEDURE	333
11.2	Summary:	
	generation and operation of an MSCF network	337
11.2.1	General preconditions for setting up an MSCF network	337
11.2.2	Additional requirements for the CCS network	338
11.2.3	Additional requirements for the shared pubset network	339
11.2.4	Additional requirements for the XCS network	339
11.2.5	Additional requirements for parallel HIPLEX	339
	Glossary	341
	Abbreviations	349
	Related publications	351
	Index	353

1 Preface

The HIPLEX product family (HIPLEX¹ = **highly integrated system complex**) implements operational, capacity and availability networks for BS2000/OSD servers.

As a member of this family, HIPLEX MSCF (MSCF = **M**ultiple **S**ystem **C**ontrol **F**acility) provides the necessary infrastructure for capacity and availability sharing, as well as the basic functions for distributed applications.

In HIPLEX MSCF a **BS2000 processor** (processor for short) is understood to be a BS2000/OSD server with the BS2000/OSD (native) operating system or a VM2000 guest system on a BS2000/OSD server with VM2000.

In other literature the term “BS2000 system” (system for short) is also used.

Fundamental to an MSCF network is the communication of the processors in it based on BCAM transport links. Jobs involving the execution of functions and check messages for monitoring the network are exchanged between the processors.

Within the MSCF network, the shared pubset network is the most important network type. In addition to the communication link, in this type of network all the participating processors have access to shared disks, the shared pubset. The processors in the network monitor each other by means of two independent data routes, i.e. with the aid of the shared disks on the one hand and communication paths on the other, so that a processor crash can be reliably detected by the redundancy of the monitoring paths. If a processor crashes or a communication path fails, appropriate reconfiguration measures guarantee the continued functionality of the shared pubset network.

The XCS network is an extension of the shared pubset network. Synchronization mechanisms for all processors permit the management of globally available resources and the operation of distributed applications with data access to the shared data volumes.

Each processor in an MSCF network is autonomous. Systems support for each processor decides whether, when and for how long each processor will participate in the network. This ability to configure the network dynamically gives it a high degree of flexibility.

¹ HIPLEX[®] is a registered trademark of Fujitsu Technology Solutions GmbH

The mechanisms for detecting a processor failure are an integral part of the shared pubset and XCS network and also of the CCS network without shared pubsets. They allow the implementation of standby configurations which can be used to minimize application downtimes. The implementation of this kind of configuration is supported by another member of the HIPLEX product family, the HIPLEX Availability Facility (HIPLEX AF). HIPLEX AF also offers a mechanism for quick and reliable crash detection, the Live Monitor, see [page 71](#).

The HIPLEX MSCF architecture

The figure below provides an overview of the architecture of HIPLEX MSCF and the basic functions and applications that are based on it.

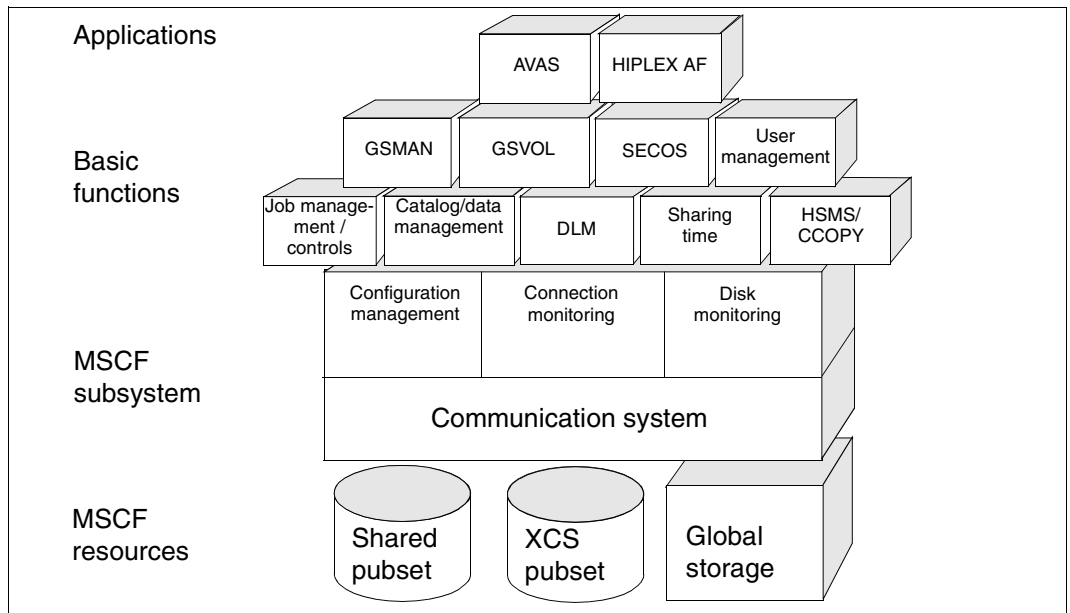


Figure 1: Architecture of the HIPLEX MSCF basic functions and applications

The product HIPLEX MSCF consists of the following DSSM subsystems:

- MSCF, responsible for communication and configuration monitoring
- NSM, DLM component (distributed lock management), implements the lock management across several systems
- XCS-TIME, responsible for processor time.

The following basic functions are grouped together in the MSCF subsystem:

- The *communication system* provides interfaces for starting and terminating the MSCF network on the local processor and for establishing and clearing down MSCF connections to partner processors. It also supports a communication interface to the partner processors.
- The *connection monitoring system* checks that the communication between each local processor and its partners is functioning properly. If necessary, it implements measures to reestablish connections that have been interrupted.
- The *configuration management system* manages the processors that are linked into an XCS network and coordinates any necessary reconfiguration measures. It also manages the shared pubset partners of the local processor.
- The *disk monitoring function* monitors the processor's shared pubset partners by means of their vital-sign messages written to the shared pubsets.

The following operating system functions and software products use the MSCF communication interface to extend their shared pubset and XCS network functionality:

- job management and checking
- catalog and data management
- HSMS with CCOPY
- user management (SRPM)
- openSM2
- SHC-OSD
- SPACEOPT
- HIPLEX-AF
- MAREN
- products of the SECOS security package

The XCS network offers the following functions:

- distributed lock management (DLM)
- time spent in the network
- management and use of a shared GS (parallel HIPLEX)
- disk access buffer (DAB)

The basic functions are used by application systems to support network-wide applications:

- AVAS supports job networks within the network, i.e. allows the load to be distributed over several systems
- HIPLEX AF offers the option of implementing standby configurations.

1.1 Target group

This manual is intended both for nonprivileged users of the BS2000/OSD operating system and for systems support staff. It describes the generation, administration, operation and application of HIPLEX MSCF.

Familiarity with the BS2000 system is a prerequisite.

1.2 Summary of the contents

This manual is divided into several chapters and indices:

- Chapter *Basic information about the MSCF network* explains the layers within the architecture of an MSCF network.
- Chapter *Connection types and network types* explains the LCS, CCS and the shared pubset and XCS networks. Connections, partner types, applications, special features and restrictions of the different network types are described in this chapter as well as entry and re-entry to and exit from the XCS network. Finally, it explains the XCS reconfiguration.
- Chapter *Applications and basic functions* deals with the basic functions and applications that are available.
- Chapter *Installation and configuration* tells you about the essential prerequisites for the installation of HIPLEX MSCF and features generation instructions for the various network types.
- Chapter *Administration of MSCF networks* gives you an overview of all the supported administration commands and explains how to start and terminate HIPLEX MSCF and establish and clear down connections. The MSCF connection messages, setting up and clearing down the LCS, CCS, the shared pubset and XCS network, fetching information and the handling of an XCS pubset are also dealt with in this chapter.
- Chapter *Error recovery* discusses error recovery in general and error recovery specifically related to the LCS/CCS network, shared pubset network and XCS network.
- Chapter *Performance and optimization* explains how the performance of the MSCF network is monitored and can be enhanced (via PFA) by the different monitoring programs of openSM2.
- Chapter *Commands* describes the commands for MSCF and NSM in alphabetical order.

- Chapter *Macros*
describes the macros for MSCF in alphabetical order followed.
- *Appendix*
describes the MSCF interface (operand HOST=*ANY) of the ENTER-JOB and ENTER-PROCEDURE commands and important points for the generation and operation of an MSCF network.

At the end of the manual you will find a *Glossary*, a list of *Abbreviations*, a list of *Related publications* and an *Index*.

README file

Any additions to the manuals are described in the Readme files for the various product versions. These Readme files are available at <http://manuals.ts.fujitsu.com> under the various products.

Readme file under BS2000/OSD

On your BS2000 system you will find Readme files for the installed products under the file name:

```
SYSRME.<product>.<version>.E
```

Please refer to the appropriate system administrator for the user ID under which the required Readme file can be found. You can also obtain the path name of the Readme file directly by entering the following IMON command:

```
/SHOW-INSTALLATION-PATH INSTALLATION-UNIT=<product>, LOGICAL-ID=SYSRME.E
```

You can view the Readme file with `/SHOW-FILE` or by opening it in an editor or print it at a standard printer using the following command (e.g. MSCF V17.0):

```
/PRINT-DOCUMENT FROM-FILE=SYSRME.MSCF.170.E,  
LINE-SPACING=*BY-EBCDIC-CONTROL
```

Additional product informations

Current information, version and hardware dependencies and instructions for installing and using a product version are contained in the associated Release Notice. These Release Notices are available at <http://manuals.ts.fujitsu.com>.

1.3 Changes since the last version of the manual

The following major changes have been made since the previous manual:

- The HIPLEX MSCF README file has been integrated.
- The manual has been brought in line with BS2000/OSD V8.0.
- HIPLEX MSCF can also be used on SQ servers.
- Monitoring groups are not longer created manually, but dynamically and automatically from the current shared pubset configuration.
- New section “[MONJV for monitoring the MSCF connection to a partner processor](#)”.
- New [section “Live Monitor of HIPLEX AF”](#).
New value RECOVERY-START=*SECURE for reliable crash detection by the Live Monitor of HIPLEX AF.
New configuration parameter FADING-INTERVAL for determining the wait time following crash detection by the Live Monitor of HIPLEX A.
- New [section “Email notification about critical states”](#).
New configuration parameter NOTIFY-BY-MAIL for providing email notification of critical system states.
- In the CCS network the failure message MCS1100 only needs to be answered on one of the surviving processors. The answer applies for all surviving processors.
- New Cluster Recovery Lock (CRL) for diffusing a “[Disk monitoring blocks](#)”.
- New macro “[DPSHSHP Output information on current shared pubset configuration](#)”.

The contents of the chapter “**HIPLEX MSCF with RFA and SPD**” have been included in the [chapter “Applications and basic functions”](#).

The “**Messages**” chapter is no longer included.

You will find the messages using an HTML application on our manual server (URL: <http://manuals.ts.fujitsu.com>) and on the DVD “BS2000/OSD SoftBooks”.

1.4 Notational conventions

The following notational conventions are used in this manual:



This symbol indicates an important passage

References to related publications are presented in inverted commas with a short title. You will find the full title in the chapter Related Publications under the number indicated. You will also find a brief description of the manual's contents.

Cross references indicate the page, and where appropriate the section or chapter. Cross references concerning subjects discussed in other manuals only give the relevant manual's short title. The subject can be found in that manual with the help of the index.

2 Basic information about the MSCF network

This chapter explains the layers within the architecture of an MSCF network.

- The basic level is the hardware, i.e. BS2000/OSD servers and their connection media.
- In the level above that, the communication system makes sure that the preconditions for communication between the processors are satisfied.
- In the level above that, the function network ensures that functions that were initially only available locally are made available throughout the network by "distributed system functions" (which use the communication system).

2.1 Hardware configuration

The prerequisites for the HIPLEX MSCF function network are firstly the communication of the BS2000/OSD servers participating in the network and secondly, the shared use of shareable resources (e.g. disks, global storage (GS)) by several processors or network participants (depending on the network type). The processors included in the network must therefore be connected with one another in terms of hardware and the shared resources (the network resources) must be accessible from each processor.

HIPLEX MSCF imposes no specifications on the type of hardware connection between the processors (apart from the fact that it must be supported by BS2000/BCAM). Depending on the functions required and the envisaged purpose, the performance and availability requirements will differ in terms of the quality and capacity of the hardware used.

The figure below shows a configuration for a 2-processor XCS network which has a global storage, a shared disk pool and a central control station. This kind of configuration can be operated with high availability when HIPLEX AF applications are used.

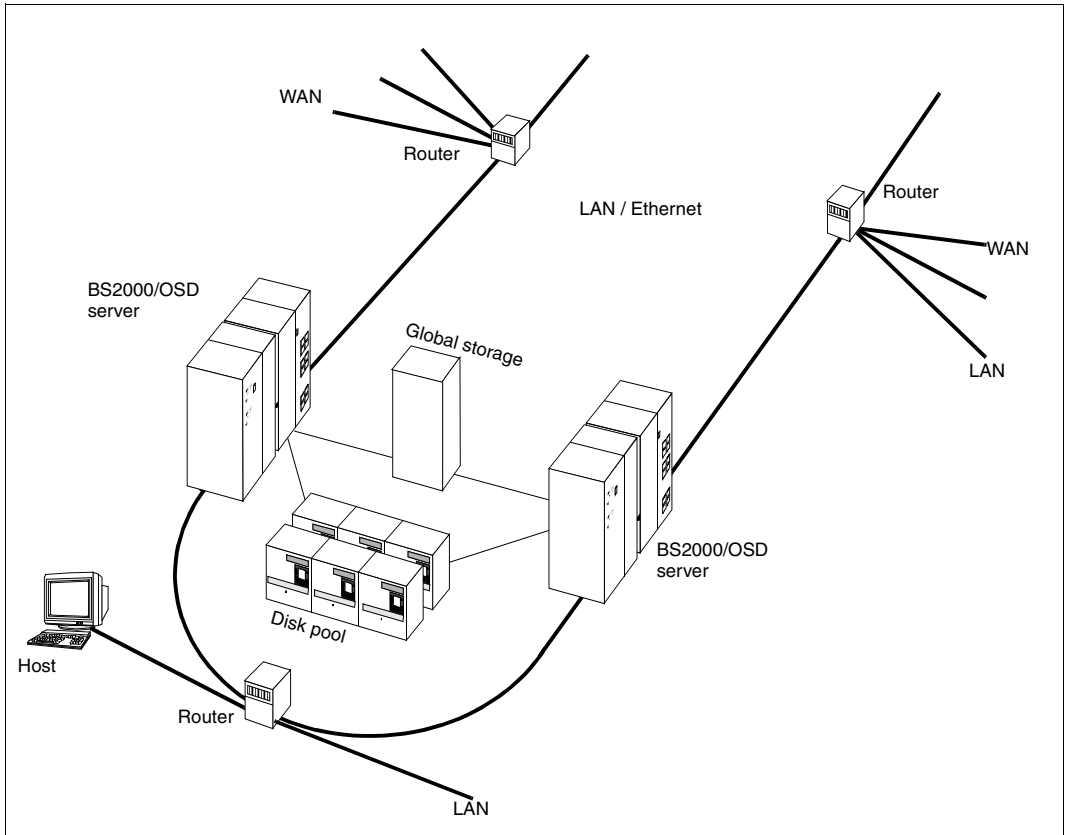


Figure 2: Hardware

To guarantee maximum functionality, the configuration shown above presupposes a powerful connection between the processors. The connection between the processors should be designed as redundant to enable HIPLEX MSCF to switch to another connection when necessary (which forestalls any disruption to operations due to a failed connection). Access-critical volumes can be moved to the global storage or buffered in main memory or global storage to increase performance. The use of disk systems with RAID functionality, together with redundant access paths, guarantees data availability at the same time as performance-efficient access. The entire system can be fully monitored and operated from the central host.

2.2 HIPLEX MSCF communication system

The BCAM-based HIPLEX MSCF communication system provides a logical connection between one processor and its partner processors. The communication system processes requests to establish and to clear down connections, monitors the connections and makes mechanisms for the communication between the processors available to the "registered functions" of the operating system.

In HIPLEX MSCF, a distinction is made between a "short connection" and a "long connection".

The "short connection" is established and cleared down on command and is equally available to all "registered functions". Job processing can, in a way, be compared with a remote procedure call, i.e. a communication interface receives the requests, transmits them to the required partner processor, has them executed there by means of the MSCF server task (one of the tasks made available by MSCF) and returns the results (via the MSCF task to the request-submitting task). A connection between the request-submitting task and the request-receiving task only exists for the ("short") duration of each individual request. The reading of a file catalog entry, for example, is processed via short connections.

Another HIPLEX MSCF communication variant is the "long connection" with which a dedicated server task/server task communication is implemented for all the processors. A long connection is established and cleared down by a "registered function" and is only available to it for communication purposes. The long connection thus links the two server tasks of the "registered function" to the processors over a ("long") time. DLM locks that apply to all processors, for example, are managed by long connections.

The communication itself is handled by BCAM. HIPLEX MSCF sets up four BCAM applications for this (six in the XCS network) on each processor. When an MSCF connection is then set up between two processors, BCAM connections are established between these applications.

The short connection between two processors consists of four or five BCAM connections:

- a connection between the \$MRSAPP applications on both processors
- a connection each between the \$MRSAPP application of one processor and the \$MRSAPPS application of the other processor
- a monitoring connection between the \$MRSAPP4 applications on both processors. Optionally, a second monitoring connection can be generated between the \$MRSAPP5 applications of both processors.

For a long connection, a BCAM connection between applications with the same name is used on both processors (applications: \$MCSNSM, \$MCSNLX). In other words, there are four or five BCAM connections (six or seven in the XCS network) between the applications:

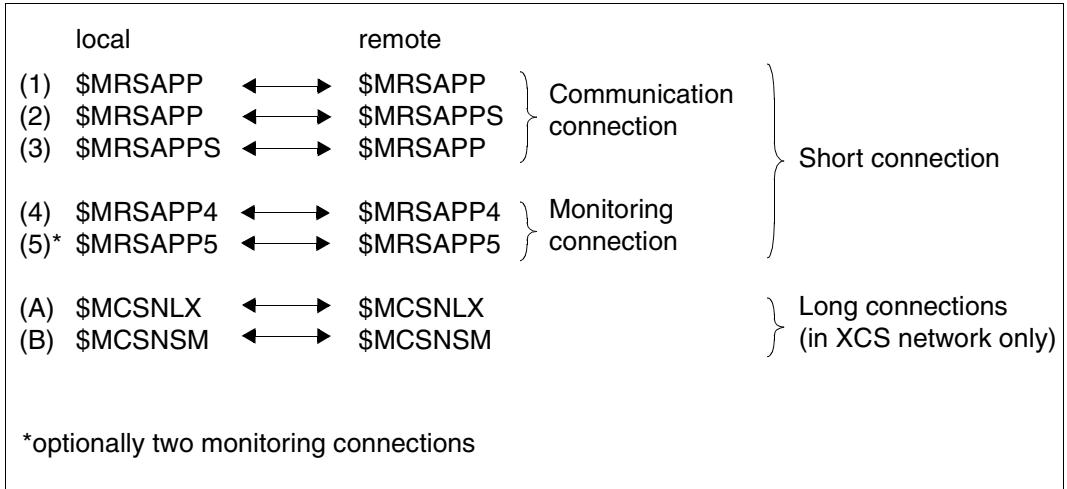


Figure 3: BCAM connection of a HIPLEX MSCF CCS connection

The communication via a short connection between two processors with HIPLEX MSCF works as follows:

- Connection (1) is used for the internal communication between the MSCF components.
- Connection (2) is used to transmit the requests (REQUEST messages) of the local processor to the remote processor and to receive the results (REPLY messages) from the remote processor.
- Connection (3) is used to receive the requests (REQUEST messages) of the remote processor and to send the results (REPLY messages) of these requests.
- Connections (4) and (5) are used solely to monitor the partner processor. Connection (4) is mandatory, connection (5) is optional.

Connections (1) and (2) are additionally used to monitor the MSCF connection and the partner processor's vital-signs.

For the long connections (A) and (B) set up by NSM, both the entire communication and the monitoring function are implemented via the one associated BCAM connection.

The figure below illustrates the interaction between the various tasks on two XCS processors during the communication and connection monitoring. The short connection here shows only the request processing by the local to remote processor.

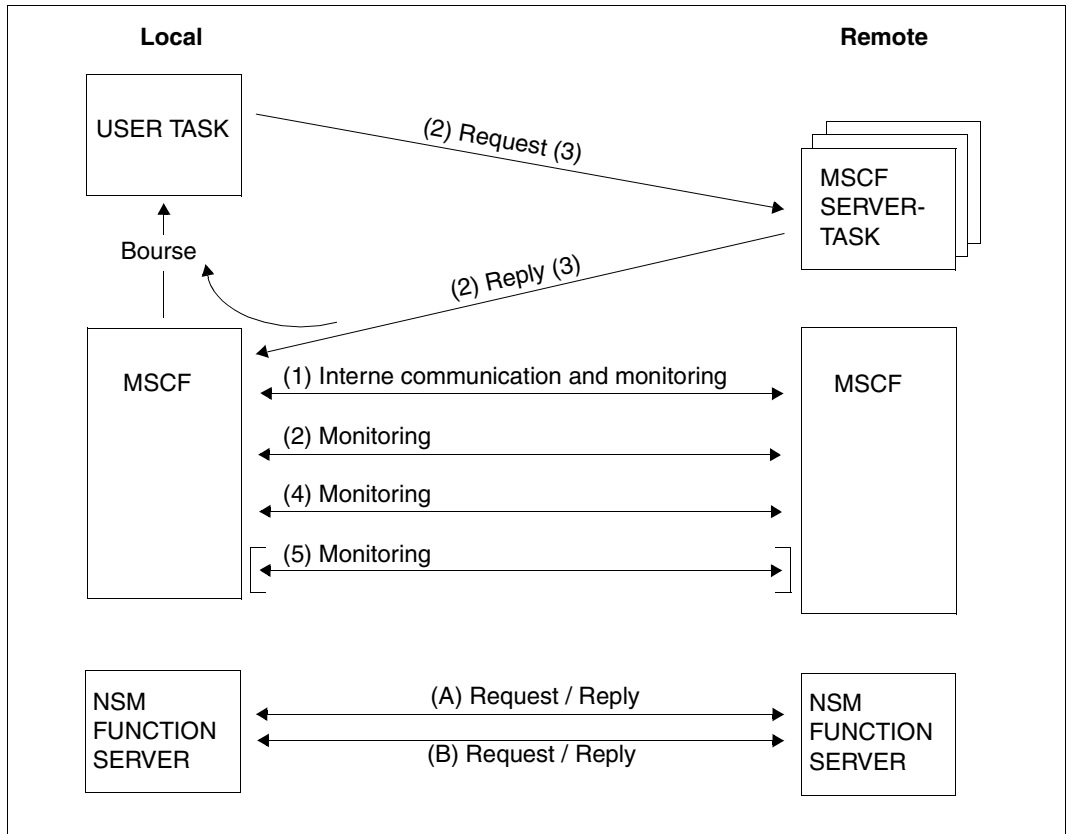


Figure 4: How the HIPLEX MSCF communication works

2.3 HIPLEX MSCF function network

The HIPLEX MSCF function network makes available the operating system functions, that were originally only available on one processor, throughout the network. A pubset can then be accessed not only by the one processor but by all of the processors in the network. The network itself may not necessarily be visible to an application, e.g. files made available on a shared pubset via the registered system function "shared pubset manager" are also available on all processors that access that pubset, as if they were local files. Only the use of processor-independent functions (e.g. shared update from several processors) can require that the network configuration is taken into account in the applications. The registered functions themselves use the services of the HIPLEX MSCF communication system to exchange requests with their partner systems on other processors.

HIPLEX MSCF recognizes different types of function network which differ in the way they cooperate. The starting of a request on a partner processor, for example, is a two-network cooperation between the two processors. Accessing shared pubset is a network cooperation between all processors that have imported the shared pubset. Global locks are a network cooperation between all of the processors that are currently participating in the same XCS network.

The following network types exist with different functionalities:

- LCS network with the "remote catalog access (RCA)", "remote enter" and "remote job control" functions.
- CCS and the shared pubset network with a "closer" coupling between two processors and management operations applicable to all the processors that simultaneously access a shared pubset.
- XCS network with a "closer" coupling between two processors and a management entity for the XCS participants. The "shared file system" and "global locks" functions, among other things, are supported in this kind of network.

Each network type includes the functionality of all the preceding network types.

The figure below illustrates the characteristics of the different network types. It shows a HIPLEX with four processors.

Processors P1 and P2 are linked in an XCS and form an XCS network. They are responsible for production. The production data is stored on shared pubsets which can be accessed by both processors. The processors form a shared pubset network in relation to the shared pubsets which both processor have jointly imported.

Processor T1 is used for testing and development. Shared tools and programs are stored on a shared pubset which processors P1, P2 and T1 can access. Together they form a shared pubset network. The connection to processor F1 is only set up when cross-matching of the data is carried out (e.g. during the night using RFA, controlled via CJC).

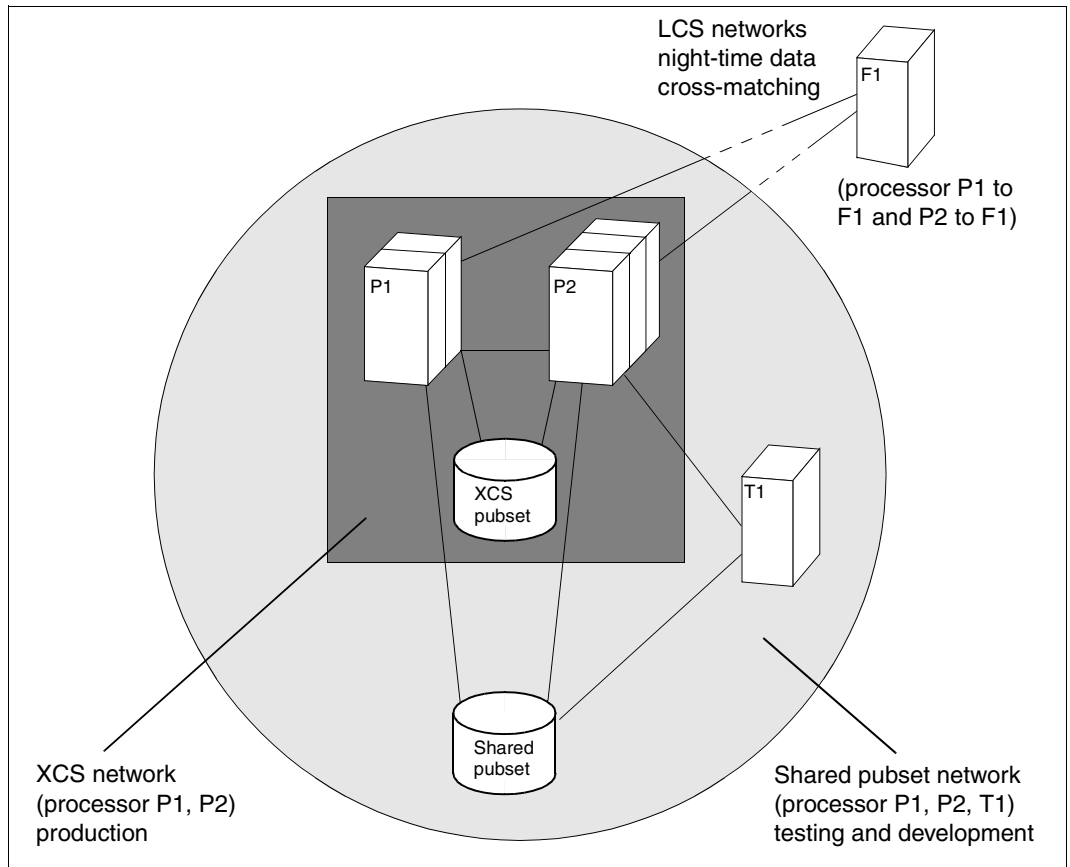


Figure 5: Network types

3 Connection types and network types

This chapter

- explains LCS, CCS, shared pubset and XCS networks
- discusses the connections, partner types, applications, special features and restrictions of the different network types
- describes entry, reentry to and exit from the XCS network
- explains the XCS reconfiguration.

Terms "connection", "network" and "partner"

The terms "connection" and "network" are defined in HIPLEX MSCF as follows:

- *Connection* denotes the communication path implemented by BCAM which links two processors to each other (even different BS2000 versions).
- *Network*, on the other hand, denotes a group of processors which work together in a functional sense. HIPLEX MSCF makes the connections and the coordination mechanisms available with which the distributed system functions form the function network.

HIPLEX MSCF makes a distinction between the following *connection types*:

- LCS connection
- CCS connection.

These connection types form the basis of the *network types* listed below.

- LCS network ("Loosely Coupled System" formed via an LCS connection) as well as via CCS connections
- CCS network ("Closely Coupled System")
- SPVS network ("Shared Pubset Network")
- XCS network ("Cross Coupled System").

All the new features and latest developments are exclusively supported via CCS connections. The functional scope supported via LCS connections will not be extended.

Example

Processor R1 is coupled via an LCS connection with processor R2 and via CCS connections with processors R3, R4 and R5. Pubsets S1 and S2 are used by processors R1 and R4, pubset S3 by processors R1, R4 and R5 as a shared pubset while pubset X1 is used by processors R1 and R5 as an XCS pubset.

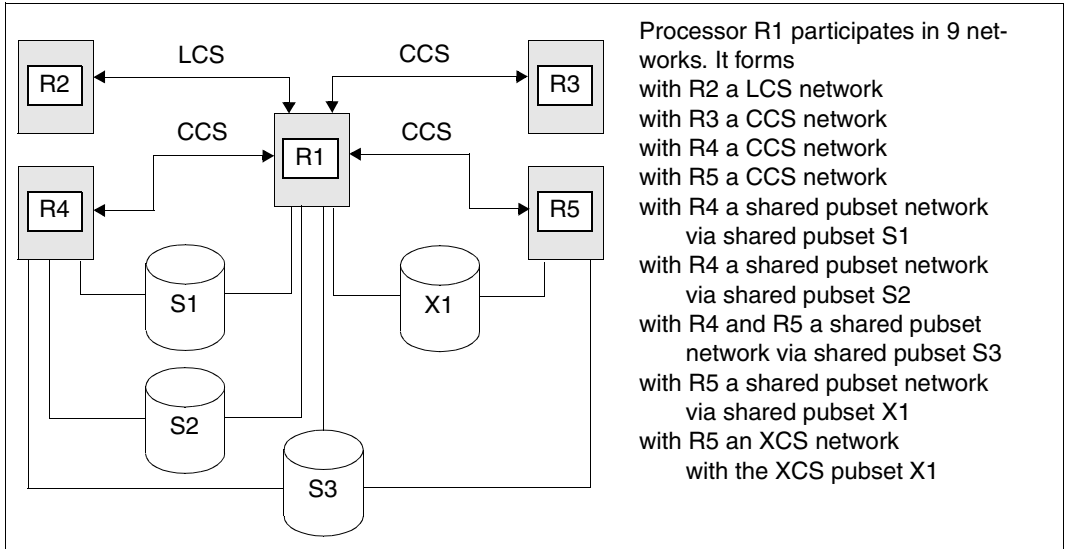


Figure 6: Network types

From the point of view of a processor, the network relationship with a partner processor is defined by the partner type. A distinction is made between the partner types LCS, CCS and XCS.

- A partner processor is called an LCS partner when it has an LCS-type MSCF connection.
- A partner processor is called a CCS partner when it has a CCS-type MSCF connection, but the two processors do not participate in the same XCS network.
- A partner processor is called an XCS partner when both processors participate in the same XCS network.

3.1 LCS network

The term LCS network denotes the function network of two processors between which an LCS connection is set up.

Connection, partner type

The LCS network is only handled via LCS connections and a partner is called an LCS partner.

Applications (functions)

An LCS network offers the following functions (altogether known as the LCS functionality):

- catalog accesses to the partner processor (remote catalog access, RCA) (e.g. with SHOW-FILE-ATTRIBUTES command)
- MRSCAT update (implicit during connection setup or explicit via command)
- the ENTER-JOB and ENTER-PROCEDURE commands for jobs started on the partner processor
- control of remote jobs with the CANCEL-JOB, CHANGE-TASK-PRIORITY and SHOW-JOB-STATUS commands
- conditional job control (CJC) for all processors by means of job variable (prerequisite: JV subsystem).

For each access, the LOGON validation of the local user is repeated on the partner processor to protect the partner processor against illegally infiltrated jobs.

3.2 CCS network

The term CCS network denotes the function network of two processors between which a CCS connection is set up.

Passwords can be allocated for the local and the partner processor which must be specified when a CCS connection is set up. Before a CCS network can be created, the local and the partner processor must know both passwords (see [section “Establishing connections” on page 147](#) and [section “Clearing down connections” on page 148](#)). This minimizes the risk for each processor of linking up with an illegal partner.

The CCS network mainly acquires practical value when it becomes part of the shared pubset or XCS network (a CCS network always consists of two processors; a shared pubset or XCS network, on the other hand, comprises up to 16 processors. The (CCS) connections necessary for this kind of network are set up between two processors at a time, i.e. a shared pubset network consisting of five processors has at least 4 CCS networks).

A CCS network can also be used to monitor partners without the aid of shared pubsets.

Connection, partner type

The CCS network is only handled via CCS connections and a partner can be a CCS or an XCS partner.

Connection monitoring

The two processors monitor the CCS connection (and thus also the relevant partner) independently of each other by periodically sending special messages which the partner must reply to within a certain time. In the event of failure or lost connection, a connection recovery action is implemented to restore the connection (and thus also the network) to normal functioning.

In this context, see also [section “Failure of BCAM connections” on page 168](#).

Applications (functions)

As a rule, the CCS network is part of a shared pubset or XCS network. As an independent network, the CCS network possesses the functionality of the LCS network and can additionally be used for partner monitoring. In addition the software monitor openSM2 is also supported.

3.3 Shared pubset network

The term shared pubset network (SPVS network) denotes a function network of up to 16 processors which use direct hardware paths to the same pubset and have imported that pubset at the same time.

Within a shared pubset network, one processor is nominated the owner of the pubset (master processor) (via the SET-PUBSET-ATTRIBUTES or IMPORT-PUBSET command) which processes the management functions for the files on the pubset, the user IDs, and the accesses for the other processors (slave processors). There must be CCS type connections between the master processor and each slave processor. While all management requests from the slave processors, which require accesses to the DMS metadata (file catalog, file lock table etc.) are directed via HIPLEX MSCF to the master processor, the read and write accesses to the user data are made via direct hardware paths. In other words, input and output operations with the user data are processed by each sharer, irrespective of whether master or slave processor, via its direct hardware path without having to access the master processor.

A shared pubset can be an SM or an SF pubset. In a shared pubset network, processors are grouped, so to speak, around a shared pubset. A processor can participate in several shared pubset networks. It can be the master processor in one shared pubset at the same time as being a slave processor in other shared pubsets. In addition to the shared pubsets, each processor can also import exclusive pubsets.

The participants in the shared pubset network monitor each other to ensure uninterrupted functioning. If a sharer fails, a network recovery is initiated. If the failed processor was the master processor, provided the system is set accordingly, another sharer assumes the master role (see section [“Failure of a processor in a shared pubset network” on page 172](#)).

Disk monitoring

Beside being monitored through the CCS connection, the participants of a shared pubset network are monitored via the pubset itself. When a pubset is imported to the master processor in shared mode, the DMS automatically sets up the watchdog file on the pubres of the SF pubset or the volres of the SM pubsets.

Each sharer (both as master and as slave processor) periodically writes an incremental counter (= vital-sign message) to the watchdog file and reads the vital-sign messages of the other sharers (disk log). A potential failure of the sharer is detected by the fact that it ceases to write vital-sign messages, i.e. its counter stops making the increments.

Control groups

Monitoring groups are formed dynamically and automatically from the current shared pubset configuration.

If several partners fail at the same time, all partners belonging to the same control group are reconfigured together. Fail reconfigurations for partners from different control groups are started independent of one another. This ensures that the master change which is automatically triggered by the fail reconfiguration is not blocked by a failed partner that has not yet been reconfigured.

Partner monitoring

Partner monitoring is partner-related. It is based on the following, mutually independent mechanisms:

- connection monitoring - see [page 30](#) and [section "Failure of BCAM connections" on page 168](#)
- disk monitoring - see [page 31](#).

While the connection monitoring takes place in the CCS network on which the shared pubset network is based (LCS networks are ignored), the disk log runs in each shared pubset network in which the local processor and the partner both participate. The failure of a sharer is only assumed if it writes no more vital-sign messages on *any* common shared pubsets and no longer replies via the MSCF connection. This also means that in the event of a partner failure, the recovery measures are introduced simultaneously for all common shared pubsets.

Connection, partner type

Before a shared pubset network can be set up, CCS connections must exist between the master processor and all slave processors. An CCS connection between two slave processors is not mandatory or the connection may be an LCS type. If the network is then reconfigured, however, the reconfiguration is subject to certain restrictions (in particular, no master change can be carried out if there is no CCS connection between the designated master processor and the slave processor).

No special partner type has been introduced for a shared pubset network. The partner type output in the SHOW-MSCF-CONFIGURATION command is based on the participation of the partner in the other network (types).

Applications (functions)

In addition to the functionality of the CCS network, the following functions are also supported in the shared pubset network:

- File management, including security functions
Virtually all of the DMS interfaces are available in a shared pubset network; the associated security functions (SECOS) are also supported throughout the network.
- Storage management and backup functions
All of the interfaces for managing the background storage facility, in particular migration and backup (also with "Concurrent Copy") are available in the shared pubset network.
- Caching in the shared pubset network
Caching on shared pubsets in the user-PFA concept is supported (see [section "Performant File Access \(PFA\) for shared pubsets" on page 194](#) and the "DAB" manual [2]).

3.4 XCS network

The XCS network is a grouping of up to 16 processors to form a performance (also called a load network) and availability network.

A processor can only participate in one XCS, although it can also participate in different LCS, CCS and shared pubset networks.

The basic feature of the XCS network is the global consistency of the XCS configuration, i.e. all distributed system functions on all processors recognize the same processor as a participant of the same XCS network. *Each* of the processors in the XCS network must therefore be connected *with every other processor of the same XCS network* via a CCS connection so that all XCS participants can communicate with each other as and when required.

In addition, there must be at least one XCS pubset in the XCS network. This is a selected shared pubset which is *automatically* imported by the system to all the participants of the XCS network - and only to them (the pubset must not be imported explicitly). In other words, XCS pubsets are only available to the processors of the same XCS network and can be used as global data repositories for that network. An XCS network is also always a shared pubset network in terms of the XCS pubsets.

An XCS pubset can be an SM or an SF pubset. A shared pubset that contains GS volumes cannot, however, be used as an XCS pubset.

Connection, partner type

A CCS connection must exist between any two processors of the same XCS network. Connections to other processors (that do not belong to the XCS network) can be LCS or CCS connections.

A partner processor within the same XCS network is called an XCS partner. Outside the XCS network (including members of other XCS networks), the partner type is defined by the respective network type.

Applications (functions)

In addition to all of the functions available in the shared pubset network, the following functions are also supported in an XCS network:

- *DLM*
Lock management for all processors for the coordination of accesses by applications to global objects.
- *Shared File System*
File access for all processors also in shared update mode, both for XCS and shared pubsets, assuming UPAM, FASTPAM or DIV access methods.
- *Shared PLAM*
Parallel accesses from different processors to libraries on shared pubsets. Shared PLAM is supported.
- *XTS (XCS Time Synchronisation)*
Synchronizes XCS time and adjusts the local processor clocks to a common time.
- *Parallel HIPLEX*
Use of a global storage (shared GS) used by *all* processors; all the processors thus form a "parallel HIPLEX" (see [page 48](#)).
- *GSVOL*
Emulation of disks with especially performance-efficient input/output processing in the global storage.

The "lock management for all processors" and "shared file system" functions permit the support and implementation of distributed database systems in the XCS network.

Configuration Manager (XCM)

The configuration of an XCS network, i.e. the number of processors belonging to the network, can change dynamically during an XCS session because processors can enter the network in any order, exit the network or cancel their XCS participation as a result of an error. The global consistency must, however, be guaranteed during all changes in the configuration. This is the job of the XCS Configuration Manager (XCM), a component of HIPLEX MSCF. XCM monitors the XCS participants and initiates the reconfiguration of the existing system functions distributed to the participants each time the configuration changes. The failure of a connection or disk log and thus the potential crash of a partner are reported to the XCM by both the connection monitoring and the disk monitoring mechanisms (independently of each other).

When the XCM then definitively determines that the partner has actually crashed, possibly after clarification of the situation by the systems support, it initiates the reconfiguration of the XCS network. Each change in the configuration is made known to all distributed system functions which automatically adapt to the new situation. When they do this, reconfiguration units are defined to the XCM which are called "registered functions".

From the point of view of the XCM, a registered function is a coherent unit which is established on all of the processors participating in the network, and whose cooperation on all processors is supported by XCM. If HIPLEX MSCF is started on a processor, XCM initializes and starts the registered functions; when HIPLEX MSCF is terminated, XCM terminates the registered functions. If a participant in the XCS network crashes, the registered functions are called on the processors that remain in the network in order to restore the network.

The following functions are known to the XCM as reconfiguration units for ensuring global consistency:

- *NSM - Node Synchronization Manager*

The NSM is the global part of the Distributed Lock Management. On entry to the XCS network, the processor is included in the global lock assignment and removed from it on exit. When a processor fails, the locks held by it are made available again to the other processors in the XCS network.

The long connections of the BCAM applications \$MCSNLX and \$MCSNSM "belong" to the NSM.

- *XTS - XCS Time Synchronization*

Creates the synchronized XCS time and ensures that the local clocks all have the same time.

- *XPM - XCS Pubset Manager*

XPM handles the availability of the XCS pubsets. All of the XCS pubsets are imported when the XCS network is started, and exported when it is terminated. In the event of a processor in the XCS network crashing, it is removed from all of the XCS pubset networks.

The number of the XCS pubsets can be changed during the XCS session via XPM (SET-XCS-PUBSET command).

On entry, exit or failure of an XCS participant, XPM guarantees the necessary automatic import, export or recovery processing.
- *GSM - Global Store Manager*

Coordinates the use of a global storage (shared GS) used by all the XCS participants.
- *NSG - NSM Global Store Part*

The NSG is the part of the NSM that uses the global storage. It is only activated in a parallel HIPLEX. If the XCS network is a parallel HIPLEX, the NSMs of the XCS participants communicate only partially via their long connections. For message transfer, they use the more performance-efficient route via the global storage. Recovery measures are processed via the long connections.
- *GSV - Global Store Shared Volume Emulation*

Offers the emulation of shared volumes in a parallel HIPLEX.
- *CPM - Shared Pubset Manager*

Manages all imported shared pubsets that are not defined as XCS pubsets. If a processor crashes, the CPM initiates the recovery of the shared pubset networks concerned.
- *DAB - Disk Access Buffer*

Manages the GS cache area in a parallel HIPLEX in its role as the central cache handler of BS2000/OSD. Network functions are used within the scope a GS reconfiguration (adding GS units). This ensures that the cache data on both GS units is synchronized.
- *SHC - Symmetrix Host Component*

Provides the functionality of external disk storage systems in BS2000/OSD. As a registered function, it is informed of all configuration changes and malfunctions.

XCS configuration

The entire number of processors participating in an XCS network form the XCS configuration which is identified by a configuration number (XCS CONFIGURATION). When it is generated, the XCS network is assigned the configuration number 1. Each time thereafter that the configuration is changed by any processor in the XCS network, i.e. after each entry, exit or failure of an XCS participant, the configuration number is increased by one. It is used as the reconfiguration number also for the unique designation of the configuration change that results in an XCS configuration of the same name.

The reconfiguration number of a processor's join reconfiguration is called the JOINING ORDER and defines the order in which the processors reenter the network.

The XCS configuration is managed consistently by the XCM on each processor, i.e. all registered functions of the processor recognize the same XCS configuration for the same configuration number. The XCS network is consistent if the configuration numbers agree on all of the processors in the XCS network, or in other words, describe the same XCS configuration. The XCS network is globally consistent (i.e. network-wide) when all of the processors have understood the same configuration at the same time, when there is no disruption due to a lost connection or an XCS participant dropping out the network, and no network entry or exit is being processed.

Reconfigurations in the XCS network are based on the principle of:

- detecting a change in the XCS configuration on the individual processors
- coordinating the participating processors
- updating the configuration on all processors
- resuming XCS operation.

Reconfigurations must be carried out together by all registered functions on all the processors participating in the XCS network. This happens when the XCM used on the processor concerned requests the registered functions to replace the configuration valid up until then with the new one. Coordinated with their partner systems, each registered function then carries out the reconfiguration of its XCS functionality and informs the XCM when the reconfiguration is successfully completed. The transition to the new configuration is made virtually simultaneously on the individual processors and in the individual functions. The XCS reconfiguration is finished when *all* XCS functions have executed the reconfiguration on *all* processors participating in the XCS network.

A distinction is made between the following three reconfiguration types:

- *Join reconfiguration*
This reconfiguration is carried out when a processor enters the XCS network
- *Leave reconfiguration*
This reconfiguration is carried out when a processor exits the XCS network
- *Fail reconfiguration*
This reconfiguration is carried out when a processor that has broken off its participation in the XCS network is removed from the XCS network.

The XCM only initiates an XCS reconfiguration if CCS connections are established between all the processors participating in the XCS network. Join and leave reconfigurations are carried out separately one after the other, i.e. the next join or leave reconfiguration is not processed until the previous XCS reconfiguration is completed. A fail reconfiguration interrupts a join or leave reconfiguration. Crashes of several processors are grouped together in one fail reconfiguration block.

If XCS operations are disrupted by a processor (e.g. failure of a registered function), the interruption is remedied by the fact that the participation of the processor in question in the XCS network is broken off and the remaining processors in the XCS network carry out a fail reconfiguration. Incomplete join and leave reconfigurations are aborted and replaced by a fail reconfiguration for the processor concerned.

If the connection between two processors has fails, the processor with the lower priority level (higher HOST-PRIORITY value or, if the same, higher JOINING ORDER value) breaks off its participation in the XCS network in automatic mode and is subsequently configured out of the network. A fail reconfiguration is initiated on the processors that remain in the network.

The fail reconfiguration of each individual processor is not complete until the target configuration (i.e. the processors remaining in the XCS network) has been fully generated.

4 Applications and basic functions

This chapter provides an overview of the global resources and distributed system functions supported by HIPLEX MSCF as well as information about the application systems which support the operation, administration and use of an MSCF network.

4.1 General overview

An MSCF network consists of two or more processors that cooperate in solving tasks. They may have a client/server relationship or they may work together as equal partners. Distributed system functions, in the sense of an "equal partners" cooperation model, are reserved for the XCS network since it has closer coordination. The functions offered by other types of networks are provided for a job submitter by a server in accordance with the first model.

An elementary requirement for several partners to be able to cooperate is an interface via which they can coordinate themselves. As in a one-processor situation, interfaces are needed to serialize accesses to jointly processed objects and for the synchronization of events that have network-wide relevance. This functionality is made available in the XCS network by the distributed lock manager (DLM).

A network-wide uniform time gives the processors in the network a uniform view when time-dependent events take place and the sequence of such events when several occur. The "GTIME" interface was extended for this purpose for the XCS network and is implemented by the subsystem XCS-TIME.

In addition to the shared pubsets, the global storage (GS) is also available in the XCS network as a fast global storage medium. On the software side, it can be divided into different partitions for different types of use either exclusively by one processor or in shared mode by all of the processors in the XCS network. Especially worth pointing out is the use of the global storage as an emulated volume (GSVOL), used as the basis for a pubset, and in particular for a shared pubset.

Data that is to be processed network-wide is stored in files on shared pubsets and access to this data is coordinated network-wide. This also applies at file level to a shared pubset network (changes to a file can only be made by one processor in the network at any one time). In addition in the XCS network, the shared file system also offers the option of a network-wide simultaneous update of files with block structure at block level and of PLAM libraries at member level. The synchronization at this level is handled by the distributed lock manager.

The network-wide file processing is supplemented by the network-wide file backup and archiving function with HSMS/ARCHIVE. With the network functionality of Concurrent Copy (CCOPY), the same high data availability can be achieved as in a one-processor situation (with parallel backup and processing across a number of processors).

The aim of the BS2000/OSD network functions is to improve, extend and optimize the system's "performance" and "availability" characteristics. The basic functions are used via various application-oriented products to attain this goal.

Distributed job networks can be defined with AVAS and the processing of these job networks monitored based on the operating system functions for the management and monitoring of jobs in the network. This allows the load to be optimally distributed between the processors of a shared pubset network.

HIPLEX AF permits an application to be *automatically* relocated both as a planned action and in the event of error. After a processor crash, for example, this improves the availability of user applications thanks to the swift restart on another processor.

The products HIPLEX OP and HIPLEX ADM (with graphical user interface) enable the MSCF network to be operated and its objects administered from a central location (or single point of operation/administration). HIPLEX AF also offers a mechanism for quick and reliable crash detection, the Live Monitor, see [page 71](#).

4.2 Synchronization and serialization

The basic mechanisms for synchronization and serialization in an XCS network are DLM locks. From the point of view of the DLM, a lock is identified by a name and an area of validity (that applies to processors or the entire network). It is up to the user to assign a name to the resources to be locked (e.g. device, block of a file) and to ensure that the name is unique within the XCS network.

A lock request can be made synchronously or asynchronously. Deadlocks are detected by means of timers.

The lock mode specifies the access permissions of the lock holder. A total of six lock modes are supported:

- *Null mode*
The holder of this lock is not permitted to access the assigned resource; all other lock requests are compatible with this mode.
- *Concurrent read mode*
The holder of this lock is entitled to read the assigned resource but must take into account the fact that other processors may be reading or writing to it at the same time.
- *Concurrent write mode*
The holder of this lock may modify the assigned resource but must take into account the fact that other processors may be reading or writing to it at the same time.
- *Protected read mode*
The holder of this lock is entitled to read the assigned resource. No other task can modify the resource at the same time.
- *Protected write mode*
The holder of this lock is entitled to modify the assigned resource. No other task may modify the resource at the same time, but can, however, read it in concurrent read mode.
- *Exclusive mode*
Only the holder of this lock can access the resource. Other tasks may neither modify nor read the resource at the same time.

In the final analysis, the lock mode selected depends on the extent to which a resource has to be protected in order to be able to carry out a processing stage.

A storage area (called the "lock value block") is optionally available for each lock with the aid of which the lock holders can exchange data. The contents of the lock value block can be modified by owners of a write lock, and read by owners of a read lock. In this sense, the lock value block can be regarded as a distributed storage facility across all the processors.

If a processor belonging to the XCS network crashes or if MSCF is terminated on the processor, the recovery initiated takes into account the necessary reconfiguration measures for the locks held on that processor. When a task is terminated, a lock recovery is implemented as part of the processing steps necessary for the task termination.

The global DLM locks are managed by a HIPLEX MSCF application. The availability of the global DLM locks on a processor with XCS capacity thus presupposes the existence of an MSCF session. It makes sense therefore for DLM locks to be used only to protect resources whose availability is also linked to an MSCF session.

The interfaces of the DLM are implemented by two DSSM subsystems. The DLMUSER subsystem is responsible for the lock handling on any single processor and receives jobs via the user interface. For lock handling that is applicable to all the processors, DLMUSER passes the jobs, if necessary, to a second DSSM subsystem, the node synchronization manager (NSM). The NSM is controlled by the XCM and in addition to the lock handling applicable to all the processors, it also carries out lock-related reconfigurations. The NSM is therefore a reconfiguration unit registered with the XCM, i.e. a "registered function".

A detailed description of the DLM is given in the "Executive Macros" manual [12].

Examples

- *Partner monitoring*

Each processor in an XCS network can be assigned an exclusive lock. The "locked" resource in that case is the processor itself and the lock name, e.g. the unique processor name. If the other processors in the XCS network then attempt to obtain this lock exclusively, they can assume that, as long as they are unable to obtain the lock exclusively or are informed that a change has occurred, the lock-holding (i.e. monitored) processor is active. The reason for this is that in the event of the crash of a monitored processor, the DLM releases the lock and assigns it exclusively to another processor. The processor that receives it can then take over the tasks of the crashed processor.

The processor that receives the exclusive lock of the partner can take over the tasks which are to be performed jointly between the two processors. However, the assignment of the partner lock does not necessarily mean that the partner has failed, but only that it has lost its multiprocessor capability. In this case the other processors can use the shared resources of the partner.

This mechanism must not be used for switching and in conjunction with exclusively assigned resources (e.g. exclusively imported pubsets), as these resources could be destroyed in the event of a communication error with subsequent unloading of MSCF.

- *Synchronization of file accesses*

PAM files can be written to and read at the same time from several processors. Generally speaking, the write accesses must be synchronized. Appropriate locks can be set on all the processors with the help of the DLM so that, for example, only one processor has write access to the object.

- *Parallel batch runs*

In an XCS network with two processors, let us assume that a file is to be created with UPAM on processor 1 in a batch run. A sequential evaluation is to be executed on processor 2 in a second batch run, also with UPAM (i.e. processor 2 purely has to handle a read process). The file creation and evaluation are to be overlapping which is why the evaluating batch run on processor 2 must be informed about the progress of the file creation on processor 1.

As soon as an appropriate number of blocks is written on processor 1, the block counter can be exchanged between processor 1 and processor 2 with the aid of a global lock using the lock value block. The counter is modified on processor 1 in lock mode "Exclusive" or "Protected Write" and read on processor 2 in lock mode "Protected Read" or "Concurrent Read". The end of this process can be indicated by depositing a special marker in the lock value block. If processor 2 reads a lock value block marked as invalid, it interprets this as an abnormal termination of processing on processor 1.

4.3 Time in the network

In an XCS network, the time management of BS2000/OSD includes two additional functions for time synchronization:

- synchronization of the clocks of the processors in the XCS network
- an XCS-wide, strictly uniformly advancing time, called the XCS time.

While the first function largely guarantees identical times on all of the processors in the XCS network and prevents discrepancies developing in the system times caused by the system clocks running at different speeds, the second function offers the option of relating globally to a shared time.

The time synchronization functions are a standard component of every XCS network. They are implemented by the DSSM subsystem XCS-TIME which is started and terminated under the aegis of the XCM (registered function XTS).

The time synchronization is initialized when an XCS network is started or when a processor enters an XCS network.

To achieve the time synchronization of the processors in the XCS network, the computer-local times are exchanged via the DLM. The XTS function initiates the exchange of time information at periodic intervals. The exchange of time information is therefore independent of any lock exchange. This means that good synchronization of the system times can be achieved and the differences are usually a matter of less than a second.

If no processor has a radio clock connection, the network time in the XCS network is defined by the processor that has the fastest running clock. The clock time of the processors with the slowest running clocks is adjusted accordingly.

If one of the processors has a radio clock connection, the entire network is synchronized according to the radio clock time.

If all participants have a radio clock connection, time synchronization no longer takes place. XCS-TIME checks only the time differences and, if necessary, initiates a network exit if the time differences become too great, e.g. because of a defective radio clock.



Avoid having a network in which multiple processors, but not all, have a radio clock connection.

Processors without a radio clock attempt to keep in line with processors with a radio clock. However, if the times of the various radio clocks arrive in the wrong order, e.g. owing to problems in the BCAM network, the synchronization requests can be contradictory and lead to errors.

The information as to whether processors have a radio clock connection is provided as a status display in the GTIMR interface and in the output field SYNCHRONIZATION of the SHOW-SYSTEM-INFORMATION command.

In addition to the synchronized system time, the XCS time is also executed on every processor in the XCS network. This XCS time applies uniformly to all the processors when they are queried under the same (distributed) lock. The XCS time can therefore be used for logging entries with a counter with UTC time semantics.

For the user, the XCS time is accessible via the GTIME interface (XCS_MODE=*YES) (see the manual "Executive Macros" [12]).



It is not the local times of the processors in the network that are synchronized, but the underlying internal time coordinate UTC, which corresponds to Greenwich Mean Time. Identical GTIME parameters result in identical local times. However, it is possible to set different local times on the XCS network participants by specifying different ZONE, DIFF or CHDATE values in the GTIME parameter set. Conversely, the network entry of a processor will be rejected despite the identical local time if the internal time coordinate UTC on it is different and the difference has merely been "corrected" by means of correspondingly adjusted GTIME parameters.

4.4 Parallel HIPLEX

The global storage (GS) can be used by all the processors in the XCS network. This global GS mode is called parallel HIPLEX. All processors that use a shared GS must belong to the same XCS network. If a shared GS exists in an XCS network, it must be connected to *all* the processors in that network. If a processor exits the XCS network, its connection to the shared GS is also terminated.

As in local mode, the GS can also be used in parallel HIPLEX by DAB (disk access buffer) as an especially performance-efficient cache memory. Furthermore, the XCS network offers the option of using the GS globally as a fast volume storage facility via the volume emulation function (GSVOL).

The GS is divided in software terms into "partitions" for the different types of use. When the system starts to use any of these partitions, it individually determines whether the partition is to be used exclusively by one processor for one type of use or whether it is to be available to all of the processors in the network in shared mode (shared partition). DAB utilizes partitions according to their use (exclusive request for exclusively imported pubsets, shared request for shared-imported pubsets). GSVOL also supports shared partitions and thus shared GS volumes.

The use of a global storage shared by all the processors is coordinated by the GSMAN subsystem (Global Storage Manager). GS mode is controlled via the parameter GSUSAGE which can accept the values GSUSAGE=LOCAL or GSUSAGE=GLOBAL. The value GLOBAL, however, presupposes a processor generated for participation in an XCS network (system parameter MCXSPXCS=Y or V). The value for GSUSAGE is determined through the GSMAN parameter file SYSSSI.GSMAN.<version>. That value can be modified at a later point in time using the command MODIFY-GSMAN-PARAMETERS.

Mixed forms involving both local and global operation are not possible (e.g. two processors in an XCS network share a global storage, while a third processor in the same XCS network has its own or has no global storage) (see [figure 7](#)).

The management and reconfiguration of a locally used global storage correspond to the functions on a processor without XCS. The same applies to the users of the global storage.

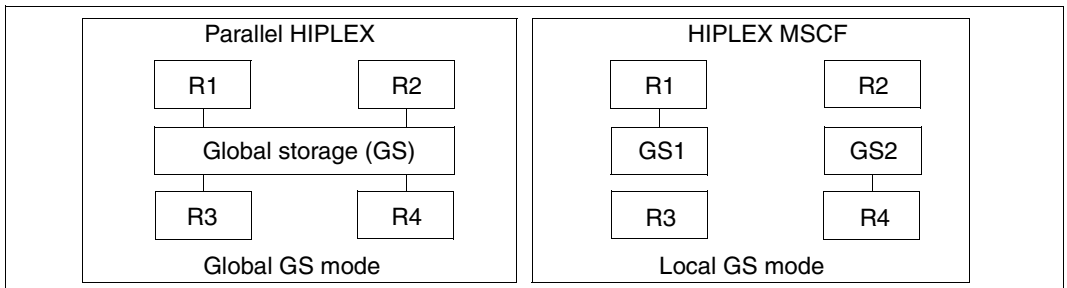


Figure 7: Permitted GS configurations

The global storage is described in detail in the manual "Introductory Guide to Systems Support" [5].

4.4.1 Network entry and exit by a processor

When a processor enters an XCS network (initiated by the startup of HIPLEX MSCF), GSMAN makes sure that the same value for GSUSAGE is set on all the processors in the network. In effect, the first processor's GSUSAGE setting defines the setpoint value for the XCS network. The entry of subsequent processors to the XCS network is aborted if their GSUSAGE setting does not agree with that of the first processor (message EGC1007 is output). If the setting is GSUSAGE=GLOBAL, GSMAN also checks whether all the processors that (want to) use the global storage belong to the same XCS network or the same parallel HIPLEX and are physically connected to the same global storage. If not, an appropriate message is output and GSMAN cancels the entry to the XCS network.

In parallel HIPLEX (GSUSAGE=GLOBAL), the GSMAN functions, i.e. all the GS commands and GS accesses, are only available after successful entry to the XCS network (XCS READY). Prior to this, all requests to GSMAN are rejected with the message EGC0010. For this reason the users of the global storage (DAB, GSVOL) may not be started until HIPLEX MSCF is ready to run. For information on how the XCS network is set up, see [section "Starting HIPLEX MSCF" on page 144](#).

When a processor leaves the XCS network (STOP-SUBSYSTEM MSCF command), GSMAN locks all GSMAN functions again. The only exceptions are GS accesses to partitions which are still in use at the time of exit from the XCS network. Prior to the exit from the XCS network, system support should ensure that all GS usages have been terminated normally on the processor concerned. The command `SHOW-GS-STATUS SELECT=PARTITIONS(SELECT=BY-ATTRIBUTES(ASSIGNED-HOST=<host-name>))` provides information about the partitions still being used by the exiting processor.

4.4.2 Configuration of the global storage

The global storage is managed and made available to the users by the GSMAN subsystem.

Configuration of the GSMAN subsystem

When the GSMAN subsystem is started the GSUSAGE subsystem parameter is read from the file SYSSSI.GSMAN.<version>. *A record containing either "GSUSAGE=LOCAL" or "GSUSAGE=GLOBAL" is expected.* In the event of any error (DVS error or wrong record contents) GSUSAGE=LOCAL is accepted and acknowledged with the message EGC1010 or EGC1011. If the local processor cannot participate in an XCS network, GSUSAGE=GLOBAL is replaced with GSUSAGE=LOCAL and the message EGC1012 is issued. The value contained in the file can be changed at a later point in time using /MODIFY-GSMAN-PARAMETERS

On delivery the file SYSSSI.GSMAN.<ver> contains the record "GSUSAGE=LOCAL". *In global GS mode "LOCAL" has to be changed to "GLOBAL" before the system is used for the first time.*

When the MSCF subsystem is started in XCS mode, GSMAN checks the hardware and software configuration and carries out the following tests:

1. All of the processors belonging to an XCS network must have the same value for the GSUSAGE subsystem parameter. The first processor that enters the XCS network determines this value for the entire network, i.e. GSMAN prevents the HIPLEX MSCF from being started on another processor if its GSUSAGE value differs from that of the first processor (message EGC1007 is output).
2. The second thing that is checked is whether another XCS network is already using the global storage. If it is, message EGC1002 queries whether the global storage is to be released or not and thus withdrawn from the other XCS network. Before responding to this message, the user must check whether the XCS network indicated in the message is using the global storage or not. Only if the global storage is not being used may the message be answered with "Y". Otherwise the message must be answered with "N" which cancels the HIPLEX MSCF startup.
3. The third thing that is checked is whether a processor that does not belong to any XCS network is using the global storage. If this is the case, message EGC1004 queries whether the global storage is to be released or not and therefore withdrawn from the other processor. Before responding to this message, the user must check whether the processor indicated in the message is using the global storage or not. Only if the global storage is not being used may the message be answered with "Y". Otherwise the message must be answered with "N" which cancels the HIPLEX MSCF startup.
4. Finally the system checks whether the processor entering the network is physically connected to the same global storage as the processors already in the XCS network. If not, the HIPLEX MSCF startup is cancelled and message EGC1005 output.

Configuration of the global storage

Configuration of the GS consists of setting up partitions for the GS users. All actions relating to the configuration of a global storage are executable on *one* processor (single point of administration) and are valid throughout the XCS because at any given time the valid configuration data is located on the global storage. All GS configuration commands behave in the XCS network exactly as in local GS mode.

When HIPLEX MSCF is terminated (normally or abnormally) on one of the processors in the XCS network, however, the GS configuration commands are rejected on that processor.

Reconfiguration of the global storage

The reconfiguration of the global storage consists of connecting or disconnecting GS units. These actions are also carried out throughout the XCS, both in the event of error (implicit DETACH) and for explicit actions initiated via the DETACH-GS-UNIT and ATTACH-GS-UNIT commands. Checks that are run before the commands are executed are made accordingly throughout the XCS.

4.4.3 Use of global storage by DAB

DAB (disk access buffer) is the central (software) cache handler for all CPU-oriented storage media (including the global storage) for accelerating inputs and outputs. To better understand the explanation below and for terminology definitions, see the "DAB" manual [2].

With respect to the use of GS partitions by DAB in the parallel HIPLEX, the following two basic scenarios can be distinguished:

- The buffered files reside on exclusively imported pubsets and are supported with user-PFA cache areas, or they are objects (pubsets/private disks, files, volumes) which are supported with ADM-PFA cache areas (on exclusive or shared pubsets, private disks or SPD). The use of the GS partitions here largely corresponds to their use in local GS mode. A GS partition is used exclusively by a processor for cache areas where files used by this processor are buffered on (usually exclusively operated) volumes.
- The buffered files reside on shared pubsets which are supported by user-PFA cache areas. In this case the GS partitions are occupied by DAB as "shared" so that they can be accessed by all pubset sharers. The GS partition contains exactly one cache area (or a maximum of one per volume set with SM pubsets), which in turn is divided into segments for each pubset sharer. These sharer-specific segments are used to buffer the files which are referenced locally by the relevant processor at any one time. For details of the concept of DAB support for shared pubsets, see [section "Performant File Access \(PFA\) for shared pubsets" on page 194](#) and the "DAB" manual [2].

DAB partitions are created using the GSMAN command CREATE-GS-PARTITION. When the first cache area is created in a partition, the partition is connected to the processor. When the last cache area of this partition is cancelled, the partition is disconnected from the processor. Before a DAB partition is cancelled, it must be disconnected, i.e. all cache areas that use that partition, must be cleared via DAB or PFA commands.

Certain conventions apply for the naming of DAB partitions. Basically, in addition to the prefix DAB, the GS unit (1,2,D for dual) and a reference to the pubset have to be specified. The catalog ID of the pubset references usually the home pubset. When caching data on the SM pubset or the PFA cache areas for shared pubsets, the relevant catalog ID must be specified.

If, after a (hardware or software) failure of a processor, another processor is to take over its applications, the latter processor needs to access the data of the applications and thus to physically access the relevant volumes. If an exclusively occupied volume is buffered in a DAB-GS partition, this partition will also be switched to the new processor (in addition to the volume). All volumes buffered in this partition have to migrate to the new processor. The GS partition is switched over automatically by DAB when the volume is started up on the new processor. If necessary, a query is output to the console as to whether the assignment of the partition is allowed to be released by the old processor (messages EGC0502 and EGC0503).

If a processor crashes in an XCS network in which shared pubsets are buffered with DAB in the global storage, the segments of cache areas assigned to the crashed processor are automatically switched to the relevant pubset master. There the data is written back to the volume and is then available to all sharers again (see also [section “Reconfiguration of DAB cache areas in parallel HIPLEX” on page 57](#)).



Use of an active cache area in the global storage is not affected by the termination of HIPLEX MSCF, be it normal or abnormal. In a parallel HIPLEX, however, the cache area in the global storage should be terminated with `/STOP-DAB-CACHING` or `/STOP-PUBSET-CACHING` before terminating HIPLEX MSCF normally.

4.4.4 Use of the global storage by GSVOL

GS volumes with particularly performance-efficient input/output processing can be emulated in the global storage by means of GSVOL (GS volume emulation). The data of a GS volume is located exclusively in the global storage. A GS volume is created via the commands CREATE-GS-PARTITION (GSMAN) and CREATE-GS-VOLUME (GSVOL) and as a result can be used in BS2000/OSD like a normal volume (device type D3409-GS) (but not in a home or paging pubset).

In the parallel HIPLEX, a GS volume can always be addressed by all of the processors (it can always be imported). Systems support must define during import whether a GS volume is used in exclusive or shared mode (as a shared pubset) via the relevant interface of the pubset management.

GSVOL is described in detail in the manual "Introductory Guide to Systems Support" [5].

Notes

- With global GS operation in the parallel HIPLEX (shared GS), *all* pubsets which contain at least one GS volume are exported when HIPLEX MSCF is terminated. By analogy therefore, the restrictions and notes in the sections "[Clearing down a shared pubset network](#)" on page 155 and "[Hanging shared pubset occupation](#)" on page 187 also apply to exclusively imported GSVOL pubsets.
- If a shared pubset contains at least one GS volume, it cannot be used as an XCS pubset.
- During the cross-matching of a dual global storage, write operations on a global storage are delayed or rejected. This global storage operation on shared SF pubsets that consist of GS volumes can result in an output error in the watchdog file (messages DMS03B9 and MCA0110 are output). When cross-matching is complete, the watchdog vital-sign messages can continue to be written without interruption (messages DMS03BB and MCA0110 are output).

4.4.5 Example of startup of parallel HIPLEX (two processors)

It is assumed that the subsystems MSCF, GSVOL and DAB have been started before the BS2000 system is ready to go (recommended in parallel HIPLEX mode). To do this, you must set the start time for the MSCF subsystem to “BEFORE-SYSTEM-READY” in the subsystem catalog:

1. System initiation on processor 1
 - Start the DLMUSER subsystem
 - ... DSSM READY
 - Start the GSMAN / GSVOL / DAB subsystems
 - DCSTART (see [section “BCAM dependencies” on page 113](#))
 - Start the MSCF subsystem (implicitly: XCM, subsystems NSM and XCS-TIME)
 - Start the DAB subsystem
 - ... SYSTEM READY
2. Start the GS usage on processor 1 (when XCS READY)
 - if necessary, set up GS partitions
 - if necessary, set up GS volumes, make them addressable (ATTACH-DEVICE command) and import the pubsets consisting of GS volumes
 - if necessary, set up DAB cache areas in the global storage
 - start applications which work with GS volumes
 - start applications which work with the pubsets concerned
3. System initiation on processor 2 - see action 1)
4. Start the GS usage on processor 2 (when XCS READY) - action 2)

Actions 1 and 3 are part of the system initiation if generated accordingly while actions 2 and 4 must be carried out by systems support.

4.4.6 Error handling in the GSMAN subsystem

The GSMAN subsystem includes functions for handling hardware, software and configuration errors. If an error occurs, the hardware can issue hardware error messages whenever the global storage is accessed. In error situations, the software can terminate partial functions or initiate an abnormal system termination (SETS). The XCS configuration may differ from the GSMAN hardware and software configuration. Software and configuration errors may require measures by systems support and hardware errors usually require service measures.

Hardware errors

In the event of hardware errors on a GS unit, accesses to mono-partitions located on the GS unit concerned are terminated with an error message. Accesses to dual partitions continue to run on the intact GS unit.

When serious hardware errors occur, the defective GS unit is implicitly deactivated, message EGC2100 is output and the write access terminated with a warning. The action is carried out XCS-wide.

XCS reconfiguration

Software errors may cause the failure of partial functions or of the entire system. The GSMAN subsystem recovers the following errors:

1. Failure of an XCS partner
If a processor within the XCS network fails, the entire GS data on the processors remaining in the network can still be used. Partitions used exclusively on the failed processor can be used by other processors if the message EGC0503 which is issued after the message EGC0502, is answered with "Y".
2. Abort of own XCS participation
GSMAN treats an aborted XCS participation in exactly the same way as a normal XCS termination, i.e. when XCS participation is aborted no further configuration or reconfiguration commands are accepted on the relevant processor. Requests for utilizing partitions are also rejected. Existing users can continue. If the user releases the partition, it can be used on other processors if the message EGC0503 which is issued after the message EGC0502 is answered with "Y"
When the processor has reentered the XCS network following the restart of MSCF, the global storage can be used again on the relevant processor without any restrictions.

4.4.7 Reconfiguration of DAB cache areas in parallel HIPLEX

Error recovery differs on the one hand between cache areas for exclusively occupied volumes (or ADM-PFA cache areas) and PFA cache areas which serve shared-imported pubsets, and on the other hand between the error situations “Failure of a processor” and “Failure of the connection between processor and buffered volumes”. These situations are described in detail below.

- *Cache area for exclusively occupied volumes*

These can be PFA cache areas for exclusively imported pubsets or ADM-PFA cache areas for exclusively occupied volumes (pubset or private disk). The possibility of ADM-PFA cache areas for volumes occupied as shared can be disregarded here, as this involves pure read caching, in which all data is located consistently on the volumes.

- Failure of a processor

If a processor fails in the case of cache areas with write caching for exclusively occupied volumes, the volumes can be switched to another processor in the parallel HIPLEX. The first time a volume is occupied (e.g. with the IMPORT-PUBSET command), the associated GS partition is automatically relocated to this processor. During this relocation, DAB reconstructs all cache areas contained in the GS partition, so that when these volumes are accessed the data is again available consistently from the cache. The switchover unit in this action is the GS partition, i.e. all volumes which were buffered with cache areas in the same GS partition must be switched to the same processor in the parallel HIPLEX.

Read cache areas can optionally be set up for exclusively occupied disks/subsets so that they can also be reconstructed from the GS partition after a processor failure. When the disks are switched to another processor in the parallel HIPLEX the relevant GS partition is relocated to that processor and all of the cache areas in the GS partition are reconstructed.

- Failure of the connection between processor and buffered volumes

If the data routes to the buffered volumes fail, it is possible to switch the volumes and the associated GS partition of the cache areas to another processor in the parallel HIPLEX during operation. The switchover can occur once it is ensured that the GS partition to be switched can no longer be accessed from the first processor. In principle there is always the option of terminating the use of a GS partition by removing the cache areas. However, if the data routes to the buffered volumes fail, normal removal is not usually possible because it does not allow the cache data to be written back. It is therefore possible to disconnect the cache areas on the first processor from the GS partition (implicit action or following the appropriate response to a message).

When a cache area is disconnected, the GS data occupied by it is retained, but the cache area is removed from the cache configuration of the current processor. When the last cache area of a GS partition is disconnected, the GS partition itself is also disconnected, i.e. the data it contains remains valid and the GS partition can be occupied again from another processor.

If the cache data is not written back successfully with the STOP-DAB-CACHING command, the message NDB0029 is issued. After the message NDB0031 that follows has been answered appropriately, DAB is disconnected for ADM-PFA areas. At the time of the STOP-DAB-CACHING command, the buffered volumes must no longer be occupied, i.e. pubsets must have been exported and the occupation of private disks must have been terminated with the command CHANGE-DISK-MOUNT UNIT=<mn>,ACTION=*CANCEL. In the case of PFA cache areas, the disconnect occurs automatically when the associated pubset is exported.

The relocation of the GS partition then occurs in the same way as when a processor fails, i.e. the first time a buffered volume on another processor in the parallel HIPLEX is occupied, the GS partition is relocated to this processor and all cache areas in the GS partition are reconstructed.

- *PFA cache areas for shared-imported pubsets*

For shared-imported pubsets, PFA cache areas in the global storage are divided into segments for the respective processors using them. If a processor fails (or if a switchover occurs because of a loss of connection to the volumes), only the segment of the cache area that is assigned to the processor is relocated. Relocation here is always to the pubset master; if the master fails, the segment is relocated to the new master that took its place (master change).

- Failure of a processor

If a pubset slave fails, the cache area it uses is relocated to the pubset master, with the relocation occurring automatically as part of the failure detection of the slave by MSCF. If the pubset master fails, its cache area segment is relocated automatically as part of the master change performed by MSCF. In both cases the cache area segment is reconstructed on the pubset master and the cache data is written back to the volumes. Once the cache data has been written back, the system resets the previously existing file locks which prevent the files concerned from being opened from processors other than the master. The files can then be accessed again from all processors of the shared pubset network.

- Failure of the connection between processor and buffered volumes

If in the case of a shared pubset the physical connection to the volumes fails while there is still an active BCAM connection, this does not lead to a failure detection by MSCF. The state is recognized by the monitoring system but the shared pubset is not exported. If this error occurs on a pubset slave, the files can still be opened (the metadata accesses, e.g. to TSOSCAT, occur on the master), but it is no longer possible for data to be read or cache data to be written back to the volumes.

Depending on the circumstances, when it comes to closing the files the DMS return code 0E27 might inform you that an I/O operation could not be performed during closing.

If the pubset on the slave is exported by systems support, the cache area segment assigned to the processor is disconnected during the export. The segment is relocated to the pubset master and the cache area is reconstructed there. The cache data is then written back and any file locks still existing are reset (as in relocation following a processor failure).

If the connection to the volumes on the pubset master fails, the metadata can no longer be accessed, i.e no file on the shared pubset can be opened or closed, even on the pubset slaves. The cache data of the files open on the master can also not be written back to the volumes. To resolve this situation, you must issue the EXPORT-PUBSET command on the pubset master and initiate a master change. During the master change the cache area segment used by the former master is automatically disconnected, relocated to the new master and reconstructed there. The cache data is then written back and any file locks still existing are reset (as in relocation following a processor failure). The buffered files are then accessible again from all pubset sharers.

Since in a parallel HIPLEX the GS-specific functions of the GSMAN subsystem are not available until "System Ready" status is reached and cache areas cannot therefore be restored until after that point, it is questionable whether cache areas that serve the data of the home pubset in write cache mode can be restored in time. This is why caching of the home pubsets in a parallel HIPLEX is not supported.

Other possible errors during DAB caching in the global storage can be caused by volume defects or cache storage defects. Such errors are handled in the parallel HIPLEX in exactly the same way as for local GS use. For further information on the subject, see the "DAB" manual [2].

4.4.8 Reconfiguration of GS volumes in parallel HIPLEX

When a processor crashes, a GS volume used by that processor remains connected to the processor. Like other volumes it can also be used by other processors after a system crash. The GS volume can only be deleted by another processor using the DELETE-GS-VOLUME command if FORCE=YES is specified. The GS volume can be disconnected from the old processor, after restarting the processor by initiating the ATTACH-DEVICE command followed by a DETACH-DEVICE command.

4.5 DMS functions in the network

4.5.1 Pubsets as file containers

In BS2000/OSD, files and job variables are stored on pubsets, as well as the associated metadata required at DMS level (e.g. file catalog, user catalog, GUARDS catalog etc.). Among other things therefore, the metadata required to address files and file contents are available on pubsets (i.e. the file catalog). A pubset therefore represents a unit in terms of the data and its associated metadata.

To be able to start a pubset on a processor, additional metadata for the pubset which is partly static and partly dynamic must be available on that processor. The metadata includes attributes such as name (= catalog ID), status, connection properties and operating parameters. These attributes are grouped together for each pubset in an entry in the MRSCAT which is stored in memory as a table during a BS2000 session. In addition, the static parts of an MRSCAT entry are also stored on the home pubset as records of the \$TSOS.SYSTEM.MRSCAT file.

A pubset is made known on or removed from a processor by adding or deleting a corresponding entry in the MRSCAT (see [page 202](#)).

For more detailed information on this subject, see the manuals "Introductory Guide to Systems Support" [5] and "System-Managed Storage" [17].

Files and job variables are addressed in two stages from one processor via MRSCAT and the file catalog TSOSCAT, as shown in [figure 8 on page 61](#) for a pubset known to two processors. This corresponds to addressing via the catalog ID (CATID) and file name with user ID.

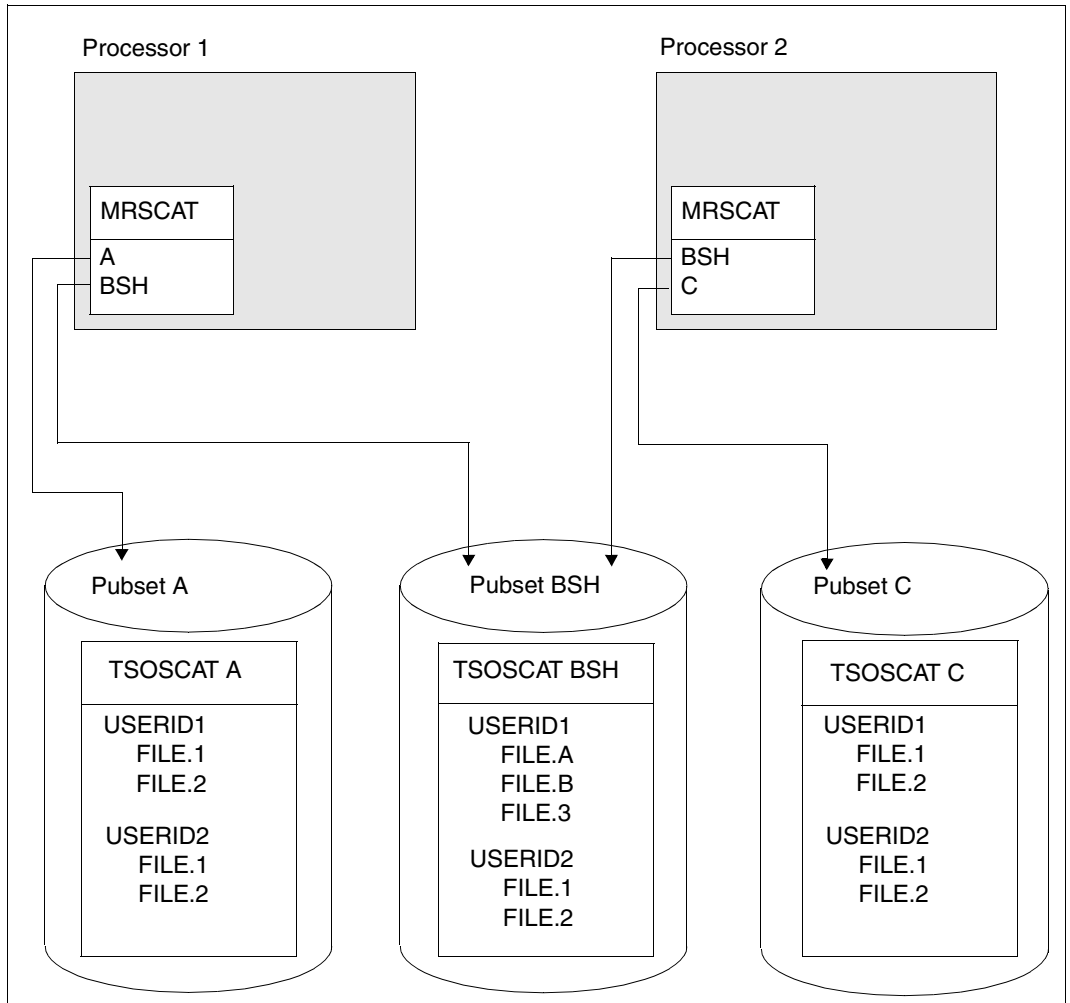


Figure 8: Two-stage addressing of files via MRSCAT and file catalog TSOSCAT

An MRSCAT entry includes the following data which is necessary for the operation of an MSCF network:

- Catalog ID(Catid)
The catalog ID unambiguously identifies each SF or SM pubset. Systems support is responsible for the processors in an MSCF network must ensure that the catalog IDs in that particular network are unique.
- Operating status:
 - INACC: the catalog cannot be reached from the processor.
 - QUIET: the connection to the catalog is temporarily interrupted.
 - ACCESSIBLE:
If the catalog is accessible, substatures decide how the pubset is to be operated:
 - LOCAL-HOME: Status of the home pubset after successful system initiation. The home pubset must be available at all times during the entire session.
 - LOCAL-IMPORTED: The pubset was put into operation as a data pubset with the IMPORT-PUBSET command. This can be done exclusively for one processor or also as a shared pubset. In the latter case, the MRSCAT is also given the specification whether the processor has assumed the role of master or slave processor.If it has become a slave processor, it is also informed of the processor name of the master processor.
 - REMOTE-HOME/
REMOTE-
IMPORTED: From a local point of view, this is a severely restricted operational startup of a pubset; the pubset is exclusively imported as the home pubset or as a data pubset or in master mode on a remote processor to which there is an MSCF connection. Only the LCS function is available.

For further details on the operating status of shared pubsets, see [page 157](#).

- Host name:
An MRSCAT entry is where the name of the processor that manages the pubset is stored. In INACCESSIBLE status, the name is not necessarily up-to-date. When a pubset is put into operation, the information is updated on transition to ACCESSIBLE status.



At the interfaces for managing the MRSCAT, processors are always identified by "HOST" or "PARTNER-NAME"; in line with the terminology introduced for HIPLEX MSCF (see [page 112](#)), this is the processor name of the processor.

Since, in addition to the data, the processing-related metadata is also stored on the pubsets, in the case of shared pubsets there is the problem of coordination across all the processors when this data is updated. This problem is solved by the fact that the master processor is declared the owner of the metadata. Any changes to the metadata must be made via the master processor. This principle is implemented in two ways:

- Operating system functions for updating are only available on the respective master processor.
- On a slave processor, functions for accessing metadata are sent to the master processor as jobs. In relation to the slave processors of a pubset, the master processor therefore acts as a server when it comes to carrying out operations on the metadata of that pubset.

It is mainly the second variant that is implemented in BS2000/OSD. For most of the interfaces relating to file management, the master/slave attributes of a processor are therefore as transparent as the shareability of a pubset.

4.5.2 Data management system and shared pubsets

The data management system is described in the manuals "Introductory Guide to DMS" [4] and "DMS Macros" [3]. Here we will just look at the special features of DMS in relation to the MSCF network.

The system parameter FSHARING is described in the "Introductory Guide to Systems Support" [5].

The interfaces of DMS include operations at file and at data level. A shared pubset network therefore presupposes that these interfaces are fully covered. In this environment, DMS operations are then implemented as follows:

- accesses to metadata (e.g. file catalog) are carried out on the master processor
- accesses to data (records, blocks) are carried out on the local processor.

This means that at file level, the entire range of functions (e.g. creation, enlarging, reducing, and deletion of files) are available as local functions. This also applies more generally to the protection of files with access profiles (GUARDS). Discrepancies only arise when opening files.

When a file is opened, a check should be made as to whether the required OPEN mode is compatible with file open operations that have already been carried out. Each opening of a file is represented by a file lock; the compatibility is checked by the "file lock manager" function unit. The file locks are also metadata which for shared pubsets is managed on the master processor. In a shared pubset network, the compatibility matrix of file locks within one processor is distinct from the compatibility matrix that relates to all the processors (see the "Introductory Guide to DMS" [4], section dealing with multi-user mode). The more restricted, network-wide compatibility is because in the shared pubset network (if the sharers do not belong to the same XCS network) there is no mechanism for synchronizing access to data on all of the processors. It does not therefore make sense to allow simultaneous file updating on all of the processors.

In contrast to HIPLEX MSCF, DMS cannot distinguish a processor failure from a connection failure. From the point of view of the DMS, both events result in a loss of the connection to the master processor, and possibly to the termination of processing due to a pubset export. In the case of a lost connection, DMS waits (see also [page 170](#)), in the case of a pubset export it terminates.

In any case, HIPLEX MSCF will periodically attempt to reestablish lost connections or change the master. If this fails, the connection remains lost for the DMS and it stays in waiting status.

4.5.3 Shared file system

As described in the section above, file access across all of the processors always requires the opening of files to be synchronized (OPEN/CLOSE). Furthermore, for a shared update on all the processors, file accesses must be synchronized (for read and write to blocks and records).

While OPEN coordination relating to all the processors can be carried out within the shared pubset network, mechanisms for the synchronization of file accesses applicable to all the processors is only available in the XCS network. A simultaneous update on all the processors is also therefore only possible in this environment.

Synchronization of the opening of files

When a file is opened, the required parallelism, among other things, is defined during file processing. The following specifications are necessary:

- OPEN mode:
Definition of the file accesses that are permissible (e.g. "INPUT" for purely read accesses, "OUTPUT" for creating a new file and any kind of processing).
- Shared update mode:
Definition as to whether parallel processing is permissible (e.g. "SHARUPD=YES" for any parallel opening of files)
- LOCK-ENVIRONMENT
Definition as to whether an environment is created when a file is opened which permits the simultaneous updating of the file across all the processors.

Example

ADD-FILE-LINK command (or SET-FILE-LINK),
LOCK-ENVIRONMENT=*HOST-SYSTEM: operand
Under OPEN, the environment is created for processor-local processing.

ADD-FILE-LINK command (or SET-FILE-LINK),
LOCK-ENVIRONMENT=*XCS: operand
Under OPEN, the environment is created for processing on all the processors.

Processing on all the processors, as explained above, is controlled by the specification in the LOCK-ENVIRONMENT operand. It should be noted, however, that a file can only be modified from several processors in parallel if the processors belong to the same XCS network.

On processors that do not belong to an XCS network, the specification LOCK-ENVIRONMENT=*XCS is treated as if *HOST-SYSTEM were defined for that operand.

Files can only be updated on all of the processors using the access methods UPAM, FASTPAM and DIV.

Synchronization for file access

During simultaneous updating of the elements of a file (in UPAM and FASTPAM this would be logical blocks while in DIV this would be file sections), a synchronization mechanism must be available which coordinates the competing accesses. For file updating on all of the processors in the XCS network, this is the distributed lock manager. It can (or must) either be used explicitly at user program level for the implementation of block locks or for locking file sections, or at operating system level for access coordination (in the case of UPAM access methods and also to a certain extent DIV). Below we explain what kind of programming effort is required from the user for the individual access methods for a global file update (i.e. adjustment in user program) and what services the operating system provides. For more information on the access methods, see the manuals "Introductory Guide to DMS" [4] and "DMS Macros" [3].

UPAM

In addition to synchronous and asynchronous read and write accesses, the UPAM interface also offers a "Set/reset of exclusive block locks" function. When a file is opened for an XCS-wide file update, the locks implemented by UPAM are effective throughout the network. An application program which coordinates parallel accesses to PAM files with UPAM locks can also always be used throughout the XCS network without requiring reprogramming; only an extension to the JCL is necessary. For example, the LOCK-ENVIRONMENT operand would have to be added to the ADD-FILE-LINK command (the network-wide access can, of course, also be controlled via the program interface).

FASTPAM

Unlike UPAM, FASTPAM provides no lock function. Block accesses must be coordinated at user program level by means of a suitable lock mechanism; the DLM interface is available for this purpose in the XCS network (the DLM interface can also be used for the coordination of a purely processor-local, parallel file access).

DIV

Basically the same applies to DIV by analogy as applies to FASTPAM: the coordination of accesses to file sections (via windows in the virtual address space) is the responsibility of the application. The DLM is the suitable interface for this purpose in the XCS network. DIV offers the setting LOCVIEW=MAP as a special mode for mapping file sections in the address space window; here the window contents are guaranteed to be consistent with the data content at the time the window was set up. The coordination necessary for this between reading in the relevant contents and writing the file accesses is handled by the DIV access method throughout the XCS network.

4.5.4 Shared PLAM

The library management system LMS creates and manages program libraries and processes the members that the libraries contain. Program libraries are PAM files with a substructure that are processed by the PLAM access method (program library access method) and used to store library members. Members are always cohesive volumes of data, such as source programs, macros, object modules and procedures.

Unrestricted multiple access is possible from different processors to libraries on a shared pubset, provided that the processors accessing the shared pubset form an XCS network. The programs necessary for the MSCF network can be made available in a shared PLAM library.

In the XCS network, synchronization takes place at the granularity of a library member. Otherwise modifying accesses are synchronized via OPEN/CLOSE by the file lock manager, i.e. at the level of the entire library.

For the general conditions that permit multiple access, see the "LMS" manual [11].

4.6 Management of the memory hierarchy with HSMS

The DMS interfaces used for data processing are supplemented by the interfaces of HSMS (Hierarchical Storage Management System). HSMS manages the storage of files on the three storage levels of BS2000/OSD and the movement of files between them:

- processing level (S0)
- background level 1 (S1) with online access
- background level 2 (S2) without online access

HSMS implements migration, backup and archiving of files and job variables on the basis of these storage levels.

The "archive definition" and "directory" metadata that HSMS requires is not linked to a pubset (with the exception of system-managed pubsets discussed in the section below). The archive definitions are stored on the default pubset of the HSMS administrator's ID, i.e. as a rule on the home pubset of the processor. The storage location of a directory can be freely selected.

All the functions of the HSMS interface, too, are available in the shared SF pubset environment (SM pubsets are dealt with in the next section). Jobs for data objects on a shared SF pubset are executed on the master processor, irrespective of which processor the job is assigned to in the network. Jobs which embrace files from several shared pubsets are only accepted if all the pubsets have the same master processor. After a master processor crashes, interrupted backup jobs cannot be moved elsewhere. Restartable jobs can, however, be continued on the crashed processor when it is available again, even if, after the system is restarted, it only participates in the MSCF network in slave mode.

The "concurrent copy" (CCOPY) function is available to reduce the time window of backup runs. It enables a consistent status of a file set to be backed up. The files can then be processed in any way in parallel after an initialization phase once the backup is logically terminated. A consistent status can be provided for the backup as follows:

- Before changes are made to a block of a file in the backup set, a copy ("before image") of this block is copied to a working file
- If additional, freely configurable mirror volumes are available, these additional mirrors can be detached when the backup is initialized. The backup can then be made from these mirrors.

CCOPY is also supported in the shared pubset network. Here you should note the following:

- *CCOPY with “before images”:*
For files on a shared pubset, the copy process to create a “before image” is always carried out on the master processor and an appropriate copy job is transmitted from the slave processor to the master processor. If the connection is lost from a slave processor to the master processor, however, the slave cannot send the job to the master processor and this results in the backup being aborted. After a master processor crash, the backup run cannot be restarted because backup jobs initiated with CCOPY are not restartable. The backup run must be restarted from scratch.
- *CCOPY with mirror disks:*
Shared pubset networks are supported.
When the backup is complete, the mirroring is resumed.
Backup jobs with mirror disks are restartable.

For further information on HSMS, see the “HSMS” manual [8].

The functions for mirror disks are described in the “SHC-OSD” manual [18].

4.7 System-managed pubsets as switchover units

When a processor crashes, the availability of critical applications can be guaranteed by relocation to another processor. An essential factor of relocating an application to a second processor is that *all* data required by the application on the substitute processor must be made available.

Determining all the data that is required for the switchover (= switchover unit) is often problematical because the application data is distributed over different pubsets with different availability and performance characteristics. In certain circumstances, it may also have to be provided from various storage levels as well (e.g. migrated files, restoration of files damaged as a result of a processor crash). For this reason, the metadata must also be made available on the substitute processor. As explained above, the metadata (at least that required by HSMS) has to be searched for in various storage locations.

The concept of system-managed pubsets (SM pubsets) represents a considerable help in forming the switchover unit. An SM pubset can be regarded as a closed container for inter-related data.

This "closedness" is basically made possible by the following configuration characteristics:

- An SM pubset contains all the associated metadata, in particular that required by HSMS.
- An SM pubset supports volumes with different formats and performance, reliability and usage attributes; each "volume set" subunit contains a set of volumes with identical attributes.
- If an S1 storage level is to be supported for the migration, it is implemented as a volume set within an SM pubset.
- For storage level S2, a description is stored of the tape pool with which it is implemented.

Essential to the practicability of the concept of an SM pubset as a self-contained, closed switchover unit is also that it can support volume sets with different attributes. Different demands may be made on the attributes of volumes of data within an application, e.g. they must be particularly reliable or access performance must be especially efficient. Only when there is attribute flexibility within a pubset can the use of several (homogeneous) pubsets with different physical compositions be avoided.

After switching an SM pubset over to a substitute processor, all the data required to operate the pubset is available (because of its self-contained, closed nature). Not only can applications be restarted, but also HSMS jobs can be resumed.

After the HSMS administrator has entered the HSMS statement RECOVER-REQUESTS, all jobs that were issued on the crashed processor are integrated in local jobs. If a SHOW-REQUESTS statement is then started, information is output about the status of the jobs at the time of the processor crash. Jobs that had not been started on the crashed processor can be processed on the substitute processor. Jobs that were actually running at the time of the crash and were interrupted, can be restarted on the substitute processor.

4.8 Support by HIPLEX AF

For further information please refer to the “HIPLEX AF” manual [6].

4.8.1 Live Monitor of HIPLEX AF

The HIPLEX AF component “Live Monitor” with its own monitoring mechanism supports HIPLEX MSCF in detecting and handling system failures. Use of the Live Monitor is independent of monitoring by HIPLEX AF.

In conjunction with HIPLEX MSCF, the Live Monitor permits even faster and more reliable failure detection and, if required, shuts down a system in the MSCF network to establish an unambiguous status.

Advantages of the Live Monitor when used with HIPLEX MSCF

- Quicker failure detection and fail reconfiguration by MSCF
The time saving for failure detection is mainly dependent on the period set in the FAIL-DETECTION-LIMIT of the SET-MSCF-ENVIRONMENT command. The time saving with the aid of the Live Monitor amounts to a few dozen seconds in the case of a short period and to several minutes in the case of a long period.
 - œ Reliable failure detection
- Preventing faulty failure detection
 - HIPLEX MSCF monitors the network partners over two independent monitoring paths. The simultaneous absence of in-service messages over both monitoring paths is a strong indication that a partner has crashed. However, as the monitoring paths can also fail independently of the monitored system, there is always a small potential of failure detection being incorrect. This potential can be excluded with MSCF and the simultaneous use of the Live Monitor.
 - Ensuring failure detection
If one (or both) MSCF monitoring path(s) has/have failed, MSCF cannot detect the failure of a partner. Cooperation with the Live Monitor ensures that a failure is detected.
- Enhanced automation
 - Avoiding the message MCS1100 which must be replied to when fail reconfiguration starts.
 - Avoiding a SETS with the message MCS1300 (system shutdown to prevent incorrect failure detection by the partners).
If the Live Monitor detects or confirms the failure of the partner system, the MSCF setting `RECOVERY-START=*SECURE` ensures that the recovery measures are instigated automatically.

- Performing measures to ensure consistency
MSCF requests the Live Monitor to shut down a system regarded as failed in order to establish an unambiguous status.

RECOVERY-START=*SECURE

Failure detection by the Live Monitor is automatically included in the failure detection of HIPLEX MSCF. To fully integrate the Live Monitor functionally in the failure processing, the setting `RECOVERY-START=*SECURE` must be made in HIPLEX MSCF. `*SECURE` can be specified as a general value in the MSCF commands `SET-MSCF-ENVIRONMENT` and `MODIFY-MSCF-ENVIRONMENT` or as a partner-specific value in `START-MSCF-CONNECTION` and `MODIFY-MSCF-CONNECTION`.

The value `*SECURE` has the same effect as

- the value `*AUTOMATIC` if the failure of the monitored partner is determined or confirmed by the Live Monitor.
- the value `*CONSISTENT-BY-OPERATOR` in all other cases.

FADING-INTERVAL

In addition, the operand `FADING-INTERVAL` determines the wait time which the Live Monitor waits before fail reconfiguration is started. `FADING-INTERVAL` may be specified in the `SET-MSCF-ENVIRONMENT` and `MODIFY-MSCF-ENVIRONMENT` commands.

This period is used to wait for open user tasks on shared pubsets to be terminated. This enables user outputs from a failed system to be completed before reconfiguration starts.

Use of the Live Monitor

The Live Monitor need not be activated on every system of the MSCF network, i.e. from the viewpoint of the Live Monitor an unhomogeneous network is possible.

From the viewpoint of MSCF it is sufficient if the Live Monitor reports the failure of a monitored system on one of the (possibly large number of) monitoring system.

4.8.2 Standby mode with HIPLEX AF

In order to ensure that critical applications continue without interruption in the event of a system crash or during maintenance work, the work environment required by those applications can be kept in a ready-to-go state on a standby system. In the event of failure, the application can be started automatically on the standby system.

The product HIPLEX AF (AF = **availability facility**) guarantees this in the shared pubset network.

If an application is running on a processor (work system) monitored by HIPLEX AF, the application together with its resources is automatically switched to another previously defined processor (i.e. the standby system) in the MSCF network if that processor crashes. Both the basic system and the communication software are immediately executable on the standby system. No time is lost therefore loading programs and the downtime for the application is minimized (circumstances which are particularly advantageous during unattended operation).

The automatic error recovery guarantees quick response times and precludes user error. It is of no significance to the user which processor the application has been relocated to.

To be able to carry out the switchover, a description of the application environment to be switched must be created for HIPLEX AF which must include the following points:

- network environment
- data environment (files, pubsets, volumes)
- application environment (start procedure, MONJVs etc.)
- system settings

It is particularly useful to store the application data in an SM pubset because the data environment is primarily identified by the specification of the pubset. The pubset can be operated as a shared pubset and at the same time imported on the work processor (as the master) and the standby processor (as a slave) so that the data is also always accessible on the standby processor.

4.9 Job assignment and monitoring

The job management system includes functions with which batch jobs can be distributed, monitored and controlled in the MSCF network. As a rule, the functions of the job management system are available in all types of network (LCS, CCS, shared pubset network or XCS).

In addition to the MSCF, the processing of batch jobs in the HIPLEX requires the following components:

- JMS (job management system also in the narrower sense), a component of the basic BS2000/OSD-BC system
- job variable subsystem

Jobs are passed to the system and batch loads administered via the JMS (see [section “Job initiation and termination” on page 78](#)). Individual jobs are checked and mutually independent jobs are controlled via job variables (see [section “Job checking” on page 86](#)).

The product AVAS (see [page 96](#)) which also supports the MSCF network is based on the functions of the JMS and job variables.

4.9.1 Job variables

Job variables (JVs) are storage areas where information is exchanged between jobs and between operating systems and jobs. They can be addressed at command and at program level.

JVs are stored in the file catalog. They are addressed in accordance with a syntax which is analogous with the path names for files (catalog ID, user ID, job variable name), and analogous name completion.

User can modify JVs themselves or instruct the operating system to set JVs to agreed values when certain events occur. The values of the JVs can be queried via commands and program interfaces.

The catalog ID in the path name of a JV defines the storage location of the JV in the MSCF network. If the catalog is accessible, JVs can be created, modified, read or deleted from any processor on any pubset in the MSCF network, i.e. JVs are available to all processors in the network.

Changes made to JVs anywhere in the MSCF network can elicit a response throughout the entire network, i.e. interdependencies can be defined between jobs and programs across all the processors.

JVs thus permit flexible job control in the MSCF network. They enable the simple definition of mutual interdependencies between complex job systems. The job variable concept is the basis of the condition-dependent job control (see [page 86](#)).

A distinction is made between the following JV types:

- *User JVs*
User JVs are created by BS2000 users and can be influenced (i.e. created, modified, deleted etc.) by the commands and macros described below.
- *Monitoring JVs (abbreviated to MONJVs)*
MONJVs are JVs linked to certain objects (e.g. batch jobs, executing programs) or even complete applications. As long as the object exists to which an MONJV is linked, the system protects it against modification or deletion by the user. If the object no longer exists (i.e. the batch job or program is finished), the MONJV becomes a normal user JV.
- *Special JVs*
Special JVs are not stored in the file catalog. They are not genuine JVs because they can only be accessed in read mode and locally (i.e. not from all the processors). The purpose of special JVs is to be able to have read access to certain system information (e.g. the current date) and to genuine JVs.

The job variable concept is described in detail in the "Job Variables" manual [\[9\]](#).

Application example illustrating job variables

In this example, the MSCF network consists of a BS2000XC processor with catalog Z and a BS2000XA processor with catalog B (see diagram below).

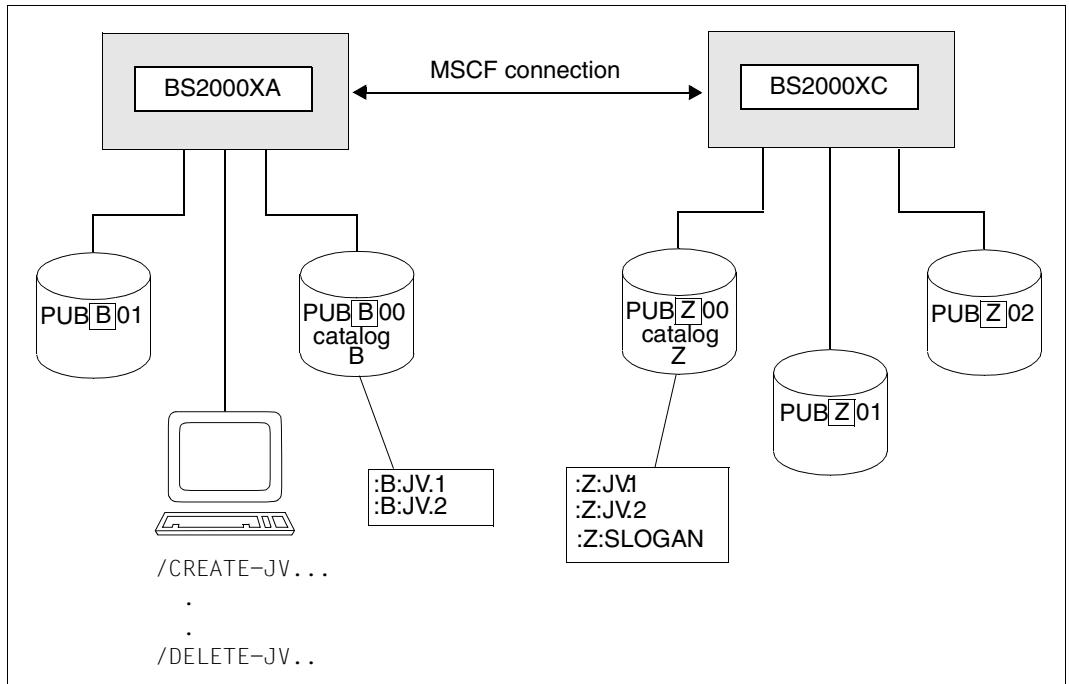


Figure 9: MSCF network with processors BS2000XA and BS2000XC

```

/CREATE-JV JV-NAME=:B:JV.1
/CREATE-JV JV-NAME=:B:JV.2
} 'Local generation' of
} job variables

/CREATE-JV JV-NAME=:Z:JV.1
/CREATE-JV JV-NAME=:Z:JV.2
/SET-JV-LINK LINK-NAME=*MSCF, JV-NAME=:Z:SLOGAN
} 'Remote generation' of
} job variables

/MODIFY-JV JV=:B:JV.1, VALUE=C'MULTIPLE PROCESSOR'
/MODIFY-JV JV=:B:JV.2, VALUE=C'SYSTEM'
/MODIFY-JV JV=:Z:JV.1, VALUE=C'IN'
/MODIFY-JV JV=:Z:JV.2, VALUE=C'BS2000'
/MODIFY-JV JV=:Z:SLOGAN, VALUE=C'/OSD-BC'

/SHOW-JV-ATTRIBUTES JV=:B:JV.
/SHOW-JV-ATTRIBUTES JV=:Z:
/SHOW-JV JV=*LINK(*MSCF)

```

```
/MODIFY-JV JV=*LINK(*MSCF),SET-VAL=:B:JV.1
/MODIFY-JV JV=*SUB(JV=:Z:SLOGAN,POSITION=12,LENGTH=6),SET-VAL=:B:JV.2
/MODIFY-JV JV=*SUB(JV=:Z:JV1,POSITION=4,LENGTH=6),SET-VAL=:Z:JV.2
/MODIFY-JV JV=*LINK(*MSCF,POSITION=19,LENGTH=9),SET-VAL=:Z:JV.1

/SHOW-JV JV=:Z:SLOGAN
/SHOW-JV-ATTRIBUTES JV=*LINK(*MSCF)

/DELETE-JV JV-NAME=:B:JV.
/DELETE-JV JV=:Z:JV.
/DELETE-JV JV=:Z:SLOGAN
```

4.9.2 Job initiation and termination

In an MSCF network, users can distribute their batch jobs to any processor in the network provided that there is a MSCF connection between the two processors. The user can therefore use the resources of any processor in the network, especially those that are only available on certain processors. Job distribution is also useful to prevent overloading individual processors.

Jobs can be routed to any processor in the MSCF network with the ENTER-JOB and ENTER-PROCEDURE commands or started with the ENTER macro. The destination processor is always addressed via the HOST operand. If the HOST specification is missing, the job is executed locally. Resource requests from the job to be executed (e.g. for devices, files, catalogs) cannot be controlled by means of these commands.

Addressing the destination processor

The destination processor can be specified as follows via the HOST operand in the ENTER-JOB command:

- HOST='<hostname>'
- HOST=<jvname>
- HOST='&&jvname>'
- HOST=*CATALOG(IDENTIFICATION='<catid>'/<jvname>)
- HOST=*ANY

If the destination processor is addressed directly via HOST='<hostname>', the job is passed straight to the destination processor where it is placed in the job queue. The destination processor can, however, also be addressed indirectly by depositing the host name in a JV. Under this method, the host name is taken from the JV either by the command server (i.e. by the routines that execute the ENTER-JOB command (HOST=<jvname>)) or by expression substitution (HOST='&&jvname>'), an SDF function which is available generally. Another indirect addressing option is to specify the catalog ID of a pubset that is imported on the destination processor (HOST=*CATALOG(IDENTIFICATION='<catid>')).

Indirect addressing generally offers greater flexibility in the formulation of procedures, as illustrated by the following example:

As agreed with the users, the computer centers that belong to an MSCF network define the names and meanings of certain JVs. Each of these JVs represents a group of resources (e.g. devices, files, jobs) which are assigned to one or more processors in the MSCF network. When a group of resources is then assigned to a specific processor, the value of the JV is set to the assigned processor/catalog name. As a result, jobs which contain the name of that JV in the HOST operand are passed to the relevant processor.

While direct addressing links a job statically to a processor or catalog, indirect addressing permits dynamic job linking to a number of processors/catalogs without changing the original job control statements. Both locally and non-locally imported catalogs can be specified in indirect addressing using a catalog ID. The job is executed on the processor on which the specified catalog was imported when the ENTER-JOB command was input. If the catalog is on a shared pubset (see page 31) that is accessible from both a local and a remote processor, the job is processed on the local processor.

This type of processor addressing is very convenient for users because they do not need to know which processor in the MSCF network has the specified catalog on it or what the processor's host name is. Not only that, but this method of addressing the destination processor has the advantage that catalogs can be exported and imported between the individual systems in an MSCF network.

This variant of destination processor addressing can also be applied independently of whether an MSCF network exists or not (because the pubsets are imported locally). Conversion to the MSCF network is possible without having to modify commands.

Notes

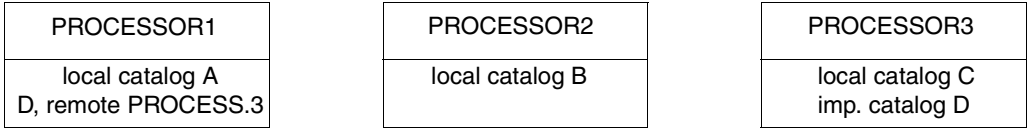
- The destination processor is defined at the time of acceptance. There can, however, be a considerable lapse of time between the acceptance of a batch job and its execution (e.g. due to a set-deadline job, overload situation etc.). It should be remembered that once a batch job is accepted, it must run on the selected destination processor, a fact that cannot even be influenced by indirect destination processor addressing using a catalog ID. The reconfiguration of a catalog (i.e. export from the selected destination processor, import to another processor in the network) *between acceptance and execution* does not automatically have the effect that the batch job will run on the new destination processor.

The only exception is a situation where the home pubset of the original destination processor was given the name of the catalog ID and a startup is carried out for the new destination processor with this home pubset.

- The ENTER-PROCEDURE command does not offer all the options of the HOST operand that the ENTER-JOB permits. This does not mean, however, that the functionality is restricted because generally indirect addressing is possible using the expression substitution (<&jvname>).

Example of indirect addressing via catalog ID

PROCESSOR1, PROCESSOR2 and PROCESSOR3 form an MSCF network in which each processor is connected with the others:



The following commands for job distribution are now input on PROCESSOR1:

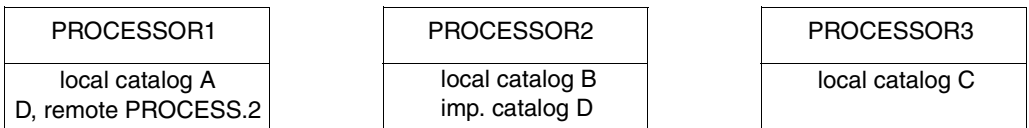
```

/ENTER-JOB FROM-FILE=job1,HOST=C'PROCESSOR2' _____ (1)
/ENTER-JOB FROM-FILE=job2,HOST=*CATALOG(IDENTIFICATION=C'D') _____ (2)
/ENTER-JOB FROM-FILE=job3,HOST=*CATALOG(IDENTIFICATION=C'A') _____ (3)
    
```

- (1) 'job1' is executed on PROCESSOR2.
- (2) 'job2' is executed on PROCESSOR3 because it manages catalog D. In the MRSCAT of PROCESSOR1, there must be an entry for 'D' with reference to PROCESSOR3.
- (3) 'job3' is executed on the local processor (PROCESSOR1).

Catalog D is then transported from PROCESSOR3 to PROCESSOR2 (systems support for PROCESSOR3 specifies the command EXPORT-PUBSET PUBSET=D and systems support of PROCESSOR2 specifies the command IMPORT-PUBSET PUBSET=D). In an existing MSCF connection the EXPORT-/IMPORT-PUBSET commands ensure that PROCESSOR1 carries out the relevant status change (ACC, PROCESSOR3 to INACC, PROCESSOR3 and ACC, PROCESSOR2).

If the user then enters command (2) on PROCESSOR1 and 'job2' is executed on PROCESSOR2. Although the catalog is now on a different processor from before (see figure below), the user does not have to change his job control commands.



Example of indirect addressing via catalog ID and JV

JOB-A, JOB-B, JOB-C and JOB-D are to be distributed to certain processors in an MSCF network:

- JOB-A is to run on PROCESSOR1.
- JOB-B is to run on the processor that manages catalog B.
- JOB-C is to run on the processor to which the laser printer is connected via a canal switch. The processor concerned in our example is identified via the 'LASERPRINTER' job variable (systems support must make sure that this job variable contains the name of the processor to which the laser printer is currently connected).
- JOB-D is to run on the processor that manages the catalog with the file 'UDS.DATA'. The catalog concerned in our example is identified via the job variable 'UDSDTA' (systems support must make sure that this job variable includes the ID of the catalog that contains the 'UDS.DATA' file).

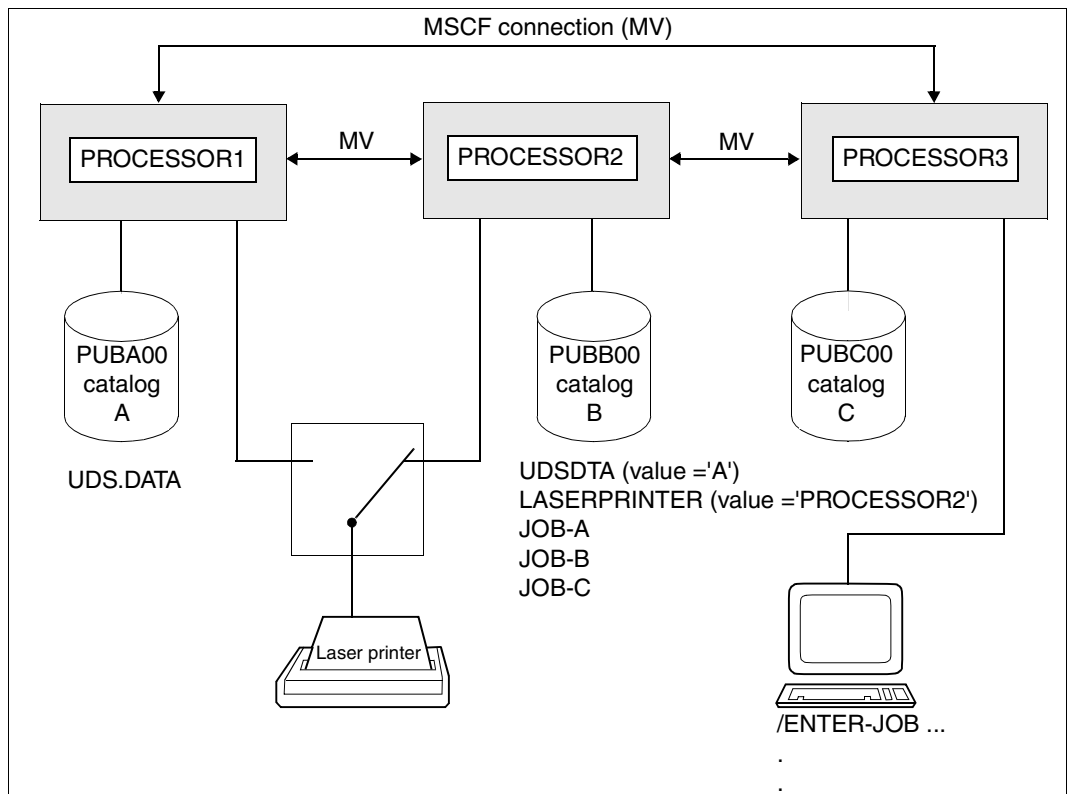


Figure 10: Job management

The jobs are started with the following commands:

```
/ENTER-JOB FROM-FILE=JOB-A,HOST=C 'PROCESSOR1 '  
/ENTER-JOB FROM-FILE=JOB-B,HOST=*CATALOG( IDENTIFICATION=C 'B' )  
/ENTER-JOB FROM-FILE=JOB-C,HOST=:C:LASERPRINTER  
/ENTER-JOB FROM-FILE=JOB-D,HOST=*CATALOG( IDENTIFICATION=:C:UDSDTA)
```

JOB-A is passed to PROCESSOR1 and JOB-B to PROCESSOR2 (PROCESSOR2 manages catalog B). The specification of the destination systems for JOB-A and JOB-B is static.

The routing of JOB-C and JOB-D depends on the values of job variables LASERPRINTER and UDSDTA. JOB-C is passed to PROCESSOR2 and JOB-D to PROCESSOR1. By simply changing the value in the job variables, the jobs can be flexibly routed in the MSCF network as appropriate to the load and resource situation. The destination systems for JOB-C and JOB-D are specified dynamically.

4.9.2.1 Access permission in the MSCF network (participant mode)

Users can only start jobs if they have access to the system. In BS2000/OSD, access permissions can be finely regulated, with a careful distinction being made in particular between dialog and batch access. Information about persons with access permissions (such as your user ID and user-specific user rights) must be entered by the user manager in the user catalogs of all the processors in the network. In the access request (e.g. SET-LOGON-PARAMETERS, i.e. request for dialog access for a specific processor), the user identifies himself to the system by means of his user ID. The system then checks whether the user is also really the owner of the specified user ID (authentication) by looking at the LOGON password supplied when the connection was set up and the accounting number supplied.

Access is only ever granted based on the user ID. If a user wants to work with several user IDs, access must be obtained for each of them.

In the MSCF network, a user who wants to use a random number of processors, must usually possess the necessary access permissions for all the processors. The access permissions are checked differently depending on the type of network (see below). When batch jobs are transmitted, it becomes clear that dialog and batch accesses are handled quite separately. The batch access permission is a requirement on the destination processor, whereas the dialog access permission is not.

As a rule, the MSCF network requires access permissions that can be set flexibly and that are guaranteed by the use of the product SECOS (see the "SECOS" manual "Access Control" [16]).

The following applies to batch access permission in the MSCF network:

- The access rules must be entered separately on all the processors in the network. We recommend selecting identical settings on all of the processors and only deviating from this recommendation for a specific purpose.
- The access permission is always checked on the destination processor. When a batch job is initiated on another processor, this means that the permission of the batch access is checked on the destination processor based on the entries there.
- Where an LCS connection exists, the processors to be linked are given general protection by means of passwords. Before the access check, the right of the user to use services on all processors under his user ID is checked via the LOGON password that is also transmitted (in this example, the right to start batch jobs on any processor). If the password of the job submitter's user ID does not match the password of a user ID with the same name on the destination processor, the destination processor rejects the job. This password check is not made in the case of a CCS connection. The access permission is checked on the destination processor irrespective of the type of connection.
- If the job submitter ID and the destination ID are different, specifications as to the destination ID must be made in the PROCESSING-ADMISSION operand (it contains the user ID, the accounting number and the password, see ENTER-JOB/ENTER-PROCEDURE) commands. Access is approved for the destination ID on the basis of these specifications.
- If the batch job is to run under the same ID, the PROCESSING-ADMISSION operand can also be omitted in the case of jobs applicable to more than one processor; the values of the job submitter task apply.

Example

A user wants to pass a job on to another processor in the MSCF network. In addition, the job is to be started under a different user ID (see [figure 11 on page 84](#)).

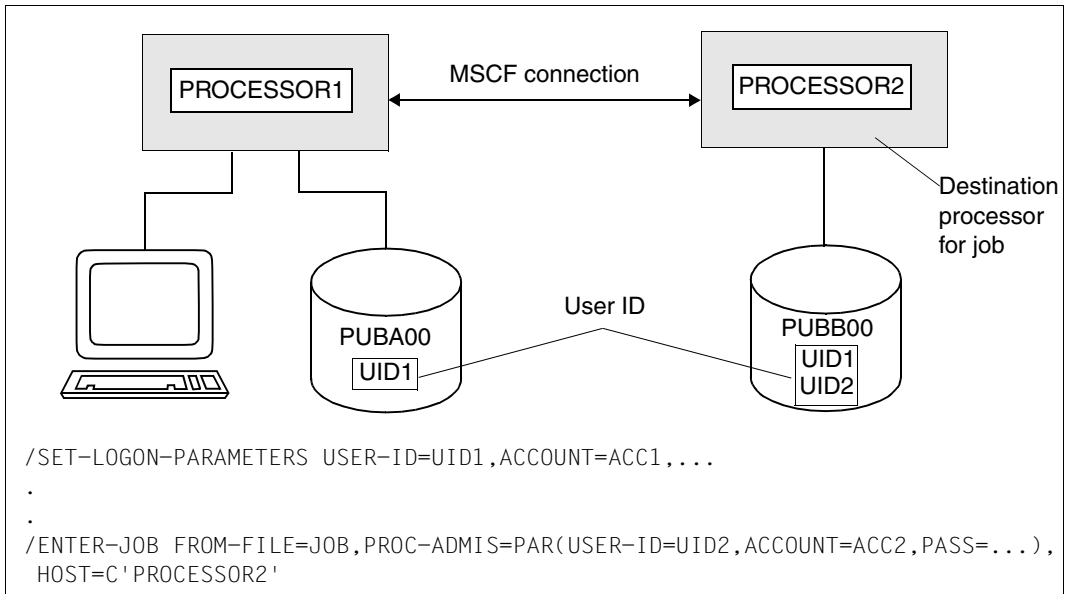


Figure 11: Access permission on a destination processor

The different user ID must be entered on PROCESSOR2 together with the accounting number and password before the job can be started. The user ID under which the command is entered, must be known on the destination processor. In the case of an LCS, the password must also agree.

4.9.2.2 Reference to jobs in the MSCF network

Once a job is passed to a processor in the MSCF network, a task sequence number (TSN) is assigned to that job on the destination processor. The TSN is only valid on the destination processor, i.e. the job can only be addressed there via the TSN. The started job can be addressed from another processor via a monitoring job variable (MONJV), provided that the product JV is loaded and the job is assigned an MONJV whose name is unique throughout the network (via the ENTER-JOB / ENTER-PROCEDURE command or the ENTER macro). This makes it possible to start the commands SHOW-JOB-STATUS, CHANGE-TASK-PRIORITY, CANCEL-JOB and FORCE-JOB-CANCEL for the job that is identifiable via the MONJV on any processor in the network. Defined values which are entered in the MONJV at certain times during the job processing also provide information about the current processing status of the job to which it is assigned (see [page 89](#)). Batch jobs can also be addressed in the said commands via the tuple TSN and host name (see the "Commands" manual [10]).

Example

JOB-A, JOB-B and JOB-C are routed to different systems in an MSCF network. The requirement is to be able to access the jobs from the initiating system so as to be able to query statuses, modify job priorities and enable a job to be aborted. The following ENTER file exists:

```

/SET-LOGON-PARAMETERS
/ENTER-JOB FROM-FILE=JOB-A,MONJV=ID.JOB.A,HOST=C'PROCESSOR1' _____ (1)
/ENTER-JOB FROM-FILE=JOB-B,MONJV=ID.JOB.B,HOST=C'PROCESSOR2' _____
/ENTER-JOB FROM-FILE=JOB-C,MONJV=ID.JOB.C,HOST=C'PROCESSOR3' _____
/SHOW-JOB-STATUS JOB-IDENTIFICATION=MONJV(MONJV=ID.JOB.A) _____ (2)
/CHANGE-TASK-PRIORITY JOB-ID=MONJV(MONJV=ID.JOB.B),RUN-PRIORITY=200 ____ (3)
/CANCEL-JOB JOB-IDENTIFICATION=MONJV(MONJV=ID.JOB.C) _____ (4)
/EXIT-JOB

```

- (1) JOB-A, JOB-B and JOB-C are distributed to different processors (PROCESSOR1, PROCESSOR2 and PROCESSOR3). Their processing status is monitored by the specified monitoring job variables (MONJV operand).
- (2) The processing status of JOB-A is queried.
- (3) The processing priority for JOB-B is set to 200.
- (4) Processing of JOB-C is aborted.

4.9.3 Job checking

In an MSCF network, dependencies between jobs and programs can be defined to apply to all the processors. Information can be exchanged between jobs and programs, and jobs and programs can be synchronized.

The job checking function consists of the following components

- conditional job control
- job/program monitoring.

4.9.3.1 Conditional job control

The conditional job control (CJC) function only allows the user to carry out actions under certain conditions. It can issue 'conditional' statements to the operating system which are only executed if or when the specified conditions are met.

The conditional job control function is based on job variables which are linked via operators to a condition. The condition may be 'met' or 'not met' depending on the values of the specified job variables. The action defined by a conditional statement is executed when the condition is fulfilled, e.g. when the value of a job variable has changed accordingly. The following actions can be defined as conditional statements:

- Initial evaluation of the condition and immediate branching to 'condition met'. If 'condition not met' applies, job processing continues with the next command (SKIP-COMMANDS)
- Transition to a wait state until the specified condition is met or a predefined period of time has elapsed, if the condition was not already fulfilled on the initial evaluation (WAIT-EVENT)
- Interruption of a job/program to execute previously specified actions when the specified condition is fulfilled (frequency can be defined) or a predefined interval of time has elapsed (ADD-CJC-ACTION).

These functions can be used locally on a single processor and are also available in the MSCF network. In particular, events on one processor can cause the execution of correspondingly defined actions on the same processor and/or on other processors in the network.

This provides a network-wide communication and control function applicable to all jobs that is based on job variables. The job variables act as shared accessible information carriers to the processors on which the jobs concerned are running.

See the "Job Variables" manual [9].

Application example

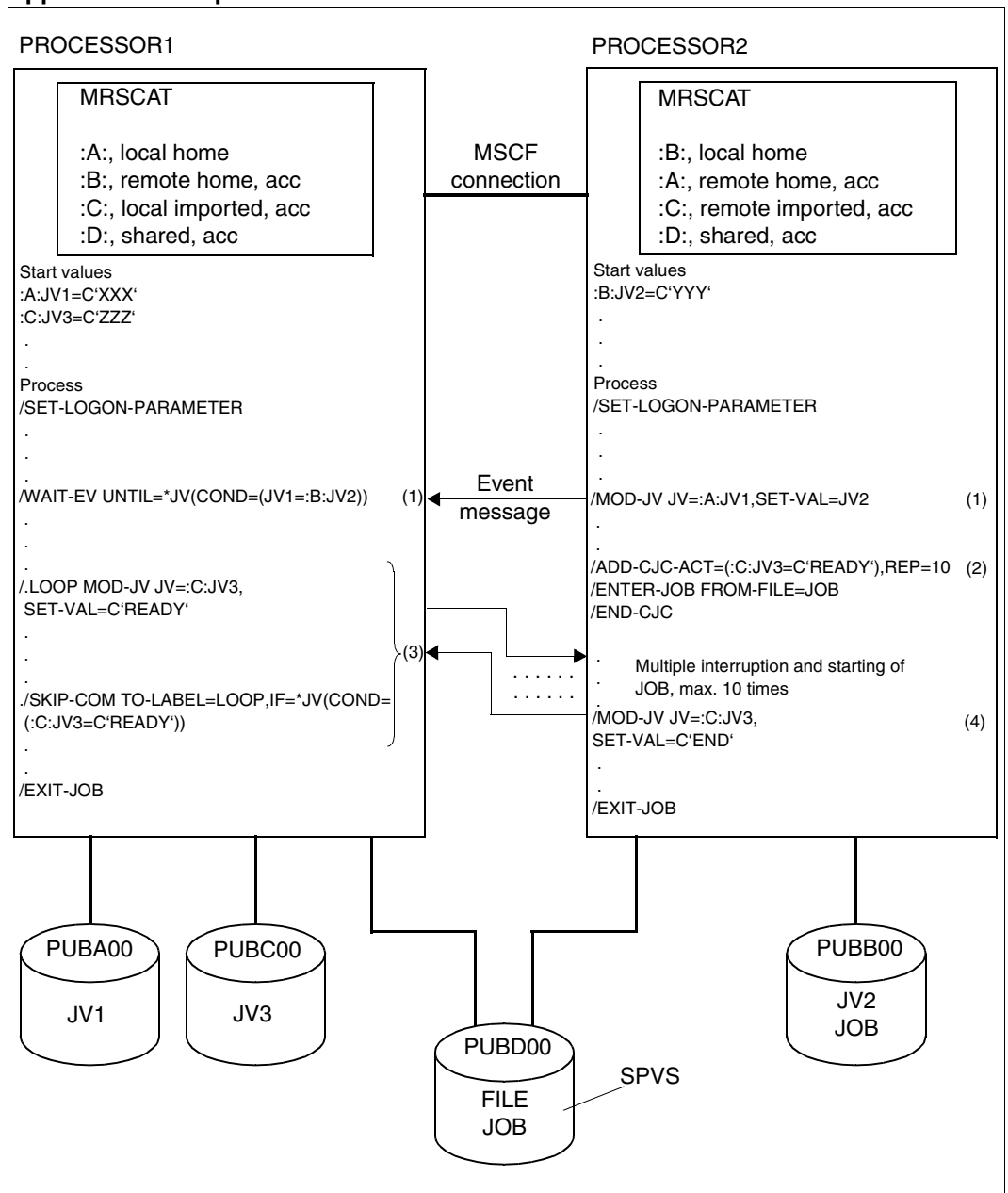


Figure 12: Job checking (see next page for key to figure)

The [figure 12 on page 87](#) illustrates job dependencies defined on processors PROCESSOR1 and PROCESSOR2 as applicable to all the processors. PROCESSOR1 and PROCESSOR2 are linked via an MSCF connection.

Key

- (1) The job on PROCESSOR1 waits until the job on PROCESSOR2 has set job variable :A:JV1 to the same value as job variable :B:JV2.
- (2) The job on PROCESSOR2 now indicates to the system that it wants to be informed of the event up to 10 times at intervals of 600 seconds (= 10 minutes)
Job variable :C:JV3 has the value C'READY'
and interrupted from starting the ENTER file JOB on the SPVS.
- (3) The job on PROCESSOR1 delivers these events in a loop which will keep rotating until the job on PROCESSOR2 has delivered the information:
Job variable :C:JV3 does not have a value corresponding to C'READY'
- (4) Job variable :C:JV3 is set to C'END'.
- (2) + (3) The condition
Job variable :C:JV3 has the value C'READY'
is now no longer fulfilled. The branch is therefore not made and both jobs are terminated with the EXIT-JOB command.

Behavior on reconfiguration

If statements of the conditional job control refer to catalogs imported to job variables, the statements still remain effective when such a catalog is identified as 'temporarily not accessible' while waiting for events. If the specified time has not elapsed and the event has not occurred when the catalog is available again, the statements are processed normally.

If an awaited event occurs while a connection is down, the waiting processor cannot be informed. Therefore the message CJC0004 is issued at the shared pubset master and the event is deactivated entirely. The processor will not be informed, even after the connection is reestablished and the event has occurred again. After the waiting time has lapsed processing continues on the waiting processor as though the event had never occurred.

If a job variable is not available while it is waiting for an event, the processing of such statements is terminated with a message and the system branches to the next SET-JOB-STEP command or to the end.

4.9.3.2 Job and program monitoring

MONJVs are used to monitor program and job runs.

The user must define a job variable that is specified as an operand in the ENTER-JOB and START-PROGRAM commands. The operating system then assigns fixed values to these job variables at certain times and the values can be queried by the user in his command sequence. The job run can be controlled by means of these values.

Program/job monitoring with job variables can be used both on a standalone BS2000 system and in an MSCF network. Program and job runs and their termination results can be queried from any processor in the network. Appropriate responses can then be initiated via the event-controlled job processing function (see also the "Job Variables" manual [9]).

Job variables for monitoring programs

Job variables for monitoring program runs contain a value that is set by the operating system and made up of the following two separate values:

- Status display, length: 2 characters
This indicates the current status of the program and can have the following values:
 - \$R Program was loaded
 - \$T Program was terminated normally
 - \$A Program was terminated due to program error or defined error output.
- Return code display, length: 4 characters
A code defined by the user in the program is returned to the job control.

A job variable for monitoring the program run can be specified in the following commands:

Command	Explanation
LOAD-PROGRAM ... ,MONJV=...	Loads a program and monitors the program run by means of specified job variables
RESTART-PROGRAM ... ,MONJV=...	Restarts a program and monitors the program run by means of job variables
START-PROGRAM ... ,MONJV=...	Loads and starts a program and monitors the program run by means of job variables

Job variables for monitoring jobs

Job variables for monitoring jobs contain a value which is set by the operating system and made up of separate values (see “Job variables“ [9]). The most important ones are:

- Status display, length: 2 characters
This indicates the current status of the job and can have the following values:
 - \$S Job placed in job queue (for batch jobs only).
 - \$R Job being processed.
 - \$T Job was terminated normally.
 - \$A Job was terminated abnormally (e.g. CANCEL-JOB, system shutdown).
 - \$M Job was exported with MOVE-JOBS command
- TSN display, length: 5 characters
Task number of monitored job.
- Catalog display, length: 8 characters
Catalog ID of the processor on which the job is being executed.

TSN and catalog display are set by the operating system when the job is initiated, i.e. either at LOGON or at the time of ENTER.

A job variable can be specified for job monitoring in the following commands:

Command	Explanation
ENTER-JOB ... ,MONJV=... ENTER-PROCEDURE ... ,MONJV=...	Initializes a job and monitors the run by means of specified job variables
SET-LOGON-PARAMETERS ... MONJV=...	Starts a job and monitors the run by means of specified job variables

If the monitoring job variable is protected by a password, the JV-PASSWORD operand must be specified in the commands.

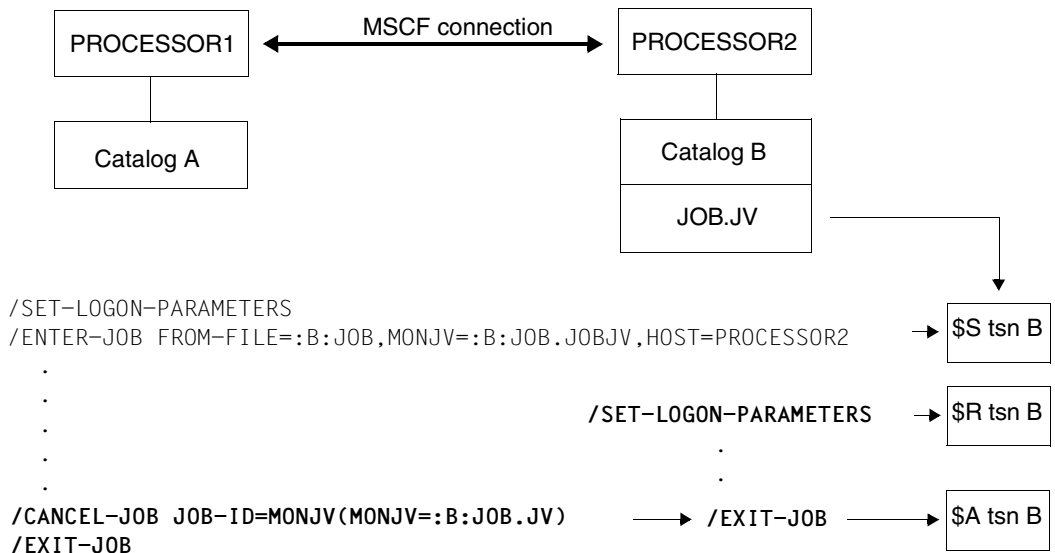
When a monitoring job variable is assigned to a job, the user can reference the job via this variable. The following commands can be used for this purpose:

Command	Explanation
CANCEL-JOB	Cancels the specified job
CHANGE-TASK-PRIORITY	Changes the run priority of the job
MODIFY-JOB	Changes the attributes of the job (to type 1)
SHOW-JOB-STATUS	Provides information about the job
SHOW-RESOURCE-ALLOCATION	Provides information about the resources assigned to the job

Notes

- The system part of job-monitoring job variables is protected by the system and cannot be changed by the user as long as the job variables are used for monitoring. At the end of the job (EXIT-JOB or LOGOFF), monitoring job variables are released again.
- Wildcard specifications may not be used for job monitoring in job variable names, temporary job variables, JV link names or partially qualified names. For detailed information, see the "Job Variables" manual [9].
- The identification of jobs by monitoring job variables is especially recommended in an MSCF network because the TSN of an ENTER job in an MSCF network is not unique, but the job variable is. A job can therefore be identified anywhere in the relevant MSCF network by using a monitoring job variable (see example on next page).

Example



4.9.3.3 Job variables for monitoring the import and export of pubsets

The functions for importing/exporting pubsets or catalogs can also be monitored with job variables.

The system uses the first two characters of the JV value to store the status value. The following status values are possible:

Value	Function	Explanation
\$A	Import	Import cancelled with error
	Export	Export cancelled with error or /CANCEL-PUBSET-EXPORT
\$E	Export	Export job started successfully
\$I	Import	Import job initiated successfully
\$R	Import	Complete pubset imported successfully
\$T	Export	Complete pubset exported successfully
\$W	Import	When a shared pubset is imported, the processor waits for confirmation from the master processor that the import has been successfully completed (i.e. until this confirmation is received, either an MSCF connection to the master does not yet exist or the import process on the master is not yet complete)

For more detailed information, see the IMPORT-PUBSET and EXPORT-PUBSET commands in the "Commands" manual [10].

4.9.4 Monitoring the network status

Monitoring job variables (MONJVs) can also be used for monitoring the network status. HIPLEX MSCF uses two classes of MONJVs to display the statuses of shared pubsets on the processor and the statuses of the processor's MSCF partners. When a shared pubset is imported or an MSCF network is set up to a partner, HIPLEX MSCF automatically makes the MONJVs available. The MONJVs of both classes have a defined name, which contains the name of the pubsets or MSCF processor concerned as a partial string. They are stored under the TSOS ID on the home pubset of the processor.

If a MONJV is not available, it is created when the new status is set with the default attributes. MONJVs that are no longer required are not automatically removed by the system and must therefore be deleted, where necessary, by systems support.

Shared pubset-specific MONJV

There is a shared pubset-specific MONJV called `$TSOS.SYS.PVS.<pvsid>.MASTER.CONTROL` (`pvsid` = PVSID of the shared pubset) for each shared pubset on the home pubset of a sharer. This is automatically set up when the pubset is imported if it does not already exist. The job variable contains information about the status of a shared pubset with the following values:

Value	Meaning	Explanation
\$R	(Running)	The shared pubset was imported to the processor or the master change was completed successfully.
\$A	(Abort)	The master processor of the shared pubset has crashed.
\$C	(Change)	Import of the shared pubset with master change has been started.
\$T	(Terminated)	The shared pubset is no longer monitored; it will be or is exported.

This enables systems support to respond appropriately to a crash, for example, using the CJC (see the "Job Variables" manual [9]).

MONJV for monitoring the network status of an MSCF participant

The MONJV contains the network status of the HIPLEX MSCF partner concerned from the point of view of the local processor. It shows whether the partner is active for the processor or has terminated abnormally. A MONJV is also created for the local processor which contains information as to whether HIPLEX MSCF is active on the processor.

The job variable has the name \$TSOS.SYS.MSCF.CONTROL-STATE.<hostname>. It is set at different times for CCS and XCS partners.

In the XCS network, the availability of the XCS functionality is still taken into account which for products based on the XCS network is more important than the setup of an MSCF connection. No MONJV is created for LCS partners.

The job variable contains information about the network status of an XCS/CCS participant with the following values:

Value	Meaning	Explanation
\$R	(Running)	The participant has entered the MSCF network. XCS processor, XCS partner: the join reconfiguration was completed. XCS Ready was reached on the local processor. When the local processor enters the XCS, MONJV is set to '\$R' for all currently participating partners in addition to the local MONJV. CCS processor: start of HIPLEX MSCF on the processor is complete, the MSCF configuration file has been processed. Shared subsets can be imported. CCS partner: the partner is the processor's CCS partner.
\$L	(Leaving)	HIPLEX MSCF is terminated on the participant. XCS-/CCS processor, XCS partner: the participant leaves the XCS/CCS network and all imported shared subsets are exported. CCS partner: is not used.
\$A	(Abort)	HIPLEX MSCF is just terminating or has terminated abnormally on the partner. XCS/CCS processor: HIPLEX MSCF is terminating abnormally due to error. XCS/CCS partner: the partner has terminated HIPLEX MSCF abnormally and is removed from the XCS network or the shared subset networks affected by the local processor by a fail reconfiguration.
\$T	(Terminated)	The participant has terminated the MSCF network. XCS/CCS processor: the HIPLEX MSCF configuration management was terminated during termination of HIPLEX MSCF. The termination of the HIPLEX MSCF communication system was initiated. All XCS and CCS partners are set to '\$T' at the same time as the processors. XCS/CCS partner: the partner has terminated the MSCF network with the processor. The partner was therefore dismissed from the XCS/CCS network and the shared subset shared with the processor was exported to the partner. The MSCF connection between the two processors was cleared down.

MONJV for monitoring the MSCF connection to a partner processor

The MONJV provides information from the viewpoint of the local system about the status of the MSCF connection to a partner processor (CCS partner, XCS partner). The MONJV has the name \$TSOS.SYS.MSCF.CONNECTION-STATE.<processor-name>.

Where <processor-name> is the processor name of the system to which an MSCF connection is to be set up (see also the description of the relevant operand with the START-MSCF-CONNECTION command on [page 293](#)).

The connection-specific MONJV is created the first time the MSCF connection concerned is started with the START-MSCF-CONNECTION command, it only exists as of this time.

The job variable shows the status of the MSCF connection concerned with the following values:

Value	Meaning	Explanation
\$R	(Running)	An MSCF connection that is completely set up from both sides exists to the relevant partner. Unrestricted use of the MSCF connection is possible.
\$S	(Starting)	Setup of an MSCF connection with the associated processor name has been initiated on the local system to the named partner.
\$A	(Aborted)	An MSCF connection to the specified partner could not be set up due to an error. The cause of the problem could, for example, be taken from the error code of the message MCS0009; the measures for eliminating the problem that occurred are dependent on this.
\$T	(Terminated)	The MSCF connection to the partner concerned was terminated normally by the local system. A new MSCF connection must be set up by the local system to the partner concerned to exchange messages with this partner.
\$P	(Partner connect necessary)	The MSCF connection to the partner concerned was only set up from the local system. However, to use the MSCF connection, the connection setup to the local system by the partner is also required.
\$W	(Recovery)	MSCF connection monitoring has detected that the relevant partner has not responded to a monitoring message and it is currently performing a recovery of the MSCF connection; MSCF itself shuts down the MSCF connection and an attempt is made to reestablish it. The connection is currently not usable. If the MSCF connection is reestablished, the MONJV is set back to the value "\$R", otherwise it is set to "\$A".

4.10 Job management and job handling with AVAS

With AVAS, the computer center can automate its job production and handle necessary planning, preparatory and checking tasks in both interactive and batch mode. Decentralization approaches (such as relocating work process scheduling to the specialist departments) are supported to the same extent as the central collection of job information is supported.

The AVAS services are available throughout the MSCF network. AVAS therefore permits the definition, scheduling and monitoring production from a central location. The user can hand over jobs for processing via AVAS to specific processors in the MSCF network and monitor the normal or abnormal end of these distributed jobs. For this, AVAS uses the HIPLEX MSCF "job distribution" and "job monitoring" functions.

AVAS dialog and job control processing is transmitted to one of the processors in the MSCF network, the AVAS master processor. Of course, jobs can also be processed on this processor. The other processors in the MSCF network are only responsible for job processing (i.e. they are AVAS slave processors).

The AVAS user can define, start and check the progress of jobs that are mutually dependent in a logical or chronological sense. Such jobs that are related on the basis of certain criteria are known as job networks, or in an AVAS context simply as networks. The figure below shows the sequence of the definition and processing of networks under AVAS control.

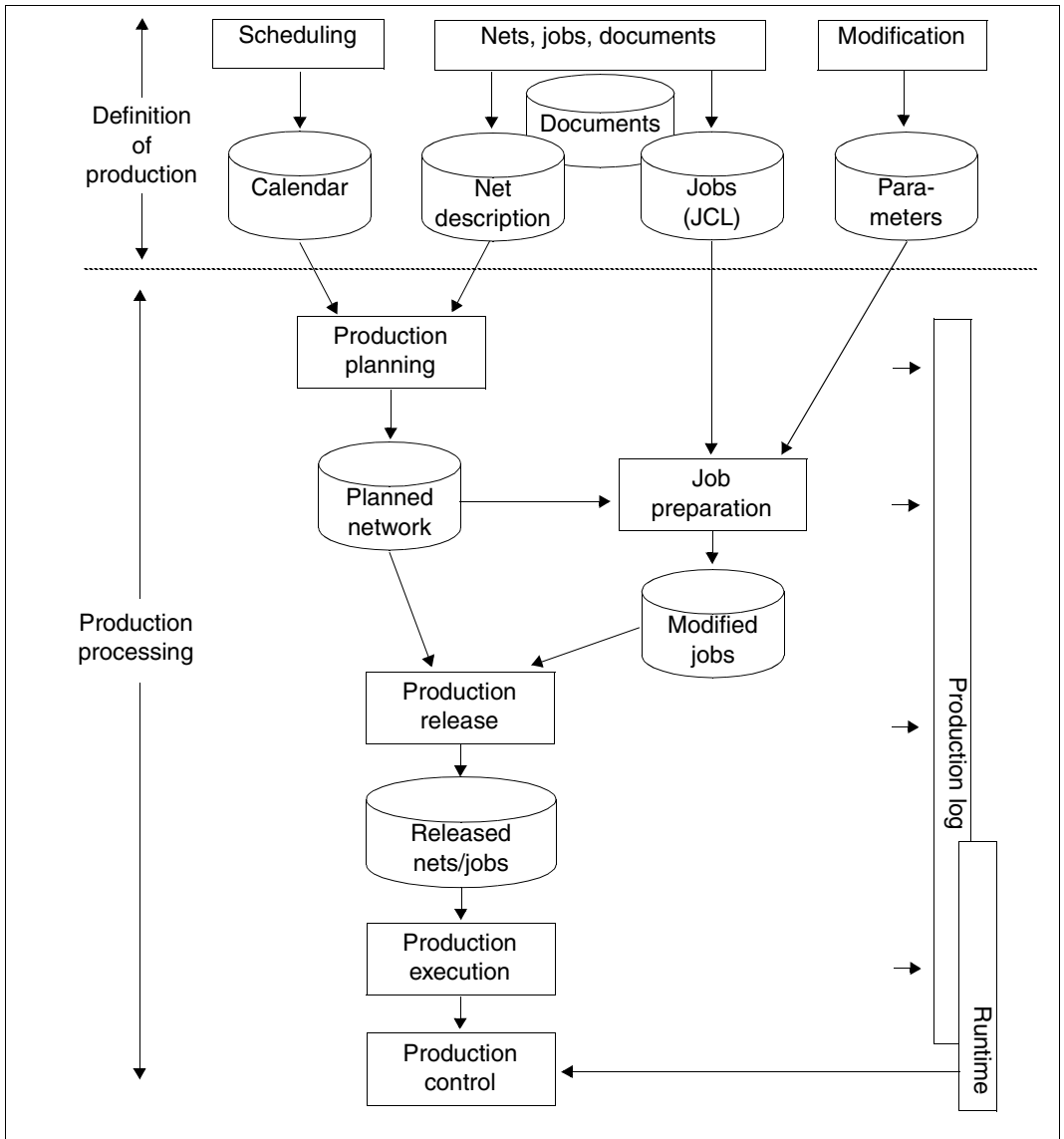


Figure 13: Job automation with AVAS

4.10.1 Job distribution with AVAS in HIPLEX

In HIPLEX MSCF, the AVAS user can distribute complete job networks or individual jobs from networks to the processors within a CCS network and have their progress checked by AVAS. AVAS uses the job distribution mechanisms available in HIPLEX MSCF for this. The network- or job-specific processor is addressed either directly by explicit specifications of the catalog IDs in the network or job descriptions of AVAS, or indirectly via a job variable. Jobs are started on the processors where the specified catalogs are located at the time of the ENTER calls.

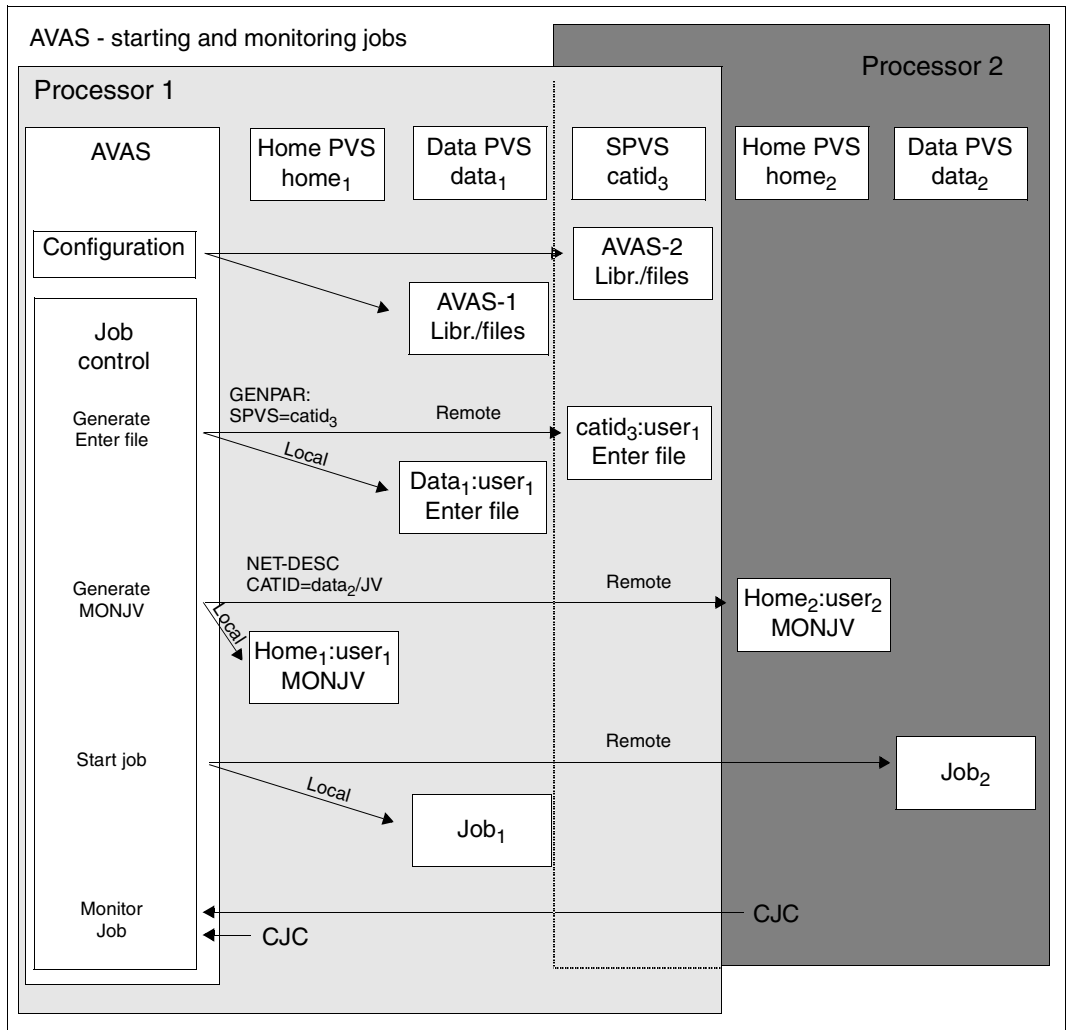


Figure 14: Distribution of networks to several processors

The respective destination processor accesses the ENTER file because AVAS stores it as "shared", e.g. on a shared pubset.

The distributed processing of networks or jobs can be used in HIPLEX to implement the optimum load distribution taking into account the logical and chronological dependencies.

4.10.2 AVAS environment in HIPLEX

The processor with the AVAS job control is called the AVAS master. Any other processors that may become active during the runtime of a job network are called AVAS slaves. The total number of processors and pubsets involved during the runtime of a network form the hardware environment of an AVAS network. It also contains all the pubsets/volumes for AVAS and all the pubsets/volumes for the jobs in the AVAS network. The hardware environment can therefore be different for different networks. In the same way, a different hardware environment may be necessary for a network in different runs or it may even be required intentionally: jobs are started on the processors where the specific catalogs are located at the time of the ENTER calls.

The software environment must be available on every processor involved in the runtime of an AVAS network: FHS, openNet Server, JV, HIPLEX MSCF, TIAM. FHS is not required on the AVAS slaves. For certain AVAS functions, MAREN, SDF-P, IFG can be used as further options.

AVAS accesses its data via the access methods PLAM and UPAM. If external data is read into AVAS or data generated by AVAS is output externally, the user can define the file name with CATID.

The CATID for the AVAS files can be specified by the user in the AVAS configuration file GENPAR. If you intend to relocate AVAS, this metadata must be located as far as possible on relocatable pubsets or a shared pubset.

ENTER files are stored by AVAS on a shared pubset in the MSCF network. The AVAS job control must be informed accordingly of the CATID of the shared pubset via a start parameter.

For more detailed information on AVAS, see the "AVAS" manual [1].

4.11 HIPLEX MSCF and RFA (Remote File Access)

For network-wide access to catalogs and files, the software product RFA is available *independently of MSCF* in addition to the shareable disk (SPD) and the File Transfer (openFT) software product.

RFA allows the user to generate, process, and delete files on a processor other than the local processor (i.e. on a remote processor). In the local processor, the user can open RFA connections to one or more remote processors (target processors) simultaneously. When opening an RFA connection, an AFR (Access From Remote) partner task is generated on each respective target processor. The file access operations performed on the target processor are controlled by this task. The user terminates the connection once processing has been completed.

RFA does not support procedure calls or the loading and execution of programs.

A detailed description of Remote File Access can be found in the "RFA" manual [15].



If accesses via RFA and MSCF take place concurrently, those via RFA always have priority.

Example

The following figure shows how the RFA function and access facilities to a private file on a shareable disk can be combined in a single task with the Remote Catalog Access (RCA) function of MSCF.

An MSCF connection exists between PROCESSOR 1 with file catalog A and PROCESSOR2 with file catalog B. Both processors have a direct hardware connection to the same shareable private disk device. FILE1 is located on the shareable disk device and is cataloged in catalog B of PROCESSOR2. FILE2 is located in the public volume area of PROCESSOR2. JOB1 is cataloged in the catalog of PROCESSOR1 and executes on this processor. The program "prog" processes the files :B:FILE1 and :B:FILE2 using link names.

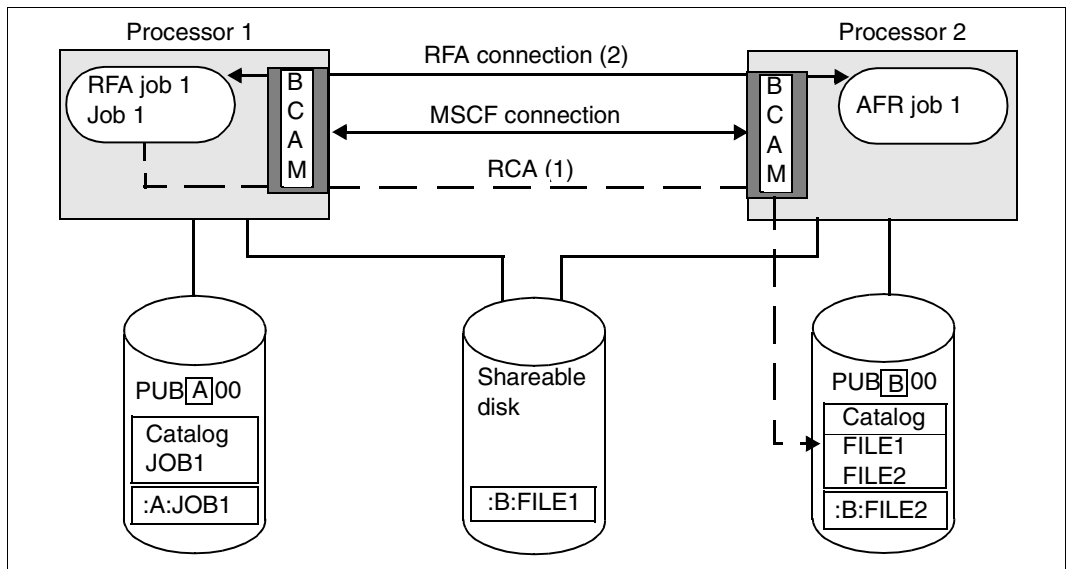


Figure 15: File access between processors

Structure of the file JOB1:

```

/SET-LOGON-PARAMETERS
/ADD-FILE-LINK LINK-NAME=RCALNK, FILE-NAME=:B: FILE1 _____ (1)
/SET-RFA-CONNECTION CATALOG-ID=B _____ (2)
/SET-FILE-LINK LINK-NAME=RFALNK, FILE-NAME=:B: FILE2 _____ (3)
/START-PROGRAM FROM-FILE=prog
...
/REMOVE-RFA-CONNECTION CATALOG-ID=B
/EXIT-JOB

```

Explanations:

- (1) Access to :B:FILE1 is effected via the direct hardware connection to the shareable private disk by means of the RCA function (via the MSCF connection), as the appropriate ADD-FILE-LINK command was issued before the RFA connection was established (SET-RFA-CONNECTION command).
- (2) An RFA connection is established to processor 2. If command (1) were issued before command (2), access to the private file :B:FILE1 would also be effected by means of the RFA function. This would, however, increase the runtime since the direct hardware connection from PROCESSOR1 to the shareable disk would not be used.
- (3) Access to :B:FILE2 located on the public volume area of PROCESSOR2 is effected by means of the RFA function (via the AFR partner task for JOB1).

4.12 HIPLEX MSCF and SPD (Shareable Private Disk)

Shareable private disks can also be used without MSCF. In this case, however, the status of the file catalog is that of the last update (command `IMPORT-FILE SUPPORT=*DISK(...)` or `IMPORT-FILE SUPPORT=*BY-FILE(...)`). If MSCF is used, the status of the file catalog is always current if the catalog is only accessed via one processor (remote catalog access, see [section "DMS functions in the network" on page 60](#)).

Each file on a shareable private disk (SPD) can be accessed at any time by one processor in write mode or by up to four processors in read mode. Access is via direct hardware paths. Files can also be processed by a single processor in "shared update" mode. User programs need not be specially prepared to make use of this function.

The fact that files to be accessed are on shareable private disks is not apparent to the user. Programs running in different processors need not implement special measures in order to access the same file simultaneously. Files stored on SPDs can be cataloged into one, several, or all catalogs of the accessing processors. A disk device is an SPD device if it is connected to more than one processor (i.e. there are at least two access paths via which the disk device can be accessed).

The term "SPD" as defined here refers to a characteristic of the disk device and should not be confused with the "shareable" characteristic of files or volumes. The shareability of a disk device is completely dependent on the given hardware configuration. An SPD disk device loses its SPD characteristic only if there is only one access path left after a hardware reconfiguration; a dynamic change is not possible.

A disk is considered to have SPD capability if:

- SPD mode was explicitly specified for it using the command `SET-DISK-PARAMETER SYSTEM-ALLOCATION=*SHARE`, or
- the default value `*ALL` is set for `SYSTEM-ALLOCATION` and the disk has been set up on a disk device with SPD capability.

The figure below illustrates simultaneous accesses to an SPD:

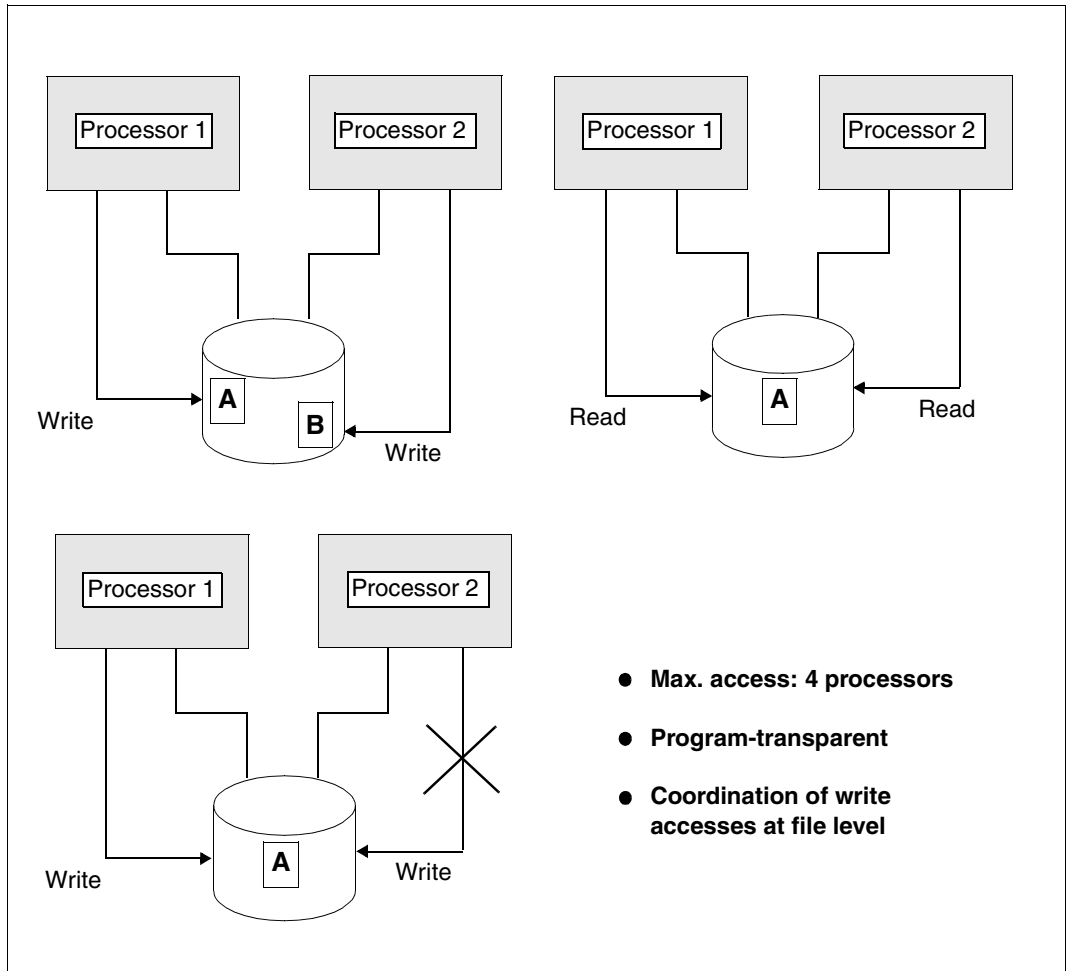


Figure 16: Shareable private disk

Preconditions for using SPD

The hardware provided must be suited to SPD operation (e.g. for fixed disk strings, string controllers and channel port extensions in the disk controllers). See also the relevant manuals for the particular products involved.

The following figure shows an example of a hardware configuration for SPD:

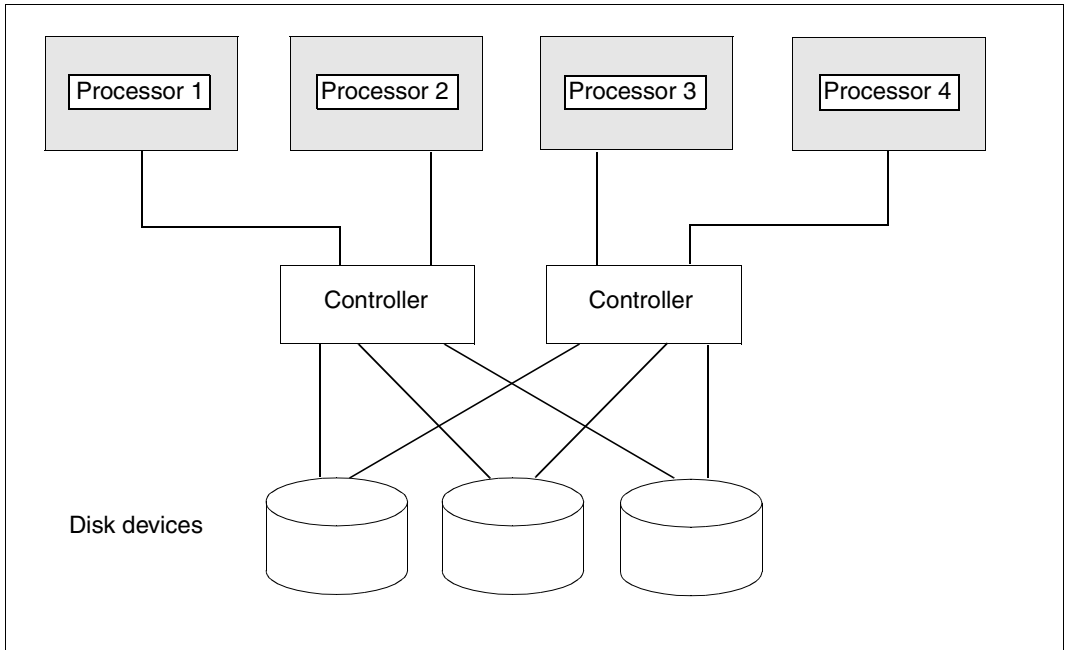


Figure 17: Example of a hardware configuration for SPD

At system generation time, unique system IDs (SYSIDs) must be assigned on *every* processor that wants to use the SPD function. This means that the SYSID must be entered in the home pubset on every processor either with SIR-IP or with the SET-PUBSET-PARAMETERS command during generation of the pubset. This is necessary because each processor accessing a shareable disk device is identified by its SYSID. The SYSID of the accessing processors is stored in the "sharer list" of the shareable disk when each processor first accesses the disk.

Notes on SPD operation

- A shareable private disk can be accessed by up to four processors, but each of these processors can also access other shareable private disks.
- SPD mode or system-exclusive allocation for a disk should always be set explicitly via the SET-DISK-PARAMETER command, so that within a session the device can be generated to match whichever event occurs (e.g. a device failure). This method ensures a high degree of flexibility in the allocation of disks and devices.
- If a user *without* RCA (Remote Catalog Access), i.e. without an existing MSCF connection between the processors which access the SPD, wishes to access a file imported on another processor, this file must be made known to his or her local system (TSOSCAT) beforehand by means of the IMPORT-FILE command or FILE macro (STATE=FOREIGN operand; see the "DMS Macros" manual [3]). This is not necessary with files on private volumes if the RCA function is used.
- To minimize the runtime of tasks using an SPD, the following should be observed:
 - a) User
 - The primary and secondary allocations in the CREATE-FILE command should correspond to the expected size of the file.
 - Catalog entries for SPD files continually accessed by several processors should be held in all catalogs of the accessing systems in order to avoid additional time requirements for communication (data transmission).
 - b) Systems support

Only private disks which can be accessed by several processors at the same time should be used in SPD mode. If only one processor accesses such a disk, the runtimes of the accessing tasks will be greater because of the additional management involved.
 - c) Users with privilege OPERATING

In the command UNLOCK-DISK SYS-ID=..., no processor may be specified that is still accessing the device specified. Otherwise, the processor involved automatically puts the disk in the "CANCELLED" state (i.e. all user accesses are blocked).
- In SPD operation, increases in runtime occur primarily at OPEN, CLOSE and SECONDARY-ALLOCATION time as a result of the necessary coordination measures. Actual file access times are no different from those for normal private disks.

Commands required for SPD operation

The following table shows the operands of BS2000 commands that are relevant for SPD operation. A gray background indicates that this command can only be used by users with the privilege OPERATING.

Command	Operands	Meaning
SHOW-FILE-ATTRIBUTES	FROM-CATALOG=*PRIVATE(...)	Displays the VTOC catalog entries
CHECK-DISK-MOUNT	UNIT=...,ACTION=...	Checks and updates the mount state of all or some devices; logs the online state and, if relevant, the device error state
SET-DISK-PARAMETER	UNIT=..., SYSTEM-ALLOCATION=..., USER-ALLOCATION=...	Determines whether a private disk in SPD mode is occupied and what it is used for
SHOW-DISK-STATUS	UNIT=...,INFORMATION=...	Queries information on specified disk(s)
UNLOCK-DISK	UNIT=...,SYSTEM-IDENTIFIER=...	Clears out the system allocation log

For a description of the commands, see the "Commands" manual [10].

5 Installation and configuration

HIPLEX MSCF incorporates the DSSM subsystems MSCF, NSM and XCS-TIME. This chapter deals with the installation of these subsystems and the parameterization of MSCF and NSM. The XCS-TIME subsystem only has to be installed (not parameterized). The chapter also includes generation instructions for the different types of network.

See also the [section “Summary: generation and operation of an MSCF network” on page 337](#).

General

The installation and parameterization of HIPLEX MSCF cannot be strictly separated from the [chapter “Administration of MSCF networks”](#) because firstly, attributes defined during generation influence the operation and secondly, some of the attributes defined during generation can be changed when the system is up and running.

HIPLEX MSCF V6.0 runs on BS2000/OSD V8.0. Furthermore, the software product JV is required to enable the full range of functions to be used.

Product maintenance in the network

Within an MSCF network the corrections to all network-relevant components must have the same release status (as well as HIPLEX MSCF, these include BS2000/OSD-BC, BCAM, DLMUSER, LNM and MSCFANC). This can also apply to add-on products which work throughout the network, e.g. JPPOPT or openSM2. It is therefore recommended that the correction statuses of all products be kept consistent.

Supported version link-ups

HIPLEX MSCF V6.0A on BS2000/OSD-BC V8.0 can work with the following HIPLEX MSCF versions:

- HIPLEX MSCF V6.0 on BS2000/OSD-BC V8.0
- HIPLEX MSCF V5.0 on BS2000/OSD-BC V7.0
- HIPLEX MSCF V4.0 on BS2000/OSD-BC V6.0 and V5.0

5.1 Installation of HIPLEX MSCF

HIPLEX MSCF must be installed using the installation monitor IMON. You must also read both the installation information in the delivery note and the release notes. Once installation is complete, HIPLEX MSCF is ready.

Details of the inputs necessary for installation with IMON and the installation procedure itself can be found in the “IMON” manual [7].

HIPLEX MSCF V6.0 is installed by installing the following DSSM subsystems MSCF V17.0A, NSM V17.0, XCS-TIME V17.0.

The version-dependent file suffix <ver> is “170” (for V17.0).

IMON implicitly imports the subsystem declarations for the subsystems of the relevant installation into the subsystem catalog of DSSM. If the MSCF subsystem is to be started automatically at startup, the subsystem catalog has to be adjusted accordingly.

5.2 Parameterization of MSCF

The subsystems of HIPLEX MSCF (MSCF, NSM and XCS-TIME) are controlled by DSSM. The MSCF subsystem uses BCAM as the basis for the communication. Furthermore, HIPLEX MSCF is executable on different hardware (processors) and different types of network can be implemented and superimposed. HIPLEX MSCF must therefore meet certain conditions for working with other products and also be parameterized for various types of use.

5.2.1 DSSM subsystem catalog

Since MSCF is managed as a subsystem of DSSM and controlled by the DSSM commands START-SUBSYSTEM and STOP-SUBSYSTEM, the subsystem declaration of MSCF (file SYSSSC.MSCF.<ver>) must be entered in the subsystem catalog of DSSM; it is imported implicitly during installation. This entry can be modified with the MODIFY-SUBSYSTEM-PARAMETER command in the dynamic subsystem catalog or with the MODIFY-SUBSYSTEM-ATTRIBUTES in the static subsystem catalog.

The two operands listed below permit modifications:

- An MSCF configuration file (see [page 115](#)) can be specified with SUBSYSTEM-INFO-FILE which is processed when HIPLEX MSCF is started. The default setting is file SYSPAR.MSCF.<ver>. The use of a configuration file is mandatory for the XCS network.
- The start time of HIPLEX MSCF can be defined with CREATION-TIME. The available choices are as follows:
 - CREATION-TIME = *AT-CREATION-REQUEST (default)
HIPLEX MSCF is started via the START-SUBSYSTEM MSCF command by starting the MSCF subsystem.
 - CREATION-TIME = *BEFORE-SYSTEM-READY
HIPLEX MSCF is started implicitly during the BS2000 system start. This presupposes that BCAM is also started during the BS2000 system start. This setting is recommended where a processor is to participate in an XCS network because global lock requests are rejected until the processor is integrated in an XCS network.

5.2.2 Processor identification

A processor is identified by the following three pieces of information:

- processor name for BCAM (for connection setup or clear-down)
- host name for applications
- system identification (SYSID) within certain applications.

Processor name

A processor name is a local representative of a partner processor. An address is assigned to the processor name by BCAM under which the partner processor is addressable in the network. Although this address is uniquely assigned to a processor, a processor can have more than one address (processor name).

Two processors can therefore know a third processor under different processor names (at the same point in time). Furthermore, a processor can also change its address in relation to a partner processor within a certain period of time. The processor name does not therefore guarantee the unambiguous identification of a processor (or more precisely a BS2000 session) within a network and over a period of time. This is why the host name exists.

Host name

The host name of a processor is the name of the own BS2000 system. It is specified in the DCSTART or DCOPT command when BCAM is started; it is not possible to start BCAM without assigning the host name.

During startup of HIPLEX MSCF, the communication system determines this name and uses it as the host name for that MSCF session.

The host names of the processors to be linked in the network must be unique. In other words, a processor cannot have MSCF connections to different partner processors at the same time with the same host name.

System identification (SYSID)

The SYSID of a processor is read out of the standard volume label (SVL) of the home pubset at BS2000 startup and is thus defined for the current BS2000 session. This means that a SYSID modified during a BS2000 session (via the SET-PUBSET-ATTRIBUTES command for the home pubset) does not become effective until the BS2000 is started up again with the same home pubset.

The following values are permissible for SYSIDs: A-Z, 0-9 and 65-192 (default value: 250).
A processor with SYSID 250 cannot participate in an MSCF network.

The SYSIDs of the processors to be linked must be unique, i.e. a processor cannot have MSCF connections to different partner processors at the same time with the same SYSID.

5.2.3 BCAM dependencies

For it to be possible to use HIPLEX MSCF, a network managed by BCAM must already exist.

The transport system BCAM is used by HIPLEX MSCF as the basis for the processing of elementary functions of the message transmission (see the "BCAM" manual [13]).

Protocols

HIPLEX MSCF supports all transport protocols known to BS2000/OSD-BC or BCAM from (ISO class 4 and NEA protocols and TCP/IP).

To avoid network problems, the RDF and SOF files should be cataloged on the home pubset. The same applies to APPLICATION-TABLE, SOCKETHOST-TABLE and PROCESSOR-TABLE.

Independent monitoring connections

If a partner is being monitored through two independent monitoring connections, a partner failure can be recognized automatically, even without a shared pubset. In that case a partner-specific setting of RECOVERY-START=*AUTOMATIC is additionally required.

The monitoring connections have to be configured as *physically independent* communication paths between the MSCF applications \$MRSAPP4 and \$MRSAPP5 on the local processor and on the partner using BCMAP commands. The communication path should preferably follow geographically different routes so that it is unlikely that they could both be destroyed through the same external event (e.g. construction work).

The specification of NUMBER-OF-CTRL-CONN=2 in the START-MSCF-CONNECTION or MODIFY-MSCF-CONNECTION command acknowledges that two physically independent monitoring connections were generated between the partners. However, MSCF cannot verify the specification, so making the wrong specification could lead to erroneous failure recognition and loss of data integrity.

Adjusted network generation

From the point of view of HIPLEX MSCF, the network generation is regarded as adjusted when the processor and host name of each processor in the network agree and the corresponding name is also used on the partner processor. If there are several routes between two processors, we recommend working with ALTERNATE-ROUTING (see the "BCAM" manual [13]) but not with different processor names (see also "Processor identification" on page 112).

Start time

If HIPLEX MSCF is started within the BS2000 startup procedure (CREATION-TIME=*BEFORE-SYSTEM-READY), BCAM must also be started within the BS2000 startup procedure. The latter can be initiated by the startup parameter service, e.g.

```
/BS2000 PARAMS
/BEGIN BCAM
DCSTART <dcstart parameter>, LWRES=NO
/EOF
/END PARAMS
```

Wait time for MSCF with the BCEND command

The BCAM command BCTIMES, MAX-MSCF-DELAY operand, enables you to set a delay time for terminating MSCF when the BCAM command BCEND is entered, see the “BCAM” manual [13]. The default value is 60 seconds.

You obtain information on the current value using the BCAM command SHOW-BCAM-ATTRIBUTES SELECT=*TIMER(*ALL / MAX-MSCF-DELAY)

The effects of BCEND on MSCF are described on [page 149](#).



When using HIPLEX MSCF in multiprocessor networks, please note that MSCF must be terminated before BCEND is issued.

You should also use MAX-MSCF-DELAY to define an interval that is long enough to ensure a structured shutdown of MSCF before BCEND. This value will depend on the size of the network.

5.2.4 MSCF configuration file

If the configuration file `SYSPAR.MSCF.<ver>` is used as delivered, only the MSCF subsystem is activated on the local processor. Connections to partner processors will not be set up automatically. They must be established explicitly for each of those processors with the `START-MSCF-CONNECTION` command.

By including the commands in the delivered configuration file or in another configuration file, these processes can be rendered automatic. The other configuration parameters can also be defined there. The configuration file is automatically processed when the MSCF subsystem is started.

- Specification

The name of the configuration file can be specified in the DSSM subsystem catalog entry by MSCF as `SUBSYSTEM-INFO-FILE` (`MODIFY-SUBSYSTEM-PARAMETERS` command for the dynamic catalog or `MODIFY-SUBSYSTEM-ATTRIBUTES` statement for the static catalog) or as a subsystem parameter when HIPLEX MSCF is started (`START-SUBSYSTEM` command). On delivery `SYSPAR.MSCF.<ver>` is specified in the static DSSM subsystem catalog.

The specification as a subsystem parameter has priority over the entry in the DSSM subsystem catalog.

- File name

The configuration file can be stored under any name on any user ID but always on the processor's home pubset. The file name must, however, be specified in full in the commands listed above.

If no user ID is specified, the specified file is searched for under the `$TSOS` ID and the `$defluid` ID (secondary read). If the configuration file is to be called with `SUBSYSTEM-PARAMETER='FILE-NAME=*SYSTEM` when the `START-SUBSYSTEM` command is initiated, it must be stored under the file name `SYS.MSCF-CONFIGURATION`.

- File attributes

The configuration file must be a SAM file with variable-length records (`FILE-STRUCT=SAM, REC-FORM=V`). It should have the attributes `USER-ACCESS=*OWNER-ONLY` and `ACCESS=*READ` and may be protected by a write password.

Configuration file format

The configuration file contains entries in command format.:

- The SET-MSCF-ENVIRONMENT command must be the first command in the configuration file (excepting remark lines). The following MSCF configuration parameters can be set with the command:

ABORT-LIMIT, *FADING-INTERVAL*, *HOST-PRIORITY*, *LEAVE-LIMIT*,
LOCAL-PASSWORD, *NOTIFY-BY-MAIL*, *XCS-NAME*, *TRACE-FILE*,
FAIL-DETECTION-LIMIT, *RECOVERY-START* (general value), *USER-TERM-LIMIT*,
NUMBER-OF-SERVERS and *SERVER-TASK-LIMIT*.

The configuration parameters presented in italics can be modified using the command MODIFY-MSCF-ENVIRONMENT.

- The START-MSCF-CONNECTION command is used to define the required MSCF connections. The partner-specific configuration parameters NUMBER-OF-CTRL-CONN and RECOVERY-START are also set with this command. They can be modified with the MODIFY-MSCF-CONNECTION command.
Up to 164 of these commands are permissible. A command may exist for the local processor but is, however, ignored. More than one entry for the same processor is not allowed.

Unambiguous abbreviations are permissible for the commands.

The commands are largely subject to the SDF rules with just the following exceptions:

- Command lines can only be continued after a comma. The continuation character (" - ") can be located in any position.
- Continuation lines must begin with "/".
- The continuation of comment lines is not permitted.
- The maximum permissible record length is 256 bytes (including record length field).

Configuration file in an XCS network

For participation in an XCS network, a configuration file must be set up. It must contain at least the SET-MSCF-ENVIRONMENT command with the name of the XCS network in which the processor is to participate.

- The configuration files of all processors participating in the same XCS network must agree in terms of the following entries:
 - In the SET-MSCF-ENVIRONMENT command, the values of XCS-NAME and FAIL-DETECTION-LIMIT must agree.
 - The configuration files of all the processors which are to participate in the same XCS network must contain the START-MSCF-CONNECTION command for a CCS connection (not necessarily with the same RECOVERY-START operands).
In addition we also recommend keeping the values of RECOVERY-START and USER-TERM-LIMIT the same in the SET-MSCF-ENVIRONMENT command.
- If no XCS name was specified when HIPLEX MSCF is started because:
 - the operand was not specified in the SET-MSCF-ENVIRONMENT command,
 - there is no configuration file,The processor is started in CCS mode. It is only possible to restart MSCF in XCS mode *during the current BS2000 session* if system parameter MCXSPXCS=V is set (see [page 120](#)).

Example of a configuration file

```

/REMARK "This is a configuration file for PROCNAM1."
/REMARK "Processors PROCNAM1, PROCNAM2 and PROCNAM3 are to"
/REMARK "form an XCS network with the name HUGO."
/REMARK "In addition, the development system PROCNAM4 is"
/REMARK "also to have access to the production processors."
/REMARK
/SET-MSCF-ENVIRONMENT XCS-NAME=HUGO,           -
/                      LOCAL-PASSWORD=C'OWN-PASS',-
/                      FAIL-DETECTION-LIMIT=308, -
/                      USER-TERM-LIMIT=240,     -
/                      LEAVE-LIMIT=600,         -
/                      HOST-PRIORITY=8,         -
/                      RECOVERY-START=*AUTOMATIC,-
/                      TRACE-FILE=*STD
/REMARK
/REMARK "Production mode"
/REMARK
/REMARK "Participant in XCS network HUGO"
/START-MSCF-CONNECTION PROCESSOR-NAME=PROCNAM1,-
/CONNECTION-TYPE=*CLOSELY-COUPLED(REMOTE-PASSWORD=C'CCS-PSW1',-
/                                RECOVERY-START=*AUTOMATIC)
/REMARK
/START-MSCF-CONNECTION PROCESSOR-NAME=PROCNAM2,-
/CONNECTION-TYPE=*CLOSELY-COUPLED(REMOTE-PASSWORD=C'CCS-PSW2')
/                                RECOVERY-START=*AUTOMATIC)
/REMARK
/START-MSCF-CONNECTION PROCESSOR-NAME=PROCNAM3,-
/CONNECTION-TYPE=*CLOSELY-COUPLED(REMOTE-PASSWORD=C'CCS-PSW3')
/                                RECOVERY-START=*AUTOMATIC)
/REMARK
/REMARK "Test and development processor"
/REMARK
/START-MSCF-CONNECTION PROCESSOR-NAME=PROCNAM4,-
/                                RECOVERY-START=*CONSISTENT-BY-OPERATOR

```

The configuration file can be copied to processors PROCNAM1, PROCNAM2, PROCNAM3 and PROCNAM4. If the processor passwords (OWN-PASS, CCS-PSW1, CCS-PSW2, CCS-PSW3) are not identical, the LOCAL-PASSWORD operand in the SET-MSCF-ENVIRONMENT command must be supplied correctly on each processor before it is used.

5.2.5 Global control parameters

The reactions of MSCF can be controlled with the parameters described below. Because several MSCF sessions (i.e. the time between START-SUBSYSTEM MSCF and STOP-SUBSYSTEM MSCF) can take place within a BS2000 session, a distinction is made between system parameters (defined in the startup parameter service) which apply to the overall BS2000 session and those parameters that purely apply to an MSCF session (defined via the SET-MSCF-ENVIRONMENT command).

System parameter MCXSPXCS

MCXSPXCS defines whether a processor is basically authorized to participate in an XCS network and how the XCS resources are managed.

- MCXSPXCS = N (default)
The processor cannot participate in the XCS network.
The XCS resources are locally managed and only the local processor can access the XCS resources.
Local resource management cannot be terminated. Therefore, the processor cannot participate in any XCS network for the duration of the entire BS2000 session. If an XCS name is specified in the MSCF configuration file, this causes the MSCF start to abort.
- MCXSPXCS = Y
The processor can take part in an XCS network.
XCS resources can be locally or globally managed. They can only be assigned after the MSCF subsystem is started. Whether the XCS resources are locally or globally managed depends on the value of the XCS-NAME operand of the SET-MSCF-CONNECTION command in the MSCF configuration file.
 - XCS-NAME=*NONE (default)
MSCF is started in CCS mode. The XCS resources are managed locally and only the local processor can access them. From this point on, the same procedure applies as for MCXSPXCS=N.
 - XCS-NAME=*SUSPEND
MSCF is started in CCS mode.
No XCS resources are allocated. However, the MSCF subsystem can also be started in XCS mode in the current BS2000 session in order to gain access to the XCS resources.
 - XCS-NAME=<alphanum-name 1..8>
MSCF is started in XCS mode.
XCS resources are managed globally. All XCS participants have access to the resources.
- MCXSPXCS=V
The processor is intended to participate in the XCS network.
XCS resources are managed globally only. They are only allocated if the MSCF subsystem is started in XCS mode. Within a BS2000 session it is possible to switch optionally between CCS mode and XCS mode.

MSCF configuration parameter ABORT-LIMIT

The MSCF configuration parameter ABORT-LIMIT defines the maximum duration of a network-participation abort for a shared pubset network or XCS network on the local processor. If the abort cannot be performed within the specified time, the BS2000 session on the processor is terminated. In this way the fail reconfiguration can be completed in the rest of the network and the operability of the network can be restored. The network-participation abort cannot be completed if, for example, the system cannot remove the DMS occupation of a pubset during export.

By default, abnormal system termination does not force termination of abort processing.

MSCF configuration parameter FADING-INTERVAL

The MSCF configuration parameter FADING-INTERVAL determines the wait time in seconds which the Live Monitor waits before fail reconfiguration is started. This period is used to wait for open user tasks on shared pubsets to be terminated. This enables user outputs from a failed system to be completed before reconfiguration starts.

MSCF configuration parameter HOST-PRIORITY

When reconfiguration is started automatically (see MSCF reconfiguration parameter RECOVERY-START=*AUTOMATIC / *SECURE on [page 124](#)), HIPLEX MSCF is automatically terminated abnormally on one of the two processors affected by the connection failure to allow full restoration of the operability of the processor network by means of a fail reconfiguration when a connection fails in the XCS network. With the HOST-PRIORITY configuration parameter you can classify a processor in the XCS network according to its importance and thus control the selection of which processor is to terminate when the connection fails. The HOST-PRIORITY determines the processor order according to their importance. Processors with a *lower value* of HOST-PRIORITY have the *higher priority*. The default value is 16. If the two processors affected by the connection failure have different HOST-PRIORITY values, the one with the higher value is removed from the XCS network. If both processors have the same HOST-PRIORITY value, the one that entered the network later is removed (higher value for JOINING ORDER). If several connections fail at the same time, this procedure is carried out for each one individually.

MSCF configuration parameter LEAVE-LIMIT

The MSCF configuration parameter LEAVE-LIMIT specifies the interval after which the coordinated exit of a processor from an XCS network (LEAVE) will be converted into a network-participation abort. The aim of this conversion is to defuse situations which delay or prevent completion of the exit and thus to make the rest of the XCS network operable again. LEAVE-LIMIT only starts after the user tasks have been terminated or after the USER-TERM-LIMIT ([page 128](#)) has expired.

By default a coordinated exit is not converted into an abort.

MSCF configuration parameter LOCAL-PASSWORD

Specification of the MSCF configuration parameter LOCAL-PASSWORD can protect the processor from illegal MSCF connections being generated by means of a password. The password defined for the processor must be specified when the START-MSCF-CONNECTION command is used for CCS outside the MSCF configuration file. The partner processor must also know the password, otherwise no (CCS) connection is established.

MSCF configuration parameter NOTIFY-BY-MAIL

The MSCF configuration parameter NOTIFY-BY-MAIL defines whether email notifications are to be sent to a user ID in critical situations, see [section “Email notification about critical states” on page 163](#). When an email is sent, the email address is taken from the EMAIL-ADDRESS field of the corresponding user entry.

MSCF configuration parameter XCS-NAME

Specification of an XCS name defines that the processor is to participate in the respective XCS network. The processor enters the network when HIPLEX MSCF is started.

The MSCF configuration parameter XCS-NAME is defined via the SET-MSCF-ENVIRONMENT command deposited in the MSCF configuration file. This configuration parameter cannot be modified once the XCS is up and running.

MSCF configuration parameter TRACE-FILE

Specifying the TRACE-FILE operand in the SET-MSCF-ENVIRONMENT command defines whether and - if so - to which file the traces of the MSCF subsystem are written. The MSCF traces represent an important document for the diagnosis of error situations. The file that is to be written to can be redefined or writing to a file can be suppressed by the MODIFY-MSCF-ENVIRONMENT command.

MSCF configuration parameter FAIL-DETECTION-LIMIT

The configuration parameter FAIL-DETECTION-LIMIT defines the time (in seconds) within which the processor must detect a failure in the MSCF network (such as a partner or connection failure). The defined value is rounded up to a multiple of 44; the default setting is 176 seconds (=minimum value).

The time for internal monitoring cycles is derived from this failure detection time (cycle time = rounded-up value of FAIL-DETECTION-LIMIT / 11).

Processor failures are detected by the monitoring mechanisms after 1-3 cycles. But this can be longer if the BCAM connections fail, depending on the maximum duration of a BCAM connection setup.

When defining the failure detection time, bear in mind that under certain conditions, the operating system is halted for long periods of time, see the [section “Disk monitoring blocks” on page 183](#). Monitoring cycles that are too short then result in errored failure detection. This must be avoided at all costs.

To avoid an erroneous failure detection or system termination, you must either specify `RECOVERY-START=*BY-OPERATOR / *CONSISTENT-BY-OPERATOR / *SECURE` or define a higher value for FAIL-DETECTION-LIMIT. Note that the processor cannot be suspended for longer than the result of FAIL-DETECTION-LIMIT / 11.

After an automatic failure detection, the system waits for a certain security interval before initiating the fail reconfiguration. The partner processor seen as crashing is thus given the opportunity to terminate if the failure detection was mistaken.

The value of FAIL-DETECTION-LIMIT should be selected to be less than the restart time (about 15 minutes) of a BS2000 system. The reason for this being that the failure of a processor must be detected on the partner processor and a fail reconfiguration carried out. Only when the fail reconfiguration is completed on the partner processor can the (crashed) processor enter its new BS2000 session in a CCS, shared pubset or XCS network.

If the processor is restarted immediately after its crash and if it wants to enter the network before the fail reconfiguration is finished on the partner processor, the host name of the processor will (still) be known to the partner processor. The processor is therefore refused entry to the network (messages MCS1005 and MCS0009).



A CCS connection can only be generated between two processors if they have the same FAIL-DETECTION-LIMIT.

MSCF configuration parameter RECOVERY-START

The behavior of partner monitoring can be influenced in a variety of ways using the RECOVERY-START configuration parameter:

Activating and deactivating MSCF partner monitoring

Partner can be monitored by MSCF through connection monitoring or disk monitoring. A partner is monitored if monitoring via at least one of the two monitoring paths is activated (for exceptions see below):

- Connection monitoring: generally activated when a CCS connection is generated and deactivated when the CCS connection is closed down (not when it fails).
- Disk monitoring: activated when a common shared pubset is imported and deactivated when the last common shared pubset is exported.

XCS partners are always monitored. Monitoring an XCS partner can only be deactivated by removing the partner from the XCS network.

A CCS partner which is not a shared pubset partner is monitored when a CCS connection is generated to it using RECOVERY-START \neq *STD. As long as no shared pubset is imported with the partner monitoring can be deactivated by modifying RECOVERY-START to $=$ *STD or. \neq *STD

A shared pubset partner is only monitored while the CCS connection is established. Connection monitoring and disk monitoring are both activated together when the CCS connection to the shared pubset partner is initially generated. Similarly, connection monitoring and disk monitoring are also both activated together when a shared pubset which is shared with the local processor is imported for the first time for the CCS partner with RECOVERY-START= $=$ *STD. In order to deactivate monitoring, both monitoring paths must be closed down by terminating the CCS connection and exporting all shared pubsets.

Partner monitoring by the Live Monitor of HIPLEX AF

Partners can also be monitored using the Live Monitor, see [page 71](#).

Inhibiting the automatic start of fail reconfiguration

When partner monitoring or the Live Monitor detects that a partner has failed, it is necessary to perform fail reconfiguration for the failed processor to restore network functionality. The setting of the configuration parameter RECOVERY-START defines whether this fail reconfiguration is started automatically or whether MSCF asks the question MSC1100 (No activity of partner detectable. Start fail-reconfiguration?) on the console and fail reconfiguration begins only after this question has been answered.

Independent of the RECOVERY-START setting, this query will also be issued if an MSCF configuration parameter has just modified RECOVERY-START.

Inhibiting an automatic start of fail recovery can be set with network-wide or partner-specific effect:

- network-wide setting:
The network-wide setting applies for all partners. It is implemented using the SET-MSCF-ENVIRONMENT or the MODIFY-MSCF-ENVIRONMENT command.
- partner-specific setting:
The partner-specific setting only applies to a specific partner. It is implemented using the START- or MODIFY-MSCF-CONNECTION command.

The partner-specific setting RECOVERY-START=*STD means that the general setting should become effective for this partner. If the partner-specific setting is neither *STD nor the same as the general setting, the value of RECOVERY-START which is effective for the partner is derived from the table below:

First setting ¹	Second setting ¹	Locally effective setting
*AUTOMATIC	*BY-OPERATOR	*BY-OPERATOR
*AUTOMATIC	*CONS-BY-OPERATOR	*CONS-BY-OPERATOR
*AUTOMATIC	*SECURE	*SECURE
*BY-OPERATOR	*CONS-BY-OPERATOR	*CONS-BY-OPERATOR
*BY-OPERATOR	*SECURE	*BY-OPERATOR
*CONS-BY-OPERATOR	*SECURE	*CONS-BY-OPERATOR

¹ The combination of a general and a partner-specific setting always results in the same locally effective setting.

If the setting which is effective locally for a partner is RECOVERY-START=*BY-OPERATOR / *CONSISTENT-BY-OPERATOR, fail reconfiguration may be started only after the question MCS1100 has been answered; automatic start is forbidden. In this case the RECOVERY-START setting on the partner is meaningless.

If the setting which is effective locally for a partner is `RECOVERY-START=*SECURE` and the Live Monitor reports or confirms the failure of this partner, MSCF starts fail reconfiguration automatically. In this case the `RECOVERY-START` setting on the partner is meaningless.

If the setting which is effective locally for a partner is `RECOVERY-START=*AUTOMATIC`, the partner will take the decision regarding the start of fail reconfiguration:

- If the setting which is effective on the partner for the partner's own computer was `RECOVERY-START=*CONSISTENT-BY-OPERATOR / *SECURE`, fail reconfiguration may be started only after the question MCS1100 has been answered.
- If the setting which is effective on the partner for the partner's own computer was `RECOVERY-START=*AUTOMATIC / *BY-OPERATOR`, automatic start of fail reconfiguration is required. However, partner monitoring cannot be certain of the failure of the partner in the following cases:
 - Failure of the last monitoring path if this does not fail “simultaneously” with another monitoring path
 - Failure of the partner during a SNAPSHOT dump
 - Failure of the partner while the Cluster Recovery Lock (CRL, see [page 183](#)) is set

If one of these cases has occurred, MSCF also poses the question MCS1100, and fail reconfiguration begins only after the question has been answered.

Error recovery when a connection fails in XCS mode

If the MSCF connection between two processors in an XCS network fails, the network functionality is disrupted.

Measures are automatically initiated to eliminate the failure if *AUTOMATIC / *SECURE is specified for the general RECOVERY-START setting and *AUTOMATIC / *SECURE or *STD is specified for the relevant partner-specific setting. In this case the rules described under “MSCF configuration parameter HOST-PRIORITY” (see [page 121](#)) apply and HIPLEX MSCF is aborted on one of the two processors and removed from the XCS network configuration.

Otherwise, the message MCS1101 prompts systems support to eliminate the connection failure.

Controlling allowed system aborts

See also [section “Abnormal termination of HIPLEX MSCF” on page 184](#).

- System termination with MSC1300 (to avoid erroneous failure detection through the partner) can either be generally or partner-specifically prohibited. A processor’s general or partner-specific setting *BY-OPERATOR stops the system abort due to erroneous failure detection by the local processor on all partners or on a specific partner. The *CONSISTENT-BY-OPERATOR setting also stops the system abort on the local processor due to erroneous failure detection by an optional or specific partner.
- The MCS1301 system abort (to allow reconfiguration on the MSCF partners) is only executed if a processor’s general RECOVERY-START setting is *AUTOMATIC / *SECURE.

MSCF configuration parameter USER-TERM-LIMIT

This configuration parameter defines the time available to the user tasks for termination, counting from the moment the XCS termination is initiated. When this time elapses, the registered functions are terminated.

If, in rare situations, the XCS functionality cannot be completely cleared locally, the MSCF subsystem is not unloaded until the system is terminated.

MSCF configuration parameter NUMBER-OF-SERVERS

This configuration parameter determines the number of server tasks that are to be provided on the processor. The default value is four server tasks, but up to ten and as few as two can be requested. The number can increase continually during the MSCF session. If necessary, MSCF will generate new server tasks, but it will also terminate them again when they are no longer needed.

The number of server tasks defined by the START-SUBSYSTEM command has priority over the number defined in the SET-MSCF-ENVIRONMENT command.

MSCF configuration parameter SERVER-TASK-LIMIT

Too heavy a load due to too many MSCF server tasks can bring a system to a standstill (e.g. due to PAGING SATURATION). To prevent this happening, the number of MSCF server tasks is limited. The value of the MSCF configuration parameter SERVER-TASK-LIMIT ("MAXIMUM" in the SM2 output under "TASK LIMITS") serves as a yardstick.

An MSCF server task is in "occupied" state when it is in a deadlock-threatened situation that affects all processors. The system components using the MSCF server tasks decide when this applies.

If the number of "occupied" MSCF server tasks exceeds an upper limit (= 75% of SERVER-TASK-LIMIT; "FLOW SET" in the SM2 output under TASK LIMITS), the status of the local processor is set to "FLOW". Partner processors are informed accordingly and respond by then sending only absolutely essential messages to the processor (which resolve the "occupied" state of an MSCF server task). Once the number of "occupied" MSCF server tasks has decreased again to a lower limit (= 50% of SERVER-TASK-LIMIT; "FLOW RESET" in the SM2 output under "TASK LIMITS"), the "FLOW" status is cancelled. The partner processors are informed about the changed situation and after that resume sending messages of any type to the processor.

Since the limit is based on the number of "occupied" MSCF server tasks, at times of sudden peak loads, more server tasks may be generated than specified in the configuration parameter SERVER-TASK-LIMIT.

5.3 Generation of the LCS network

As regards BCAM, the dependencies described in [section “BCAM dependencies” on page 113](#), (MAXTSDU, protocols) must be taken into account.

The processor name must be unique.

We recommend generating an MSCF configuration file (see [page 115](#)).

Decentralized administration must ensure that the processors in an LCS network have the following attributes:

- unique SYSIDs
- unique catalog IDs of the home pubset
- same value for the system parameter ENCRYPT
If password coding is required, *all* of the processors in the network must work with it.
- Both the user ID and the LOGON password must be the same on the local processor and on the LCS partner. If that is not the case, the LCS functionality will not be available. This also applies to the \$TSOS user ID.

The local security criteria of a processor cannot be bypassed by HIPLEX MSCF.

5.4 Generation of the CCS network

As regards BCAM, the dependencies described in [section “BCAM dependencies” on page 113](#), (MAXTSDU, protocols) must be taken into account. BCMAP commands for \$MRSAPP and \$MRSAPPS are recommended. The host names assigned on different processors by the DCOPT or DCSTART commands must be unique and should be the processor names.

We recommend generating an MSCF configuration file for the CCS network (see [page 115](#)). The configuration file can be entered in the DSSM subsystem catalog as SUBSYSTEM-INFO-FILE.

If HIPLEX MSCF is to be started during the BS2000 startup process, the start time must be set to BEFORE-SYSTEM-READY and BCAM must also be started during the BS2000 startup.

If CCS connections are to be established automatically when HIPLEX MSCF is started (see [page 115](#)) or if the processor is to be protected in the network by a password, an MSCF configuration file is mandatory (see [page 122](#)).

A decentralized administration function must ensure that the processors in a CCS network have the following attributes:

- unique SYSIDs
- unique catalog IDs of the home pubsets
- same value for the system parameter ENCRYPT
If password coding is required, *all* the processors in the network must work with it.
- mutually adjusted SRPM databases
User jobs are processed on the remote side by MSCF server tasks. The server tasks "mask" themselves for this purpose with the ID and privileges of the local job submitter.
- same values for FAIL-DETECTION-LIMIT.

5.5 Generation of the shared pubset network

Because the shared pubset (SPVS) network is based on the CCS network (at least one CCS network between the master processor and each slave processor), the preconditions listed in section [section “Generation of the CCS network” on page 130](#) also apply to the shared pubset network. Furthermore, the CATIDs and the VSNS of the shared pubsets on all processors in the shared pubset network must be unique.

5.5.1 Entering the shared pubsets in the MRSCAT

The MRSCAT contains entries for the pubsets defined on the processor. A shared pubset that is to be imported to the processor must first be entered in the MRSCAT with the `ADD-MASTER-CATALOG-ENTRY` or `MODIFY-MASTER-CATALOG-ENTRY` command:

```
/ADD-MASTER-CATALOG-ENTRY ENTRY-NAME=<catid>, -
/                               SHARED-PUBSET=*YES, -
/                               DIALOG-WAIT-TIME=<time1>, -
/                               BATCH-WAIT-TIME=<time2>
```

or

```
/MODIFY-MASTER-CATALOG-ENTRY ENTRY-NAME=<catid>, -
/                               SHARED-PUBSET=*YES, -
/                               DIALOG-WAIT-TIME=<time1>, -
/                               BATCH-WAIT-TIME=<time2>
```

The time values `<time1>` and `<time2>` specify for batch and dialog tasks how long the system should wait for the (re)establishment of the connection in the event that it is lost. For compatibility reasons, the `DIALOG-WAIT-TIME` and `BATCH-WAIT-TIME` operands have the same default values as in the previous versions. However, we recommend using the value defined for `FAIL-DETECTION-LIMIT` (`SET-MSCF-ENVIRONMENT` command) for `DIALOG-WAIT-TIME`.

5.5.2 Creating a pubset as a shared pubset

A pubset is declared a potential shared pubset with the SET-PUBSET-ATTRIBUTES command. This command is also used to predefine the master processor of the pubset and the attributes for a master change. The following setting is recommended:

```
/SET-PUBSET-ATTRIBUTES PUBSET=<catid>, -
/ SHARED=*YES, -
/ MASTER=<sysid1>, -
/ BACKUP-MASTER=<sysid2>, -
/ ALTERNATE-BACKUP=*BY-SHARER
```

If MASTER-CATALOG-ENTRY (see [section “Entering the shared pubsets in the MRSCAT” on page 131](#)) contains SHARED-PUBSET=*YES the following applies to the *next* pubset session:

When the IMPORT-PUBSET command is used *without* specification of the SHARER-TYPE operand on the processor with <sysid1>, this starts a master import, i.e. the shared pubset session begins and the processor with <sysid1> becomes the master processor of the shared pubset (<catid>). During a master change (see [page 172](#)), one of the two processors with <sysid2> and <sysid1> is a backup master while the other is the master, and vice versa. If a master change is necessary at a time when the current backup master is not active, a third processor that is currently an active slave, is implicitly named the backup master and the master change is initiated.

The SET-PUBSET-ATTRIBUTES command causes entries to be made in the SVL of the pubres of a pubset. The specification SHARED=*YES identifies a pubset (in its SVL) as a potential shared pubset. Whether a pubset that is thus identified is actually used as a shared pubset is decided when the pubset is imported and depends on whether local multiple usability is entered in the MRSCAT, or if the pubset is entered as multiply usable in the MRSCAT, by the specifications in the IMPORT-PUBSET command, USE and SHARER-TYPE operands.

5.5.3 Parameters for defining the shared pubset master

The assignment of the master attribute when a network is set up and to another specific processor when the previous master crashes can be predefined via the SET-PUBSET-ATTRIBUTES command. The specifications are stored in the SVL of the PUBRES of the SF pubsets or the VOLRES of the SM pubsets. These specifications determine how the sharer will react when the pubset is imported, when the current master processor crashes and when a pubsets is exported during a master change (EXPORT-PUBSET command with MASTER-CHANGE=*YES).

The following specifications are possible for the master attribute:

- *MASTER*
The MASTER operand defines which processor is to be the master processor when the pubset is imported, in as far as the SHARER-TYPE operand is not specified in the IMPORT-PUBSET command.
The SHOW-PUBSET-ATTRIBUTES command displays this value under "DESIGNATED MASTER" and the current master processor under "CURRENT MASTER".
- *BACKUP-MASTER*
The BACKUP-MASTER operand can be used to define a processor which is to automatically assume the role of master after a master change.
- *ALTERNATE-BACKUP*
The ALTERNATE-BACKUP operand defines a method for selecting the backup master if at the time of the master change, no backup master can be defined or the processor designated as the backup master is not active.

Selection of actual master processor

When the IMPORT-PUBSET command is issued, the master processor is selected, after normal termination of the previous pubset session, as follows:

1. If a "current master" is already entered in the SVL, that specification applies.
2. If SHARER-TYPE=*MASTER is specified in the IMPORT-PUBSET command, the imported processor becomes the master processor.
3. If a "designated master" is specified in the SVL, this becomes the master processor.
4. If none of conditions 1 to 3 is met, the imported processor becomes master processor (i.e. the processor on which the IMPORT-PUBSET command is first issued becomes the master processor).

Selection of the actual backup master

At the time of a master change, the actual backup master is selected as follows:

1. If a "backup master" (SET-PUBSET-ATTRIBUTES command, BACKUP-MASTER operand) is entered in the SVL and if that processor is an active slave processor, it becomes the backup master.
2. If the "backup master" entered in the SVL is the current master, the MASTER (= designated master) entered in the SVL is the backup master
3. If no "backup master" is entered in the SVL or if that processor is not active, the SVL ALTERNATE-BACKUP entry is evaluated:
 - ALTERNATE-BACKUP=*NONE
There is no alternative backup master, no master change takes place.
 - ALTERNATE-BACKUP=*BY-SHARER
The processor that is the first active sharer in the current SVL sharer list becomes backup master.
 - ALTERNATE-BACKUP=*BY-OPERATOR
No backup master is defined; a master change is not initiated automatically but must be started by systems support by means of an import with master change (see [page 154](#)).

After a successful master change, the backup master is the new master processor (i.e. current master), otherwise it remains a slave processor.

The master change is rejected there is no CCS connection between the backup master and a slave processor (it has not been established or it has been interrupted).

If an automatic master change is not initiated or fails, the pubset is put into INACC,QUIET status.

If a master change is not explicitly prohibited (by the specification BACKUP-MASTER=*NONE and ALTERNATE-BACKUP=*NONE), or if a master change is not initiated or fails, it can be initiated subsequently by the commands EXPORT-PUBSET or IMPORT-PUBSET with MASTER-CHANGE=*YES (see [chapter "Administration of MSCF networks" on page 143](#)).

5.5.4 General information about shared pubset mode

Shared pubset mode requires the following conditions to be met:

- The pubset must be on devices to which the processors have direct hardware paths.
- The configuration must be a balanced one.
A low-performance master with possibly several powerful slaves which furthermore generate a high DMS management load (e.g. Open/Close), can only be regarded as an incorrectly configured shared pubset network. For performance reasons, the processor expected to handle the high DMS management load should, if possible, be defined as the master processor.
- Number of MSCF server tasks
A high DMS management load can result in the multiplication of MSCF server tasks. To prevent the number of server tasks escalating out of hand, a maximum value must be defined. When this limit is reached, the DMB management load is automatically restricted by the system, a fact that becomes noticeable by the significantly poorer performance. Powerful configurations permit a higher limit for the number of server tasks. This is set in the MSCF configuration file with the SET-MSCF-ENVIRONMENT command, SERVER-TASK-LIMIT operand. When deciding on the maximum number of server tasks (see [“MSCF configuration parameter SERVER-TASK-LIMIT” on page 128](#)), a decisive factor is the number of tasks or system tasks permitted in the overall system.
- Reliability of disk monitoring
The permissible disks (with respect to disk inputs/outputs) should always be used as shared pubsets.
A separate disk log is kept for each shared pubset irrespective of whether the shared pubset is an SF or an SM pubset. Each shared pubset is thus a separate monitoring path, a fact that makes the disk monitoring function highly reliable.

The SM pubset concept permits all the shared pubsets to be grouped into a single shared pubset. This solution, however, runs contrary to the requirement for a reliable, failsafe disk monitoring function. We recommend using *at least* two physically separate shared pubsets (i.e. separate channel paths, disk controllers etc.).
- For all processors that access a shared pubset we recommend setting identical storage space saturation levels (in the MRSCAT) because otherwise a master change can result in problems.
- System parameter TEMPFILE
If the processor is to work with temporary files (TEMPFILES) or job variables, then all of the processors in the network must work with them. If they work with TEMPFILES, the same prefixes must be defined.

- System parameter FSHARING
This access protection for pubsets must agree on all the processors that access a shared pubset (see the the “Introductory Guide to Systems Support [5]).
- SYSEAM on shared pubsets
Whether SYSEAM files on shared pubsets are supported depends on the setting of the system parameter EAMSPVS:
setting "00": SYSEAM files on shared pubsets can only be created by the master processor of the pubset
setting "01": SYSEAM files can be created on all shared pubsets.
- VM mode with shared pubsets
If a shared pubset is imported to guest systems of the same hypervisor, the real volumes of the shared pubset (not emulated in the shared GS) must be defined in the VM as "shared". This means that each input/output on these disks causes two hypervisor interruptions. When there is a high input/output load, the performance of the entire VM processor can be seriously affected.
- System files on shared pubsets
Message files, SDF syntax files, SM2 monitoring files and POSIX file systems (except for /root and /var) can be stored on shared pubsets.
Other files that are occupied by system tasks (e.g. accounting files) should not be placed on a shared pubset because they could prevent the export of a shared pubset and thus the reconfiguration of the shared pubset network.

5.5.5 Special operating modes

Mode without master change

If the command SET-PUBSET-ATTRIBUTES with BACKUP-MASTER=*NONE, ALTERNATE-BACKUP=*NONE is executed, a master change *is fundamentally precluded* (irrespective of the MASTER operand).

Targeted master change between two processors

The command SET-PUBSET-ATTRIBUTES with MASTER=<sysid1>,BACKUP-MASTER=<sysid2>,ALTERNATE-BACKUP=*NONE (and corresponding handling of the IMPORT-PUBSET command) can be used to ensure that, without explicit actions, only the processors with <sysid1> and <sysid2> can become master processors of a shared pubset. If sysid1 is the master processor and sysid2 the active slave processor of the shared pubset at the time of a master change, <sysid2> is the backup master and becomes the new master as a result of the master change. If conversely <sysid2> is the master processor and <sysid1> the active slave processor, <sysid1> is the backup master which becomes the new master as a result of a master change if <sysid2> crashes. In other words, the role of master is assumed alternately by the two processors. If one of the two processors is the master processor but the other is not an active slave processor, no master change takes place. The specification of MASTER-CHANGE=*YES in the commands EXPORT-PUBSET and IMPORT-PUBSET ensures that in emergency situations, a third processor can assume the master role.

Operation with relevant and non-relevant partners

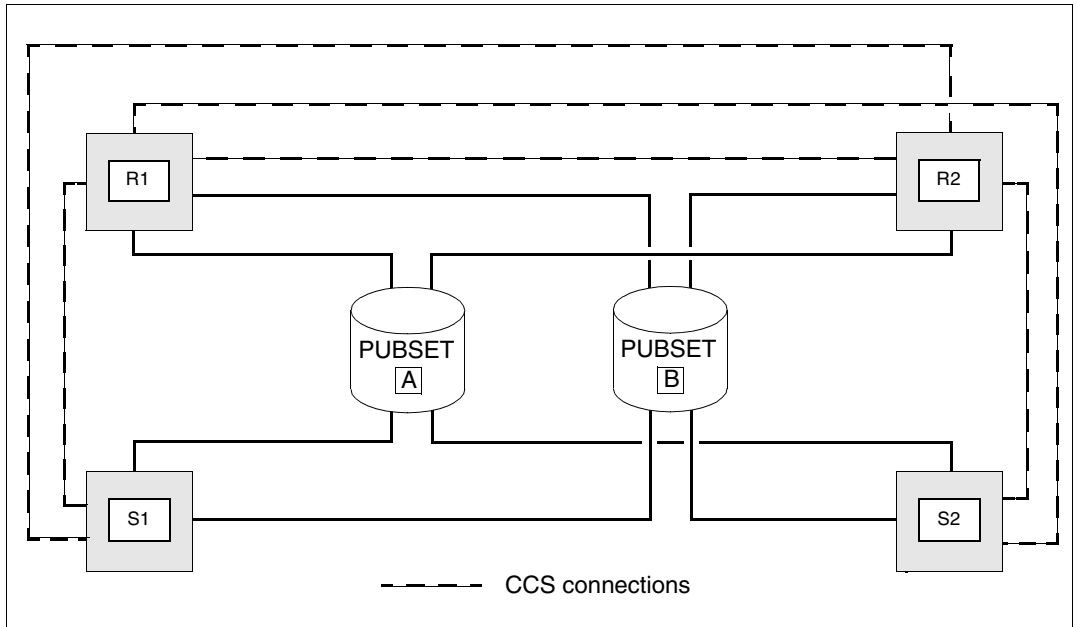
The partner monitoring function in a shared pubset network is partner-based (see [page 32](#)). The partners are divided into two categories: "monitored partners" and "non-monitored partners".

Only those shared pubset partners are monitored to which there is a CCS connection or to which there was a CCS connection at any given time within the current MSCF session. The other partner processors are not monitored (see also section "[Activating and deactivating MSCF partner monitoring](#)" on [page 124](#)).

There has to be a CCS connection between processors in a master/slave relationship, i.e. the processors monitor each other. There should also be a CCS connection between processors in a backup master/slave relationship, since a master change would otherwise not be possible.

No CCS connection is required between processors that are in a slave/slave relationship regarding all shared pubsets that they use together. The processors do not monitor each other and can therefore be operated independently of one another, i.e. in a decentralized mode. The following example describes a configuration in which two centrally operated processors alternately take on the role of master and backup master processor and in which other, decentrally operated processors are always in the role of slave processors of the shared pubsets.

Example



Processors R1 and R2 on which the production is running, are operated centrally. They are alternately master processor and backup master for shared pubsets A and B. The centrally operated processors S1 and S2 are departmental processors or test systems; from time to time these processors have to access central data on shared pubsets A and B. All four processors form shared pubset networks in relation to the two shared pubsets. For a successful master change to be possible, CCS connections must exist between processor R1 and all the other processors and between processor R2 and all the other processors but not between processors S1 and S2. In our example there is no CCS connection between processors S1 and S2, i.e. they do not monitor each other. A failure of processor S2 would not affect S1. In particular, the shared pubset functionality with R1 and R2 remains intact.

The situation is a different one if there is a CCS connection between S1 and S2. Then the two processors monitor each other and S2 has to react to the failure of S1 by starting a fail reconfiguration.

If the fail reconfiguration is not started, e.g. because MCS1100 is not answered, then S2 cannot import pubsets A and B after it has been restarted. But if the fail reconfiguration is performed and S2 it subsequently transpires that S2 had not failed at all, MSCF will abort on S1. Without partner monitoring between S1 and S2 this problem cannot occur.

5.6 Generation of the XCS network

Since a CCS connection must exist between every two XCS participants, the preconditions listed in [section “Generation of the CCS network” on page 130](#) also apply to the XCS network.

The system parameter MCXSPXCS must be set to "Y" or "V" on all the XCS participants. This can be done via the startup parameter service.

In order to participate in an XCS network the name of the network must be specified in the MSCF configuration file in the SET-MSCF-ENVIRONMENT command via the XCS-NAME operand.

The MSCF configuration files of all the XCS participants must agree in the following points:

- The definitions made for the XCS-NAME and FAIL-DETECTION-LIMIT operands in the SET-MSCF-ENVIRONMENT command must be the same on all of the processors in the network. We recommend setting the USER-TERM-LIMIT operand to be in agreement on all of the XCS participants. The possible definitions for the FAIL-DETECTION-LIMIT operand are described in detail on [page 123](#).
- The configuration files of all processors in the XCS network must contain a START-MSCF-CONNECTION command for each processor in the XCS network (the operands do not necessarily have to agree for participants).

At least one XCS pubset must be available in the XCS network. Since an XCS pubset is a special shared pubset, the preconditions required of a shared pubset network must also be observed for the XCS network (see [section “Generation of the shared pubset network” on page 131](#)).

When entering the XCS pubsets in the MRSCAT, XCS-CONFIGURATION=*YES must also be specified in the ADD-MASTER-CATALOG-ENTRY or MODIFY-MASTER-CATALOG-ENTRY command, and ALTERNATE-BACKUP must be set to *BY-SHARER with the SET-PUBSET-ATTRIBUTES command.

The processor that enters the XCS network first determines which pubsets are actually used as XCS pubsets. When another processor enters the XCS network, the only pubsets automatically imported to it as XCS pubsets are those which the first XCS participant defined. Pubsets that are defined as XCS pubsets on other XCS participants but not on the first processor, cannot be accessed until they have been explicitly imported. The pubsets intended for use as XCS pubsets should therefore be the same on all of the processors of an XCS network.

5.7 Parameterization of NSM

The NSM subsystem only becomes active in an XCS network. NSM supports options for setting delay times for the communication token in the network. Within a parallel HIPLEX (i.e. where the global storage (GS) is shared by all processors in the network), NSM supports the global storage as a supplementary, fast communication medium in addition to MSCF.

5.7.1 Configuration file

The subsystem information file SYSSSI.NSM.<ver> entered in the DSSM subsystem catalog serves as a configuration file for NSM (provided no other name was specified during installation). The file must be a SAM file with variable record length (FILE-STRUC=SAM,RECFORM=V) and should be protected against unauthorized modification with USER-ACCESS=*OWNER-ONLY and, if appropriate, ACCESS=*READ or a write password.

The only command allowed in the configuration file is SET-NSM-ENVIRONMENT (see [page 222](#)). This specifies whether the global storage is to be used as a fast communication medium. Only users with a very good knowledge of the system should use the other setting options for the circulation times of the communication token in the network.

In the information file generated during installation, the command is specified such that NSM attempts to use any existing global storage as a fast communication medium. But all settings can also be changed later with the MODIFY-NSM-ENVIRONMENT command.

The settings defined in the information file apply to the first processor that enters the corresponding XCS network (and thus establishes it). All other processors entering the XCS network take over the settings in effect at the time they enter (see the /MODIFY-NSM-ENVIRONMENT command on [page 212](#)).

5.7.2 GS partition

To be able to use the global storage as a communication medium, NSM requires a currently 3-Mbyte GS partition with the ID 'NSM'. If no partition of this name exists yet, NSM attempts to create one if possible with MODE=*DUAL.

If NSM is to work with a MONO partition despite an existing DUAL GS, the MONO partition must be created explicitly using the following command before MSCF is started:

```
/CREATE-GS-PARTITION PARTITION-ID=NSM, SIZE=3, MODE=*MONO(...)
```

6 Administration of MSCF networks

This chapter provides an overview of all supported administration commands and contains information on starting and stopping HIPLEX MSCF as well as on connection setup and teardown. It also describes MSCF connections, the setup and teardown of LCS, CCS, shared pubset, and XCS networks, the retrieval of information, and the handling of XCS pubsets.

Overview of administration commands

Processors are linked by a data communication network to form a function network. They can be connected to or disconnected from the network dynamically. The administration commands required to administer the DSSM subsystem MSCF and the MSCF network are summarized below.

Command	Function	User
MODIFY-MSCF-CONNECTION	Modify partner-specific monitoring parameters	System support
MODIFY-MSCF-ENVIRONMENT	Modify MSCF configuration parameters	Systems support
SET-XCS-PUBSET	Declare a shared pubset as an XCS pubset	Systems support
SHOW-MSCF-CONFIGURATION	Request information on the HIPLEX MSCF configuration	Systems support, user
SHOW-XCS-OCCUPATION	Output information on user tasks that occupy XCS functionality	Systems support
START-MSCF-CONNECTION	Establish an MSCF connection to a partner processor	Systems support
START-SUBSYSTEM MSCF	Activate HIPLEX MSCF	Systems support
STOP-MSCF-CONNECTION	Close down MSCF connection to a partner processor	Systems support
STOP-SUBSYSTEM MSCF	Deactivate HIPLEX MSCF	Systems support,

6.1 Starting HIPLEX MSCF

HIPLEX MSCF is started separately on each processor, either automatically when BS2000 starts up or manually via the START-SUBSYSTEM MSCF command. Starting HIPLEX MSCF establishes the respective processor as a sharer in the HIPLEX MSCF communication network and a sharer in the MSCF network.

The communication system initializes the BCAM applications, creates the server tasks, and initializes configuration management. The registered functions required for entering the network, which create their network entities on the processor, are in turn initialized by configuration management. This concludes the "local initialization", and message MCS0011 is output.

Network entry occurs as soon as the local initialization process is completed successfully. Of the total of four possible network types, LCS and particularly CCS form the basis for communication between the processors connected in the MSCF network. The shared pubset and XCS network types depend on the existence of the CCS network type. If HIPLEX MSCF is started with the delivered configuration file or without a configuration file, the local processor remains isolated and does not join any network. But if the configuration file contains START-MSCF-CONNECTION commands, the local processor enters into the relevant networks if the partners have already expressed similar connection requests to the local processor. Once a CCS connection exists a shared pubset can be imported in order to establish a shared pubset network with the relevant partner.

If the configuration file permits the processor to participate in an XCS network, the processor joins the network by establishing the CCS connection to an already active partner of the same XCS network. At the same time a shared pubset network is established or extended with the XCS partner processor because of the mandatory XCS pubset. If no active XCS partner processor is available as yet, the local processor starts the XCS network (in which case it is the sole member of the XCS network). An MSCF connection to partner processors that have not yet expressed a connection request to the local processor is postponed until the partner processors actually request a connection.

So, depending on the contents of the configuration file, the processor can join either one, several, all or none of the networks of the available network types when the START-SUBSYSTEM MSCF command is executed. The command itself is always the same. Independent of the contents of the configuration file, the local processor can join further LCS, CCS, or shared pubset (SPVS) networks by entering the relevant START-MSCF-CONNECTION or IMPORT-PUBSET command during operation. Joining an existing XCS network, however, is only possible by including the XCS name and the relevant START-MSCF-CONNECTION command in the MSCF configuration file.

At startup HIPLEX attempts to establish MSCF connections to the processors named in the configuration file (see [section “Establishing connections” on page 147](#) and [section “MSCF configuration file” on page 115](#)). If the processor is to join an XCS network, the registered XCS and CCS functions are initialized and after the connection establishment attempts have been concluded, entry of the processor into the XCS network is initiated. If participation in an XCS network is not provided for, only CCS functions are started. From this point on the registered CPM function allows shared pubsets to be imported. A successful network entry concludes the start of HIPLEX MSCF on the processor.

Coordinating the start of BCAM and MSCF

When MSCF is started, MSCF connections can only be established (initiated by the relevant commands in the parameter file) once BCAM startup is successfully completed and the routes required for the MSCF connections are activated.

As well as BCAM commands you can also enter BS2000 commands in the SOF file. To make sure that the routes required by MSCF are available, we recommend setting up the SOF file as follows:

1. Activate the routes required by MSCF
2. Start the MSCF subsystem
3. Other BCAM and/or BS2000 commands.

To start MSCF from the BCAM-SOF file, SET-CODE CODE=R,CONSOLE=BCAM must be specified in the OPR parameter service.

Example

```

/BCIN <route 1>
/BCIN <route 2>
.
.
/BCIN <route n>
START-SUBSYSTEM MSCF
.
.

```

} Activate the routes required by MSCF

} Start MSCF

} Other BCAM or BS2000 commands

Notes

- The start of HIPLEX MSCF on a processor with SYSID 250 is rejected with message MCS0007 (error code: 01).
- Message MCS0029 indicates which configuration file was used when starting HIPLEX MSCF.
- Message MCS0030 indicates which MSCF trace file was opened.
- When message MCS0011 is output, a connection setup to partner processors is initiated but not yet concluded.
- It is not possible to import shared pubsets before the start of HIPLEX MSCF. The import job is either delayed (slave import) or rejected (master import).
- If MSCF is to be started within the BS2000 startup procedure (see [page 114](#)), BCAM and MSCF cannot be coordinated from the SOF file with the START-SUBSYSTEM MSCF command (in this system state only DSSM can start a subsystem). However, if the BCAM routes needed by MSCF are activated at the beginning of the SOF file, once MSCF is loaded the MSCF connections specified in the configuration file are established in the BS2000 startup.

6.2 Establishing connections

The basic requirements for a successful MSCF connection setup are described in [section “BCAM dependencies” on page 113](#).

The connections to partner processors can be set up for all of the desired connections via the configuration file, or for each individual connection by entering a command (START-MSCF-CONNECTION, see [page 293](#)). The type (LCS or CCS) of connection to the respective partner processor is also defined at this point.

If a connection is lost (indicated by message MCS0014), it can also be re-established with the START-MSCF-CONNECTION command.

The connection to a partner processor is only established if the corresponding command for the local processor has already been issued on the partner processor named in the command. If this is not the case (message MCS0009 is output), the local processor notes that a connection is to be established to the partner processor. The connection is set up as soon as the START-MSCF-CONNECTION command for the local processor is issued on the partner processor (message MCS0013 is output on the local processor, message MCS0010 is output on the partner processor).

With each connection setup, HIPLEX MSCF updates the availability of the of the home pubset in MRSCAT in both the requesting and the accepting processor ([page 62](#)).

Notes

- An MSCF connection cannot be established if the SYSID of a processor is the same as that of another processor in the same network. The MSCF connection setup is rejected with message MCS0009 (error code: 08).
- A processor cannot be connected to more than 163 partners at the same time (restriction imposed by the SYSID).
- If the BCAM-SOF file was not processed in full due to an error when loading BCAM, the Start Option File (SOF) must be explicitly executed again with CALL-BCAM-COMMANDS before MSCF connections are established (reason: incomplete network or BCMAP configuration).

6.3 Clearing down connections

The STOP-MSCF-CONNECTION command clears down a connection to a partner processor (see [page 303](#)). Message MCS0015 is output on the local processor, and message MCS0014 is output on the partner processor.

If the command is executed when no connection exists to the partner processor specified in the command, the request to clear down a connection to the partner processor is canceled (message MCS0015 is output).

Clearing down the MSCF connection between two processors can result in restrictions to the shared pubset or XCS functionality on both processors. For this reason, the clear down request in the STOP-MSCF-CONNECTION command for a connection between a master and slave processor in a shared pubset network will be rejected (message MCS0021 is output), but can be forced by specifying the operand FORCED=*YES.

The MSCF connection between two processors is cleared down implicitly when HIPLEX MSCF is terminated on either of the two processors.

For information on connection problems and how to handle them, see [chapter “Error recovery” on page 163](#).

6.4 Terminating HIPLEX MSCF

HIPLEX MSCF is terminated either directly via the STOP-SUBSYSTEM MSCF command or indirectly in the course of the shutdown procedure. Termination not only clears down the MSCF connections, it also removes the processor from *all* networks and exports all shared pubsets. In a parallel HIPLEX all exclusive pubsets that have at least one GS volume are also exported. If the processor was the master processor for a shared pubset, a master change is initiated by the processors remaining in the network. XCS processors exit the XCS network.

If HIPLEX MSCF is terminated on a processor, the processor is removed from the MSCF network and all tasks of the processor must return their HIPLEX MSCF resources. For this reason, the STOP-SUBSYSTEM command is rejected if shared pubsets are still imported on the processor (message MCS0021 is output). If MSCF termination is forced by the command STOP-SUBSYSTEM MSCF, SUBSYSTEM-PARAMETER='FORCED=YES', all user tasks still using shared pubset or XCS functionality are aborted with CANCEL-JOB before the processor is disconnected from the network.

The MSCF subsystem does not terminate until the entire HIPLEX MSCF functionality has been shut down in full on the processor, because only then is the processor permitted to re-enter the network.

If HIPLEX MSCF terminates automatically (e.g. due to a configuration error), the processor disconnects from the network and all shared pubsets are exported by force. HIPLEX MSCF is unloaded after the last shared pubset has been exported successfully.

Notes

- If serious errors occur within the MSCF network, the MSCF subsystem terminates automatically. The processor disconnects from the network, and message MCS1003 is output.
- The MSCF network is based on BCAM. BCAM commands can therefore be used for HIPLEX MSCF.
 - Issuing the BCAM command BCOUT for a processor connected to the MSCF network disconnects this processor from the network.
 - When the BCAM command BCEND is executed, BCAM checks whether MSCF is still running. If it is, first the STOP-SUBSYSTEM MSCF, SUBSYSTEM-PARAMETER='FORCED=YES' command is executed, and then BCAM delays further processing of BCEND until either MSCF has terminated or the maximum wait time set in the BCAM parameter MAX-MSCF-DELAY (see [page 114](#)) has elapsed.

- If BCAM commands are executed, the functionality of the commands STOP-MSCF-CONNECTION and STOP-SUBSYSTEM MSCF does not come into play, which can have serious consequences for the multiprocessor system and in particular for shared pubset or XCS operation (risk of erroneous crash detection, DMS return codes on the slave processors of a shared pubset).
- When HIPLEX MSCF is terminated, all shared pubsets are exported; the system waits for the release of DMS occupation. This can cause a delay in the unloading of MSCF if applications which occupy the shared pubset are not terminated before shutdown. UTM applications that use shared pubsets are terminated automatically by the export processing.
- When a system shutdown is to take place on multiple processors in an MSCF network, it is recommended that these shutdowns should be initiated one after the other on the processors concerned and that completion of an initiated shutdown should be waited for to give the other processors which are (still) in the MSCF network the opportunity to register the exit from the MSCF network of a processor which is shutting down.

When multiple processors in an MSCF network are to be terminated more or less simultaneously (e.g. monitor system and guest systems), this can lead to blocks when a system is shut down. Such blocks are cleared only by shutting down BCAM and subsequent abnormal termination of HIPLEX MSCF (after 10 minutes).

6.5 Displaying MSCF connections

Information on the MSCF connections of the processor can be requested at any time by issuing the SHOW-MSCF-CONFIGURATION command (see [page 227](#)).

From the point of view of the local processor, the result of an attempt to establish a connection with another processor is one of the following basic states (connection statuses):

- NOT MSCF-CONNECTED: The local processor is ready to establish an MSCF connection with this partner processor, but the partner processor does not want the connection, is not yet active, or the connection to the partner has failed.
- CONNECTED: An MSCF connection has been established to the partner processor.
- UNDEFINED: The local processor does not want an MSCF connection to the partner processor.

6.6 Shared pubset network

The prerequisites for the CCS network also apply to a shared pubset (SPVS) network. Please also note the following information:

6.6.1 Setting up a shared pubset network

A shared pubset network comprises a master processor and a maximum of 15 slave processors with hardware paths to a shared pubset.

The prerequisite for starting the shared pubset network is that HIPLEX MSCF is loaded on all participating processors and the necessary MSCF connections are established between the processors. The minimum requirement is one CCS connection (see [page 30](#)) between the master and slave processor. For automatic master changes and comprehensive network monitoring, CCS connections must also exist between all slave processors.

The shared pubset network is established when the pubset is imported on the master processor. The network is cleared down when the pubset is exported from the last sharer.

MRSCAT entries (catalog IDs) must be created on the processors for the shared pubset involved. The HIPLEX MSCF requirements for the associated SYSIDs of the home pubsets are explained on [page 112](#). The BCAM dependencies that must be taken into account in relation to the shared pubset network are described on [page 113](#).

Importing a shared pubset

The shared pubsets provided for the shared pubset network must be imported on all processors of the network. The shared pubsets are imported explicitly using the command `IMPORT-PUBSET PUBSET=<catid>,USE=*SHARE,SHARER-TYPE=*MASTER` (import to master processor) or `IMPORT-PUBSET ... ,SHARER-TYPE=*SLAVE` (import to slave processor). If the operand `SHARER-TYPE` is not specified, its value is determined implicitly using the value of `CURRENT-MASTER` currently entered in the SVL for the pubset and the defined `DESIGNATED-MASTER` (for more information, see [section "Generation of the shared pubset network" on page 131](#)).

The following preconditions must be fulfilled to import the shared pubset:

- The processor must have direct hardware paths to the devices on which the pubset resides.
- All devices which contain the pubset to be imported must be "attached" (`ATTACH-DEVICE` command on the console or corresponding generation).
- The pubset to be imported must be declared as "shared" in the SVL entry (on the pubres) (`SET-PUBSET-ATTRIBUTES` command).
- The pubset to be imported must be declared as "shared" in the MRSCAT entry (`ADD-` or `MODIFY-MASTER-CATALOG-ENTRY` command).
- The pubset must not be the home pubset or the paging pubset of the processor.
- The status of the pubset in the MRSCAT must be "inaccessible". For a slave import, "remote accessible" is also permitted.
- The pubset must not already be imported exclusively on other processors (if necessary, export from there).
- HIPLEX MSCF must be loaded and a connection must be established to the current master processor (CCS connections must exist between the master processor and all slave processors). If this is not the case, a slave import is pended (message `DMS037C`); a master import is rejected with message `DMS037B`.
- In VM2000 operation, the devices containing the pubset must be assigned to the VM of the importing guest system. This assignment is implemented on the monitor system with the VM2000 command `ADD-VM-DEVICES`. The operand `TYPE=*SD` is required if the pubset is to be imported to several guest systems of the same VM system, and should only be used in this case.

Master import of a pubset

When the pubset is imported to the master processor, the processor establishes the shared pubset network. At this time, there need be no MSCF connections to the slave processors. However, the pubset can only be imported to the master processor if it is not imported on any other processor. If the previous pubset session was not terminated properly, i.e. if processors which might still be accessing the pubset are still entered on the pubset, the import procedure will be discontinued and systems support will be requested to intervene.

- If the pubset session during which the pubset was imported exclusively was not terminated properly, message NKA0019 is output. If it is guaranteed that the respective sharer is no longer accessing the pubset, the sharer can be removed from the sharer list using the UNLOCK-DISK command. Following the appropriate response to message NKVD072 for UNLOCK-DISK, the master import is continued automatically.
- If the pubset was previously imported as "shared" and if the former master processor is still entered, message DMS13C7 is output and a response must be given. If it is guaranteed that the processor is no longer accessing the pubset, the message can be answered with "<tsn>.FORCE-MASTER-IMPORT" and the master import thus implemented by force. The import procedure is aborted if "<tsn>.N".
- If the pubset was previously imported as "shared" and if slave processors are still entered and it is determined that the pubset is still imported on at least one of these, the import process is aborted. A master import of this pubset is only possible if one of the remaining slave processors performs a master change (see "Importing a shared pubset with master change" on the next page) or if all slave processors that are still active export the pubset.
- If the pubset was previously imported as "shared" and if slave processors are still entered in the SVL and it cannot be determined whether or not the pubset is still imported on these processors, the DMS13D3 message, which requires a response, is output for each slave processor. If it is guaranteed that the processor is no longer accessing the pubset, the message can be answered with "<tsn>.Y", thus forcing the master import. If the answer "<tsn>.N" is given, the import process is aborted.

Slave import of a pubset

If the shared pubset is imported on the slave processor, this means that the processor wants to join an *existing* shared pubset network. The following conditions must be fulfilled in this case:

- HIPLEX MSCF must be loaded.
- A master processor must be established.
- A CCS connection must exist to the master processor.

If any of these conditions is not fulfilled, the slave import is halted until all conditions are fulfilled. Only then can the processor connect to the network.

The status of a shared pubset can be queried using the commands SHOW-MASTER-CATALOG-ENTRY, SHOW-PUBSET-PARAMETER and possibly SHOW-SHARED-PUBSET. If a pubset is imported as a shared pubset, the master processor is also specified in the output of the SHOW-MASTER-CATALOG-ENTRY command (on the master processor: MASTER-HOST=OWN-HOST, on the slave: MASTER-HOST=<processor-name of master-processor>).

Importing a shared pubset with master change

In addition to the master and slave import, there is also the special case of "import with master change".

If a master change fails, the cause of the error that led to the failure must first be eliminated. The shared pubset can then be imported on the slave processor which is to be the new master processor using the IMPORT-PUBSET PUBSET=<catid>,SHARER-TYPE=*MASTER(MASTER-CHANGE=*YES) command. The processor is thereby established as the new master processor for the shared pubset. However, the command can only be applied in this form after a master change has failed or has not been initiated (i.e. the pubset is in the INACC,QUIET state).

Differences from exclusively imported pubsets

Modifications to the user catalog via the commands ADD-USER, MODIFY-USER-ATTRIBUTES, LOCK-/UNLOCK-USER and REMOVE-USER can only be implemented from the current master processor.

SCA (Speed Catalog Access) for a shared pubset can only be started and stopped on the current master processor of the shared pubset.



A slave import which is waiting for the import of the master processor or for the connection to be established to the master processor, can be terminated by an export command.

6.6.2 Clearing down a shared pubset network

When the shared pubset is exported, a processor is disconnected from the shared pubset network.

Exporting a shared pubset

A shared pubset is exported when the EXPORT-PUBSET command is issued on the processor or HIPLEX MSCF is terminated. A distinction is made between a master and slave export, and with a master export an additional distinction is whether or not a master change is to take place.

Slave export

If a slave processor exports a shared pubset, its resources are released to the master processor (including those resources that could not be released during the slave pubset session due to communication problems).

Master export without master change

If the EXPORT-PUBSET command is executed on the master processor without explicitly requesting a master change, the entire shared pubset network is cleared down. The pubset is therefore exported not only from the master processor but from all slave processors as well.

Master export with master change

The EXPORT-PUBSET <catid>,SHARER-TYPE=*MASTER(MASTER-CHANGE=*YES) command executed on the master processor has the following effects:

- After the command starts, the DMS management operations are halted on the slave processors.
- The shared pubset is exported from the master processor.
- When the export procedure has terminated on the master processor, a master change takes place automatically on the slave processors. A new master processor is selected from the set of slave- processors.
- After the master change has concluded, the pubset is again available in full on the former slave processors (with new master processor). As far as the applications on the slave processors are concerned, this master change is just as transparent as the master change following a shutdown or failure of the master processor.

An export with master change is initiated automatically when MSCF is terminated on the master processor.

Notes

- A master change is only possible if it is permitted (see [page 133](#)) and the necessary CCS connections are established.
- If EXPORT-PUBSET TERMINATE-JOBS=*YES is specified UTM applications that use the shared pubset receive an end signal when they use the pubset.
- The POSIX file systems /root and /var should not be placed on shared pubsets, otherwise POSIX must be terminated manually so that the affected pubsets can be exported outside of a shutdown. This could lead to blocks, especially during automatic fail reconfiguration (RECOVERY-START=*AUTOMATIC / *SECURE), because MSCF cannot be terminated until all shared pubsets have been exported.

6.6.3 Retrieving information on shared pubsets

The SHOW-PUBSET-ATTRIBUTES command provides information on the current values of the pubset attributes (defined using the SET-PUBSET-ATTRIBUTES command). The SHARE attribute of the pubset, the current master processor, and the specifications for master processor and master change are displayed.

The SHOW-PUBSET-PARAMETERS command provides information on the defined operating parameters of the shared pubset. The INFORMATION operand allows you to specifically query which tasks are currently occupying the pubset (*PUBSET-OCCUPATION); *GLOBAL-PARAMETERS returns information on the values of the predefined and current network parameters of the pubset, the import mode, whether or not the pubset is an XCS pubset, etc.

The SHOW-MASTER-CATALOG-ENTRY command outputs the operating states of the pubsets defined on the processor. Selection can be made on the basis of shared pubsets, XCS pubsets, and pubsets with incorrect master change, amongst other things.

The SHOW-SHARED-PUBSET command provides information on the shared pubset configuration of the processor. The processors connected to the network, together with their network status, are output for each imported shared pubset.

6.6.4 Operating states of a shared pubset

The SHOW-MASTER-CATALOG-ENTRY command contains information about the operating state of a shared pubset. The following operating states are possible:

- **INACC**
The shared pubset is not imported; it is in inaccessible state.
- **LOCAL-IMPORTED, SHARED**
The shared pubset is imported and in accessible state. The catalog can be accessed from the processor. In addition, information is provided via MASTER-HOST: If OWN-HOST is specified, the local processor is the master processor. If <processor-name> is specified, the local processor is a slave processor and the specified partner is the master processor.
- **LOCAL-IMPORTED, QUIET**
The MSCF connection to the master processor is temporarily interrupted or an export with master change has been initiated on the master processor but the master change has not yet been started.
- **LOCAL-IMPORTED, QUIET, MASTER-CHANGE-IN-PROCESS**
The master change is in progress.
- **INACC, QUIET**
The automatic master change was rejected or terminated abnormally. The pubset is in a "masterless" state. Since DMS metaoperations are cannot be executed in this state, systems support should eliminate the cause of the error as quickly as possible and then start a master change on a slave processor with the `IMPORT-PUBSET <catid>,SHARER-TYP=MASTER(MASTER-CHANGE=*YES)` command. The pubset session can be resumed when the master change is successfully completed.
The SHOW-MASTER-CATALOG-ENTRY `SELECT=*MASTER-CHANGE-ERROR` command contains information about shared pubsets in "INACC, QUIET" state.
- **IMPORT-IN-PROCESS**
The pubset is in the process of being imported. This state occurs when a slave import process is in wait state due to one of the following causes (message DMS037C is output):
 - master processor not available
 - MSCF connection is not set up
- **EXPORT-IN-PROCESS**
The pubset is in the process of being exported. This state also occurs when
 - the export processing is waiting for the shared pubset assignment to be returned or
 - during a master export the system is waiting for the slave export to be terminated.

6.7 XCS network

The XCS network connects up to 16 processors to form a function network. CCS connections are established between all processors and the processors have imported at least one shared pubset (XCS pubset).

A processor joins the XCS network when HIPLEX MSCF is started, and leaves the XCS network again when HIPLEX MSCF is terminated. The entry, exit, and failure of a partner processor result in the reconfiguration of the XCS network on all processors connected to the network.

The current status of the XCS network can be determined by issuing the SHOW-MSCF-CONFIGURATION command.

The conditions that must be fulfilled for the shared pubset network also apply to the XCS network.

6.7.1 Entry/re-entry to the XCS network

A processor enters the XCS network when HIPLEX MSCF is started (see [page 144](#)). Depending on the entry in the subsystem catalog, HIPLEX MSCF is started either when the MSCF subsystem is started in the course of the BS2000 startup procedure, or at a later stage through an explicit START-SUBSYSTEM MSCF command. Use of a configuration file is mandatory for XCS entry.

The registered functions necessary for entering the network are initialized by configuration management (XCM) in the local initialization phase of HIPLEX MSCF. The NSM and XCS-TIME subsystems required for this are automatically loaded later by MSCF. If this is not possible, MSCF too terminates again.

If the local initialization phase is concluded successfully, the communication system initiates the connection setup to the partner processors predefined in the configuration file. If a partner is active and is defined as the HIPLEX MSCF partner, an MSCF connection is established between the local processor and the partner processor. Otherwise, the partner processor is not noted as connected (see [page 147](#)).

When the connection setup phase has concluded, configuration management determines the XCS status of the active HIPLEX MSCF partner. If the XCS network does not yet exist, the processor establishes the network (Create). If the XCS network already exists, the processor enters this network (Join) and the partner processors add the processor to the network (join reconfiguration).

The registered functions are started in succession on the processor joining the network. The XCS functionality managed by each of the functions is made available on the processor. When a processor joins a network, each function implements a join reconfiguration in conjunction with its partner entities.

The network setup is concluded when all functions have been started successfully. Network entry is not completed, however, until the processor has also been successfully added to the network on all partner processors.

If HIPLEX MSCF has been terminated on a processor, the processor can only rejoin the XCS network when HIPLEX MSCF is restarted if the appropriate leave or fail reconfiguration has been completed in full on the partner processors and the processor is no longer known to the network. If this condition is not fulfilled, the active partner processors will reject the connection request of the processor.

Notes

- Connection requests from CCS partners will be rejected at first. After the processor joins the XCS network, connections will be established to these partners.
- It is only possible for a processor to join an existing XCS network if CCS connections have been established to all active sharers. If this is not the case, the start of HIPLEX MSCF will be aborted.
- If no connections are set up to any of the active XCS partners due to an error, the processor itself attempts to establish the XCS network (Create). When importing the XCS pubset, however, the processor determines that other processors are entered on the pubset. Message DMS13C7 is then output, requesting systems support to decide whether the master processor entered can be ignored. However, since the other XCS partners are still active, the import must be aborted in this case (input: <tsn>.N). HIPLEX MSCF is terminated abnormally.
- While a processor is joining the network, no other processor can be added to the network or leave the network. Such operations are delayed until the network entry is completed.
- If the network is in a reconfiguration phase, the entry of a processor is delayed until this phase is concluded. If necessary, the reconfiguring processors should be checked to determine whether the reconfiguration is blocked.
- If the XCS network is corrupted due to a connection failure or a processor crash, or if such an error occurs while a processor is joining the network, the start of HIPLEX MSCF is aborted and the MSCF subsystem is terminated. A fail reconfiguration is initiated on the partner processors. When the fail reconfiguration is concluded on the partner processors and the error has been rectified, the network entry can be restarted by issuing the START-SUBSYSTEM MSCF command.

6.7.2 Leaving the XCS network

A processor leaves the XCS network when HIPLEX MSCF is terminated. HIPLEX MSCF is terminated when the MSCF subsystem is terminated in the course of the BS2000 shutdown procedure, or can be terminated explicitly by issuing the `STOP-SUBSYSTEM MSCF,SUBSYSTEM-PARAMETER='FORCED=YES'` command. When this command is executed, the entire HIPLEX MSCF functionality of the processor is cleared down and all imported shared pubsets are exported. The remaining (local) functionality of the processor is retained.

HIPLEX MSCF is terminated in a number of phases. In the first phase, the user tasks still using XCS functionality are terminated with `CANCEL`. Network exit is initiated at the latest with the expiration of the period defined by the MSCF configuration parameter `USER-TERM-LIMIT`. If the processor is the last sharer in the XCS network, the network is cleared down (termination). Otherwise, the processor leaves the network and the partner processors remaining in the network disconnect the processor from their network (leave reconfiguration). The registered functions are terminated in succession on the processor. The functions first of all terminate throughout the network (global termination) and then clear down their local resources (local termination). The network clear-down concludes when the global termination of the last function is completed. When a processor leaves the network, its exit is not completed until the partner processors have also disconnected this processor from the network. Following the network exit and the local termination of the functions, the MSCF connections are cleared down and the MSCF subsystem is terminated.

Notes

- The exit from the XCS network is coordinated with the partners. The exiting processor only disconnects from the network in uncoordinated mode if cooperation between the network sharers has been disrupted due to the loss of a connection or some other error. Following a crash, the XCS network must then be recovered by the remaining sharers in the network by means of a fail reconfiguration.
- While a processor is leaving the network, no other processor can be added to the network or leave the network. Such operations will be delayed until the network exit has concluded.
- If the network is in a reconfiguration phase, the exit of the processor (and hence also the shutdown) is delayed until the reconfiguration is completed. However, if the reconfiguration lasts longer than 10 minutes, the processor disconnects from the network (message `MCS1003`, reason 14).

6.7.3 Handling an XCS pubset

The existence of an XCS pubset is part of the functionality of the XCS network and can be a prerequisite of the distributed system functions and the XCS applications. The import of the XCS pubset at network entry and export at network exit are implemented automatically by MSCF, as is the master change if the current master leaves the network or fails. Systems support must *not* import XCS pubsets. Operator intervention is only required in the event of an aborted master change or an input/output error.

When the XCS network is active, additional pubsets can be added to the XCS pubsets already active (i.e. shared pubsets declared as XCS pubsets). On the other hand, XCS pubset can also be removed from the XCS network, provided the last pubset is not removed (the XCS network requires at least one XCS pubset).

To add an XCS pubset to the network, the pubset must first be imported as a shared pubset on all processors in the XCS network and then declared an XCS pubset via the SET-XCS-PUBSET command. The SET-XCS-PUBSET definition only applies for the duration of the active XCS session.

If an XCS pubset is to be removed from the network, the pubset must be exported from the current master processor of the pubset.

In the event of write errors in the disk protocol of an XCS pubset, the pubset can be exported from the respective processor (e.g. to perform hardware repairs). During the period when no XCS pubset is available or an XCS pubset is not imported on every processor in the network, a join reconfiguration is not possible. The faulty XCS pubset should be exported from all processors connected in the network. If this was the last XCS pubset of the network, a new XCS pubset should be added to the network as soon as possible.

6.7.4 Displaying the XCS configuration

The SHOW-MSCF-CONFIGURATION command provides information on the current status of the XCS network from the point of view of the local processor. The command allows you to query the current XCS configuration, the current status of the XCS network on the processor, and the view of the processor on the partners. Any error and wait states (e.g. blockings) are also displayed.

7 Error recovery

Any important changes of status and any errors detected are reported; blocks are highlighted by periodic messages. When potential conflicts arise, messages are output and systems support must make a decision. Non-critical errors are logged in the SERSLOG, which is also a source of supplementary information for any error displayed.

Regardless of the type of network, it is essential that systems support be kept informed of the status of all processors connected in the network. It is therefore advisable to check the MSCF connections and the status of shared pubsets on a regular basis using the following commands:

```
SHOW-MSCF-CONFIGURATION
SHOW-MASTER-CATALOG-ENTRY
SHOW-SHARED-PUBSET
```

When analyzing network problems, particular consideration should also be given to the messages DMS03Bx, MCAxxxx and MCSxxxx in the conslog files of all processors in the network.

7.1 Email notification about critical states

Critical states can be reported by sending an email to an instance provided for this purpose. The email describes the situation. Thus an opportunity is provided to solve the problem occurred by manual intervention within short time.

Critical states concerning MSCF network functionality for example are caused by unexpected loss of MSCF connection to a network partner or by master change failure for a shared pubset.

The catalog entry of the user to be destined as instance for monitoring and error recovery has to be supplemented with one email address or multiple email addresses by using the MODIFY-USER-ATTRIBUTES or ADD-USER commands. In case of critical problems, an email describing the problem occurred will be sent to the mail address(es) found in the catalog entry of the user ID that has been specified in the NOTIFY-BY-MAIL operand of the SET-MSCF-ENVIRONMENT or MODIFY-MSCF-ENVIRONMENT commands.

7.2 Analyzing the system status

Errors occurring within an MSCF network can be multifaceted. The following commands help to analyze the particular situation and determine the cause of the error.

Information functions (see the "Commands" manual [10])

Command	Function
SHOW-MASTER-CATALOG-ENTRY SELECT=SHARED	Displays the pubsets currently used as shared pubsets. The information is taken from the local MRSCAT.
SHOW-MSCF-CONFIGURATION	Provides information on the status of the MSCF connections, the MSCF networks, and the reconfiguration entities from the point of view of the local processor, as well as information on the global control parameters.
SHOW-PUBSET-ATTRIBUTES	Displays the current pubset settings with regard to shareability and mastership or master change, which were defined via the SET-PUBSET-ATTRIBUTES command. The information is taken from the SVL of the first pubset volume.
SHOW-PUBSET-PARAMETERS	Displays the preset and currently used operating parameters of the pubset, as well as the occupying tasks. Selection based on shared or XCS pubsets is supported. The command provides information for the error situation "hanging shared pubset occupation". The information is taken from the local MRSCAT.
SHOW-SHARED-PUBSET	Displays the sharer configuration of the individual shared pubsets (from the point of view of the local processor) if the local processor is connected to shared pubset networks. Selection based on shared or XCS pubsets is supported. The information is taken from the local watchdog tables.
SHOW-SUBSYSTEM-STATUS MSCF	In addition to the status of the MSCF subsystem from the point of view of DSSM, this command displays the TSNs of the tasks connected to the subsystem. With MSCF, these are the tasks MSCC, MSCM, MXCM, WDGS und WDWT and the MSCF server tasks.
SHOW-XCS-OCCUPATION	If the local processor is connected to an XCS network, this command is used to determine the local TU tasks that are currently using XCS-wide locks. This command provides information for the error situation "hanging XCS occupation".

Table 1: Command overview of information functions

In the event of disk or connection problems, the following commands may be useful:

Command	Function												
<pre>SHOW-DISK-STATUS UNIT=<mnemonic> *VOLUME(VOLUME=<vsn>), INFORMATION=*SYSTEMS *ALL</pre> <p>(see the "Commands" manual [10])</p>	<p>The SYSIDs of the processors in a shared pubset network which are entered in the SVL of the pubset are displayed under SYSTEMS.</p>												
<p>BCSHOW (see the manual "BCAM" [13])</p>	<p>Displays detailed information on the BCAM network on which the MSCF network is based.</p>												
<pre>BCSHOW APPLICATION=... , SHOW=ROUTE</pre> <p>(see the "BCAM" manual [13])</p>	<p>For each application, displays the established BCAM connections and their data (e.g. protocol used (TCP/IP, ISO4), MAXTSDU setting). An MSCF connection is made up of four or five BCAM connections, one between each of the following applications:</p> <table border="0" style="margin-left: 40px;"> <tr> <td style="padding-right: 20px;">(local)</td> <td>(remote)</td> </tr> <tr> <td>\$MRSAPP</td> <td>\$MRSAPP</td> </tr> <tr> <td>\$MRSAPP</td> <td>\$MRSAPPS</td> </tr> <tr> <td>\$MRSAPPS</td> <td>\$MRSAPP</td> </tr> <tr> <td>\$MRSAPP4</td> <td>\$MRSAPP4</td> </tr> <tr> <td>\$MRSAPP5</td> <td>\$MRSAPP5 (optional)</td> </tr> </table> <p>Two routes for the \$MRSAPP application and one route each for the \$MRSAPPS, \$MRSAPP4 and possibly the \$MRSAPP5 application are therefore displayed on a processor for each partner, if the MSCF connection is intact (i.e. no recovery is active).</p> <p>All sharers in the XCS network are connected via the \$MCSNLX applications. In an intact XCS network, one route is therefore displayed per partner on each processor for the \$MCSNLX application.</p> <p>The sharers of the XCS network are connected to form a ring via the \$MCSNSM applications; for this application, a route is therefore not displayed for each partner.</p>	(local)	(remote)	\$MRSAPP	\$MRSAPP	\$MRSAPP	\$MRSAPPS	\$MRSAPPS	\$MRSAPP	\$MRSAPP4	\$MRSAPP4	\$MRSAPP5	\$MRSAPP5 (optional)
(local)	(remote)												
\$MRSAPP	\$MRSAPP												
\$MRSAPP	\$MRSAPPS												
\$MRSAPPS	\$MRSAPP												
\$MRSAPP4	\$MRSAPP4												
\$MRSAPP5	\$MRSAPP5 (optional)												

Table 2: Command overview of information functions for disk and connection problems

Documentation for error diagnosis

The following documentation should be made available for error diagnosis:

- CONSLOG file
- SERSLOG file
- MSCF trace file
- BCAM trace

7.3 Server task bottleneck

Additional MSCF server tasks are generated to cope with an increase in the number of jobs sent to the processor from the partner processors, and also to avoid inter-processor deadlock situations. If the number of jobs decreases, the superfluous additional server tasks are cleared down. The number of server tasks on the master processor fluctuates in particular in a shared pubset network. An incorrect configuration of the shared pubset network (e.g. low-performance master processor and high-performance slave processor) can overload the master processor in extreme cases. A dramatic increase in the number of MSCF server tasks can result in a memory saturation or can even bring the system to a standstill. The existence of more than a certain number of server tasks on the master processor (see the [“MSCF configuration parameter SERVER-TASK-LIMIT” on page 128](#)) will mean that jobs originating from the partner processors (slave processors) will therefore be put on hold until there are sufficient free MSCF server tasks on the local processor.

The line "NUMBER OF SERVERS: CURRENT=x FREE=y" in the output of the command SHOW-MSCF-CONFIGURATION INF=*LOCAL indicates the number of actual and free MSCF server tasks. If the value for FREE is "0" and the value for ACTUAL is the same as that set for SERVER-TASK-LIMIT or even greater, this indicates a high load from the partner processors. If this status persists, suitable measures should be taken to resolve it (e.g. increasing the server task limit, master change on the processor with the highest catalog load).

7.4 Communication bottleneck

If an MSCF connection is overloaded, MSCF may not be able to transfer any more messages to BCAM for transmission to the partner processors. This type of overload can occur sporadically, but should rectify itself again. The internal mechanisms of HIPLEX MSCF can bridge short-term overloads. However, if the bottleneck does not resolve itself, the connection to the respective partner processor can be cleared down as an emergency measure with the STOP-MSCF-CONNECTION command, possibly with FORCED=*YES (see [page 303](#)), and can then be re-established using the START-MSCF-CONNECTION command.

The SHOW-MSCF-CONFIGURATION command can be used to determine whether a connection is overloaded as outlined above. In the command output, the column "OVL" (OVERLOADED) indicates whether or not the connection is overloaded:

- NN: Congestion-free transmission
- YN: Congestion for requests (=jobs from the local processor to the partner processor)
- NY: Congestion for replies (=responses from the local processor to jobs from the partner processor)
- YY: Congestion for requests and replies (i.e. congestion in both directions) or a connection recovery is active

7.5 Failure of BCAM connections

In general, BCAM detects the loss of a BCAM connection. In particular situations, however, BCAM cannot determine the loss of connection itself (e.g. if the remote processor fails). For this reason, MSCF monitors the MSCF connections by periodically sending monitoring messages to the partner, which must be answered within the defined time span (FAIL-DETECTION-LIMIT / 11). If the response is not received or if BCAM reports a loss of connection, for MSCF there is an error in the MSCF connection and a connection recovery is initiated. In this case, the associated BCAM connections are first cleared down. An attempt is then made to reestablish these connections. During the recovery procedure, the command SHOW-MSCF-CONFIGURATION indicates the connection as CONNECTED, even though jobs to the partner can be rejected during this time.

If the connection can be reestablished (successful recovery), message MCS0018 is output on the console of both processors. Due to internal temporal dependencies during the recovery, it may happen that messages MCS0014 and (later) MCS0013 are output on one of the processors, while message MCS0018 is output on the other processor.

If the connection cannot be reestablished (connection failure), message MCS0014 is output on the console. The START-MSCF-CONNECTION command can be issued in an attempt to reestablish the connection manually.

A connection error impairs the partner monitoring mechanisms. If a connection error occurs in conjunction with a write error during the disk monitoring, this can lead to erroneous crash detection by the partner and, if operation continues, to inconsistencies in the data. This situation is dealt with in [section “Input/output errors when accessing the watchdog file” on page 181](#).

7.6 Loss of connection in an LCS/CCS network

If the MSCF connection to an LCS/CCS partner fails or is cleared down (messages MCS0014 and MCS0015 are output), the partner is no longer accessible. Pubsets imported remotely via the partner are switched to the QUIET state. JMS, CJC, and catalog accesses to the partner are rejected.

If the connection is cleared down by a command or by the termination of MSCF (but not by the failure of the connection), the catalog locks maintained by the partner are reset on the processor.

For further information, see also section [“Wait time for reestablishment of MSCF connection” on page 170](#).

7.7 Connection cleardown/loss in a shared pubset network

The cleardown or loss of an MSCF connection (messages MCS0014 and MCS0015 are output) between the sharers of a shared pubset network affects the functionality of the network. The only immediate effect on the users of a shared pubset is the interruption of the MSCF connection between the master and slave processor. In this case, the MRSCAT entry of the pubset on the slave processor is switched to the QUIET state. DMS metaoperations on the slave processor cannot be executed until the connection to the master processor is reestablished. The loss of a connection between two slave processors is of no significance to the other sharers of the shared pubset network. The pubset remains fully available to the processors.

If the shared pubset is an SM pubset, a loss of connection during a configuration modification may result in inconsistencies between the image of the pubset configuration in the MRSCAT on the slave processors and the pubset configuration file updated by the master processor. The only functions affected by this are the information functions, which display outdated information. The RESUME-PUBSET-RECONFIGURATION command can be issued after the MSCF connection has been reestablished in order to update the pubset configuration of an SM pubset in the MRSCAT using the information in the pubset configuration file.

In addition to the interference in the availability of the pubset, the loss of a connection between the master and slave processor affects the reconfiguration capability and the monitoring of the shared pubset network. A distinction is made between the following cases:

- *Export of pubset without master change*

If the MSCF connection between the master and slave processor is interrupted, the export of the pubset from the exporting processor is implemented without the cooperation of the partner processor.

If the pubset is exported from the slave processor, its locks are not released on the master processor. If the pubset is exported from the master processor, it remains imported on the slave processor.

At the end of the export procedure, the processor notes the status "EXPORT" in its assigned control blocks of the watchdog file and releases the pubset. The disk monitoring mechanism of the partner processor thus identifies that the status of the processor has changed and can implement the measures that have not been performed:

If the pubset is exported from a slave processor, the master processor resets the locks of the slave processor by means of a slave crash reconfiguration; if the pubset is exported from the master processor, the slave processor likewise exports the pubset.

If the EXPORT status cannot be written in the watchdog file (cause: write error or volume was removed), the partner processor cannot detect that the pubset has been exported until the MSCF connection is reestablished. Only then can the master processor reset the locks of the slave processor, or the slave processor export the pubset.

- *Master change due to export or failure*
The behavior for a master change initiated by an export with master change or by the failure of the master processor corresponds to the behavior in the event of an incompletely meshed network and is described in [section “Failure of a processor in a shared pubset network” on page 172](#) and in [section “Clearing down a shared pubset network” on page 155](#).

Wait time for reestablishment of MSCF connection

If a task wants to access a pubset that is QUIET (see [section “Loss of connection in an LCS/CCS network” on page 168](#)), it first waits for the MSCF connection to be reestablished. The maximum wait time is defined or modified using the BATCH-WAIT-TIME and DIALOG-WAIT-TIME operands of the ADD-MASTER-CATALOG-ENTRY and MODIFY-MASTER-CATALOG-ENTRY commands. The default value is 28800 seconds (= 8 hours) for batch jobs and 30 seconds for dialog jobs.

If the task is still waiting after 600 seconds, the message DMS03A8 is issued at the console so that systems support knows the connection has failed and can terminate the wait status. In general, the task recognizes status changes itself, withdraws its query and continues: If the pubset is available again, the current operation is continued normally. If the pubset has been exported and the answer to the message DMS03A8 requires processing to be aborted or if the wait time has expired, the current operation is aborted and a message (usually DMS0502) is issued.

If the connection failure interrupts internal system processes at an adverse point, it may occur that the task will not continue before the message has been answered, even if the connection has been reestablished in the meantime or the pubset has been exported. To stop the message DMS03A8 from being issued at all in unattended operation, we recommend that both wait times be set to 600 seconds in the MRSCAT entry.

7.8 Loss of connection in an XCS network

The XCS network is based on the use of shared resources by connected processors. Disruptions in the ability of the sharers to cooperate therefore represent a serious restriction to the functionality of XCS. If non-redundant connections fail, the MSCF connection must be reestablished quickly or the processor disconnected, so that the XCS network can continue to operate unhindered.

Since network entry and exit require an intact XCS configuration, all outstanding join and leave reconfigurations are aborted if a connection is lost between two XCS sharers. The processors entering and leaving the network disconnect from the MSCF network and terminate MSCF.

Crash detection and fail reconfigurations are only possible if the target configuration is error-free.

If a connection is lost between two active XCS sharers, automatic disconnection from the XCS network is not always acceptable. The system behavior can therefore be controlled by MSCF configuration parameters RECOVERY-START (see [page 124](#)).

Automatic error recovery (RECOVERY-START=*AUTOMATIC / *SECURE)

If the MSCF connection between two active processors in the XCS network fails, the one with the higher value for HOST-PRIORITY in the configuration file disconnects from the network. If both these processors have the same value, the one that entered the network later disconnects from it.

Operator decision

The failure of an MSCF connection between two processors is announced on the consoles of both processors by the question MCS1101. The question can only be answered indirectly by carrying out one of the measures cited in the message text. Systems support can do the following:

1. Eliminate the cause of the connection failure or provide a new path and then reestablish the MSCF connection via the START-MSCF-CONNECTION command.
2. Disconnect one of the XCS sharers affected by the loss of connection via the command STOP-SUBSYSTEM MSCF,SUBSYSTEM-PARAMETER='FORCED=YES'.

The fault remains until the MSCF connection is reestablished or one of the two processors disconnects from the XCS network.

7.9 Failure of an LCS/CCS partner

The failure of the partner is only noted as the failure of the MSCF connection. Catalog locks of the partner on all pubsets imported remotely by the partner can be reset on the processor by clearing down the MSCF connection or by terminating MSCF.

7.10 Failure of a processor in a shared pubset network

If a processor connected to a shared pubset network crashes, the network can be continued by the remaining sharers (following a reconfiguration). If a slave processor crashes, the master processor must therefore release the locks of the slave processor (slave crash processing); if the master processor crashes, a slave processor (backup master) must take on the role of the master processor (master change import).

The mechanisms for crash detection and the actions taken in the event of a crash depend on the BS2000 versions of the connected processors. The differences are explained in the individual reconfiguration steps.

1. Crash detection

Monitoring and crash detection for partners are partner-specific for all shared pubsets or is done by the Live Monitor of HIPLEX AF. For error recovery, it is possible to define whether these tasks are to be carried out automatically or whether they are to be initiated by systems support.

- A partner processor has crashed when this is detected by the Live Monitor.
- A partner processor may have crashed when the partner's disk protocol was no longer detected by the processor on all pubsets shared with the partner processor, and the MSCF connection monitoring function has found a connection error or no MSCF connection exists.
- If both monitoring paths are free of errors from the point of view of the processor, and if the vital-sign messages of the partner failed to appear on both monitoring paths simultaneously (i.e. within the intervals of the monitoring mechanisms) the partner is automatically declared as crashed if this is permitted by the rules described under [“Inhibiting the automatic start of fail reconfiguration” on page 125](#).

In all other cases, systems support must make a decision (message MCS1100); the partner is not declared as crashed until the message is answered with “CRASH” or “MTERM” (or more precisely “MXCM-<akz>.CRASH” or “MXCM-<akz>.MTERM”).

2. Failure of a sharer

If the partner processor has crashed, error recovery is started in parallel in the respective local tasks for the individual pubsets concerned. With a master-slave relationship, the failure of the partner is indicated by message DMS03B0; with a slave-slave relationship, message MCA0110 is output on the console.

If a slave processor crashes, slave crash processing is initiated on the master processor. No recovery measures are necessary on the other slave processors.

If the master processor crashes, one of the slave processors becomes the backup master. The backup master starts a master change import.

3. Determining the backup master

If the master processor crashes, each processor determines the backup master on the basis of its local watchdog sharer list, the sharers currently entered in the SVL, and the master change control parameters specified with the SET-PUBSET-ATTRIBUTE command.

If the processor itself is the backup master, it checks whether a master change can be implemented. If a master change is not possible (e.g. there is no connection to a slave processor), it is rejected with message MCA0104. Otherwise, a master change import is started.

If the processor is not the backup master, it waits for the start of the master change import on the backup master. However, if no connection exists to the backup master, the master change cannot take place (message MCA0104 is output).

4. Master change import

With a master change import, the individual DMS components (e.g. CMS, Allocator) must establish their master entity on the new master processor. In this case, the management data which has been lost must be recovered on the other processors in the shared pubset network in cooperation with the slave entities. During the master change import, all new requirements on the master processor are deferred until the master change is completed or has been aborted.

5. Slave crash processing

The master processor releases all the resources held by the slave processor. All locks (file and catalog entry locks) are reset.



Message MCS1100 is output on all affected processors which have noted that the partner has crashed. The message must only be answered on one processor. This answer applies for the entire network.

7.11 Termination of the master change import

If a master change is not possible (message MCA0104 is output) or if it is terminated due to an error (message MCA0103 is output), systems support can initiate a new master change after the cause of the error has been eliminated (IMPORT-PUBSET command with master change). If the master change is not possible with this command either, the pubset session must be ended and the pubset exported from all sharers.

7.12 Manual removal of a crashed sharer

If the pubset session cannot be ended properly (i.e. one or more sharers cannot export the pubset, error-free recovery was not possible), a new master import is only possible after the UNLOCK-DISK command has been executed. This command resets all the file locks set by a sharer with respect to a shared pubset. In addition, the SYSID of the crashed sharer is deleted from the SVL sharer list of the disk.

The current SVL sharer list can be obtained via the SHOW-DISK-STATUS command.

The command must only be used for sharers that are definitely not occupying the pubset. Otherwise, the integrity and consistency of the pubset cannot be guaranteed. If the respective pubset is an XCS pubset, an operating error of this type can endanger the integrity and consistency of the entire XCS network.

7.13 Failure of an XCS sharer

The XCS network is based on the use of shared resources by the connected processors. If a processor fails, the resources utilized by the crashed processor must be made available again so that the XCS network can continue to operate unhindered.

The failure of a partner is determined by the individual sharers using the two mechanisms "disk monitoring" and "connection monitoring" or by the Live Monitor of HIPLEX AF. In addition, the processors connected in the XCS network agree whether the partner is actually considered no longer active by all remaining sharers in the network. Based on the status of the partner with respect to these three factors, the partner can be classified into one of the error classes described below.

- *Partner crashed*

A processor assumes that a partner has crashed if the Live Monitor reports so or the following conditions apply simultaneously (i.e. within the interval defined for the monitoring algorithms and derived from FAIL-DETECTION-LIMIT):

 1. The disk monitoring mechanism detects the absence of vital-sign messages from the partner on all shared pubsets.
 2. The connection monitoring mechanism detects the absence of vital-sign messages from the partner to the monitoring telegram of the processor via the MSCF connection.
 3. The partner is no longer considered active by all other processors connected to the XCS network.
- *Partner status unknown*

The partner is no longer considered active by the processor and all other processors connected to the XCS network. However, due to the time difference between the absence of vital-sign messages and the connection failure, or due to another error in relation to the monitoring path, a crash cannot be assumed.
- *Loss of connection*

The partner is no longer considered active by the processor, but is considered active by another sharer in the XCS network. No automatism is initiated in this case, rather a decision is requested from systems support (see [section "Loss of connection in an XCS network" on page 171](#)).

The behavior in the event of a "partner crash" can be controlled by the MSCF configuration parameter RECOVERY-START (see [page 124](#)): either the system handles the situation automatically or systems support is requested to decide on the relevant measures. In the event of "partner status unknown", a decision is always requested from systems support.

Automatic error recovery

In the event that a “partner failed” the fail reconfiguration needed to release the global resources occupied by the processor that has crashed is started automatically.

Systems support decision

In the event of “partner status unknown” or if a RECOVERY-START setting does not permit automatic error handling, the message MCS1100 which requires an answer is issued to the consoles of all partners remaining in the network and a systems support decision is requested. Systems support can do the following:

- Start the fail reconfiguration with an appropriate response to the message from any of the participants. The following input choices are available:
 - MXCM-<order code of the console message>.CRASH
(CRASH: The BS2000 system on the partner has crashed)
 - MXCM-<order code of the console message>.MTERM
(MTERM: The MSCF subsystem on the partner was terminated).

The processors still connected to the XCS network perform a recovery. The failed processor is removed from the XCS network.

A fail reconfiguration must only be started if the processor has actually crashed or if it can be guaranteed that the processor can no longer access the shared resources (communication, shared pubsets, and shared GS).

- Reestablish the failed MSCF connection to the partner by issuing the START-MSCF-CONNECTION command, provided only the connection to the partner has failed and not the partner itself.
- Disconnect the remote processor from the XCS network by issuing the command STOP-SUBSYSTEM MSCF,SUBSYSTEM-PARAMETER='FORCE=YES', if the connection cannot be reestablished and then answer the message MCS1101 with MTERM.

The fault in the XCS configuration remains until a fail reconfiguration is implemented and the crashed processor has been removed from the XCS network or, if the processor has not crashed, the connections to the partner are reestablished.

7.14 Time elapsed until crash recovery

When the crash is detected by the Live Monitor, the `FADING-INTERVAL` operand determines the time which is to be waited following crash detection, see [page 72](#).

In the shared pubset network and the XCS network a connection failure is detected by MSCF at the latest after two monitoring intervals. The automatic fail reconfiguration is initiated after one further interval.

This gives the partners time to react to a possibly erroneous failure detection. A monitoring interval is calculated using `FAIL-DETECTION-LIMIT / 11`. So, with a setting of `FAIL-DETECTION-LIMIT=176` (minimum), MCS1100 is issued at the latest after 32 seconds.

If automatic failure handling is permitted, failure reconfiguration is started at the latest after 48 seconds.

7.15 Operation during crash recovery

If a processor crashes, within as short a time as possible the remaining processors in the network must be allowed to continue working and the crashed processor must be allowed to reenter the network. It is therefore very important that the crash recovery is completed as quickly as possible.

Normally the crash recovery occurs largely automatically, so it is not necessary for systems support to intervene. However, the automatic mechanisms offered by the system have their limits in the following cases:

- if the default settings are not correct
- if required task resources are not released
- if errors occur during processing.

If message MCS1100 (requiring a response) is output, you must check whether the displayed processor has merely lost its multiprocessor capability (MSCF has terminated) or has actually crashed. Only after this check can the message be answered with “MXCM-<order code of the console message>.MTERM” or with “MXCM-<order code of the console message>.CRASH”.

Often resources occupied by the crashed processor cannot be released for the remaining processors in the network or a function of the crashed processor which was necessary for the network cannot be taken over by another processor in the network. This is the case, for example, if the automatic master change in the shared pubset network has failed (you can display the affected pubsets with the command SHOW-MASTER-CATALOG-ENTRY SELECT=*MASTER-CHANGE-ERROR).

If the master change is rejected, a corresponding message (MCA0103, MCA0104) is output. When analyzing this message, do not forget that a master change must basically have been permitted via the pubset parameters (SHOW-PUBSET-ATTRIBUTES or SET-PUBSET-ATTRIBUTES command). There must also be an MSCF connection between the processor intended as the new master and all other sharers.

If there is a non-active sharer in the SVL (can be displayed with the command SHOW-DISK-STATUS <mn>,*ALL), this entry must be deleted with the UNLOCK-DISK <mn>,<sysid> command before the master change is initiated (the command can only be issued if the specified processor does not actually occupy the pubset). Immediately after this the master change can be started again with the command IMPORT-PUBSET <catid>, SHARER-TYPE=*MASTER(MASTER-CHANGE=*YES).

7.16 Erroneous partner crash detection

An erroneous partner crash detection occurs if a partner is declared as crashed without justification, either by systems support by answering message MCS1100 with “MTERM” or “CRASH” or by the system itself (see [“MSCF configuration parameter RECOVERY-START” on page 124](#)), even though the processor is still active. Since the global resources of the partner can be reused on the remaining processors, inconsistencies may arise. If a processor discovers when the MSCF connection is reestablished or by reading the vital-sign messages that a partner was falsely declared as crashed, configuration management disconnects from the network and HIPLEX MSCF terminates.

If two master processors have nonetheless been set up as a result of an error, the integrity of the shared pubset is no longer guaranteed; systems support must export the pubset immediately from all network sharers.

If data inconsistencies arise on the global resources, shared pubsets etc., it may be necessary to revert to an older backup version before MSCF is restarted.

7.17 Rejection of a partner's entry to the network

A connection setup to a partner or the import of a pubset shared by the partner will be rejected by MSCF configuration management if the integrity of the network would otherwise be endangered. The following points are checked:

- *Coordination of monitoring characteristics*
FAIL-DETECTION-LIMIT must be the same on both processors.
- *Uniqueness of partner*
The host name of the partner must be unique. A partner can therefore only re-enter the MSCF network of the processor when the previous network session with this partner has ended in full.
- *Uniqueness of XCS network*
The XCS name must be unique. Two different XCS networks with the same name must not be interconnected directly by HIPLEX MSCF or indirectly via another processor.

If a connection to a partner is rejected, this is indicated by the message MCS1005 being issued.

The rejection of the partner by configuration management results in the termination of the connection setup or the forced export (see [page 185](#)) of the shared pubsets by the watchdog. After a partner crashes, it should particularly be ensured that no unanswered failure inquiry (message MCS1100) is outstanding and that all reconfigurations are completed before HIPLEX MSCF is restarted on the crashed partner.

7.18 Input/output errors when accessing the watchdog file

To constantly monitor the shared pubset configuration, it is essential that the disk protocol be written and/or read on a continuous basis (see [page 31](#)). Input/output errors when accessing the disk therefore represent a restriction to the monitoring and reconfiguration capabilities of the shared pubset network. The use of several shared pubsets increases the effectiveness of disk monitoring with respect to errors on a disk.

The monitoring mechanism tolerates the interruption of a monitoring path. However, a simultaneous error on the other monitoring path (here, loss of connection), may result in an erroneous crash detection of the processor by the partner. Emergency measures (either immediate export or termination of system by MSCF) are unavoidable in this case.

Input/output errors can be classified into temporary and permanent (i.e. irrecoverable) errors. A temporary input/output error can be a delayed input/output operation or an input/output error rectified automatically by the system. The reading and writing of the watchdog file is not halted, rather it is continued periodically. When the input/output error is rectified, the disruption of the disk path is also eliminated.

A permanent input/output error, i.e. an irrecoverable error, can only be eliminated by the intervention of systems support (and possibly with hardware repairs). The pubset is switched to the WRTErr or READERR state (the SHOW-SHARED-PUBSET command provides information on the disk states). In the event of a permanent input/output error, the pubset should be exported as soon as possible and should not be imported again until the error has been rectified.

Errors when writing vital-sign messages

A write error which occurs when a sharer accesses the watchdog file of the shared pubset decreases the reconfiguration capability of the network. A master change cannot take place and, with an XCS pubset, leave or fail reconfigurations are blocked. Apart from the reconfiguration problem, a write error has no further consequences per se. However, a simultaneous communication failure with the partner may result in erroneous crash detection and may necessitate the implementation of emergency measures. This depends on the BS2000 version running on the partner processor.

Sharers are subject to partner-related failure monitoring. If an error occurs when writing to the watchdog file of a shared pubset, monitoring is only affected if the error occurs with all pubsets shared with a partner.

In this case, if the MSCF connection has already failed or fails shortly afterwards, an erroneous crash detection is imminent by the partner due to the temporal relationship and the RECOVERY-START settings. The system may be terminated by MSCF to avoid the erroneous crash detection and message MCS1300 will be output.

Write errors are indicated by message DMS03B9. Rectification is announced by message DMS03BB.

The partner is notified of the occurrence and rectification of a write error by message MCA0110.



When leveling a dual global storage area, temporary write errors may occur in the watchdog file for shared GS volumes. These errors will be rectified automatically once the leveling procedure is completed.

Errors when reading vital-sign messages

If a read error prevents a sharer from reading the vital-sign messages entered in the watchdog file by the partners, the sharer is no longer able to detect the failure of these partners. If a partner is considered no longer active by configuration management due to a read error and the loss of an MSCF connection, the partner may have crashed. Systems support is then requested to make a decision (message MCS1100), regardless of the RECOVERY-START setting.

The read error is indicated by message DMS03B2, and the rectification of the error is indicated by message DMS03B8. If a partner which was suspected of having crashed is identified as being active again, the pending failure inquiry is cancelled.

7.19 Failure of the watchdog file

A defective watchdog file blocks the monitoring and reconfiguration of the shared pubset network. For this reason, the file must be deleted by systems support so that a new file can be created by the system. It is only possible to delete the file if the pubset is exported from all sharers and is then imported exclusively on one processor. The watchdog file (`$TSOS.SYS.PVS.SHARER.CONTROL` with SF pubsets; `$TSOS.SYS.PVS.SHARER.CONTROL.<control-volume-set-id>` with SM pubsets) can now be deleted. The pubset must then be exported again from the processor. Finally, the shared pubset network can be reestablished. A new watchdog file is automatically created during the master import.

7.20 Disk monitoring blocks

If disk monitoring is not in control again within a specific period of time (FAIL-DETECTION-LIMIT / 11 seconds), it cannot write the vital-sign messages in time. This has the same effect as a write error on all shared pubsets. If a connection is also lost at this time, the partner may incorrectly diagnose a crash (for information on how to prevent an erroneous crash detection, see [section “Input/output errors when accessing the watchdog file” on page 181](#)).

A temporary write error is rectified when new vital-sign messages are written successfully.

The disk monitoring blocks are indicated by message DMS03B7.

If a connection is lost at the same time as a disk monitoring block exists, MSCF may initiate an abnormal system termination (SETS); message MCS1300 is then output (see [section “System termination by MSCF” on page 186](#)).

Disk monitoring blocks may be caused by excessively long wait states, for example due to:

- extreme overloading
(a paging bottleneck does not have any effect, however, because vital-sign messages are written by a resident routine),
- performing a CPU reconfiguration

If these situations cannot be excluded, it is advisable to use the general or partner-specific setting of RECOVERY-START=*CONSISTENT-BY-OPERATOR - possibly temporarily (see [“Inhibiting the automatic start of fail reconfiguration” on page 125](#)).

7.21 Cluster Recovery Lock (CRL)

Under certain special conditions of the operating system (e.g. during CPU reconfiguration) the operating system is halted for a certain time. To prevent the risk of errored failure detection by MSFC, the system component responsible for the special condition can reserve the Cluster Recovery Lock (CRL) during the critical period. While the CRL is reserved, a fail reconfiguration by MSCF is started automatically if the failure was detected or confirmed by the Live Monitor.

Current users of the CRL are the VM2000 command MOVE-VM and, on the SX 150 model series, dynamic reconfiguration. SNAPSHOT does not use the CRL, but has its own connection to disk monitoring.

Normally, the CRL is released after the special conditions have finished. If this does not happen (e.g. because of an error), automatic error handling would be blocked if it were not for the Live Monitor. In this case, the CRL can be released manually with the RELEASE-CLUSTER-RECOVERY-LOCK command (see [page 214](#)).

7.22 Abnormal termination of HIPLEX MSCF

HIPLEX MSCF terminates automatically in the event of irrecoverable errors. The entire HIPLEX MSCF functionality on the processor is cleared down and the processor is disconnected from an XCS network. All imported shared pubsets are exported. The MSCF connections are cleared down and the MSCF subsystem is terminated.

Normally the processor can release all its shared resources. It is possible, however, that tasks might hold up the release of XCS locks (can be displayed via the SHOW-XCS-OCCUPATION command) or the export of shared pubsets (can be displayed via the command SHOW-PUBSET-PARAMETER <catid>,INFORMATION=*PUBSET-OCCUPATION). But MSCF cannot be unloaded until these tasks have released their shared resources. For this reason it might be necessary to terminate the tasks.

A "global crash" occurs if the processor cannot be disconnected from the network due to an error in a registered function or in configuration management. In this case, MSCF can no longer manage the global resources on the processor or cannot terminate access to the global resources, which is why the system must be terminated. If the general configuration parameter RECOVERY-START is set to *AUTOMATIC, an abnormal system termination is initiated (message MCS1301 is output); if *BY-OPERATOR or *CONSISTENT-BY-OPERATOR is set, systems support is requested to shut down the system (message MCS1102 is output).

7.23 Disconnection of a processor from the XCS network

If the XCS configuration contains errors, a coordinated exit with a leave reconfiguration is not possible. If HIPLEX MSCF is terminated in this state, the processor is removed from the XCS network by disconnecting itself from the network. The registered functions are terminated in succession. When the XCS functionality has been cleared down in full, the processor disconnects from the partners. The partners then perform a fail reconfiguration, whereby the relevant processor is removed from the XCS configuration and the global resources are recovered.

7.24 Abnormal termination of an MSCF system task

The MSCF subsystem uses the system tasks MSCF, MSCC, MSCM, MXCM, WDGS and WDWT the MSCF server tasks, and a variable number of auxiliary tasks for temporary jobs. The tasks are displayed by issuing the SHOW-SUBSYSTEM-STATUS command (assuming MSCF is the DSSM holder task).

- *Abnormal termination of the WDGS task*
If the WDGS task is terminated abnormally due to a program error, all shared pubsets are exported with a watchdog forced export in the task error recovery procedure. This export process essentially corresponds to an implicit FORCE-PUBSET-EXPORT command, with which the occupying tasks are terminated immediately with CANCEL-JOB and then the disks are removed from the system. The export process is concluded when all tasks have canceled their shared pubset occupation. After being exported, the pubset can therefore be imported again in the same BS2000 session. In addition, the registered function CPM (plus XPM in XCS mode) reports the termination to configuration management. MSCF is terminated as a result.
- *Abnormal termination of the MSCF task*
If the MSCF task is terminated abnormally due to a program error, MSCF clears down the connections to the partner processors in the DSSM task recovery; the processor is disconnected from the network and MSCF terminates.
- *Abnormal termination of the MXCM task*
If the MXCM task is terminated abnormally due to a program error, XCM disconnects the processor from the network in the error recovery and initiates the termination of MSCF.
- *Abnormal termination of an MSCF server task*
A new MSCF server task is created if necessary.

7.25 Termination of MSCF by a registered function

In the event of irrecoverable errors, configuration management allows the registered functions to request that the processor be disconnected from the network. This termination is indicated by message MCS1008.

7.26 System termination by MSCF

Abnormal system termination (SETS) is initiated by MSCF in four defined situations. The cases indicated by messages MCS1300 and MCS1301 are, however, "normal" terminations, i.e. system terminations permitted by the defined operating parameters. Only message MCS1302 indicates an error that must be diagnosed.

- **MCS1300 SYSTEM TERMINATION TO PREVENT ERRONEOUS CRASH DETECTION BY PARTNERS**
If the processor is no longer able to send vital-sign messages to a partner due to the loss of the MSCF connection and the simultaneous occurrence of a write error on the watchdog files of all shared pubsets, and if there is a danger (due to time conditions and the RECOVERY-START settings) that a fail reconfiguration will be initiated by a partner, a SETS is initiated with MCS1300 in order to prevent data inconsistency on the global resources. See also [“Controlling allowed system aborts” on page 127](#).
- **MCS1301 SYSTEM TERMINATION TO ALLOW THE RECONFIGURATION AT THE MSCF-PARTNERS**
If MSCF terminates on a processor and the network functionality used by the processor cannot be released due to an error in configuration management or in the registered functions, it is not possible for the remaining processors to reestablish the XCS network. When operating with the general RECOVERY-START=*AUTOMATIC setting, a SETS is therefore initiated with MCS1301 to enable the reconfiguration of the XCS network on the partner processors.
- **MCS1302 SYSTEM TERMINATION BECAUSE OF IRRECOVERABLE ERROR IN CONFIGURATION MANAGEMENT (failed error recovery)**
If the MXCM task terminates abnormally, a task error recovery is initiated in which MSCF is terminated on the processor. If a program error occurs again, a SETS is initiated with MCS1302 because the consistency of the network can no longer be guaranteed.
- **MCS1304 SYSTEM TERMINATION BECAUSE OF EXPIRATION OF ABORT-LIMIT TIME**
If a network-participation abort for a shared pubset network or XCS network cannot be completed within the interval specified by ABORT-LIMIT, a SETS is initiated with MCS1304.

7.27 Timeout of reconfiguration

XCM signals configuration changes to the registered functions and controls their execution. No upper time limit can be defined for the duration of a reconfiguration, which can depend on many factors. However, if the time required for the reconfiguration exceeds the FAILURE-DETECTION-LIMIT, a warning is output in message MCS1006 and systems support is requested to check the function cited in the message.

7.28 Hanging shared pubset occupation

When HIPLEX MSCF is terminated, all shared pubsets are always exported. However, it is possible that pubsets are still occupied by the DMS and the export is therefore blocked. For this reason, message DMS039B indicates which pubsets are still occupied on which sharers. Message MCS4122 indicates which pubsets have not yet been exported. Message MCS1006 indicates whether the registered function CPM or XPM is preventing the termination of the XCS reconfiguration.

Systems support can issue the following command to determine the blocking tasks, and can eliminate the cause of the block:

```
/SHOW-PUBSET-PARAMETERS <catid>, INFORMATION=*PUBSET-OCCUPATION
```

If EXPORT-PUBSET TERMINATE-JOBS=*YES is specified UTM applications receive an end signal when they use the pubset.

The POSIX file systems /root and /var should not be placed on shared pubsets, otherwise POSIX must be terminated manually so that the affected pubsets can be exported outside of a shutdown. This could lead to blocks, especially when fail reconfiguration of MSCF (RECOVERY-START=*AUTOMATIC / *SECURE) is started automatically, because MSCF cannot be terminated until all shared pubsets have been exported.

7.29 Hanging XCS occupation

When HIPLEX MSCF is terminated, all TU tasks which occupy XCS functionality are canceled (CANCEL); system tasks are terminated by terminating the registered functions. The MSCF subsystem does not terminate until there is no further occupation of XCS functionality (the MSCF subsystem can then be restarted if necessary).

If the CANCEL job of a task is deferred as a result of higher priority states, the termination of MSCF is blocked. The block is indicated at regular intervals by message MCS1021 on the main console. Systems support can issue the SHOW-XCS-OCCUPATION command to determine the blocking tasks, and can eliminate the cause of the block.

7.30 Error recovery for GS caches in parallel HIPLEX mode

Error recovery in parallel HIPLEX mode is described in detail in [section “Parallel HIPLEX” on page 48](#).

8 Performance and optimization

8.1 Software Monitor openSM2

The openSM2 monitoring system (abbreviated to: SM2) provides the user with statistical data on the performance of the DP system and the utilization of resources. SM2 records measurement values and outputs them to the data display terminal and/or to a file. Output to the data display terminal is suitable for real-time monitoring (online). When the data is output to a file, the stored measurement values can be analyzed at a later stage (offline).

The following explanations are limited to the multiprocessor-specific aspects of SM2. A detailed description of openSM2 can be found in the manual of the same name [14].

In the MSCF network, SM2 provides functions for central online monitoring and control of monitored data acquisition. Since the internal structure of the monitored data depends to a high degree on the software version used, the SM2 network functionality can only be used with identical SM2 versions. Exchanging monitored data between different SM2 versions is not possible.

The monitored data is recorded independently on each processor in the network by a local SM2 performance monitor and is exchanged between the processors via the MSCF connections. In this way, the online reports of all processors can be output on any processor in the network.

There are also reports which combine all the measurement values of the various processors.

All control statements for monitored data acquisition (change of monitoring cycle, switch of monitoring programs, etc.) can be entered on each processor for all processors in the network.

8.1.1 Prerequisites for implementing SM2 in the MSCF network

- An MSCF connection of type CCS (see [page 27](#)) must exist between all processors in the network (SM2 does not distinguish between a CCS, shared pubset, or XCS network).
- SM2 is already started on all processors in the network (e.g. with START-SM2) or has at least been loaded with the DSSM command /START-SUBSYSTEM SUBSYSTEM-NAME=SM2.
- The SM2 functions for monitored data acquisition in a network can only be used for those processors in the network that run the same SM2 version.

8.1.2 Users in the MSCF network

SM2 distinguishes between privileged users ("primary" administrators, "secondary" administrators and other privileged users) and nonprivileged users in the MSCF network. However, no distinction is made between the rights of a local and a remote user. A user can have different rights on different processors, depending on the privileges of this user and the rights already assigned. In particular, the primary administrator of a processor can originate from another processor. Such primary administrators can be identified on the MEASUREMENT-STATUS screen by the additional specification of the host name. A limited set of administration instructions is available depending on the rights of the user. The set of permitted instructions can therefore differ on the various processors. Any instructions which are not permitted on the individual processors will be rejected on those processors. If several processors are selected, the instructions of the first processor specified are displayed in the dialog procedure.

8.1.3 Screen output

The screens displaying the local data of a processor are output in succession, followed by the screens of the next processor, etc. The host name is specified in the top left-hand corner of the screen to indicate which processor is involved.

Screens with summarized network data can be output before the local screens of the individual processors. Such screens include the following reports:

- The GLOBAL report provides an overview of the utilization of all processors in the network.
- The NSM report outputs the NSM subsystem data for all processors in the network.
- The SHARED-PVS report indicates the accesses to shared disk devices.
- The CMS report in the multiprocessor system indicates the access values originating from the tasks of the system which manages the catalog (LOCAL), as well as the accesses of tasks running on remote processors. These access values have the identifier REMOTE.

In a shared pubset network, accesses by slave processors are also indicated on the master processor (identifier SHARED).



The GLOBAL, NSM and SHARED-PVS reports can only take account of the processors previously selected with the SM2 instruction SELECT-HOSTS.

8.1.4 Standard SM2 output file

The standard SM2 output file is created on each processor on the home pubset. To allow the SM2 output files of the various processors to be distinguished later, the file name also contains the host name.

8.1.5 Restrictions

- The BS2000 command START-TASK-MEASUREMENT can only be used on the local processor.
- It is not possible to page back with "-" or "R" beyond processor boundaries, i.e. you cannot page from the first report of a processor to the last report of the previous processor. It is only possible to switch to the first report of the preceding processor using the paging instruction "-N" (node).
The reports containing the combined measurement values can likewise only be reached with "-N" when paging backward.
The first selected report is always output with the paging instruction "--".

8.1.6 Monitoring programs and reports

SM2 offers a range of mostly switchable monitoring programs, which record additional monitored data. Some of the important monitoring programs in multiprocessor operation are listed below:

Monitoring programs

- *DAB (Disk Access Buffer)*
provides data on read and write accesses and the associated hit rates for files buffered via the cache.
- *DLM (Distributed Lock Manager)*
provides data on lock requirements from TU and TPR.
- *GS (Global Storage)*
provides data on read and write accesses to the global storage, divided according to partitions.
- *GSVOL*
provides data on read and write accesses to volumes emulated in the global storage.
- *MSCF*
provides data on the communication between the local processor and other processors.
- *NSM (Node Synchronization Manager)*
monitoring program provides data on this basic function of HIPLEX in the XCS network.

Reports

- *GLOBAL*
provides an overview of the utilization (CPU utilization, I/O rate, memory utilization) of all processors in the network.
- *SHARED-PVS*
shows the accesses and queue lengths for shared disk devices.

8.2 Performance aspects for shared pubsets

Disks on shared pubsets are accessed directly by all sharers via the input/output path. The possible performance enhancements in data accesses are described in [section “Performant File Access \(PFA\) for shared pubsets” on page 194](#) and [section “Emulation of shared pubsets in global storage \(GSVOL\)” on page 200](#).

DMS metaoperations (e.g. Open, Close, Allocation) on the shared pubsets can be processed more effectively with the following measures:

- Start SCA on the master processor of the shared pubset. The indicator START-SPEEDCAT in the MRSCAT entry should also be set on the slave processors to ensure that SCA is also started in the event of a master change.
- In a shared pubset network containing processors with different levels of performance, it must be ensured that the DMS management requests of all sharers can be processed efficiently by the master processor. This requirement should also be fulfilled after a master change (see [section “Generation of the shared pubset network” on page 131](#)).
- If there is a permanent and intensive DMS metaoperation load, the processor on which the greatest load of DMS metaoperations is expected should be used as the master processor, in order to avoid superfluous communication.
- The MSCF connections should be established via the most reliable and fastest communication paths (BCAM routes).

8.3 Performant File Access (PFA) for shared pubsets

The aim of the BS2000 HIPERFILE concept is to implement high-performance file processing and thereby increase the performance of the entire system (PFA concept). The basis of the HIPERFILE concept is the utilization of particularly fast storage media as buffers (or caches) in order to avoid input/output bottlenecks in file processing and in particular for BS2000 applications. As a direct result, response time behavior is improved in transaction and dialog applications, batch job processing is accelerated, and the performance of the entire system is enhanced.

The HIPERFILE concept is described in the manual "Introductory Guide to Systems Support" [5].

Shared pubsets can also be supported by the cache media main memory (all BS2000/OSD server) and global storage (S servers) which are included in the HIPERFILE strategy.

Cache media: main memory and global storage

The cache media main memory and global storage (GS) can be used for caching shared pubsets. These cache media offer significantly improved response time and throughput values. In particular, a two-stage caching with DAB in main memory and global storage is possible.

Each pubset sharer manages a separate cache area (or part of one) itself. Files are buffered in the relevant cache area of a pubset sharer if they are opened locally there. Files which are processed system-wide in shared update mode (shared file system) are not buffered.

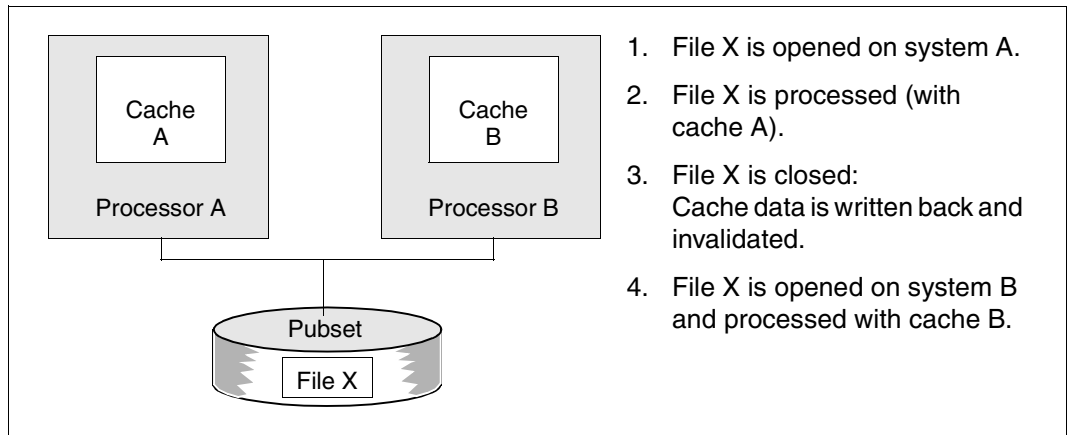


Figure 18: Concept of support for shared pubsets by DAB

The cache areas on the relevant pubset sharers can either reside in a local medium (main memory, local GS) or in a globally used GS (= parallel HIPLEX).

The cache areas are configured via the relevant MRSCAT entry of the shared pubset (MODIFY-PUBSET-CACHE-ATTRIBUTES command). With SF pubsets the cache parameters can be set separately for each pubset sharer; with SM pubsets the settings made apply to all participants in the shared pubset network. These cache areas are created on the relevant pubset sharer during import and are removed again on export. The cache medium used must be homogeneous in the shared pubset network, i.e. it is not possible to buffer on one pubset sharer with main memory and on another with (local) GS. This is checked by the system when the cache areas are created. Conditions of homogeneity also apply to the BS2000 versions participating in the shared pubset network, to the GS unit and to the GS use (local/global) with GS cache areas.

Points to note when using the individual cache media are explained below.

Caching in main memory

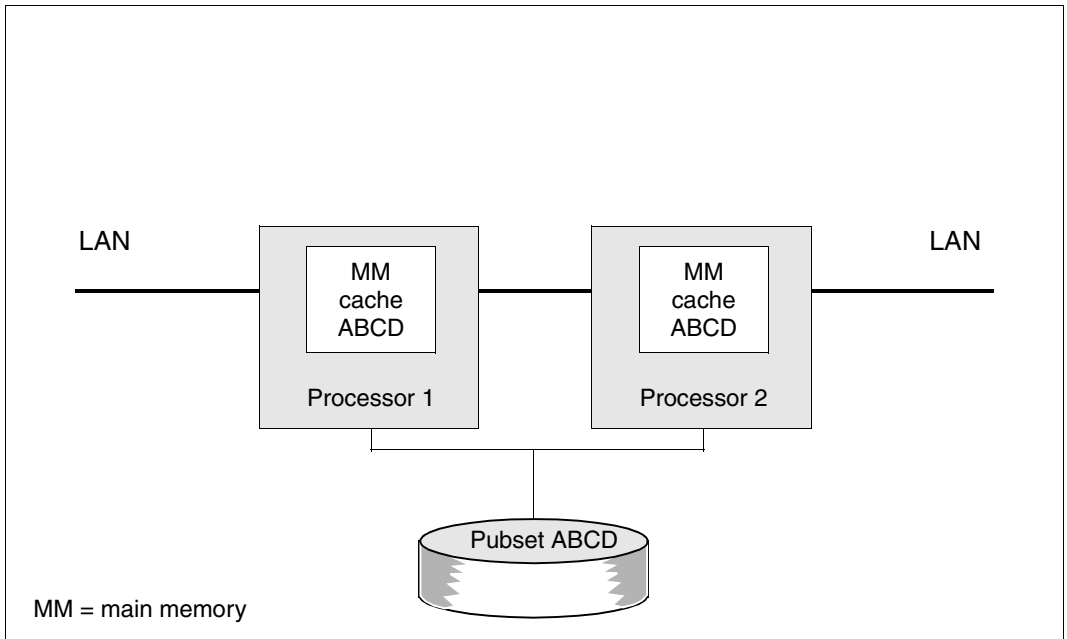


Figure 19: Shared pubset network with main memory caching

The medium main memory is primarily suitable for read caching, as there is no crash resistance against a processor crash. DMS also sets this mode for all permanent files by default. Write caching is only performed for temporary files or if the user has explicitly given the file the attribute DISK-WRITE=*BY-CLOSE. A processor crash is handled here in the same way as with exclusively imported pubsets. The cache data of the crashed processor is lost, but this is not a problem because it is exclusively data from unclosed temporary files which is deleted automatically anyway.

Caching in global storage

Battery-backed, the global storage is a nonvolatile medium and therefore suitable for use as write cache. DAB supports both local and global use of the global storage, with the following distinctions:

Caching in the locally used GS

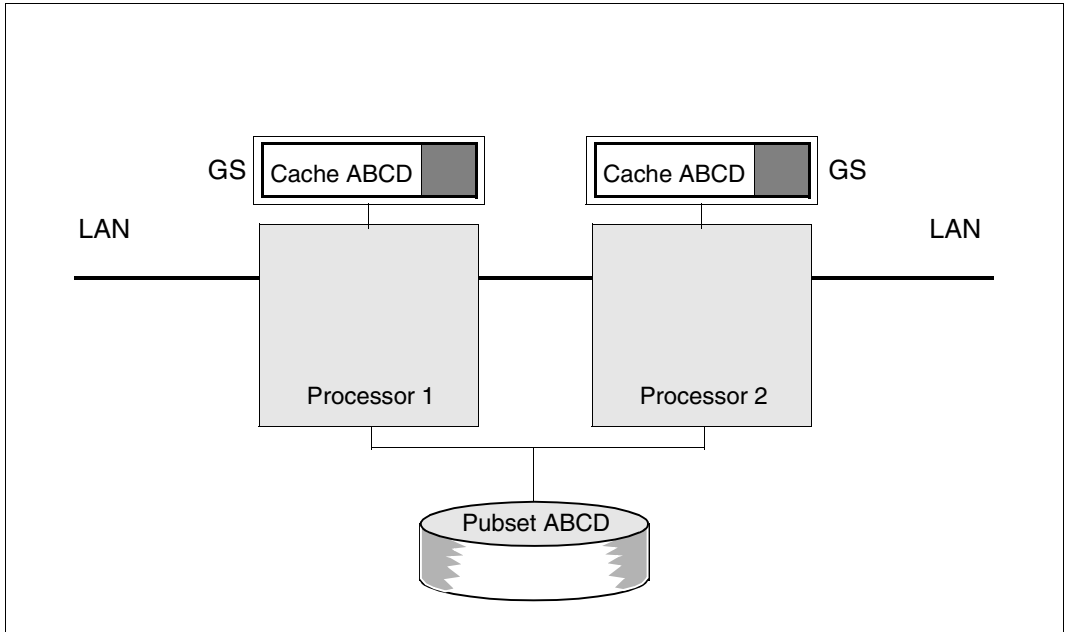


Figure 20: Shared pubset network with caching in local GS

When a local GS configuration is used, at least one pubset sharer has a global storage (equally, several or all pubset sharers can have a global storage). Because each existing global storage is used locally (each processor has its own GS), the processors do **not** form a parallel HIPLEX.

A GS partition for a cache area must be created on every processor by systems support. The cache area on each respective processor is created either when the shared pubset is imported or later when the START-PUBSET-CACHING command is issued. The cache area local to the processor is removed when the pubset is exported or via the STOP-PUBSET-CACHING command.

Note that the use of a local GS configuration is subject to restrictions regarding the reconfiguration if a processor or the connection between processor and buffered disks crashes.

In a local GS configuration, access to the global storage is only possible from one processor. If this processor crashes, the associated cache area cannot be relocated to the pubset master. All files which had not been closed normally when the processor failed and were used for the write caching (open mode=OUTPUT/INOUT/OUTIN) remain locked, as they contain cache data on the crashed processor (any attempt to open these files from another pubset sharer is rejected with the DMS return code 0D1F, and the status is displayed with the SHOW-FILE-LOCKS command). The file lock is not released until the crashed processor is restarted and the pubset has been reimported on the processor. During the re-import the GS cache area is connected to the pubset again, the cache data is written back to the disks and the file locks are reset. The files are then accessible again from all pubset sharers.

Caching in the globally used GS (parallel HIPLEX)

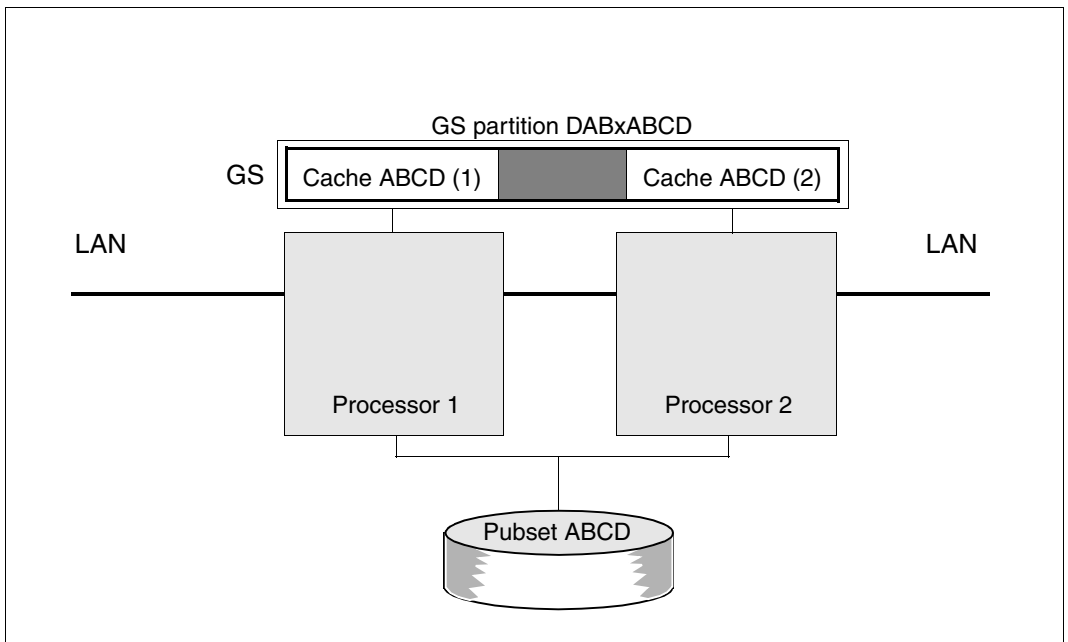


Figure 21: XCS network with caching in the globally used GS (parallel HIPLEX)

In a global GS configuration all pubset sharers are connected to the same global storage and form a parallel HIPLEX. This provides functions which allow central management of the global storage and its use as a shared medium.

For use as a cache medium, systems support must create a GS partition whose size corresponds to the sum of the sizes of all cache areas of the pubset sharers that are to be included (plus 1 Mbyte for DAB management data). In this GS partition, cache areas are set up on the respective pubset sharers. The GS partition is divided internally into segments for the pubset sharers which correspond to the cache size. The GS partition is occupied as “shared”, so all pubset sharers have access to the same GS partition.

If a processor or the connection between processor and buffered disks crashes, in a global GS configuration a cache area of a pubset sharer can be relocated to the pubset master. In the case of a processor crash (slave or master), this occurs as part of the crash detection by MSCF. On the pubset master the data in the relocated cache area is written back to the disks. The file locks, which were preventing the buffered files from being opened from pubset slaves, are then released (on the pubset master they can already be opened once the cache area has been relocated, i.e. after reconfiguration of the shared pubset network by MSCF). The files are then accessible again from all pubset sharers. See also [section “Reconfiguration of DAB cache areas in parallel HIPLEX” on page 57](#).

For further information on the concept of shared pubset caching via main memory or global storage, see the “DAB” manual [2].



The functionality described above for when a processor in the shared pubset network crashes is only available in a parallel HIPLEX. Use of the global storage as a “shared” medium is recommended.

8.4 Emulation of shared pubsets in global storage (GSVOL)

The relocation of an SF pubset or the volume sets of an SM pubset to the global storage greatly increases the efficiency of system and user data accesses to these disks, in the same way as DAB caching in the global storage (see [section “Performant File Access \(PFA\) for shared pubsets” on page 194](#)). Although this method requires a large global storage expansion, it does mean that every file access is processed “quickly” (with caching “only” the cache hits and only exclusive (i.e. not distributed) applications).

Shared pubsets in the global storage (GS volumes) naturally require a global GS (i.e. a parallel HIPLEX).

GSVOL is described in [section “Parallel HIPLEX” on page 48](#) and in detail in the manual "Introductory Guide to Systems Support" [5].

9 Commands

This chapter contains the following:

- an overview of the commands in HIPLEX MSCF described in this manual
- an overview of the commands for managing shared pubsets
- the description of the HIPLEX MSCF commands in alphabetical order

Overview of the commands in HIPLEX MSCF

Command	Function	Page
MODIFY-MSCF-CONNECTION	Modify partner-specific monitoring parameters	203
MODIFY-MSCF-ENVIRONMENT	Modify MSCF configuration parameters	206
MODIFY-NSM-ENVIRONMENT	Modify NSM configuration parameters.	212
RELEASE-CLUSTER-RECOVERY-LOCK	Release the Cluster Recovery Lock (CRL) manually	214
SET-MSCF-ENVIRONMENT	Define MSCF configuration parameters	216
SET-NSM-ENVIRONMENT	Define NSM configuration parameters	222
SET-XCS-PUBSET	Declare a shared pubset an XCS pubset	225
SHOW-MSCF-CONFIGURATION	Retrieve information on the MSCF configuration	227
SHOW-NSM-CONFIGURATION	Display NSM configuration	277
SHOW-SHARED-PUBSET	Request information on shared pubsets	285
SHOW-XCS-OCCUPATION	Output information on user tasks occupying XCS functionality	290
START-MSCF-CONNECTION	Establish MSCF connection to a partner processor	293
START-SUBSYSTEM	Activate MSCF	299
STOP-MSCF-CONNECTION	Clear down MSCF connection to a partner processor	303
STOP-SUBSYSTEM	Deactivate MSCF	305

These commands are only available to the user if the chargeable software product HIPLEX MSCF is in use.

For a detailed description of the SDF syntax please refer to the “Commands” manual [[10](#)].

Commands for managing shared pubsets

The table below lists the commands for managing shared pubsets. These commands are described in detail in the “Commands” manual [10].

Command	Function	User
ADD-MASTER-CATALOG-ENTRY	Create an MRSCAT entry	System management
MODIFY-MASTER-CATALOG-ENTRY	Modify an MRSCAT entry	System management
REMOVE-MASTER-CATALOG-ENTRY	Delete an MRSCAT entry	System management
SHOW-MASTER-CATALOG-ENTRY	Retrieve information on MRSCAT entries ¹	Systems support, nonprivileged user
CANCEL-PUBSET-EXPORT	Cancel the export of a pubset	Systems support
CANCEL-PUBSET-IMPORT	Cancel the import of a pubset	Systems support
EXPORT-PUBSET	Export a pubset	Systems support
FORCE-PUBSET-EXPORT	Force the export of a pubset	Systems support
IMPORT-PUBSET	Import a pubset	Systems support
SET-PUBSET-ATTRIBUTES	Define characteristics of a pubset	Systems support
SHOW-PUBSET-ATTRIBUTES	Output characteristics of a pubset	Systems support
SHOW-PUBSET-IMPORT-EXPORT	Output pubset processing status	Systems support
SHOW-PUBSET-PARAMETERS	Retrieve information on MRSCAT entries	Systems support
SHOW-SHARED-PUBSET	Retrieve information on shared pubsets	Systems support, nonprivileged user

Table 3: Commands for managing the MRSCAT

¹ The function "retrieve information on MRSCAT entries" is also available as the STAMCE macro (see the manual "Executive Macros" [12])

MODIFY-MSCF-CONNECTION

Modify partner-specific monitoring parameters

Description status:	MSCF V17.0A
Functional area:	Pubset and MRSCAT management
Domain:	MULTI-CATALOG-AND-PUBSET-MGMT
Privileges:	TSOS OPERATING
Routing code:	R

Function

The MODIFY-MSCF-CONNECTION command is used to modify partner-specific monitoring parameters:

- Number of monitoring connections
- Defines how recovery measures are to be initiated

This command is evaluated only if a CCS connection to the partner exists. Use the START-MSCF-CONNECTION command to modify the monitoring parameters of an existing connection request or of an interrupted connection.

Format

MODIFY-MSCF-CONNECTION

PROCESSOR-NAME = <alphanum-name 1..8>

,**NUMBER-OF-CTRL-CONN** = ***UNCHANGED** / 1 / 2

,**RECOVERY-START** = ***UNCHANGED** / ***STD** / ***AUTOMATIC** / ***BY-OPERATOR** /
***CONSISTENT-BY-OPERATOR** / ***SECURE**

Operands

PROCESSOR-NAME = <alphanum-name 1..8>

Name of the partner whose monitoring parameters are to be modified.

NUMBER-OF-CTRL-CONN = *UNCHANGED / 1 / 2

Number of independent monitoring connections through which the partner is to be monitored.

NUMBER-OF-CTRL-CONN = *UNCHANGED

The number of monitoring connections is not changed.

NUMBER-OF-CTRL-CONN = 1

The partner is to be monitored using one monitoring connection.

NUMBER-OF-CTRL-CONN = 2

The partner is to be monitored using two monitoring connection. If they fail concurrently, a partner failure can be detected automatically even without a shared pubset.

Note

Independent monitoring connections are only available if *physically independent* communication paths were configured between the same-name MSCF applications \$MRSAPP4 and \$MRSAPP5 on the local system and on the partner system using BCMAP commands. MSCF cannot verify this prerequisite, i.e. NUMBER-OF-CTRL-CONN=2 confirms that the specified communication paths are independent. An incorrect entry can lead to erroneous failure detection and a loss of data integrity.

RECOVERY-START = *UNCHANGED / *STD / *AUTOMATIC / *BY-OPERATOR / *CONSISTENT-BY-OPERATOR / *SECURE

Defines partner-specifically when the partner is monitored (see the section [“Activating and deactivating MSCF partner monitoring” on page 124](#)) and how the system is to respond in the event of the failure of the partner system (= partner-specific RECOVERY-START setting).

Note

The fail reconfiguration can only be started automatically if partner monitoring is certain that the partner has failed and this is permitted by both the general *and* the partner-specific RECOVERY-START settings on the local system and on the partner system (see the section [“Inhibiting the automatic start of fail reconfiguration” on page 125](#)).

RECOVERY-START = *UNCHANGED

The RECOVERY-START setting in not changed.

RECOVERY-START = *STD

Monitoring of the partner begins only when both the MSCF connection has been established and at least one joint shared pubset has been imported.

The general setting for the RECOVERY-START parameter determines the start time of fail reconfiguration.

Irrespective of this, fail configuration can never be initiated automatically if the partner is monitored only via two independent monitoring connections.

RECOVERY-START = *AUTOMATIC

Partner monitoring begins with the establishment of the MSCF connection.

The local system can automatically initiate fail reconfiguration for the partner even if it is only monitored via two independent monitoring connections (see NUMBER-OF-CTRL-CONN operand).

RECOVERY-START = *BY-OPERATOR

Partner monitoring begins with the establishment of the MSCF connection.

The local system does not initiate fail reconfiguration for the partner unless confirmed by systems support.

RECOVERY-START = *CONSISTENT-BY-OPERATOR

Same behavior as on RECOVERY-START=*BY-OPERATOR. In addition, the partner system does not initiate fail reconfiguration for the local system unless confirmed by the Live Monitor or by systems support.

RECOVERY-START = *SECURE

Partner monitoring begins with the establishment of the MSCF connection.

The local system only starts fail reconfiguration automatically for this partner if failure of the partner has been detected or confirmed by the Live Monitor.

The partner only starts fail reconfiguration automatically for the local system if failure has been detected or confirmed by the Live Monitor.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without error
	1	CMD2201	Parameter error
	2	CMD2202	MSCF subsystem not defined
	3	CMD2203	Installation error
	32	MCS0025	Internal error (bourse call, request memory)
	64	CMD0216	User is not privileged
	64	MCS0009	Local password invalid or local system specified as partner
	65	CMD2241	MSCF subsystem is not loaded
	65	MCS0003	MSCF subsystem currently being loaded
	65	MCS0004	MSCF subsystem currently being unloaded
	66	CMD2242	No connection to the MSCF subsystem
	128	CMD2280	MSCF subsystem temporarily not available
	129	CMD2281	MSCF subsystem unavailable for an extended period
	130	CMD2282	MSCF subsystem unavailable for an undefined period

MODIFY-MSCF-ENVIRONMENT

Modify HIPLEX MSCF environment

Description status:	MSCF V17.0A
Functional area:	Pubset and MRSCAT management
Domain:	MULTI-CATALOG-AND-PUBSET-MGMT
Privileges:	TSOS OPERATING
Routing code:	R

Function

With MODIFY-MSCF-ENVIRONMENT, settings defined for the MSCF session in the MSCF configuration file using SET-MSCF-ENVIRONMENT can be modified dynamically. The following settings can be changed:

- The recovery measures to be initiated in the case of a failure
- Specification whether MSCF traces are to be generated and the file to which they are to be written
- Value for limiting MSCF server tasks
- How much time is available to the user tasks to close themselves down when terminating participation in an XCS network
- Interval after which abort processing will be performed for an incomplete exit from the XCS network (i.e. interval after which a blocked leave reconfiguration is to be converted into an abort)
- Time after which an abort which is not concluded properly leads to the system being closed down abnormally
- Priority of a system in an automatic recovery following the loss of a connection in an XCS network
- Period of time after which fail reconfiguration is started following failure detection by the Live Monitor
- Specification whether a user ID is notified by email, and if so which user ID this is, when particular errors and critical situations occur

The MSCF configuration parameters are described in the [section “Parameterization of MSCF” on page 111](#). For details on the Live Monitor see the “HIPLEX AF” manual [6].

Format

MODIFY-MSCF-ENVIRONMENT
<pre> RECOVERY-START = <u>*UNCHANGED</u> / *AUTOMATIC / *BY-OPERATOR / *CONSISTENT-BY-OPERATOR / *SECURE ,TRACE-FILE = <u>*UNCHANGED</u> / *NONE / *STD / <partial-filename 2..31> ,SERVER-TASK-LIMIT = <u>*UNCHANGED</u> / *STD / <integer 10..500> ,USER-TERM-LIMIT = <u>*UNCHANGED</u> / *STD / *UNLIMITED / <integer 0..6000 seconds> ,LEAVE-LIMIT = <u>*UNCHANGED</u> / *UNLIMITED / <integer 1..6000 seconds> ,ABORT-LIMIT = <u>*UNCHANGED</u> / *UNLIMITED / <integer 0..6000 seconds> ,HOST-PRIORITY = <u>*UNCHANGED</u> / *STD / <integer 1..32> ,FADING-INTERVAL = <u>*UNCHANGED</u> / *STD / <integer 0..300 seconds> ,NOTIFY-BY-MAIL = <u>*UNCHANGED</u> / *NO / *YES(...) *YES(...) USER-ID = <u>TSOS</u> / <name 1..8> </pre>

Operands

RECOVERY-START = *UNCHANGED / *AUTOMATIC / *BY-OPERATOR / *CONSISTENT-BY-OPERATOR / *SECURE

Defines how the system is to respond to the failure of a partner system, i.e. how the recovery mechanisms are to be started. The setting applies to all partners (= RECOVERY-START default value).

Note

- The setting of RECOVERY-START cannot be modified while a system component holds the Cluster Recovery Lock (CRL) (see also the RELEASE-CLUSTER-RECOVERY-LOCK command).
- The fail reconfiguration can only be started automatically if partner monitoring is certain that the partner has failed and this is permitted by both the general *and* the partner-specific RECOVERY-START settings on the local system and on the partner system (see the section [“Inhibiting the automatic start of fail reconfiguration” on page 125](#)).

RECOVERY-START = *UNCHANGED

The current setting remains unchanged.

RECOVERY-START = *AUTOMATIC

The local system can initiate recovery measures automatically.

RECOVERY-START = *BY-OPERATOR

The recovery measures are not to be initiated until the action has been confirmed by systems support.

RECOVERY-START = *CONSISTENT-BY-OPERATOR

The same behavior as on RECOVERY-START=*BY-OPERATOR. In addition, the partner system does not initiate fail reconfiguration for the local system unless confirmed by the Live Monitor or by systems support.

RECOVERY-START = *SECURE

The local system only starts fail reconfiguration automatically if failure of the partner has been detected or confirmed by the Live Monitor. Likewise partner systems only initiate fail reconfiguration for the local system automatically if failure has been detected or confirmed by the Live Monitor.

TRACE-FILE = *UNCHANGED / *NONE / *STD / <partial-filename 2..31>

Governs whether the traces generated by the MSCF subsystem are to be logged in a file and if so, which file.

TRACE-FILE = *UNCHANGED

The current setting remains unchanged.

TRACE-FILE = *NONE

MSCF traces are not logged in a file. If a trace file is open, it will be closed.

TRACE-FILE = *STD

The MSCF traces are to be logged in a file with the default name

SYS.MSCF-TRACE.<datum>.<sysid>.<session>.<index>.

Key:

<date>	Date in yyyy-mm-dd format
<sysid>	External representation of the system ID of the local system (range A ... Z, 0 ... 9, 65 ... 192)
<session>	Number of the current BS2000 session
<index>	Number of the MSCF trace file in the current BS2000 session (range 001...999).

Example: SYS.MSCF-TRACE.2009-01-25.A.001.002

TRACE-FILE = <partial-filename 2..31>

Defines the name of the file to which the MSCF traces are to be written. The file name is composed as follows:

<partial-filename 2..31><date>.<sysid>.<session>.<index>
(for the meaning of <date>, <sysid>, <session> and <index>
see the operand TRACE-FILE=*STD).

Example: SYSTEM=TRACES.MSCF.2009-01-26.150.010.003

SERVER-TASK-LIMIT = *UNCHANGED / *STD / <integer 10..500>

Number of MSCF server tasks. This number defines the maximum number of tasks accepted on the partner system.

SERVER-TASK-LIMIT = *UNCHANGED

The server task limit is not changed.

SERVER-TASK-LIMIT = *STD

The server task limit is set to 20.

SERVER-TASK-LIMIT = <integer 10..500>

The server task limit is set to the specified number.

USER-TERM-LIMIT = *UNCHANGED / *STD / *UNLIMITED / <integer 0..6000 seconds>

Time available to a user task that is using the XCS functionality, during which it can terminate, when MSCF is being unloaded. After this interval by the latest, MSCF termination is continued with the termination of the registered function.

USER-TERM-LIMIT = *UNCHANGED

The time remaining for a user task to be terminated after the shutdown of XCS has been initialized remains unchanged.

USER-TERM-LIMIT = *STD

The time available for terminating user tasks using global locks is to be set to the default value (300 seconds).

USER-TERM-LIMIT = *UNLIMITED

The system waits with terminating the registered functions until the XCS functionality used on the local system is released, i.e. the tasks canceled with CANCEL-JOB have released their global locks.

USER-TERM-LIMIT = <integer 0..6000 seconds>

Specifies the length of time in seconds available to the user tasks occupying XCS functionality to terminate while MSCF is unloading.

LEAVE-LIMIT = *UNCHANGED / *UNLIMITED / <integer 1..6000 seconds>

Specifies the period of time within which the computer has to conclude a LEAVE reconfiguration. If the LEAVE reconfiguration cannot be concluded within the given time, ABORT reconfiguration is performed instead (see [page 122](#)).

LEAVE-LIMIT = *UNCHANGED

The period of time available to the computer for performing a LEAVE reconfiguration remains unchanged.

LEAVE-LIMIT = *UNLIMITED

No time monitoring is performed, i.e. there is no time limit on a LEAVE reconfiguration.

LEAVE-LIMIT = <integer 1..6000 seconds>

Defines the period of time to be available to the computer for performing a LEAVE reconfiguration.

ABORT-LIMIT = *UNCHANGED / *UNLIMITED / <integer 0..6000 seconds>

Specifies the length of time within which the system must have completed an abort process. When this time expires, the system's exit from the network is forced by an abnormal system termination (SETS) with MCS1304.

ABORT-LIMIT = *UNCHANGED

The period of time available to the computer to perform an ABORT operation remains unchanged.

ABORT-LIMIT = *UNLIMITED

No time monitoring is performed, i.e. there is no time limit on an ABORT operation.

ABORT-LIMIT = <integer 0..6000 seconds>

Defines the period of time in Sekunden to be available to the computer for performing an ABORT operation.

HOST-PRIORITY = *UNCHANGED / *STD / <integer 1..32>

In an XCS network, this defines a system's priority for automatic recovery after a connection breakdown. The system to which the *higher* value was assigned is removed from the XCS network with an ABORT. If the systems have the same value assigned to them, then the system that entered the network last will be removed.

The default value *STD corresponds to the value 16 (see [page 121](#)).

HOST-PRIORITY is only defined for XCS systems. The attempt to change the priority of a CCS system is rejected with message MCS1029.

FADING-INTERVAL = *UNCHANGED / *STD / <integer 0..300 seconds>

Determines the wait time in seconds which the Live Monitor waits after failure detection before fail reconfiguration is started. The specification *STD corresponds to a wait time of 5 seconds.

This period is used to wait for open user tasks on shared pubsets to be terminated. This enables user outputs from a failed system to be completed before reconfiguration starts.

NOTIFY-BY-MAIL = *UNCHANGED / *NO / *YES(...)

Specifies whether, in addition to a message being output on the console, notification is sent by email when the following problems occur:

- In the event of unexpected loss of an MSCF connection
- If there is a danger of a change of master not succeeding

NOTIFY-BY-MAIL = *NO

No notification is to be provided by email.

NOTIFY-BY-MAIL = *YES(...)

Notifications are to be sent by email to a user ID. When an email is sent, the email address is taken from the EMAIL-ADDRESS field of the user entry concerned (see the MAIL-FILE command).

USER-ID = TSQS / <name 1..8>

Receiver's user ID. The default is TSQS.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
	1	CMD2201	Parameter error
	2	CMD2202	MSCF subsystem not defined
	3	CMD2203	Installation error
	32	MCS0025	Internal error (bourse call, request memory)
	64	CMD0216	User is not privileged
	65	CMD2241	MSCF subsystem is not loaded
	65	MCS0003	MSCF subsystem currently being loaded
	65	MCS0004	MSCF subsystem currently being unloaded
	66	CMD2242	No connection to the MSCF subsystem
	128	CMD2280	MSCF subsystem temporarily unavailable
	129	CMD2281	MSCF subsystem unavailable for an extended period
	130	CMD2282	MSCF subsystem unavailable for an undefined period

MODIFY-NSM-ENVIRONMENT

Modify NSM configuration parameters

Description status:	NSM V17.0A
Functional area:	Pubset and MRSCAT management
Domain:	MULTI-CATALOG-AND-PUBSET-MGMT
Privileges:	TSOS OPERATING SW-MONITOR-ADMINISTRATION
Routing code:	R

Function

The NSM configuration parameters can be modified dynamically in an XCS network using the MODIFY-NSM-ENVIRONMENT command.

The possible settings of the TOKEN-DELAY-TIME operand should only be made use of if you have extremely good knowledge of the system.

Format

MODIFY-NSM-ENVIRONMENT

COMMUNICATION = *UNCHANGED / *BY-MSCF / *BY-GS-IF-AVAILABLE

,TOKEN-DELAY-TIME = *UNCHANGED / *BY-PARAMETER(...)

***BY-PARAMETER(...)**

MINIMUM-DELAY-TIME = *UNCHANGED / <integer 0..32767 milliseconds>

MAXIMUM-DELAY-TIME = *UNCHANGED / <integer 1..32767 milliseconds>

Operands

COMMUNICATION = *UNCHANGED / *BY-MSCF / *BY-GS-IF-AVAILABLE

Specifies whether the NMS subsystem communicates exclusively via HIPLEX MSCF or whether the global storage (GS) is also be used (if available).

COMMUNICATION = *BY-MSCF

NSM is to communication solely via HIPLEX MSCF.

COMMUNICATION = *BY-GS-IF-AVAILABLE

NSM is to use the global storage as an additional communication path.

NSM uses a GS partition called "NSM" (if it does not already exist and if possible, the partition will be created). If the partition can be allocated, the global storage will be used as an additional communication medium alongside HIPLEX MSCF. If the partition cannot be allocated (e.g. because there is no shared global storage or there is not sufficient storage space in the global storage area), communication is performed solely via HIPLEX MSCF.

TOKEN-DELAY-TIME = *UNCHANGED / *BY-PARAMETER(...)

Specifies the token delay in milliseconds before the token is forwarded to the next system. This operand can be used to control the network load generated by NSM, should this be necessary. Note that the NSM reaction time may be impaired by choosing too high a value. As a matter of principle, the settings should only be made by a person with good system knowledge.

Settings made with this operand are taken over automatically by all XCS partner systems.

TOKEN-DELAY-TIME = *BY-PARAMETER(...)

The desired cycle time for the token is to be modified. The settings for normal operation and standby operation can be modified using the following operands:

MINIMUM-DELAY-TIME = *UNCHANGED / <integer 0..32767 milliseconds>

Defines the time (in milliseconds) by which the token is delayed in normal operation, before it is passed on to the next computer. The value "0" means that there is to be no delay. The value has to be smaller than the value defined for standby operation.

MAXIMUM-DELAY-TIME = *UNCHANGED / <integer 1..32767 milliseconds>

Defines the time (in milliseconds) by which the token is delayed in standby operation, before it is passed on to the next computer.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
	3	CMD2203	Incorrect syntax file
	32	CMD0221	System error (internal error)
	64	CMD0216	No authorization to issue command
	64	NLX1044	MINIMUM-DELAY-TIME not smaller than MAXIMUM-DELAY-TIME

RELEASE-CLUSTER-RECOVERY-LOCK

Release the Cluster Recovery Lock (CRL) manually

Description status:	MSCF V17.0A
Functional area:	Pubset and MRSCAT management
Domain:	MULTI-CATALOG-AND-PUBSET-MGMT
Privileges:	OPERATING TSOS
Routing code:	R

Function

Systems support can use the RELEASE-CLUSTER-RECOVERY-LOCK command to manually release the Cluster Recovery Lock (CRL).

Under certain special conditions of the operating system (e.g. during CPU reconfiguration) the operating system is halted for a certain time. To prevent the risk of errored failure detection by MSFC, the system component responsible for the special condition can reserve the Cluster Recovery Lock (CRL) during the critical period. While the CRL is reserved, a fail reconfiguration by MSCF is started automatically if the failure was detected or confirmed by the Live Monitor.

Current users of the CRL are the VM2000 command MOVE-VM and, on the SX 150 model series, dynamic reconfiguration.

Reservation and release of the CRL by a system component is indicated by the messages MCS1048 and MCS1049.

Application for manual release of the CRL

Normally the CRL is reset after the special condition has terminated. If this is not done (e.g. because of an error), automatic failure handling without the Live Monitor is blocked. In this case the CRL can be released using a command.

Reservation of the CRL is indicated in the SHOW-MSCF-CONFIGURATION command with RECOVERY-START=LOCKED.

Format

RELEASE-CLUSTER-RECOVERY-LOCK

Return codes

(SC2)	SC1	Maincode	Meaning
1	0	CMD0001	Command executed without error
	0	CMD0001	No action required
	2	CMD2202	MSCF subsystem not defined
	3	CMD2203	Installation error
	64	CMD0216	User is not privileged
	65	CMD2241	MSCF subsystem is not loaded
	65	MCS0003	MSCF subsystem currently being loaded
	65	MCS0004	MSCF subsystem currently being unloaded
	66	CMD2242	No connection to the MSCF subsystem
	128	CMD2280	MSCF subsystem temporarily not available
	129	CMD2281	MSCF subsystem unavailable for an extended period
	130	CMD2282	MSCF subsystem unavailable for an undefined period

SET-MSCF-ENVIRONMENT

Define MSCF environment

Description status:	MSCF V17.0A
Functional area:	Pubset and MRSCAT management
Domain:	MULTI-CATALOG-AND-PUBSET-MGMT
Privileges:	TSOS OPERATING
Routing code:	R

Function

The SET-MSCF-ENVIRONMENT command conveys global system and network data to the MSCF subsystem. The following configuration parameters can be set:

- password for the local system
- connection to an XCS network
- minimum number of MSCF server tasks
- dimension figure for limiting the MSCF server tasks
- maximum time for network error detection
- time available for the termination of user tasks when disconnecting from an XCS network
- how recovery measures are to be initiated
- whether or not MSCF traces are to be written and to which file
- interval after which abort processing will be performed for an incomplete exit from the XCS network (i.e. interval after which a blocked leave reconfiguration is to be converted into an abort)
- interval after which incomplete abort processing triggers abnormal system termination
- priority of a processor for automatic recovery triggering after a loss of connection in XCS.
- period of time after which fail reconfiguration is started following failure detection by the Live Monitor
- specification whether a user ID and which user ID is notified by email when particular errors and critical situations occur

The command is evaluated only as part of an MSCF configuration file. Outside the configuration file the command is rejected with message MCS0032.

Some of the configuration parameters can be modified during an MSCF session with the MODIFY-MSCF-ENVIRONMENT command.

The MSCF configuration parameters are described in [section “Parameterization of MSCF” on page 111](#). For details on the Live Monitor see the “HIPLEX AF” manual [6].

Format

SET-MSCF-ENVIRONMENT
<pre> LOCAL-PASSWORD = *NONE / <c-string 0..8> / <x-string 1..16> ,XCS-NAME = *NONE / *SUSPEND / <alphanum-name 1..8> ,NUMBER-OF-SERVERS = *STD / <integer 2..10> ,SERVER-TASK-LIMIT = *STD / <integer 10..500> ,FAIL-DETECTION-LIMIT = *STD / <integer 176..3300 seconds> ,USER-TERM-LIMIT = *STD / *UNLIMITED / <integer 0..6000 seconds> ,RECOVERY-START = AUTOMATIC / *BY-OPERATOR / CONSISTENT-BY-OPERATOR / *SECURE ,TRACE-FILE = *NONE / *STD / <partial-filename 2..31> ,LEAVE-LIMIT = *UNLIMITED / <integer 1..6000 seconds> ,ABORT-LIMIT = *UNLIMITED / <integer 0..6000 seconds> ,HOST-PRIORITY = *STD / <integer 1..32> ,FADING-INTERVAL = *STD / <integer 0..300 seconds> ,NOTIFY-BY-MAIL = *NO / *YES(...) *YES(...) USER-ID = TSOS / <name 1..8> </pre>

Operands

LOCAL-PASSWORD = ***NONE** / <c-string 0..8> / <x-string 1..16>

Password for the local system. A password defined here must be specified outside the configuration file by the local system and each partner system when CCS connections are established. By default, no password is assigned.

XCS-NAME = ***NONE** / ***SUSPEND** <alphanum-name 1..8>

Specifies whether the system is participate in an XCS network and, if so, which one.

XCS-NAME = *NONE

By default, the system does not participate in any XCS network. MSCF is started in CCS mode.

If the system parameter MCXSPXCS is set to “Y” or “N” then XCS functionality is made available locally. The system cannot join any XCS network during the current BS2000 session. If the system parameter MCXSPXCS is set to “V” then no XCS functionality is made available. However, MSCF can be restarted in XCS mode following the end of the MSCF session.

XCS-NAME = *SUSPEND

The system is not permitted to participate in any XCS network.

MSCF is started in CCS mode. No XCS functionality is made available. However, the system can be started in a new MSCF session in XCS mode within the same BS2000 session. The value *SUSPEND is not permitted if the system parameter MCXSPXCS is set to “N”.

XCS-NAME = <alphanum-name 1..8>

Name of the XCS network in which the system is to participate. MSCF is started in XCS mode. XCS functionality is made available across the various systems.

This value may not be specified if the system parameter MCXSPXCS is set to “N” or if the load operation was performed with the setting MCXSPXCS=“Y” MSCF in the same BS2000 session with XCS-NAME=*NONE.

If one of these prohibited specifications is used, the message MCS1014 is output and MSCF is terminated.

NUMBER-OF-SERVERS = *STD / <integer 2..10>

Number of MSCF server tasks that are to be made available on the local system during an MSCF session.

The default is 4 MSCF server tasks.

Note

Values specified in the START-SUBSYSTEM MSCF command, SUBSYSTEM-PARAMETER=' NUMBER-OF-SERVERS' take precedence.

SERVER-TASK-LIMIT = *STD / <integer 10..500>

Number of server tasks used by MSCF as a yardstick to limit the number of jobs accepted on the job side. The default value is 20.

FAIL-DETECTION-LIMIT = *STD / <integer 176..3300 seconds>

Specifies the failure detection time in an MSCF network (in seconds, multiples of 44). The intervals at which monitoring mechanisms run are defined through this value. The default setting is 176 seconds. The value has to be coordinated with the partner system (see [page 123](#)).

USER-TERM-LIMIT = *STD / *UNLIMITED / <integer 0..6000 seconds>

Time left to the user to terminate jobs once XCS shutdown has been initialized. The default setting is 300 seconds. The default is 300 seconds.

RECOVERY-START = *AUTOMATIC / *BY-OPERATOR / *CONSISTENT-BY-OPERATOR / *SECURE

Defines how the system is to initiate recovery measures in the event of an error (e.g. failure of a system or connection).

Note

The fail reconfiguration can only be started automatically if partner monitoring is certain that the partner has failed and this is permitted by both the general *and* the partner-specific RECOVERY-START settings on the local system and on the partner system (see section [“Inhibiting the automatic start of fail reconfiguration” on page 125](#)).

RECOVERY-START = *AUTOMATIC

The recovery measures are to be initiated automatically.

RECOVERY-START = *BY-OPERATOR

The recovery measures are not to be initiated by the local system until the action is confirmed by systems support.

RECOVERY-START = *CONSISTENT-BY-OPERATOR

The same behavior as on RECOVERY-START=*BY-OPERATOR. In addition, the partner system does not initiate fail reconfiguration for the local system unless confirmed by systems support.

RECOVERY-START = *SECURE

The local system only starts fail reconfiguration automatically if failure of the partner has been detected or confirmed by the Live Monitor. Likewise partner systems only initiate fail reconfiguration for the local system automatically if failure has been detected or confirmed by the Live Monitor.

TRACE-FILE = *NONE / *STD / <partial-filename 2..31>

Governs whether MSCF traces are to be logged in a file and if so, which file.

TRACE-FILE = *NONE

MSCF traces are not to be logged in a file.

TRACE-FILE = *STD

MSCF traces are to be logged in a file with the default name
 SYS.MSCF-TRACE.<date>.<sysid>.<session>.<index>

Key:

<date>	Date in yyyy-mm-dd format
<sysid>	External representation of the system ID of the local system (from the set A ... Z, 0 ... 9 in the range 65 ... 192)
<session>	Number of the current BS2000 session
<index>	Number of the MSCF trace file in the current BS2000 session (range 001 ... 999)

Example: SYS.MSCF-TRACE.2009-01-14.152.030.001

TRACE-FILE = <partial-filename 2..31>

HIPLEX MSCF traces are to be logged in a file named
 <partial-filename 2..31><date>.<sysid>.<session>.<index>
 (for <date>, <sysid>, <session> and <index> see TRACE-FILE=*STD above).

Example: SYSTEM-TRACES.MSCF.2009-01-15.152.030.002

LEAVE-LIMIT = *UNLIMITED / <integer 1..6000 seconds>

Determines the time within which a system must have concluded a LEAVE reconfiguration. If the LEAVE reconfiguration cannot be concluded within this time span, an ABORT reconfiguration is performed. By default, there is no time limit set for a LEAVE reconfiguration. See also [page 122](#).

ABORT-LIMIT = *UNLIMITED / <integer 0..6000 seconds>

Specifies the interval within which the system must have completed an abort process. When this interval expires, the system's exit from the network is forced by an abnormal system termination (SETS) with MCS1304. By default no time limit is set for the system for its abort process.

HOST-PRIORITY = *STD / <integer 1..32>

In an XCS network, this defines a system's priority for automatic recovery after a connection breakdown. The system to which the *higher* value was assigned is removed from the XCS network with an ABORT. If the systems have the same value assigned to them, then the system that entered the network last will be removed. The default value *STD corresponds to the value 16. See also [page 121](#).

FADING-INTERVAL = *STD / <integer 0..300 seconds>

Determines the wait time in seconds which the Live Monitor waits before fail reconfiguration is started. The default *STD corresponds to a wait time of 5 seconds.

This period is used to wait for open user tasks on shared pubsets to be terminated. This enables user outputs from a failed system to be completed before reconfiguration starts.

NOTIFY-BY-MAIL = *UNCHANGED / *NO / *YES(...)

Specifies whether a user should also be notified by email when the following problems occur:

- In the event of unexpected loss of an MSCF connection
- If there is a danger of a change of master not succeeding

NOTIFY-BY-MAIL = *NO

No notification is to be provided by email.

NOTIFY-BY-MAIL = *YES(...)

Notifications are to be sent by email to a user ID. When an email is sent, the email address is taken from the EMAIL-ADDRESS field of the user entry concerned (see the MAIL-FILE command).

USER-ID = TSQS / <name 1..8>

Receiver's user ID. The default is TSQS.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
	1	CMD2201	Parameter error
	1	MCA0032	Command only permitted within an MSCF configuration file
	2	CMD2202	MSCF subsystem not defined
	3	CMD2203	Installation error
	32	MCS0025	Internal error (bourse call, request memory)
	64	CMD0216	User is not privileged
	65	CMD2241	MSCF subsystem is not loaded
	65	MCS0003	MSCF subsystem currently being loaded
	65	MCS0004	MSCF subsystem currently being unloaded
	66	CMD2242	No connection to the MSCF subsystem
	128	CMD2280	MSCF subsystem temporarily unavailable
	129	CMD2281	MSCF subsystem unavailable for an extended period
	130	CMD2282	MSCF subsystem unavailable for an undefined period

SET-NSM-ENVIRONMENT

Define NSM configuration parameters

Description status:	NSM V17.0A
Functional area:	Pubset and MRSCAT management
Domain:	MULTI-CATALOG-AND-PUBSET-MGMT
Privileges:	TSOS

Function

The NSM subsystem is supplied with parameters by means of the SET-NSM-ENVIRONMENT command. The command is only relevant in XCS networks, and it is only allowed within the NSM configuration file.

The following configuration parameters can be set:

- communication path (via HIPLEX MSCF or global storage)
- token idling

The settings are made in the SYSSSI.NSM.<version> parameter file of the subsystem for the first system to join the XCS network. All other systems that participate in an XCS network use the settings valid at the time of their entry into the network (see [section “Network entry and exit by a processor” on page 49](#) and the MODIFY-NSM-ENVIRONMENT command on [page 212](#)).

Note

The options offered by the COMMUNICATION-TYPE operand should only be used by persons with good system knowledge.

Format

SET-NSM-ENVIRONMENT
<pre> COMMUNICATION = *BY-GS-IF-AVAILABLE / *BY-MSCF / ,COMMUNICATION-TYPE = *TOKEN-RING(...) *TOKEN-RING(...) MINIMUM-DELAY-TIME = *STD / <integer 0..32767 milliseconds> MAXIMUM-DELAY-TIME = *STD / <integer 1..32767 milliseconds> </pre>

Operands

COMMUNICATION = *BY-GS-IF-AVAILABLE / *BY-MSCF

Governs whether the NSM subsystem is to communicate solely via HIPLEX MSCF or if global storage (GS) may be used as well (if available).

COMMUNICATION = *BY-GS-IF-AVAILABLE

Default setting. NSM shall use the global storage as an additional fast communication medium.

NSM requires a GS partition called “NSM” (ID) to do so. If it does not already exist, the partition is created, if possible. If it is possible to allocate the partition, global storage is used for communication alongside HIPLEX MSCF. If the partition cannot be allocated or if errors occur when it is being accessed (e.g. shared GS is not available or the global storage area is not sufficiently large), messages NLX1015 and NLX1016 are issued. In such cases, communication only uses HIPLEX MSCF.

COMMUNICATION = *BY-MSCF

NSM is to communicate solely via HIPLEX MSCF, even if a global storage unit is available.

COMMUNICATION-TYPE = *TOKEN-RING(...)

Defines the communication method for NSM.

With the “token ring” method, all participants in the XCS network are linked in a virtual ring. The token is passed on from system to system for jobs to be bundled, if this is possible, thus reducing the system load.

MINIMUM-DELAY-TIME = *STD / <integer 0..32767 milliseconds>

Defines the time (in milliseconds) which the token is delayed before being passed on again to the next system. The value “0” means that the token is passed on without any delay. The default setting is *STD, i.e. in normal operation the token is delayed for 15 milliseconds.

Note

The smaller the value that is set for MINIMUM-DELAY-TIME, the greater the network load will be. It is therefore possible to regulate the network load by setting an appropriate value. However, it should be noted that setting too high a value may have a negative effect on the reaction time of NSM. As a matter of principle, these settings should only be made by people with good system knowledge.

MAXIMUM-DELAY-TIME = *STD / <integer 1..32767 milliseconds>

Defines the time (in milliseconds) which the token is idle before being passed on to the next system.

The default setting is *STD, i.e. the token is idle for 100 milliseconds.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
	1	NLX1009	Commando only allowed in the NSM configuration file
	3	CMD2203	Wrong syntax file
	32	CMD0221	System error (internal error)
	64	CMD0216	Caller is not privileged

SET-XCS-PUBSET

Declare shared pubset as the XCS pubset

Description status:	MSCF V17.0A
Functional area:	Pubset and MRSCAT management
Domain:	MULTI-CATALOG-AND-PUBSET-MGMT
Privileges:	TSOS OPERATING
Routing code:	R

Function

The SET-XCS-PUBSET command lets you redefine the specified shared pubset as an XCS pubset online. The command will only work correctly if the shared pubset

- has been imported exclusively from systems belonging to an XCS network
- has been imported from all the systems belonging to the XCS network This XCS network must be globally consistent, i.e. there must be no connection errors and an XCS reconfiguration must not be taking place.
- does not contain any GS volumes

The command can be issued on any system in the XCS network.

Note

An XCS pubset cannot be redefined online as an ordinary shared pubset. This can only be done by exporting and then reimporting the pubset.

For more information on XCS pubsets see [section “XCS network” on page 34](#).

Format

SET-XCS-PUBSET
PUBSET = <cat-id 1..4>

Operands**PUBSET = <cat-id 1..4>**

Defines which shared pubset is to be redefined as an XCS pubset.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without error
	1	CMD0202	Syntax error
	64	MCS4127	Error occurred during processing of the command

Note

An XCS network needs to have at least one XCS pubset. If the last XCS pubset has to be exported, typically due to a pubset failure, no more systems can join the XCS network. This situation can be resolved by importing a shared pubset into all the XCS participants and then declaring this as an XCS pubset by means of the SET-XCS-PUBSET command.

SHOW-MSCF-CONFIGURATION

Show information on HIPLEX MSCF configuration

Description status:	MSCF V17.0A
Functional area:	Pubset and MRSCAT management
Domain:	MULTI-CATALOG-AND-PUBSET-MGMT
Privileges:	TSOS OPERATING STD-PROCESSING
Routing code:	@

Function

The command provides information on the current status of the MSCF network as seen by the local system.

The command supports structured output in S variables (see [page 251](#) and the “Commands” manual [10]).

Privileged functions

Only systems support (TSOS or OPERATING privilege) is able to use INFORMATION=*REGISTERED-FUNCTIONS to request information on the current states of the registered functions on the local system. This information is likewise not included in the output for INFORMATION=*ALL unless the command is issued by systems support.

Format

```

SHOW-MSCF-CONFIGURATION

INFORMATION = *MINIMAL / *SUMMARY / *ALL / *LOCAL-ATTRIBUTES / *REGISTERED-FUNCTIONS /
              *HOSTS(...)

  *HOSTS(...)
    SELECT = *ALL / *BY-HOST-NAME(...) / *BY-PROCESSOR-NAME(...) / *BY-SYSID(...) /
            *BY-PARTNER-TYPE(...)
            *BY-HOST-NAME(...)
              | HOST-NAME = <alphanum-name 1..8>
            *BY-PROCESSOR-NAME(...)
              | PROCESSOR-NAME = <alphanum-name 1..8>
            *BY-SYSID(...)
              | SYSID = <alphanum-name 1..3>
            *BY-PARTNER-TYPE(...)
              | PARTNER-TYPE = *XCS / *CCS / *LCS

,OUTPUT = *SYSOUT / *SYSLST(...)

  *SYSLST(...)
    | SYSLST-NUMBER = *STD / <integer 1..99>

```

Operands

INFORMATION = *MINIMAL / *SUMMARY / *ALL / *LOCAL-ATTRIBUTES / *REGISTERED-FUNCTIONS / *HOSTS(...)

Indicates what information is required.

INFORMATION = *MINIMAL

Default value. An overview of the connection and network data of the systems in HIPLEX MSCF is issued.

INFORMATION = *SUMMARY

Outputs an overview of the connection and network data of the systems known in HIPLEX MSCF. With the privileges TSOS and OPERATING, the status of the registered functions on the local system is also output.

INFORMATION = *ALL

Outputs all the available information on the MSCF network. This comprises:

- the local MSCF configuration parameters and the current status (configuration) of the XCS network to which the local system belongs
- the current states of the registered functions on the local system (output only for users with TSOS or OPERATING privilege)
- the connection and network data of all systems with which there is a HIPLEX MSCF network relationship.

INFORMATION = *LOCAL-ATTRIBUTES

Outputs the local HIPLEX MSCF configuration parameters and the current status (configuration) of the XCS network to which the local system belongs.

INFORMATION = *REGISTERED-FUNCTIONS

Only privileged users (TSOS or OPERATING privilege) can select this value.

Outputs the current states of the registered functions on the local system.

INFORMATION = *HOSTS (...)

Outputs the connection and network data of all systems with which there is a HIPLEX MSCF network relationship. SELECT can be used to restrict the output to specific systems.

SELECT = *ALL / *BY-HOST-NAME / *BY-PROCESSOR-NAME / *BY-SYSID / *BY-PARTNER-TYPE

Selects the systems for which the connection and network data is required.

SELECT = *ALL

Default: outputs the information for all systems with which there is a HIPLEX MSCF network relationship.

SELECT = *BY-HOST-NAME(...)

The systems for which information is required are identified by their host names.

HOST-NAME = <alphanum-name 1. 8>

Host name.

SELECT = *BY-PROCESSOR-NAME(...)

The system for which information is required is identified by its BCAM processor name.

PROCESSOR-NAME = <alphanum-name 1. 8>

BCAM processor name of system.

SELECT = *BY-SYSID(...)

The system for which information is required is identified by its system ID (SYSID).

SYSID = <alphanum-name 1..3>

System ID of system.

SELECT = *BY-PARTNER-TYPE(...)

The systems for which information is required are identified by their partner type.

PARTNER-TYPE = *XCS / *CCS / *LCS

Partner type.

OUTPUT = *SYSOUT / *SYSLST(...)

Specifies where the output information is to be sent.

OUTPUT = *SYSOUT

Default: output is sent to the SYSOUT system file.

OUTPUT = *SYSLST(...)

Output is sent to the SYSLST system file.

SYSLST-NUMBER = *STD / <integer 1..99>

Specifies whether the information is to be sent to the SYSLST system file or to a SYSLST file from the set SYSLST01 through SYSLST99.

The default is *STD, i.e. output is directed to the SYSLST system file.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without error
	0	MCS3000	No information for current selection
	3	CMD2203	Invalid syntax file
	32	CMD0221	Internal error (system error)
	32	CMD2009	Error on structured output in S variables
	32	MCS0025	Error in command processing
	64	CMD0216	Syntax error
	128	MCS0025	Error in command processing

Output format

The exact nature of the output depends on which operand value is selected:

Output with INFORMATION=*LOCAL-ATTRIBUTES

```

SHOW-MSCF-CONFIGURATION                                DATE: yyyy-mm-dd, TIME: hh:mm:ss
-----
LOCAL ATTRIBUTES
-----
HOST NAME                = aaaaaaaaa   HOST START DATE (UTC)   = yyyy-mm-dd
PROCESSOR NAME           = aaaaaaaaa   HOST START TIME (UTC)  = hh:mm:ss
SYSID                    = nnn
SYSTEM VERSION           = Vmm.na
MSCF VERSION             = Vmm.na

SERVER TASK LIMIT        = nnn             FAILURE DETECTION LIMIT = nnnn
NUMBER OF SERVERS:
    CURRENT              = nnn             USER TERM LIMIT        = nnnn
    FREE                 = nnn             LEAVE LIMIT            = nnnn
PROTECTED                = kkk             ABORT LIMIT            = nnnn
FADING INTERVAL         = nnn             RECOVERY START         = kkkkkkkkkkkk
                                REQ RECOVERY START = kkkkkkkkkkkk

XCS NAME                 = aaaaaaaaa   XCS STATE              = kkkkkkkkkkkkkk
XCS CREATION DATE (UTC) = yyyy-mm-dd   XCS CONFIGURATION      = nnnnnnnnnn
XCS CREATION TIME (UTC) = hh:mm:ss     HOST PRIORITY          = nn
-----
SHOW-MSCF-CONFIGURATION                                END OF DISPLAY
    
```

Notes on output

Output is split up into three groups:

- The first group identifies the local system, the BS2000 and MSCF software running on it and the current MSCF session.
- The second group shows the values currently set for the MSCF configuration for the local system.
- The third group identifies the XCS network in which the system is participating and shows the values currently set for the XCS configuration parameters of the local system.

Example

```

/show-mscf-configuration inf=*local-attributes
SHOW-MSCF-CONFIGURATION DATE: 2009-01-27, TIME: 16:40:04
-----
LOCAL ATTRIBUTES
-----
HOST NAME = D017ZE15 HOST START DATE (UTC) = 2009-01-27
PROCESSOR NAME = D017ZE15 HOST START TIME (UTC) = 13:43:53
SYSID = 129
SYSTEM VERSION = V17.0A
MSCF VERSION = V17.0A

SERVER TASK LIMIT = 50 FAILURE DETECTION LIMIT = 352
NUMBER OF SERVERS:
  CURRENT = 4 USER TERM LIMIT = UNDEFINED
  FREE = 4 LEAVE LIMIT = UNLIMITED
  ABORT LIMIT = UNLIMITED
PROTECTED = NO RECOVERY START = AUTOMATIC
FADING INTERVAL = 5 REQ RECOVERY START = AUTOMATIC

XCS NAME = UNDEFINED XCS STATE = NOT ALLOWED
XCS CREATION DATE (UTC) = UNDEFINED XCS CONFIGURATION = UNDEFINED
XCS CREATION TIME (UTC) = UNDEFINED HOST PRIORITY = UNDEFINED
-----
SHOW-MSCF-CONFIGURATION END OF DISPLAY

```

*Output with INFORMATION=*REGISTERED-FUNCTIONS*

```

SHOW-MSCF-CONFIGURATION DATE: yyyy-mm-dd, TIME: hh:mm:ss
-----
REGISTERED FUNCTIONS
-----
NUMBER OF FUNCTIONS = nn
-----
FUNCTION NAME = aaa FUNCTION STATE = kkkkkkkkkk
FUNCTION TYPE = kkk FUNCTION ERROR = kkkkkkkk
FUNCTION ATTRIBUTE = kkkkkkkkkk
-----
FUNCTION NAME = aaa FUNCTION STATE = kkkkkkkkkk
FUNCTION TYPE = kkk FUNCTION ERROR = kkkkkkkk
FUNCTION ATTRIBUTE = kkkkkkkkkk
-----
SHOW-MSCF-CONFIGURATION END OF DISPLAY

```


Example

```

/show-mscf-configuration inf=*registered-functions
SHOW-MSCF-CONFIGURATION DATE: 2009-01-27, TIME: 16:40:42
-----
REGISTERED FUNCTIONS
-----
NUMBER OF FUNCTIONS = 9
-----
FUNCTION NAME      = NSM          FUNCTION STATE = NOT ACTIVE
FUNCTION TYPE      = XCS          FUNCTION ERROR  = NO ERROR
FUNCTION ATTRIBUTE = MANDATORY
-----
FUNCTION NAME      = XTS          FUNCTION STATE = NOT ACTIVE
FUNCTION TYPE      = XCS          FUNCTION ERROR  = NO ERROR
FUNCTION ATTRIBUTE = MANDATORY
-----
FUNCTION NAME      = XPM          FUNCTION STATE = NOT ACTIVE
FUNCTION TYPE      = XCS          FUNCTION ERROR  = NO ERROR
FUNCTION ATTRIBUTE = MANDATORY
-----
FUNCTION NAME      = GSM          FUNCTION STATE = NOT ACTIVE
FUNCTION TYPE      = XCS          FUNCTION ERROR  = NO ERROR
FUNCTION ATTRIBUTE = OPTIONAL
-----
FUNCTION NAME      = NSG          FUNCTION STATE = NOT ACTIVE
FUNCTION TYPE      = XCS          FUNCTION ERROR  = NO ERROR
FUNCTION ATTRIBUTE = OPTIONAL
-----
FUNCTION NAME      = GSV          FUNCTION STATE = NOT ACTIVE
FUNCTION TYPE      = XCS          FUNCTION ERROR  = NO ERROR
FUNCTION ATTRIBUTE = OPTIONAL
-----
FUNCTION NAME      = CPM          FUNCTION STATE = READY
FUNCTION TYPE      = CCS          FUNCTION ERROR  = NO ERROR
FUNCTION ATTRIBUTE = MANDATORY
-----
FUNCTION NAME      = DAB          FUNCTION STATE = NOT ACTIVE
FUNCTION TYPE      = XCS          FUNCTION ERROR  = NO ERROR
FUNCTION ATTRIBUTE = OPTIONAL
-----
FUNCTION NAME      = SHC          FUNCTION STATE = NOT ACTIVE
FUNCTION TYPE      = CCS          FUNCTION ERROR  = NO ERROR
FUNCTION ATTRIBUTE = OPTIONAL
-----
SHOW-MSCF-CONFIGURATION END OF DISPLAY

```

*Output with INFORMATION=*HOSTS*

```

SHOW-MSCF-CONFIGURATION                      DATE: yyyy-mm-dd, TIME: hh:mm:ss
-----
HOSTS
-----
NUMBER OF HOSTS = nn
-----
HOST NAME           = aaaaaaaaa   HOST START DATE (UTC) = yyyy-mm-dd
PROCESSOR NAME     = aaaaaaaaa   HOST START TIME (UTC) = hh:mm:ss
SYSID              = nnn         HOST SITE             = kkkkkk
SYSTEM VERSION     = Vmm.na      PARTNER TYPE         = kkk
MSCF VERSION       = Vmm.na      HOST STATE           = kkkkkkkkkk

CONTROL CONNECTIONS:
  (REQUESTED,IN USE) = (1,m)     CONNECTION TYPE      = kkkkkkkkkk
OVERLOADED:
  (REQUEST,REPLY)   = (k,k)     CONNECTION STATE     = kkkkkkkkkkkk
REQ CONN RECOVERY START = kkkkkkkkkkkk  PROTECTED           = kkk
REQ CONTROL GROUP    = a

CONN RECOVERY START  = kkkkkkkkkkkk CONTROL GROUP       = a
PARTNERS RECOVERY START = kkkkkkkkkkkk CONTROL STATE      = kkkkkkkkkkkk

XCS NAME            = aaaaaaaaa   RECONFIGURATION STATE = kkkkkkkkkkkkkk
XCS CREATION DATE (UTC) = yyyy-mm-dd JOINING ORDER        = nnnnnnnnnn
XCS CREATION TIME (UTC) = hh:mm:ss  HOST PRIORITY        = nn
-----
SHOW-MSCF-CONFIGURATION                      END OF DISPLAY

```

Note on output

Output is split up into four groups:

- The first group identifies the partner system, the BS2000 and MSCF software running on it and its MSCF session, and it shows the network status.
- The second group identifies the status of the MSCF connection to the partner.
- The third group shows the current settings for partner-specific monitoring parameters and the control status of the partner.
- The fourth group shows the current settings for the partner-specific XCS configuration parameters.

Example

```

/show-mscf-configuration inf=*hosts
SHOW-MSCF-CONFIGURATION                                     DATE: 2009-01-27, TIME: 16:41:09
-----
HOSTS
-----
NUMBER OF HOSTS = 3
-----
HOST NAME           = D017ZE15      HOST START DATE (UTC) = 2009-01-27
PROCESSOR NAME     = D017ZE15      HOST START TIME (UTC) = 13:43:53
SYSID              = 129          HOST SITE              = LOCAL
SYSTEM VERSION     = V17.0A      PARTNER TYPE          = CCS
MSCF VERSION       = V17.0A      HOST STATE            = ACTIVE

CONTROL CONNECTIONS:
  (REQUESTED,IN USE) = (-,-)      CONNECTION TYPE        = UNDEFINED
OVERLOADED:
  (REQUEST,REPLY)    = (-,-)      CONNECTION STATE       = LOCAL
REQ CONN RECOVERY START = UNDEFINED  PROTECTED              = NO
REQ CONN RECOVERY START = UNDEFINED  REQ CONTROL GROUP     = UNDEFINED

CONN RECOVERY START = UNDEFINED  CONTROL GROUP         = UNDEFINED
PARTNERS RECOVERY START = UNDEFINED  CONTROL STATE        = UNDEFINED

XCS NAME           = UNDEFINED    RECONFIGURATION STATE = NOT ACTIVE
XCS CREATION DATE (UTC) = UNDEFINED  JOINING ORDER         = UNDEFINED
XCS CREATION TIME (UTC) = UNDEFINED  HOST PRIORITY         = UNDEFINED
-----
HOST NAME           = JUANA1       HOST START DATE (UTC) = 2009-01-21
PROCESSOR NAME     = JUANA1       HOST START TIME (UTC) = 08:58:50
SYSID              = 150          HOST SITE              = REMOTE
SYSTEM VERSION     = V17.0A      PARTNER TYPE          = CCS
MSCF VERSION       = V17.0A      HOST STATE            = ACTIVE

CONTROL CONNECTIONS:
  (REQUESTED,IN USE) = (1,1)      CONNECTION TYPE        = CCS
OVERLOADED:
  (REQUEST,REPLY)    = (N,N)      CONNECTION STATE       = CONNECTED
REQ CONN RECOVERY START = STD          PROTECTED              = NO
REQ CONN RECOVERY START = STD          REQ CONTROL GROUP     = NONE

CONN RECOVERY START = STD          CONTROL GROUP         = NONE
PARTNERS RECOVERY START = CONS BY OP  CONTROL STATE        = ACTIVE

XCS NAME           = DANTEXCS     RECONFIGURATION STATE = NOT ACTIVE
XCS CREATION DATE (UTC) = 2008-11-26  JOINING ORDER         = UNDEFINED
XCS CREATION TIME (UTC) = 13:43:20  HOST PRIORITY         = UNDEFINED
-----
HOST NAME           = JUANA11      HOST START DATE (UTC) = 2009-01-27
PROCESSOR NAME     = JUANA11      HOST START TIME (UTC) = 10:39:37
SYSID              = 115          HOST SITE              = REMOTE
SYSTEM VERSION     = V17.0A      PARTNER TYPE          = CCS
MSCF VERSION       = V17.0A      HOST STATE            = ACTIVE

CONTROL CONNECTIONS:
  (REQUESTED,IN USE) = (1,1)      CONNECTION TYPE        = CCS
OVERLOADED:
  (REQUEST,REPLY)    = (N,N)      CONNECTION STATE       = CONNECTED
REQ CONN RECOVERY START = STD          PROTECTED              = NO
REQ CONN RECOVERY START = STD          REQ CONTROL GROUP     = NONE

CONN RECOVERY START = STD          CONTROL GROUP         = NONE
PARTNERS RECOVERY START = AUTOMATIC  CONTROL STATE        = ACTIVE

XCS NAME           = UNDEFINED    RECONFIGURATION STATE = NOT ACTIVE
XCS CREATION DATE (UTC) = UNDEFINED  JOINING ORDER         = UNDEFINED
XCS CREATION TIME (UTC) = UNDEFINED  HOST PRIORITY         = UNDEFINED
-----
SHOW-MSCF-CONFIGURATION                                     END OF DISPLAY

```

*Output with INFORMATION=*SUMMARY*

```

SHOW-MSCF-CONFIGURATION                                DATE: yyyy-mm-dd, TIME: hh:mm:ss
-----
XCS NAME = aaaaaaaa                                XCS STATE = kkkkkkkkkkkkk
-----
HOSTS:
HOST      PROCESSOR MSCF   SYS CONN  CTRL OVL PART HOST  CONTROL  RECOVERY
NAME      NAME      VERS   ID  STATE  CONN  TYPE STATE  G STATE  START  P
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
aaaaaaa  aaaaaaaa  Vmm.na nnn LOCAL          kkk  kkkkkk          kkkkkk
[        aaaaaaaa          kkkkkk ]
aaaaaaa  aaaaaaaa  Vmm.na nnn kkkkkk 1,m  k,k kkk  kkkkkk k kkkkkkkkk kkkkkk k
[        aaaaaaaa          k          kkkkkk ]
:        :        :        :        :        :        :        :        :        :        :
:        :        :        :        :        :        :        :        :        :        :
aaaaaaa  aaaaaaaa  Vmm.na nnn kkkkkk 1,m  k,k kkk  kkkkkk k kkkkkkkkk kkkkkk k
-----
REGISTERED FUNCTIONS:
NAME TYPE ATTRIBUTE STATE      ERROR
-----+-----+-----+-----+-----
aaa  kkk  kkkkkkkkk kkkkkkkkk kkkkkkkkk
:    :    :        :        :
aaa  kkk  kkkkkkkkk kkkkkkkkk kkkkkkkkk
-----
SHOW-MSCF-CONFIGURATION                                END OF DISPLAY

```

Notes on output

- The first block or the first line of the section “HOSTS” always describes the local system. The information relating to an MSCF partner comprises identification, connection, network and monitoring data.
- The connection is identified by the processor name, i.e. the name under which the partner can be accessed from the system via BCAM. Several connections can exist simultaneously to a partner using different processor names, however, only one MSCF connection can be established between two systems. The systems within the network are identified by the system name (see [section “Processor identification” on page 112](#)).
- The connection is managed by the communication system; the network is managed by configuration management. In the output, the data is combined partner-specifically into logical blocks, provided an MSCF connection exists to the partner. If there is no MSCF connection to the partner, the data is output in two blocks, one containing the connection data and the other containing the network and monitoring data. The connection data is identified through the processor name and the network and monitoring data through the system name.

- During the connection recovery the connection status remains as “connected”, which means that the MSCF connection exists. But communication is not possible because the connection is displayed as overloaded. The non-availability of the connection is displayed in the monitoring status and the reconfiguration status of the partner.

Example

```

/show-mscf-configuration inf=*summary
SHOW-MSCF-CONFIGURATION                               DATE: 2009-01-27, TIME: 16:41:44
-----
XCS NAME = UNDEFINED                                XCS STATE = XCS NOT ALLOWED
-----
HOSTS:
HOST NAME      PROCESSOR     MSCF   SYS  CONN  CTRL  OVL  PART  HOST  CONTROL  RECOVERY
NAME           NAME         VERS  ID   STATE  CONN  STATE  TYPE STATE  G STATE  START  P
-----
D017ZE15      D017ZE15     V17.0A 129  LOCAL
JUANA1        JUANA1       V17.0A 150  CONN   1,1  N,N  CCS  NOT-J   COMM-ACT  STD  C
JUANA11       JUANA11      V17.0A 115  CONN   1,1  N,N  CCS  ACTIVE  ACTIVE     STD  A
-----
REGISTERED FUNCTIONS:
NAME TYPE ATTRIBUTE STATE      ERROR
-----
CPM  CCS  MANDATORY  READY  NO ERROR
SHC  CCS  OPTIONAL   NOT-ACTIVE  NO ERROR
-----
SHOW-MSCF-CONFIGURATION                               END OF DISPLAY
    
```

Key to output

If the SHOW-MSCF-CONFIGURATION command is issued with INFORMATION=*SUMMARY, the names of the output fields and the field values may be output in a different format (shown in parenthesis in the table below).

Output field	Meaning / Information
ABORT LIMIT	Maximum period within which the system must have left the network abnormally within an abort reconfiguration. <ul style="list-style-type: none"> - <nnnn>: integer, 0..6000 seconds - UNLIMITED - UNDEFINED: No information available
CONNECTION STATE (CONN STATE)	Status of the MSCF connection: <ul style="list-style-type: none"> - LOCAL: Local system - CONNECTED (CONN): MSCF connection exists to partner - NOT CONNECTED (N-CONN): No MSCF connection exists to partner, but local system desires connection to partner - UNDEFINED (---): Local system does not desire connection to partner.
CONNECTION TYPE (PART TYPE)	Connection type: <ul style="list-style-type: none"> - LCS: Loosely Coupled System - CCS: Closely Coupled System - UNDEFINED (---): No information available.

Table 4: Output values of the SHOW-MSCF-CONFIGURATION command

(part 1 of 14)

Output field	Meaning / Information
CONN RECOVERY START (RECOVERY START)	<p>Method of starting recovery measures (current value of partner-specific setting). The partner-specific setting is displayed in the block or line of the relevant partner (INFORMATION=*HOSTS/ *SUMMARY). <kkkkkkkkkkk> (<kkkkkkk>):</p> <ul style="list-style-type: none"> - AUTOMATIC (AUTO): The local system may automatically start recovery measures for a partner, even if the partner is only monitored through two independent monitoring connections. The partner may automatically start recovery measures for the local system. - BY OPERATOR (BY-OP): The local system may not initiate recovery measures for a partner until the action is confirmed by systems support. The partner may automatically start recovery measures for the local system. - CONS BY OP (CONSOP): The local system may not start recovery measures for a partner until the action is confirmed by systems support. The partner may not start recovery measures for the local system until the action is confirmed by systems support. - SECURE: The local system starts fail reconfiguration for the partner automatically only if failure of the partner was detected or confirmed by the Live Monitor. Likewise the partner only starts fail reconfiguration for the local system automatically if failure was detected or confirmed by the Live Monitor. - STD: The partner is not monitored after it has imported the same shared subset as the local system. The method of starting recovery measures is defined through the general setting of RECOVERY START (see page 248). - UNDEFINED (---): No information available. <p>A recovery measure can only be started automatically if both the general and the partner-specific setting on the partner and on the local system allow it.</p>
CONTROL CONNECTIONS (CTRL CONN)	<p>Number of monitoring connections in the format <l>,<m>:</p> <ul style="list-style-type: none"> - <l> numeric 1..2 or - Number of monitoring connections requested via START-MSCF-CONNECTION and MODIFY-MSCF-CONNECTION. "-" means that a value was not defined. - <m> numeric 0..2 or - Number of monitoring connections currently used. "-" means that the value was not defined.

Table 4: Output values of the SHOW-MSCF-CONFIGURATION command

(part 2 of 14)

Output field	Meaning / Information
CONTROL GROUP (CONTROL G)	<p>Partner's monitoring group, current value:</p> <ul style="list-style-type: none"> – <n> numeric 1..9 Fail configurations are started concurrently for partners in the same control group. – NONE (): No concurrent fail reconfiguration. – UNDEFINED (-): No information available.
CONTROL STATE	<p>Partner's monitoring status:</p> <ul style="list-style-type: none"> – NOT ACTIVE (NOT-ACT): The partner is not monitored. – COMM ACTIVE (COMM-ACT): The partner is only active via connection monitoring. – DISK ACTIVE (DISK-ACT): The partner is only active via disk monitoring. The connection is interrupted, has failed, has been closed down or does not exist. – !CONNECT: Same as "DISK ACTIVE". Additionally, the message MCS1101 requests the connection to be re-established. – ACTIVE: The partner is active through both monitoring mechanisms (connection and disk monitoring). – LOST: The partner is monitored. Partner activities can neither be detected through connection monitoring nor disk monitoring. The partner has either failed or the connection is disrupted. – !CONNECTLOST (!CONNLOST): Same as "LOST". Additionally, the message MCS1101 requests the connection to be re-established. – ?CRASHED: Same as "LOST". Additionally, the failure query MCS1100 is open. – ?CRASHED: Same as "LOST". Additionally, the failure query MCS1100 is open. – CRITICAL: Due to a write error to the watchdog files of the shared pubsets used by all sharers, the partner can no longer be informed of the local system's activity. If connection monitoring also fails within a critical period an erroneous failure detection through the partner becomes probable. In this case abnormal system termination must be forced using SETS. – CRASHED: The system / systems support has declared the partner to have failed and remove it from the MSCF network. – MSCF TERMED (MSCF-TERM): Systems support has confirmed the MSCF has been terminated on the partner system. The partner is removed from the MSCF network. – UNDEFINED (-):No information available.

Table 4: Output values of the SHOW-MSCF-CONFIGURATION command

(part 3 of 14)

Output field	Meaning / Information
FADING INTERVAL	<p>Wait time between notification of a partner failure by the Live Monitor and the start of fail reconfiguration:</p> <ul style="list-style-type: none"> – <nnn>: Numeric, 0..300 seconds. – UNDEFINED: No information available.
FAILURE DETECTION LIMIT	<p>Maximum crash detection time:</p> <ul style="list-style-type: none"> – <nnnn>: Numeric, 176..3300 seconds – UNDEFINED (---): No information available.
FUNCTION ATTRIBUTE (ATTRIBUTE)	<p>Availability attribute of registered function:</p> <ul style="list-style-type: none"> – OPTIONAL: Registered function need not be available or can terminate during operation. – MANDATORY: Registered function must be available, as otherwise the MSCF network is terminated on the local system and HIPLEX MSCF terminates.
FUNCTION ERROR (ERROR)	<p>Error state of registered function with respect to MSCF network management:</p> <ul style="list-style-type: none"> – NO ERROR: No errors known. – PROTOCOL: Registered function responded incorrectly to a reconfiguration job. – ABORT: Registered function requested termination of MSCF on the system. – CRASHED: Registered function terminated abnormally. Hinders the reconfiguration capability of the other network sharers or a restart of MSCF, depending on its "FUNCTION STATE".
FUNCTION NAME (NAME)	<p>Name of the registered function. The following functions are defined:</p> <ul style="list-style-type: none"> – NSM - node synchronization manager – XTS - XCS time synchronization – XPM - XCS pubset manager – GSM - global store manager – NSG - NSM global store part – GSV - global store volume emulation – CPM - shared pubset manager – DAB - disk access buffer – SHC - symmetrix host component

Table 4: Output values of the SHOW-MSCF-CONFIGURATION command

(part 4 of 14)

Output field	Meaning / Information
FUNCTION STATE (STATE)	<p>Reconfiguration state of the registered function:</p> <ul style="list-style-type: none"> – NOT ACTIVE (NOT-ACTIVE): Function not active (not yet initialized or already terminated). – INITIATED: Function is currently being initialized; network entry is being prepared. – ENABLED: Function has concluded local initialization and is ready to join network. – STARTED: Function has been started; it joins the network with its partner entities. – READY: Function has successfully concluded network entry/reconfiguration; network functionality is available. – IN RECONF (IN-RECONF): Function is being reconfigured; network functionality may be restricted depending on the reconfiguration and the function. – IN TERM (IN-TERM): Function is being terminated; it leaves the network with partner entities and releases occupied network resources (network-wide termination phase). – DISABLED: Function has left the network and is terminating (local termination phase).
FUNCTION TYPE (TYPE)	<p>Type of registered function:</p> <ul style="list-style-type: none"> – CCS: Registered function is a CCS function – XCS: Registered function is an XCS function
HOST NAME	<p>Host name of the system</p> <ul style="list-style-type: none"> – <nnnnnnnn>: HOST NAME is the name assigned to the system at the start of BCAM; exchanged when the MSCF connections are established; uniquely identifies the system in the MSCF network (see page 112). – UNDEFINED: No information available
HOST PRIORITY	<p>Priority of a system in the XCS network with automatic recovery after a loss of connection. The system with the higher value is removed from the XCS network with ABORT. If both systems have the same priority, the system which entered the XCS network last is removed.</p> <ul style="list-style-type: none"> – <nn>: numeric, 1..32 – UNDEFINED: No information available
HOST SITE	<p>Distinguishes the local entry from the entries of the partner</p> <ul style="list-style-type: none"> – LOCAL: Entry of the local system – REMOTE: Entry of a partner system – UNDEFINED: No information available

Table 4: Output values of the SHOW-MSCF-CONFIGURATION command

(part 5 of 14)

Output field	Meaning / Information
HOST START DATE	<p>The time stamp (UTC time) of the MSCF start with start date and start time on the system; used for network status comparison when re-establishing the MSCF connection between two systems.</p> <ul style="list-style-type: none"> – <yyyy-mm-dd>: Start date – UNDEFINED: No information available
HOST START TIME	<p>see “HOST START DATE”</p> <ul style="list-style-type: none"> – <hh:mm:ss>: Start time – UNDEFINED: No information available
HOST STATE	<p>Status of the local system/partner: <nnnnnnnnnn> Status of the local system/partner in relation to the network specified by “PARTNER TYPE”; <i>PARTNER TYPE=*XCS</i> indicates connection to the XCS network; <i>*CCS</i> indicates participation in shared pubset networks of the system.</p> <ul style="list-style-type: none"> – NOT JOINED (NOT-J): <i>PARTNER-TYPE=*XCS</i>: Local system has not joined the XCS network. Partner is not connected to the XCS network of the system. <i>PARTNER-TYPE=*CCS</i>: Local system is not yet or no longer ready to import a shared pubset. Partner does not share a common pubset with the system. – JOIN: <i>PARTNER-TYPE=*XCS</i>: Local system is joining or establishing the XCS network. Partner is added to the network; system is implementing join reconfiguration. <i>PARTNER-TYPE=*CCS</i>: Local system is not yet ready to import shared pubsets. “JOIN” is not defined for CCS partners. – ACTIVE: <i>PARTNER-TYPE=*XCS</i>: Local system is connected to XCS network. Partner is connected to the XCS network of the system. <i>PARTNER-TYPE=*CCS</i>: Local system is ready to import shared pubsets. Partner has imported a shared pubset in common with the local system. – LEAVE: <i>PARTNER-TYPE=*XCS</i>: Local system is leaving the XCS network. Partner is removed from the network; system is implementing leave reconfiguration. <i>PARTNER-TYPE=*CCS</i>: Local system is terminating CCS network; all imported shared pubsets are exported. LEAVE is not defined for the CCS partner.

Table 4: Output values of the SHOW-MSCF-CONFIGURATION command

(part 6 of 14)

Output field	Meaning / Information
HOST STATE (continued)	<ul style="list-style-type: none"> – CRASH: “CRASH” is not defined for the local system . The partner is automatically recognized as crashed. Before starting a fail reconfiguration, the local system waits a certain time in order to let the partner react to an erroneous failure detection by terminating the system. – FAIL: “FAIL” is not defined for the system. <i>PARTNER-TYPE=*XCS:</i> Partner has disconnected from the XCS network; the local system removes partner from XCS network, re-establishes XCS network with remaining sharers and implements fail reconfiguration. <i>PARTNER-TYPE=*CCS:</i> Partner abnormally terminated CCS network with local system; at least one shared pubset network of the system is affected and is re-established. – ABORT: MSCF terminated abnormally on the local system. “ABORT” is not defined for partners. – UNDEFINED (---): No information available.
JOINING ORDER	<p>This number defined the order in which sharers join the XCS network. When a join reconfiguration is performed, the number of the current XCS configuration (see XCS CONFIGURATION, page 249) is incremented by 1 and becomes the JOINING ORDER of the new system in the network.</p> <ul style="list-style-type: none"> – <nnnnnnnnnn>: integer 0..2147483647. – UNDEFINED: No information available.
LEAVE LIMIT	<p>Maximum period within which the local system must have completed a normal network exit (leave reconfiguration). When this period expires, the normal exit from the network is converted into an abnormal exit (ABORT).</p> <ul style="list-style-type: none"> – <nnnn>: numeric, 1..6000 seconds – UNLIMITED (default): A leave does not become an abort. – UNDEFINED: No information available
NUMBER OF FUNCTIONS	<p>Number of functions registered with configuration management for which information is provided.</p> <p><nn>: numeric, 0..16</p>
NUMBER OF HOSTS	<p>Number of systems for which information is provided:</p> <ul style="list-style-type: none"> – <nnn>, numeric, 1..164

Table 4: Output values of the SHOW-MSCF-CONFIGURATION command

(part 7 of 14)

Output field	Meaning / Information
NUMBER OF SERVERS CURRENT FREE	<p>Number of MSCF server tasks: <nnn>:</p> <p>Current number of MSCF server tasks Number of free MSCF server tasks</p> <p>With periodic peak loads, there may be more MSCF server tasks than specified for the SERVER-TASK-LIMIT operand in the SET-MSCF-ENVIRONMENT or MODIFY-MSCF-ENVIRONMENT command.</p>
OVERLOADED (OVL)	<p>Overloaded connections:</p> <ul style="list-style-type: none"> – (Y,Y): Congestion in both directions or connection recovery is active – (Y,N): Congestion for REQUESTs – (N,Y): Congestion for REPLYs – (N,N): No congestion – (-,-): Local system or there is no connection to the partner
PARTNER TYPE (PART TYPE)	<p>Partner type:</p> <ul style="list-style-type: none"> – LCS: Partner is an LCS partner of the local system – CCS: Partner is a CCS partner of the local system – XCS: Partner is an XCS partner of the local system – UNDEFINED: No information available <p>In the case of INFORMATION=*SUMMARY the connection type is output (see CONNECTION TYPE, page 237).</p> <p>For the local system itself, PARTNER TYPE defines the types of network the system can form with partners.</p>
PARTNERS RECOVERY START (RECOVERY START P)	<p>Partner setting with regard to the start method for recovery measures for the local system (combined general and partner-specific setting). The setting is shown in the block or line of the relevant partner.</p> <p>(INFORMATION=*HOSTS/*SUMMARY) <kkkkkkkkkk> (<k>):</p> <ul style="list-style-type: none"> – AUTOMATIC (A): The partner can automatically start recovery measures for the local system. It also allows the local system to start recovery measures automatically for the partner. – BY OPERATOR (O): The partner starts recovery measures for the local system only after the action has been confirmed by systems support. But it allows the local system to automatically start recovery measures for the partner. – CONS BY OP (C): The partner starts recovery measures for the local system only after the action has been confirmed by systems support. It does not allow the local system to automatically start recovery measures for the partner. – LOCKED (L): A partner system component holds the Cluster Recovery Lock Effect as for SECURE. – NO CONTROL (N): The partner does not monitor the local system.

Table 4: Output values of the SHOW-MSCF-CONFIGURATION command

(part 8 of 14)

Output field	Meaning / Information
PARTNERS RECOVERY_START (RECOVERY START_P) (continued)	<ul style="list-style-type: none"> - SECURE (S): The partner starts fail reconfiguration for the local system automatically only if failure was detected or confirmed by the Live Monitor. It permits automatic start of fail reconfiguration by the local system only if failure was detected or confirmed by the Live Monitor. - UNDEFINED (-): No information available.
PROCESSOR_NAME	<ul style="list-style-type: none"> - <nnnnnnnn>: Processor name - UNDEFINED (--): No information available
RECONFIGURATION STATE	<p>Reconfiguration state: Status of a configuration modification or of a pending malfunction in relation to the system or partner from the point of view of the system. RECONFIGURATION_STATE contains states visible throughout the network relating to the corresponding configuration modification in the XCS network. The same procedures also apply in the CCS network. In contrast to the XCS network, modifications in the CC network are not made known to the sharers. The states relevant to the CCS network are listed in the second part of the explanations below (under "<i>If PARTNER-TYPE=*CCS</i>"). <i>If PARTNER-TYPE=*XCS</i></p> <ul style="list-style-type: none"> - NOT ACTIVE: From the point of view of the local system, there is no pending configuration modification for the local system or the partner, and no configuration fault is known. - JOIN REQUEST: The local system wants to join the XCS network or is to add the partner to the network. The system waits for the start of the join reconfiguration. - JOIN START: The local system joins the XCS network or adds the partner to the XCS network. The registered functions are started or reconfigured. The network implements a join reconfiguration. - JOIN END: The local system has joined the XCS network or has added the partner to the network. The system waits for the other sharers to reach this state and then terminates the join reconfiguration. - LEAVE REQUEST: The local system wants to leave the XCS network or is to remove the partner from the network. The XCS functionality used by the applications on the exiting sharer has been released. The system waits for the start of the leave reconfiguration. - LEAVE START: The local system is leaving the XCS network or is removing the partner from the network. The registered functions are being terminated or reconfigured. The network implements a leave reconfiguration.

Table 4: Output values of the SHOW-MSCF-CONFIGURATION command

(part 9 of 14)

Output field	Meaning / Information
RECONFIGURATION STATE (continued)	<ul style="list-style-type: none"> <li data-bbox="458 211 1266 329">– LEAVE END: The local system has left the XCS network or has removed the partner from the network. The system waits for the other sharers to reach this state and then terminates the leave reconfiguration. <li data-bbox="458 329 1266 388">– ABORT REQUEST: The partner is to disconnect from the XCS network due to an error. <li data-bbox="458 388 1266 530">– ABORT START: The local system or partner disconnects from the XCS network and abnormally terminates the registered functions. The system waits for the end of the global termination phase of the registered functions on the disconnecting sharer. <li data-bbox="458 530 1266 707">– ABORT END: The disconnecting system or partner has concluded the global termination of registered functions and is waiting for their local termination. Any system remaining in the network waits until all other remaining sharers in the network have also reached this state and then switches to the FAIL REQUEST reconfiguration state. <li data-bbox="458 707 1266 799">– CONN ACT: The MSCF connection to the partner exists or has been re-established. However, other sharers may have lost the connection to this partner. <li data-bbox="458 799 1266 917">– CONN LOST: The MSCF connection to the partner is faulty, has crashed or has been cleared down. However, disk monitoring still identifies the partner as active. <li data-bbox="458 917 1266 1001">– CONTROL FAIL: Both monitoring paths to the partner have failed. However, at least one other network sharer still identifies the partner as active. <li data-bbox="458 1001 1266 1093">– CONTROL CRASH: Both monitoring paths to the partner have failed. No other network sharer identifies the partner as still active. The partner may have crashed. <li data-bbox="458 1093 1266 1202">– FAIL REQUEST: The partner has abnormally terminated MSCF and is to be removed from the XCS network. The local system waits for the start of the fail reconfiguration. <li data-bbox="458 1202 1266 1320">– FAIL START: The partner has abnormally terminated MSCF and is being removed from the XCS network. The remaining sharers in the network re-establish the network by implementing a fail reconfiguration. <li data-bbox="458 1320 1266 1496">– FAIL END: The partner has abnormally terminated MSCF and has been removed from the XCS network from the local point of view. The system waits until all other sharers remaining in the network have also reached this state and then terminates the fail reconfiguration. Only then is the network re-established.

Table 4: Output values of the SHOW-MSCF-CONFIGURATION command

(part 10 of 14)

Output field	Meaning / Information
RECONFIGURATION STATE (continued)	<ul style="list-style-type: none"> – UNDEFINED: No information available. <i>If PARTNER-TYPE=*CCS</i> – NOT ACTIVE: No configuration modification is pending for the local system or no crash detection or reconfiguration is pending for the partner. – LEAVE START: The local system terminates MSCF. All imported shared pubsets are exported. – ABORT START: The local system aborts MSCF due to an error. All imported shared pubsets are exported. – ABORT END: The local system has abnormally terminated MSCF. All imported shared pubsets have been exported. Disk monitoring is terminated. – CONTROL FAIL: The MSCF connection to the partner has failed or has been cleared down. The partner is no longer identifiable to the local system as active on all shared pubsets. The partner may have crashed. Operator action has been requested. – FAIL REQUEST: The partner has been declared crashed by the local system. The system waits for some time to allow the partner to initiate a SETS if necessary. – FAIL START: The partner has abnormally terminated MSCF. The affected shared pubset networks are re-established. – UNDEFINED: No information available.

Table 4: Output values of the SHOW-MSCF-CONFIGURATION command

(part 11 of 14)

Output field	Meaning / Information
RECOVERY START	<p>Type of recovery measures (general setting):</p> <p>In the case of LOCAL-ATTRIBUTES and output SUMMARY, the information is issued in the line for the local system.</p> <p><kkkkkkkkkk> (<kkkkkk>):</p> <ul style="list-style-type: none"> – AUTOMATIC (AUTO): The local system can automatically start recovery measures for the partner, partners can automatically start recovery measures for the local system. – BY OPERATOR (BY-OP): The local system starts recovery measures for the partner only after the action has been confirmed by systems support. – CONS BY OP (CONSOP): The local system starts recovery measures for the partner only after the action has been confirmed by systems support. Partners can only start recovery measures for the local system only after the action has been confirmed by systems support. – LOCKED: A partner system component holds the Cluster Recovery Lock Effect as for SECURE. – SECURE: The local system starts recovery measures for the partner automatically only if failure of the partner was detected or confirmed by the Live Monitor. Partners may only start recovery measures for the local system automatically if failure was detected or confirmed by the Live Monitor. – UNDEFINED (---): No information available. – A recovery measure can only be started automatically if this is permitted both by the general and the partner-specific settings of the local system and of the partner
REQ CONN RECOVERY START (RECOVERY START)	<p>Method of starting recovery measures. Partner-specific setting last requested with /START-MSCF-CONNECTION or /MODIFY-MSCF-CONNECTION. The requested setting may temporarily differ from the current setting (until the partner has acknowledged the change).</p> <p>In the SUMMARY output, the requested value is only issued in a separate line for the partner if it differs from the current value.</p> <p>See CONN RECOVERY START (page 238) for an explanation of the values.</p>
REQ CONTROL GROUP (CONTROL G)	<p>Partner's control group. Setting last requested with /START-MSCF-CONNECTION.</p> <p>The requested setting may temporarily differ from the current setting (while the connection is inactive). In the SUMMARY output, the requested value is only issued in a separate line for the partner if it differs from the current value.</p> <p>See CONTROL GROUP (page 239) for an explanation of the values.</p>

Table 4: Output values of the SHOW-MSCF-CONFIGURATION command

(part 12 of 14)

Output field	Meaning / Information
REQ RECOVERY START (RECOVERY START)	Method of starting recovery measures. General setting last requested with /START-MSCF-CONNECTION or /MODIFY-MSCF-CONNECTION. The requested setting may temporarily differ from the current setting (until all partners have acknowledged the change). In the SUMMARY output, the requested value is only issued in a separate line for the local system if it differs from the current value. See RECOVERY START (page 248) for an explanation of the values.
SERVER TASK LIMIT	Maximum number of MSCF server tasks: – <nnn>: numeric, 10..500 – UNDEFINED (---): No information available.
SYSID	– <nnn>: System identification of the system – UNDEFINED (---): No information available.
SYSTEM VERSION (VERS)	BS2000 version of the system: – V<mm>.<n><a> or UNDEFINED (---)
USER TERM LIMIT	Time interval available to the tasks using XCS functionality to release the XCS functionality when disconnecting from the network before the registered XCS functions are terminated. – <nnnn>: numeric, 0..6000 seconds – UNLIMITED: The registered XCS functions are not terminated until all tasks using XCS functionality have been terminated. – UNDEFINED: No information available.
XCS CONFIGURATION	Number of the current XCS configuration. A number is assigned to the global status of the XCS network. It identifies the current configuration and is incremented by 1 with each configuration change. – <nnnnnnnnnn>: numeric, 1..2147483647 – UNDEFINED: No information available.
XCS CREATION DATE	The time stamp (UTC time) of the XCS start, comprising the start date and start time, marks the time at which the XCS network was established by the first joining system. Together with XCS NAME, the time stamp uniquely identifies the XCS network. – <yyyy-mm-dd>: Start date – UNDEFINED: No information available.
XCS CREATION TIME	See XCS CREATION DATE – <hh:mm:ss>: Start time – UNDEFINED: No information available.
XCS NAME	Name of the XCS network: – <nnnnnnnn>: alphanumeric, 1..8 characters – UNDEFINED: No information available.

Table 4: Output values of the SHOW-MSCF-CONFIGURATION command

(part 13 of 14)

Output field	Meaning / Information
XCS STATE	<p>State of XCS functionality on the local system:</p> <ul style="list-style-type: none"> <li data-bbox="458 245 1272 357">– NOT ALLOWED (XCS NOT ALLOWED): The system is not generated as an XCS sharer. MSCF only supports an LCS or CCS network; XCS functionality (global resources) is available on the local system. <li data-bbox="458 362 1272 505">– XCS SUSPENDED: The system is generated as an XCS sharer, but it did not join an XCS network with the START-SUBSYSTEM command previously executed. XCS functionality is not available on the system at present. However, MSCF can be restarted in XCS mode. <li data-bbox="458 510 1272 593">– NOT ACTIVE: The system is not yet or no longer connected to the XCS network; XCS functionality is not available on the system at present. <li data-bbox="458 598 1272 710">– ACTIVE: The system is connected to the XCS network; complete XCS functionality is available. An XCS reconfiguration is not implemented; no fault is known. <li data-bbox="458 715 1272 828">– INTERRUPTED: XCS network is interrupted due to a fault (cause of fault indicated in "RECONF STATE"). XCS functionality is available on the system to a limited extent, depending on the type of fault. <li data-bbox="458 833 1272 976">– RECONF JOIN: XCS network is expanding, a join reconfiguration of the XCS functionality is being implemented (system joining network/adding partner to network). XCS functionality not yet available when system joining, but available in full on the system if a partner is being added. <li data-bbox="458 981 1272 1147">– RECONF LEAVE: XCS network is reducing, a leave reconfiguration of the XCS functionality is being implemented (system leaving the network/partner being removed from the network). XCS functionality no longer available when system leaving the network, but available in full on the system if a partner is being removed from the network. <li data-bbox="458 1152 1272 1359">– RECONF FAIL: One or more partners have disconnected from the XCS network; the XCS network is re-established by the partner by removing the crashed partners from the XCS network. A fail reconfiguration of the XCS functionality is implemented; depending on the type of error that occurred, functionality is only available in full on the system after the reconfiguration is completed.

Table 4: Output values of the SHOW-MSCF-CONFIGURATION command

(part 14 of 14)

Output in S variables

The INFORMATION operand of the command determines which S variables are to be created. INFORMATION accepts the following values:

Notation used in command	Condition column
INFORMATION = *ALL	1
INFORMATION = *ALL and var(*LIST).ALL.HOSTS.NUM-HOSTS > 0	2
INFORMATION = *ALL and var(*LIST).ALL.REG-FUNC.NUM-FUNC > 0	3
INFORMATION = *HOSTS ¹	4
INFORMATION = *LOCAL-ATTRIBUTES	5
INFORMATION = *REGISTERED-FUNCTIONS ^{1 2}	6
INFORMATION = *SUMMARY	7
INFORMATION = *SUMMARY and var(*LIST).SUMMARY.HOSTS.NUM-HOSTS > 0	8
INFORMATION = *SUMMARY and var(*LIST).SUMMARY.REG-FUNC.NUM-FUNC > 0 ²	9
INFORMATION = *MINIMAL	10

¹ The variables for INFORMATION = *HOST or *REGISTERED-FUNCTION (indicated by condition 4 or 6) are only created if information on at least one element exists (i.e. at least one system or one registered function was selected). If no element exists, the maincode MCS3000 is returned in the command return code.

² The variables that output information on registered functions (indicated by condition 6 or 9) are only displayed if the user has operator privileges.

Output information	Name of the S variable	T	Contents	Condition
<p>Start of partner monitoring and method of starting recovery measures (current value of partner-specific setting). A recovery can only be started automatically if this is permitted by both the general and the partner-specific settings of the local system and the partner.</p> <p>*AUTO= partner monitoring begins with the establishment of the MSCF connection. The local system can automatically start recovery measures for a partner, even if the partner is only monitored via two independent monitoring paths.</p> <p>*BY-OPER= partner monitoring begins with the establishment of the MSCF connection. The local system only starts recovery measures for a partner after the action has been confirmed by systems support.</p>	var(*LIST).ALL.HOSTS.HOST(*LIST).CONN-RECOV-START	S	*AUTO *BY-OPER *CONSISTENT *SECURE *STD *UNDEFINED	2

(part 1 of 25)

Output information	Name of the S variable	T	Contents	Condition
<p>*CONSISTENT= the local system only starts recovery measures for a partner after the action has been confirmed by systems support. The partner may start recovery measures for the local system only after the action has been confirmed by the Live Monitor or systems support.</p> <p>*SECURE=partner monitoring begins with the establishment of the MSCF connection. The local system only starts recovery measures for the partner automatically if failure of the partner has been detected or confirmed by the Live Monitor. Likewise the partner only starts recovery measures for the local system automatically if the failure has been detected or confirmed by the Live Monitor.</p> <p>*STD= partner monitoring begins only when both the MSCF connection has been established and at least one joint shared pubset has been imported. The method for starting recovery measures is defined by the general setting of "RECOVERY-START".</p> <p>*UNDEFINED= no information available.</p>	var(*LIST).ALL.HOSTS.HOST(*LIST). CONN-RECOV-START (continued)	S	*AUTO *BY-OPER *CONSISTENT *SECURE *STD *UNDEFINED	2
<p>Status of the MSCF connection:</p> <p>*MSCF-CONN=MSCF connection exists to partner.</p> <p>*MSCF-LOCAL=local system.</p> <p>*NOT-MSCF-CONN=no MSCF connection to partner, but local system wants connection to partner.</p> <p>*UNDEFINED=local system does not want connection to partner.</p>	var(*LIST).ALL.HOSTS.HOST(*LIST). CONN-STA	S	*MSCF-CONN *MSCF-LOCAL *NOT-MSCF-CONN *UNDEFINED	2
<p>Connection type:</p> <p>*CCS=closely coupled system</p> <p>*LCS=loosely coupled system</p> <p>*UNDEFINED=no information</p>	var(*LIST).ALL.HOSTS.HOST(*LIST). CONN-TYPE	S	*CCS *LCS *UNDEFINED	2
<p>Number of monitoring connections currently used.</p>	var(*LIST).ALL.HOSTS.HOST(*LIST). CONTR-CONN	S	<integer 0..2> *UNDEFINED	2
<p>Current value of the partner's control group. For partners in the same control group, fail reconfigurations are started concurrently.</p>	var(*LIST).ALL.HOSTS.HOST(*LIST). CONTROL-GROUP	S	<integer 1..9> *NONE *UNDEFINED	2

(part 2 of 25)

Output information	Name of the S variable	T	Contents	Condition
<p>Monitoring status of partner:</p> <p>*ACTIVE=system is monitored by connection monitoring and disk monitoring.</p> <p>*COMM-ACTIVE=partner only controlled by connection monitoring.</p> <p>CONNECT-LOST-REQ=same as *LOST. Additionally, MCS1101 requests the connection to be re-established.</p> <p>*CONNECT-REQ=same as *DISK-ACTIVE. Additionally, MCS1101 requests the connection to be re-established.</p> <p>*CRASHED=partner was declared crashed by system/systems support and is removed from MSCF network.</p> <p>CRASHED-QUEST=same as *LOST. Additionally, the failure query MSC1100 is open.</p> <p>*CRITICAL=due to write error on the watchdog files of all shared pubsets, the activity of the system can no longer be signaled to the partner.</p> <p>*DISK-ACTIVE=partner is only controlled by disk monitoring; connection is faulty, has crashed, has been cleared down or does not exist.</p> <p>*LOST=activity of the partner cannot be determined either by connection monitoring or disk monitoring. The partner has either crashed or a fault has occurred.</p> <p>*MSCF-TERMINATED=systems support has confirmed that MSCF was terminated on the partner. The partner is removed from the MSCF network.</p> <p>*NOT-ACTIVE=partner is not monitored.</p> <p>*UNDEFINED=no information</p>	var(*LIST).ALL.HOSTS.HOST(*LIST).CONTROL-STATE	S	*ACTIVE *COMM-ACTIVE *CONNECT-LOST-REQ *CONNECT-REQ *CRASHED *CRASHED-QUEST *CRITICAL *DISK-ACTIVE *LOST *MSCF-TERMINATED *NOT-ACTIVE *UNDEFINED	2
Host name of the system	var(*LIST).ALL.HOSTS.HOST(*LIST).HOST-NAME	S	<c-string 1..8> *UNDEFINED	2
Priority of the partner in an XCS network in the case of an automatic recovery start following the loss of a connection. The system with the higher value is removed from the network with ABORT. If both systems have the same priority, the system to have joined the XCS network later is removed from the XCS network.	var(*LIST).ALL.HOSTS.HOST(*LIST).HOST-PRIO	S	<integer 0..32> (0=UNDEFINED)	2

(part 3 of 25)

Output information	Name of the S variable	T	Contents	Condition
Distinguishes the local entry from the entries of the partners: *LOCAL= entry of the local system *REMOTE= entry of a partner system *UNDEFINED= no information	var(*LIST).ALL.HOSTS.HOST(*LIST).HOST-SITE	S	*LOCAL *REMOTE *UNDEFINED	2
Time stamp (UTC time) of the MSCF start on the system	var(*LIST).ALL.HOSTS.HOST(*LIST).HOST-START-TIME	S	<yyyy-mm-dd/ hh:mm:ss> *UNDEFINED	2
Status of the system/partner with respect to the network with the system specified by "PARTNER-TYPE" (<i>P-T</i>); <i>P-T</i> =*XCS indicates connection of the system to the XCS network, <i>P-T</i> =*CCS indicates participation in shared pubset networks of the system: *ABORT= (only for system) MSCF terminates abnormally on the local system. *ACTIVE= <i>P-T</i> =*XCS: System participates in XCS network. Partner participates in XCS network of the system. <i>P-T</i> =*CCS: System is ready to import shared pubsets. Partner uses the same shared pubset as the system. *CRASH: (only for partner) The partner was automatically recognized as crashed. Before beginning the fail reconfiguration the system waits for a certain time to give the partner the opportunity of reacting to an erroneous failure detection by terminating the system. *FAIL= (only for partner) <i>P-T</i> =*XCS: Partner has disconnected from XCS network; system starts a fail reconfiguration. <i>P-T</i> =*CCS: Partner has abnormally terminated CCS network with system; at least one shared pubset network of the system is affected and is re-established.	var(*LIST).ALL.HOSTS.HOST(*LIST).HOST-STATE	S	*ABORT *ACTIVE *CRASH *FAIL *JOIN *LEAVE *NOT-JOINED *UNDEFINED	2

(part 4 of 25)

Output information	Name of the S variable	T	Contents	Condition
<p>*JOIN= <i>P-T=*XCS:</i> System joins or establishes XCS network. Partner is added to the network; system implements join reconfiguration.</p> <p>*LEAVE= <i>P-T=*XCS:</i> System leaves XCS network. Partner is removed from network; system implements leave reconfiguration. <i>P-T=*CCS</i> (only for local system): System terminates CCS network; all imported shared pubsets are exported.</p> <p>*NOT-JOINED= <i>P-T=*XCS:</i> System is not connected to XCS network. Partner is not connected to XCS network of the system. <i>P-T=*CCS:</i> Partner is not ready to import shared pubsets. Partner does not have a shared pubset with the system.</p> <p>*UNDEFINED=no information</p>	var(*LIST).ALL.HOSTS.HOST(*LIST). HOST-STATE (continued)	S	*ABORT *ACTIVE *FAIL *JOIN *LEAVE *NOT-JOINED *UNDEFINED	2
Order of entry into XCS network.	var(*LIST).ALL.HOSTS.HOST(*LIST). JOINING-ORDER	I	<integer 0..2147483647> (0=UNDEFINED)	2
MSCF version running on the system with release status (format 'Vmm.na')	var(*LIST).ALL.HOSTS.HOST(*LIST). MSCF-VERSION	S	<c-string 1..6> *UNDEFINED	2
Overloaded connections: --=local system or there is no connection to the partner NN=no congestion NY=congestion for REPLYs YN=congestion for REQUESTs YY=congestion in both directions or connection recovery is active	var(*LIST).ALL.HOSTS.HOST(*LIST). OVERLOAD-STA	S	-- NN NY YN YY	2
Partner type: *CCS=partner is CCS partner *LCS=partner is LCS partner *UNDEFINED=no information *XCS=partner is XCS partner For the local system itself, the network types the system can form with partners are indicated.	var(*LIST).ALL.HOSTS.HOST(*LIST). PARTNER-TYPE	S	*CCS *LCS *UNDEFINED *XCS	2

(part 5 of 25)

Output information	Name of the S variable	T	Contents	Condition
<p>Recovery start setting for the partner (combined general and partner-specific setting with regard to local system).</p> <p>*AUTO= The partner can automatically start recovery measures for the local system if that is allowed by the system. It also allows the local system to automatically start recovery measures for a partner.</p> <p>*BY-OPER= The partner starts recovery measures for the local system after the action is confirmed by systems support. But it does allow the local system to automatically start recovery procedures for the partner.</p> <p>*CONSISTENT= The partner starts recovery measures for the local system after the action is confirmed by systems support. It does not allow the local system to automatically start recovery procedures for the partner.</p> <p>*LOCKED= A system component of the partner holds the Cluster Recovery Lock. Acts like *SECURE.</p> <p>*SECURE= The partner only starts recovery measures for the local system automatically if a failure has been detected or confirmed by the Live Monitor. The partner only permits the recovery measures to be started automatically by the local system if the failure has been detected or confirmed by the Live Monitor.</p> <p>*NO-CONTROL= The local system is not monitored by the partner.</p> <p>*UNDEFINED= no information.</p> <p>*UNKNOWN= *AUTO or *BY-OPER (only for partners with MSCF versions < V14.0A).</p>	var(*LIST).ALL.HOSTS.HOST(*LIST).PART-RECOV-START	S	*AUTO *BY-OPER *CONSISTENT *LOCKED *SECURE *NO-CONTROL *UNDEFINED	2
<p>Password protection:</p> <p>*NO=no password has been agreed for the system.</p> <p>*UNDEFINED=no information.</p> <p>*YES=system is password-protected.</p>	var(*LIST).ALL.HOSTS.HOST(*LIST).PASS-SPEC	S	*NO *UNDEFINED *YES	2
Processor name	var(*LIST).ALL.HOSTS.HOST(*LIST).PROCESSOR-NAME	S	<c-string 1..8> *UNDEFINED	2

(part 6 of 25)

Output information	Name of the S variable	T	Contents	Condition
<p>Status of a configuration modification/fault relating to the system/partner from the point of view of the system. States visible throughout the network regarding the corresponding modification in the XCS network are output.</p> <p>For CCS networks, only a subset is output (identified with “<i>P-T=*CCS</i>”) because modifications in CCS are not made known to the sharers.</p> <p><i>*ABORT-END</i>= disconnecting system/partner has completed the global termination of registered functions and is now waiting for local termination. Any system remaining in the network waits until all other sharers remaining in the network have also reached this state before switching to the FAIL REQUEST reconfiguration state.</p> <p><i>P-T=*CCS</i>: system has abnormally terminated MSCF; all imported shared pubsets are exported, disk monitoring is terminated.</p> <p><i>*ABORT-REQUEST</i>=partner is to disconnect from the XCS network due to an error.</p> <p><i>*ABORT-START</i>=system/partner disconnects from the XCS network and abnormally terminates the registered functions. System waits for the end of the global termination phase of the registered functions on the disconnecting sharer.</p> <p><i>P-T=*CCS</i>: system aborts MSCF due to errors; all imported shared pubsets are exported.</p> <p><i>*CONN-ACT</i>=MSCF connection to partner exists or has been re-established; however, other sharers may have lost their connection to this partner.</p> <p><i>*CONN-LOST</i>=MSCF connection to partner is faulty, has crashed or has been cleared down; however, disk monitoring still identifies the partner as active.</p>	<p>var(*LIST).ALL.HOSTS.HOST(*LIST).RECONF-STATE</p>	<p>S</p>	<p><i>*ABORT-END</i> <i>*ABORT-REQUEST</i> <i>*ABORT-START</i> <i>*CONN-ACT</i> <i>*CONN-LOST</i> <i>*CONTROL-CRASH</i> <i>*CONTROL-FAIL</i> <i>*FAIL-END</i> <i>*FAIL-REQUEST</i> <i>*FAIL-START</i> <i>*JOIN-END</i> <i>*JOIN-REQUEST</i> <i>*JOIN-START</i> <i>*LEAVE-END</i> <i>*LEAVE-REQUEST</i> <i>*LEAVE-START</i> <i>*NOT-ACTIVE</i> <i>*UNDEFINED</i></p>	<p>2</p>

(part 7 of 25)

Output information	Name of the S variable	T	Contents	Condition
<p>*CONTROL-CRASH=both monitoring paths to the partner have failed; no other network sharer identifies the partner as still active; partner may have crashed.</p> <p>*CONTROL-FAIL=both monitoring paths to the partner have failed; however, at least one other network sharer still identifies the partner as active.</p> <p><i>P-T=*CCS</i>: MSCF connection to partner has failed or been cleared down; partner no longer identifiable as active for systems on all shared pubsets; partner may have crashed; systems support action requested.</p> <p>*FAIL-END=partner has abnormally terminated MSCF and has been removed from the XCS network from the local point of view; system waits until all other sharers remaining in the network have reached this state before terminating the fail reconfiguration. Only then is the network re-established.</p> <p>*FAIL-REQUEST=partner has abnormally terminated MSCF and is to be removed from the XCS network; system waits for the start of the fail reconfiguration.</p> <p><i>P-T=*CCS</i>: partner has been declared crashed by the system; system waits for some time to allow the partner to initiate SETS if necessary</p> <p>*FAIL-START=partner has abnormally terminated MSCF and is removed from the XCS network; the remaining sharers in the network re-establish the network by implementing a fail reconfiguration.</p> <p><i>P-T=*CCS</i>: partner has abnormally terminated MSCF; affected shared pubset networks are re-established.</p> <p>*JOIN-END=system has joined XCS network or has added partner to the network; system waits until all other sharers have also reached this state before terminating the join reconfiguration.</p> <p>*JOIN-REQUEST=system wants to join XCS network or is to add partner to the network; system waits for the start of the join reconfiguration</p>	<p>var(*LIST).ALL.HOSTS.HOST(*LIST).RECONF-STATE (continued)</p>	S	<p>*ABORT-END *ABORT-REQUEST *ABORT-START *CONN-ACT *CONN-LOST *CONTROL-CRASH *CONTROL-FAIL *FAIL-END *FAIL-REQUEST *FAIL-START *JOIN-END *JOIN-REQUEST *JOIN-START *LEAVE-END *LEAVE-REQUEST *LEAVE-START *NOT-ACTIVE *UNDEFINED</p>	2

(part 8 of 25)

Output information	Name of the S variable	T	Contents	Condition
<p>*JOIN-START=system is joining the XCS network or is adding a partner to the XCS network; registered functions on the system are started/reconfigured; network implements join reconfiguration.</p> <p>*LEAVE-END=system has left XCS network or has removed partner from the network; system waits until all other sharers have also reached this state before terminating the leave reconfiguration.</p> <p>*LEAVE-REQUEST=system wants to leave the XCS network or remove partner from the network; the XCS functionality used by the applications on the exiting sharer have been released; system waits for the start of the leave reconfiguration.</p> <p>*LEAVE-START=system leaves the XCS network or removes partner from the network; registered functions on the system are terminated/reconfigured; network implements leave reconfiguration.</p> <p><i>P-T=*CCS</i>: system terminates MSCF; all imported shared pubsets are exported.</p> <p>*NOT-ACTIVE=from the point of view of the system, no reconfiguration pending / no configuration fault known for the system/partner.</p> <p><i>P-T=*CCS</i>: no reconfiguration pending for system / no crash detection or reconfiguration pending for partner.</p> <p>*UNDEFINED=No information</p>	var(*LIST).ALL.HOSTS.HOST(*LIST).RECONF-STATE (continued)	S	*ABORT-END *ABORT-REQUEST *ABORT-START *CONN-ACT *CONN-LOST *CONTROL-CRASH *CONTROL-FAIL *FAIL-END *FAIL-REQUEST *FAIL-START *JOIN-END *JOIN-REQUEST *JOIN-START *LEAVE-END *LEAVE-REQUEST *LEAVE-START *NOT-ACTIVE *UNDEFINED	2
<p>Method of starting the recovery measures (the partner-specific setting previously requested with /START-MSCF-CONNECTION).</p> <p>The requested value may temporarily differ from the current value (while the connection is inactive).</p> <p>See ALL.HOSTS.HOSTS(*LIST).CONN-RECOV-START (page 251) for the meaning of the value.</p>	var(*LIST).ALL.HOSTS.HOST(*LIST).REQ-CONN-RECOV-START	S	*AUTO *BY-OPER *CONSISTENT *SECURE *STD *UNDEFINED	2
Number of requested monitoring connections	var(*LIST).ALL.HOSTS.HOST(*LIST).REQ-CONTR-CONN	S	<integer 1..2> *UNDEFINED	2

(part 9 of 25)

Output information	Name of the S variable	T	Contents	Condition
Value of the partner's control group previously requested with /START-MSCF-CONNECTION. The requested value may temporarily differ from the current value (while the connection is inactive). The fail reconfiguration is started concurrently for all partners in the same monitoring group.	var(*LIST).ALL.HOSTS.HOST(*LIST).REQ-CONTROL-GROUP	S	<integer 1..9> *NONE *UNDEFINED	2
System identification of the system	var(*LIST).ALL.HOSTS.HOST(*LIST).SYS-ID	S	<c-string 1..3> *UNDEFINED	2
Release status of the BS2000 version of the system	var(*LIST).ALL.HOSTS.HOST(*LIST).SYS-SUBVERSION	S	<c-string 1..1> *UNDEFINED	2
BS2000 version of the system without release status (format 'Vmm.n')	var(*LIST).ALL.HOSTS.HOST(*LIST).SYS-VERSION	S	<c-string 5..5> *UNDEFINED	2
Name of the XCS network	var(*LIST).ALL.HOSTS.HOST(*LIST).XCS-NAME	S	<c-string 1..8> *UNDEFINED	2
Time stamp (UTC time) of the establishment of the XCS network consisting of start date and start time	var(*LIST).ALL.HOSTS.HOST(*LIST).XCS-START-TIME	S	<yyyy-mm-dd/ hh:mm:ss> *UNDEFINED	2
Number of systems for which information is provided	var(*LIST).ALL.HOSTS.NUM-HOSTS	I	<integer 0..164>	1
Maximum period within which the system must have left the network abnormally as part of an abort reconfiguration.	var(*LIST).ALL.LOC-ATTR.ABORT-LIM	S	<integer 0..6000> *UNLIMITED *UNDEFINED	1
Current number of MSCF server tasks	var(*LIST).ALL.LOC-ATTR.ACT-SERVER-NUM	I	<integer 2..4000>	1
Wait time between failure of a partner being reported by the Live Monitor and the start of the fail reconfiguration	var(*LIST).ALL.LOC-ATTR.FADING-INTERVAL	S	<integer 0..300> *UNDEFINED	1
Maximum error detection time in seconds	var(*LIST).ALL.LOC-ATTR.FAIL-DETEC-LIM	S	<integer 176..3300> *UNDEFINED	1
Number of free MSCF server tasks	var(*LIST).ALL.LOC-ATTR.FREE-SERVER-NUM	I	<integer 0..4000>	1
Host name of system	var(*LIST).ALL.LOC-ATTR.HOST-NAME	S	<c-string 1..8> *UNDEFINED	1
Priority of a system in the XCS network with automatic recovery after a loss of connection. The system with the higher value is removed from the XCS network with ABORT. If both systems have the same priority, the system which entered the XCS network last is removed.	var(*LIST).ALL.LOC-ATTR.HOST-PRIO	I	<integer 0..32> (0 = UNDEFINED)	1

(part 10 of 25)

Output information	Name of the S variable	T	Contents	Condition
Time stamp (UTC time) of the MSCF start on the system	var(*LIST).ALL.LOC-ATTR. HOST-START-TIME	S	<yyyy-mm-dd/ hh:mm:ss> *UNDEFINED	1
Maximum period within which the system must have completed a normal network exit (leave reconfiguration). When this period expires, the normal exit from the network is converted into an abnormal exit (ABORT).	var(*LIST).ALL.LOC-ATTR. LEAVE-LIM	S	<integer 1..6000> *UNLIMITED *UNDEFINED	1
MSCF version running on the system with release status (format 'Vmm.na')	var(*LIST).ALL.LOC-ATTR. MSCF-VERSION	S	<c-string 6..6> *UNDEFINED	1
Password protection: *NO=no password has been agreed for the system. *UNDEFINED=no information. *YES=system is password-protected.	var(*LIST).ALL.LOC-ATTR. PASS-SPEC	S	*NO *UNDEFINED *YES	1
Processor name	var(*LIST).ALL.LOC-ATTR. PROCESSOR-NAME	S	<c-string 1..8> *UNDEFINED	1
Type of recovery measures (general setting). A recovery measure can only be started automatically if this is allowed by both the general and the partner-specific setting on the system and on the partner. *AUTO= the local system can automatically start recovery measures for the partner, partners can automatically start recovery measures for the local system. *BY-OPER= the local system can only start recovery measures for the partner after the action has been confirmed by systems support. *CONSISTENT= the local system can only start recovery measures for the partner after the action has been confirmed by systems support. Partners can only start recovery measures for the local system after the action has been confirmed by systems support. *LOCKED= a system component of the partner holds the Cluster Recovery Lock. Acts like *SECURE.	var(*LIST).ALL.LOC-ATTR. RECOVERY-START	S	*AUTO *BY-OPER *CONSISTENT *LOCKED *SECURE *UNDEFINED	1

(part 11 of 25)

Output information	Name of the S variable	T	Contents	Condition
*SECURE= the partner only starts recovery measures for the local system automatically if a failure has been detected or confirmed by the Live Monitor. The partner only permits the recovery measures to be started automatically by the local system if the failure has been detected or confirmed by the Live Monitor. *UNDEFINED=no information.	var(*LIST).ALL.LOC-ATTR. RECOVERY-START (continued)	S	*AUTO *BY-OPER *CONSISTENT *LOCKED *SECURE *UNDEFINED	1
Method of recovery start (value of general setting previously requested with /SET-MSCF-ENVIRONMENT or /MODIFY-MSCF-ENVIRONMENT). The value may temporarily differ from the current value. See ALL.LOC-ATTR.RECOVERY-START above for the meaning of the value.	var(*LIST).ALL.LOC-ATTR. REQ-RECOVERY-START	S	*AUTO *BY-OPER *CONSISTENT *LOCKED *SECURE *STD *UNDEFINED	
Maximum number of MSCF server tasks	var(*LIST).ALL.LOC-ATTR. SERVER-LIM	S	<integer 10..500> *UNDEFINED	1
System identification of the system	var(*LIST).ALL.LOC-ATTR.SYS-ID	S	<c-string 3..3> *UNDEFINED	1
Release status of the BS2000 version of the system	var(*LIST).ALL.LOC-ATTR. SYS-SUBVERSION	S	<c-string 1..1> *UNDEFINED	1
BS2000 version of the system without release status (format'Vmm.n')	var(*LIST).ALL.LOC-ATTR. SYS-VERSION	S	<c-string 1..5> *UNDEFINED	1
Time available to the tasks using XCS functionality to release the XCS functionality when disconnecting from the network, before the registered XCS functions are terminated	var(*LIST).ALL.LOC-ATTR. USER-TERM-LIM	S	<integer 0..6000> *UNLIMITED *UNDEFINED	1
Current XCS reconfiguration number, which is incremented by 1 with each configuration modification	var(*LIST).ALL.LOC-ATTR. XCS-CONFIGURATION	I	<integer 0.2147483647> (0 = UNDEFINED)	1
Name of the XCS network	var(*LIST).ALL.LOC-ATTR. XCS-NAME	S	<c-string 1..8> *UNDEFINED	1
Time stamp (UTC time) of the XCS network start	var(*LIST).ALL.LOC-ATTR. XCS-START-TIME	S	<yyyy-mm-dd/ hh:mm:ss> *UNDEFINED	1

(part 12 of 25)

Output information	Name of the S variable	T	Contents	Condition
<p>Status of XCS functionality on the system:</p> <p>*ACTIVE=system actively participating in XCS network; full XCS functionality is available. XCS reconfiguration is not implemented, no fault is known.</p> <p>*INTERRUPTED=XCS network is interrupted due to error (for cause, see "RECONF STATE"); XCS functionality is available on the system to a limited extent, depending on the type of fault.</p> <p>*NOT-ACTIVE=system is not yet or no longer connected to the XCS network; XCS functionality not available on the system.</p> <p>*NOT-GENERATED=system is not generated as XCS sharer. MSCF only supports LCS and CCS networks; local system XCS functionality is available.</p> <p>*RECONF-FAIL=one or more partners have disconnected from the XCS network; the XCS network is re-established by the system by removing the crashed partners from the XCS network. Depending on the type of error, a fail reconfiguration of the XCS functionality is not available in full on the system until the reconfiguration is completed.</p> <p>*RECONF-JOIN=XCS network expanding, a join reconfiguration of the XCS functionality is implemented (system joins network/adds partner to network). The XCS functionality is not available when the system is joining, but is available in full on the system when a partner is being added to the network.</p> <p>*RECONF-LEAVE=XCS network is being reduced, a leave reconfiguration of the XCS functionality is implemented (system leaves network/removes partner from network). The XCS functionality is no longer available when the system leaves the network, but is available in full on the system when a partner is removed.</p> <p>*UNDEFINED=no information</p> <p>*XCS-SUSPENDED= system is intended as an XCS sharer, but it did not enter an XCS network with the previously initiated START-SUBSYSTEM-MSCF command. XCS functionality is not available at present. However, a restart of MSCF in XCS mode is possible.</p>	var(*LIST).ALL.LOC-ATTR. XCS-STATE	S	*ACTIVE *INTERRUPTED *NOT-ACTIVE *NOT-GENERATED *RECONF-FAIL *RECONF-JOIN *RECONF-LEAVE *UNDEFINED *XCS-SUSPENDED	1

(part 13 of 25)

Output information	Name of the S variable	T	Contents	Condition
Availability attribute of a registered function: *MANDATORY=must be available *OPTIONAL=not mandatory	var(*LIST).ALL.REG-FUNC. FUNC(*LIST).FUNC-ATTR	S	*MANDATORY *OPTIONAL	3
Error state of the registered function with respect to MSCF network management: *ABORT=registered function requested termination of MSCF. *CRASH=registered function has terminated abnormally. The reconfiguration capability of the other network sharers or the MSCF restart may be hindered, depending on the "FUNCTION STATE". *NONE=no errors known. *PROTOCOL=registered function responded incorrectly to a reconfiguration job.	var(*LIST).ALL.REG-FUNC. FUNC(*LIST).FUNC-ERROR	S	*ABORT *CRASH *NONE *PROTOCOL	3
Name of registered function: CPM=shared pubset manager DAB=disk access buffer GSM=global storage manager GSV=global storage volume emulation NSG=node serialization global store part NSM=node serialization manager SHC=symmetrix host component XPM=XCS pubset manager XTS=XCS time synchronization	var(*LIST).ALL.REG-FUNC. FUNC(*LIST).FUNC-NAME	S	CPM DAB GSM GSV NSG NSM SHC XPM XTS	3
Reconfiguration state of registered function: *DISABLED=function has exited network and terminates (local termination phase). *ENABLED=function has completed local initialization and is ready to join network. *INITIATED=function is now being initialized; network entry is being prepared. *IN-RECONF=function is being reconfigured; network functionality may be restricted, depending on the reconfiguration and function. *IN-TERM=function is being terminated; it exits the network with partner entities and releases occupied network resources (network-wide termination phase). *NOT-ACTIVE=function is not active (not yet initialized or already terminated). *READY=function has successfully completed connection entry/reconfiguration; network functionality is available. *STARTED=function has been started/joins network with partner entities.	var(*LIST).ALL.REG-FUNC. FUNC(*LIST).FUNC-STATE	S	*DISABLED *ENABLED *INITIATED *IN-RECONF *IN-TERM *NOT-ACTIVE *READY *STARTED	3

(part 14 of 25)

Output information	Name of the S variable	T	Contents	Condition
Type of registered function: *CCS=CCS function *XCS=XCS function	var(*LIST).ALL.REG-FUNC. FUNC(*LIST).FUNC-TYPE	S	*CCS *XCS	3
Number of registered functions for which information is provided	var(*LIST).ALL.REG-FUNC. NUM-FUNC	I	<integer 0..16>	1
Method of starting recovery measures (partner-specific setting). See ALL.HOSTS.HOST(*LIST).CONN-RECOV-START on page 251 for the meaning of the values.	var(*LIST).HOSTS.HOST(*LIST). CONN-RECOV-START	S	*AUTO *BY-OPER *CONSISTENT *SECURE *STD *UNDEFINED	4
Status of MSCF connection: *MSCF-CONN=MSCF connection exists to partner. *MSCF-LOCAL=local system. *NOT-MSCF-CONN=no MSCF connection to partner, but local system wants connection to partner. *UNDEFINED=local system does not want connection to partner.	var(*LIST).HOSTS.HOST(*LIST). CONN-STA	S	*MSCF-CONN *MSCF-LOCAL *NOT-MSCF-CONN *UNDEFINED	4
Connection type: *CCS=closely coupled system *LCS=loosely coupled system *UNDEFINED=no information	var(*LIST).HOSTS.HOST(*LIST). CONN-TYPE	S	*CCS *LCS *UNDEFINED	4
Number of monitoring connections currently in use.	var(*LIST).HOSTS.HOST(*LIST). CONTR-CONN	S	<integer 0..2> *UNDEFINED	4
Current value of a partner's control group. A failure reconfiguration is started concurrently for all partners in the same control group.	var(*LIST).HOSTS.HOST(*LIST). CONTROL-GROUP	S	<integer 1..9> *NONE *UNDEFINED	4
Monitoring status of the partner. For precise meaning, see ALL.HOSTS.HOST(*LIST).CONTROL-STATE on page 253	var(*LIST).HOSTS.HOST(*LIST). CONTROL-STATE	S	*ACTIVE *COMM-ACTIVE *CONNECT-LOST-REQ *CONNECT-REQ *CRASHED *CRASHED-QUEST *CRITICAL *DISK-ACTIVE *LOST *MSCF-TERMINATED *NOT-ACTIVE *UNDEFINED	4
Host name of system	var(*LIST).HOSTS.HOST(*LIST). HOST-NAME	S	<c-string 1..8> *UNDEFINED	4

(part 15 of 25)

Output information	Name of the S variable	T	Contents	Condition
Priority of a system in the XCS network with automatic recovery after a loss of connection. The system with the higher value is removed from the XCS network with ABORT. If both systems have the same priority, the system which entered the XCS network last is removed.	var(*LIST).HOSTS.HOST(*LIST).HOST-PRIO	I	<integer 0..32> (0 = UNDEFINED)	4
Distinguishes local entry from the entries of the partners: *LOCAL=entry of local system *REMOTE=entry of a partner system *UNDEFINED=no information	var(*LIST).HOSTS.HOST(*LIST).HOST-SITE	S	*LOCAL *REMOTE *UNDEFINED	4
Time stamp (UTC time) of the MSCF start on the system	var(*LIST).HOSTS.HOST(*LIST).HOST-START-TIME	S	<yyyy-mm-dd/hh:mm:ss> *UNDEFINED	4
Status of the system with respect to the network with the system specified by "PARTNER-TYPE" (<i>P-T</i>); for precise meaning, see ALL.HOSTS.HOST(*LIST).HOST-STATE on page 255 .	var(*LIST).HOSTS.HOST(*LIST).HOST-STATE	S	*ABORT *ACTIVE *CRASH *FAIL *JOIN *LEAVE *NOT-JOINED *UNDEFINED	4
Order of entry into the XCS network	var(*LIST).HOSTS.HOST(*LIST).JOINING-ORDER	I	<integer 0..2147483647> (0=UNDEFINED)	4
MSCF version running on the system with release status (format 'Vmm.na')	var(*LIST).HOSTS.HOST(*LIST).MSCF-VERSION	S	<c-string 1..6> *UNDEFINED	4
Overloaded connections: --=local system or there is no connection to the partner NN=no congestion NY=congestion for REPLYs YN=congestion for REQUESTs YY=congestion in both directions or connection recovery is active	var(*LIST).HOSTS.HOST(*LIST).OVERLOAD-STA	S	-- NN NY YN YY	4
Partner type: *CCS=partner is CCS partner *LCS=partner is LCS partner *UNDEFINED=no information *XCS=partner is XCS partner For the local system itself, the connection types the system can form with partners are indicated.	var(*LIST).HOSTS.HOST(*LIST).PARTNER-TYPE	S	*CCS *LCS *UNDEFINED *XCS	4

(part 16 of 25)

Output information	Name of the S variable	T	Contents	Condition
RECOVERY-START setting of the partner in relation to the local system (combined general and partner-specific setting). For precise meaning, see ALL.HOSTS.HOST(*LIST).PART-RECOV-START (page 256).	var(*LIST).HOSTS.HOST(*LIST).PART-RECOV-START	S	*AUTO *BY-OPER *CONSISTENT *LOCKED *SECURE *NO-CONTROL *UNDEFINED	4
Password protection: *NO=no password has been agreed for the system. *UNDEFINED=no information. *YES=system is password-protected.	var(*LIST).HOSTS.HOST(*LIST).PASS-SPEC	S	*NO *UNDEFINED *YES	4
Processor name	var(*LIST).HOSTS.HOST(*LIST).PROCESSOR-NAME	S	<c-string 1..8> *UNDEFINED	4
Status of a configuration modification / a fault in relation to the system or partner from the point of view of the system; for precise meaning, see ALL.HOSTS.HOST(*LIST).RECONF-STATE on page 258 .	var(*LIST).HOSTS.HOST(*LIST).RECONF-STATE	S	*ABORT-END *ABORT-REQUEST *ABORT-START *CONN-ACT *CONN-LOST *CONTROL-CRASH *CONTROL-FAIL *FAIL-END *FAIL-REQUEST *FAIL-START *JOIN-END *JOIN-REQUEST *JOIN-START *LEAVE-END *LEAVE-REQUEST *LEAVE-START *NOT-ACTIVE *UNDEFINED	4
Method of starting recovery measures (the value of the partner-specific setting requested with the previous /START-MSCF-CONNECTION). The requested value may temporarily differ from the current value (while the connection is inactive). For precise meaning, see ALL.HOSTS.HOST(*LIST).CONN-RECOV-START on page 251 .	var(*LIST).HOSTS.HOST(*LIST).REQ-CONN-RECOV-START	S	*AUTO *BY-OPER *CONSISTENT *SECURE *STD *UNDEFINED	4
Number of requested monitoring connections	var(*LIST).HOSTS.HOST(*LIST).REQ-CONTR-CONN	S	<integer 1..2> *UNDEFINED	4

(part 17 of 25)

Output information	Name of the S variable	T	Contents	Condition
Value of the monitoring group requested with the previous /START-MSCF-CONNECTION. The requested value may temporarily differ from the current value (while the connection is inactive). Fail reconfigurations are started concurrently for partners in the same control group.	var(*LIST).HOSTS.HOST(*LIST).REQ-CONTROL-GROUP	S	<integer 1..9> *NONE *UNDEFINED	4
System identification of the system	var(*LIST).HOSTS.HOST(*LIST).SYS-ID	S	<c-string 1..3> *UNDEFINED	4
Release status of the BS2000 version of the system	var(*LIST).HOSTS.HOST(*LIST).SYS-SUBVERSION	S	<c-string 1..1> *UNDEFINED	4
BS2000 version of the system without release status (format 'Vmm.n')	var(*LIST).HOSTS.HOST(*LIST).SYS-VERSION	S	<c-string 1..5> *UNDEFINED	4
Name of the XCS network	var(*LIST).HOSTS.HOST(*LIST).XCS-NAME	S	<c-string 1..8> *UNDEFINED	4
Time stamp (UTC time) of the XCS connection set-up consisting of start date and start time	var(*LIST).HOSTS.HOST(*LIST).XCS-START-TIME		<yyyy-mm-dd/ hh:mm:ss> *UNDEFINED	4
Number of systems for which information is provided	var(*LIST).HOSTS.NUM-HOSTS	I	<integer 1..164>	4
Maximum period within which the system must have left the network abnormally as part of an abort reconfiguration.	var(*LIST).LOC-ATTR.ABORT-LIM	S	<integer 0..6000> *UNLIMITED *UNDEFINED	5
Current number of MSCF server tasks	var(*LIST).LOC-ATTR.ACT-SERVER-NUM	I	<integer 2..4000>	5
Wait time between failure of a partner being reported by the Live Monitor and the start of the fail reconfiguration	var(*LIST).LOC-ATTR.FADING-INTERVAL	S	<integer 0..300> *UNDEFINED	5
Maximum error detection time in seconds	var(*LIST).LOC-ATTR.FAIL-DETEC-LIM	S	<integer 176..3300> *UNDEFINED	5
Number of free MSCF server tasks	var(*LIST).LOC-ATTR.FREE-SERVER-NUM	I	<integer 0..4000>	5
Host name of system	var(*LIST).LOC-ATTR.HOST-NAME	S	<c-string 1..8> *UNDEFINED	5
Priority of a system in the XCS network with automatic recovery after a loss of connection. The system with the higher value is removed from the XCS network with ABORT. If both systems have the same priority, the system which entered the XCS network last is removed.	var(*LIST).LOC-ATTR.HOST-PRIO	I	<integer 0..32> (0 = UNDEFINED)	5

(part 18 of 25)

Output information	Name of the S variable	T	Contents	Condition
Time stamp (UTC time) of the MSCF start on the system	var(*LIST).LOC-ATTR. HOST-START-TIME	S	<yyyy-mm-dd/ hh:mm:ss> *UNDEFINED	5
Maximum period within which the system must have completed a normal network exit (leave reconfiguration). When this period expires, the normal exit from the network is converted into an abnormal exit (ABORT).	var(*LIST).LOC-ATTR.LEAVE-LIM	S	<integer 1..6000> *UNLIMITED *UNDEFINED	5
MSCF version running on the system with release status (format 'Vmm.na')	var(*LIST).LOC-ATTR.MSCF-VERSION	S	<c-string 6..6> *UNDEFINED	5
Password protection: *NO=no password has been agreed for the system. *UNDEFINED=no information. *YES=system is password-protected.	var(*LIST).LOC-ATTR.PASS-SPEC	S	*NO *UNDEFINED *YES	5
Processor name	var(*LIST).LOC-ATTR. PROCESSOR-NAME	S	<c-string 1..8> *UNDEFINED	5
Type of recovery measures (general setting).For meaning see ALL.LOC-ATTR.RECOVERY-START on page 262 .	var(*LIST).LOC-ATTR. RECOVERY-START	S	*AUTO *BY-OPER *CONSISTENT *LOCKED *SECURE *UNDEFINED	5
Method of starting recovery measures (value of general setting last requested with /SET-MSCF-ENVIRONMENT or /MODIFY-MSCF-ENVIRONMENT). The requested value may temporarily differ from the current value. For meaning see ALL.LOC-ATTR.RECOVERY-START on page 262 .	var(*LIST).LOC-ATTR. REQ-RECOVERY-START	S	*AUTO *BY-OPER *CONSISTENT *LOCKED *SECURE *UNDEFINED	5
Maximum number of MSCF server tasks	var(*LIST).LOC-ATTR.SERVER-LIM	S	<integer 10..500> *UNDEFINED	5
System identification of the system	var(*LIST).LOC-ATTR.SYS-ID	S	<c-string 1..3> *UNDEFINED	5
Release status of the BS2000 version of the system	var(*LIST).LOC-ATTR. SYS-SUBVERSION	S	<c-string 1..1> *UNDEFINED	5
BS2000 version of the system without release status (format 'Vmm.n')	var(*LIST).LOC-ATTR.SYS-VERSION	S	<c-string 1..5> *UNDEFINED	5
Time available to the tasks using XCS functionality to release the XCS functionality when disconnecting from the network, before the registered XCS functions are terminated	var(*LIST).LOC-ATTR. USER-TERM-LIM	S	<integer 0..6000> *UNDEFINED *UNLIMITED	5

(part 19 of 25)

Output information	Name of the S variable	T	Contents	Condition
Number of the current XCS configuration. The number is incremented by 1 each time the configuration is modified.	var(*LIST).LOC-ATTR.XCS-CONFIGURATION	I	<integer 0..2147483647> (0 = UNDEFINED)	5
Name of the XCS network	var(*LIST).LOC-ATTR.XCS-NAME	S	<c-string 1..8> *UNDEFINED	5
Time stamp (UTC time) of XCS network establishment	var(*LIST).LOC-ATTR.XCS-START-TIME	S	<yyyy-mm-dd/ hh:mm:ss> *UNDEFINED	5
Status of XCS functionality on the system; for precise meaning, see ALL.LOC-ATTR.XCS-STATE on page 263 .	var(*LIST).LOC-ATTR.XCS-STATE	S	*ACTIVE *INTERRUPTED *NOT-ACTIVE *NOT-GENERATED *RECONF-FAIL *RECONF-JOIN *RECONF-LEAVE *UNDEFINED	5
Availability attribute of a registered function: *MANDATORY=must be available *OPTIONAL=not mandatory	var(*LIST).REG-FUNC.FUNC(*LIST). FUNC-ATTR	S	*MANDATORY *OPTIONAL	6
Error state of the registered function with respect to MSCF network management: *ABORT=registered function requested termination of MSCF. *CRASH=registered function has terminated abnormally. The reconfiguration capability of the other network sharers or the restart of MSCF may be hindered, depending on "FUNCTION STATE". *NONE=no errors known. *PROTOCOL=registered function responded incorrectly to a reconfiguration job.	var(*LIST).REG-FUNC.FUNC(*LIST). FUNC-ERROR	S	*ABORT *CRASH *NONE *PROTOCOL	6
Name of registered function: CPM=shared pubset manager DAB=disk access buffer GSM=global store manager GSV=global store volume emulation NSG=node serialization global store part NSM=node serialization manager SHC=symmetrix host component XPM=XCS pubset manager XTS=XCS time synchronization	var(*LIST).REG-FUNC.FUNC(*LIST). FUNC-NAME	S	CPM DAB GSM GSV NSG NSM SHC XPM XTS	6

(part 20 of 25)

Output information	Name of the S variable	T	Contents	Condition
Reconfiguration state of registered function; for precise meaning, see ALL.REG-FUNC.FUNC(*LIST).FUNC-STATE on page 264 .	var(*LIST).REG-FUNC.FUNC(*LIST).FUNC-STATE	S	*DISABLED *ENABLED *INITIATED *IN-RECONF *IN-TERM *NOT-ACTIVE *READY *STARTED	6
Type of registered function: *CCS=CCS function *XCS=XCS function	var(*LIST).REG-FUNC.FUNC(*LIST).FUNC-TYPE	S	*CCS *XCS	6
Number of registered functions for which information is provided	var(*LIST).REG-FUNC.NUM-FUNC	I	<integer 1..16>	6
Method of starting recovery measures (current value of partner-specific setting): For meaning of value see ALL.HOSTS.HOST(*LIST).CONN-RECOV-START on page 251 .	var(*LIST).SUMMARY.HOSTS.HOST(*LIST).CONN-RECOV-START	S	*AUTO *BY-OPER *CONSISTENT *SECURE *STD *UNDEFINED	8, 10
Status of MSCF connection: *MSCF-CONN=MSCF connection exists to partner. *MSCF-LOCAL=local system. *NOT-MSCF-CONN=no MSCF connection to partner, but local system wants connection to partner. *UNDEFINED=local system does not want connection to partner.	var(*LIST).SUMMARY.HOSTS.HOST(*LIST).CONN-STA	S	*MSCF-CONN *MSCF-LOCAL *NOT-MSCF-CONN *UNDEFINED	8, 10
Number of monitoring connections currently used	var(*LIST).SUMMARY.HOSTS.HOST(*LIST).CONTR-CONN	S	<integer 0..2> *UNDEFINED	8, 10
Current value of the partner's control group. The fail reconfiguration is started concurrently for all partners in the same control group.	var(*LIST).SUMMARY.HOSTS.HOST(*LIST).CONTROL-GROUP	S	<integer 1..9> *NONE *UNDEFINED	8, 10

(part 21 of 25)

Output information	Name of the S variable	T	Contents	Condition
Monitoring status of the partner. For precise meaning, see ALL.HOSTS.HOST(*LIST).CONTROL-STATE on page 252 .	var(*LIST).SUMMARY.HOSTS. HOST(*LIST).CONTROL-STATE	S	*ACTIVE *COMM-ACTIVE *CONNECT-LOST-REQ *CONNECT-REQ *CRASHED *CRASHED-QUEST *CRITICAL *DISK-ACTIVE *LOST *MSCF-TERMINATED *NOT-ACTIVE *UNDEFINED	8, 10
Host name of system	var(*LIST).SUMMARY.HOSTS. HOST(*LIST).HOST-NAME	S	<c-string 1..8> *UNDEFINED	8, 10
Status of the system with respect to the network with the system specified by "PARTNER-TYPE" (<i>P-T</i>); for precise meaning, see ALL.HOSTS.HOST(*LIST).HOST-STATE on page 253 .	var(*LIST).SUMMARY.HOSTS. HOST(*LIST).HOST-STATE	S	*ABORT *ACTIVE *FAIL *JOIN *LEAVE *NOT-JOINED *UNDEFINED	8, 10
MSCF version and release status running on the system (format 'Vmm.na')	var(*LIST).SUMMARY.HOSTS. HOST(*LIST).MSCF-VERSION	S	<c-string 6..6> *UNDEFINED	5
Overloaded connections: --=local system or there is no connection to the partner NN=no congestion NY=congestion for REPLYs YN=congestion for REQUESTs YY=congestion in both directions or connection recovery is active	var(*LIST).SUMMARY.HOSTS. HOST(*LIST).OVERLOAD-STA	S	-- NN NY YN YY	8, 10
Partner type: *CCS=partner is CCS partner *LCS=partner is LCS partner *UNDEFINED=no information *XCS=partner is XCS partner For the local system itself, the connection types the system can form with partners are indicated.	var(*LIST).SUMMARY.HOSTS. HOST(*LIST).PARTNER-TYPE	S	*CCS *LCS *UNDEFINED *XCS	8, 10

(part 22 of 25)

Output information	Name of the S variable	T	Contents	Condition
RECOVERY-START setting of the partner (combined setting in relation to the local system). For meaning see ALL.HOSTS.HOST(*LIST).PART-RECOV-START on page 256 .	var(*LIST).SUMMARY.HOSTS. HOST(*LIST).PART-RECOV-START	S	*AUTO *BY-OPER *CONSISTEND *LOCKED *SECURE *NO-CONTROL *UNDEFINED	8, 10
Processor name	var(*LIST).SUMMARY.HOSTS. HOST(*LIST).PROCESSOR-NAME	S	<c-string 1..8> *UNDEFINED	8, 10
Status of a configuration modification / a fault in relation to the system or partner from the point of view of the system; for precise meaning, see ALL.HOSTS.HOST(*LIST).RECONF-STATE on page 257 .	var(*LIST).SUMMARY.HOSTS. HOST(*LIST).RECONF-STATE	S	*ABORT-END *ABORT-REQUEST *ABORT-START *CONN-ACT *CONN-LOST *CONTROL-CRASH *CONTROL-FAIL *FAIL-END *FAIL-REQUEST *FAIL-START *JOIN-END *JOIN-REQUEST *JOIN-START *LEAVE-END *LEAVE-REQUEST *LEAVE-START *NOT-ACTIVE *UNDEFINED	8, 10
Method of starting recovery measures (requested value of partner-specific setting last requested with /START-MSCF-CONNECTION). The requested value may differ temporarily from the current value (while the connection is inactive. For meaning see ALL.HOSTS.HOST(*LIST).CONN-RECOV-START on page 251 .	var(*LIST).SUMMARY.HOSTS. HOST(*LIST).REQ-CONN-RECOV-START	S	*AUTO *BY-OPER *CONSISTENT *SECURE *STD *UNDEFINED	8, 10
Number of requested monitoring connections	var(*LIST).SUMMARY.HOSTS. HOST(*LIST).REQ-CONTR-CONN	S	<integer 1..2> *UNDEFINED	8, 10
Value of the partner's control group last requested with /START-MSCF-CONNECTION. The requested value may differ temporarily from the current value (while the connection is inactive). Fail configurations for partners in the same control group are started concurrently.	var(*LIST).SUMMARY.HOSTS. HOST(*LIST).REQ-CONTROL-GROUP	S	<integer 1..9> *NONE *UNDEFINED	8, 10

(part 23 of 25)

Output information	Name of the S variable	T	Contents	Condition
System identification	var(*LIST).SUMMARY.HOSTS. HOST(*LIST).SYS-ID	S	<c-string 1..3> *UNDEFINED	8, 10
Release status of the BS2000 version of the system	var(*LIST).SUMMARY.HOSTS. HOST(*LIST).SYS-SUBVERSION	S	<c-string 1..1> *UNDEFINED	8, 10
BS2000 version of the system without release status (format 'Vmm.n')	var(*LIST).SUMMARY.HOSTS. HOST(*LIST).SYS-VERSION	S	<c-string 1..5> *UNDEFINED	8, 10
Number of systems for which information is provided	var(*LIST).SUMMARY.HOSTS. NUM-HOSTS	I	<integer 0..164>	7, 10
Availability attribute of a registered function: *MANDATORY=must be available *OPTIONAL=not mandatory	var(*LIST).SUMMARY.REG-FUNC. FUNC(*LIST).FUNC-ATTR	S	*MANDATORY *OPTIONAL	9
Error state of the registered function with respect to MSCF network management: *ABORT=registered function requested termination of MSCF on the system. *CRASH=registered function has terminated abnormally. The reconfiguration capability of the other network sharers or the restart of MSCF may be hindered, depending on "FUNCTION STATE". *NONE=no errors known. *PROTOCOL=registered function responded incorrectly to a reconfiguration job.	var(*LIST).SUMMARY.REG-FUNC. FUNC(*LIST).FUNC-ERROR	S	*ABORT *CRASH *NONE *PROTOCOL	9
Name of registered function: CPM=shared pubset manager DAB=disk access buffer GSM=global store manager GSV=global store volume emulation NSG=node serialization global store part NSM=node serialization manager SHC=symmetrix host component XPM=XCS pubset manager XTS=XCS time synchronization	var(*LIST).SUMMARY.REG-FUNC. FUNC(*LIST).FUNC-NAME	S	CPM DAB GSM GSV NSG NSM SHC XPM XTS	9
Reconfiguration state of registered function; for precise meaning, see ALL.REG-FUNC.FUNC(*LIST).FUNC-STATE on page 265	var(*LIST).SUMMARY.REG-FUNC. FUNC(*LIST).FUNC-STATE	S	*DISABLED *ENABLED *INITIATED *IN-RECONF *IN-TERM *NOT-ACTIVE *READY *STARTED	9
Type of registered function: *CCS=CCS function *XCS=XCS function	var(*LIST).SUMMARY.REG-FUNC. FUNC(*LIST).FUNC-TYPE	S	*CCS *XCS	9

(part 24 of 25)

Output information	Name of the S variable	T	Contents	Condition
Number of registered functions for which information is provided	var(*LIST).SUMMARY.REG-FUNC.NUM-FUNC	I	<integer 0..16>	7, 10
Name of the XCS network	var(*LIST).SUMMARY.XCS-NAME.	S	<c-string 1..8> *UNDEFINED	7, 10
Status of the XCS functionality on the system; for the exact meaning, see under ALL.LOC-ATTR.XCS-STATE on page 263	var(*LIST).SUMMARY.XCS-STATE.	S	*ACTIVE *INTERRUPTED *NOT-ACTIVE *NOT-GENERATED *RECONF-FAIL *RECONF-JOIN *RECONF-LEAVE *UNDEFINED *XCS-SUSPENDED	7, 10

(part 25 of 25)

Note

The S variables listed in the following table continue to be supported for reasons of compatibility only. As of MSCF V14.0A, the S variables `***.HOST-CONTR` have been replaced by the HOST-STATE independent S variables `***.CONTROL-STATE`. The S variables `***.RECONF-NUM` and `***.XCS-RECONF-NUM` have been renamed into `***.JOINING-ORDER` and `***.XCS-CONFIGURATION`.

As of MSCF V13.0A, the S variables `***-LIMIT` have been replaced by the S variables `***-LIM`. For new programming only the S variables contained in the above table (as of [page 251](#)) should be used.

Output information	Name of the S variable	T	Contents	Condition
Partner monitoring status: For meaning of values see <code>ALL.HOSTS.HOST(*LIST).CONTROL-STATE</code> on page 252 .	<code>var(*LIST).ALL.HOSTS.HOST(*LIST).HOST-CONTR</code>	S	*ACTIVE *COMM-ACTIVE *CRASHED *CRITICAL *DISK-ACTIVE *LOST *NOT-ACTIVE *UNDEFINED	2
Order of joining XCS	<code>var(*LIST).ALL.HOSTS.HOST(*LIST).HOST-RECONF-NUM</code>	I	<integer 0..2147483647> (0=UNDEFINED)	2
Number of the current XCS configuration. The number is incremented by 1 each time the configuration is modified.	<code>var(*LIST).ALL.LOC-ATTR.XCS-RECONF-NUM</code>	I	<integer 0..2147483647> (0=UNDEFINED)	1
Partner monitoring status: For meaning of values see <code>ALL.HOSTS.HOST(*LIST).CONTROL-STATE</code> on page 252 .	<code>var(*LIST).HOSTS.HOST(*LIST).HOST-CONTR</code>	S	*ACTIVE *COMM-ACTIVE *CRASHED *CRITICAL *DISK-ACTIVE *LOST *NOT-ACTIVE *UNDEFINED	4
Order of joining XCS	<code>var(*LIST).HOSTS.HOST(*LIST).HOST-RECONF-NUM</code>	I	<integer 0..2147483647> (0=UNDEFINED)	4
Number of the current XCS configuration. The number is incremented by 1 each time the configuration is modified.	<code>var(*LIST).LOC-ATTR.XCS-RECONF-NUM</code>	I	<integer 0..2147483647> (0=UNDEFINED)	5
Partner monitoring status: For meaning of values see <code>ALL.HOSTS.HOST(*LIST).CONTROL-STATE</code> on page 252 .	<code>var(*LIST).SUMMARY.HOSTS.HOST(*LIST).HOST-CONTR</code>	S	*ACTIVE *COMM-ACTIVE *CRASHED *CRITICAL *DISK-ACTIVE *LOST *NOT-ACTIVE *UNDEFINED	8

SHOW-NSM-CONFIGURATION

Show information on state of NSM and its partners

Description status:	NSM V17.0A
Functional area:	Pubset and MRSCAT management
Domain:	MULTI-CATALOG-AND-PUBSET-MGMT
Privileges:	TSOS OPERATING SW-MONITOR-ADMINISTRATION
Routing code:	@

Function

The SHOW-NSM-CONFIGURATION command provides information on the state of the NSN subsystem and on the partners in an XCS network. The information is issued from the viewpoint of the local system.

The command supports structured output in S variables (see [page 281](#) and the “Commands” manual [10]).

Format

```

SHOW-NSM-CONFIGURATION

INFORMATION = *SUMMARY / *ALL / *HOSTS(...)
  *HOSTS(...)
    | SELECT = *ALL / <alphanum-name 1..8>
,OUTPUT = *SYSOUT / *SYSLST(...)
  *SYSLST(...)
    | SYSLST-NUMBER = *STD / <integer 1..99>

```

Operands

INFORMATION = *SUMMARY / *ALL / *HOSTS(...)

Specifies the information that the user wishes to view.

INFORMATION = *SUMMARY

Default: a summary of the local NSM settings and the state of the connection system is output.

INFORMATION = *ALL

All information on the NSM subsystem in the current XCS network of the own system is output. This information comprises:

- the local NSM configuration parameters
- the connection and network data of all systems in the current XCS network.

INFORMATION = *HOSTS(...)

The connection and network data of the systems specified with the SELECT operand is output.

SELECT = *ALL / <alphanum-name 1..8>

Specifies for which systems information is output.

SELECT = *ALL

Information is issued for all systems to which there is an NSM network relationship.

SELECT = <alphanum-name 1..8>

Information is issued for the specified system.

OUTPUT = *SYSOUT / SYSLST(...)

Defines the destination to which output is directed.

OUTPUT = *SYSOUT

Default: output is directed to the SYSOUT system file.

OUTPUT = SYSLST(...)

Output is directed to the SYSLST system file.

SYSLST-NUMBER = *STD / <integer 1..99>

Specifies whether output is to be directed to the SYSLST system file or to one of the SYSLST files from SYSLST01 to SYSLST99.

SYSLST-NUMBER = *STD

Default: output is to be directed to the SYSLST system file.

SYSLST-NUMBER = <integer 1..99>

Output is to be directed to one of the SYSLST files from SYSLST01 to SYSLST99.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
	0	NLX1020	No information for current selection
	3	CMD2203	Wrong syntax file
	32	CMD0221	System error (internal error)
	32	CMD2009	Error during output to a structured S variable
	32	NLX1019	Error while processing the command
	64	CMD0216	User does not have the necessary privileges
	130	NLX1019	Error while processing the command; repeat possible later

Output format

The contents of the output depends on the selection made:

*Output for INFORMATION=*SUMMARY*

```
/SHOW-NSM-CONFIGURATION INFORMATION=*SUMMARY
```

```
SHOW-NSM-CONFIGURATION DATE: yyyy-mm-dd, TIME: hh:mm:ss
```

```
-----
```

```
SUMMARY
```

```
OWN HOST NAME = D017ZE30
```

```
XCS NAME = XCS
```

```
NSM VERSION = V17.0A
```

```
COMM = *BY-MSCF
```

```
COMM-STATE = *MSCF-ACTIVE
```

```
COMM-TYPE = *TOKEN-RING
```

```
MINIMUM TOKEN-DELAY TIME = 15 MAXIMUM TOKEN-DELAY TIME = 100
```

```
-----
```

```
SHOW-NSM-CONFIGURATION
```

```
END OF DISPLAY
```

*Output for INFORMATION=*HOSTS(*ALL)*

```
/SHOW-NSM-CONFIGURATION INFORMATION=*HOSTS(*ALL)
```

```
SHOW-NSM-CONFIGURATION DATE: yyyy-mm-dd, TIME: hh:mm:ss
```

```
-----
```

```
HOSTS
```

```
NUM OF HOSTS = 2
```

```
HOST NAME COMM STATE SYS VERSION NSM VERSION
```

```
-----
```

```
D016ZE01 MSCF ACTIVE V17.0L V1V176.0A
```

```
D016ZE02 LOCAL HOST V17.0L V17.0A
```

```
-----
```

```
SHOW-NSM-CONFIGURATION
```

```
END OF DISPLAY
```

Key to output

The output fields and their possible values are listed in alphabetical order in the table below. Note that each output field may also contains the keyword UNDEFINED (UNDEFINED: no information available) which we have not expressly included in the table.

Output field	Keyword / meaning
COMM STATE	System status with regard to the connection: <ul style="list-style-type: none"> – *NONE: the NSM subsystem is still in the installation phase or the local system is the only system in the network. – *LOCAL-HOST: local system – *MSCF-ACTIVE: NSM only uses the MSCF connection as a communication path – *GS-ACTIVE: NSM uses the MSCF and the shared GS connection as communication paths – *GS-RECONF: the connection via GS is currently being reconfigured, the MSCF connection to the partner is still functioning. – *INTERRUPTED: the connection to the partner is currently interrupted
COMM TYPE	Communication method: *TOKEN-RING NSM uses the token ring communication method.
COMMUNICATION	Communication path: <ul style="list-style-type: none"> – *BY-MSCF: NSM uses MSCF – *BY-GS: NSM also uses the GS – *SWITCHING-TO-MSCF: the MODIFY-NSM-CONFIGURATION command was issued. NSM will therefore eliminate the communication path via GS. – *SWITCHING-TO-GS: the MODIFY-NSM-CONFIGURATION command was issued. NSM will therefore add the communication path via GS.
HOST NAME	Host name of the partner (as issued by the SHOW-MSCF-CONFIGURATION command)
MAXIMUM TOKEN DELAY TIME	Time (1..32767 ms) which an idle token requires to complete one cycle.
MINIMUM TOKEN DELAY TIME	Time (0..32767 ms) which a token requires to complete one cycle in normal operation.

Table 5: Output fields of the SHOW-NSM-CONFIGURATION command (part 1 of 2)

Output field	Keyword / meaning
NSM VERSION	NSM version running on the host or on the partner. V17.0A: BS2000/OSD-BC V8.0 *UNDEFINED: version unknown
NUMBER OF HOSTS	Number of hosts (1..16) on which information is issued.
OWN HOST NAME	Name of own (local) host.
SYS VERSION	BS2000-Version running on the host/partner. V17.0A: BS2000/OSD-BC V8.0
XCS NAME	Name of the XCS network: – <nnnnnnnn>: alphanumeric, 1..8 characters – UNDEFINED: no information available.

Table 5: Output fields of the SHOW-NSM-CONFIGURATION command (part 2 of 2)

Output in S variables

The INFORMATION operand of the command determines which S variables are to be created. INFORMATION accepts the following values:

Notation used in command	Abbreviated form used in table
INFORMATION = *SUMMARY	1
INFORMATION = *HOSTS	2
INFORMATION = *ALL	3

Output information	Name of the S variable	T	Contents	Condition
Number of hosts about which information is provided	var(*LIST).ALL.HOSTS.NUM-OF-HOSTS	I	<integer 1.. 16>	3
Unique name of the local host/partner	var(*LIST).ALL.HOSTS.HOST(*LIST).HOST-NAME	S	<c-string 1 .. 8> UNDEFINED	3
Status of hosts in respect of the connection	var(*LIST).ALL.HOSTS.HOST(*LIST).COMM-STATE	S	*NONE *LOCAL-HOST *MSCF-ACTIVE *GS-ACTIVE *GS-RECONF *INTERRUPTED	3
BS2000 version running on the local host/partner	var(*LIST).ALL.HOSTS.HOST(*LIST).SYS-VERSION	S	<c-string 1 .. 6> UNDEFINED	3
NSM version running on the local host/partner	var(*LIST).ALL.HOSTS.HOST(*LIST).NSM-VERSION	S	<c-string 1 .. 6> UNDEFINED	3

(part 1 of 3)

Output information	Name of the S variable	T	Contents	Condition
Communication path	var(*LIST).ALL.SUMMARY.COMM	S	*BY-MSCF *BY-GS *SWITCHING-TO-GS *SWITCHING-TO-MSCF	3
Communication method	var(*LIST).ALL.SUMMARY.COMM-TYPE	S	*TOKEN-RING	3
Status of hosts in respect of the connection	var(*LIST).ALL.SUMMARY.COMM-STATE	S	*NONE *LOCAL-HOST *MSCF-ACTIVE *GS-ACTIVE *GS-RECONF *INTERRUPTED	3
NSM version running on the local processor V13.0A: BS2000/OSD-BC V4.0 V14.0A: BS2000/OSD-BC V5.0 V15.0A: BS2000/OSD-BC V6.0 V16.0A: BS2000/OSD-BC V7.0 V17.0A: BS2000/OSD-BC V8.0	var(*LIST).ALL.SUMMARY.NSM-VERSION	S	V13.0A V14.0A V15.0A V16.0A V17.0A	3
Specified minimum time required for a token to complete a round-trip in the no-load state.	var(*LIST).ALL.SUMMARY. MAX-DELAY-TIME	I	<integer 1 .. 32767>	3
Specified minimum time required for a token to complete a round-trip in normal operation.	var(*LIST).ALL.SUMMARY.MIN-DELAY-TIME	I	<integer 0 .. 32767>	3
Unique name of the local host	var(*LIST).ALL.SUMMARY.HOST-NAME	S	<c-string 1 .. 8>	3
Unique name of the XCS network	var(*LIST).ALL.SUMMARY.XCS-NAME	S	<c-string 1 .. 8> UNDEFINED	3
Number of hosts about which information is provided	var(*LIST).HOSTS.NUM-OF-HOSTS	I	<integer 1 .. 16>	2
Unique name of the local host/ partner	var(*LIST).HOSTS.HOST(*LIST). HOST-NAME	S	<c-string 1 .. 8> UNDEFINED	2
Status of hosts in respect of the connection	var(*LIST).HOSTS.HOST(*LIST). COMM-STATE	S	*NONE *LOCAL-HOST *MSCF-ACTIVE *GS-ACTIVE *GS-RECONF *INTERRUPTED	2
BS2000 version running on the local host	var(*LIST).HOSTS.HOST(*LIST). SYS-VERSION	S	<c-string 1 .. 6> UNDEFINED	2
NSM version running on the local host	var(*LIST).HOSTS.HOST(*LIST). NSM-VERSION	S	<c-string 1 .. 6> UNDEFINED	2

(part 2 of 3)

Output information	Name of the S variable	T	Contents	Condition
Communication path	var(*LIST).SUMMARY.COMM	S	*BY-MSCF *BY-GS *SWITCHING-TO-GS *SWITCHING-TO-MSCF	1
Communication method	var(*LIST).SUMMARY.COMM-TYPE	S	*TOKEN-RING	1
Status of hosts in respect of the connection	var(*LIST).SUMMARY.COMM-STATE	S	*NONE *LOCAL-HOST *MSCF-ACTIVE *GS-ACTIVE *GS-RECONF *INTERRUPTED	1
NSM version running on the local processor V13.0A: BS2000/OSD-BC V4.0 V14.0A: BS2000/OSD-BC V5.0 V15.0A: BS2000/OSD-BC V6.0 V16.0A: BS2000/OSD-BC V7.0 V17.0A: BS2000/OSD-BC V8.0	var(*LIST).SUMMARY.NSM-VERSION	S	V13.0A V14.0A V15.0A V16.0A V17.0A	1
Specified minimum time required for a token to complete a round-trip in the no-load state.	var(*LIST).SUMMARY.MAX-DELAY-TIME	I	<integer 1 .. 32767>	1
Specified minimum time required for a token to complete a round-trip in normal operation.	var(*LIST).SUMMARY.MIN-DELAY-TIME	I	<integer 0 .. 32767>	1
Unique name of the local host	var(*LIST).SUMMARY.HOST-NAME	S	<c-string 1 .. 8>	1
Unique name of the XCS network	var(*LIST).SUMMARY.XCS-NAME	S	<c-string 1 .. 8> UNDEFINED	1

(part 3 of 3)

Examples

```

/declare-variable var-name=var(type=*structure),multiple-elem=*list
/exec-cmd (show-nsm-configuration inf=*summary),text-output=*none,structure-
  output=var
/show-var var
VAR(*LIST).SUMMARY.HOST-NAME = LOLITA2
VAR(*LIST).SUMMARY.XCS-NAME = OS36XCS
VAR(*LIST).SUMMARY.NSM-VERSION = V17.0A
VAR(*LIST).SUMMARY.COMM-TYPE = *TOKEN-RING
VAR(*LIST).SUMMARY.MIN-DELAY-TIME = 15
VAR(*LIST).SUMMARY.MAX-DELAY-TIME = 100
VAR(*LIST).SUMMARY.COMM = *BY-MSCF
VAR(*LIST).SUMMARY.COMM-STATE = *MSCF-ACTIVE

/exec-cmd (show-nsm-configuration inf=*host),text-output=*none,structure-
  output=var
/show-var var
VAR(*LIST).HOSTS.NUM-OF-HOSTS = 2
VAR(*LIST).HOSTS.HOST(*LIST).HOST-NAME = STARTA2
VAR(*LIST).HOSTS.HOST(*LIST).COMM-STATE = *MSCF-ACTIVE
VAR(*LIST).HOSTS.HOST(*LIST).SYS-VERSION = V17.0A
VAR(*LIST).HOSTS.HOST(*LIST).NSM-VERSION = V17.0A
VAR(*LIST).HOSTS.HOST(*LIST).HOST-NAME = LOLITA2
VAR(*LIST).HOSTS.HOST(*LIST).COMM-STATE = *LOCAL-HOST
VAR(*LIST).HOSTS.HOST(*LIST).SYS-VERSION = V17.0A
VAR(*LIST).HOSTS.HOST(*LIST).NSM-VERSION = V17.0A

/exec-cmd (show-nsm-configuration inf=*all),text-output=*none,structure-
  output=var
/show-var var
VAR(*LIST).ALL.SUMMARY.HOST-NAME = LOLITA2
VAR(*LIST).ALL.SUMMARY.XCS-NAME = OS36XCS
VAR(*LIST).ALL.SUMMARY.NSM-VERSION = V17.0A
VAR(*LIST).ALL.COMM-TYPE = *TOKEN-RING
VAR(*LIST).ALL.SUMMARY.MIN-DELAY-TIME = 15
VAR(*LIST).ALL.SUMMARY.MAX-DELAY-TIME = 100
VAR(*LIST).ALL.SUMMARY.COMM = *BY-MSCF
VAR(*LIST).ALL.SUMMARY.COMM-STATE = *MSCF-ACTIVE
VAR(*LIST).ALL.HOSTS.NUM-OF-HOSTS = 2
VAR(*LIST).ALL.HOSTS.HOST(*LIST).HOST-NAME = STARTA2
VAR(*LIST).ALL.HOSTS.HOST(*LIST).COMM-STATE = *MSCF-ACTIVE
VAR(*LIST).ALL.HOSTS.HOST(*LIST).SYS-VERSION = V17.0A
VAR(*LIST).ALL.HOSTS.HOST(*LIST).NSM-VERSION = V17.0A
VAR(*LIST).ALL.HOSTS.HOST(*LIST).HOST-NAME = LOLITA2
VAR(*LIST).ALL.HOSTS.HOST(*LIST).COMM-STATE = *LOCAL-HOST
VAR(*LIST).ALL.HOSTS.HOST(*LIST).SYS-VERSION = V17.0A
VAR(*LIST).ALL.HOSTS.HOST(*LIST).NSM-VERSION = V17.0A

```

SHOW-SHARED-PUBSET

Request information on shared pubsets

Description status:	MSCF V17.0A
Functional area:	Pubset and MRSCAT management
Domain:	MULTI-CATALOG-AND-PUBSET-MGMT
Privileges:	STD-PROCESSING TSOS OPERATING
Routing code:	E

Function

The SHOW-SHARED-PUBSET command provides information on the main characteristics of the “sharers” for shared pubsets.

“Sharers” are processors that have imported a particular pubset simultaneously.

If no shared pubset is available, the message MCA0201 PUBSET(S) NOT SHARED is output.

The command supports structured output in S variables (see [page 288](#) and the “Commands” manual [10]).

Format

SHOW-SHARED-PUBSET
PUBSET = *ALL / <cat-id 1..4> / *XCS / *CCS

Operands

PUBSET =

ID of the pubset on whose sharers information is requested.

PUBSET = *ALL

Default value: information given on all imported shared pubsets.

PUBSET = <cat-id 1..4>

ID of the pubset on which information is to be provided.

PUBSET = *XCS

Information is to be provided about all shared pubsets which have been imported as XCS pubsets.

PUBSET = *CCS

Information is to be provided about all shared pubsets which have been imported purely as CCS pubsets (i.e. not as XCS pubsets).

Output format

The following data is output for each shared pubset (from an internal table):

```
SHARER CONFIGURATION OF SHARED PUBSET :<catid>:
PARTNER  HOME  HOME  SHARER  SHARER  BS2000
NAME     CATID  SYSID  TYPE     STATE   VERSION
@@@@@@@@ @@@@  @@@   @@@@@@@ @@@@@@@ @@@@@@
```

Example

```
SHARER CONFIGURATION OF SHARED PUBSET :M1D1:
% PARTNER  HOME  HOME  SHARER  SHARER  BS2000
% NAME     CATID  SYSID  TYPE     STATE   VERSION
% D016ZE07 10SH  152   SLAVE   IMCAT   V17.0
% SQHAV101 C3H1  176   SLAVE   SHUTD   V17.0
% D016ZE10 20V1  157   SLAVE   IMCAT   V16.0
% D016ZE04 20SH  163   SLAVE   IMCAT   V17.0
% D016ZE00 20V0  155   MASTER  IMCAT   V15.0
```

Key to output

Output field	Meaning / Information
BS2000 VERSION	BS2000 version of the shared-pubset sharer (e.g. V15.0, V16.0, V17.0)
HOME CATID	Home catid of the shared-pubset sharer
HOME SYSID	System ID of the shared-pubset sharer
PARTNER NAME	Host name of the shared-pubset sharer
SHARER STATE	Partner status from the viewpoint of the local host. The values refer to the system specified under PARTNER-NAME. Meaning: <ul style="list-style-type: none"> - CHECK: The disk protocol has crashed. Monitoring possible only via the MSCF connections - CRASH: The partner system has failed. - EXCAT: Partner system has correctly exported the pubset

Table 6: Output values of the SHOW-SHARED-PUBSET command (part 1 of 2)

Output field	Meaning / Information
SHARER STATE (continued)	<ul style="list-style-type: none"> – IMCAT: Pubset is imported by the system – MCHANGE: A master change is currently running for the pubset – READERR: Permanent read error on the disk – SHUTDOWN Partner system correctly exported the pubset in the course of MSCF or BS2000 shutdown processing – WRTErr: Permanent write error on the disk
SHARER TYPE	Sharer type of the shared-pubset sharer: <ul style="list-style-type: none"> – BACKUP: A master change is currently running; the sharer is the backup master of the shared pubset – MASTER: The sharer is the master of the shared pubset – SLAVE: The sharer is the slave of the shared pubset

Table 6: Output values of the SHOW-SHARED-PUBSET command (part 2 of 2)

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
	1	CMD2201	Parameter error
	2	CMD2202	MSCF subsystem not defined
	32	MCS0025	Internal error (bourse call, memory request)
	32	MCA0204	Error on output to SYSOUT
	64	MCA0201	Pubset is not shareable
	64	MCA0202	Pubset cannot be accessed
	66	CMD2242	No connection to the MSCF subsystem
	128	CMD2280	MSCF subsystem temporarily unavailable
	129	CMD2281	MSCF subsystem unavailable for an extended period
	130	CMD2282	MSCF subsystem unavailable for an undefined period

Output in S variables

Output information	Name of the S variable	T	Contents	Condition
Catalog ID of the home pubset	var(*LIST).LIST(*LIST).HOME-PUBSET	S	<cat-id 1..4>	
Name of the partner system	var(*LIST).LIST(*LIST).PARTNER-NAME	S	<alphan.-name 1..8>	
Catalog ID of the shared pubset	var(*LIST).SHARED-PUBSET	S	<cat-id 1..4>	
Status of the partner system *CHECK=The disk log has failed. Monitoring is only being performed via the MSCF connection. *CRASH=The partner computer has failed. *EXP-PUBSET=The partner computer has exported the pubset correctly. *IMP-PUBSET=The pubset has been imported by the computer. *MASTER-CHA=A master change for the pubset is currently being run. *READ-ERROR=A permanent read error has occurred on the disk. *SHUTDOWN=The partner computer exported the pubset correctly when MSCF was shut down. *WRITE-ERROR=A permanent write error has occurred on the disk.	var(*LIST).LIST(*LIST).SHARER-STA	S	*CHECK *CRASH *EXP-PUBSET *IMP-PUBSET *MASTER-CHA *READ-ERROR *SHUTDOWN *WRITE-ERROR	
Partner system type *BACKUP=A master change is being performed; the sharer is the backup master of the shared pubset. *MASTER=The sharer is the master of the shared pubset. *SLAVE=The sharer is slave of the shared pubset.	var(*LIST).LIST(*LIST).SHARER-TYPE	S	*BACKUP *MASTER *SLAVE	
System ID	var(*LIST).LIST(*LIST).SYS-ID	S	<alphan.-name 1..3>	
System version	var(*LIST).LIST(*LIST).SYS-VERSION	S	<version>	

Example

```

/declare-var var-name=var(type=*structure),multiple-elem=*list
/exec-cmd cmd=(show-shared-pubset swkl),text-output=*none,struct-output=var
/show-var var,inf=*par(list-index=*yes)
OUT#1.SHARED-PUBSET = SWKL
OUT#1.LIST#1.PARTNER-NAME = D016ZE07
OUT#1.LIST#1.HOME-PUBSET = 10SH
OUT#1.LIST#1.SYS-ID = 152
OUT#1.LIST#1.SHARER-TYPE = *SLAVE
OUT#1.LIST#1.SHARER-STA = *IMP-PUBSET
OUT#1.LIST#1.SYS-VERSION = V17.0
OUT#1.LIST#2.PARTNER-NAME = SQHAV101
OUT#1.LIST#2.HOME-PUBSET = C3H1
OUT#1.LIST#2.SYS-ID = 176
OUT#1.LIST#2.SHARER-TYPE = *SLAVE
OUT#1.LIST#2.SHARER-STA = *SHUTDOWN
OUT#1.LIST#2.SYS-VERSION = V17.0
OUT#1.LIST#3.PARTNER-NAME = D016ZE10
OUT#1.LIST#3.HOME-PUBSET = 20V1
OUT#1.LIST#3.SYS-ID = 157
OUT#1.LIST#3.SHARER-TYPE = *SLAVE
OUT#1.LIST#3.SHARER-STA = *IMP-PUBSET
OUT#1.LIST#3.SYS-VERSION = V16.0
OUT#1.LIST#4.PARTNER-NAME = D016ZE04
OUT#1.LIST#4.HOME-PUBSET = 20SH
OUT#1.LIST#4.SYS-ID = 163
OUT#1.LIST#4.SHARER-TYPE = *SLAVE
OUT#1.LIST#4.SHARER-STA = *IMP-PUBSET
OUT#1.LIST#4.SYS-VERSION = V17.0
OUT#1.LIST#5.PARTNER-NAME = D016ZE00
OUT#1.LIST#5.HOME-PUBSET = 20V0
OUT#1.LIST#5.SYS-ID = 155
OUT#1.LIST#5.SHARER-TYPE = *MASTER
OUT#1.LIST#5.SHARER-STA = *IMP-PUBSET
OUT#1.LIST#5.SYS-VERSION = V15.0

```

Note

The command's source of information is the sign-of-life messages which the sharers exchange (disk log). As the current status of the shared pubset configuration is logged on the master system, a certain delay is to be expected on the slave systems, depending on the value of the FAIL-DETECTION-LIMIT configuration parameter.

SHOW-XCS-OCCUPATION

Show occupation information for TU tasks using XCS

Description status:	MSCF V17.0A
Functional area:	Pubset and MRSCAT management
Domain:	MULTI-CATALOG-AND-PUBSET-MGMT
Privileges:	TSOS OPERATING
Routing code:	@

Function

The SHOW-XCS-OCCUPATION command provides information on the TU tasks which are currently using global XCS-wide locks on the local host system. Up to 130 tasks can be shown. If information is available for more than that number of tasks, output breaks off after 130 tasks and an appropriate warning is issued.

The TU tasks for which information is supplied make use of XCS functionality and therefore have to be terminated if HIPLEX MSCF is terminated on the local host. Thus if HIPLEX MSCF is terminated or aborted, the associated tasks are aborted.

If this is not possible, HIPLEX MSCF cannot be terminated. The command then indicates which tasks are affected and thus enables systems support staff to take appropriate action.

The command supports structured output in S variables (see [page 292](#) and the “Commands” manual [10]).

Format

```
SHOW-XCS-OCCUPATION
```

```
OUTPUT = *SYSOUT / *SYSLST(...)
```

```
*SYSLST(...)
```

```
  | SYSLST-NUMBER = *STD / <integer 1..99>
```

Operands**OUTPUT = *SYSOUT / *SYSLST(...)**

Specifies where the information is to be output.

OUTPUT = *SYSOUT

The information is output to the system file SYSOUT.

OUTPUT = *SYSLST(...)

The information is output to the system file SYSLST.

SYSLST-NUMBER = *STD / <integer 1..99>

Specifies whether the information is to be output to the system file SYSLST or to a SYSLST file from the set SYSLST01 through SYSLST99.

The default is *STD, i.e. output is directed to the system file SYSLST.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without error
2	0	CMD0002	Warning, output incomplete
	1	CMD0202	Syntax error
	32	CMD0221	Internal error (system error)
	32	CMD2009	Error on OPS variable output
	64	CMD0216	User does not have required privilege
	64	MCS1019	No XCS network defined
	128	MCS1020	Not enough space

Output format

Assuming that 34 tasks are using XCS, the following information is output to SYSOUT:

```
/SHOW-XCS-OCCUPATION OUTPUT=*SYSOUT
```

```
SHOW-XCS-OCCUPATION
```

```
-----
```

```
NUMBER OF TASKS USING XCS: 34
```

```
LOCAL TASKS USING XCS:
```

```
  OT01  OT02  OT03  OT04  OT05  OT06  OT07  OT08  OT09  OT10  OT11  OT12  OT13
  OT14  OT15  OT16  OT17  OT18  OT19  OT20  OT21  OT22  OT23  OT24  OT25  OT26
  OT27  OT28  OT29  OT30  OT31  OT32  OT33  OT34
```

```
-----
```

```
SHOW-XCS-OCCUPATION
```

```
END OF DISPLAY
```

Key to the output

Output field	Meaning / Information
NUMBER OF TASKS USING XCS:	Number of tasks currently using XCS; numeric, 1 - 10 bytes long.
LOCAL TASKS USING XCS:	TSNs of the tasks currently using XCS List of TSNs, alphanumeric, 4 bytes long; breaks off after 130th task.

Table 7: Output values for SHOW-XCS-OCCUPATION command

Output in S variables

Output information	Name of the S variable	T	Contents	Condition
Number of tasks currently using the XCS	var(*LIST).NUMB-OF-TASKS	I	<integer 0..2147483647>	
TSNs of the tasks currently using the XCS (output is terminated after 130 TSNs)	var(*LIST).TSN-OF-TASKS	S	<alphan.-name 1..4>	Number of tasks > 0

START-MSCF-CONNECTION

Dynamically establish MSCF connection to partner system

Description status:	MSCF V17.0A
Functional area:	Pubset and MRSCAT management
Domain:	MULTI-CATALOG-AND-PUBSET-MGMT
Privileges:	TSOS OPERATING
Routing code:	R

Function

The START-MSCF-CONNECTION command is used to set up the connection to a partner system. This can either be an LCS or CCS connection. CCS connections are required for shared pubset or XCS networks. If passwords have been specified for local systems and/or partner systems, they have to be entered when a connection of the CCS type is established.

When MSCF is started, the connection is set up via entries in a configuration file (entered in the subsystem catalog or specified with SUBSYSTEM-PARAMETER='FILE-NAME=...' in the START-SUBSYSTEM command). The START-MSCF-CONNECTION and STOP-MSCF-CONNECTION commands allow MSCF connections to be set up or closed down. This does not modify the configuration file.

The connection to a partner system will only be established if the relevant command has already been executed on the partner system named in the command. Otherwise the connection request is only provisionally recorded.

Format

START-MSCF-CONNECTION
<p>PROCESSOR-NAME = <alphanum-name 1..8></p> <p>,CONNECTION-TYPE = *CLOSELY-COUPLED(...) / *LOOSELY-COUPLED</p> <p>*CLOSELY-COUPLED(...)</p> <p style="padding-left: 2em;">REMOTE-PASSWORD = *NONE / *SECRET / <c-string 0..8> / <x-string 1..16></p> <p style="padding-left: 2em;">,LOCAL-PASSWORD = *NONE / *SECRET / <c-string 0..8> / <x-string 1..16></p> <p style="padding-left: 2em;">,NUMBER-OF-CTRL-CONN = *NOT-SPECIFIED / 1 / 2</p> <p style="padding-left: 2em;">,RECOVERY-START = *NOT-SPECIFIED / *STD / *AUTOMATIC / *BY-OPERATOR / *CONSISTENT-BY-OPERATOR / *SECURE</p>

Operands

PROCESSOR-NAME = <alphanum-name 1..8>

Name of the system to which a connection is to be set up.

CONNECTION-TYPE =

Specifies the type of connection.

CONNECTION-TYPE = ***CLOSELY-COUPLED(...)**

Default. Selects CCS (closely-coupled system). CCS is a prerequisite for shared pubset and XCS network operation.

REMOTE-PASSWORD = ***NONE** / ***SECRET** / <c-string 1..8> / <x-string 1..16>

Here you have to give the password that the partner system defined with its LOCAL-PASSWORD operand. Otherwise the partner system will reject the connection. The REMOTE-PASSWORD operand has the following special characteristics:

- The password entered is not logged.
- The input field is automatically blanked out in the guided dialog.
- In unguided dialog and foreground procedures, the entry ***SECRET** or **^**, SDF provides a blanked out input field for inputting the password .

LOCAL-PASSWORD = *NONE / *SECRET / <c-string 1..8> / <x-string 1..16>

A password can be defined for an MSCF session on a system through an MSCF configuration file (see command SET-MSCF-ENVIRONMENT). If this has been done for the local system, then the password has to be entered here.

Otherwise the command will be rejected.

On the partner system, the password must be specified for the local system in the REMOTE-PASSWORD operand of the START-MSCF-CONNECTION command.

The LOCAL-PASSWORD operand has the following special characteristics:

- The password entered is not logged.
- The input field is automatically blanked out in the guided dialog.
- In unguided dialog and foreground procedures, the entry *SECRET or ^, SDF provides a blanked out input field for inputting the password .

NUMBER-OF-CTRL-CONN = *NOT-SPECIFIED / 1 / 2

Defines the number of independent monitoring connections through which the partner is monitored.

NUMBER-OF-CTRL-CONN = *NOT-SPECIFIED

When the connection is established for the first time, the partner is monitored via a monitoring connection. Otherwise the value set with the last START-MSCF-CONNECTION or MODIFY-MSCF-CONNECTION command remains invalid.

NUMBER-OF-CTRL-CONN = 1

The partner is monitored via a monitoring connection.

NUMBER-OF-CTRL-CONN = 2

The partner is monitored via two independent monitoring connections. If they both fail simultaneously a partner failure can be detected automatically, even without using a shared subset if RECOVERY-START= *AUTOMATIC is set partner-specifically.

Note

Independent monitoring connections only exist if *physically independent* communication paths were configured between the same-name MSCF applications \$MRSAPP4 and \$MRSAPP5 on the local system and on the partner system using the BCMAP command. MSCF cannot check whether this is actually the case. This means that NUMBER-OF-CTRL-CONN=2 confirms that the communication paths specified are physically independent. An incorrect specification may lead to erroneous failure detection and the loss of data integrity.

RECOVERY-START = *NOT-SPECIFIED / *STD / *AUTOMATIC / *BY-OPERATOR / *CONSISTENT-BY-OPERATOR / *SECURE

Defines partner-specifically when the partner is monitored or not (see [“Activating and deactivating MSCF partner monitoring” on page 124](#)) and how recovery measures are to be initiated in the case of an error (e.g. failure of a system or a connection).

Note

- The setting for RECOVERY-START cannot be changed while a system component holds the Cluster Recovery Lock (CRL) (see also the RELEASE-CLUSTER-RECOVERY-LOCK command).
- A fail reconfiguration can only be started automatically if partner monitoring is certain that the partner has failed and if this is allowed both by the general *and* the partner-specific RECOVERY-START settings on the local system and on the partner system (see [“Inhibiting the automatic start of fail reconfiguration” on page 125](#)).

RECOVERY-START = *NOT-SPECIFIED

When the connection is established for the first time, system behavior is the same as for RECOVERY-START=*STD (see below). Otherwise the value set when the START-MSCF-CONNECTION or MODIFY-MSCF-CONNECTION command last executed in the current MSCF session applies.

RECOVERY-START = *STD

Partner monitoring begins only when both the MSCF connection has been established and at least one joint shared pubset has been imported.

The general setting for the RECOVERY-START parameter determines the start time of fail reconfiguration.

Irrespective of this, fail configuration can never be initiated automatically if the partner is monitored only via two independent monitoring connections.

RECOVERY-START = *AUTOMATIC

Partner monitoring begins with the establishment of the MSCF connection.

The local system can start fail reconfiguration for the partner automatically even if the partner is only being monitored via two independent monitoring connections (see NUMBER-OF-CTRL-CONN operand).

RECOVERY-START = *BY-OPERATOR

Partner monitoring begins with the establishment of the MSCF connection.

The local system does not start fail reconfiguration for this partner until the action is confirmed by systems support.

RECOVERY-START = *CONSISTENT-BY-OPERATOR

Same behavior as with RECOVERY-START=*BY-OPERATOR. Additionally, the partner system does not start fail reconfiguration for the local system until the action is confirmed by the Live Monitor or systems support.

RECOVERY-START = *SECURE

Partner monitoring begins with the establishment of the MSCF connection.

The local system starts fail reconfiguration for the partner automatically only if failure of the partner was detected or confirmed by the Live Monitor.

The partner starts fail reconfiguration for the local system automatically only if failure was detected or confirmed by the Live Monitor.

CONNECTION-TYPE = *LOOSELY-COUPLED

LCS (loosely-coupled system) does not support shared pubset or XCS network operation.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
	1	CMD2201	Parameter error
	2	CMD2202	MSCF subsystem not defined
	3	CMD2203	Installation error
	32	MCS0025	Internal error (bourse call, memory request)
	64	CMD0216	Necessary privileges not available
	64	MCS0009	Local password invalid or local system specified as partner
	65	CMD2241	MSCF subsystem is not loaded
	65	MCS0003	MSCF subsystem currently being loaded
	65	MCS0004	MSCF subsystem currently being unloaded
	66	CMD2242	No connection to the MSCF subsystem
	128	CMD2280	MSCF subsystem temporarily unavailable
	129	CMD2281	MSCF subsystem unavailable for an extended period
	130	CMD2282	MSCF subsystem unavailable for an undefined period

Notes

- The commands START- and STOP-MSCF-CONNECTION do not modify entries in the (static) MSCF configuration file.
- If an MSCF connection is lost due to an error, MSCF automatically starts a connection recovery. If this is successful the message MCS0018 is issued on the console and data communication continues. Remote tasks may be lost during connection recovery and may cause an error on the caller.
- The system automatically and if necessary repeatedly tries to establish a new connection, if connection recovery was not successful. The attempts are stopped when there is no hope of reestablishing a connection (e.g. the partner has failed or is unknown to BCAM). During the connection recovery attempts there is no connection to the partner, so remote tasks are rejected.
- If the MSCF connection fails due to BCAM, a new MSCF connection is not automatically established if the BCAM network is re-established through external influences (e.g. BCIN-/BCACT command). The START-MSCF-CONNECTION command has to be entered again. If the START-MSCF-CONNECTION command is entered anew values other than the standard value *NONE for the REMOTE-PASSWORD and LOCAL-PASSWORD operands have to be re-specified. However, the current value of the NUMBER-OF-CTRL-CONN and RECOVERY-START monitoring parameters remains in place. This is also the case following an explicit connection establishment.

START-SUBSYSTEM

Activate subsystem

Description status:	DSSM V4.3A
Functional area:	Subsystem management
Domain:	SYSTEM-MANAGEMENT
Privileges:	OPERATING SUBSYSTEM-MANAGEMENT
Routing code:	R

Function

The following command description only applies to the MSCF subsystem. A full description of the command can be found in the “Commands” manual [10].

The following information from the dynamic subsystem catalog is used for activating the MSCF subsystem:

- data concerning the loading and linking of the subsystem
- data concerning initialization/deinitialization and termination of job relations
- data concerning entry points, secondary components, and operational dependencies (see the relevant SSCM statements in the "Subsystem Management" manual [19])

The command is rejected if:

- the subsystem could not be found in the dynamic subsystem catalog
- the subsystem is already active
- subsystems on which the subsystem to be activated depends are not loaded
- a necessary file (e.g. message file, library) is missing

An appropriate message informs systems support staff whether the command has been accepted or rejected.

Format (excerpt)

START-SUBSYSTEM
<p>SUBSYSTEM-NAME = MSCF</p> <p>,SUBSYSTEM-PARAMETER = *NONE / <c-string 1..254></p> <p> <c-string 1..254></p> <p> FILE-NAME = *NONE / *SYSTEM / <filename 1..54></p> <p> ,NUMBER-OF-SERVERS = *STD / <integer 2..10></p> <p>,RESET = *NO / *YES</p> <p>,SYNCHRONOUS = *NO / *YES</p>

Operands**SUBSYSTEM-NAME = MSCF**

Name of the subsystem to be activated.

SUBSYSTEM-PARAMETER = *NONE / C' ... '

Specifies whether special parameters which can only be evaluated by the specified subsystem (MSCF) are to be processed.

SUBSYSTEM-PARAMETER = *NONE

Default value. The file specified in the DSSM catalog as the subsystem file is interpreted as the MSCF configuration file. If no file is specified in the DSSM catalog, only the MSCF subsystem is loaded. No connections are established.

Any connections required must be established explicitly via the START-MSCF-CONNECTION command.

SUBSYSTEM-PARAMETER = <c-string 1..254>

Special parameters that can only be evaluated by the MSCF subsystem are to be processed. The following must be specified as C strings (C'...').

FILE-NAME = *NONE / *SYSTEM / <filename 1..54>

Name of the MSCF configuration file to be evaluated and processed when the subsystem is loaded. The MSCF configuration file contains the MSCF configuration parameters and a description of the MSCF connections to be established automatically (see [page 115](#)).

If the FILE-NAME parameter is not specified explicitly, the file specified in the DSSM catalog as the subsystem file is used as the configuration file.

FILE-NAME = *NONE

No MSCF configuration file is to be used. Any configuration file specified as the subsystem file in the DSSM catalog is also ignored.

The MSCF subsystem is loaded. No connections are established; they must be set up explicitly via the START-MSCF-CONNECTION command.

FILE-NAME = *SYSTEM

The MSCF configuration file with the name SYS.MSCF-CONFIGURATION is to be used. The file is sought under the user ID TSOS and, if not found, under the system default ID (secondary read). Any configuration file specified as the subsystem file in the DSSM catalog is ignored.

FILE-NAME = <filename 1..54>

The specified file is to be used as the MSCF configuration file. If no user ID is specified in the file name, the user ID \$TSOS and, if not found, the system default ID (secondary read) are used.

Any configuration file specified as the subsystem file in the DSSM catalog is ignored.

NUMBER-OF-SERVERS = *STD / <integer 2..10>

Number of MSCF server tasks to be made available on this processor (at start time). This number can be between 2 and 10, depending on the operand value. If this operand is not specified, the value set in the configuration file applies (see SET-MSCF-ENVIRONMENT command). A value defined via this operand has priority over any specification in the configuration file. The number may increase in the course of an MSCF session.

NUMBER-OF-SERVERS = *STD

Four server tasks are made available.

NUMBER-OF-SERVERS = <integer 2..10>

The specified number of server tasks is made available.

RESET = *NO / *YES

Determines the mode and urgency of command processing.

RESET = *NO

If the respective subsystem is being deactivated, the command is rejected until this blocking process has been completed.

RESET = *YES

The command is accepted irrespective of any outstanding deactivation process, and the subsystem or some of its components are initialized.

The version parameter is mandatory for this operand.

SYNCHRONOUS = *NO / *YES

Enables synchronous or asynchronous processing to be selected.

SYNCHRONOUS = *NO

The command is to be processed asynchronously. As soon as the command is entered, it is possible to enter further commands. Error messages relating to the execution of the command are output on the console.

SYNCHRONOUS = *YES

Further commands cannot be entered until the command is executed. Any error messages regarding the execution of the command are output.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
1	0	CMD0001	No action required
	32	ESM0224	Command not processed
	32	ESM0228	Command terminated abnormally

Note

To ensure a high degree of parallelism and data integrity, time-consuming management functions are not performed under the control of the calling task, rather are transferred to a DSSM task.

As a rule, the requested function is checked *synchronously* (i.e. involving a wait state for the calling task). The actual processing, however, is performed by DSSM *asynchronously* and independently of the calling task.

STOP-MSCF-CONNECTION

Clear down MSCF connection to partner system

Description status:	MSCF V17.0A
Functional area:	Pubset and MRSCAT management
Domain:	MULTI-CATALOG-AND-PUBSET-MGMT
Privileges:	TSOS OPERATING
Routing code:	R

Function

The STOP-MSCF-CONNECTION command clears down a connection to a partner system.

Format

STOP-MSCF-CONNECTION
PROCESSOR-NAME = <alphanum-name 1..8> , FORCED = * NO / * YES

Operands

PROCESSOR-NAME = <alphanum-name 1..8>

Name of the partner system for which a connection is to be cleared down.

FORCED = *NO / *YES

Specifies whether or not execution of the command is to be forced.

Before the connection to the partner is cleared down, MSCF checks whether this connection is still required by the local system. It is required if both systems are sharing a pubset or one of the two systems is the master system. In this situation, the connection clear-down is rejected, but can be forced by specifying **FORCED = *YES**. The connection should subsequently be re-established with the **START-MSCF-CONNECTION** command.

The **FORCED = *YES** operand should as a rule be used on a restricted basis and only in acute error situations.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
	1	CMD2201	Parameter error
	2	CMD2202	MSCF subsystem not defined
	3	CMD2203	Installation error
	32	MCS0025	Internal error (REQM, bourse call)
	64	CMD0216	User is not privileged
	64	MCS0009	Local system specified as partner
	64	MCS0021	Command can only be executed in conjunction with the FORCED=*YES operand
	65	CMD2241	MSCF subsystem is not loaded
	65	MCS0003	MSCF subsystem currently being loaded
	65	MCS0004	MSCF subsystem currently being unloaded
	66	CMD2242	No connection to the MSCF subsystem
	128	CMD2280	MSCF subsystem temporarily unavailable
	129	CMD2281	MSCF subsystem unavailable for an extended period
	130	CMD2282	MSCF subsystem unavailable for an undefined period

STOP-SUBSYSTEM

Deactivate subsystem

Description status:	DSSM V4.3A
Functional area:	Subsystem management
Domain:	SYSTEM-MANAGEMENT
Privileges:	OPERATING SUBSYSTEM-MANAGEMENT
Routing code:	R

Function

The following command description only applies to the MSCF subsystem. A complete description of the command can be found in the “Commands” manual [10].

Systems support can use this command to deactivate the MSCF subsystem. The command has the following results:

- MSCF is locked for all new callers
- MSCF is unloaded
- All resources allocated are released

The STOP-SUBSYSTEM command is rejected if:

- The MSCF subsystem is not found in the dynamic subsystem catalog
- Subsystems that are already active or are in the process of activation are dependent on the MSCF subsystem to be deactivated
- Subsystems that are already active or are in the process of activation have linkage relations with the MSCF subsystem to be deactivated

Format (extract)

STOP-SUBSYSTEM

SUBSYSTEM-NAME = MSCF

,VERSION = *STD / <c-string 4..4> / <c-string 7..7>

,SUBSYSTEM-PARAMETER = *NONE / 'FORCED = NO' / 'FORCED = YES'

,FORCED = *NO / *YES

,SYNCHRONOUS = *NO / *YES

Operands

SUBSYSTEM-NAME = MSCF

The MSCF subsystem is to be deactivated.

VERSION = *STD / <c-string 4..4> / <c-string 7..7>

Version number of the MSCF subsystem. The format specified here must be identical to the format used in the subsystem definition (release and correction level are either mandatory or must not be specified, see SDF notational conventions in the “Commands” manual [10]).

VERSION = *STD

If only one version of the subsystem is loaded, that version is chosen. If there are several appropriate versions, a version number must be defined.

SUBSYSTEM-PARAMETER = *NONE / 'FORCED = NO / YES'

Specifies whether special parameters are to be evaluated by MSCF.

'FORCED = NO'

MSCF can reject execution of the STOP-SUBSYSTEM MSCF command if one or more shared pubsets are imported on the local processor.

'FORCED = YES'

Forces execution of the STOP-SUBSYSTEM MSCF command, even if MSCF has already rejected execution of the command. If 'FORCED=YES' is specified, the local processor is disconnected from an XCS or shared pubset network.

FORCED = *NO / *YES

Determines the mode and urgency of command processing.

FORCED = *NO

MSCF is not terminated until the tasks linked to the MSCF subsystem (MSCF holder task, XCM and watchdog task TSN: MXCM, WDGS, MSCF server tasks) are processed and terminated normally.

FORCED = *YES

Not permitted for MSCF.

SYNCHRONOUS = *NO / YES

Enables synchronous or asynchronous processing to be selected.

SYNCHRONOUS = *NO

The command is to be processed asynchronously. As soon as the command is entered, it is possible to enter further commands. No error messages relating to the execution of the command are output.

SYNCHRONOUS = *YES

The command must first be executed before any other commands can be entered. Any error messages relevant to command execution are output.

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command executed without errors
1	0	CMD0001	No action required
	32	ESM0224	Command not processed
	32	ESM0228	Command terminated abnormally

Notes

- There are usually a variety of relationships between subsystems (dependencies, load relations).

In order to guarantee the performance of individual subsystems, these relationships must be taken into account. DSSM attempts to avoid potential conflicts that may arise from the user's requirements and therefore rejects certain commands. Illegal actions, such as the unloading of subsystems on which active subsystems depend, are not executed.

However, if the user also generates complex subsystems using the operand CHECK-REFERENCE=*NO (see the SSCM statement SET-SUBSYSTEM-ATTRIBUTES), DSSM executes the requested functions *despite* possible conflicts.
- To ensure a high degree of parallelism and data integrity, time-consuming management functions are not performed under the control of the calling task, rather are transferred to a DSSM task.

As a rule, only the checking of the requested function is carried out *synchronously* (i.e. involving a wait state for the calling task). The actual processing is carried out by DSSM *asynchronously* and independently of the calling task.

10 Macros

This chapter contains the following:

- an overview of the macros in HIPLEX MSCF described in this manual
- the description of the HIPLEX MSCF macros in alphabetical order

For a detailed description of the macro syntax please refer to the “Executive Macros” manual [12].

Overview of the macros

Macro	Function	Page
DPSHSHP	Output information on current shared pubset configuration	310
MCSINFO	Output information on current MSCF configuration	317

DPSHSHP

Output information on current shared pubset configuration

Macro description

DPSHSHP provides information about the current sharers of a shared pubset. The requested information is placed in an output area. The user must provide an output area of adequate size with the necessary access rights by means of the OUTAREA operand.

The DPSHSHP interface supplies information that corresponds to the output of the SHOW-SHARED-PUBSET command.

If INFO=*ALL, INFO=*XCS or INFO=*CCS is specified, the storage space required for the output depends on the number of existing shared pubsets and the number of sharers per shared pubset. 24 bytes are required for one sharer of a shared pubset (a shared pubset may have a maximum of 16 sharers).

If the DPSHSHP interface call does not return any output information due to an unexpected watchdog mechanism return code, this return code is placed in the parameter list as output information (in the DPSHWDRC (“watchdog return code”) field). In this case, the return code field of the DPSHSHP interface contains the main return code DPSHINTE (“internal error”).

The output area provided by the interface caller must be at least long enough to accommodate a single sharer of a single shared pubset. The following is required for this: 8 bytes for the output area header, 8 bytes for the header of the shared pubset configuration and 24 bytes for at least one sharer (i.e. 40 bytes in all). If the output area is dimensioned with a smaller length, this is acknowledged with the main return code DPSHOISF (“not minimal output length”).

If the output area specified by the user is too small to accommodate the returned information, the currently needed value for the length of the output area DPSHRQLG (“requested length of output area”) is returned as an output parameter. In addition, a return code is set that draws attention to this situation (main return code DPSHOTSH (“not enough space for info”).

Dependencies between the operands

If INFO=*ONEPVS is specified, the identification of a pubset must also be specified using the input parameter PUBSET.

Format and operands

Operation	Operands
DPSHSHP	MF=D / C / L / M / E ,XPAND=PARAM / OUTPUT ,VERSION= <u>1</u> / <integer 1..3> ,INFO= <u>*ALL</u> / *ONEPVS / *XCS / *CCS ,PUBSET=<var: char:4> / (<reg: pointer>) ,OUTAREA=(adresse, länge) adresse: <var:pointer> / (<reg: pointer>) länge: <integer 1..32767> / <var:int:4> / (<reg:int:4>) [,PREFIX= <u>D</u> / p] [,MACID= <u>PSH</u> / macid] [,PARAM=<name 1..27>]

MF

The MF operand defines the type of macro generation.

Depending on the operand values for MF, a distinction is made between five forms of the macro call:

Operands
MF=L [,operand ₁ ,...,operand _n]
MF=M ,operand ₁ ,...,operand _n [,PREFIX= <u>M</u> / p] [,MACID= <u>PSH</u> / macid]
MF=D ,XPAND=PARAM/OUTPUT[,PREFIX= <u>M</u> / p]
MF=C ,XPAND=PARAM/OUTPUT[,PREFIX= <u>M</u> / p] [,MACID= <u>PSH</u> / macid]
MF=E [,PARAM=<name 1..27>]

For information on the MF operand values, see the manual "Executive Macros" [12].

XPAND = PARAM / OUTPUT

Defines whether or not a DSECT is to be generated. Specification only in conjunction with MF=D/C.

PARAM

The structure description of the parameter list is created.

OUTPUT

The structure description of the output area is created.

VERSION = 1 / <integer 1..3>

Defines the desired interface version. Interface version 1 (currently the only version) is supported as of MSCF V15.0.

1

Default value. Interface version 1 is generated.

<integer 1..3>

Direct specification of the desired interface version.

INFO = *ALL / *ONEPVS / *XCS / *CCS

Specifies which shared pubsets information was requested for.

***ALL**

Information about all existing shared pubsets is to be provided.

***ONEPVS**

Requests information about a selected shared pubset. The pubset is identified in the PUBSET operand.

***XCS**

Outputs information about all XCS pubsets.

***CCS**

Outputs information about all CCS pubsets.

PUBSET =

Only evaluated with INFO=*ONEPVS.

Identifies a pubset.

OUTAREA = (address, length)

Defines the output area in which the processor information is output.

address: <var: pointer> / (<reg: pointer>)

<var: pointer>

The output area must be aligned on word boundaries.

The following two alternatives are possible:

- address of the output area as direct address specification in the form A(area)
- name of a field with the address of the output area. Specification only in conjunction with MF=M

(<reg: pointer>)

Register with the address of the output area. Specification only in conjunction with MF=M.

length: <integer 1..32767> / <var: int:4> / (<reg: int:4>)

Defines the length of the output area. If the output area specified is too small, the required size is stored in the Input_Param.Area_Length field of the parameter list.

<integer 1..32767>

Length of the output area in bytes as direct specification.

In Assembler a symbolic specification must be enclosed in square brackets.

<var: int:4>

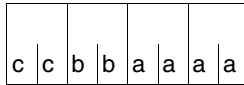
Name of a field with the length specification as an integer; field length = 4 bytes. Specification only in conjunction with MF=M.

<reg: int:4>

Register containing the length specification for the output area. Specification only in conjunction with MF=M.

Return codes

Standard header:



The following return code regarding the success of the execution of the DPSHSHP macro is transferred in the standard header (cc = SUBCODE2, bb = SUBCODE1, aaaa = MAINCODE):

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'00'	X'0000'	No errors
	X'40'	X'0001'	Parameter list not accessible
	X'40'	X'0002'	Output address not accessible
	X'40'	X'0003'	Output address not on word boundary
	X'40'	X'0004'	INFO parameter is invalid
	X'40'	X'0005'	PUBSET parameter is invalid
	X'40'	X'0006'	Not enough storage space for requested information
	X'40'	X'0007'	Not enough storage space for minimum output information
	X'40'	X'0008'	No shared pubsets exist
	X'40'	X'0009'	MSCF subsystem not available
	X'20'	X'000A'	Internal error
	X'01'	X'FFFF'	Interface error, wrong unit or function number
	X'03'	X'FFFF'	Version not supported

Table 8: DPSHSHP macro: return codes

Layout of the parameter area

```

DPSHSHP MF=D,XPAND=PARAM,VERSION=1
1      MFTST MF=D,PREFIX=D,MACID=PSH,ALIGN=F,                                C
1      DMACID=PSH,SUPPORT=(E,D,C,M,L),DNAME=DPSHPAR
2 DPSHPAR DSECT ,
2      *,##### PREFIX=D, MACID=PSH #####
1 *    DPSHSHP parameter area
1 DPSHFHDR FHDR MF=(C,MCSI),EQUATES=NO                                     Standardheader
2 DPSHFHDR DS    OA
2 DPSHFHE  DS    OXL8              0 GENERAL PARAMETER AREA HEADER
2 *
2 DPSHFID  DS    OA                0 INTERFACE IDENTIFIER
2 DPSHFCTU DS    AL2              0 FUNCTION UNIT NUMBER
2 *
2 *                                BIT 15    HEADER FLAG BIT,
2 *                                MUST BE RESET UNTIL FURTHER NOTICE
2 *                                BIT 14-12 UNUSED, MUST BE RESET
2 *                                BIT 11-0   REAL FUNCTION UNIT NUMBER

```

2	DPSHFCT	DS	AL1	2	FUNCTION NUMBER
2	DPSHFCTV	DS	AL1	3	FUNCTION INTERFACE VERSION NUMBER
2	*				
2	DPSHRET	DS	0A	4	GENERAL RETURN CODE
2	DPSHSRET	DS	0AL2	4	SUB RETURN CODE
2	DPSHSR2	DS	AL1	4	SUB RETURN CODE 2
2	DPSHSR1	DS	AL1	5	SUB RETURN CODE 1
2	DPSHMRET	DS	0AL2	6	MAIN RETURN CODE
2	DPSHMR2	DS	AL1	6	MAIN RETURN CODE 2
2	DPSHMR1	DS	AL1	7	MAIN RETURN CODE 1
2	DPSHFHL	EQU	8	8	GENERAL OPERAND LIST HEADER LENGTH
2	*				
1	*	main return codes			
1	DPSHNOER	EQU	0		no error
1	DPSHPLNA	EQU	1		parameterlist not accessible
1	DPSHONAC	EQU	2		output area not accessible
1	DPSHONAL	EQU	3		output area not aligned
1	DPSHINFI	EQU	4		invalid INFO parameter
1	DPSHINVP	EQU	5		invalid PUBSET parameter
1	DPSHOTSH	EQU	6		not enough space for info
1	DPSHOISF	EQU	7		not minimal output length
1	DPSHNSHP	EQU	8		no shared pubsets
1	DPSHMNAV	EQU	9		subsystem MSCF not available
1	DPSHINTE	EQU	10		internal error
1	*				
1	DPSHINFO	DS	FL1		par.INFO
1	*	values for operand INFO			
1	DPSHALLP	EQU	1		info for all shared pubsets
1	DPSHONEP	EQU	2		info for one shared subset
1	DPSHXCSP	EQU	3		info for all XCS pubsets
1	DPSHCCSP	EQU	4		info for all CCS pubsets
1	*				
1	DPSHRSV1	DS	XL3		reserved - first
1	DPSHPBID	DS	CL4		par. PUBSET
1	DPSHARAD	DS	A		address of output area
1	DPSHARLG	DS	F		length of output area
1	DPSHRSV2	DS	XL4		reserved - second
1	DPSHRQLG	DS	F		req. length of output area
1	DPSHWDRG	DS	F		watchdog return code
1	DPSH#	EQU	*-DPSHFHDR		

Layout of the output area

```

DPSHSHP MF=D,XPAND=OUTPUT
1 MFTST MF=D,PREFIX=D,MACID=PSH,ALIGN=F, C
1 DMACID=PSH,SUPPORT=(E,D,C,M,L),DNAME=PSHOUT
2 DPSHOUT DSECT ,
2 *,##### PREFIX=D, MACID=PSH #####
1 * output area layout
1 DPSHNSPV DS F number of spvs
1 DPSHTLTH DS F total length
1 *
1 DPSHCONF DS 0XL392 configuration entry
1 DPSHNOSH DS H number of sharer
1 DPSHRESV DS XL2 reserved bytes
1 DPSHSPVS DS CL4 spvs catid
1 DPSHSHAR DS 0XL24 sharer data of pubset
1 DPSHHNAM DS CL8 sharer host name
1 DPSHHCTI DS CL4 sharer home catid
1 DPSHBSVR DS CL3 sharer bs version
1 DPSHBSSV DS CL1 sharer bs subversion
1 DPSHESYS DS CL3 sharer ext sysid
1 DPSHSYSI DS X sharer sysid
1 DPSHSTYP DS FL1 sharer type
1 * values for output SHARER TYPE
1 DPSHTUDF EQU 0 undefined sharer type
1 DPSHTMAS EQU 1 sharer is pubset master
1 DPSHTSLV EQU 2 sharer is pubset slave
1 DPSHTBAM EQU 3 sharer is pubset backup master
1 DPSHSTAT DS FL1 sharer state
1 * values for output SHARER STATE
1 DPSHSUDF EQU 0 undefined sharer state
1 DPSHSACT EQU 1 sharer state active
1 DPSHSCRS EQU 2 sharer state crashed
1 DPSHSEXC EQU 3 sharer state excat
1 DPSHSDPR EQU 4 sharer state disc prot error
1 DPSHSHUT EQU 5 sharer state shutdown
1 DPSHSMCH EQU 6 sharer state master change
1 DPSHSRDE EQU 7 sharer state disc read error
1 DPSHSWRE EQU 8 sharer state disc write error
1 DPSHRES3 DS XL2 reserved - third
DS 15XL24
1 DPSHSHAR# EQU 16
1 *
1 DPSHOUTPUT# EQU *-DPSHNSPV

```

MCSINFO

Output information on current MSCF configuration

Macro description

The MCSINFO macro call provides information on the current MSCF configuration associated with the processor. If information is reserved for privileged callers, any nonprivileged callers receive the information "UNDEFINED".

The desired information is stored in an output area. The user must define the OUTAREA operand to ensure that the output area is large enough and that the necessary access rights are assigned. If the area is too small, a corresponding return code is supplied and the actual length required is stored in the parameter file so that a suitable output area can be requested.

If INFO=*ALLHOSTS and INFO=*XCShOSTS are specified, the memory area required for the output depends on the number of processors connected to the network.

Format and operands

Operation	Operands
MCSINFO	MF=D / C / L / M / E ,XPAND=PARAM / OUTPUT ,VERSION= <u>4</u> / <integer 1..3> ,INFO= <u>*NOHOST</u> / *ONEHOST / *XCShOSTS / *ALLHOSTS ,KEY= <u>*NONE</u> / *HOST / *PROC / *SYSID ,HOST=<var: char:8> / (<reg: pointer>) ,PROC=<var: char:8>/ (<reg: pointer>) ,SYSID=<var: char:3>/ (<reg: pointer>) ,OUTAREA=(address, length) address: <var:pointer> / (<reg: pointer>) length: <integer 1..32767> / <var:int:4> / (<reg:int:4>) [,PREFIX= <u>M</u> / p] [,MACID= <u>CSI</u> / macid] [,PARAM=<name 1..27>]

MF

The MF operand defines the type of macro generation.

Depending on the operand values for MF, a distinction is made between five forms of the macro call:

Operands
MF=L [,operand ₁ ,...,operand _n]
MF=M ,operand ₁ ,...,operand _n [,PREFIX= <u>M</u> / p] [,MACID= <u>CSI</u> / macid]
MF=D ,XPAND=PARAM/OUTPUT[,PREFIX= <u>M</u> / p]
MF=C ,XPAND=PARAM/OUTPUT[,PREFIX= <u>M</u> / p] [,MACID= <u>CSI</u> / macid]
MF=E [,PARAM=<name 1..27>]

For information on the MF operand values, see the manual "Executive Macros" [12].

XPAND = PARAM / OUTPUT

Defines whether or not a DSECT is to be generated. Specification only in conjunction with MF=D/C.

PARAM

The structure description of the parameter list is created.

OUTPUT

The structure description of the output area is created.

VERSION = 4 / <integer 1..3>

Defines the desired interface version. Interface version 1 is supported as of MSCF V12.0 and MSCF V13.0, version 2 is supported as of MSCF V13.0, version 3 as of MSCF V14.0, version 4 as of MSCF V17.0.

4

Default value. Interface version 4 is generated.

<integer 1..3>

Direct specification of the desired interface version.

INFO = *NOHOST / *ONEHOST / *XCSTHOSTS / *ALLHOSTS

Defines whether additional information is requested on processors.

***NOHOST**

Default value. Parameters of the local processor are stored in the parameter list. This information is also supplied for the following operand values of INFO.

***ONEHOST**

Information is stored in the output area on the processor defined by the KEY operand.

***XCShOSTS**

Information is stored in the output area on all XCS partners.

***ALLHOSTS**

Information is stored in the output area on all processors.

KEY = *NONE / *HOST / *PROC / *SYSID

Only evaluated if INFO=*ONEHOST is specified.

Defines whether the processor on which information is requested is specified by its host or processor name or by its SYSID.

***NONE**

Preset value if INFO is not equal to *ONEHOST.

No processor is selected.

***HOST**

The processor on which information is requested is defined by the HOST operand.

***PROC**

The processor on which information is requested is defined by the PROC operand.

***SYSID**

The processor on which information is requested is defined by the SYSID operand.

HOST = <var: char:8> / (<reg: pointer>)

Name of the processor on which information is requested. The name can be up to 8 characters long. Specification only in conjunction with KEY=*HOST.

<var: char:8>

Name of a field containing the host name of the processor on which information is requested.

<reg: pointer>

Register with the address of a field containing the host name of the processor on which information is requested. Specification only in conjunction with MF=M.

PROC = <var: char:8> / (<reg: pointer>)

BCAM name of the processor on which information is requested. The name can be up to 8 characters long. Specification only in conjunction with KEY=*PROC.

<var: char:8>

Name of a field containing the name of the connection to the processor on which information is requested.

(<reg: pointer>)

Register with the address of a field containing the name of the processor on which information is requested. Specification only in conjunction with MF=M.

SYSID = <var: char:3> / (<reg: char:3>)

SYSID of the processor on which information is requested. The name can be up to 3 characters long. Specification only in conjunction with KEY=*SYSID.

<var: char:3>

Name of a field containing the SYSID of the processor on which information is requested.

(<reg: pointer>)

Register with the address of a field containing the SYSID of the processor on which information is requested. Specification only in conjunction with MF=M.

OUTAREA = (address, length)

Defines the output area in which the processor information is output.

address: <var: pointer> / (<reg: pointer>)**<var: pointer>**

The output area must be aligned on word boundaries.

The following two alternatives are possible:

- address of the output area as direct address specification in the form A(area)
- name of a field with the address of the output area. Specification only in conjunction with MF=M

(<reg: pointer>)

Register with the address of the output area. Specification only in conjunction with MF=M.

length: <integer 1..32767> / <var: int:4> / (<reg: int:4>)

Defines the length of the output area. If the output area specified is too small, the required size is stored in the Input_Param.Area_Length field of the parameter list.

<integer 1..32767>

Length of the output area in bytes as direct specification.

In Assembler a symbolic specification must be enclosed in square brackets.

<var: int:4>

Name of a field with the length specification as an integer; field length = 4 bytes. Specification only in conjunction with MF=M.

<reg: int:4>

Register containing the length specification for the output area. Specification only in conjunction with MF=M.

Return codes

Standard
header:

c	c	b	b	a	a	a	a
---	---	---	---	---	---	---	---

The following return code regarding the success of the execution of the MCSINFO macro is transferred in the standard header (cc = SUBCODE2, bb = SUBCODE1, aaaa = MAINCODE):

X'cc'	X'bb'	X'aaaa'	Explanation
	X'00'	X'0000'	No errors
	X'01'	X'0001'	Illegal operand specified
X'01'	X'01'	X'0001'	Illegal INFO operand
X'02'	X'01'	X'0001'	Illegal KEY operand
X'03'	X'01'	X'0001'	Illegal HOST operand
X'04'	X'01'	X'0001'	Illegal PROC operand
X'05'	X'01'	X'0001'	Illegal SYSID operand
X'06'	X'01'	X'0001'	Illegal CALLER parameter
X'07'	X'01'	X'0001'	Illegal OUTAREA operand
	X'20'	X'0004'	Internal error
X'01'	X'20'	X'0004'	Internal error when requesting memory
X'02'	X'20'	X'0004'	Internal error when calling configuration management
	X'40'	X'0006'	Illegal output address
	X'40'	X'0007'	Output address not aligned on word boundaries
	X'40'	X'0009'	Insufficient memory for requested information
	X'40'	X'000A'	Illegal address of parameter list
	X'40'	X'0040'	Specified host name unknown
	X'40'	X'0041'	Specified processor name unknown
	X'40'	X'0042'	Specified SYSID unknown
	X'01'	X'FFFF'	Interface error, wrong unit or function number
	X'03'	X'FFFF'	Version not supported

Table 9: MCSINFO macro: return codes

Layout of the parameter area

```

PARAM    MCSINFO MF=D,XPAND=PARAM,VERSION=4
1 PARAM    MFTST MF=D,PREFIX=M,MACID=CSI,ALIGN=F,
1          DMACID=CSI,SUPPORT=(E,D,C,M,L),DNAME=CSI_PL_MDL
2 PARAM    DSECT ,
2          *,##### PREFIX=M, MACID=CSI #####
1 * values for sub return codes 2
1 MCSIINFO_PARAM EQU 1 INFO-Parameter
1 MCSIKEY_PARAM EQU 2 KEY-Parameter
1 MCSIHOST_PARAM EQU 3 HOST-Parameter
1 MCSIPROC_PARAM EQU 4 PROC-Parameter
1 MCSISYSID_PARAM EQU 5 SYSID-Parameter
1 MCSICALLER_PARAM EQU 6 CALLER-Parameter
1 MCSIOUTAREA_PARAM EQU 7 OUTAREA-Parameter
1 MCSIGETMEM EQU 1 at $GETMEMI interface
1 MCSIXCS_SHOW EQU 2 at $MCXSHCI interface
1 *
1 * MCSINFO parameter area
1 MCSIFHDR FHDR MF=(C,MCSI),EQUATES=NO Standardheader
2 MCSIFHDR DS OA
2 MCSIFHE DS OXL8 0 GENERAL PARAMETER AREA HEADER
2 *
2 MCSIIFID DS OA 0 INTERFACE IDENTIFIER
2 MCSIFCTU DS AL2 0 FUNCTION UNIT NUMBER
2 * BIT 15 HEADER FLAG BIT,
2 * MUST BE RESET UNTIL FURTHER NOTICE
2 * BIT 14-12 UNUSED, MUST BE RESET
2 * BIT 11-0 REAL FUNCTION UNIT NUMBER
2 MCSIFCT DS AL1 2 FUNCTION NUMBER
2 MCSIFCTV DS AL1 3 FUNCTION INTERFACE VERSION NUMBER
2 *
2 MCSIRET DS OA 4 GENERAL RETURN CODE
2 MCSISRET DS OAL2 4 SUB RETURN CODE
2 MCSISR2 DS AL1 4 SUB RETURN CODE 2
2 MCSISR1 DS AL1 5 SUB RETURN CODE 1
2 MCSIMRET DS OAL2 6 MAIN RETURN CODE
2 MCSIMR2 DS AL1 6 MAIN RETURN CODE 2
2 MCSIMR1 DS AL1 7 MAIN RETURN CODE 1
2 MCSIFHL EQU 8 8 GENERAL OPERAND LIST HEADER LENGTH
2 *
1 * main return codes
1 MCSISUCCESSFUL EQU 0 successfully processed
1 MCSIPARAM_ERROR EQU 1 invalid parameter specified
1 MCSIINTERFACE_ERROR EQU 65535
1 * Interface error, wrong unit
1 * or function number specified
1 MCSIVERSION_ERROR EQU 65535

```

```

1 *                               Version not supported
1 MCSIINTERNAL_ERROR             EQU 4   Internal error
1 MCSIINV_OUTPUT_ADDR           EQU 6   invalid output address
1 *                               specified
1 MCSIINV_OUTPUT_ALIGN          EQU 7   output address not word
1 *                               aligned
1 MCSIAREA_TO_SHORT             EQU 9   not enough space for all
1 *                               requested information
1 MCSIINV_PARAM_ADDR            EQU 10  invalid address of
1 *                               parameterlist
1 MCSIHOST_UNKNOWN              EQU 64  specified host is unknown
1 MCSIPROCESSOR_UNKNOWN        EQU 65  specified processor is
1 *                               unknown
1 MCSISYSID_UNKNOWN            EQU 66  specified sysid is unknown
1 *
1 *
1 MCSIINPUT_PARAM               DS     0XL32
1 *                               INPUT PARAMETER
1 MCSIINFO                       DS     FL1  parameter INFO
1 *   values for operand INFO
1 MCSINO_HOST                    EQU 1   no host information requested
1 MCSIONE_HOST                   EQU 2   information requested for
1 *                               host is specified by operand
1 *                               *KEY
1 MCSIXCS_HOSTS                  EQU 3   information requested for all
1 *                               hosts in current XCS
1 MCSIALL_HOSTS                  EQU 4   information requested for all
1 *                               connected hosts
1 *
1 MCSIKEY                         DS     FL1  parameter KEY
1 *   values for operand KEY
1 MCSIHOST_GIVEN                 EQU 1   host specified by operand
1 *                               HOST
1 MCSIPROCESSOR_GIVEN            EQU 2   host specified by operand
1 *                               PROC
1 MCSISYSID_GIVEN                EQU 3   host specified by operand
1 *                               SYSID
1 MCSINO_KEY_GIVEN               EQU 4   no host specified
1 *
1 MCSISYSID_IN                   DS     CL3  parameter SYSID
1 MCSIRESERVED04                 DS     XL3  reserved
1 MCSIHOST_NAME_IN               DS     CL8  parameter HOST
1 MCSIPROCESSOR_NAME_IN          DS     CL8  parameter PROCESSOR
1 MCSIAREA_ADDRESS               DS     A    address of output area
1 MCSIAREA_LENGTH                DS     F    length of output area
1 *
1 *
1 MCSIOUTPUT_PARAM               DS     0XL120

```

```

1 *                               OUTPUT PARAMETER
1 *
1 MCSILOCAL_ATTRIBUTES           DS    0XL112
1 *                               LOCAL ATTRIBUTES, see command
1 *                               SHOW-MSCF-ENVIRONMENT for
1 *                               meaning of fields
1 MCSIENV_HN                     DS    CL8    HOST NAME
1 MCSIENV_PN                     DS    CL8    PROCESSOR NAME
1 *
1 MCSI1ENV_HT                    DS    0XL14
1 *                               HOST START DATE/TIME
1 MCSI1YEAR                      DS    CL4    yyyy
1 MCSI1MONTH                     DS    CL2    mm
1 MCSI1DAY                       DS    CL2    dd
1 MCSI1HOUR                      DS    CL2    hh
1 MCSI1MINUTE                    DS    CL2    mm
1 MCSI1SECOND                    DS    CL2    ss
1 *
1 MCSIXCS_NAME                   DS    CL8    XCS NAME
1 *
1 MCSI2ENV_XT                    DS    0XL14
1 *                               XCS CREATION DATE/TIME
1 MCSI2YEAR                      DS    CL4    yyyy
1 MCSI2MONTH                     DS    CL2    mm
1 MCSI2DAY                       DS    CL2    dd
1 MCSI2HOUR                      DS    CL2    hh
1 MCSI2MINUTE                    DS    CL2    mm
1 MCSI2SECOND                    DS    CL2    ss
1 *
1 MCSISERVER_TASK_LIMIT         DS    F      SERVER TASK LIMIT
1 MCSICURRENT_NUM_SERVERS       DS    F      NUMBER OF SERVERS
1 MCSICURRENT_NUM_FREE_SERVERS DS    F      NUMBER OF FREE SERVERS
1 MCSIXCS_RECONF_NUMBER         DS    F      XCS RECONFIGURATION NUMBER
1 MCSIENV_SID                   DS    CL3    SYSID
1 MCSIRESERVED01                DS    XL1    reserved
1 MCSIENV_SVERS                 DS    CL4    SYSTEM VERSION
1 MCSIENV_MVERS                 DS    CL4    MSCF VERSION
1 MCSIFAILURE_DETECTION_LIMIT   DS    H      FAILURE DETECTION LIMIT
1 MCSIUSER_TERMINATION_LIMIT    DS    H      USER TERMINATION LIMIT, value
1 *                               32767 means *UNLIMITED
1 MCSIENV_PWR                   DS    FL1    processor password required
1 *                               for confidential
1 *                               communication
1 *   values for field password_required
1 MCSIENV_N                     EQU    0    no password required
1 MCSIENV_Y                     EQU    1    password required
1 MCSIENV_NDEF                   EQU    255   protected or no information
1 *

```

```

1 MCSIRECOVERY_START          DS    FL1  RECOVERY START (general
1 *                            setting)
1 * specifies, whether fail-reconfiguration for partners may be
1 * started automatically by host and partners may start
1 * fail-reconfiguration for host automatically (general setting)
1 MCSIRECOV_NDEF              EQU    0    no information
1 MCSIRECOV_BY_SYS            EQU    1    reconfigurations for partners
1 *                            may be started
1 *                            automatically, as long as
1 *                            allowed by partnerspecific
1 *                            setting of host and settings
1 *                            of partner.
1 MCSIRECOV_BY_OP              EQU    2    reconfigurations for partners
1 *                            are started by operator only
1 MCSIRECOV_CBY_SYS           EQU    3    as by_operator, additionally
1 *                            reconfiguration for HOST
1 *                            must not be started
1 *                            automatically by any partner
1 *                            :* 418
1 MCSIRECOV_SECURE            EQU    8    reconfiguration of host or
1 *                            partner is started
1 *                            automatically only when LIVE
1 *                            MONITOR detects crash :* 502
1 MCSIRECOV_LOCKED            EQU    9    automatic reconfiguration of
1 *                            host or partner is locked by
1 *                            system component :* 520
1 MCSIRECOV_INVALID           EQU   255    invalid :* 408
1 MCSIREQ_RECOVERY_START      DS    FL1  RECOVERY START ( requested
1 *                            value of general setting) :*
1 *                            408
1 * specifies, whether fail-reconfiguration for partners should be
1 * started automatically by host and partners may start
1 * fail-reconfiguration for host automatically (requested value of
1 * general setting) :* 408
1 MCSIRRECOV_NDEF             EQU    0    no information
1 MCSIRRECOV_BY_SYS           EQU    1    reconfigurations for partners
1 *                            may be started
1 *                            automatically, as long as
1 *                            allowed by partnerspecific
1 *                            setting of host and settings
1 *                            of partner.
1 MCSIRRECOV_BY_OP            EQU    2    reconfigurations for partners
1 *                            are started by operator only
1 MCSIRRECOV_CBY_SYS           EQU    3    as by_operator, additionally
1 *                            reconfiguration for HOST
1 *                            must not be started
1 *                            automatically by any partner

```

```

1 *                               :* 418
1 MCSIRECOV_SECURE                EQU 8   reconfiguration of host or
1 *                               partner is started
1 *                               automatically only when LIVE
1 *                               MONITOR detects crash :* 502
1 MCSIRECOV_LOCKED                EQU 9   automatic reconfiguration of
1 *                               host or partner is locked by
1 *                               system component :* 520
1 MCSIRECOV_INVALID               EQU 255 invalid :* 408
1 *
1 MCSIXCS_STATE                   DS   FL1 XCS STATE
1 *   values for XCS STATE
1 MCSIXCS_UNDEFINED               EQU 0   XCS not allowed in this
1 *   session, because of
1 *   MCXSPXCS=N or XCS-NAME=*NONE
1 *   in last MSCF session: XCS
1 *   ressources may be allocated
1 *   locally :* 420
1 MCSIXCS_INACTIVE                EQU 1   XCS inactive
1 MCSIXCS_ACTIVE                  EQU 2   XCS active, xcs ressources
1 *   are allocated globally
1 MCSIXCS_RECONF_JOIN             EQU 3   XCS in join reconfiguration
1 MCSIXCS_RECONF_LEAVE           EQU 4   XCS in leave reconfiguration
1 MCSIXCS_RECONF_FAIL             EQU 5   XCS in fail reconfiguration
1 MCSIXCS_INTERRUPTED            EQU 6   XCS interrupted due to an
1 *   error
1 MCSIXCS_SUSPENDED              EQU 7   last MSCF session is or was
1 *   CCS, but XCS allowed because
1 *   of MCXSPXCS=V or
1 *   XCS-NAME=*SUSPEND :* 420
1 *
1 MCSIENV_HP                       DS   X   HOST PRIORITY
1 MCSIRESERVED02                  DS   XL3 reserved
1 MCSILEAVE_LIMIT                 DS   H   LEAVE LIMIT
1 MCSIABORT_LIMIT                 DS   H   ABORT LIMIT
1 MCSIFADING_INTERVAL            DS   H   FADING INTERVAL :* 502
1 MCSIRESERVED03                  DS   XL14 reserved
1 *
1 MCSIGLOBAL_STATI_NUM_ENTRIES    DS   F   number of host entries
1 MCSIGLOBAL_STATI_AREA_ADDRESS   DS   A   address of host area
1 *
1 MCSIRESERVED05                  DS   XL4 reserved
1 MCSI#                            EQU   *-MCSIFHDR

```

Layout of the output area

```

OUTPUT   MCSINFO MF=D,XPAND=OUTPUT,VERSION=4
1 OUTPUT   MFTST MF=D,PREFIX=M,MACID=CSI,ALIGN=F,           C
1          DMACID=CSI,SUPPORT=(E,D,C,M,L),                 C
1          DNAME=CSI_GLOBAL_STATUS_MDL
2 OUTPUT   DSECT ,
2          *,##### PREFIX=M, MACID=CSI #####
1 *
1 *
1 MCSIHOST_IDENTIFICATION      DS      0XL48
1 *                               host identification,
1 *                               information derived from XCM
1 *                               HCTE or MSCF-HOSTTAB
1 MCSIOUT_HN                    DS      CL8   HOST NAME
1 *
1 MCSI3OUT_HT                   DS      0XL14
1 *                               HOST START DATE/TIME
1 MCSI3YEAR                     DS      CL4   yyyy
1 MCSI3MONTH                    DS      CL2   mm
1 MCSI3DAY                      DS      CL2   dd
1 MCSI3HOUR                     DS      CL2   hh
1 MCSI3MINUTE                   DS      CL2   mm
1 MCSI3SECOND                   DS      CL2   ss
1 *
1 MCSISYSID                     DS      CL3   SYSID
1 MCSIRESERVED06                DS      XL1   reserved
1 MCSISYSTEM_VERSION            DS      CL4   SYSTEM VERSION
1 MCSIMSCF_VERSION              DS      CL4   MSCF VERSION
1 MCSIHOST_SITE                 DS      FL1   HOST SITE
1 *   locality of the host
1 MCSIIS_REMOTE                 EQU     0     remote host
1 MCSIIS_LOCAL                  EQU     1     local host
1 *
1 MCSIRESERVED07                DS      XL13  reserved
1 *
1 *
1 MCSIHOST_STATUS               DS      0XL48
1 *                               host states, information
1 *                               derived from XCM HCTE
1 MCSIHOST_RECONF_NUMBER        DS      F     joining order
1 MCSIHOST_STATE                DS      FL1   HOST STATE
1 *   values for HOST STATE
1 MCSIHOST_NOT_ACTIVE           EQU     0     Host/partner not part of
1 *                               XCS/CCS
1 MCSIHOST_JOIN                 EQU     1     Host/partner joins to XCS
1 MCSIHOST_ACTIVE               EQU     2     Host/partner is active in
1 *                               XCS/CCS

```

1	MCSIHST_HOST_LEAVE	EQU	3	Host/partner leaves XCS/CCS
1	MCSIHST_HOST_FAIL	EQU	4	Partner has terminated
1	*			XCS/CCS abnormally or
1	*			crashed
1	MCSIHST_HOST_ABORT	EQU	5	Host/partner performs
1	*			abnormal XCS/CCS termination
1	MCSIHST_HOST_CRASH	EQU	6	Partner has crashed :*330
1	MCSIHST_HOST_UNDEF	EQU	255	no information
1	*			
1	MCSIPARTNER_TYPE	DS	FL1	PARTNER TYPE
1	*			values for PARTNER TYPE
1	MCSILCS_PARTNER	EQU	0	LCS
1	MCSICCS_PARTNER	EQU	1	CCS
1	MCSIXCS_PARTNER	EQU	2	XCS
1	MCSIPTYPE_UNDEF	EQU	255	no information
1	*			
1	MCSIRECONFIGURATION_STATE	DS	FL1	RECONFIGURATION STATE
1	*			current reconfiguration state of host or partner
1	MCSIRCF_INACTIVE	EQU	0	no reconfiguration active
1	MCSIRCF_JOIN_REQUEST	EQU	3	host or partner waits for
1	*			join
1	MCSIRCF_JOIN_START	EQU	4	host or partner is joining
1	MCSIRCF_JOIN_END	EQU	5	host or partner has joined
1	MCSIRCF_LEAVE_REQUEST	EQU	6	host or partner waits for
1	*			leave
1	MCSIRCF_LEAVE_START	EQU	7	host or partner is leaving
1	MCSIRCF_LEAVE_END	EQU	8	host or partner has left
1	MCSIRCF_FAIL_REQUEST	EQU	10	abnormal termination of
1	*			partner assumed
1	MCSIRCF_FAIL_START	EQU	11	recovery for abnormally
1	*			terminated partner
1	MCSIRCF_FAIL_END	EQU	12	recovery for abnormally
1	*			terminated partner finished
1	MCSIRCF_ABORT_REQ	EQU	20	abort of host or partner
1	*			requested
1	MCSIRCF_ABORT_START	EQU	13	host or partner is in
1	*			abnormal termination
1	MCSIRCF_ABORT_END	EQU	14	host or partner has
1	*			abnormally terminated
1	MCSIRCF_CONN_ACT	EQU	15	mcsf connection to partner
1	*			established
1	MCSIRCF_CONN_LOST	EQU	16	mcsf connection to partner
1	*			lost
1	MCSIRCF_CONTROL_FAIL	EQU	17	control of partner lost
1	MCSIRCF_CONTROL_CRASH	EQU	18	partner possibly crashed
1	MCSIRCF_UNDEF	EQU	255	no information
1	*			
1	MCSIHST_HOST_CONTROL	DS	FL1	control state


```

1 * values for HOST CONTROL
1 MCSIALL_INACTIVE EQU 0 partner is not monitored
1 MCSICOMMUNICATION_ACTIVE EQU 1 partner is monitored by
1 * communication only
1 MCSIDISK_ACTIVE EQU 2 partner is monitored by by
1 * disk only
1 MCSIALL_ACTIVE EQU 3 partner is monitored by
1 * communication and disk
1 MCSILOST EQU 4 both monitoring paths of
1 * partner are lost
1 MCSICRITICAL EQU 5 disk protokoll for partner
1 * not possible
1 MCSICRASHED EQU 6 partner has crashed
1 MCSIMSCF_TERMINATED EQU 7 mscf terminated on partner :*
1 * 410
1 MCSIHCTRL_NDEF EQU 255 no information
1 *
1 MCSIFLAGS DS AL1 flags :* 404
1 MCSICRASHED_QUEST EQU X'80'
1 * Is set together with
1 * host_control lost (MCS1100
1 * open) :* 408
1 MCSICONNECT_REQ EQU X'40'
1 * Is set together with
1 * host_control disk_active or
1 * lost (MCS1101 open) :* 408
1 MCSIRESERVED08 EQU X'3F'
1 * filler for alignment
1 MCSIHOST_PRIORITY DS X HOST PRIORITY
1 MCSICONN_RECOVERY_START DS FL1 RECOVERY START
1 * (partnerspecific setting) :*
1 * 402
1 * specifies, whether fail-reconfiguration for this partner may be
1 * started automatically by host and this partner may start
1 * fail-reconfiguration for host automatically (partnerspecific
1 * setting) :* 402
1 MCSICRECOV_NDEF EQU 0 no information
1 MCSICRECOV_BY_SYS EQU 1 host may start
1 * reconfiguration for partner
1 * automatically as longas
1 * allowed by partner
1 MCSICRECOV_BY_OP EQU 2 host starts reconfiguration
1 * for partner initiated by
1 * operator only
1 MCSICRECOV_CBY_SYS EQU 3 as by_operator, additionally
1 * reconfiguration for HOST
1 * must not be started
1 * automatically by this

```

```

1 * partner. :* 418
1 MCSICRECOV_STD EQU 5 This partner is controlled
1 * only after it has shared a
1 * pubset with host. When it is
1 * controlled, general
1 * recovery-start setting comes
1 * into effect. :* 402
1 *
1 MCSIPARTN_RECOVERY_START DS FL1 RECOVERY START (combined
1 * general and partnerspecific
1 * setting of partner) :* 402
1 * specifies the combined general and partnerspecific recovery-start
1 * settings of this partner as they are effective for host. :* 402
1 MCSIPRECOV_NDEF EQU 0 no information
1 MCSIPRECOV_BY_SYS EQU 1 partner may start
1 * reconfiguration for host
1 * automatically, as long as
1 * allowed by host
1 MCSIPRECOV_BY_OP EQU 2 partner starts
1 * reconfiguration for host
1 * initiated by operator only
1 MCSIPRECOV_CBY_SYS EQU 3 as by_operator, additionally
1 * host must not start
1 * reconfiguration for this
1 * partner automatically). :*
1 * 418
1 MCSIPRECOV_NO_CTRL EQU 6 partner does not control
1 * host. :* 418
1 MCSIPRECOV_UNKNOWN EQU 7 setting of partner with
1 * mscf_version < V14 not
1 * known. :* 426
1 *
1 MCSICONTROL_GROUP DS CL1 multiple fail
1 * reconfigurations for
1 * partners in same control
1 * group are started
1 * simultaneously :* 402
1 MCSIRESERVED09 DS XL3 filler for structure
1 * alignment
1 MCSIHOST_XCS_NAME DS CL8 XCS NAME :* 402
1 *
1 MCSI4XCS_START_TIME DS 0XL14
1 * XCS CREATION DATE/TIME :* 402
1 MCSI4YEAR DS CL4 yyyy
1 MCSI4MONTH DS CL2 mm
1 MCSI4DAY DS CL2 dd
1 MCSI4HOUR DS CL2 hh
1 MCSI4MINUTE DS CL2 mm

```

```

1 MCSI4SECOND          DS    CL2   ss
1 *
1 MCSIRESERVED10      DS    XL10  reserved
1 *
1 *
1 MCSICONNECTION_STATUS DS    0XL32
1 *                      connection states,
1 *                      information derived from
1 *                      MSCF-HOSTTAB
1 MCSIPROCESSOR_NAME   DS    CL8   PROCESSOR NAME
1 MCSICONNECTION_TYPE DS    FL1   CONNECTION TYPE
1 * values for CONNECTION TYPE
1 MCSINOT_CONFIDENTIAL EQU    0    loosely coupled
1 MCSICONFIDENTIAL     EQU    1    closely coupled
1 MCSICTYPE_NDEF       EQU    255   undefined
1 *
1 MCSIOUT_PWR          DS    FL1   states whether processor
1 *                      password is required for
1 *                      confidential communication
1 * values for field password_required
1 MCSIOUT_N            EQU    0    no password required
1 MCSIOUT_Y            EQU    1    password required
1 MCSIOUT_NDEF        EQU    255   protected or no information
1 *
1 MCSICONNECTION_STATE DS    FL1   CONNECTION STATE
1 * values for CONNECTION STATE
1 MCSICONNECTION_LOCAL EQU    0    host is local
1 MCSICONNECTION_CONNECTED EQU    1  host is connected
1 MCSICONNECTION_NOT_CONNECTED EQU    2  host is not connected
1 MCSICSTAT_NDEF     EQU    255   no information
1 *
1 *
1 MCSIOVERLOAD        DS    0XL2  OVERLOADED (REQUEST,REPLY)
1 MCSIREQUEST         DS    CL1   information for request
1 *                      messages
1 MCSIREPLY           DS    CL1   information for reply
1 *                      messages
1 *
1 MCSIREQ_N_CTRL_CONN DS    FL1   requested number of control
1 *                      connections : * 420
1 * values of N_Cntrl_Conn Parameters
1 MCSINO_CNTRL        EQU    0    no control connection
1 MCSIONE_CNTRL       EQU    1    one control connections
1 MCSITWO_CNTRL       EQU    2    two control connections
1 MCSIUNDEF_CNTRL     EQU    127   control connections not
1 *                      possible to partners with
1 *                      MSCF < V14
1 *

```

1	MCSIUSED_N_CTRL_CONN	DS	FL1	number of control connections
1	*			currently in use :* 420
1	MCSIREQ_CONN_RECOVERY_START	DS	FL1	RECOVERY START (requested
1	*			value of partnerspecific
1	*			setting) :* 404
1	MCSIREQ_CONTROL_GROUP	DS	CL1	multiple fail
1	*			reconfigurations for
1	*			partners in same control
1	*			group are started
1	*			simultaneously :* 404
1	MCSIRESERVED11	DS	XL15	reserved
1	*			
1	MCSIRESERVED12	DS	XL16	reserved
1	MCSIOOUTPUT#	EQU		*-MCSIOUT_HN

11 Appendix

11.1 MSCF interface in /ENTER-JOB and /ENTER-PROCEDURE

The HOST operand of the ENTER-JOB and ENTER-PROCEDURE commands includes the value HOST=*ANY. Its meaning is described below. For a description of the complete commands, see the "Commands" manual [10].

Operands

HOST=*ANY

Defines that the job can run on any processor in the XCS network. The job is transferred for processing by the distributor component of the JMS to the processor that has the lowest batch load.

Notes

The operand HOST=*ANY causes the JMS itself to define a destination processor in the XCS network of the job submitter. The selection criterion on which the selection of the job processing processor is based is the batch load of the processors in the XCS network, i.e. the job processing is handed to the processor that has the lowest batch. In this context, the following should be noted:

- The processor's load is in each case determined on the basis of the job class assignment during the job acceptance phase, i.e. the JMS determines the destination processor of a job by comparing the load at the time the job is started. The job classes are categorized by the amount of system resources they will take up (CPU use, run priority etc.) by means of the job classes of the JMS (see the "Introductory Guide to Systems Support" [5]). The number of active, i.e. simultaneously running, jobs of a job class is restricted by the CLASS-LIMIT attribute. If more jobs of a job class than are permitted by CLASS-LIMIT are sent to the system for processing, the jobs are accepted by the JMS but they are *not active* and lay no claim to resources. Scheduling, repeat and calendar jobs are also as a rule not active at first because they have not yet reached their start time.

In order to determine the destination processor, the job processor (i.e. the processor of the MSCF network which first received the job for acceptance, e.g. from the ENTER-JOB command) sends requests to all the processors in the XCS network with information about the upcoming batch job. A "prevalidation" is carried out on all the processors reached in the XCS network (including the job processor itself). This involves determining the basic acceptability of the job on each processor (the job would not be accepted, for instance, if its job class is not defined on the potential destination processor).

As a result of the prevalidation, the job processor receives responses containing the following information:

- acceptable/not acceptable
- indicators of system overload, category overload
(when this overload situation occurs, JMS does not start any more jobs even if job class assignments would allow it)
- job class counter and job class status
- number of non-active jobs of the job class

From this returned information, the job processor determines the processor with the greatest remaining acceptance capacity from the matrix of possible destination processors whose prevalidation confirmed the basic acceptability. The JMS selects the destination processor according to the following priorities:

1. Stream/job class status (if, for instance, the stream in question is not active on a processor or the job class in question has been put on hold on the destination processor by the HOLD-JOB-CLASS command or its CLASS-LIMIT is temporarily 0, the job cannot be started on that processor immediately after acceptance)
2. Processor load (system/category overload)
3. Job class assignment (CLASS-LIMIT minus the jobs currently active in the job class)
4. Job class optimum (value for CLASS-OPTIMUM minus the jobs currently active in the job class)

In other words, first the loads involving active jobs (i.e. jobs in progress) are compared as at the time of the job acceptance on the individual processors in the network. Under normal operation (i.e. no overload, streams and job classes active on all the processors, CLASS-OPTIMUM=0), the processor with the highest value for CLASS-LIMIT minus the jobs currently in the job class has the highest acceptance capacity. If the job classes are assigned on all the available processors in the network, the values for CLASS-LIMIT minus the total number of accepted jobs of the relevant job class on the processors becomes negative but the comparison, even in this case, still pinpoints the potentially least loaded processor in the network. If a processor indicates that the job cannot be started due to the current setting (e.g. stream not active, job class in HOLD state) or due to system/category overload, it ceases to be a possible destination processor, even

if it has a lower batch load according to the job class counter. If all of the above criteria prevent a destination processor from being decided on, the job remains on the job processor.

The job class optimum is another parameter with which in some cases the batch job distribution in the XCS network can be controlled/improved. The job class optimum is a number (see CLASS-OPTIMUM operand of the MODIFY-JOB-CLASS command) lower than the maximum number of possible active jobs in a job class (see CLASS-OPTIMUM operand of the DEFINE- and MODIFY-JOB-CLASS commands). It is used to define the optimum assignment of a job class by systems support. Usually this is an empirical value which can only be defined based on the individual situation (e.g. due to the best observed throughput of batch jobs in an application). If CLASS-OPTIMUM is not 0, the distributor component of the JMS first considers all the processors on which the CLASS-OPTIMUM has not yet been reached in the job class concerned. If several processor are available, the one with the highest value for CLASS-OPTIMUM minus the jobs currently in the job class is preferred.

- Once the destination processor has been decided on, the job is sent to that processor (in the same way as under the existing function in which a destination processor is explicitly determined by the user with the HOST operand of the ENTER-JOB command). The job then runs through the acceptance phase on that processor and the result is returned to the job submitter (message JPM0500 and other messages, usually JMS0066). The accepted job on the destination processor is transferred to the job scheduler there which releases it for start. Once a job has been accepted on a processor, it stays there.

The result of the prevalidation does not always have to agree with the final acceptance. This applies when JMS settings on the destination processor are modified by systems support between prevalidation and acceptance on the destination processor.

CLASS-LIMIT may in the interim, for example, have been set to 0. Such interventions by systems support are not prevented by the JMS (no global lock on JMS data in the XCS network). In situations like this, the JMS attempts to find another processor in the network (after receiving the information from the destination processor as a result of the prevalidation that the job is not accepted).

- If no processor can be found in the entire network that can accept the job, the reasons are sent in messages from each processor to the job processor (see message JDS0322).
- Default response: if no specification is made for the following job attributes in the ENTER-JOB command, the default response is as follows:
 - PROCESSING-ADMISSION (default value = *SAME):
User ID of the batch job to be generated is the user ID of the job submitter task on the job processor. The attributes of the user ID of the same name of the destination processor are decisive for the authorization checks.

- JOB-CLASS (default value = *STD):
Job class of the batch jobs to be generated is the default batch job class of the user ID on the destination processor.
- Job attributes whose default values are taken from the job class; the default value of the job class definition of the destination processor applies.
- Specifying the value *ANY in the HOST operand is pointless for scheduling, calendar and repeat jobs, but is not prevented by the JMS. At the time of acceptance when the distribution is made, the JMS cannot forecast the load for scheduling jobs at the time the job is initiated. Although repeat jobs are generated with a new TSN for each type, they still remain on the processor that has been selected. Calendar jobs have only one TSN and must therefore remain on the processor that has been selected. If these jobs are transferred to the system under specification of HOST=*ANY, they are distributed as described above. The JMS makes no plans about the expected loads; here too it determines the acceptance capacity of a processor based on the differences between optimum and actual jobs as described above.

The same applies by analogy to all other jobs that cannot start immediately after acceptance (e.g. CLASS-LIMIT=0, job class or job stream in HOLD status or job stream not started). If, for instance, jobs are transferred during the day to the stream that is to be active during the night, the JMS decides on the distribution on the basis of the available job figures at the time of acceptance. It is up to the user to decide whether the response he requires is feasible.

- Restriction:
If a task with the OPERATOR privilege is the job submitter of a batch job, the SET-LOGON-PARAMETERS command in the Enter command file is not evaluated if HOST=*ANY is specified.
- Recommendations for system settings (JMS parameters, user attributes):
We recommend assigning the same JMS parameters (job class definition, default job classes), user attributes and contingents on all participants of the XCS network. The JMS carries out the checks described above (prevalidation) so that the executability of the batch job on the selected processor is guaranteed, but the uniform assignment is not compulsory. Inconsistencies which only become apparent at the runtime of the batch jobs will therefore not be revealed here (e.g. different syntax files, different privilege assignment etc.). If the assignment of uniform system settings is deviated from, this should be done consciously for a specific effect. If, for example, CLASS-LIMITs are differently assigned on the processors for the same job class, this has the effect that processors with the larger value are given priority, even if the number of jobs that such a processor has already accepted in the corresponding job class is greater than that on other processors.

11.2 Summary: generation and operation of an MSCF network

11.2.1 General preconditions for setting up an MSCF network

- The BCAM network must be generated (see the “BCAM” manual [13]). BCAM and MSCF are loaded on each processor in the network independently of each other. BCAM must be loaded before MSCF.
- The message length (MAXTSDU) in the BCAM definition must be at least 8 Kbytes.
- BCMAP commands for the applications \$MRSAPP and \$MRSAPPS must be started (where necessary).
- A DSSM subsystem catalog entry must be created with the MSCF declarations (start time for MSCF, where applicable the name of the configuration file).

Starting the MSCF network or entering an existing network and manually setting up connections to partner processors (without configuration file)

- Start HIPLEX MSCF with the START-SUBSYSTEM MSCF command
- The START-MSCF-CONNECTION command must be started for each processor to which a connection is to be set up.

Starting the MSCF network or entering an existing network and setting up connections to partner processors (where configuration file exists)

The following two options are possible:

- Start HIPLEX MSCF with the START-SUBSYSTEM MSCF command
When the subsystem is started, the connections to the processors for which the configuration file contains the START-MSCF-CONNECTION command are automatically set up.
- Start HIPLEX MSCF implicitly with the BS2000 startup
 - Start BCAM in the BS2000 startup
 - Enter "BEFORE-SYSTEM-READY" as the MSCF start time in the DSSM subsystem catalog
 - Enter the configuration file in the DSSM subsystem catalog as the subsystem information file

- The RDF should be set up in such a way that the necessary things are executed for the home processor before the BCAM network is fully activated. If the BCAM-SOF file is used, the commands for the BCAM connections used by HIPLEX MSCF and the corresponding BCMAP commands (if needed) should therefore be placed as near to the beginning as possible.

Manually clearing down connections to partner processors and terminating the MSCF network

The STOP-MSCF-CONNECTION command can be issued for each partner processor to which a connection is set up.

If HIPLEX MSCF is terminated with the STOP-SUBSYSTEM MSCF command, the STOP-MSCF-CONNECTION command is implicitly executed with the command for all the partner processors to which a connection is set up.

11.2.2 Additional requirements for the CCS network

- Uniquely assigned host name for DCSTART or DCOPT
- Unique SYSIDs for local pubsets
- Unique catalog IDs and VSNs
- Mutually adjusted user IDs (access rights, quotas). User jobs are processed on the remote side by MSCF server tasks. The server tasks "mask" themselves for this purpose with the IDs and privileges of the local job submitter)
- Mutually adjusted system parameters (e.g. ENCRYPTION, DEFLUID)
- Same values for FAIL-DETECTION-LIMIT.

11.2.3 Additional requirements for the shared pubset network

The master processor must be linked to each slave processor via a CCS connection. To ensure that a master change is possible, CCS connections should also be set up between the backup master and all slave processors. In addition, the following requirements must be met:

- The pubset must be declared a potential shared pubset in its SVL.
- Where applicable, the master processor, backup master and alternative backup must be predefined in the SVL.
- The pubset must be entered as a shared pubset in the MRSCAT.

11.2.4 Additional requirements for the XCS network

- Fully meshed XCS configuration
- BCMAP commands for the applications \$MCSNSM and \$MCSNLX
- Value for system parameter MCXSPXCS must be set to "Y" or "V"
- XCS pubset(s) must be set up with the command ADD-MASTER-CATALOG-ENTRY SHARED-PUBSET=*YES,XCS=*YES.
- XCS name must be defined in the configuration file with the SET-MSCF-ENVIRONMENT command.
- The START-MSCF-CONNECTION command must be specified in the configuration file for all the XCS partner processors.

11.2.5 Additional requirements for parallel HIPLEX

- Central global storage connected to all XCS processors
- The GSUSAGE parameter (SYSSSI.GSMAN.<ver> file) must be set to the value GLOBAL on all the XCS processors.

Glossary

backup master

A slave processor which is to become the new master processor of a shared pubset following a master change.

batch mode

Job started with the ENTER command. In contrast to interactive mode, execution is predefined and specified in an ENTER file.

BCAM

BCAM (Basic Communication Access Method). This is a functional unit of the data communication system DCM. BCAM implements transport services and the associated administrative functions.

catalog ID (= CATID)

See "CATID".

CATID (= catalog ID)

Identifies a pubset (see "volume serial number", "pubset"). It is specified in the complete file name / path name in the form :catid:.

console

Terminal of systems support, used for system operation.

control volume set

The control volume set is the container for global pubset metadata files of SM pubsets.

exported pubset

See "imported pubset".

heterogeneous network

MSCF network comprising hosts with different BS2000 versions.

HIPERFILE concept

BS2000 concept which aims to implement high-performance file processing and thereby increase the performance of the overall system.

home pubset

The home pubset is required to initialize the BS2000 operating system and during the entire session. Each processor running under BS2000 has one home pubset.

User access to BS2000 is always via the home pubset. This is where the logon is validated and the access permissions and availability of resources are checked.

imported pubset

MPVS (Multiple Public Volume Sets), a functional extension of the Data Management System, allows several sets of public disks (pubsets) to be installed within a session. Further pubsets can be imported as needed onto the system of the home pubset, which is required throughout the session; these can then be accessed directly by the importing system. The properties of imported pubsets can be queried using the SHOW-MASTER-CATALOG-ENTRY and SHOW-PUBSET-PARAMETERS commands.

Each user can create, edit and delete files and job variables on an imported pubset, provided this authorization is granted by the system management. Of course, an imported pubset can also be exported again, i.e. moved from the system. The home pubset cannot be exported.

long connection

Communication variant in HIPLEX MSCF. A long connection is set up and cleared down by MSCF applications and is available exclusively to these applications for communication purposes. The link to the server tasks of the MSCF applications remains open for the entire lifetime of the connection, i.e. for a long period.

master change

Process in which the master attribute of a processor in a shared pubset network is transferred to a different sharer. This process is initiated by unloading MSCF on the master processor (also with shutdown), by the failure of the master processor, and by a master export with master change. An unsuccessful master change can be restarted by issuing the IMPORT-PUBSET command.

master processor

The processor on which all DMS administration functions of a shared pubset are executed.

MRSCAT

Data structure which assigns a catalog ID to the BCAM name of the BS2000 processor managing the catalog.

The pubsets are addressed using the MRSCAT catalog directory. Although any pubset can possess such a catalog directory, only the MRSCAT of the home pubset is used for pubset addressing during the session.

MSCF

Subsystem of BS2000/OSD which allows privileged components (MSCF applications) of the operating system to communicate with other BS2000 entities. MSCF communicates by means of BCAM.

MSCF configuration file

The configuration file is used to define the configuration parameters and the processors with which an MSCF connection is to be established. The configuration file is executed automatically when HIPLEX MSCF is started. A configuration file in command format is mandatory for the XCS network.

MSCF connection

Communication paths which exist between two processors and are managed by the MSCF subsystem.

MSCF server task

Special system task which executes an MSCF job of a partner processor on the local processor. The number of MSCF server tasks is increased and decreased dynamically by MSCF.

MSCF trace file

File containing the traces of the MSCF subsystem. The MSCF traces are an important source of information for diagnosing error situations.

non-relevant partner

See "relevant partner".

partner

Remote BS2000 system which can also be a guest system under VM2000.

PFA concept

Increasing the performance of the overall system through high-performance file processing.

(BS2000) processor

BS2000/OSD server with the BS2000/OSD (native) operating system or a VM2000 guest system on a BS2000/OSD server with VM2000.
In other literature the term "BS2000 system" (system for short) is also used.

pubset (= PVS, public volume set)

A pubset (PVS) is a public storage area (i.e. memory available to all users) on disk storage units combined to form one administrative unit. A pubset must contain at least one disk. The disk numbered '0' is called the PUBRES; in particular, this contains the references to any other disks in the pubset.

Pubsets are divided into SF and SM pubsets.

A single-feature pubset (SF pubset) is made up of volumes with identical physical characteristics.

A system-managed pubset (SM pubset), which comprises one or more volume sets (max. 255), can be viewed as an independent repository for logically related data.

In addition to the data, the SM pubset contains the associated metadata and thus represents a switch unit.

See "control volume set".

See "imported pubset".

See "pubset configuration file".

pubset configuration file

The volume sets belonging to an SM pubset and their characteristics are detailed in the pubset configuration file :<catid>:

\$TSOS.SYS.PUBSET.CONFIG. This file is created implicitly when the pubset is generated and is always contained on the control volume set.

See "pubset".

PVS

See "pubset".

PVS ID

Pubset identifier; specified in the form :catid: in the full file name / path name.

registered function

Functional unit whose entities are established on all processors in the XCS network and whose inter-processor cooperation is supported by XCM.

relevant partner

From the point of view of partner monitoring, the term "relevant partners" refers to the partner processors on which BS2000 version OSD-BC \geq V3.0 is running and to which a CCS connection is established or has been established at any time in the current HIPLEX MSCF session. All other partner processors are called "non-relevant partners".

RFA

Remote File Access; software product that enables access to files located on non-local (i.e. remote) BS2000 processors. RFA can be used with or without MSCF.

SF pubset

See "pubset".

shareable private disk (= SPD)

A disk mounted in a disk drive that can be accessed by more than one processor via direct hardware connections.

shared pubset

Public volume set which can be accessed by two or more processors concurrently.

shared pubset network

Network of processors which access a shared pubset.

shared resource

Shared resources are utilized by all the users of the respective network (e.g. disks, global memory). Shared resources must therefore be accessible from all processors.

sharer

Processors which have simultaneously imported a particular pubset (shared pubset).

sharer in the SVL

SVL entry of a sharer on each volume of a shared pubset. The entry is normally made at import and deleted at export or in the event of a processor failure.

short connection

Communication variant in HIPLEX MSCF. A short connection is set up with the START-MSCF-CONNECTION command and cleared down with STOP-MSCF-CONNECTION, and is available equally to all MSCF applications. A link between the job-submitting task and the job-receiving task only exists for the short duration of a job.

slave processor

Any sharer of a shared pubset which is not the master processor.

SM pubset

See "pubset".

Software Monitor openSM2

The openSM2 monitoring system provides the user with statistical data on the performance of the DP system and the utilization of resources.

standby mode

In the event of a processor failure (and when maintenance tasks are being carried out), the continued availability of critical applications is ensured if the environment required for such an application is maintained on a standby system and the application is automatically started on this system should the processor fail.

This is guaranteed by the product HIPLEX AF (AF = Availability Facility) in the shared pubset network.

SVL (= standard volume label)

Standard volume label with volume serial number.

SYSID

System identifier comprising one to three characters. Possible values are: A...Z; 0...9 and 65...192; (default: 250).

user catalog

A user catalog contains the pubset-specific permissions for each user entered in this file. The entries in the user catalog control which user ID can access a pubset. If a user does not have an entry for a particular pubset in the user catalog, this user cannot access this pubset or any shareable files (unless general permission is defined via the system parameter FSHARING). Users can check which entries exist for them by issuing the SHOW-USER-ATTRIBUTES command.

The user catalog is sometimes known as the JOIN file.

volume serial number (= VSN)

A 6-position number assigned to the volume during initialization (VOLIN or INIT). This number is contained in the standard volume label and is used to identify the volume.

The VSN of public volumes begins with the character string "PUB" (PUB notation) or contains a period between the 1st and 6th character (dot notation). In PUB notation, the 4th character is the SF pubset ID or (with SM pubsets) the volume set ID.

volume set

Component of an SM pubset comprising one or more volumes (max. 255). Each volume set contains complete files. The control volume set is a special volume set as it contains global pubset metadata files.

VSN

See "volume serial number".

watchdog

Functional unit of the MSCF subsystem. Manages the sharer configuration of all shared pubsets, monitors the sharer using a disk protocol, and implements the necessary DMS recovery measures in the event of a failure.

watchdog file

The file \$TSOS.SYS.PVS.SHARER.CONTROL on a shared pubset. It is required for the disk protocol (reading/writing vital-sign messages). The file is automatically created in shared mode the first time the pubset is imported and must not be modified. Otherwise, the pubset cannot be used as a shared pubset until it has been exclusively imported and the watchdog file has been deleted.

XCS Configuration Manager (XCM)

Component of HIPLEX MSCF. Ensures global consistency in the XCS network by monitoring the XCS sharers and, in the event of configuration changes, by reconfiguring the distributed system functions which exist on the sharers.

XCS network

An XCS network combines up to 16 processors into one function network. CCS connections are established between all processors and the processors have imported at least one shared pubset (XCS pubset).

XCS pubset

A special shared pubset which is mandatory for the XCS network and is imported exclusively by the processors in an XCS network. The XCS pubset must be imported by all processors in this XCS network. An XCS network requires at least one XCS pubset.

Abbreviations

AFR	Access From Remote
AVAS	job management and scheduling system (German acronym)
BCAM	Basic Communication Access Method
CCS	Closely Coupled System
CJC	Conditional Job Control
CMS	Catalog Management System
CPM	shared pubset manager
DAB	Disk Access Buffer
DIV	Data In Virtual
DLM	Distributed Lock Manager
DMS	Data Management System
DSSM	Dynamic SubSystem Management
FASTPAM	FAST Primary Access Method
GS	Global Storage
GSM / GSMAN	Global Store Manager
GSV / GSVOL	Global Storage Volume emulation
GUARDS	Generally Usable Access contRol aDministration System
HIPERFILE	High PERformant FILE
HIPLEX	Highly Integrated system compLEX
IMON	Installation MONitor
ISAM	Index Sequential Access Method
JMS	Job Management System
JV	Job Variable
LCS	Loosely Coupled System
MONJV	MONitoring Job Variable

Abbreviations

MSCF	Multiple System Control Facility
NSG	Node Synchronization - Global storage part
NSM	Node Synchronization Manager
PCA	Peripheral Cache Handler
PFA	Performant File Access
PLAM	Program Library Access Method
PPD	Protected Private Disk
PVS	Public Volume Set
RAID	Redundant Array of Independent Disks
RCA	Remote Catalog Access
RDF	Resource Definition File (BCAM)
RFA	Remote File Access
RMS	REP Mounting System
SCA	Speed Catalog Access
SOLIS	software delivery and information system (German acronym)
SPD	Shareable Private Disk
SRPM	System Resources and Privileges Management
SVL	Standard Volume Label
TSN	Task Serial Number
UPAM	User Primary Access Method
UTC	Universal Time Coordinate
UTM	Universal Transaction Monitor
VSN	Volume Serial Number
VTOC	Volume Table Of Contents
XCM	XCS Configuration Manager
XCS	Cross-Coupled System
XPM	XCS Pubset Manager
XTS	XCS Time Synchronization

Related publications

The manuals listed below are available from these sources:

- On the Internet from our manual server (URL: <http://manuals.ts.fujitsu.com>)
- On the DVD “BS2000/OSD SoftBooks”.
This CD contains practically all the manuals and readme files for the BS2000 system software of the latest BS2000/OSD version.

Printed copies of the manuals can be purchased separately over the Internet from:
<http://manualshop.ts.fujitsu.com>.

- [1] **AVAS / AVAS-SV** (BS2000/OSD)
AVAS for the Administrator
System Administrator Guide
- [2] **DAB** (BS2000/OSD)
Disk Access Buffer
User Guide
- [3] **BS2000/OSD-BC**
DMS Macros
User Guide
- [4] **BS2000/OSD-BC**
Introductory Guide to DMS
User Guide
- [5] **BS2000/OSD-BC**
Introductory Guide to Systems Support
User Guide
- [6] **HIPLEX AF** (BS2000/OSD)
Automatic Switching of Applications between BS2000 Systems
User Guide
- [7] **IMON** (BS2000/OSD)
Installation Monitor
User Guide

- [8] **HSMS / HSMS-SV** (BS2000/OSD)
Hierarchical Storage Management System
User Guide
- [9] **JV** (BS2000/OSD)
Job Variables
User Guide
- [10] **BS2000/OSD-BC**
Commands
User Guide
- [11] **LMS** (BS2000/OSD)
SDF Format
User Guide
- [12] **BS2000/OSD-BC**
Executive Macros
User Guide
- [13] **openNet Server** (BS2000/OSD)
BCAM
User Guide
- [14] **openSM2** (BS2000/OSD)
Software Monitor
User Guide
- [15] **RFA** (BS2000/OSD)
Remote File Access
User Guide
- [16] **SECOS** (BS2000/OSD)
Security Control System - Access Control
User Guide
- [17] **BS2000/OSD-BC**
System-Managed Storage
User Guide
- [18] **SHC-OSD** (BS2000/OSD)
Symmetrix Host Component
User Guide
- [19] **BS2000/OSD-BC**
Subsystem Management (DSSM / SSCM)
User Guide

Index

\$MCSNLX (application) [22](#)
\$MCSNSM (application) [22](#)
\$MRSAPP (application) [21](#)
\$MRSAPPS (application) [21](#)

A

abbreviations [349](#)
abnormal termination
 HIPLEX MSCF [184](#)
 MSCF system task [185](#)
 network sharer [149](#)
ABORT-LIMIT
 MSCF configuration parameter [121](#)
access permission in the MSCF network [82](#)
 batch [82](#)
 dialog [83](#)
access request [82](#)
activate subsystem [299](#)
activating and deactivating partner
 monitoring [124](#)
ADD-MASTER-CATALOG-ENTRY
 command [131](#), [140](#), [152](#)
addressing destination processor [78](#)
addressing files [60](#)
addressing, indirect [78](#)
 example [80](#)
adjusted network generation [113](#)
administration commands
 overview [143](#)
ALTERNATE-BACKUP [133](#)
analysis of system status [164](#)
ARCHIVE [42](#)
authentication [82](#)
automatic error recovery [176](#)
availability network [34](#)

AVAS (job management and scheduling system) [42](#), [96](#)

B

backup master [172](#), [173](#), [341](#)
 select [134](#)
BATCH-WAIT-TIME (operand) [170](#)
BCAM [341](#)
 BCEND command [114](#), [149](#)
 BCOUT command [149](#)
 BCTIMES command [114](#)
 commands [150](#)
 MAX-MSCF-DELAY [149](#)
 network [113](#)
 SHOW-BCAM-ATTRIBUTES command [114](#)
 trace [165](#)
before images [68](#)

C

cache media [194](#)
catalog entry [163](#)
CATID [62](#), [341](#)
CCS network [30](#)
 connection monitoring [30](#)
 failure of a partner [172](#)
 generation [130](#)
 loss of connection [168](#)
 partner type [30](#)
CCS partner, failure [172](#)
CJC [86](#)
Cluster Recovery Lock (CRL) [214](#)

- command
 - MODIFY-MSCF-CONNECTION 203
 - MODIFY-MSCF-ENVIRONMENT 206
 - MODIFY-NSM-ENVIRONMENT 212
 - RELEASE-CLUSTER-RECOVERY-LOCK 214
 - SET-MSCF-ENVIRONMENT 216
 - SET-NSM-ENVIRONMENT 222
 - SET-XCS-PUBSET 225
 - SHOW-MSCF-CONFIGURATION 227
 - SHOW-NSM-CONFIGURATION 277
 - SHOW-SHARED-PUBSET 285
 - SHOW-XCS-OCCUPATION 290
 - START-MSCF-CONNECTION 293
 - STOP-MSCF-CONNECTION 303
- command overview 201
 - information functions 164
- commands 201
- communication bottleneck 167
- communication path 213, 222
- computer network
 - prerequisites for SM2 implementation 190
- concurrent copy (CCOPY) 68
- conditional job control (CJC) 86
- configuration file 301
 - example 118
 - format 116
 - NSM 141
 - XCS network 117
- configuration number 38
- connection
 - display 150
 - establish to partner processor 147
- connection clear-down 148
- connection monitoring 30
- connection setup, reject 180
- connection statuses 150
- connection type 27
 - CCS 30
 - LCS 29
- CONSLOG file 165
- control group 32
- control parameters, global 119
- control volume set 341
- coordination of BCAM/MSCF starts 145
- CPM (XCS function) 37
- crash detection 172
- crash recovery, operation 178
- CREATION-TIME (operand) 111
- critical states 163
- CRL (Cluster Recovery Lock) 214
- cross-matching, global store 54
- cycle time
 - normal operation 213
 - standby mode 213
- D**
- DAB 52
 - emulated disks 54
- DAB (SM2 monitoring program) 192
- DAB caching, processor crash 53
- data management system and shared pubsets 64
- deactivate subsystem 305
- deadlock, detect 43
- define
 - MSCF environment 216
- define shared pubset master 133
- destination processor
 - addressing 78
- destination processor, addressing 78
- determine sharers on shared pubset 285
- DIALOG-WAIT-TIME (operand) 170
- disk access buffer 52
- disk monitoring 31
 - block 183
 - reliability 135
- display VTOC catalog entries 107
- distributed lock manager (DLM) 41
- DLM (Distributed Lock Manager) 43
 - interfaces 44
- DLM (SM2 monitoring program) 192
- DLMUSER (subsystem) 44
- DMS functions 60
- documentation for error diagnosis 165
- DPSHSH 310
- DSSM subsystem catalog 111

E

Email notification [163](#)
ENTER-JOB, command extension [333](#)
ENTER-PROCEDURE, command extension [333](#)
entry to XCS network [158](#)
erroneous partner crash detection [179](#)
error
 when writing vital-sign messages [181](#)
error diagnosis, documentation [165](#)
error handling, GSVOL [59](#)
error recovery [163](#)
 automatic [176](#)
 GSMAN [56](#)
 with parallel HIPLEX [57](#)
establish connection [293](#)
example, configuration file [118](#)
exit from XCS network [160](#)
EXPORT-PUBSET command [137](#)

F

FADING-INTERVAL [72](#)
 MSCF configuration parameter [121](#)
fail reconfiguration [39](#)
 prohibiting automatic starts [125](#)
FAIL-DETECTION-LIMIT
 MSCF configuration parameter [123](#)
failure
 BCAM connection [168](#)
 LCS/CCS partner [172](#)
 of sharer [173](#)
 processor in shared pubset network [172](#)
 XCS sharer [175](#)
file addressing [60](#)
file lock [64](#)
file lock manager [64](#)
format of the configuration file [116](#)
FSHARING
 system parameter [136](#)
function network [24](#)
 prerequisites [19](#)

G

generation
 CCS network [130, 131](#)
 LCS network [129](#)
 shared pubset network [131](#)
 XCS network [140](#)
GLOBAL (SM2 monitoring program) [192](#)
global control parameters [119](#)
global crash [184](#)
global storage [194](#)
 caching [197](#)
 configuration/reconfiguration [50](#)
 cross-matching [54](#)
 emulated volume (GSVOL) [41](#)
 level [182](#)
 shared partitions [48](#)
 single point of administration [51](#)
 use by DAB [52](#)
 use by GSVOL [54](#)
GS (SM2 monitoring program) [192](#)
GS volume emulation [54](#)
GSM (XCS function) [37](#)
GSMAN [48](#)
 error recovery [56](#)
GSUSAGE [48](#)
GSV (XCS function) [37](#)
GSVOL [41, 48, 54](#)
GSVOL (SM2 monitoring program) [192](#)
GTIME (interface) [41](#)
GTIMR interface [47](#)
GUARDS [64](#)

H

heterogeneous network [341](#)
HIPLEX ADM [42](#)
HIPLEX AF [42, 71, 73](#)
 Live Monitor [71](#)
HIPLEX concept [11](#)
HIPLEX MSCF
 abnormal termination [184](#)
 adjusted network generation [113](#)
 architecture [12](#)
 BCAM dependencies [113](#)
 communication, prerequisites [113](#)

HIPLEX MSCF

- connection problems 166
 - function network 24
 - hardware connection 19
 - HIPLEX MSCF/RFA relationship 100
 - installation 110
 - installation files 110
 - loss of connection/clear-down 169
 - network, architecture 19
 - processor identification 112
 - product maintenance in the network 109
 - start 144
 - start time 111, 114
 - subsystems 109
 - terminate 149
 - type of function network 24
- HIPLEX MSCF, modify environment 206
- HIPLEX OP 42
- home pubset 288
- HOST-NAME 112
- HOST-PRIORITY
- MSCF configuration parameter 121
- HSMS 42, 68

I

- IMON 110
- IMPORT-PUBSET command 137
- indirect addressing 78
- example 80
- information
- on MSCF connection 150
 - on shared pubset 156
 - on XCS configuration 161
- information functions
- command overview 164
- information on
- NSM and partner systems 277
- input/output errors
- access to watchdog file 181
- installation
- of NSM subsystem 110
 - of the HIPLEX MSCF subsystems 109
 - of the MSCF subsystem 110
 - of XCS-TIME subsystem 110

J

- job checking 86
- job control in the network 75
- job initiation and termination 78
- job management and scheduling system (AVAS) 96
- job management system 74
- job checking 86
 - job variables 75
- job monitoring by job variable 85, 89
- job variable 75
- application example 76
 - conditional job control 86
 - conditional statement 86
 - job/program monitoring 89
 - monitoring IMPORT-/EXPORT-PUBSET 92
- job variables
- MONJV for monitoring the MSCF connection to a network partner 95
- jobs in the MSCF network 85
- JOIN file 346
- join reconfiguration 39

L

- LCS network 29
- generation 129
 - loss of connection 168
 - partner type 29
- LCS partner, failure 172
- LEAVE reconfiguration 210
- leave reconfiguration 39
- LEAVE-LIMIT
- MSCF configuration parameter 122
- level global storage area 182
- Live Monitor 71
- load network 34
- LOCAL-PASSWORD
- MSCF configuration parameter 122
- lock
- identification of DLM 43
 - recovery 44
- lock modes 43
- lock requirement 43
- lock value block (LVB) 44

- long connection 21, 342
 - communication 22
- loss of connection
 - between master and slave processor 169
 - between two slave processors 169
 - LCS/CCS network 168
 - shared pubset network 169
 - XCS network 171
- M**
- macro
 - DPSHSHP 310
 - MCSINFO 317
- macros 309
- main memory 194
 - caching 196
- MAIN-MEMORY 194
- MASTER 133
- master change 134, 151, 172, 342
 - failed 178
 - rejected 178
 - targeted 137
- master change import 173
 - termination 174
- master export
 - with master change 155
 - without master change 155
- master import of pubset 153
- master processor 342
 - select 133
- MAX-MSCF-DELAY 114, 149
- MCSINFO macro 317
- MCXSPXCS 140
 - system parameter 120
- memory saturation by server tasks 166
- mixed network, BACKUP-MASTER 133
- modify
 - configuration 293
 - MSCF network 293
- MODIFY-MASTER-CATALOG-ENTRY
 - command 131, 140
- MODIFY-MSCF-CONNECTION 203
- MODIFY-MSCF-ENVIRONMENT 206
- MODIFY-NSM-ENVIRONMENT 212
- monitoring characteristics, coordinate 180
- monitoring job variable (MONJV) 93
- monitoring programs (SM2) 192
- MRSCAT 60, 343
 - accessibility 62
 - catalog ID 62
 - connection setup 147
 - enter shared pubset 131
 - entry 62
- MSCF
 - abnormal system termination 186
 - activate 299
 - administration commands 143
 - basic functions 13
 - commands 201
 - communication network 144
 - configuration file 111, 216
 - connection clear-down 148
 - connection setup 147
 - coordination with BCAM 145
 - deactivate 305
 - define configuration parameters 216
 - display connections 150
 - DSSM subsystem catalog 111
 - establish connection 293
 - installation 110
 - job checking 86
 - job distribution 78
 - job management system 74
 - macros 309
 - modify communication network
 - dynamically 293
 - modify configuration parameters 206
 - modify network configuration 293
 - modify partner-specific monitoring parameters 203
 - release Cluster Recovery Lock 214
 - SCA implementation 154
 - server task 128, 130
 - start 144, 299
 - start at BS2000 startup 114
 - subsystem declaration 111
 - terminate 149, 305
 - termination by registered function 185

MSCF

- trace file [208, 219](#)
- traces [219](#)
- MSCF (SM2 monitoring program) [192](#)
- MSCF configuration parameter
 - ABORT-LIMIT [121](#)
 - FADING-INTERVAL [121](#)
 - FAIL-DETECTION-LIMIT [123](#)
 - HOST-PRIORITY [121](#)
 - LEAVE-LIMIT [122](#)
 - LOCAL-PASSWORD [122](#)
 - NOTIFY-BY-MAIL [122](#)
 - NUMBER-OF-SERVERS [128](#)
 - RECOVERY-START [124](#)
 - SERVER-TASK-LIMIT [128](#)
 - TRACE-FILE [122](#)
 - USER-TERM-LIMIT [128](#)
 - XCS-NAME [122](#)
- MSCF connection
 - modify dynamic [203](#)
 - stop [303](#)
- MSCF problem
 - notification by email [211, 221](#)
- MSCF server task [343](#)
 - abnormal termination [185](#)
- MSCF structure [12](#)
- MSCF system task
 - abnormal termination [185](#)
- MSCF task, abnormal termination [185](#)
- MSCF trace file [165, 343](#)
- MXCM task, abnormal termination [185, 186](#)

N

- network cooperations [24](#)
- network entry, reject partner [180](#)
- network generation, adjusted [113](#)
- network time in the XCS network [46](#)
- network types [24, 27](#)
- non-relevant partner [137, 343](#)
- NOTIFY-BY-MAIL
 - MSCF configuration parameter [122](#)
- NOTIFY-BY-MAIL (operand) [163](#)

NSM

- configuration file [141](#)
- installation [110](#)
- parameterization [141](#)
- subsystem [44](#)
- XCS function [36](#)
- NSM (SM2 monitoring program) [192](#)
- NSM configuration file [222](#)
- NSM configuration parameters
 - modify dynamically [212](#)
 - specify [222](#)
- NUMBER-OF-SERVERS
 - MSCF configuration parameter [128](#)

O

- openSM2 [189](#)
- operating modes, special [137](#)
- operational states of a shared pubset [157](#)
- overview
 - of network clear-down [338](#)
 - of network setup [337](#)

P

- parallel batch runs (example) [45](#)
- parallel HIPLEX [48, 194, 199](#)
 - caching the home pubsets [59](#)
 - DAB partition [52](#)
 - error recovery [57](#)
 - GS volume [54](#)
 - GSUSAGE setting [49](#)
 - mixed forms [48](#)
 - startup example [55](#)
 - switch GS partition to other processors [57](#)
 - termination of XCS participation [56](#)
- parameterization
 - MSCF [109, 111](#)
 - NSM [109, 141](#)
 - XCS-TIME [109](#)
- partner
 - non-relevant [137](#)
 - reject network entry [180](#)
 - relevant [137](#)
- partner crash detection, erroneous [179](#)

- partner monitoring 32, 137
 - example 44
 - partner system 288
 - partner type
 - CCS network 30
 - LCS network 29
 - shared pubset network 32
 - XCS network 34
 - password coding 130
 - performance network 34
 - PFA 194
 - PFA concept 343
 - prevalidation 334
 - processor identification 112
 - PROCESSOR name 112
 - program monitoring by JV 89
 - protocols 113
 - pubset
 - configuration file 344
 - determine sharers 285
 - set up as shared pubset 132
 - startup 60
 - pubset-specific MONJV 93
- R**
- radio clock 46
 - RCA (Remote Catalog Access) 101
 - README file 15
 - Readme file 15
 - reconfiguration
 - number 38
 - of GS volumes 59
 - timeout 186
 - type 39
 - RECOVERY-START
 - MSCF configuration parameter 124
 - network-wide setting 125
 - partner-specific setting 125
 - settings 124
 - RECOVERY-START=*SECURE 71
 - registered function 44, 344
 - rejection, network entry of partner 180
 - Related publications 17
 - RELEASE-CLUSTER-RECOVERY-LOCK 183, 214
 - relevant partner 137, 345
 - reliability of the disk monitoring 135
 - relocation of applications 42
 - Remote File Access (RFA) 100
 - interaction with MSCF 100
 - remove crashed sharer 174
 - requirements, shared pubset network 152
 - resources, release failed 178
 - RFA 100
- S**
- SCA 154
 - SECOS 82
 - SERSLOG file 165
 - server task 128, 343
 - bottleneck 166
 - distributed functions 21
 - number 128
 - SERVER-TASK-LIMIT
 - MSCF configuration parameter 128
 - set up connection to partner system 293
 - SET-MSCF-ENVIRONMENT 216
 - SET-NSM-ENVIRONMENT 222
 - SET-XCS-PUBSET 225
 - Shareable Private Disk (SPD) 103
 - shared file system 65
 - synchronization of file access 66
 - synchronization of file opening 65
 - shared GS 48
 - shared GS volume 48
 - shared GSVOL 200
 - shared partitions 48
 - shared PLAM 67
 - shared pubset 31
 - access to user catalog 154
 - CATID 151
 - clear down network 151, 155
 - commands for managing 202
 - declare as XCS pubset 225
 - determine sharers SHOW-SHARED-PUBSET 285
 - disk monitoring 32

- shared pubset
 - ENCRYPTION 338
 - enter in the MRSCAT 131
 - errored configuration 135
 - establish network 151
 - explain 132
 - export 155
 - FSHARING 136
 - hanging occupation 187
 - hardware caching 194
 - import 152
 - import with master change 154
 - master import 153
 - mode, conditions 135
 - operational states 157
 - performance 193
 - PFA 194
 - processor crash 172
 - retrieve information 156
 - slave import 153
 - slave sharer 31
 - Speed Catalog Access (SCA) 154
 - SYSEAM 136
 - system files 136
 - TEMPFILE 135
 - VM mode 136
- shared pubset master 31, 131
 - change at shutdown 155
 - define 133
- shared pubset network
 - and SECOS 33
 - and storage management/backup
 - functions 33
 - generation 131
 - loss of connection 169
 - partner monitoring 32, 137
 - partner type 32
 - processor crash during DAB caching 53
 - requirements 152
 - time until detection of partner crash/initiation of reconfiguration 177
- shared resource 19, 345
 - show occupants 184
- SHARED-PVS (SM2 monitoring program) 192
- sharer
 - crash 173
 - remove manually 174
- short connection 21, 346
 - communication 22
- SHOW-FILE-ATTRIBUTES command 107
- SHOW-MASTER-CATALOG-ENTRY command 154
- SHOW-MSCF-CONFIGURATION 227
- SHOW-NSM-CONFIGURATION 277
- SHOW-PUBSET-ATTRIBUTES command 156
- SHOW-PUBSET-PARAMETER command 154
- SHOW-SHARED-PUBSET 285
- SHOW-SHARED-PUBSET command 154
- SHOW-SYSTEM-INFORMATION 47
- SHOW-XCS-OCCUPATION 290
- simultaneous update
 - of all processors 65
- slave crash processing 173
- slave import of pubset 153
- SM pubset 69
- SM2 189
 - implementation requirements 190
 - monitoring programs 192
 - reports 191
 - restrictions 191
 - screen output 191
 - standard SM2 output file 191
 - users 190
- SoftBooks 16, 351
- Software Monitor 189
- SPD (shareable private disk) 103
 - capability 103
 - commands 107
 - conditions for use 105
 - notes on operation 106
- Speed Catalog Access 154
- SPVS, see shared pubset 31
- standby mode 73, 346
- standby system 73
- start at BS2000 startup 114
- start of HIPLEX MSCF
 - coordination with BCAM start 145
- start of MSCF 299

- start of MSCF subsystem
 - coordination with BCAM start 145
 - from BCAM-SOF file 145
 - start time 114
 - START-MSCF-CONNECTION 293
 - START-SUBSYSTEM command 299
 - status of XCS network 158
 - STOP-MSCF-CONNECTION 303
 - STOP-SUBSYSTEM command 305
 - subsystem
 - activate 299, 305
 - deactivate 305
 - declaration 110
 - initialize 299
 - unload 305
 - SUBSYSTEM-INFO-FILE (operand) 111
 - SVL 346
 - switch GS partition to other processors 57
 - switchover unit 69
 - synchronization of file accesses (example) 45
 - synchronized time 46
 - SYSEAM 136
 - SYSID 112
 - SYSSII.NSM 141
 - SYSSSC.MSCF 111
 - system files on shared pubsets 136
 - system ID 288
 - system identification (SYSID) 112
 - system parameter
 - FSHARING 136
 - MCXSPXCS 120, 140
 - TEMPFILE 135
 - system termination by MSCF 186
 - system version 288
- T**
- TCP/IP 113
 - TEMPFILE system parameter 135
 - termination
 - HIPLEX MSCF 149
 - master change import 174
 - time in network 46
 - time synchronization (functions) 46
 - timeout of reconfiguration 186
 - token idling 222
 - TRACE-FILE
 - MSCF configuration parameter 122
 - TSOSCAT 60
- U**
- unload subsystem 305
 - user catalog 346
 - USER-TERM-LIMIT
 - MSCF configuration parameter 128
 - UTC (internal time coordinate) 47
- V**
- version link-ups 109
 - vital-sign message 31
 - definition 31
 - error when reading 182
 - error when writing 181
 - VM mode with shared pubset 136
 - volume serial number 347
 - VSN 347
 - VTOC catalog entries, display 107
- W**
- wait time for reestablishment of MSCF
 - connection 170
 - watchdog file 31, 347
 - failure 182
 - input/output errors 181
 - name 182
 - watchdog mechanism, block 183

X

XCM [36](#)

XCS

define pubset [225](#)

disconnection of processor from network [184](#)

functions [36](#)

generation notes [140](#)

hanging occupation [187](#)

MONJV [94](#)

occupation by TU tasks [292](#)

pubset [34](#), [347](#)

pubset handling [161](#)

show information on occupation [290](#)

tasks using XCS [292](#)

time [46](#)

XCM [36](#)

XCM configuration manager [36](#)

XCS network [34](#), [158](#)

applications (functions) [35](#)

configuration file [117](#)

consistent network [38](#)

determine status [158](#)

display configuration [161](#)

entry [158](#)

join existing network [159](#)

leave [160](#)

loss of connection [171](#)

maximum number of sharers [158](#)

partner type [34](#)

sharer crash [175](#)

time until detection of partner crash/initiation of
reconfiguration [177](#)

XCS time [47](#)

XCS-NAME

MSCF configuration parameter [122](#)

XCS-TIME

installation [110](#)

subsystem [41](#), [46](#)

subsystem declaration [110](#)

XPM (XCS function) [37](#)

XTS (XCS function) [36](#)