

FOCUS for S/390

MSO Installation and Technical Reference Guide
Version 7.2

Contents

1	Features and Components	1-1
	Installation Requirements	1-2
	MSO Features.....	1-2
	VTAM® Access to MSO.....	1-3
	Shared MSO Access.....	1-3
	Multiple MSO Regions	1-3
	Usage Accounting	1-3
	MSO Initialization and Termination Exit.....	1-3
	User Common Memory.....	1-4
	Information Builders Snapshot Facility.....	1-4
	The MSO Console.....	1-4
	Operator Commands	1-4
	Security	1-5
	The FastLoad Facility.....	1-5
	The MSO Resource Manager.....	1-6
	Trace Management.....	1-6
	Viewing the Output of Your Batch Job from the SDSF Main Panel.....	1-6
	MSO Load Balancing.....	1-6
	Failure Processing	1-6
	MSO FOCUS Operation.....	1-7
	Profile Operation	1-7
	The Terminal Operator Environment	1-7
	Attention Key Support.....	1-9
	The DYNAM Command.....	1-10
	Public and Private DDname for MSO	1-10
	Printer Support	1-10
	REBUILD/INDEX	1-10
	Graphics	1-10
	Submitting Batch Jobs.....	1-11
	Simultaneous Usage	1-11
	Retrieving Environmental Information	1-11
	SmartMode® for FOCUS®	1-11
	HiperFOCUS	1-11
2	Preparing for MSO	2-1
	MSO Issues.....	2-2
	OS/390® Issues	2-4
3	MSO Technical Description	3-1
	Technical Overview.....	3-2
	Control Tasks	3-3
	User Tasks	3-5

CICS Access to MSO	3-7
MSO Resource Manager	3-7
The FastLoad Facility	3-8
Internal APF Authorization	3-9
4 MSO Capacity Planning and Performance	4-1
Virtual Memory Requirements	4-2
Below-the-line Memory Requirements	4-2
Above-the-line Memory Requirements	4-4
Region Size	4-4
Specifying Region Size	4-4
Monitoring Region Requirements	4-5
Real Memory Requirements	4-6
Using Expanded Storage	4-6
A Typical MSO Region	4-6
Preventing Thrashing	4-7
MVS™ Performance Group Recommendations	4-7
Data Set Allocation Limitations	4-8
OS/390 DDNAME Limitations: Task Input/Output Table (TIOT) Restrictions	4-8
MSO DDNAME Limitations	4-8
5 The MSO Configuration File	5-1
Global Configuration	5-2
Connectivity Options	5-7
MSO Console Security	5-8
MSO Diagnostic Facilities	5-11
The Service Block	5-12
MSO Load Balancing Configuration Parameters	5-15
Obsolete Configuration File Commands	5-16
6 Installing MSO	6-1
Install FOCUS	6-3
System Programmer Tasks	6-3
Authorize the FOCUS Load Library	6-3
Make the MSO Address Space Non-Swappable	6-3
Provide JES2 Banner Support	6-4
Install the IBI Subsystem, If Required	6-4
Set up System Security	6-4
Assign a VTAM APPLID to MSO	6-5

File Set Up.....	6-6
Unload the MSO Data sets	6-6
Copy FUSELIB.LOAD into FOCLIB.LOAD	6-6
Create MSPUT and MSGET	6-6
Protect Module MSDEB	6-6
Create MSOPROF.....	6-7
Create member PROFILE in MSOPROF.....	6-7
Basic MSO Customization	6-7
Create the Configuration File	6-7
Create the MSO JCL	6-8
Set up MSOKILL JCL	6-10
Installing MSO Components	6-11
Install the MSO Console	6-11
Set Up CICS Access.....	6-11
Customize the VTAM Logon Screen	6-12
Install Exits and ZAPs, as Required	6-12
7 Load Balancing	7-1
Overview	7-2
MSO Load Balancing Requirements.....	7-3
MSO Load Balancing Approach	7-4
Load Balancing Sequence	7-4
Setting Load Balancing Defaults in the IBI Subsystem	7-6
MSO Load Balancing Configuration Parameters	7-6
Minimum Parameters for Load Balancing	7-7
Load Balancing Logon Procedures.....	7-10
VTAM Logon Request.....	7-11
TSO Logon Request	7-12
CICS Terminal Logon Request	7-12
CICS Load Balancing API	7-13
Planning for MSO Load Balancing	7-15
MSO Load Balancing Requirements.....	7-15
Load Balancing Defaults	7-15
Consistent Destination Environment.....	7-16
Define Load Balancing Group Names to SAF	7-17
Consistently use the APPLICATION Parameter in All Service Blocks.....	7-17
MSO Region Capacity.....	7-17
All MSO Regions Should Have a Console Available	7-17
Minimize Dividing the MSO Load Balancing Group	7-17
Operational Troubleshooting	7-18
CICS Problem Determination Procedures.....	7-20
Operational Benefits of MSO Load Balancing	7-20

8 MSO/CICS Installation	8-1
Preparing for Installation.....	8-2
Installation Requirements.....	8-2
Planning.....	8-2
Required Data Sets.....	8-4
MSO SASLIB Installation Guideline.....	8-5
MSO User Exits.....	8-5
Considerations for Multiple Address Spaces.....	8-5
Installation Steps.....	8-6
MSO Setup.....	8-6
CICS Setup.....	8-6
CICS Setup for MRO.....	8-9
Multiple Address Spaces.....	8-9
Installation Testing.....	8-10
Testing Basic Functionality.....	8-10
Testing Attention Key Support.....	8-10
Testing PLT Functions.....	8-11
Operations.....	8-11
MSO/CICS Startup.....	8-11
Reusable MSO/CICS Control Blocks.....	8-11
Startup for Attention Key Support.....	8-11
MSMT - MSO Master Terminal.....	8-12
CICS Shutdown.....	8-12
End-User Issues.....	8-13
Starting a FOCUS Session.....	8-13
Ending a FOCUS Session.....	8-13
Keyboard Locking.....	8-13
Attention Handling.....	8-14
Session Termination Key.....	8-14
MSO and Other CICS Programs.....	8-14
MSO/CICS Technical Description.....	8-14
Steady-State Operation.....	8-15
Initialization and Termination of the MSO Link.....	8-16
Initialization and Termination of a User Session.....	8-16
Attention Key Handling.....	8-18
MRO Issues.....	8-18
Transaction Security.....	8-19
Userid Processing.....	8-19
Summary of MSO Transactions and Programs.....	8-20
Troubleshooting.....	8-22
Diagnostic information.....	8-23

9	Testing and Logging on to MSO	9-1
	VTAM Access	9-2
	Interfacing with VTAM Session Managers.....	9-3
	CICS Access.....	9-3
	TSO Access	9-4
	Accessing Multiple MSO Regions	9-4
10	MSO Usage Accounting	10-1
	Enabling Usage Accounting.....	10-2
	Processing SMF Records Using FOCUS	10-2
	SMF Fields Recorded.....	10-3
	Modifying and Redirecting SMF Records.....	10-4
11	MSO Operator Commands.....	11-1
	MVS Operator Commands	11-2
	The STOP Command	11-2
	The MODIFY Command	11-2
	Examples	11-5
12	MSO Security.....	12-1
	RACF [®] Security for MSO.....	12-3
	CA-ACF2 [®] Security for MSO	12-4
	CA-TOP SECRET [®] Security for MSO.....	12-5
13	The MSO Console.....	13-1
	Overview	13-2
	Installation	13-2
	Installation Steps	13-2
	Security	13-4
	Tailoring Console Authority	13-5
	Operation.....	13-6
	Logon Screen.....	13-7
	Primary Option Menu.....	13-8
	Operator Commands	13-9
	Display User	13-9
	WHOHAS	13-12
	Display Servers	13-14
	Log Viewer.....	13-15
	Trace Handling.....	13-16
	Data Set Browser.....	13-19
	Help Facility	13-21
	MSO Monitoring and Statistics.....	13-24
	Messages for On Demand and Monitoring.....	13-29
	MSO Dynamic VTAM Re-configuration.....	13-30

14	Common Installation Problems and Solutions.....	14-1
	Diagnosing Installation Problems.....	14-2
	Diagnosing Storage Problems	14-3
	Diagnosing System Failures	14-3
	Summary of the Environment	14-4
	About the Abend	14-5
	About the MSO Address Space.....	14-6
	About the Failing Task.....	14-6
	Applying Maintenance	14-8
15	Site-Specific Customizations.....	15-1
	Exits.....	15-2
	MSIDVER: User Authentication Exit.....	15-2
	MSIDTR: Userid Translation Exit	15-2
	MSCXUID: Supply CICS Userid.....	15-2
	MSUSRXT: The MSO Initialization and Termination Exit.....	15-2
	DYNALLOC: The DYNAM ALLOC User Exit	15-3
	MSVTXT: VTAM Text User Exit	15-4
	SSUSRACT: VTAM Account Field Validation	15-4
	MSSMFXT: Customized SMF Records.....	15-4
	MSCXWRT: Customized CICS Termination Processing	15-5
	FOCUSID: Setting the FOCUS DBA Password	15-5
	User Written Subroutines	15-5
	User Common Memory.....	15-6
	MSODBLW: Retrieve the User Doubleword Address.....	15-6
	DDname Translation: The MSODDX Subroutine	15-6
	MSODDX Calling Environment	15-7
	Usage of MSODDX in a Typical Application.....	15-8
	MSODDX Parameter List	15-8
	Sample Assembler Call	15-11
	Link-Edit Example	15-11
	ZAPs.....	15-12
	IBISNAP -- Disable User Screen Display in IBISNAP Dumps.....	15-12
	SUBMITZ2 -- Enable a Site-Specific SUBMIT Exit.....	15-12
	DYNZAP -- Enable Allocation of WRITER INTRDR.....	15-12
	DYNEXZAP -- Enable the DYNAM ALLOC User Exit.....	15-12
	Environmental Subroutines and MSOINFO.....	15-13

A	MSO/CICS Cooperative Processing	A-1
	MSO FOCEXEC Cooperative Processing Service.....	A-3
	MSO/CICS Cooperative Processing Services	A-3
	How to Use the CMSOLGN Function	A-4
	How to Use the CMSORCV Function	A-6
	How to Use the CMSOREC Function.....	A-7
	How to Use the CMSOSTP Function.....	A-8
	How to Use the CMSORSM Function	A-8
	CMSORCV Function Codes	A-9
	Examples	A-10
	Reconnection Capability	A-10
	Suspend key	A-11
B	Sample JCL for Installing MSO and CICS.....	B-1
	Sample JCL for Installing MSO	B-2
	Sample JCL to Unload MSO Datasets	B-4
	Sample JCL for Installing CICS.....	B-5
C	Installing MSO-Supported Data Adapters.....	C-1
	MSO-Supported Data Adapters.....	C-2
D	Security Implementation for DB2.....	D-1
	Steps for Modifying DSN3SATH	D-2
	Modifying DSN3SATH.....	D-3
	Modifying DSN3SATH for RACF and CA-TOP SECRET.....	D-3
	Modifying DSN3SATH for CA-ACF2	D-6
	Link JCL for DSN3SATH.....	D-6
E	MSO Glossary of Terms	E-1
	MSO Glossary of Terms.....	E-2
	Index	I-1

Cactus, EDA, FIDEL, FOCCALC, FOCUS, FOCUS Fusion, Information Builders, the Information Builders logo, SmartMode, SNAPpack, TableTalk, and Web390 are registered trademarks and Parlay, SiteAnalyzer, SmartMart, WebFOCUS, and WorldMART are trademarks of Information Builders, Inc.

Acrobat and Adobe are registered trademarks of Adobe Systems Incorporated.

NOMAD is a registered trademark of Aonix.

UniVerse is a registered trademark of Ardent Software, Inc.

IRMA is a trademark of Attachmate Corporation.

Baan is a registered trademark of Baan Company N.V.

SUPRA and TOTAL are registered trademarks of Cincom Systems, Inc.

Impromptu is a registered trademark of Cognos.

Alpha, DEC, DECnet, NonStop, and VAX are registered trademarks and Tru64, OpenVMS, and VMS are trademarks of Compaq Computer Corporation.

CA-ACF2, CA-Datcom, CA-IDMS, CA-Top Secret, and Ingres are registered trademarks of Computer Associates International, Inc.

MODEL 204 and M204 are registered trademarks of Computer Corporation of America.

Paradox is a registered trademark of Corel Corporation.

StorHouse is a registered trademark of FileTek, Inc.

HP MPE/iX is a registered trademark of Hewlett Packard Corporation.

Informix is a registered trademark of Informix Software, Inc.

Intel is a registered trademark of Intel Corporation.

ACF/VTAM, AIX, AS/400, CICS, DB2, DRDA, Distributed Relational Database Architecture, IBM, MQSeries, MVS, OS/2, OS/390, OS/400, RACF, RS/6000, S/390, VM/ESA, and VTAM are registered trademarks and DB2/2, Hiperspace, IMS, MVS/ESA, QMF, SQL/DS, z/VM, z/OS, VM/XA, SQL/DS, and WebSphere are trademarks of International Business Machines Corporation.

INTERSOLVE and Q+E are registered trademarks of INTERSOLVE.

Orbit is a registered trademark of Iona Technologies Inc.

Approach and DataLens are registered trademarks of Lotus Development Corporation.

ObjectView is a trademark of Matesys Corporation.

ActiveX, FrontPage, Microsoft, MS-DOS, PowerPoint, Visual Basic, Visual C++, Visual FoxPro, Windows, and Windows NT are registered trademarks of Microsoft Corporation.

Teradata is a registered trademark of NCR International, Inc.

Netscape, Netscape FastTrack Server, and Netscape Navigator are registered trademarks of Netscape Communications Corporation.

NetWare and Novell are registered trademarks of Novell, Inc.

CORBA is a trademark of Object Management Group, Inc.

Oracle is a registered trademark and Rdb is a trademark of Oracle Corporation.

PeopleSoft is a registered trademark of PeopleSoft, Inc.

INFOAccess is a trademark of Pioneer Systems, Inc.

Progress is a registered trademark of Progress Software Corporation.

Red Brick Warehouse is a trademark of Red Brick Systems.

SAP and SAP R/3 are registered trademarks and SAP Business Information Warehouse and SAP BW are trademarks of SAP AG.

Silverstream is a trademark of Silverstream Software.

ADABAS is a registered trademark of Software A.G.

CONNECT:Direct is a trademark of Sterling Commerce.

Java, JavaScript, NetDynamics, Solaris, SunOS, and iPlanet are trademarks of Sun Microsystems, Inc.

PowerBuilder and Sybase are registered trademarks and SQL Server is a trademark of Sybase, Inc.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd.

Allaire and JRun are trademarks of Allaire Corporation.

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. In most, if not all cases, these designations are claimed as trademarks or registered trademarks by their respective companies. It is not this publisher's intent to use any of these names generically. The reader is therefore cautioned to investigate all claimed trademark rights before using any of these names other than to refer to the product described.

Copyright © 2001, by Information Builders, Inc. All rights reserved. This manual, or parts thereof, may not be reproduced in any form without the written permission of Information Builders, Inc.

Printed in the U.S.A.

Preface

This documentation describes how to use and install the FOCUS® Multi-Session Option (MSO). It is intended for end users, system administrators, and application developers.

How This Manual Is Organized

This manual includes the following chapters:

Chapter/Appendix		Contents
1	<i>Chapter 1</i>	Features and Components
2	<i>Chapter 2</i>	Preparing for MSO
3	<i>Chapter 3</i>	MSO Technical Description
4	<i>Chapter 4</i>	MSO Capacity Planning and Performance
5	<i>Chapter 5</i>	The MSO Configuration File
6	<i>Chapter 6</i>	Installing MSO
7	<i>Chapter 7</i>	Load Balancing
8	<i>Chapter 8</i>	MSO/CICS® Installation
9	<i>Chapter 9</i>	Testing and Logging on to MSO
10	<i>Chapter 10</i>	MSO Usage Accounting
11	<i>Chapter 11</i>	MSO Operator Commands
12	<i>Chapter 12</i>	MSO Security
13	<i>Chapter 13</i>	The MSO Console
14	<i>Chapter 14</i>	Common Installation Problems and Solutions
15	<i>Chapter 15</i>	Site-Specific Customizations
A	<i>Appendix A</i>	MSO/CICS Cooperative Processing
B	<i>Appendix B</i>	Sample JCL for Installing MSO and CICS
C	<i>Appendix C</i>	Installing MSO-Supported Data Adapters
D	<i>Appendix D</i>	Security Implementation for DB2®
E	<i>Appendix E</i>	MSO Glossary of Terms

Documentation Conventions

The following conventions apply throughout this manual:

Convention	Description
<code>THIS TYPEFACE</code> or <code>this typeface</code>	Denotes syntax that you must enter exactly as shown.
<i>this typeface</i>	Represents a placeholder (or variable) in syntax for a value that you or the system must supply.
<u>underscore</u>	Indicates a default setting.
<i>this typeface</i>	Represents a placeholder (or variable) in a text paragraph, indicates a cross-reference, or emphasizes an important term.
this typeface	Highlights file names and commands (in a text paragraph) that must be lowercase.
this typeface	Indicates buttons, menu items, and dialog box options you can click or select.
Key + Key	Indicates keys that must be pressed simultaneously.
{ }	Indicates two choices from which you must choose one. You type one of these choices, not the braces.
[]	Indicates a group of optional parameters. None are required, but you may select one of them. Type only the information within the brackets, not the brackets.
	Separates two mutually exclusive choices in a syntax line. You type one of these choices, not the symbol.
...	Indicates that you can enter a parameter multiple times. Type only the parameters, not the ellipsis points (...).
.	Indicates that there are (or could be) intervening or additional commands.

Related Publications

FOCUS for S/390® OS/390® and MVS® Installation Guide (DN1000994.1100)

See the *Information Builders Technical Publications Catalog* for the most up-to-date listing and prices of technical publications, plus ordering information. To obtain a catalog, contact the Publications Order Department at (800) 969-4636.

You can also visit our World Wide Web site, <http://www.informationbuilders.com>, to view a current listing of our publications and to place an order.

Customer Support

Do you have questions about MSO?

Call Information Builders Customer Support Service (CSS) at (800) 736-6130 or (212) 736-6130. Customer Support Consultants are available Monday through Friday between 8:00 a.m. and 8:00 p.m. EST to address all your MSO questions. Information Builders consultants can also give you general guidance regarding product capabilities and documentation. Please be ready to provide your six-digit site code number (xxxx.xx) when you call.

You can also access support services electronically, 24 hours a day, with InfoResponse Online. InfoResponse Online is accessible through our World Wide Web site, <http://www.informationbuilders.com>. It connects you to the tracking system and known-problem database at the Information Builders support center. Registered users can open, update, and view the status of cases in the tracking system and read descriptions of reported software issues. New users can register immediately for this service. The technical support section of www.informationbuilders.com also provides usage techniques, diagnostic tips, and answers to frequently asked questions.

To learn about the full range of available support services, ask your Information Builders representative about InfoResponse Online, or call (800) 969-INFO.

Information You Should Have

FOCUS:

To help our consultants answer your questions most effectively, be ready to provide the following information when you call:

- Your six-digit site code number (*xxxx.xx*).
- The FOCEXEC procedure (preferably with line numbers).
- Master file with picture (provided by CHECK FILE).
- Run sheet (beginning at login, including call to FOCUS), containing the following information:
 - ? RELEASE
 - ? FDT
 - ? LET
 - ? LOAD
 - ? COMBINE
 - ? JOIN
 - ? DEFINE
 - ? STAT
 - ? SET/? SET GRAPH
 - ? USE
 - ? TSO DDNAME OR CMS FILEDEF
- The exact nature of the problem:
 - Are the results or the format incorrect? Are the text or calculations missing or misplaced?
 - The error message and code, if applicable.
 - Is this related to any other problem?
- Has the procedure or query ever worked in its present form? Has it been changed recently? How often does the problem occur?
- What release of the operating system are you using? Has it, FOCUS, your security system, or an interface system changed?
- Is this problem reproducible? If so, how?

- Have you tried to reproduce your problem in the simplest form possible? For example, if you are having problems joining two data sources, have you tried executing a query containing just the code to access the data source?
- Do you have a trace file?
- How is the problem affecting your business? Is it halting development or production? Do you just have questions about functionality or documentation?

User Feedback

In an effort to produce effective documentation, the Documentation Services staff at Information Builders welcomes any opinion you can offer regarding this manual. Please use the Reader Comments form at the end of this manual to relay suggestions for improving the publication or to alert us to corrections. You can also use the Document Enhancement Request Form on our Web site, <http://www.informationbuilders.com>.

Thank you, in advance, for your comments.

Information Builders Consulting and Training

Interested in training? Information Builders Education Department offers a wide variety of training courses for this and other Information Builders products.

For information on course descriptions, locations, and dates, or to register for classes, visit our World Wide Web site (<http://www.informationbuilders.com>) or call (800) 969-INFO to speak to an Education Representative.

CHAPTER 1

Features and Components

Topics:

- Installation Requirements
- MSO Features
- Failure Processing
- MSO FOCUS Operation
- SmartMode for FOCUS
- HiperFOCUS

This chapter describes the requirements for installation and the options available for the FOCUS Multi-Session Option (MSO). Chapter 6, *Installing MSO*, includes the actual installation instructions.

Installation Requirements

- The MSO version of FOCUS runs on any IBM® or plug-compatible computer that supports OS/390.
- This installation guide applies to FOCUS Version 7.2 and higher.
- MSO supports CICS Release 4.1 and above.

MSO Features

This section contains brief descriptions of the following features:

- VTAM access to MSO
- Shared MSO access
- Multiple MSO regions
- Usage accounting
- MSO initialization and termination exit
- User common memory
- Information Builders® Snapshot Facility
- The MSO Console
- Operator commands
- Security
- The FastLoad Facility
- The MSO Resource Manager
- Trace Management
- Viewing the output of your batch job from the SDSF main panel
- MSO Load Balancing

VTAM Access to MSO

Access to MSO is available from any VTAM terminal by logging on to the MSO applid set up by a VTAM system programmer.

Shared MSO Access

VTAM, CICS, and TSO users may use a single MSO region. The CICS users may reside in one or multiple CICS regions.

Multiple MSO Regions

Multiple MSO regions can run simultaneously on a single OS/390 system. Users accessing MSO from VTAM, CICS, and TSO may choose which MSO region to use. VTAM users specify the appropriate LUname, CICS users run the appropriate transaction, and TSO users allocate the appropriate communications data sets. Each different MSO region must have different communications data sets, which are shared by all CICS regions and TSO users accessing that region.

Usage Accounting

MSO provides the facilities for a site to monitor and trap usage accounting information on a per-user basis by cutting SMF records from an exit supplied by MSO. Your site can modify this exit to suit any in-house accounting package. See Chapter 10, *MSO Usage Accounting*, for detailed information.

MSO Initialization and Termination Exit

MSO provides a user exit that is invoked when MSO is brought up or terminated. This allows initialization or termination of any site-specific code when an MSO address space starts or ends. See Chapter 15, *Site-Specific Customization*, for detailed information concerning the MSUSRXT exit.

User Common Memory

If multiple users in an MSO address space must communicate with each other or coordinate services between themselves, MSO provides common memory to use in conjunction with user-written subroutines. MSO provides eight bytes of memory, the address of which can be obtained in a variety of ways. The most common use of such memory would be to store the address of GETMAINED memory obtained by the MSUSRXT user-written exit. This memory can then be shared among all the users in the MSO address space. More information on user common memory and the MSODBLW subroutine is present in Chapter 15, *Site-Specific Customization*.

Information Builders Snapshot Facility

In the event of a system failure, MSO provides the Snapshot Facility, MSO's own dump snapshot formatter, which enhances problem determination and serviceability by assisting dump interpretation by support personnel. This facility is automatic, and provides a clear record of a problem even when no dump is available.

The MSO Console

MSO provides a full-screen real-time monitor console, accessed directly via VTAM. You can use the MSO Console to:

- Display users, cancel sessions, and change priorities.
- Display CPU, storage, and EXCP statistics.
- Display the current contents of a user's screen.
- Control the MSO address space via operator commands.
- Display, browse, and control address space and user data set allocations.

All of these capabilities require the appropriate authorization, either in the MSO configuration file or from an external security package.

Operator Commands

MSO provides the MVS operator with a STOP command to stop a given MSO address space, as well as MODIFY commands to cancel individual users, monitor CPU and EXCP statistics, control traces, and perform other tasks. See Chapter 11, *MSO Operator Commands*, for detailed information.

Security

Security in MSO is implemented using IBM's System Authorization Facility (SAF), which allows a variety of security systems (RACF, CA-ACF2, etc.) to be employed transparently to MSO. A userid is obtained from the TSO userid, or the CICS signon ID (usually obtained from the CICS CSSN or CESN transaction), or the VTAM logon screen. This userid is then used for security purposes for the task assigned to the user.

This userid, when obtained from CICS or TSO, is assumed to be already authenticated. Because VTAM does not use CICS or TSO, SAF is used to authenticate the userid and password that come from the MSO VTAM logon screen. This VTAM-only first level authentication can be bypassed by using an MSO exit called MSIDVER, which allows an MSO site to do its own first level authentication for VTAM MSO users. A sample MSIDVER assembler program is available as member MSIDVER in the MSO.DATA partitioned data set. To implement this exit, you must create an OS/390 module called MSIDVER and put this module into the authorized MSO library.

Another MSO user exit allows a site to translate userids obtained by CICS, TSO, or the Information Builders VTAM logon screen to security userids known by MSO. To use this exit, you must create an OS/390 module called MSIDTR and put this module into the authorized MSO library. The exit accepts the original userid as input, and has as output the security ID by which the user will be known, and the prefix ID by which any dynamically allocated files will be allocated (such as dynamically allocated FOCUS files). A sample MSIDTR program is present as member MSIDTR in the MSO.DATA partitioned data set.

If external security is in effect, a userid that is not known to SAF will not be allowed to log on to MSO. The MSIDTR exit may be used in conjunction with the MSIDVER exit.

For information on how to implement MSO security with a particular security package, see Chapter 12, *MSO Security*.

The FastLoad Facility

The MSO FastLoad Facility preloads a copy of any module, such as FOCUS, and eliminates further loads of the module. MSO provides a unique copy to each user as needed, greatly speeding up logon to MSO and startup of FOCUS applications.

The MSO Resource Manager

The MSO Resource Manager balances the resources allocated to MSO users, and allows the assignment of relative priorities to users. This prevents a resource-intensive task, or a looping task, from monopolizing all resources available to the region, and allows all work in the region to proceed concurrently.

Trace Management

MSO provides tools to activate and manage the various traces that Information Builders products and options may generate. These tools are available to the OS/390 operator and users of the MSO Console.

Viewing the Output of Your Batch Job from the SDSF Main Panel

You can view the output of your batch job from the SDSF main panel. From an MSO user standpoint this practically eliminates the need for TSO, ISPF and SDSF and provides an easy transition to MSO for TSO users.

MSO Load Balancing

Load Balancing allows MSO administrators to distribute FOCUS users across multiple MSO regions to reduce contention for system resources and provides a transparent and efficient way to distribute users across multiple MSO regions.

Failure Processing

When a user who is logged on to MSO exits from FOCUS or abends, all resources associated with the session are freed, including data areas, programs, control blocks, and data sets.

Information Builders Snapshot Facility is provided in case of an abend (see Chapter 16, *Common Installation Problems and Solutions*). The facility supplements the standard OS/390 dump capabilities by producing a summary mini-dump whenever a task in the MSO address space abends. This snapshot aids in failure identification and diagnosis.

In addition, trace facilities, for example, FSTRACE, are provided by individual FOCUS components to assist in problem determination. These traces can be controlled from the OS/390 system console or from the MSO Console, as well as from individual FOCUS sessions.

MSO FOCUS Operation

All of the FOCUS features described in the FOCUS documentation are available under MSO, except as documented in this section. TABLE, MAINTAIN, MODIFY, Dialogue Manager, TED, FOCCALC, and the Talk Technologies are present in MSO.

Profile Operation

In addition to the standard profiles executed by FOCUS (See *FOCUS for S/390® Overview and Operating Environments*), two additional profiles are executed when running MSO (See the *OS/390 and MVS Installation Guide*). The profiles follow the execution of SHELPROF in the following order:

- In the JCL used to start MSO, the partitioned data set allocated to ddname MSOPROF may have a member PROFILE. If this member is present it will be executed for all users who log on. This profile is executed in the MSO environment only.
- In the JCL used to start MSO, the partitioned data set allocated to ddname MSOPROF may have members with different userids as the member names. The member corresponding to the MSO security userid will be executed for that user. This profile is executed in the MSO environment only.
- In the partitioned data set allocated to FOCEXEC by the user, there may be a member PROFILE, which will be executed for the user. This is a standard FOCUS profile.
- The ddname FOCEXEC may be allocated in the MSO JCL or by a standard FOCUS PROFILE. If the partitioned data set userid.FOCEXEC.DATA exists and the FOCEXEC ddname has not been allocated, the FOCEXEC ddname will be allocated by default to this data set. The same is true for the MASTER ddname.
- In the JCL used to start MSO, the partitioned data set allocated to ddname FOCEXEC has a member SHELPROF that will be executed for all users who log on. This is a standard FOCUS profile, which is described further in *FOCUS for S/390 Overview and Operating Environments*.

The Terminal Operator Environment

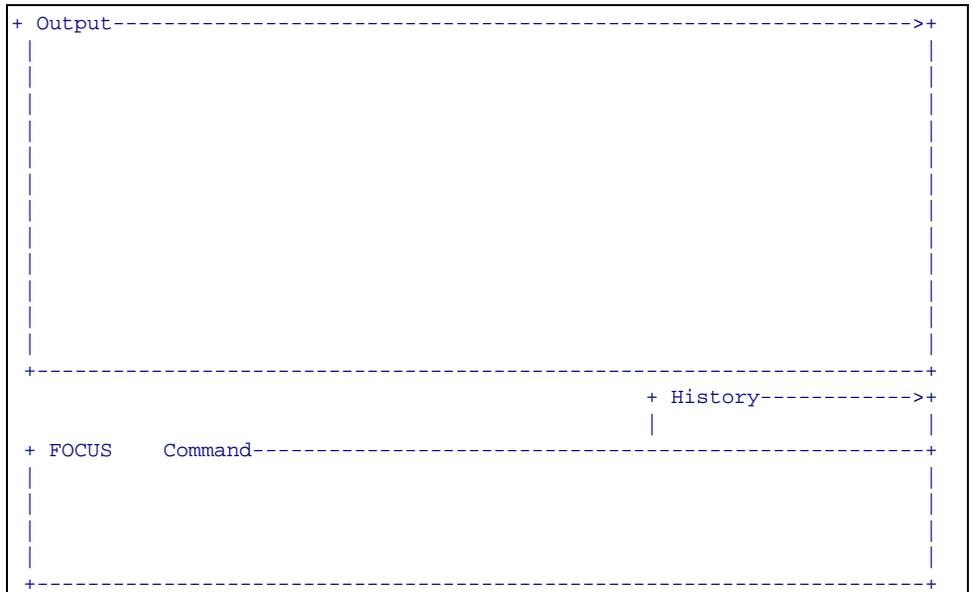
MSO FOCUS is a full-screen-only product. MSO FOCUS users are always in the Terminal Operator Environment (TOE), which provides windows for various terminal activities such as data input, output, and errors. The Terminal Operator Environment is described in detail in *Overview and Operating Environments*.

The figure on the following page shows the default Terminal Operator Environment screen with three windows: the Output window, the FOCUS Command window, and a History window. All input is done in the Command window. You can navigate between windows using the TAB key or the PF12 key.

If the keyboard is locked, the cursor may not be properly positioned on the Command window. The RESET and TAB keys can be used to position the cursor correctly in the Command window.

All output is sent to the Output window. If the output is too large to fit in the Output window, you must press the ENTER key or PF8 to display the rest of the output. When there is no more output to display, the cursor returns to the Command window.

Errors are displayed in a dynamically created window positioned in the Output window. The command that caused the error remains in the Command window. To clear the Error window, just erase the command (by blanking it out or pressing the Erase EOF key) and press ENTER. Alternatively, you can correct the command in error and press ENTER.



FOCUS attempts to draw the TOE windows with solid borders on terminals that are capable of displaying them. Otherwise, broken lines, drawn with standard 3270 characters, are used to create the window borders.

MSO issues a query command to the terminal to determine whether the terminal is capable of drawing solid lines. In some cases, a terminal will erroneously report the ability to display solid lines, resulting in various errors, most commonly PROGnnn communications errors or incorrectly drawn boxes. In this case, the command

```
SET SBORDER=OFF
```

may be issued, either from the terminal, or in a profile (see *Profile Operation* on page 1-7). This command may also be used if you prefer broken borders to solid borders.

It is also possible that a terminal will reply that it cannot display solid lines when it actually can. In this situation, the command

```
SET SBORDER=ON
```

may be issued.

Please note that some PC-based terminal emulators require enabling the APL character set in order to activate this capability.

Note:

The TOE output screen will buffer any output messages during FOCEXEC program execution until control is returned to command level FOCUS. If the TOE output buffer fills to capacity, the screen will appear, even if control is still within a FOCEXEC. The window command

```
WINDOW SET CONTINUE ON
```

prevents this from happening.

A full set of commands is available for manipulating and customizing the Terminal Operator Environment. Refer to FOCUS documentation for a complete description of TOE and the TOE commands.

Attention Key Support

MSO sessions can be terminated by pressing an attention key in VTAM, CICS, and TSO.

- From a VTAM, CICS, or TSO terminal, use the 3270 Attention key to interrupt a FOCUS command or end an MSO session.
- Under CICS, MSO allows sites to specify a session termination key via the CICSBREAK parameter in the MSO configuration file. See Chapter 5, *The MSO Configuration File*, for further details. When you press the designated key, the MSO session is terminated and all the session resources are freed as if there was a normal exit from FOCUS.

Note:

To exit from FOCUS normally, use the FIN command rather than the attention key.

The DYNAM Command

The DYNAM command can be used to allocate and free MVS data sets. For information about the DYNAM Command consult the *FOCUS for S/390 Overview and Operating Environments* documentation.

Public and Private DDname for MSO

MSO may be used in different ways. There are users who run production applications and use the Global allocations for the MSO region as a whole. There are also users who need to allocate their own MASTER or FOCEXEC PDSs. These users can use PRIVATEDDD to perform allocations of PDSs for their exclusive use.

PRIVATEDDD allows users to put their allocations in front of, in back of, or override the allocations for the MSO region. Applications can be tested, or new ones put into production without affecting the region as a whole.

Printer Support

To send output to a remote printer, use the DYNAM command. With DYNAM, the DEST parameter can be specified to cause a file to be sent to a JES printer.

REBUILD/INDEX

The INDEX option of REBUILD must be executed as a batch job, using DD statements for the sort files. The sort files cannot be allocated via the DYNAM command.

For an example of REBUILD/INDEX, see *FOCUS for S/390 Maintaining Databases*.

Graphics

GDDM graphics support is not available under MSO.

Submitting Batch Jobs

Submitting batch jobs via TED is supported in MSO. Enter TED to edit or view the JCL to be submitted, and issue the TED command:

`SUBMIT`

Simultaneous Usage

Simultaneous Usage (SU) is supported in MSO. The SU needs to be allocated either in the MSO startup JCL or via a DYNAM command. Once this is done, access to SU applications is exactly as described in *FOCUS for S/390 Maintaining Databases*.

Retrieving Environmental Information

The MSOINFO subroutine returns information on the environment in which it is run. The subroutine passes two parameters: the entity being requested and a return field where the results are placed.

MSOINFO is described in Chapter 15, *Site-Specific Customizations*.

SmartMode for FOCUS

SmartMode for FOCUS is fully supported with MSO. Consult the *SmartMode for FOCUS Installation and Operations Manual* for complete information.

HiperFOCUS

HiperFOCUS for MVS/ESA[®] is fully supported with MSO. Refer to the *FOCUS for S/390 MVS/TSO Installation Guide* for complete information.

CHAPTER 2

Preparing for MSO

Topics:

- MSO Issues
- OS/390 Issues

As you read the following chapters, keep the following issues in mind. You should know the answers to all of these questions before you start the installation.

Some installation steps are only required if you will be using specific MSO features, and may be omitted if the feature will not be used.

MSO Issues

- How will users access MSO?
MSO supports access from VTAM terminals, from CICS sessions, and from TSO sessions. These access methods may be used singly or in any combination.
- Will MSOKILL be used?
The MSOKILL program allows you to stop the MSO region by submitting a batch job. If MSOKILL is not used, stopping MSO requires access to either an MVS console or the MSO Console.
- Will the IBI Subsystem be needed?
The IBI Subsystem is used by MSO for CICS and TSO logons, and for the MSOKILL program. It is also used by FOCUS for Simultaneous Update capability to FOCUS databases, and for access to the IMS™ BMP.
- What VTAM applids will be used?
MSO requires a VTAM applid for access from VTAM terminals, and a second applid for the MSO Console.
- Will MSO interface with an external security system?
MSO can validate logons and data set access through an external security system such as RACF, CA-ACF2, or CA-TOP SECRET. This capability is described in Chapter 12, *MSO Security*.
- Will the MSO Console be installed?
The MSO Console allows control of MSO without requiring access to an MVS console. It is described in Chapter 13, *The MSO Console*. Even if it will be used infrequently, Information Builders recommends that you always make the MSO console available.
- Will the MSO Console interface with an external security system?
Security for the MSO Console can be controlled by records in the MSO configuration file, or by an external security system. This capability is described in Chapters 5, *The MSO Configuration File*, and 13, *The MSO Console*.
- Will JES2 Banner Support be used?
JES2 Banner Support, described in Chapter 6, *Installing MSO*, allows printouts to contain a user's userid instead of the jobid of the MSO address space. This support requires updating a JES exit.

- Will Usage Accounting be activated?

The Usage Accounting feature causes MSO to produce session-level SMF accounting data for each individual user session. It is also possible to activate Usage Accounting without actually producing SMF records. This is necessary in order to activate the MSO Resource Manager and to enable CPU and storage statistics in the MSO Console.
- What SMF record number will be used?

If Usage Accounting will be used to produce SMF records, you must select a number for MSO to use for these user records.
- Will the FastLoad facility be used?

The FastLoad facility, described in Chapter 3, *MSO Technical Description*, enables high-speed loading of MSO and FOCUS program modules, thus speeding up logon to MSO and FOCUS application startup.
- How many users will be using MSO?

This affects the REGION parameter in the JCL, and the specification of the NUMBER_READY and MAXIMUM parameters in the FOCUS service block in the configuration file.

A typical MSO region can support up to eighty users but Information Builders recommends a second MSO region if there are more than fifty users.
- Will MSO need to start a CICS transaction when users exit?

MSO allows chaining to another transaction when a user FINs from FOCUS, if desired.
- Should unique userids be enforced?

MSO can enforce the use of unique userids. Either the logon ID or the security ID may be checked for uniqueness.
- Should idle users be forced off?

MSO has the capability of forcing idle users off to conserve resources.
- What SYSOUT class should be used for the Information Builders Snapshot Facility?

The Information Builders Snapshot Facility is always active, and produces a SYSOUT file every time a task abends. This facility, and the ability to redirect the output, is described in Chapter 5, *The MSO Configuration File*.
- What SYSOUT class should be used for traces?

Traces should be directed to a held SYSOUT class, as described in Chapter 5, *The MSO Configuration File*. If desired, traces may be kept in sequential data sets or in a PDSE.

- Do any exits need to be installed?
MSO provides several exit points, which are described in Chapter 15, *Site-Specific Customizations*. You should read their descriptions to determine if any exits are needed in your environment.
- Do any zaps need to be applied?
Several optional zaps are documented in Chapter 15, *Site-Specific Customizations*. You should read the description of each zap to decide if it is applicable in your environment.
- Will MSO Load Balancing be used?
See Chapter 7, *Load Balancing*, for complete information on Load Balancing.

OS/390 Issues

- Will MSO need to run APF-authorized?
Many MSO features, including External Security, Usage Accounting, FastLoad, and the Resource Manager, require that MSO run APF-authorized. APF authorization is also required if MSO is to be brought up non-swappable.
By using MSO's Internal APF Authorization, it is possible to allocate unauthorized libraries to ddnames other than STEPLIB (USERLIB for FOCUS code; product-dependent ddnames for third-party products) while running MSO with APF authorization.
- Will MSO run non-swappable?
For performance reasons, it is recommended that MSO be run non-swappable. MSO will still be subject to normal paging operations, but will not be swapped out.
- Will user-level security be required for UDB access?
If user-level security is required, then the UDB exit DSN3SATH must be updated as described in this book.
- In which performance group will MSO run? Will it require its own performance group?
MSO is a large on-line interactive address space. It should run in an appropriate performance group, and should most definitely not be run in the same performance group as typical batch jobs.

CHAPTER 3

MSO Technical Description

Topics:

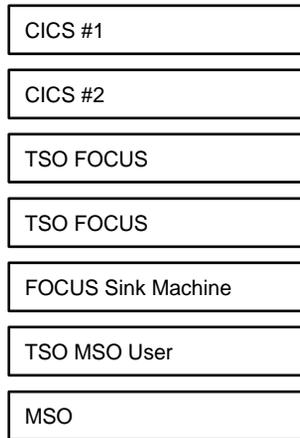
- Technical Overview
- CICS Access to MSO
- MSO Resource Manager
- The FastLoad Facility
- Internal APF Authorization

MSO (the Multi-Session Option) is a server that allows many users to run FOCUS within the same address space. This technical description of MSO assumes you are familiar with the OS/390 concepts of address spaces, tasks, enqueueing and dequeuing, and wait/post.

FOCUS was traditionally run by a TSO user from a CLIST or ISPF panel, or via a batch job using submitted JCL. In either case, the copy of FOCUS that was run required a separate address space (either TSO's or the batch job's) to execute. Use of TSO and batch also generally requires the end user to have some knowledge of OS/390, and for many applications a simpler interface was preferable.

MSO provides an alternative to these methods. It allows many FOCUS users to run in one address space. It gives users a simpler environment, allowing them to log on directly from VTAM, or to connect from a CICS region, or from TSO. A single MSO region allows users to connect from any or all of these sources simultaneously. Resources such as allocations to files may be shared as desired, while full access control is maintained using the site's normal security package.

MVS

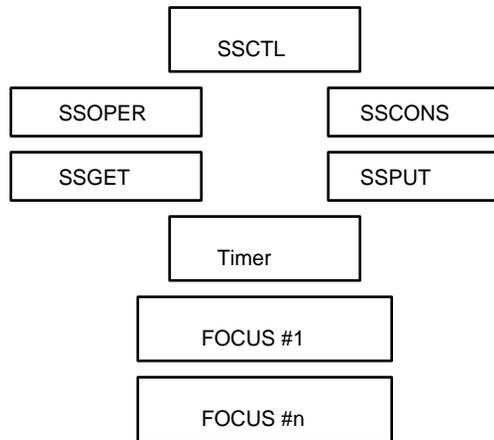


Many of the internal components of MSO are described in more detail in the sections that follow.

Technical Overview

Running SSCTL starts MSO. This program becomes the main task of the region (normally SSCTL runs as the "job-step task", in OS/390 terms). This task spawns many other OS/390 tasks, often called subtasks because they are children of the main task. The two main categories of tasks in MSO are the control tasks and the user tasks. Control tasks help MSO to perform its job of managing the users. User tasks are the ones where each FOCUS user has their own copy of FOCUS.

The following is a schematic of the MSO address space.



Control Tasks

SSCTL is the task that manages the overall execution of the MSO region. When the MSO server starts up, it reads the configuration file (FOCM_{SO}) to determine what options are requested for this execution, such as security, usage accounting (SMF records), fast loading of modules, console options, how many FOCUS tasks to start initially, the maximum number of users to allow, and so on. It also reads the name of the VTAM applid assigned, and if specified, opens the VTAM ACB to allow logons. SSCTL then attaches all of the control tasks and user tasks. When the message

`(MSO13023) ALL INITIAL SERVERS STARTED`

appears, initialization is complete, and logons can be processed.

During normal operation, SSCTL handles users logging on and off. Requests may come either via the VTAM logon exit or from the SSGET control task (for CICS and TSO). Work given to SSCTL is queued in the order received and executed sequentially. Because of this, most requests are broken into many parts, each of which can be speedily dispatched while SSCTL moves on the next work item.

The goal is to make sure SSCTL is never waiting for a unit of work to complete on behalf of a user. When the first portion of a request completes, it schedules the next part to SSCTL, and so on. Thus a logon from a CICS user to a FOCUS task may take either one or two steps: first, the logon request from SSGET is received. If a FOCUS task is already loaded and available, the user is assigned to that task, and it is woken up to begin execution for the user. If a ready task is not available, one is started. SSCTL then goes on to service someone else. When the FOCUS task started is ready to accept the logon request, it sends a queue element to SSCTL, who then completes the process, and the user's FOCUS session goes live.

VTAM users log on to MSO by first driving the VTAM logon exit established for the VTAM ACB opened when MSO came up. This logon exit operates asynchronously in the SSCTL task; it queues a message to the synchronous part of SSCTL to report that a logon request has been received. The VTAM logon request passes certain information to MSO, which determines what MSO will do next. If the terminal logging on supports the "queryable" attribute, MSO will issue a "query" to the terminal. This is simply a VTAM operation that returns additional data about a terminal and its capabilities (screen size, support for color, and so on). If user data was passed in the logon, it is analyzed for an MSO "fastpath" logon. The user data can contain the userid and password of the user logging on, thus allowing MSO to bypass the standard MSO logon screen and go straight into FOCUS. If there is no user data, then MSO next puts up its logon screen to solicit the users' userid and password.

All VTAM operations discussed so far occur as asynchronous exits running within the SSCTL task. Once a user has finished the logon process and is assigned to a FOCUS subtask, however, the FOCUS task takes over the VTAM conversation and begins to converse with the terminal using synchronous communications. SSCTL is no longer needed for the traffic of user screens being passed back and forth, since the FOCUS task is speaking directly to VTAM.

SSCTL is posted whenever any FOCUS task ends. SSCTL makes sure that all resources belonging to that user are cleaned up, such as allocations, memory, communications thread, and so on. After detaching the old FOCUS task, SSCTL may start a new task so that it will be initialized and ready when the next logon request is made.

SSOPER is another MSO control task. Its function is to process commands (MODIFY or STOP, usually abbreviated F or P) from the MVS operator's console. Even though SSOPER is a subtask of SSCTL, it runs at a higher priority than SSCTL does within the MSO region. This is done to ensure that no matter how busy SSCTL becomes, MSO will always respond to operator commands or queries.

SSCON is the control task where the MSO Console runs. This subtask accepts connections via VTAM, from multiple terminals and allows authorized users to view and/or control the operation of the MSO region. A complete description of this process is provided in Chapter 13, *The MSO Console*. SSSCON runs at a priority below SSOPER but still higher than SSCTL, again so that control and query operations will take precedence.

SSGET is the subtask responsible for receiving messages from TSO and CICS users. MSO uses an inter-address space communications protocol to send messages from TSO and CICS regions to SSGET in the MSO region. The protocol uses an OS/390 subsystem to accomplish this communication. When a user sends their first message to MSO, SSGET queues it to SSCTL as a logon request, as discussed above. Messages coming in for users already logged on are given directly to the FOCUS task that was assigned to this user by posting the ECB that the task is waiting on.

SSPUT is a companion subtask to SSGET, and is used to send data from MSO to a connected TSO or CICS user. SSPUT waits for a message from any of the FOCUS tasks, as well as from SSCTL, and sends them out to the associated user. Most messages consist primarily of the screen contents to be displayed on the users' TSO or CICS terminal. After a user ends his FOCUS session, whether by typing the FIN command or any other means, a final message is sent to let the user's address space know that the conversation with MSO has ended. This allows the MSO program running on the user side to know that it should terminate and return the user to TSO or CICS.

SSTIME is the control task that handles MRM (the MSO Resource Manager). It wakes up at a preset interval to perform analysis of the FOCUS subtasks running within the MSO address space. This feature is described in detail in the MSO Resource Manager on page 3-7. When MRM is not active, SSTIME continues to wake up periodically to prevent the region from being terminated with an S522 abend due to inactivity.

User Tasks

Each copy of FOCUS runs in a separate subtask dedicated to a particular user. The task does not start up by attaching FOCUS; instead, a program named SSFOC is attached to begin the user subtask. When FOCUS runs in MSO there is a terminal user entering screens, from which it accepts commands and writes screens out in return. SSFOC is responsible for providing the "smoke and mirrors" by which FOCUS is fooled. When FOCUS thinks it is issuing a TPUT to a TSO screen, SSFOC is the one that intercepts the request and routes it either to SSPUT (for CICS and TSO) or to VTAM.

SSFOC also handles file allocations. All FOCUS users need certain ddnames to be allocated. MASTER, FOCEXEC, FOCSORT, and OFFLINE are several examples of standard ddnames used in a FOCUS session. However, in MSO, users need their own personal allocation for some of these files. FOCSORT, for example, is a work file that two or more users cannot share; each must have their own. Since MVS shares allocations between all subtasks in an address space, SSFOC provides FOCUS with a mechanism called "ddname translation" to ensure that each task uses a unique ddname with MVS. When a particular FOCUS allocates a ddname, SSFOC first creates a translated name. The new name has the same two leading characters, a 3-digit "uniqueness" value, and a 3-digit "task number" value. Thus FOCSORT might be translated into FO001002 for one user, and FO003007 for another.

Whenever FOCUS references a ddname, the translated name is substituted instead. MSO handles the issue of shared allocations by dividing ddnames into two groups: global ddnames and local ddnames. Local ddnames are those for which SSFOC provides the translations. Global ddnames are those that are shared between all of the FOCUS tasks. The global ddname table is built by reading the MSO server startup JCL at initialization; all of the JCL-allocated files go into the global table. Additionally, if files are allocated or freed via either of the console services (SSCON or SSOPER), then that global table is updated as needed. Certain ddnames which we know should never be shared are automatically excluded from the global table, names such as OFFLINE, FOCSORT, HOLD, HOLDMAST, ADMDECK, FOCSML, FOCSTACK, SAVE and STDOUT.

Since OS/390 knows the FOCUS ddnames under different ddnames than the user does, this can sometimes be confusing when trying to reconcile what a given ddname represents. The relationship between a translated ddname and the ddname known to the user are shown in several places. One is via the MSO Console. In the DU display, use of either the WHOHAS command, or the prefix command W, brings an authorized user into a display showing the private allocations of the selected user(s), with the ddname known to the user shown on the left side of the screen, and the actual MVS ddname on the right. The other place where this relationship is easy to see is in an IBISNAP (the IBI Snapshot Facility) formatted dump printout, where the entire local ddname table is printed for the user. This is especially useful after a file-related abend such as an SB37, since the MVS-generated message has the translated ddname. Just look in the IBISNAP to see the corresponding ddname by which the user knows it.

User tasks also are responsible for other parts of the MSO system, such as establishing security and writing SMF records. Security is validated prior to allowing a user to connect to the subtask. Deletion of the security environment will occur whether the task terminates normally or abnormally. The same holds true for SMF records; the logon record is created before the user receives control, and the logoff record is created from normal or abnormal termination. A flag is set when a logon record is created; this flag ensures that the logoff record will be created later when the task terminates.

MSO frees user-allocated files in three separate steps. The first is during normal termination within FOCUS itself. Second, any ddnames remaining after normal or abnormal termination are cleaned up, which is performed while the users' security environment is still active. Finally, the SSCTL task will look for any that may have been difficult to free (such as a file which had been left open for update), and will free them after the SSFOC subtask has been detached and is totally ended. This takes place under the server's security environment.

I/O is not directly reported as there is no way to correlate an I/O to a particular subtask in any way useful to MSO. The same holds true for SRB time, which is why MSO records only TCB time. MVS keeps CPU time on a per-task basis, which MSO uses to report CPU time.

Several other components of the MSO Server are worth looking into in greater detail, and are covered in the following sections.

CICS Access to MSO

An MVS site may have one or more CICS regions active, with each CICS region handling many users. Refer to Chapter 7, *Load Balancing*, for a complete technical description.

MSO Resource Manager

The MSO Resource Manager is used by MSO to ensure that system resources are allocated appropriately among all of the tasks in the MSO address space. In addition to balancing the CPU allocated to each task, MRM allows assignment and control of the relative priorities of the FOCUS tasks in the region.

When MRM is active, the SSTIME task operates periodically (every 10 seconds or the value specified via the MRM_INTERVAL configuration keyword) and analyzes the tasks in the region. MVS's dispatcher, once it has selected the MSO address space, scans the TCB chain until it finds the first ready TCB, and gives it control. This means that tasks early in the TCB chain get more access to the CPU, and a CPU-bound (or looping) task can prevent other tasks from running. Within each MRM priority group, tasks compete with each other for CPU time. MRM looks at the tasks within a priority group, and if the top task is using a percentage of the CPU exceeding 50%, or the MRM_PERCENT value, the MRM changes the task's MVS priority to place it at the end of the group. In addition, totally inactive tasks, which consume no CPU and are in a wait, are given a very low background priority. This helps the MVS dispatcher by making the chain shorter for active users, and therefore getting them dispatched faster. When a "background" task becomes active again, it is reassigned its original priority.

There are 16 priority groups, 0 through 15, with 15 being the highest. The default priority for a user task is 7, unless it is changed as described below. Tasks in priority group 0 are suspended. They will not run until they are reassigned a higher priority.

MSO's internal tasks all run at priorities above 15. Their relative priorities, from highest to lowest are:

- SSOPER
- SSSCON
- SSCTL
- SSTIME
- SSGET/SSPUT

I/O and storage utilization are not factored into MRM decisions, which are based strictly on the amount of CPU being used.

MRM is activated when the following two conditions are met:

- MSO is running APF-authorized.
- SMFNUM is specified in the MSO configuration file.

If both of these conditions are met, MSO issues the message

(MSO13370) SERVER RESOURCE MANAGER IS ACTIVE.

Each service block specified in the MSO configuration may contain a PRTYGROUP record, which assigns a priority to all users of that service. If not specified, PRTYGROUP defaults to 7. This priority may be displayed or changed by using the DU command in the MSO console (see Chapter 13, *The MSO Console*). In addition, the priority may be changed from the MVS system console by using the MODIFY command (see Chapter 11, *MSO Operator Commands*).

The primary purpose of MRM is improving throughput by balancing and controlling resource allocation. It is expected that most MSO sites will run all tasks at the default priority, and run multiple MSO regions, at different priorities when the workload requires, giving some tasks higher priorities than others.

The FastLoad Facility

The FastLoad Facility is an MSO feature which greatly reduces the time required to load program modules into storage.

Using the FastLoad facility enables faster logons, allowing applications to start more quickly, and reducing storage fragmentation in the MSO address space.

FastLoad works by pre-fetching a copy of a module and saving it, along with relocation information, in the MSO address space. When the module needs to be loaded, FastLoad makes an in-storage copy, thus bypassing MVS program fetch, and eliminating the need to perform any I/O (other than paging I/O) to load the module. This reduction in I/O results in a substantial reduction in the elapsed time required to load a module.

FastLoad may be used with non-reentrant modules that normally load above the 16-megabyte line (i.e., they have an RMODE of ANY). Reentrant modules may also be FastLoaded. In this case, no copy is made, but the initial copy is kept in storage for all other users, thus reducing storage fragmentation. FastLoad may not be used for modules that have RMODE 24.

FastLoad is enabled by specifying the FASTLOAD keyword in the MSO configuration file (see Chapter 5, *The MSO Configuration File*). Each module to be FastLoaded requires a separate record in the file.

We recommend the use of FastLoad for modules that are over 200K and will be used by one or more MSO users. Consider FastLoading the following modules:

Module	Use
FOCUS	Main FOCUS program module.
HLI	Used by FOCMAN and HLI user-written programs.
FSROOT	Reentrant portion of the main FOCUS module.
SQLFOC	SQL translator.

Reentrant modules should not be FastLoaded if they already reside in the XLPA. FSROOT is often installed into the XLPA during FOCUS installation. You should verify if this took place before using FastLoad with F\$ROOT or other reentrant module.

Modules over 200K in size are good candidates for FastLoad. The benefits of using FastLoad on smaller modules are low, resulting in only minor savings. Never specify FOCDIR or FSASLIB because they will not work properly if FastLoaded.

When using FastLoad, preloading copies of FOCUS by specifying a high NUMBER_READY in the configuration file (see Chapter 5, *The MSO Configuration File*) is not particularly necessary, since FastLoad can dynamically initialize a copy of FOCUS relatively quickly. However, by moving this work from a user's logon to the MSO region's startup, there is still some benefit available from specifying a NUMBER_READY equal to the anticipated user count. Both capabilities can be used together: copies of FOCUS loaded by NUMBER_READY do take advantage of FastLoad.

In addition, using the MVS LLA and VLF facilities on FOCLIB.LOAD is not as beneficial as it would be without the FastLoad Facility, although it will still result in a reduction of I/O to FOCLIB. This reduction will be more pronounced when multiple MSO address spaces are used, or when there is batch or TSO-based usage of FOCUS.

Internal APF Authorization

Internal APF authorization allows MSO to run with APF authorization without requiring that every load library allocated to the MSO address space also be APF authorized.

Normally, when an address space is running with APF authorization, all load libraries allocated to the address space must be APF-authorized libraries. With Internal APF Authorization, only the libraries allocated to ddname STEPLIB must be APF authorized. Unauthorized libraries may be allocated to ddname USERLIB or to product-specific ddnames, as required. Both Information Builders products and third-party products may take advantage of this feature.

This feature works by turning off the APF-authorization bit after MSO starts up. MSO retains the ability to make authorized requests to the operating system when it needs to (for security validation, writing SMF records, etc.), but prevents users from doing so. The APF bit remains off even when MSO makes these requests, so that integrity is ensured without forcing other tasks to be in a "status stop" state. Thus multi-tasking continues even when this feature is active.

When Internal APF Authorization is active, ddnames USERLIB and FOCLIB may not be allocated with the DYNAM command; if used, they must be allocated in the MSO JCL.

When using Internal APF Authorization, ensure that the data set FOCLIB.LOAD is only allocated to STEPLIB, and not to USERLIB and/or FOCLIB. In general, libraries should only be allocated to either STEPLIB or USERLIB; allocating libraries to both is not supported.

Note:

Any third party software package that issues MODESET and TESTAUTH macros may not be compatible with APFAUTH=INTERNAL. In order to avoid incompatibility with third party software, use APFAUTH=EXTERNAL (default) in conjunction with APF authorized load libraries allocated to ddnames STEPLIB and USERLIB. MSO then uses MVS standard APF authorization techniques and provides accurate information to third party software packages.

Internal APF Authorization is activated by use of the APFAUTH keyword in the MSO configuration file, as described in Chapter 5, *The MSO Configuration File*.

CHAPTER 4

MSO Capacity Planning and Performance

Topics:

- Virtual Memory Requirements
- Region Size
- Real Memory Requirements
- MVS Performance Group Recommendations
- Data Set Allocation Limitations

The FOCUS Multi-Session Option is a software platform that allows multiple FOCUS users to run in a single MVS address space. MSO communications services allow different terminal environments to talk to FOCUS sessions in MSO; terminals in the CICS, TSO, or VTAM environments are supported.

The basic FOCUS programs in the OS/390 environment are identical for both MSO and TSO FOCUS. Because the same programs are used when either an MSO or a TSO user accesses FOCUS, the overall resources consumed by FOCUS functions are essentially the same for both. In fact, for most FOCUS users, operating under MSO is indistinguishable from operating under TSO. However, since OS/390 treats swappable TSO FOCUS users differently than non-swappable MSO FOCUS users, it is possible for TSO and MSO response time characteristics for the same FOCUS application to vary greatly. Generally, the choice depends on resources available to the FOCUS application, as well as the site's tuning parameters.

Each FOCUS user requires a substantial user data area that cannot be shared. These data areas are the main factors in determining how to plan for resource requirements for FOCUS in the MSO environment. CPU and I/O requirements depend on the particular FOCUS applications being used, and are not dependent on the FOCUS environment (either MSO or TSO). This means that only real and virtual memory requirements must be taken into account for MSO capacity planning.

Virtual Memory Requirements

Each MSO user requires a certain amount of virtual memory within the MSO address space. OS/390 splits virtual memory into two main types: "above the line" and "below the line." The "line" being referred to is the 16-megabyte line, which divides 24-bit-addressable storage from 31-bit-addressable storage. Above-the-line virtual memory is where most of the memory for each user resides. Certain OS/390 functions and data areas, however, must reside below the line. Because OS/390 restricts below-the-line memory more severely than above-the-line memory, it is below-the-line memory that most impacts MSO capacity planning.

Below-the-line Memory Requirements

Each MSO user requires some memory residing below the 16-megabyte line. Certain operating system control blocks (such as TCBs) and I/O access method buffers must reside below the line. Usually, this below-the-line memory must fit in the user private area of virtual address space; the size varies from site to site but is usually from 6 to 10 megabytes. Thus, all users in a single MSO address space must share a site-limited below-the-line user private area that may only be 6 megabytes in size.

Most below-the-line memory requirements are due to MVS access methods such as BSAM, BPAM, QSAM, and BDAM. These access methods require one to five blocks of memory below the line while the file is open. Suppose a FOCUS MODIFY FIXFORM command opening a QSAM file, requires 5 buffers below the line for an MVS GETMAIN. If the blocksize of the input file is 12K, then this file will require 60K of below-the-line private user area while the MODIFY is executing.

In addition to MVS access method buffers, FOCUS requires that each MSO user have a non-shareable load module called XAIO resident below the line. Currently XAIO is 20K in size. Most of the space required by XAIO is for data areas that cannot be shared.

Finally, there are some miscellaneous MVS system-oriented data areas that must be resident for each MSO user. These data areas total about 16K per user.

It is important to remember that only open files require buffer space below the line, and that these buffers are freed as soon as the user closes the file. Thus, in the FIXFORM example above, as soon as the MODIFY command finishes its processing, the 60K needed by QSAM is freed, and the memory can then be used for another MSO user.

As an example, if each MSO user requires the following open files, we can estimate the total user private area requirements:

Use	Blocksize	Number of Blocks	Total
MASTER	4K	1	4K
FOCEXEC	4K	1	4K
FOCFILE	4K	1	4K
INFILE	12K	5	60K
XAIO	25K		25K
Miscellaneous	16K		16K
TOTAL			113K

If the site has a 9-megabyte below-the-line user private area, then we can potentially fit 9216/108 or 85 users in this MSO address space.

As another example, if we design our application to use FOCUS SU and we make the blocksizes smaller, we can get the following:

Use	Blocksize	Number of Blocks	Total
MASTER	1K	1	1K
FOCEXEC	2K	1	2K
XAIO	25K		25K
Miscellaneous	16K		16K
Total			44K

If the site has a 9-megabyte below-the-line user private area, then we can potentially fit 9216/39 or 236 users in the MSO address space.

By using the MSO Console, you can see at a glance how much memory is being used by each task in an MSO address space. The DU display includes columns for below-the-line and above-the-line virtual storage, and makes it possible to easily project the capacity of the MSO address space, in terms of virtual storage restrictions.

Above-the-line Memory Requirements

While below-the-line memory in an MVS address space is severely restricted, above-the-line memory is essentially unbounded. Potentially in an OS/390 address space, the user private area can be over 2000 megabytes in size. Most of the FOCUS programs and data areas are above the line. Each FOCUS MSO user will require roughly 4 megabytes of virtual memory above the line, which means that over 500 users can potentially fit into above-the-line memory in a single address space.

In practice, MVS limits the amount of virtual memory that a single address space can access. In early versions of MVS/XA, the site codes an IBM exit called IEFUSI to change the default above-the-line virtual memory limit for an address space, which is 32 megabytes. This means that at 4MB per user, only about 8 users can fit into the default limits. This 32-megabyte default, however, can be changed to any value desired by the site. To support 50 users at 4MB per user, the IEFUSI limit must be changed to 200MB. The configuration file parameter MAXIMUM should be set to this number (50).

In MVS/ESA or later versions of XA, the job's REGION parameter can be directly set to any limit desired for the MSO job. To support 50 MSO users, one would simply specify REGION=200M.

The 4MB virtual memory requirement per user used above can vary, depending on the FOCUS applications used. For example, large MODIFY programs may require 5MB of memory, the FOCUS cache facility can dramatically increase memory requirements for each user, and FOCUS database interfaces may require additional memory. Once again, the MSO Console DU display can be used to determine the actual amount of above-the-line memory being used by each user.

Region Size

You should specify a region size for each user and MSO will help you determine when it is necessary to increase a user's storage needs.

Specifying Region Size

As a starting point, you should specify a region size of 4 MB per logged in user. You should add 4MB for region overhead, and an additional 4MB if the FastLoad Facility is active. Depending on the FOCUS applications in use, you may need to increase this later.

For example, if you have one service block in the MSO configuration file, with `MAXIMUM` set to 20 and the FastLoad facility activated, you should specify a region size of $(20+2)*4$, or `REGION=88M`.

On some MVS systems, the region size request may be overridden by the IEFUSI exit in MVS. You can determine whether this has happened by examining the message MSO13065, which MSO issues at startup. This message displays the amount of storage actually assigned to the address space. If the system has overridden the requested region size, you will need to change IEFUSI to allow MSO to obtain adequate virtual storage.

You should not specify a region size of less than 16MB, since MVS may interpret this as a request for below-the-line storage instead of above the line. In such a case, the total region size will be a site-dependent default.

Monitoring Region Requirements

MSO issues messages MSO13145 and MSO13144 when a user logon is blocked due to insufficient free storage in the address space. Use these messages to determine when storage needs to be increased. You can increase above-the-line storage by requesting a larger region in the MSO JCL. Increasing below-the-line storage requires either application tuning or splitting the MSO address space into multiple regions.

You can also use the Display Users (DU) display in the MSO Console to monitor actual storage allocations for individual MSO users. With this display, you can project the capacity of the region, assuming the users monitored are representative of your MSO workload.

Keep the following points in mind when viewing the storage allocated to the region, as opposed to individual MSO users, in the MSO Console's DU display:

- FOCUS modules are preloaded (based on the `NUMBER_READY` parameter in the configuration file) and reloaded (after a user logs off, to speed up the next logon). The storage for these modules, and their data areas, is displayed as part of the region's storage until a user logs on.
- Some task-level storage cannot be accounted for accurately, and is displayed only as part of the region's storage.
- FOCUS and MSO contain many reentrant modules which have their storage allocated to the MSO region. Since the modules are reentrant, their storage doesn't increase as the number of users increases.

Real Memory Requirements

While virtual memory limits do exist within MSO, they do not imply any kind of performance characteristic. With virtual memory, the site simply needs to ensure that adequate paging space is available on disk to hold the size of the MSO address space. It is the real memory requirements that dictate how much physical memory will be needed to support a given number of MSO users. Thus, even though MSO may require as much as 4MB of virtual memory per user, the entire 4MB need not be real in the sense that a site must purchase this much main memory per user. The actual real memory requirements are very different from the virtual memory requirements.

The real memory requirement for MSO FOCUS use is essentially identical to the real memory requirement for TSO FOCUS use, and is roughly 1 to 1.5 MB per active user. An active user is not strictly defined, but can be loosely described as any user who has pressed the enter key on the terminal in the past five minutes or so.

Thus, to calculate how much real memory is required to support a given number of users, it is important to know what percentage of them will be active. If only two users are active out of a logged on population of 50, then only 2-3 MB of real memory will be required. On the other hand, if all 50 users are equally active, then 50 to 75 MB of real memory will be required to support them.

Using Expanded Storage

It is important to understand that real memory requirements for MSO can be fulfilled using either central or expanded storage. This means that the 1-1.5 MB can essentially all be expanded storage, which is much less expensive than central storage.

A certain amount of central storage will, of course, be needed to support individual transactions. A very rough rule of thumb would be 100K per active user, except for users running very long running jobs, which require the full 1-1.5 MB of central storage.

A Typical MSO Region

Experience has shown that typical FOCUS sites have roughly the same proportion of transaction types: 50% of users are using pre-written FOCUS programs as on-line transactions, 40% are doing essentially FOCUS development (such as writing a new report) and 10% may be executing long running tasks (such as large reports). In this type of site, perhaps one out of three logged on users is active. This means that to support 50 logged on users, 17 to 25 MB of expanded storage is required, along with 2 to 3 MB of main memory. For each long running job, we would add an additional 1 MB of main memory.

Another fairly common MSO FOCUS scenario would be a small number of data entry operators executing a transaction-oriented application. Here, all users are always active and thus each require 1-1.5 MB of expanded storage. To handle 15 such users in an MSO address space would require about 15-22 MB of expanded storage and 2 to 3 megabytes of central storage. Eighty users is a typical maximum number of users per address space.

Preventing Thrashing

If too many MSO users are competing for an inadequate amount of central or expanded storage, thrashing will result. When this happens, MVS tries to page in and out memory pages from auxiliary storage (DASD). If there is an inadequate number of paging actuators available to the system, the entire system can stall (except for those address spaces that are storage isolated). Since, unlike TSO FOCUS, MVS cannot distinguish between users in MSO, it is important not to allow too many MSO users to log onto the system when there is inadequate memory to support them. This is done via the MAXIMUM configuration file parameter.

It can be difficult to know under MVS how much expanded storage is actually available for use, because all the MVS address spaces share expanded storage. Thus, when planning for a certain number of active MSO users, the safest bet is to plan to add the required expanded storage if it appears to be needed. If paging rates dramatically increase while adding a few active MSO users, the extra-expanded storage is probably required.

MVS Performance Group Recommendations

A site implementing MSO may desire to create separate performance groups for MSO address spaces. Because MSO is designed to run non-swappable, the regular parameters needed for SRM swapping recommendations are not required. However, the dispatching priority may be changed depending on the type of work to be executed in an MSO region. A high priority MSO region using efficient, pre-written routines would be assigned a favorable dispatch priority, while an MSO region for users reporting from large databases might get a low dispatch priority.

In general, MSO should have dispatch priorities just below those for production CICS and IMS regions, as well as TSO period 1. MSO should get a higher dispatch priority than test CICS, IMS, and batch regions, as well as lower period TSO regions.

Using storage isolation for very high priority applications, such as intensive data entry applications, can be appropriate. In such applications, assigning a minimum working set size of 1.5MB per user will ensure that MSO region always has adequate main memory to provide good response time for users. For a three-user region, specifying

```
PGN=... , (... , PWSS=(1000,1000))
```

ensures that this small MSO address space will always have about 4MB of main storage dedicated to it.

Data Set Allocation Limitations

Both OS/390 and MSO place restrictions on ddnames.

OS/390 DDNAME Limitations: Task Input/Output Table (TIOT) Restrictions

OS/390 restricts the storage allocated to the Task Input/Output Table (TIOT), which contains data associated with ddnames, to 32K bytes. As each TIOT entry is 20 bytes, there is a limit of about 1600 total allocations possible in an MSO address space. It is possible to increase the size of the TIOT to a maximum of 64K to give a total of about 3200 total allocations. If an MSO address space is anticipated to require this many allocations, the system programmer maintaining MVS should change the default TIOT size to 64K.

MSO DDNAME Limitations

In addition to OS/390 restrictions on the number of ddnames that may be allocated to an address space, MSO imposes restrictions on the number of ddnames that may be allocated concurrently.

Up to 1024 ddnames may be globally allocated to the MSO address space. Global allocations are those made in the MSO JCL or allocated with the DYNAM command from the MSO Console.

Up to 256 ddnames may be allocated to each MSO user. These local allocations are made using the DYNAM command in the user's session.

To see the impact of Load Balancing on DDNAME limitations see Chapter 7, *Load Balancing*.

CHAPTER 5

The MSO Configuration File

Topics:

- Global Configuration
- Connectivity Options
- MSO Console Security
- MSO Diagnostic Facilities
- The Service Block
- Obsolete Configuration File Commands

The MSO configuration file contains start-up parameters processed when MSO is initialized. It is allocated to ddname FOCMSO in the MSO JCL.

This chapter contains descriptions of configuration file records for:

- Global processing and options.
- Connectivity options.
- Console security.
- Diagnostic facilities.
- Subtask characteristics.
- Obsolete records.

Any line in the configuration file starting with an asterisk (*) is treated as a comment.

All data in the configuration file must be between columns 1 and 72; anything after column 72 is ignored.

For commands that accept wildcards, "*" and "?" may be used. "*" will match zero or more characters, and "?" will match exactly one character.

Global Configuration

```
EXTSEC = {YES|NO}
```

where:

YES

Specifies the use of an external security system for validating access to MSO and to MVS data sets and resources. User Exit MSIDVER can effectively bypass External Security Verification. Each user receives a separate security profile. The MSO library FOCLIB.LOAD must be APF-authorized.

NO

Bypasses User Exit MSIDVER.

All users inherit the security profile of the MSO region itself. See Chapter 12, *MSO Security*, for further information. This command is mandatory in the configuration file; it does not default.

APF authorization is required.

Default: None.

EXTSEC must be specified.

`SMFNUM = smfnumber`

where:

smfnumber

Is a site-selected number for a user SMF record, from 128 to 255.

When SMFNUM=0 is specified, it has the effect of turning on internal usage accounting without writing SMF records; this would be used to enable utilization statistics to be displayed in the MSO Console and from the MVS console. The MSO Resource Manager also requires this specification. See Chapter 10, *MSO Usage Accounting*, for more information.

MSO must be APF-authorized if SMFNUM is specified. We recommend specifying SMFNUM=0 if you don't want to write SMF records, due to the other MSO features which require it.

APF authorization is required.

Default: 0, when running with APF authorization; otherwise disabled.

`UNIQUE = {logid|secid}`

where:

logid

Is a login ID.

secid

Is a security ID.

If this entry is omitted, MSO will allow any number of users to log on with the same LOGID or SECID.

APF authorization is not required.

Default: None

`STORAGEBELOW = n`

where:

n

Is the amount of virtual storage in kilobytes that must be available in the MVS below-the-line private area before a new user is allowed to log on. If omitted, the default is 200 (that is, 200K). This value may be specified as described in The Service Block, on page 5-12 in which case it overrides the globally specified value. If the number you specify is beyond your maximum memory you will be unable to start your service.

APF authorization is not required.

Default: 200

`STORAGEABOVE = n`

where:

n

Is the amount of virtual storage in kilobytes that must be available in the MVS above-the-line private area before a new user is allowed to log on. If omitted, the default is 4096 (that is, 4M). This value may be specified as described in The Service Block, on page 5-12, in which case it overrides the globally specified value.

APF authorization is not required.

Default: 4096

`MAXSERVERS = n`

where:

n

Is the maximum number of servers and FOCUS users that can be present on MSO at a time. MAXSERVERS may be specified to allow for adding additional FOCUS tasks, and should be set to the total number of subtasks that will be active in the address space.

APF authorization is not required.

Default: None

`IDLELIM = nnnnnnn`

where:

nnnnnnn

Is the number of seconds of idle time that are allowed before the MSO address space is shut down. All FOCUS users and all server tasks must be idle in order for this to take place.

IDLELIM may also be specified in the service block in which case it applies to a logged-on user, rather than the entire MSO address space. The maximum value that may be specified is 9999999.

APF authorization is not required.

Default: None

`RACSP = nnn`

where:

nnn

Allows you to select the MVS storage subpool used when SAF creates its security control blocks. In some circumstances, CA-ACF2 sites will experience better results by using subpool 255.

APF authorization is required.

Default: 230

`FASTLOAD = load module`

where:

load module

Names a reentrant or non-reentrant load module to be preloaded by the FastLoad Facility. This module must have an RMODE of ANY; FastLoad may not be used for modules that have RMODE 24. If the module specified is reentrant, then no copy is made, but the initial copy is kept in storage for all MSO users. Using FastLoad on a reentrant module causes the module to be loaded at MSO initialization time rather than at the time the module is first referenced by an MSO user; this has the effect of reducing storage fragmentation by preventing the use count from ever going back down to zero.

The FastLoad Facility should not be used on modules smaller than 200K or modules installed in the XLPA. See Chapter3, *MSO Technical Description*, for a discussion of FastLoad Facility processing.

Use a separate record for each module to be preloaded.

APF authorization is required.

Default: None

`APFAUTH = {EXTERNAL|INTERNAL}`

where:

`EXTERNAL`

Allows you to allocate non-APF-authorized libraries (other than STEPLIB) to the MSO address space, or to individual MSO users, even when MSO itself is APF-authorized.

`INTERNAL`

Activates Internal APF Authorization. This feature is described in Chapter 3, *MSO Technical Description*.

APF authorization is required.

Default: EXTERNAL

`SZERO = {YES|NO}`

where:

`YES`

Causes subpool 0 to be shared across tasks. It is required when VSAM files are allocated in the MSO JCL.

`NO`

Is used when VSAM files are not accessed, or when they are allocated to individual users with the DYNAM command.

APF authorization is not required.

Default: NO

MRM = {ON|OFF}

where:

ON

Turns the MSO Resource Manager on.

OFF

Turns the MSO Resource manager off.

MRM should normally be allowed to default; it will be activated whenever MSO is running with APF authorization and the SMFNUM has been specified or allowed to default. You should only deactivate MRM if a problem is suspected with the resource manager.

APF authorization is required.

Default: ON, if running with APF authorization and SMFNUM has been specified or allowed to default to 0.

FASTPDS = {ON|OFF}

where:

ON

Activates FASTPDS, causing MSO to read all partitioned data sets in the server's startup JCL and copy their directory information into memory. These tables of members are used for all future references to those libraries. The I/O to find the members is only done once, at initialization, and this results in faster access to data in the globally allocated libraries. When activated, these global files become read-only, and may not be updated by MSO users.

OFF

Deactivates FASTPDS.

APF authorization is not required.

Default: OFF.

MRM_PERCENT = *nn*

where:

nn

Is the threshold percentage of CPU that will cause a user to be reduced in priority by MRM. This number can range from 1 to 99.

APF authorization is required.

Default: 50

MRM_INTERVAL = *nnnn*

where:

nnnn

Is the second interval in which MRM will monitor all users to prevent any users from exceeding the MRM_PERCENT value. This number can range from 1 to 1440.

APF authorization is required.

Default: 10

SOS_PERCENT = nn

where:

nn

Is an integer representing a percentage between 0 and 99 percent.

Short on Storage specifies the percentage of storage utilization, which when exceeded, will cause a warning message to be issued via non-scrollable messages to WTO and MSOPRINT. The message destination can be altered with the Enhanced Message Routing Facility. If storage utilization falls below this value, a message is issued stating that the storage constraint has been relieved and the WTO message is deleted (unless suppressed by the Enhanced Message Routing Facility). The SOS_PERCENT check, if active, is made at every MRM_INTERVAL interval (MSO Resource Manager Interval).

Specifying 0 disables the SOS monitoring facility. SOS monitoring may be enabled or disabled dynamically by setting SOS_PERCENT to a non-0 or 0 value, respectively.

APF authorization is required.

Default: 0

Connectivity Options

APF authorization is not required for the connectivity options.

LU2_NAME = applid

where:

applid

Tells MSO which VTAM applid to use. Users logging onto MSO use an applid to log onto MSO from a VTAM screen.

Default: None

`TRANSID = CICStransactionID`

where:

`CICStransactionID`

Is a CICS-only parameter that specifies the CICS transaction that is to be executed when an MSO/CICS user returns to CICS. If TRANSID is not specified, MSO clears the screen when done.

Default: None, unless CMSOTABL is reassembled.

`CICSBREAK = {NONE|PA n |PF nn }`

where:

`NONE`

Means that there is no CICS Session Termination key.

`PA n`

Sets any PA key (from 1 to 3) as a CICS Termination key.

`PF nn`

Sets any PF key (from 1 to 24) as a CICS Termination key.

This is a CICS-only parameter. When a CICS Termination key is pressed, the user's MSO session is terminated and all the user's resources freed. Care should be taken that CICSBREAK does not specify the same key used for CICS screen printing, and that it does not specify a key that will be used in any FOCUS applications that might be run.

CICSBREAK replaces the ATTNKEY parameter from Release 6.8 and earlier.

It is recommended that this feature not be used. The FIN command should be used to terminate an MSO/FOCUS session.

Default: NONE

MSO Console Security

The descriptions of configuration file records for MSO Console security assume that standard command authority assignments are in effect. These may be changed by customizing SSSCONSEC, the MSO Console command security table. This customization allows commands to be moved from one authorization class to another. SSSCONSEC is discussed in Chapter 13, *The MSO Console*.

`CONSEC = {INTERNAL|EXTERNAL}`

where:

`INTERNAL`

Allows you to define security rules using the CONSOPER, CONSDTL and CONSUSER configuration statements.

EXTERNAL

Allows the MSO Console to use external SAF-based security. EXTERNAL may only be specified when EXTSEC = YES is specified.

APF authorization is required for EXTERNAL only.

Default: INTERNAL

CONSOPER = *userid1,userid2...*

where:

userid1,userid2...

Specifies the list of userids that may issue commands from the MSO Console which alter the behavior of MSO or MSO users. This is applicable only if CONSEC is set to INTERNAL. Wildcards using '*' and '?' are permitted. Multiple records may be present if the userid list does not fit on a single line. Blank spaces are not allowed in the list of userids.

APF authorization is not required.

Default: None

CONSDTL = *userid1,userid2...*

where:

userid1,userid2...

Specifies the list of userids that may see detailed information from other MSO user's sessions (files allocated, screen contents, current statement being executed, etc.). This is applicable only if CONSEC is set to INTERNAL. Wildcards using '*' and '?' are permitted. Multiple records may be present if the userid list does not fit on a single line. Blank spaces are not allowed in the list of userids.

APF authorization is not required.

Default: None

CONSUSER = *userid1,userid2...*

where:

userid1,userid2...

Specifies the list of userids that are permitted to logon to MSO Console. If a userid appeared in the CONSOPER or CONSDTL statement it is already authorized to logon to Console and need not be specified here. This is applicable only if CONSEC is set to INTERNAL. Wildcards using '*' and '?' are permitted. Multiple records may be present if the userid list does not fit on a single line. Blank spaces are not allowed in the list of userids.

APF authorization is not required.

Default: None

`CONSCCLASS = classname`

where:

classname

Specifies the name of the resource class that is specified to SAF for authorization checking. Users must have READ access to this resource class and the appropriate entity name in order to access MSO Console. This is applicable only if CONSEC is set to EXTERNAL.

APF authorization is required.

Default: FACILITY

`CONSENTL = entity-name`

where:

entity-name

Specifies the entity name that will be specified to SAF, in conjunction with the resource class specified in CONCLASS, in order to validate logon access to MSO Console. This is applicable only if CONSEC is set to EXTERNAL.

APF authorization is required.

Default: IBI.CONSOLE.LOGON

`CONSENTO = entity-name`

where:

entity-name

Specifies the entity name that will be specified to SAF, in conjunction with the resource class specified in CONCLASS, in order to validate Operator-style command authority (see CONSOPER, above) in MSO Console. This is applicable only if CONSEC is set to EXTERNAL.

APF authorization is required.

Default: IBI.CONSOLE.OPERATOR

`CONSENTD = entity-name`

where:

entity-name

Specifies the entity name which will be specified to SAF, in conjunction with the resource class specified in CONCLASS, in order to validate authority to see details on other user's sessions (see CONSDTL, above) in MSO Console. This is applicable only if CONSEC is set to EXTERNAL.

APF authorization is required.

Default: IBI.CONSOLE.DETAIL

MSO Diagnostic Facilities

None of the diagnostic facilities require APF authorization.

`IBISNAP = class`

where:

`class`

Specifies the SYSOUT class to use for dumps produced by the Snapshot Facility. This can be overridden by allocating ddname IBISNAP in the MSO JCL.

Default: A

`SNAPEXCLUDE = cccc,rrrr`

where:

`cccc`

Is the abend completion code.

`rrrr`

Is the reason code.

For example:

In FOCMSO:

`SNAPEXCLUDE = S913,U1010*`

The S is required in the abend code and the U is required in the reason code. Trailing asterisks are permitted. They denote that any character may follow. To specify multiple abend codes it is necessary to use multiple SNAPEXCLUDE cards.

A new message

`(MSO13277) >>>> IBISNAP WILL BE SUPPRESSED FOR ABEND S913,* <<<<`

displays in the JES Message Log to list all currently active SNAPECLUDEs.

Default: None

`TRACECLASS = class`

where:

`class`

Specifies the SYSOUT class to use for traces produced by FOCUS. When traces are directed to SYSOUT, they may be printed, or they may be browsed with a tool such as SDSF. Information Builders recommends using a held SYSOUT class for traces.

Default: A

`TRACELIB = dsname`

where:

dsname

Specifies the library in which to store traces produced by FOCUS. This file must be a PDSE. Specifying TRACELIB causes TRACECLASS to be ignored.

Default: None

`TRACEPREF = specification`

where:

specification

Specifies sequential files to which traces produced by FOCUS will be written. The specification may include the special names "&USER" and "&TRACE," which will be replaced with the userid and trace ID, respectively. For example, the following specifications might be used:

```
TRACEPREF = &USER . .FOCTRACE .&TRACE
```

```
TRACEPREF = MSO.TRACE.&TRACE
```

Specification of &USER and &TRACE generates unique data set names, and will thus prevent traces from overwriting older traces. Specifying TRACEPREF causes TRACECLASS and TRACELIB to be ignored.

Default: None

The Service Block

The service block replaces the MAXFOC and NUMFOC specifications, and defines the characteristics of the subtasks that MSO is to run. Multiple service blocks may be specified. The parameters in the service block are positional, and only apply to the service block in which they are placed. All service blocks must be placed after all other entries in the configuration file.

The service specified in the first service block will become the default service for VTAM users and will be shown on the VTAM logon screen (see Figure 4). If multiple service blocks are specified, you should ensure the desired default service is specified first. The maximum number of service blocks that may be specified is 20.

Note:

CICS- and TSO-connected users may only access SERVICE=FOCUS. Only VTAM-connected users have access to other services. APF authorization is not required in the service block.

`SERVICE = name`

where:

name

Specifies FOCUS as the service name. The service named in the first service block in the configuration file will be the default service on the VTAM logon screen.

Default: None

`PROGRAM = name`

where:

name

Specifies FOCUS as the program name.

`NUMBER_READY = nnn`

where:

nnn

Is the number of initial FOCUS copies loaded into the MSO address space. These FOCUS copies are the non-sharable part of FOCUS. All MSO users share one copy of the sharable part of FOCUS. When installing MSO, it is convenient to make `NUMBER_READY = 1`, because the higher this number is, the longer it takes MSO to initialize. `NUMBER_READY` can later be increased to a larger number. This value is equivalent to the old `NUMFOC` parameter.

Default: 1

`MAXIMUM = nnn`

where:

nnn

Is the maximum number of users allowed to log onto MSO. This value is equivalent to the old `MAXFOC` parameter.

Default: 1

`TIMEOUT = nnn`

where:

nnn

Specifies a maximum time, in seconds, a FOCUS task may be in a terminal wait. If the limit is exceeded, the task is ended and its resources are freed.

Note:

If a user is waiting for FOCUS to reply, the `TIMEOUT` clock is not running.

Thus, to cause a user to timeout after 20 minutes, one would code

`TIMEOUT = 1200`

The maximum value that may be specified is 9999999. The default is to disable the timeout facility.

Default: None

IDLELIM = *nnn*

where:

nnn

Specifies a maximum time, in seconds that a user's task may be in a wait. This includes waiting on all external events, not just terminal wait (in contrast to TIMEOUT above). If the limit is exceeded, the task is ended and its resources are freed. This is distinct from the global IDLELIM setting (see *Global Configuration* on page 5-2), which applies to the entire MSO region. Whichever IDLELIM is smaller will apply to users of this service. The maximum value that may be specified is 9999999.

Default: None

PRTYGROUP = *n*

where:

n

Specifies the default MRM priority for all users of this service, from 1 to 15.

Default: 7

STORAGEBELOW = *n*

where:

n

Specifies STORAGEBELOW for this service only, overriding the default or globally-specified value. See *Global Configuration* on page 5-2 for a description of the global STORAGEBELOW parameter.

Default: Use the global STORAGEBELOW value.

STORAGEABOVE = *n*

where:

n

Specifies STORAGEABOVE for this service only, overriding the default or globally-specified value. See *Global Configuration* on page 5-2 for a description of the global STORAGEABOVE parameter.

Default: Use the global STORAGEABOVE value.

`LOGON_TIMEOUT = nnn`

where:

nnn

Is a positive number between 30 and 100000, representing the number of seconds the MSO VTAM Logon screen is displayed before being canceled.

It allows you to set a maximum number of seconds that the MSO Logon screen is displayed. When that period expires the connection is broken and the user is returned to the VTAM Logon screen. This frees up the VTAM session.

MSO Load Balancing Configuration Parameters

MSO Load Balancing provides the capability of grouping multiple MSO regions into a logical entity. Dividing the MSO Load Balancing group into subgroups is also possible. This mapping of a MSO Load Balancing group of MVS regions and routing of users into one or more regions within this group is done through the specification of three parameters.

`APPLGROUP = MSO_group_name`

where:

MSO_group_name

This specification activates the Load Balancing function. It appears once in the global section of the MSO Configuration File. All MSO regions that have an identical *MSO_group_name* are grouped together and become selection candidates for MSO users. A user enters an MSO group by establishing an initial connection to a MSO region that has APPLGROUP specified. This results in the IBI Subsystem scanning the current MSO group population of MSO regions to properly place the Logon request of the user unless the SERVICE(...) parameter was specified at Logon.

`SERVICE=name`

where:

name

Specifies a service block name in a Logon request. It is mutually exclusive with specifying APPLNAME.

This specification is not new with MSO Load Balancing. The functional characteristics of using the Logon parameter SERVICE(...) change when Load Balancing is active for the MSO region. The SERVICE(...) Logon request will not be able to locate the service block of the same name unless APPLNAME=NONE has been specified in that service block.

The specified service block name will only be used if it has PROGRAM=FOCUS specified.

When APPLNAME is not specified for the service block, the service name becomes the default application name for that service block.

```
APPLNAME=name1
APPLNAME=name2
.....
APPLNAME=name8 /* max of 8 appl names per service block */
APPLNAME=NONE
```

This is a Load Balancing keyword. It is specified within a service block in the MSO Configuration File. It identifies the service block as being a valid Load Balancing Logon destination for those Logon requests that specify an APPLNAME. Specifying APPLNAME is mutually exclusive with “service” in the Logon request.

When an application name is specified in the Logon request, a subset of MSO regions within the Load Balancing group is determined on the basis of which ones have a service block with the name specified as an applname.

The application name of NONE is reserved. If this is specified, the service block is not eligible for Load Balancing and is ignored for routing purposes. This is also the only way to be able to connect to a service block in a Load Balancing region by SERVICE(...) in the Logon request.

NOTE:

If APPLNAME is not specified for a service block that is in an MSO Load Balancing region, a default application name is assumed for the service block using the service block name.

Obsolete Configuration File Commands

The following parameters are obsolete and should not be specified in the configuration file. Their use is not supported. In particular, specification of RUSIZE or BANNER may cause unpredictable results.

```
NUMFOC = n
MAXFOC = n
MSOLOG = {ON|OFF}
SVC = {YES|NO}
SERVINIT = name
RUNCOUNT = n
BANNER = AUTO
RUSIZE = n
```

CHAPTER 6

Installing MSO

Topics:

- Install FOCUS
- System Programmer Tasks
- File Set Up
- Basic MSO Customization
- Installing MSO Components
- Install Exits and Zaps, as Required

These installation instructions assume you have available FOCUS documentation.

There are three ways to access MSO: from CICS terminals, from TSO sessions, and directly from VTAM. Depending upon which of the access methods that are to be installed, not all installation steps need to be performed.

The installation steps include:

1. FOCUS installation
2. System programmer tasks
3. File set up
4. Basic MSO customization: configuration file and JCL
5. MSO subsystems set up
6. Database data adapters set up
7. Advanced MSO customization: exits and zaps

The following table summarizes the data sets and files that are used by MSO:

DDName	DSName	Purpose
	MSO JCL	JCL for the MSO address space.
	CICS JCL	JCL for the CICS address space.
	FOCCTL.DATA	Contains FOCUS sample JCL, zaps, assembly and link-edit jobs.
	MSO.DATA	Contains MSO sample JCL, zaps, assembly and link-edit jobs, and source code for modules that are locally customizable.
	MSO.LOAD	Contains the MSO CICS programs.
STEPLIB	FOCLIB.LOAD Other load libraries	Contains FOCUS and MSO product code. Contain other required product code.
USERLIB	FUSELIB.LOAD Other load libraries	Contains FUSELIB subroutine library. Contain other required product code.
ERRORS	ERRORS.DATA	Contains all FOCUS error message text and explanations.
FOCMSO		MSO Configuration File.
FOCCONS		MSO Console Configuration File.

MSGET MSPUT	Communication Data sets	Data sets used by MSO for inter-address-space communications. One pair per MSO address space is needed. Used for CICS and TSO access, and the MSOKILL program, in conjunction with the IBI MVS Subsystem.
MSOPROF		Profile library.
MSOPRINT	SYSOUT	MSO log file.
STDOUT		MSO Console communications traces.
IBISNAP	SYSOUT	IBI Snapshot Facility dumps.

Install FOCUS

Install FOCUS Version 7.1 or higher. Use the instructions in the *OS/390 and MVS Installation Guide*.

Notes:

The data sets FOCLIB.LOAD, FOCCTL.DATA, FUSELIB.LOAD, and ERRORS.DATA should be unloaded from the tape during the installation of FOCUS.

We recommend re-blocking FOCLIB.LOAD for faster MSO startup.

System Programmer Tasks

The following tasks must be implemented in order for MSO to operate correctly:

Authorize the FOCUS Load Library

FOCLIB.LOAD, the FOCUS load library, must be an APF-authorized library, in order to activate MSO security and many other MSO features.

Note:

If FOCLIB.LOAD is authorized, then all other load libraries allocated to MSO must be authorized, unless Internal APF Authorization is activated.

Make the MSO Address Space Non-Swappable

Make sure that the MSO address space is non-swappable. An entry in the MVS PPT should be added to SYS1.PARMLIB(SCHEdnn) as follows:

```
PPT PGMNAME(SSCTL) NOSWAP CANCEL
```

Do not use the KEY 0 parameter, or any other parameter (such as NOPASS), unless the consequences of adding the parameter are completely understood by the system programmer.

Note:

FOCLIB.LOAD must be APF-authorized in order for MSO to be non-swappable.

If MSO is running as a started task, make sure that MSO will not have special security authorization, as is common with started tasks. This will interfere with the user-level security of MSO. Under RACF, there is a Started Task Table that allows the site to specify that MSO will not have special authorization. Other security packages have different requirements. See Chapter 12, *MSO Security*, for further information.

Provide JES2 Banner Support

Print files produced by MSO users have the MSO job's name printed on the banner pages. If desired, you can update JES2 to print MSO users' userids instead.

Install the IBI Subsystem, If Required

The IBI Subsystem must be installed if any of the following MSO or FOCUS facilities will be used:

- CICS access to MSO.
- TSO access to MSO.
- MSOKILL, an alternate method of stopping MSO.
- Simultaneous Usage (SU).
- The BMP extension of the FOCUS/IMS-DB data adapter.
- SmartMode, which uses SU.

If all access to MSO will be via VTAM, and neither SU, MSOKILL, nor the BMP extension will be used, then skip this step and do not allocate ddnames MSGET and MSPUT in the MSO JCL.

Set up System Security

If you are using external security you must set up your security package to work properly with MSO. Chapter 12, *MSO Security*, provides details on setting up RACF, CA-ACF2, and CA-TOP SECRET to work with MSO security.

Assign a VTAM APPLID to MSO

Assign a VTAM APPLID for each MSO address space which you will run, and, if desired, one for the MSO Console for each address space.

Each applid must be able to support multiple LU2 sessions, much like a CICS APPLID.

This sample VTAMLST member for MSO may be found in member LU2DEF of MSO.DATA:

```

*=====*
* MSO VTAM Definitions                                     *
*-----*
* This member defines the APPL statements and major node *
* needed by MSO to properly utilize VTAM. Copy it to   *
* SYS1.VTAMLST (or your site's equivalent data set). Make *
* sure it is added to ATCCON00 (or your site's member) *
* so it will automatically be started during VTAM      *
* initialization.                                       *
*-----*
*
APPLMSO  VBUILD TYPE=APPL                               Application major node
*
*-----*
*
FOCUS    APPL    AUTH=(VPACE),EAS=10,PARSESS=YES
CONSOLE  APPL    AUTH=(VPACE),EAS=10,PARSESS=YES
*
*-----*

```

File Set Up

The following steps detail the unloading of the MSO data sets and subsequent steps required in your file set up.

Unload the MSO Data sets

Unload the MSO data sets from the tape. Two data sets are required for MSO in addition to the data sets required for FOCUS:

1. MSO.LOAD (data set 35) contains the MSO load libraries for running MSO under CICS. If MSO will not be installed for CICS, you will not need this library.
2. MSO.DATA (data set 36) contains JCL and other samples needed to install MSO.

Copy FUSELIB.LOAD into FOCLIB.LOAD

If you are not using Internal APF Authorization, copy all of the modules in FUSELIB.LOAD into FOCLIB.LOAD. This removes the need for FUSELIB.LOAD to be authorized.

With Internal APF Authorization, FUSELIB.LOAD may be allocated to ddname USERLIB, and does not need to be authorized beforehand.

Create MSPUT and MSGET

If access to MSO from CICS or TSO is required, or MSOKILL will be used, create the communications data sets MSPUT and MSGET. These data sets are used in conjunction with the IBI MVS Subsystem.

These data sets should have LRECL=16, RECFM=F, BLKSIZE=16, and should be one track each. All MSO users must have read access to them; the MSO job itself will need read/write access.

Protect Module MSDEB

Module MSDEB in FOCLIB.LOAD is a debugging tool to be used with instruction from Information Builders support personnel. This module should be protected from unauthorized execute access by your site's security package. Alternatively, it can be renamed to prevent users from accessing it.

Create MSOPROF

Create the MSOPROF profile partitioned data set. This file contains the entries used by MSO for profile support. The file should have LRECL=80, RECFM=FB, BLKSIZE=1600, with DIR=30. Twenty tracks should be adequate space for this file to start.

Create member PROFILE in MSOPROF

Member PROFILE of MSOPROF should contain any FOCUS commands to be executed by all MSO users at logon time.

If you copied part or the entire FOCPARM member of ERRORS.DATA into member PROFILE in Release 6.8, you should delete those commands now. FOCPARM processing has changed, removing the necessity of duplicating any of it in MSO's profile processing.

Details on profile processing may be found in Chapter 1, *Features and Components*.

Basic MSO Customization

The following describes how to customize MSO for your site.

Create the Configuration File

The MSO configuration file is a sequential data set that is allocated to ddname FOCMSO in the MSO job. It should have LRECL=80 and RECFM=F or RECFM=FB. A sample configuration file is supplied in MSO.DATA(FOCMSO). For further details see Chapter 5, *The MSO Configuration File*.

To install MSO, the configuration file must contain at least the following entries:

```
EXTSEC  
LU2_NAME  
SERVICE  
PROGRAM  
NUMBER_READY  
MAXIMUM
```

Specify EXTSEC = YES if external security is being used.

Specify LU2_NAME if MSO is being accessed from VTAM. If not, then omit this record.

We recommend initial values for NUMBER_READY and MAXIMUM of 1 and 8, respectively.

Create the MSO JCL

Create the MSO JCL for your site. A sample is in MSO.DATA(MSO), as shown below.

```
/**** Your JOB card
//MSO      EXEC  PGM=SSCTL,REGION=32M
//STEPLIB  DD   DSN=prefix.FOCLIB.LOAD,DISP=SHR
//ERRORS   DD   DSN=prefix.ERRORS.DATA,DISP=SHR
//FOCMO    DD   DSN=prefix.MSO.MSOCONFIG,DISP=SHR
//MSOPRINT DD   SYSOUT=*,DCB=(LRECL=133,BLKSIZE=133,RECFM=FA)
//MSOPROF  DD   DSN=prefix.MSO.MSOPROF,DISP=SHR
//MSGET    DD   DSN=prefix.MSO.MSGET,DISP=SHR
//MSPUT    DD   DSN=prefix.MSO.MSPUT,DISP=SHR
//FOCCONS  DD   DSN=prefix.MSO.MSOCONS,DISP=SHR
//STDOUT   DD   SYSOUT=*,DCB=(LRECL=133,BLKSIZE=133,RECFM=F)
```

Started Task vs. Batch Job

MSO can be run as either a started task or a batch job, depending on your operational requirements.

Started tasks may only be started by MVS operators, and generally have their JCL stored in libraries that may only be updated by authorized personnel.

Batch jobs may be started by MVS operators or by anybody with access to the JCL, and can be stored in libraries that are easier to update.

Once MSO is running, there is no difference in MSO's behavior between running as a started task or as a batch job. It is important, however, that MSO not have any special security authorization if it is running as a started task; such authorization will interfere with MSO's user-level security.

Mandatory Allocations

The following ddnames are mandatory, and should be allocated as discussed here:

STEPLIB	Is mandatory unless FOCLIB.LOAD is accessible via JOBLIB or the link list. If FOCLIB.LOAD is APF-authorized, then all libraries in the concatenation must also be APF-authorized. If Internal APF Authorization is not being used, then ddnames FOCLIB and USERLIB should not be allocated; all load libraries should be allocated in STEPLIB. With Internal APF Authorization, non-authorized libraries, such as FUSELIB.LOAD, may be allocated to USERLIB.
ERRORS	Points to the ERRORS.DATA library. This must be from the same FOCUS release as FOCLIB.LOAD. If any data adapters are being used, their error message libraries must either be concatenated to, or copied into, ERRORS.DATA.
FOCMISO	Contains the MSO configuration file, as described in Chapter 5, <i>The MSO Configuration File</i> . This file may be a sequential file, an in-stream (SYSIN) data set, or a PDS member.
MSOPRINT	Points to the MSO log file. The file must be allocated with LRECL=133 and RECFM=FA or FBA. It is recommended that MSOPRINT be allocated to SYSOUT.
MSOPROF	Contains the profiles for MSO users, to be executed upon logon to MSO. While this library must be allocated, it is not required that it contain any profiles.

Optional Allocations

The following ddnames are optional, and should be allocated as needed:

MSGET MSPUT	Point to the inter-address-space communications data sets used by MSO. These are mandatory if access to MSO is being made from CICS and/or TSO or if MSOKILL is used. If all MSO access is from VTAM, then these may be omitted.
FOCCONS	Contains the configuration file for the MSO Console. It must be allocated if MSO Console is installed. This file may be a sequential file, an in-stream (SYSIN) data set, or a PDS member.
STDOUT	Is used by the MSO Console for trace information. If MSO Console is installed, and communications traces are needed, it must be allocated.

IBISNAP	Is used by the IBI Snapshot Facility. If allocated, it overrides the default allocation of SYSOUT=A and/or the default SYSOUT class specified in the MSO configuration file. In addition, allocating this in the MSO JCL causes all snapshots to be written to the same allocation, whereas the default behavior would cause a separate JES allocation for each snapshot. It is recommended that you omit IBISNAP from the JCL, and instead select a held SYSOUT class via the IBISNAP configuration parameter. For further information see Chapter 5, <i>The MSO Configuration File</i> .
USERLIB	When Internal APF Authorization is active, load libraries that are not APF authorized may be allocated to USERLIB.

Allocations in JCL

In general, any files that are read-only for all users may be allocated in the MSO JCL, and any that must be written by one or more MSO users (except IBISNAP) must not be allocated in the MSO JCL.

Files that are read-only and are being used by most or all MSO users should be allocated in the MSO JCL. However, allocations that are done in the MSO JCL may not be freed or changed by a user, so this may only be done for allocations that are completely static.

If VSAM files are allocated in the MSO JCL, you must ensure that SZERO=YES is specified in the MSO configuration file.

If you are using Internal APF Authorization (i.e., if APFAUTH=INTERNAL is specified in the MSO configuration file), then you must allocate ddname USERLIB in the MSO JCL.

The ddnames SYSIN, SYSPRINT, OFFLINE, and FOCSORT should not be allocated in the MSO JCL. If they are allocated, they will be ignored -- they must be allocated by the individual FOCUS subtasks, if needed.

Set up MSOKILL JCL

Create the MSOKILL JCL if required. The MSOKILL job is an alternate method of bringing down MSO without requiring access to an MVS Console or the MSO Console facility.

Sample JCL for MSOKILL is available in MSO.DATA(MSOKILL). Make sure that the communications data sets MSGET and MSPUT are the same data sets as in the MSO JCL.

If MSOKILL will not be used, MSO must be brought down using the STOP or MODIFY operator commands (see Chapter 11, *MSO Operator Commands*), or the STOP or SHUTDOWN commands in the MSO Console (see Chapter 13, *The MSO Console*).

Installing MSO Components

The following sections describe how to install the components of MSO.

Install the MSO Console

In order to activate the MSO Console, perform the following steps:

1. Install a VTAM applid for the Console (see Assign a VTAM APPLID to MSO on page 6-5).
2. Create a Console configuration file, using the sample in MSO.DATA(SSCHCNFG) as a template. The only record that should be changed is the "local_lu_name" record; it should be updated to reflect the VTAM applid assigned (This record must be in lower case).
3. Allocate ddname FOCCONS in the MSO JCL; it should point at the Console configuration file.

If communication traces are needed, ddname STDOUT must be allocated as well.

4. If you will be using a VTAM session manager, refer to Chapter 10, *Testing and Logging on to MSO*.
5. Set up security for the Console. Security specifications are placed in the MSO configuration file, and are described in Chapter 5, *The MSO Configuration File*. You may further customize Console security by tailoring the SSCONSEC table, as described in Chapter 13, *MSO Security*.
6. If MSO is running with APF authorization, ensure that SMFNUM either is allowed to default or is explicitly specified in the MSO configuration file.

See Chapter 13, *The MSO Console*, for detailed installation procedures, and for a discussion of MSO Console security.

Set Up CICS Access

Installing CICS access to MSO requires:

- Updating the MSO JCL.
- Updating the MSO configuration file.
- Updating the CICS JCL.
- Installing CICS programs.
- Installing CICS transactions.
- Link-editing CICS programs.

- Updating the CICS PLT.
- Optional customization.

Appendix D, *Installing MSO Supported Data Adapters*, contains detailed instructions for installing the CICS components of MSO. Refer to these instructions if you will be accessing MSO from CICS.

Customize the VTAM Logon Screen

The Information Builders supplied logon screen may be modified within certain restrictions. Sample source code for the assembler routine SSLGSCRN is available in member SSLGSCRN of FOCCTL.DATA. This routine contains the logic required to build the Information Builders-supplied logon screen, as well as any restrictions to modifications. It is recommended that experienced assembler programmers do any modification to this routine.

JCL to assemble and link the routine into the MSO modules is available in member SSLGJCL of FOCCTL.DATA. The screen may have a maximum length of 1920 bytes, including all 3270 orders; as shipped, it is approximately 1000 bytes.

Install Exits and Zaps, as Required

MSO provides several exit points and zaps which may be used to customize MSO for your environment. You should review these and install those that are appropriate for your system.

Chapter 15, *Site-Specific Customizations*, provides documentation on the following exits:

- MSIDVER: VTAM user authentication.
- MSIDTR: Userid translation.
- MSCXUID: Supply CICS userid.
- MSUSRXT: MSO initialization and termination exit.
- MSVTXT: Parse VTAM Text logon string.
- SSUSRACT: Account field validation (VTAM only).
- MSSMFXT: Customize SMF records.
- MSCXWRT: Customize CICS termination processing.
- FOCUSID: Set the FOCUS DBA password.

In addition, Chapter 15, *Site-Specific Customizations*, documents the following zaps:

- IBISNAP: Disable display of user screens in IBISNAP dumps.

- SUBMITZ2: Enable a site-specific job submission exit.
- DYNZAP: Enable allocation of the internal reader.

CHAPTER 7

Load Balancing

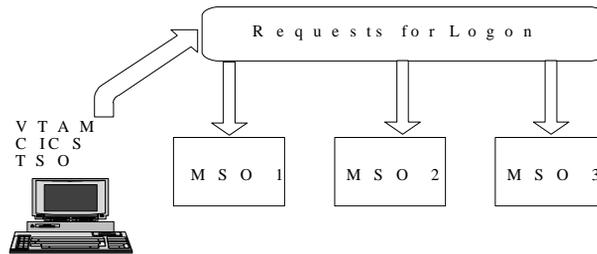
Topics:

- Overview
- MSO Load Balancing Configuration Parameters
- Load Balancing Logon Procedures
- Operational Troubleshooting
- Operational Benefits of MSO Load Balancing

Load Balancing allows MSO administrators to distribute FOCUS users across multiple MSO regions to reduce contention for system resources and provides a transparent and efficient way to distribute users across multiple MSO regions.

With Load Balancing, the MSO administrator can control access to and the use of resources. The administrator's job is simplified with Load Balancing because users are routed only to the MSO regions that are set up for their applications. All users have access to MSO FOCUS with a single profile and are automatically routed to the region that has sufficient capacity.

Users are routed to the MSO region with the applications they wish to run. For MSO administrators, all communications setup is done once. Implementing Load Balancing is as simple as adding a keyword to the configuration file.



Overview

As MSO users Logon, Load Balancing has the capability of automatically distributing them across a group of up to 32 MSO regions within a single MVS image. Users may connect to MSO via TSO, VTAM, or CICS. MSO, in conjunction with the IBI Subsystem at the time of initial connection, determines which MSO address space in the group of defined and started address spaces is the “best one” to connect this MSO user.

The actual routing is based upon the number of users currently logged on to each MSO region, the level of CPU activity each MSO region has incurred during its last MSO Resource Monitor (MRM) interval, and finally, all things being equal, a “Round Robin” selection.

Load Balancing simplifies the setup and installation of multiple MSO regions. Communication is set up with one primary MSO region. The MSO Logon procedure is just one VTAM applid or VTAM menu selection, or one CICS transaction. The search for an available MSO region is transparent to the user.

Load Balancing introduces the concepts of groups or application groups and applications. Put simply, a group is a set of MSO regions participating in Load Balancing. The MSO administrator defines the group. There can be one or more groups but there must be at least one. The MSO administrator chooses a single group when the user community is fairly homogeneous. There can be up to 32 MSO regions in a group.

MSO administrators can use application groups to segregate users into homogenous groups of MSO regions. For example, you might create TEST, PROD, and STAGE groups. Or, an application group may be created to segregate users into non-competing (for resources) groups. To keep users in Accounting from competing for resources with Engineering, the MSO administrator could set up three application groups, giving each the number of MSO regions required for that user community.

A given MSO region can belong to only one group. If it does not belong to any group, that region does not participate in Load Balancing.

Groups can be further subdivided into mutually exclusive applications. An application is a name representing users with similar needs and Logon Profiles. It is not necessary for the MSO administrator to assign applications. If no application names are assigned, all users will be balanced across all MSO regions assigned to the application group.

If a single group has many applications, the MSO administrator can assign application names to one or many of the participating MSO regions, effectively restricting those users with the same application to a subset of the total number of MSO regions available.

Applications further restrict competition between MSO regions. If for example the engineering group has a very important application (Cost Analysis), the MSO administrator can assign that application to many of the MSO regions to optimize Load Balancing. Less important applications (Vacation Requests) can be assigned to only a few of the group's regions.

An application name can appear in multiple groups and has different characteristics in each group. If the FOCUS users are updating an existing application then that application name can exist in the PROD, TEST, and STAGE groups. For example, the On-line Ordering System has been extensively changed. Some users are using the new system (STAGE), some users are using the old system (PROD) and there are the developers making changes (TEST). Each set of users is running the application named On-line Ordering System but the On-line Ordering System is different in each group.

MSO Load Balancing Requirements

The IBI Subsystem must be active and the MSO program libraries must be APF authorized to support this feature.

MSO Load Balancing Approach

MSO Load Balancing depends upon a repository of information that pertains to all MSO regions in the MVS system. Each MSO region updates its own status information stored in ECSA. In this Common Storage Area, above the 16-megabyte virtual storage line, storage is not taken up in what is termed the Private area. It does not affect the amount of private storage below the 16-megabyte line available to all MVS programs. MSO uses the IBI Subsystem to update this information in ECSA. It stores information about the number of users currently logged on and the amount of CPU that it consumed during its last MRM interval.

Load Balancing Sequence

Logon Request: to Primary or initially contacted MSO Region

1. The user initiates the MSO Logon request to an MSO region that is part of an MSO Load Balancing group. This initial connection is based upon the MSO communication methods supported from TSO, VTAM, and CICS. The initial logon request may specify the Load Balancing Application Group and application name to override the settings in the IBI Subsystem.
2. CICS users make initial contact based upon the MSO transaction name they specified to invoke MSO.
3. TSO users are connected to the primary MSO region based upon the current allocation of the MSGET and MSPUT DDnames when the MSUSER command is executed.
4. The VTAM user establishes communication based upon the VTAM applid specified by the user in a VTAM Logon command or supplied by a VTAM selection menu displayed to the user.

If the VTAM Logon request did not contain a userid, the MSO Logon Screen is displayed to the user by the primary or initially contacted region.

All three communication environments allow the user to specify the Load Balancing Group name and the application name that controls MSO region selection by Load Balancing. If the connection request does not specify an MSO Application Group or application, the defaults that are set in the IBI subsystem are used. The initially contacted MSO region does not do any Logon verification of the user. After entering a userid, the next step begins. The CICS and TSO user does not specify userid or password as this is carried over from the originating environment.

5. The primary or initially contacted MSO region now calls the IBI Subsystem, which selects the Load Balancing Group and particular MSO region to route the user to, based upon the Logon distribution algorithm described previously. The selected region may be the one initially contacted. If there are no MSO regions with the specified application name, the Logon ends and the user returns to their initial environment with an explanatory message.
6. The Primary MSO region dynamically passes the user over to the selected MSO region. The destination MSO region now processes the Logon request normally.

The VTAM user sees the MSO Logon Screen if the minimum required information for Logon to the target region was not specified.

For the VTAM Logon, the minimum information required is Userid and Password if EXTSEC=YES is specified in the MSO Configuration File and Userid is required if EXTSEC=NO.

6. A security check is done for all Logons specifying the class, APPL, using the entity name of the application name.

At this point Logon processing usually completes and the TOE screen is displayed. However, the Logon request may fail because of resource constraints in the destination MSO region. .

```

          FFFFFFFF      00000      CCCCC      UU      UU      SSSS
          FFFFFFFF      0000000      CCCCCC      UU      UU      SSSSSS
          FF          OO      OO      CC      CC      UU      UU      SS      SS
          FF          OO      OO      CC          UU      UU      SSS      INFORMATION
          FFFFF      OO      OO      CC          UU      UU      SSSSS
          FFFFF      OO      OO      CC          UU      UU      SSS      BUILDERS ,
          FF          OO      OO      CC      CC      UU      UU      SS      SS
          FF          0000000      CCCCCC      UUUUU      SSSSS      INCORPORATED
          FF          00000      CCCCC      UU      SSSS

          USERID      ==>
          PASSWORD      ==>
          SERVICE      ==>
          APPLICATION ==>
          APPL GROUP  ==>
          ACCOUNT      ==>
          USER PARM   ==>

          NEW PASSWORD ==>
          REPEAT NEW PSWD ==>
          SECURITY GROUP ==>

          SERVER REGION      PGMRAMM2
          VTAM APPLID       PMRMM203
          TERMINAL           T37CT01B

          PRESS PF3/PF15 TO EXIT

```

7. The Destination MSO region calls the IBI Subsystem to update the Load Balancing information stored by the subsystem to reflect the addition of an another MSO user.

Setting Load Balancing Defaults in the IBI Subsystem

Defaults for both the Load Balancing Group name and the application may be set in the IBI Subsystem. These defaults take effect whenever a user accesses an MSO Load Balancing Region without specifying an APPLGROUP name or an APPL name. When these defaults are set in the IBI Subsystem for the group or application name it takes effect for all non-specific MSO access requests. The initially contacted MSO region, in this case, does not control the Load Balancing Group that the user is placed into.

The Load Balancing Subsystem defaults are set via subsystem commands that are entered as follows:

- MVS Console

```
IBIS SET APPLNAME = applname  
IBIS SET APPLGROUP = groupname
```

- SYSIN to the SUBSYSI job when starting the IBI Subsystem or while it is active

```
SET APPLNAME = applname  
SET APPLGROUP = groupname
```

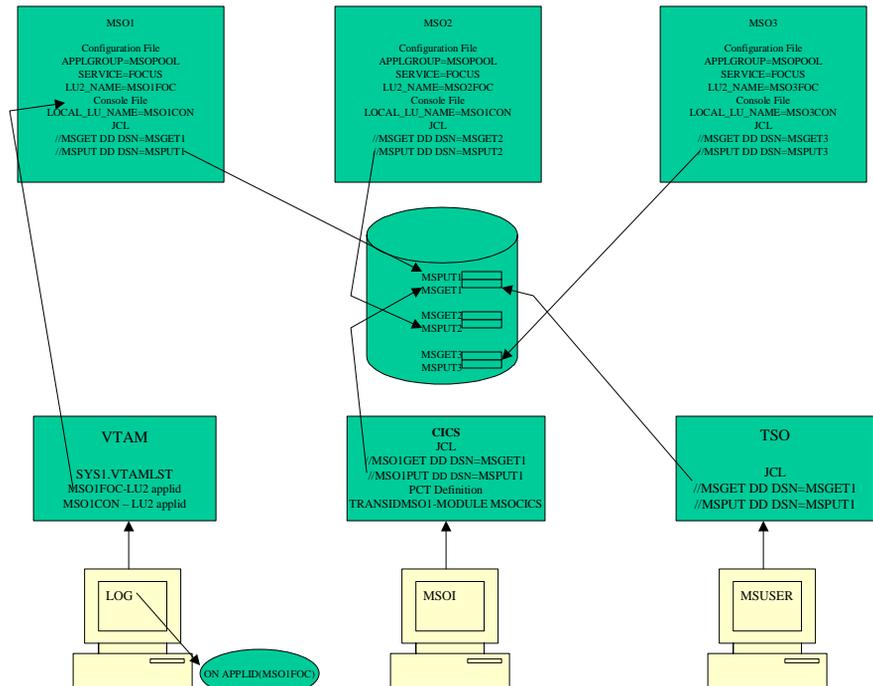
The current IBI Subsystem settings for the Load Balancing defaults may be displayed by executing the command in any of the above formats:

```
IBIS DISPLAY LOADBALANCING
```

MSO Load Balancing Configuration Parameters

See Chapter 5, *The MSO Configuration File*, for details on Load Balancing configuration parameters.

Minimum Parameters for Load Balancing

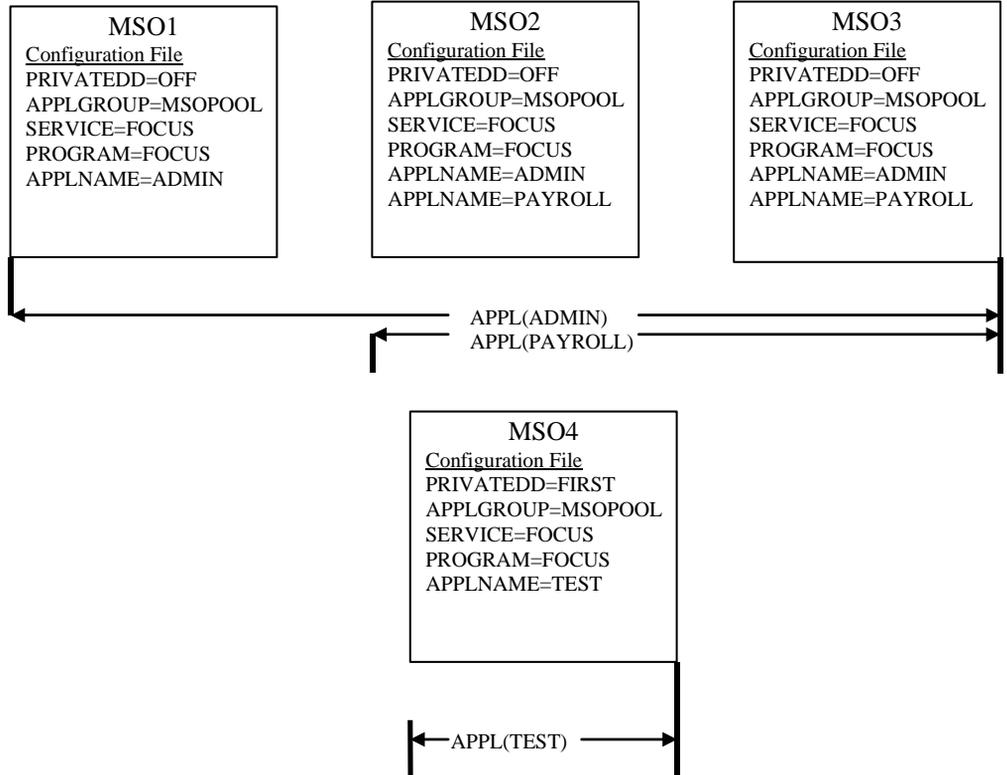


The preceding figure illustrates the minimum entries that must be present in the MSO Configuration File to implement Load Balancing. It also shows the connectivity methodology used by MSO for CICS, VTAM, and TSO. The `APPLGROUP=groupname` parameter must be present in all MSO Configuration Files of those MSO regions that are to participate in the same MSO Load Balancing group. This is the minimum entry required to activate MSO Load Balancing. The IBI Subsystem must be active at the time that these MSO regions are started.

MSO Load Balancing configuration parameters do not have to be specified. This works because the IBI Subsystem defaults for Load Balancing group and application are used if set and if not, the default APPLNAME for a service block is its service name (SERVICE=). The user trying to Logon who does not specify APPL(..) picks up the default APPLNAME of the first service block of the MSO region that is initially contacted. If all MSO users intend to run in the same environment, then there is no need to segment the Load Balancing group. APPLNAME then serves no purpose, as all users would be specifying the same application name. This also simplifies the MSO Logon specification for the user and bypasses the requirement in CICS for the installation to supply a parameter list to the CICS MSO transaction when a selection menu is presented to the CICS user.

In the preceding figure, the MSO region, MSO1, is only the primary or initially connected MSO region. Using a single Load Balancing connection region is not required. Any of the MSO regions can be connected to by the user as the initial connection. Dedicating a single region has the advantage of not having to define the other regions to the connection logic. For the TSO user only a single set of communication data sets has to be defined. This is also true for the CICS region and in addition, only one CICS MSO transaction has to be defined. The VTAM user only has to know a single applid to enter on the LOGON command.

Segmenting an MSO Load Balancing group



The preceding figure illustrates an MSO Load Balancing group that has been segmented into several target sub-groups. All of the MSO regions are in the same Load Balancing group. This is indicated by the `APPLGROUP` parameter in each MSO configuration file. All four have “MSOPOOL” specified.

It can be advantageous to do this type of segmenting. Frequently there is a need to set up a special environment for a group of users or a particular application. It may be necessary to separate these different environments because of setup conflicts or to reserve certain resources to a group of users or to a particular application.

A user first enters the MSO Load Balancing group by initiating a Logon request to an MSO region that is part of the group. Any of the four MSO regions shown above may be the one that the user initially contacts. The destination or target sub-group of regions is based upon the parameters specified in the Logon request. When that sub-group is determined, an individual MSO region in that sub-group is selected based upon the Load Balancing selection criteria.

The preceding figure has three separate sub-groups an MSO user can Logon to. The following are the different groups and the parameters that determine them. None of the three sub-groups includes all four MSO regions that make up this Load Balancing group.

1. 3 MSO regions, MSO1 - MSO3

`APPLNAME=ADMIN`

MSO1 through MSO3 have ADMIN specified as one of their eligible application names. When the Logon request specifies APPL(ADMIN), these regions are selected for Load Balancing routing.

This sub-group is also selected if the Logon request does not have an application name. Default processing is to inspect the first service block in the initially contacted MSO region. The first application name in the first service block is selected as the default application name. When there are no application names defined to the first service block, the service block name is selected as the default. In this case, ADMIN defaults as the application name.

2. 2 MSO regions, MSO2 - MSO3

`APPLNAME=PAYROLL`

For this sub-group to be selected for Logon routing, APPL(PAYROLL) must be present in the Logon request. There is no way to default to these groups as the default application name for the server block is ADMIN.

3. 1 MSO region, MSO4

`APPLNAME=TEST`

For this sub-group to be selected for Logon routing, APPL(TEST) must be present in the Logon request when the initially contacted MSO region is not MSO4. When the initially contacted MSO region is MSO4 the default application name is TEST as this is the first (and only) application name specified.

In this example, MSO4 is intended to be used as a maintenance or test MSO region. PRIVATEDD has been set to FIRST so a private set of libraries may be allocated to override the general set.

Load Balancing Logon Procedures

An initial MSO Logon request may contain any of the following parameters:

- Applgroup - The name of the MSO Load Balancing Group that is to be accessed.
- Application - The application name used to sub-divide the total MSO Load Balancing group.

- Service - The name of the service block of the initially contacted region. This service block name is only searched for in the initially contacted region. The service block will be found only if APPLNAME=NONE has been coded for it.
- Account - The MSO account number for this Logon. To validate or alter the supplied account number, use exit SSUSRACT. MSO.DATA includes this member. A maximum of 40 characters may be specified in this field.
- Userparm - The arbitrary string of data that is available in the FOCUS environment using the MSOINFO module. A maximum of 256 characters may be specified for userparm.

The following FOCEXEC lines will retrieve the value that was specified at Logon into the variable, USERPARM.

```
-SET &USERPARM = MSOINFO('USERPARM', 'A256');
-TYPE USERPARM IS: &USERPARM
```

VTAM Logon Request

The VTAM user initiates the MSO Logon by entering the VTAM LOGON command and specifying an applid as follows:

LOGON APPLID(msoapplid)

The user sees the MSO Logon Screen as shown in the MSO FOCUS Logon Screen. At that time the lower right hand section of the screen displays the default service and application. When Load Balancing is turned on for the region presenting the screen, application is the field that controls the connection possibilities for the user. The default application is the APPLNAME= setting for the first service block in the MSO region that is displaying the screen.

Only one of the fields, SERVICE or APPLICATION may be specified. If there are no entries in these fields, the Logon request is processed using the default application name displayed on the panel. If using Fast Logon with only the userid specified, the MSO Logon Screen is displayed with both the default service and default application names in the input fields. The user should blank out one of these fields, enter any of the other fields they desire and press Enter.

VTAM users may also use a Fast Logon to specify one or more of the fields that appear on the MSO VTAM Logon Screen. The following is the syntax for VTAM fast Logon:

```
LOGON APPLID(msoapplid) DATA(userid/password/service/account/newpass/group/
application/userparm/applgroup)
```

Parameters not specified are indicated by a double /. Below is an illustration of a Fast Logon request specifying userid, password, account number, application name, and user information.

```
LOGON APPLID(MMMMM203)
DATA(MYUSR1/PASSWORD//ACCT99476///PAYROLL/YEAREND/MSOGRP1)
```

TSO Logon Request

TSO users must pre-allocate certain data sets and then invoke the MSUSER module to connect to MSO. See the *OS/390 and MVS Installation Guide* for a CLIST example that allocates the required data sets.

The MSUSER module may be invoked as a command or via the CALL command. The format for the command invocations is:

```
MSUSER application,applgroup,service,account,userparm
```

or

```
MSUSER 'application,applgroup,service,account,userparm'
```

The fields are positional and must be specified in the order listed. Skipped fields are represented by null commas. To specify only account and userparm use the following syntax.

```
MSUSER ',,ACCT123,USERDATA'
```

The parameter string must be enclosed in single quotes if initial input fields are skipped. The commas are unconditional field delimiters and may not be specified as part of the data. Single quotes may be included within the data by entering two single quotes. Entering single quotes around a comma insures that the comma is not treated as data.

The format for invoking MSUSER with the CALL command is below. The parameter string must always be enclosed in single quotes. The parameter parsing is the same as described for the command invocation above.

```
CALL 'prefix.FOCLIB.LOAD(MSUSER)'  
'application,applgroup,service,account,userparm'
```

CICS Terminal Logon Request

The Information Builders provided module, MSOCICS, provides the capability of specifying the same four parameters. The format for the command is:

```
MSO APPL(application_name)|SERVICE(service_name)  
APPLGROUP(groupname)  
ACCT(account_information)  
USERPARAM(user_information)
```

The MSOCICS module must have a CICS transaction ID assigned to it. The screen should be cleared before entering the MSO transaction.

CICS Load Balancing API

Those installations that call Information Builders MSOCICS transaction from a program must pass the control block structure illustrated below. The CICS-MSO Interface block should be passed in the CICS commarea when the MSO module MSOCICS is invoked via EXEC CICS XCTL. The storage for the second block, CICS - MSO Logon Buffer, should be obtained using EXEC CICS GETMAIN.

Prior to entering data into any of the fields in this structure, the interface block should be initialized to nulls (X'00') and the Log buffer should be initialized to blanks (X'40').

MSO-CICS Interface Block				
+0	func set to x'01'	msoname	reserved	reserved
+10		reserved		
+20	logbuf->	reserved		
+30		reserved		
+40		reserved		

MSO-CICS Logon Buffer				
+0	application name		service name	
+10		reserved		
+20	group name	length of X'08'	reserved	
+30		reserved		
+40	account	length of X'28'		
+68	user parm	length of X'100'		

The field contents for these two control blocks are:

Field Name	Length	Type	Contents
func	4 bytes	numeric	1 - signifies MSO Logon
msoname	4 bytes	character	MSO CICS transaction name
logbuff	4 bytes	31 bit address	Pointer to MSO log buffer
application	8 bytes	character	Load Balancing application name
service	8 bytes	character	Service Block name
group name	8 bytes	character	Load Balancing Group name
account	40 bytes	character	MSO account information
user parm	256 bytes	character	User information available to executing FOCEXEC

Planning for MSO Load Balancing

The most important consideration in planning for Load Balancing is to make sure that when the user arrives at their destination it is where they expected to be. The MSO user expects the characteristics of their environment to be the same every time they Logon with the same application name. Application names are merely addresses to MSO Load Balancing. They are a way of ensuring that the user is serviced by an MSO region that has the application as a valid address. The actual names have no meaning beyond this.

MSO Load Balancing Requirements

The IBI Subsystem must be active and the MSO program libraries must be APF authorized to support this feature.

Load Balancing Defaults

When the Load Balancing defaults are set in the IBI Subsystem for the group or application name, it takes effect for all non specific MSO access requests. The initially contacted MSO region, in this case, does not control the Load Balancing Group that the user is placed into.

Consistent Destination Environment

All MSO regions running the same applications should have the same operating environment. The same application must run identically in each region.

For planning purposes, the following considerations affect the execution characteristics of an MSO region.

- Global Allocations

The libraries allocated for FOCUS libraries should be identical. If different Master File allocations are made, for example, then the same application may behave differently.

- Profile Data Sets

The need for profile data sets is reduced with the availability of Load Balancing. Profiles tailor the environment when users were logging on to the same MSO region. If used, it is imperative that all MSO regions in a target sub-group have identical profiles for each user. The easiest way to affect this is to allocate the same MSO profile data set.

- Configuration Parameters

The following MSO Configuration File parameters affect the execution environment of an MSO region. They should be identical across Load Balancing sub-groups.

EXTSEC

When set to NO, all users have the same security access as the MSO region in which they are executing. When set to YES, each user executes with the established security for the userid.

PRIVATEDDD

This parameter affects the logical search order of FOCUS libraries and files.

- MSO Exits

Ensures that all MSO exits are identical.

Define Load Balancing Group Names to SAF

Define Load Balancing APPLGROUP names as entities in the APPL class. There is a SAF READ access check based upon the APPLGROUP name for jobs attempting to start a MSO Load Balancing Region.

The IBM RACF® product does not fail the request for an undefined entity. However, the security check is effectively nullified.

The CA-ACF2® SAF product fails a request for an undefined entity. In this environment, Load Balancing requires defined application names and users with READ access.

Consistently use the APPLICATION Parameter in All Service Blocks

When the APPLICATION parameter is introduced into a MSO Load Balancing group all service blocks should have this parameter specified. This avoids confusion resulting from the application name defaulting to the service block name when the parameter is not used.

MSO Region Capacity

The MAXIMUM setting in the MSO Configuration File for each MSO region should not be set too low as compared to the amount of virtual storage that is available below and above the 16-megabyte line. This avoids the operational difficulty of having a MSO Logon request denied for a particular region when less than the minimum amount of storage is left above or below the 16m line.

All MSO Regions Should Have a Console Available

The console provides a look into the MSO region showing very useful information about the MSO users' activities. It is a valuable aid in problem determination.

Minimize Dividing the MSO Load Balancing Group

To make administration easier, keep the number of subsets of MSO regions in an APPLGROUP to a minimum. The less subsetting, the more effectively the Load Balancing algorithm can balance Logons across MSO regions.

Operational Troubleshooting

- Why don't my users end up in the regions that I expect them to?

Ensure all the regions have the same name specified for APPLGROUP.

If the Logon specifies an application name, make sure the name is in a service block in each MSO region you want available to the MSO Load Balancing group.

If the Logon does not specify application name ensure that all targeted MSO regions in the group have the same service name as specified for the first service block in the initially contacted MSO region.

If the logon does not specify a Load Balancing Group name or application name, then the defaults may be set in the IBI Subsystem. Issue the subsystem command:

```
IBIS DISPLAY LOADBALANCING
```

This shows the current default settings. Once default settings are set in the IBI subsystem, all access requests that are not supposed to use these defaults will require that the group and application name be specified.

- The distribution of users does not seem to be even across MSO Load Balancing regions.

Load Balancing only has an effect at Logon time. If users FIN out of one or more regions faster than the rate of new users logging on, then the distribution will become unbalanced.

Dividing the Load Balancing group can produce imbalances across MSO regions in the APPLGROUP.

- One region is rejecting all MSO Logons and users cannot get on even when other regions are available.

The placement algorithm for Logon routing MSO users into load balanced regions does not guarantee the region can handle it. If one region becomes overstressed due to the amount of activity but it has the lowest number of users, users are routed to it and fail in Logon. This can be remedied in one of two ways:

1. Start a new MSO region with identical parameters to the target set that is having problems. A region is then available with 0 users logged on. All new users are routed to this address space.
2. Issue the MSO QUIESCE command for the region rejecting Logons. This region then removes itself from eligibility in the Load Balancing group. The other regions are selected for Logons.

- My payroll application fails in one Load Balancing region.

Ensure that the environment for the failing region is identical to the region in which the application works properly. Before you make any changes, review the other applications assigned to that region making sure the correction will not affect those applications adversely.

- A user is having a problem. How do I find out in which MSO region they are executing.

Have the user execute a FOCEXEC that calls the MSOINFO module. This module now returns the application name in addition to the service block name and server name. The following lines in a FOCEXEC accomplish this:

```
-SET &SERVER = MSOINFO('SERVER ', 'A8');  
-TYPE SERVER JOBNAME IS &SERVER  
-SET &SERVICE = MSOINFO('SERVICE ', 'A8');  
-TYPE SERVICE BLOCK NAME IS &SERVICE  
-SET &APPLICAT = MSOINFO('APPLICAT', 'A8');  
-TYPE APPLICATION NAME IS &APPLICAT  
-SET &APPLGROUP = MSOINFO('APPLGROUP', 'A8')  
-TYPE GROUP NAME IS &APPLGROUP
```

The first parameter in the invocation of the MSOINFO subroutine must be padded to 8 characters.

- A Logon request with SERVER(FOCUS) is being rejected with the message LOGON FAILED: SPECIFIED SERVICE UNKNOWN when there is a service block with FOCUS as its service name.

The ability to access a service block with the SERVER(...) parameter on an MSO Logon request becomes disabled when Load Balancing is active in the region. The service block can only be reached this way if APPLNAME=NONE is specified and the region is initially contacted by setting Load Balancing defaults in the IBI Subsystem

CICS Problem Determination Procedures

MSO users may capture a dump of an IBI internal trace file. CSS researchers or IBI programming may request this trace for problem determination. This function requires additional DDnames be allocated to the CICS region. FOCDMP00 through FOCDMPnn may be allocated. At any time a CICS user may issue the following command to cause the current contents of the MSO CICS trace to be written to the selected DDname.

```
MSMT DUMP {*|transaction_name} DDSUFFIX(nn)
```

where:

*

Denotes printing the trace information for all MSO regions connected to this CICS region.

transaction_name

Specifies the associated MSO transaction ID or the MSO region you wish to trace to limit the listing to that one connection.

nn

Specifies the corresponding FOCDMPnn pre-allocated file to which the trace information is written. DDSUFFIX defaults to 00.

Allocate the FOCDMPnn data sets to a SYOUT class or to a sequential file of 4 tracks. These data sets should have an LRECL of 102.

Operational Benefits of MSO Load Balancing

The primary benefit derived from MSO Load Balancing is the ability to evenly distribute MSO Logon requests across multiple MSO regions. This makes monitoring of MSO regions less costly to administrative personnel. It also ensures that MSO users will get more even-handed performance characteristics when they Logon to MSO.

There is more flexibility for the MSO Administrator in controlling how many resources to give to MSO Users. The connection to all but the primary or initially contacted MSO region need not be known by the connecting environments, including the primary MSO region. Additional MSO regions may be started via the console or by submitting a job. These new regions must have their own unique connection data sets allocated (MSGET/MSPUT) and VTAM applids if they are to be accessible from VTAM. However, they become automatically available to the APPLGROUP when they have completely initialized.

The connecting environments, TSO, CICS, and VTAM only have to be set up to connect to the primary or initially contacted MSO region.

- The TSO CLIST/EXEC only has to allocate the MSGET/MSPUT communication data sets of one of the MSO regions in the MSO Load Balancing group.

- The VTAM Logon command/session manager screen only has to specify the MSO applid (lu2_name in the MSO Configuration File) of one of the MSO regions in the Load Balancing group.
- The CICS regions that connect MSO Users to the MSO Load Balancing group only have to have one pair of communication data sets allocated. In addition, only one MSO transaction id has to be defined.

CHAPTER 8

MSO/CICS Installation

Topics:

- Preparing for Installation
- Installation Steps
- Installation Testing
- Operations
- End-User Issues
- MSO/CICS Technical Description
- Troubleshooting

Please read this chapter carefully before starting the installation of MSO/CICS. It provides important information that must be considered during installation.

Preparing for Installation

The installation procedure assumes that all of MSO has been installed except for the CICS feature. If possible, you should test the MSO installation with VTAM or TSO access before starting the CICS portion of the installation. We strongly recommend installing MSO/TSO access for diagnostic purposes, even if it is not otherwise needed.

The installation procedure also assumes that the IBI Subsystem has been installed to provide inter-address-space communications. This should be tested before attempting to use it with MSO/CICS, ideally by installing and running MSO/TSO.

Installation Requirements

Any MVS and CICS releases supported by IBM are supported for the MSO/CICS Interface, although running CICS Version 3 or later provides enhanced trace support. Information Builders currently supports any release of OS/390 currently supported by IBM, and CICS 4.1 or higher.

Both FOCUS and MSO must be installed from the same tape. No mixing of releases is allowable.

All three MSO access methods — VTAM, CICS, and TSO — can be used concurrently in the same MSO address space.

Planning

Before starting the installation, you should decide:

- How many MSO address spaces each CICS needs to communicate with. You will need a separate transaction id and a separate pair of communication data sets for each MSO address space.
- How many CICS address spaces will be running MSO/CICS. You will need to repeat the installation procedure for each CICS.
- If you will be using MRO. Some of the MSO/CICS code must run in the TOR, although most of it runs in the AOR.
- If you will be using the Attention Key support. This allows users to interrupt a running FOCUS command, and optionally end the command or the entire FOCUS session.
- If you will be using the Session Termination Key. This allows a user to end a FOCUS session by pressing a PA or PF key, and is provided for compatibility with previous releases.

- Whether or not you need to start another transaction upon exiting from FOCUS. You can specify this for all MSO address spaces, by updating and reassembling a table, or in individual MSO address spaces, by specifying it in the configuration file. You may choose, instead, to run a user exit when the MSO transaction ends, which also allows suppression of the session termination message.
- If ASSIGN USERID functions properly at your site, or if you need to supply the proper userid through a user exit.
- If you want to automate startup and shutdown through the PLT, or if you want to manually start and stop MSO/CICS operation.

Required Data Sets

Please note that all Information Builders-supplied data sets must come from the same tape. It is not permissible to mix data sets from different releases of FOCUS. In addition, any sample jobs, such as table updates, zaps, assemblies, and link-edits, should be taken from the current tape. These samples change significantly from release to release.

In addition, you should review your MSO JCL and configuration file against the current samples and documentation to confirm their continued correctness in the current release.

File	Description
FOCLIB.LOAD	Contains FOCUS and MSO product code, and MVS subtasks that run in the CICS address space.
MSO.LOAD	Contains the MSO CICS programs: MSOCICS, MSOUTIL, MSOPLT, MSOATTN, and MSOTOR. Also contains the locally assembled MSOEXIT (the XZCATT exit) and any MSO/CICS user exits that are in use.
MSO.DATA	Contains sample JCL, zaps, assembly and link-edit jobs, and source code for CMSOTABL and MSOEXIT.
ERRORS.DATA	Contains all FOCUS error message text and explanations. Member CMSO1ERR contains messages issued by MSO/CICS.
MSO Configuration File	Specified by ddname FOCMSO in MSO JCL; is modified during the MSO/CICS installation.
MSO JCL	JCL for the MSO address space; is modified during the MSO/CICS installation.
CICS JCL	JCL for the CICS address space; is modified during the MSO/CICS installation.
Communication data sets	These data sets are used by MSO for inter-address-space communications. One pair per MSO address space is needed. The pair is shared by all CICS regions that communicate with the MSO address space. DDnames used are MSGET/MSPUT in the MSO address space, and xxxxGET/xxxxPUT in the CICS address space, where "xxxx" is the MSO transaction id.

MSO SASLIB Installation Guideline

Customers have a choice between two versions of SASLIB. For specifics see the *OS/390 and MVS Installation Guide*. The TRANSIENT version of SASLIB may cause conflicts in versions of SASC when operating in an address space in conjunction with other software using SASC runtime routines.

As a result of this conflict, the following messages may appear on the JESLOG or MVS console

```
+LSCX048 Most recent C runtime library modules not available
+          Use version 93028C5E ( 5.50C) or later to avoid problems.
```

Information Builders recommends that all MSO customers use the ALLRES version of SASLIB in all MSO/CICS regions.

Affected release levels are:

- MVS release 5.2 and above. This includes OS/390.
- FOCUS Version 7.0 and above.

MSO User Exits

The following user exits have not changed since Version 6.8:

- The MSCXUID exit, used to supply a userid when ASSIGN USERID does not function properly or when a different userid is desired.
- The MSCXWRT exit, used to override the message issued by MSO at normal session termination and the normal next-transaction processing.

Considerations for Multiple Address Spaces

You may interconnect multiple CICS and MSO address spaces. When doing so, ensure that all MSO address spaces, and all MSO programs in the CICS address spaces, are at the exact same release level of MSO and FOCUS. While crossing of releases might work under some circumstances, it is not a supported configuration. There are no restrictions on mixing different releases of CICS, however. Any combination may be used, as long as each release used is supported by MSO.

Installation Steps

The following sections describe the steps necessary to install MSO/CICS.

MSO Setup

The following steps are used to prepare the MSO address space for communications with one or more CICS address spaces. They should be performed once for each MSO address space that will be accessed from CICS.

- As mentioned above, the IBI Subsystem must be installed in the MSO address space before the MSO/CICS installation is performed. Ensure that this has been tested before continuing. (The best test is to set up access to MSO from TSO and to confirm that this works properly.)
- Add the MSGET and MSPUT ddnames to the MSO JCL. Note that the ddnames are always MSGET and MSPUT in the MSO JCL regardless of the ddnames that are used in the CICS JCL. There is only one pair of communication data sets per MSO address space; it is shared by all CICS and TSO address spaces that are communicating with that MSO. An example of these DD cards is:

```
//MSGET DD DSN=PROD.MSO.MSGET,DISP=SHR
//MSPUT DD DSN=PROD.MSO.MSOPUT,DISP=SHR
```

- Update the MSO configuration file:
 1. Ensure SERVICE=FOCUS is specified. Only MSO/VTAM users may connect to a service with a name other than FOCUS; MSO/CICS (and MSO/TSO) users are restricted to SERVICE=FOCUS.
 2. If desired, code the TRANSID parameter to cause the MSO transaction to start another transaction when it completes. This specification overrides the default that is set by re-assembling CMSOTABL.
 3. If desired, code the CICSBREAK parameter. This parameter activates the Session Termination Key, which replaces the old CICS Attention Key feature.

CICS Setup

Perform these steps once for each CICS address space to be connected to an MSO address space:

- Choose transaction ids for the two MSO transactions. This manual uses the name MSO for the transaction used to access MSO, and MSMT for the utility transaction used to terminate MSO operations. Any valid transaction ids may be used, but keep in mind that the name of the MSO transaction will affect the ddnames used in the CICS JCL for the communication data sets.

- Change the CICS JCL:
 1. Include the data set MSO.LOAD to the DFHRPL concatenation. If required by your version of MVS, ensure the blocksizes of the data sets in the concatenation are in descending order, or override them if necessary.
 2. Add a DD statement for FOCLIB, pointing to FOCLIB.LOAD.
 3. Add a DD statement for ERRORS, pointing to ERRORS.DATA.
 4. Add the MSOGET/MSOPUT ddnames. These are the communications data sets created above. The actual ddnames used should be the transaction id used for the MSO transaction, with GET and PUT appended to the name. For example, if you use transaction id FOC, you would use ddnames FOCGET and FOCPUT.
 5. If using a version of CICS that uses the OSCOR parameter, ensure that it is set to a large enough value. MSO needs at least 165K above current needs.

Thus, the following might be added to the CICS JCL:

```
//DFHRPL DD DSN=prefix.MSO.LOAD,DISP=SHR,DCB=BLKSIZE=23476
//FOCLIB DD DSN=prefix.FOCLIB.LOAD,DISP=SHR
//ERRORS DD DSN=prefix.ERRORS.DATA,DISP=SHR
//MSOPUT DD DSN=prefix.MSO.MSOPUT,DISP=SHR
//MSOGET DD DSN=prefix.MSO.MSOGET,DISP=SHR
```

Do not put these DD statements into the CICS FCT. These files are not going to be opened by CICS, but rather are used via MVS facilities.

- Add the following programs to the PPT:
 - MSOCICS - This is the MSO transaction, run by end users.
 - MSOUTIL - This is the MSMT utility transaction, currently used for stopping MSO communications links.
 - MSOPLT - This program is run from the PLT to shut down all active MSO communications at CICS termination.
 - MSOATTN - This program is part of the Attention Key support under MRO.
 - MSOTOR - This program is part of the Attention Key support under MRO.
 - MSOEXIT - This program is used as the XZCATT exit if Attention Key support is needed.

The following RDO (Resource Definition Online) definitions are compatible with the storage protection attribute STGPROT=YES in the DFHSIT (System Initialization Table for the MSO region):

- Information Builders-supplied PROGRAMS that are defined to CICS should specify EXECKEY(CICS).
- The MSO/CICS TRANSACTION definition should specify TASKDATAKEY(USER).

All of the above programs should have LANG(ASSEMBLER) specified.

While DTIMOUT is valid for program MSOCICS, you should consider using the TIMEOUT parameter in the MSO configuration file instead, as it provides for an orderly shutdown of the FOCUS session when the session times out.

The DTIMOUT parameter may not be specified for program MSOTOR.

- Add the following transactions to the PCT:

Transid	Program
MSO	MSOCICS
MSMT	MSOUTIL
MTOR	MSOTOR (MRO only)
MSAT	MSOATTN (MRO only)

- Set up transaction security as required:
 1. MSO must be run by all users.
 2. MSMT is run by personnel authorized to stop MSO communications.
 3. MTOR is started at CICS startup.
 4. MSAT is started by MTOR when required.
- Link the IBM Command-Level stubs into the programs in MSO.LOAD. Sample JCL is provided in MSO.DATA(MSLNKSTB).
- Change the PLT for proper operation at CICS startup and shutdown. Details on PLT processing are given in the Operations section below.
 1. If Attention Key support is needed, enable MSOEXIT (an XZCATT exit) at startup.
 2. Run MSOPLT at shutdown.
- If Attention Key support is needed, update, assemble and link-edit MSOEXIT. You must update the table at the end of this exit with the transaction ids that you will be using for MSO. Source code is provided in MSO.DATA(MSOEXIT), and sample JCL is provided in MSO.DATA(MSXZCATJ).
- Set up the MSCXUID exit if required. The MSCXUID exit is used to supply a CICS userid when ASSIGN USERID is not supported, or when it is desired to use a different userid than the one returned by ASSIGN USERID. This exit is described in MSO/CICS Technical Description on page 8-14.

- Set up the MSCXWRT exit if required. The MSCXWRT exit allows customization or elimination of the MSO13206 message when the MSO transaction completes normally; it also suppresses the next-transaction processing. This exit is described in MSO/CICS Technical Description on page 8-14.
- If you need to modify the defaults set in CMSOTABL, update and re-assemble it. Source and JCL for this are provided in members CMSOTABL and MSOTABLJ in MSO.DATA. You may change the user auxtrace number, the default value for TRANSID, and the flag to suppress the session initialization message.

CICS Setup for MRO

The following steps are only required when Attention key support is required in an MRO environment. Except as described below, all MSO transactions and programs run in the AOR and are fully compatible with MRO.

1. Define transaction MTOR to run in the TOR.
2. Ensure that exit XZCATT (program MSOEXIT) is enabled in the TOR at CICS startup.
3. Start transaction MTOR from the PCT after XZCATT is enabled.
4. Ensure that transaction MSAT, which runs in the AOR, can be started by transaction MTOR in the TOR.

Multiple Address Spaces

When connecting multiple CICS address spaces to a single MSO address space, simply perform the installation procedure once in the MSO address space (adding a single pair of MSGET and MSPUT data sets), and once per CICS address space (adding the MSO transaction and xxxxGET and xxxxPUT data sets under any valid names, which may be different in each CICS address space).

When connecting to multiple MSO address spaces from a single CICS address space, install an extra instance of the MSO transaction (using program MSOCICS) and an extra pair of communication data sets (appending GET" and PUT to the transaction id to create the ddnames) for each MSO address space to be used. For example, assuming transactions MSO and MFOC were both installed pointing to program MSOCICS, you would have the following allocations in the JCL for CICS:

```
//MSOPUT DD DSN=prefix.MSO.MSOPUT,DISP=SHR
//MSOGET DD DSN=prefix.MSO.MSOGET,DISP=SHR
//MFOCPUT DD DSN=prefix.MSO.MFOCPUT,DISP=SHR
//MFOCGET DD DSN=prefix.MSO.MFOCGET,DISP=SHR
```

The first MSO would have the following allocation for its communication data sets:

```
//MSPUT DD DSN=prefix.MSO.MSPUT,DISP=SHR
//MSGET DD DSN=prefix.MSO.MSGET,DISP=SHR
```

The second MSO would have its communication data sets allocated as follows:

```
//MSPUT DD DSN=prefix.MSO.MFOCPUT,DISP=SHR
//MSGET DD DSN=prefix.MSO.MFOCGET,DISP=SHR
```

Installation Testing

To confirm proper installation and operation of MSO/CICS, use the following steps:

Testing Basic Functionality

1. Start CICS and MSO. Check their log files to ensure normal startup.
2. Run the MSO transaction. You should get message MSO13239, followed by the FOCUS Terminal Operator Environment Screen. Message MSO13029 should appear in the MSO log.
3. Run a few commands in FOCUS to ensure the session is behaving properly, and then issue FIN to end the session.
4. Verify that the message 13028 appears in the MSO log and that MSO13300 appears on your CICS terminal.
5. Run MSO again, to ensure communication restarts properly.
6. Run MSO from a second terminal, to ensure multiple sessions can be run concurrently.
7. Run MSMT.
8. Bring down CICS and MSO; to ensure both can come down without any problem.

Testing Attention Key Support

1. As described in Operations on page 8-11, enable XZCATT and run transaction MTOR.
2. Start an MSO session.
3. Start a long-running TABLE request.
4. Press Attention while the request is running, then press Enter to resume the request and ensure the table completes normally.
5. Run two more TABLE requests, trying the KX and HX options at the Attention prompt.

6. Press Attention during a terminal read and confirm that it is ignored.

Testing PLT Functions

1. If using Attention Key support, enable the XZCATT exit and (if using MRO) run MTOR from the PLT.
2. Run MSO. Make sure Attention Key support works if you are using it.
3. Stop CICS and ensure MSOPLT runs so CICS comes down cleanly.

Operations

MSO/CICS is designed to run with little or no operator intervention. All programs needed to establish and destroy the environment can be run from the PLT at CICS startup and shutdown. These programs may also be run manually, if desired.

MSO/CICS Startup

MSO/CICS is started when the first user runs the MSO transaction. This causes the communication tasks to start and establishes communication with the MSO address space that corresponds to the specified transaction. (The first user to run the MSO transaction will encounter a slightly longer wait as communication is established.)

Reusable MSO/CICS Control Blocks

MSO provides a facility to re-use orphaned control blocks. If you are timed out or disconnected the control block you were using goes into a pool. If you then come back you will get your old control block as long as it's still available. If the system runs out of control blocks it will start re-using the oldest orphaned control blocks. When you time out or if your session was cancelled, restarting your MSO connection after your control block was reissued, you will receive the following message:

```
FOCUS SESSION ENDED: ERROR UNKNOWN OR NO LONGER AVAILABLE
```

Startup for Attention Key Support

If you are using the optional Attention key support, you must explicitly enable the XZCATT exit. In addition, if Attention support is desired in an MRO environment, the MTOR transaction must be run. Both of these steps must be performed before the first use of the MSO transaction in order for the Attention key support to be active.

The XZCATT exit must be enabled in each CICS region that contains terminals that will run MSO. Under MRO, it must be enabled in each TOR that will access MSO. It is not necessary to enable it in the AOR. The following command may be used to enable XZCATT:

```
CECI ENABLE EXIT(XZCATT) PROGRAM(MSOEXIT) GALEN(nnn) START
```

The value for GALEN is calculated by multiplying the maximum number of termids that MSO/CICS will encounter by 8, and then adding 8 bytes for overhead. It is recommended that you use double the number of anticipated termids, to allow for growth. There is no indication of overflow, and the Attention key is ignored from termids not in the table.

After XZCATT is enabled, run the MTOR transaction. MTOR is a long-running transaction (it runs in a loop), and under most circumstances should not be run from a terminal. One way to start it is with the command:

```
CECI START TRANSID(MTOR)
```

MSMT - MSO Master Terminal

The MSMT transaction can be used to stop communications with one or more MSO address spaces from a CICS address space. To run MSMT, issue the command:

```
MSMT STOP(transid)
```

where:

transid

Is the transaction being used to stop communications between an MSO address space and a CICS address space.

```
MSMT STOP(tran1,tran2,...)
```

where:

tran1, tran2

Are the transactions being used to stop communications between multiple MSO address spaces and a CICS address space.

Note that MSMT doesn't delete the MTOR transaction, which can be stopped in the same manner as any other CICS transaction.

CICS Shutdown

When CICS is shut down, the program MSOPLT should be invoked from the PLTSD list. MSOPLT performs the same processing as MSMT, stopping communication with all attached MSO address spaces and deleting the MVS subtasks that perform the communication.

End-User Issues

The following issues may impact end-users.

Starting a FOCUS Session

A FOCUS session is started under MSO/CICS by running the MSO transaction. This transaction may have any name, as long as the installation procedures are properly followed.

While waiting for the session to start, the user will see message MSO13239. (This message may be suppressed by updating CMSOTABL.) After the session is established, the user sees the FOCUS Terminal Operator Environment screen and may begin using FOCUS.

Ending a FOCUS Session

A FOCUS session is normally ended by issuing the FIN command or by executing the -QUIT FOCUS command in a Dialogue Manager procedure. It is also possible to end a session by using the Attention key or the Session Break key, as discussed below.

It is not possible to temporarily suspend a FOCUS session. The session must be ended in order to return to CICS.

Depending on how the session is ended (FIN,abend, Attention key, Session Termination Key, or lost communications), the appropriate MSO132xx message is issued. The only exception to this is that when a session is ended normally (via FIN) and the MSCXWRT user exit is installed, message MSO13206 will not be issued.

After the session ends, and the appropriate message (if any) is displayed, MSO performs one of the following actions:

- If MSCXWRT is installed, transfer control to the exit.
- If TRANSID is specified in the MSO configuration file, start the specified transaction.
- If CMSOTABL has been updated and NEXTTRAN is not blank, start the specified transaction.
- Wait for the user to press Enter, and then clear the screen.

Keyboard Locking

Just as with MSO/VTAM, MSO/TSO, and TSO FOCUS, the 3270 keyboard is now locked during processing. It is only unlocked when FOCUS is waiting for input from the terminal. As discussed below, the Attention key may be used to break the lock.

Attention Handling

When the Attention key is pressed while the keyboard is locked, the processing is identical with the processing under MSO/VTAM. The user is presented the options of ending the current command, ending the session, or resuming processing.

The Attention key is ignored if pressed when the keyboard is not locked.

Session Termination Key

A PA key or a PF key may be defined as an MSO session termination key with MSO/CICS. If pressed when the keyboard is unlocked, this causes immediate termination of the session.

It is recommended that the standard FIN command be used instead of this feature. The Session Termination Key feature is only available under MSO/CICS. It is not active under MSO/VTAM or MSO/TSO. This feature is identical to the CICS Attention Key feature in previous releases of MSO, and is provided for upward compatibility.

MSO and Other CICS Programs

MSO/CICS uses the transaction id, as read from the EIB, to identify which communication data sets to use. Therefore, the MSOCICS program can only be run under the proper transaction name. You must issue a START for the MSO transaction to run it from another program.

You can incorporate MSO into a menuing system by issuing a START for the MSO transaction, and by using the TRANSID parameter in the MSO configuration file (or by re-assembling CMSOTABL) to cause MSO to run the proper transaction when a session ends.

MSO/CICS Technical Description

This section provides information about the internal structure and functionality of MSO/CICS. This information will help you understand MSO/CICS processing, make appropriate decisions during installation, and troubleshoot installation and operational problems.

Included are discussions of steady-state operation, starting and stopping links to MSO address spaces, starting and stopping user sessions, attention key handling, operations under MRO, transaction security, retrieving userids, and MSO/CICS user exits. There are also tables giving information on all MSO/CICS transactions and programs, including user exits.

Steady-State Operation

When MSO/CICS is active, the following run in the CICS address space:

- One pair of MVS subtasks, running programs MSXGET and MSXPUT, for each MSO address space with which the CICS address space is communicating.
- One series of pseudo-conversational MSOCICS transactions for each user currently in session with MSO.

The MSXGET and MSXPUT subtasks communicate with their counterparts in the MSO address space, MSPUT and MSGET. They pass messages in processor storage, using the IBI Subsystem to perform the communications. These communication methods are described in the *OS/390 and MVS Installation Guide*. Please note that the communication data sets that are set up during installation do not contain the actual messages being passed; they are used only for initialization and control of the cross-address-space communications.

The MSOCICS program is installed with transid MSO (default). This pseudo-conversational transaction:

- Reads the user's screen.
- Passes it to MSXPUT to be shipped to the MSO address space.
- Waits for a response from the MSO address space to be received by MSXGET.
- Displays the response on the screen.
- Ends the transaction.

Please note that all actual MSO/FOCUS processing is performed in the MSO address space; the only work done in the CICS address space is screen I/O.

In addition to the above, when using Attention Key support, the program MSOEXIT must be installed as an XZCATT exit. Furthermore, when using Attention Key support in an MRO environment, the long-running transaction MTOR is running in the TOR, and the transaction MSAT is started in the AOR when Attention is pressed. Further details on Attention Key support are provided below.

Initialization and Termination of the MSO Link

Initialization of the MSO Link by MSOCICS

Other than the special considerations required for Attention Key support, nothing need be done to start the link from CICS to an MSO address space. It is automatically activated when the first user runs the MSOCICS program, normally installed with transid MSO. When a user invokes the MSO transaction, it checks to see if the link to the MSO address space is running. If not, it performs initialization tasks and starts MSXGET and MSXPUT (the communications subtasks, described above) to activate the link.

Termination of the MSO Link by MSOUTIL

MSOUTIL, the MSO utility program, is normally installed as the MSMT (MSO Master Terminal) transaction. It can be used to bring down the link(s) to one or multiple MSO address spaces. It stops the MSXGET and MSXPUT subtasks and frees the resources associated with the link(s) to each MSO specified.

Termination of the MSO Link by MSOPLT

The MSOPLT program runs at CICS shutdown. Like MSOUTIL, it brings down all links to MSO address spaces, stops all of the MSXGET and MSXPUT subtasks (which is necessary in order to prevent SA03 abends at CICS termination), and frees all MSO/CICS resources.

Initialization and Termination of a User Session

MSO/CICS Initialization

When invoked, the MSOCICS program (by default, installed with transid MSO) reads its transid from the EIB. Using this transid, it determines which pair of MVS communications subtasks and which pair of communications data sets are to be used in communicating with the MSO address space. The transid and the ddnames of the communications data sets must correspond. It is also crucial that MSO/CICS always be run by invoking its associated transaction; Calling it directly causes it to read an incorrect transid and prevent proper communications.

As discussed above, if the link to the appropriate MSO address space is not up, MSOCICS starts the MSXPUT and MSXGET subtasks. If the link is up, MSOCICS initializes an MSO/FOCUS session for the user, and then starts normal processing.

MSO/CICS Termination

When a user's MSO/FOCUS session is ended, MSO/CICS will:

1. Determine if the MSCXWRT user exit is installed.
2. For normal termination of a session (via the FIN command), display the MSO13206 message only if the exit is not installed.
3. For all other types of session termination (e.g., Attention key, Session Break key, abend, MSMT brought down the link, MSO has been shut down), display an appropriate message about why the session has been ended.
4. If the exit is not installed, and if TRANSID has been specified (either in the MSO configuration file or by reassembling CMSOTABL), then start the desired transaction. (Note that the transid in the configuration file will override the transid in CMSOTABL.)
5. If the exit is installed, run it.

Should a session failure (such as TIMEOUT) occur while the keyboard is unlocked, the appropriate message will display when the user next presses the Enter key.

The MSCXWRT User Exit

As described above, by installing the MSCXWRT user exit, you can suppress both the display of the MSO13206 message and the next-transaction processing.

MSCXWRT is started via XCTL, and is run:

- Immediately, after normal session termination (suppressing the MSO13206 message)
- After the appropriate message is displayed, in the event of an abnormal session termination

Source code for a sample MSCXWRT exit is provided in member MSCXWRTS of MSO.DATA, with JCL in member MSCXWRTJ. This sample executes a transaction without displaying any messages or waiting for the user to press Enter. You may modify this sample to perform any processing you desire.

To install this exit, assemble and link the source code to create a module called MSCXWRT. This module should be added to the PPT, and made available in one of the DFHRPL libraries, such as MSO.LOAD.

Attention Key Handling

MSO's Attention key handling is an optional feature; if not installed, MSO will ignore the Attention key. The feature requires installation of the XZCATT exit. CICS allows more than one user program to run at this exit point, so MSO's processing will not interfere with any other use of this exit on your system. Source code for the exit is supplied in member MSOEXIT of MSO.DATA; it must be assembled and linked with the libraries for your release of CICS.

Attention Key support is more elaborate in an MRO environment, as described in the next section.

MRO Issues

MSO is fully compatible with MRO. You need to ensure the MRO link is configured properly, and to perform additional setup for Attention key support.

Userid Propagation Under MRO

MSO depends upon proper functioning of the ASSIGN USERID command in order to support user-level security. Therefore, in order for the userid to be transferred from the TOR to the AOR, the connection between the TOR and the AOR must have the ATTACHSEC parameter set to IDENTIFY.

Attention Key Support Under MRO

When Attention Key support is desired in an MRO environment, please note the following:

- The XZCATT exit is loaded in the TOR, not the AOR, as it is handling terminal-related control blocks.
- The MTOR transaction must be running in the TOR. This transaction, which may have any name, is a long-running transaction that must be active while any MSO sessions are running.
- The XZCATT exit must be enabled before the first invocation of the MSO transaction. If MSO is run first, it will determine that XZCATT is not enabled and will permanently (for the life of the CICS address space) mark Attention Key support as not installed.
- When it detects that Attention has been pressed, MTOR starts the MSAT transaction in the AOR. MSAT then notifies the MSO transaction. Please note that the MSAT transid is hard-coded in MSOTOR. If necessary, you may zap MSOTOR with another transid.

Transaction Security

There are no special requirements for transaction security with the MSO/CICS transactions. The Summary of MSO Transactions and Programs on page 8-20 lists the circumstances under which each transaction is run. You should ensure that end users who are authorized to access MSO/FOCUS can run MSO. The other transactions are started by operators, the PLT, or other transactions, as shown.

Userid Processing

MSO needs to determine the user's userid during session establishment, in order to activate user-level security. This allows MSO to properly identify users to MVS, to ensure proper access to files and other protected resources.

MSO uses the ASSIGN USERID command to determine the userid. If your system does not support this command, the userid is not available to MSO, and user-level security is not available for MSO/CICS sessions.

You can test whether this works on your system by using the IBM-supplied CECI transaction to execute the ASSIGN USERID command and examine the results. In an MRO environment, as mentioned above, you must specify ATTACHSEC(IDENTIFY) for ASSIGN USERID to work properly in an MRO environment.

If you desire, you can turn external security off (via the EXTSEC parameter in the MSO configuration file), allowing any user to log onto MSO and have the same security rights as the MSO address space. This means, however, that MSO features such as GETUSER will not operate properly, and your system's security policy could be compromised, since users might have access to files for which they are not authorized.

If your system does not support ASSIGN USERID, or you wish to use a different userid than is provided by CICS, MSO provides the MSCXUID user exit. If the CICS program MSCXUID exists, it is called with an 8-byte COMMAREA, into which the program must insert the userid to be used by MSO. If the program does not exist, then MSO executes an ASSIGN USERID to get the userid.

The MSCXUID User Exit

Source code for a sample MSCXUID exit is provided in member MSCXUIDS of MSO.DATA, with JCL in member MSCXUIDJ. This sample does an ASSIGN OPID to retrieve the operator ID and puts this into the COMMAREA. This causes MSO to use the OPID instead of the CICS userid. You may modify this sample to provide any userid you desire.

To install this exit, assemble and link the source code to create a module called MSCXUID. This module should be added to the PPT, and made available in one of the DFHRPL libraries, such as MSO.LOAD.

Summary of MSO Transactions and Programs

Transaction	Program	Runs in	MRO Only?	Run by
MSO	MSOCICS	AOR	No	End user
MSMT	MSOUTIL	AOR	No	Operator
MTOR	MSOTOR	TOR	Yes	PLT, at startup
MSAT	MSOATTN	AOR	Yes	MTOR, upon ATTN

Program	CICS or MVS?	Transid	Runs In	MRO Only?	Use
MSOCICS	CICS	MSO	AOR		User's MSO session.
MSOUTIL	CICS	MSMT	AOR		Utility transaction, used for stopping link(s) to MSO address space(s).
MSOPLT	CICS		AOR		Shuts down all MSO links and MVS subtasks at CICS termination; run from the PLT.
MSOATTN	CICS	MSAT	AOR	Yes	Part of Attention Key support under MRO; started by MSOTOR when Attn is pressed.
MSCXUID	CICS		AOR		Optional user exit to replace ASSIGN USERID call.
MSCXWRT	CICS		AOR		Optional user exit to replace display of final message and link to transaction specified by TRANSID.
MSOTOR	CICS	MTOR	TOR	Yes	Part of Attention Key support under MRO; loops until Attn has been pressed, then invokes MSAT. Must be started before first use of MSO transaction.
MSOEXIT	CICS		TOR		CICS task attach exit (XZCATT). Must be enabled before first use of MSO transaction.
MSXGET	MVS		AOR		Part of inter-address-space communications; receives data from MSO address space.
MSXPUT	MVS		AOR		Part of inter-address-space communications; transmits data to MSO address space.

Troubleshooting

In the event of a problem with MSO/CICS, the following points may prove helpful before contacting Information Builders technical support:

- MSO/TSO and MSO/CICS use similar communications paths. If TSO access to MSO does not work properly, it is extremely unlikely that CICS access will work. We recommend that you install MSO/TSO for testing purposes, even if it will not normally be used for access to MSO. Once TSO access works properly, troubleshooting CICS access is much simpler.
- The following are the most common causes of problems with MSO/CICS:
 - The MSO configuration file does not contain a definition for SERVICE=FOCUS.
 - The CICS command-level stubs have not been link-edited into the MSO/CICS modules.
 - The MSO address space is not running, or is malfunctioning. This is most easily tested by trying to access it via TSO or VTAM.
 - The communication data sets have not been created properly.
 - The communication data sets are not allocated to the correct ddname or dsname. (Remember, the ddnames are different in the MSO and the CICS address spaces.)
 - The FOCUS libraries allocated to FOCLIB and to DFHRPL come from different releases of FOCUS.
 - The FOCUS libraries allocated to the MSO address space and to the CICS address space come from different releases of FOCUS.
 - All of the FOCUS files are not allocated properly in the CICS address space.
 - The userid is not available via the ASSIGN USERID command, requiring use of the MSCXUID user exit.
 - Program MSOPLT is not being called from the PLT at CICS shutdown. This is a common cause of SA03 abends in the CICS address space.
 - The PPT and PCT entries for the MSO/CICS programs and transactions have not been created or are incorrect.

Diagnostic information

When contacting Information Builders technical support for MSO/CICS problems, please have the following information available:

- The JCL for the MSO and CICS address spaces.
- The MSO configuration file (ddname FOCMSO in the MSO JCL).
- RDO definitions of all MSO/CICS transactions and programs.
- Any PLT entries created for MSO/CICS.
- All installation and customization job streams.
- An exact description of the failure symptoms, including error message numbers.

In addition, a CICS auxtrace of the MSO and/or MSMT transaction might be requested, depending on the nature of the problem.

In CICS V3 and above, this will include MSO's user-defined records as well as standard CICS records.

CHAPTER 9

Testing and Logging on to MSO

Topics:

- VTAM Access
- CICS Access
- TSO Access
- Accessing Multiple MSO Regions

After MSO is properly installed it is imperative to conduct a series of tests involving VTAM, CICS, and TSO.

After the installation of MSO, the MSO job is submitted or the MSO started task is activated. When MSO is ready to accept logons, the message

```
(MSO13023) ALL INITIAL SERVERS STARTED
```

is displayed on the system console and in the MSO log.

You should log on to MSO using each access method that has been installed, and verify that the Terminal Operator Environment Screen appears on your terminal.

VTAM Access

From a VTAM terminal, access to MSO is accomplished by logging onto MSO's APPLID. If you have set up a special VTAM command or option on a menu to enter MSO, simply enter the VTAM command on the screen or select the option. If not, access MSO by entering a VTAM logon command as follows:

```
LOGON APPLID(msoappl)
```

where:

```
msoappl
```

Is the MSO APPLID set up by the VTAM system programmer.

The Information Builders-supplied MSO VTAM Logon Screen is displayed:

```

      FFFFFFFF      00000      CCCCCC      UU      UU      SSSS
      FFFFFFFF      000000      CCCCCC      UU      UU      SSSSS
      FF          00      00      CC      CC      UU      UU      SS      SS
      FF          00      00      CC          UU      UU      SSS
      FFFFF      00      00      CC          UU      UU      SSSSS
      FF          00      00      CC      CC      UU      UU      SS      SS
      FF          000000      CCCCCC      UUUUU      SSSSS
      FF          00000      CCCCC      UUU      SSSS
      INFORMATION
      BUILDERS,
      INCORPORATED

      USERID      ==>
      PASSWORD    -->
      SERVICE     ==>
      APPLICATION ==>
      APPL GROUP  ==>
      ACCOUNT     ==>
      USER PARM  -->

      NEW PASSWORD ==>
      REPEAT NEW PSWD ==>
      SECURITY GROUP ==>

      DEFAULT SERVICE      FOCUS
      DEFAULT APPLICATION
      DEFAULT APPL GROUP

      SERVER REGION      PMSMSOR1
      VTAM APPLID        QA38M218
      TERMINAL            VTAM220

      PRESS PF3/PF15 TO EXIT
```

From this screen, the userid and password are entered. The MSO Terminal Operator Environment screen is then displayed.

Interfacing with VTAM Session Managers

If you are using MSO with a VTAM session manager, the session manager can send the logon information to MSO and automatically log users on to MSO. You can bypass the MSO VTAM Logon Screen by specifying the appropriate information in the LOGON command:

For MSO FOCUS:

```
LOGON APPLID(msoappl) DATA(userid/password/service/account/newpass/group)
```

For the MSO Console:

```
LOGON APPLID(msocons) DATA(userid/password)
```

Enclose the parameters in parentheses and separate each parameter with a slash (/). The maximum length of the DATA string for MSO FOCUS is 85 bytes (five 8-byte fields, one 40-byte field, and five slashes). The maximum length of the DATA string for the MSO Console is 17 bytes (two 8-byte fields and one slash). None of the fields may contain blanks or slashes.

If only a userid is entered after DATA, the user will be prompted for a password on the logon screen, and, for MSO FOCUS, the default service will be used.

CICS Access

For CICS access, the MSO resources must be defined to the CICS in which users are running. The user enters the MSO transaction by clearing the screen and entering the command:

```
MSO
```

The FOCUS Terminal Operator Environment screen is displayed and the user may begin using FOCUS.

The name of the MSO transaction is selected when MSO/CICS is installed. MSO/CICS uses the transaction id, as read from the EIB, to identify the communications data sets to use. Therefore, the MSOCICS program can only be run under the proper transaction name. You must issue a START for the MSO transaction to run it from another program.

TSO Access

From TSO, each user executes the MSOUSER CLIST (provided by the MSO installer) to be used for the MSO region desired. This CLIST calls the MSUSER module that connects a user to the MSO region.

The following sample MSO CLIST is provided in MSO.DATA(MSOUSER).

```
ALLOC F(FOCLIB) DA('prefix.FOCLIB.LOAD') SHR REU
ALLOC F(USERLIB) DA('prefix.FOCLIB.LOAD') SHR REU
ALLOC F(ERRORS) DA('prefix.ERRORS.DATA') SHR REU
ALLOC F(MSGET) DA('prefix.MSO.MSGET') SHR REU
ALLOC F(MSPUT) DA('prefix.MSO.MSPUT') SHR REU
ALLOC F(MSOPRINT) DA(*)
CALL 'prefix.FOCLIB.LOAD(MSOUSER)'
```

After a few seconds, the FOCUS Terminal Operator Environment screen appears and the user may begin using FOCUS.

We recommend that CICS sites test the installation of MSO using TSO if possible, before testing access from CICS.

Accessing Multiple MSO Regions

If multiple MSO regions are active, MSO users must select the proper region. This is done differently for each access method.

VTAM users select an MSO region by specifying the proper applid in the VTAM LOGON command.

CICS users select an MSO region by executing the proper version of the MSO transaction, which has a different name for each MSO region.

TSO users select an MSO region by allocating the corresponding MSGET and MSPUT files in the CLIST that calls MSUSER.

CHAPTER 10

MSO Usage Accounting

Topics:

- Enabling Usage Accounting
- Processing SMF Records Using FOCUS
- SMF Fields Recorded
- Modifying and Redirecting SMF Records

MSO can write SMF records that give details on individual users' sessions. These records are written at logon and logoff time, and provide information about the session, the userid, and resources consumed.

You can use your standard SMF accounting software to process these records or, if desired, you can use FOCUS to produce MSO usage reports.

MSO provides the capability of modifying these records and redirecting them to somewhere other than the SMF datasets, if desired.

Enabling Usage Accounting

Usage accounting requires that MSO be running from an APF-authorized library. If this is the case, then usage accounting can be activated by specifying SMFNUM in the MSO configuration file. See Chapter 5, *The MSO Configuration File*, for details on SMFNUM.

Processing SMF Records Using FOCUS

Because FOCUS Master Files are provided for the SMF VSAM and dump files, it is a simple matter to create FOCUS TABLE requests to generate any report of usage accounting that you might require. To do this, you must make the Master Files available by either:

- Allocating the MASTER ddname, using the DYNAM command, to the MSO.DATA dataset, or
- Copying members SMFVSAM and SMFFIX into a MASTER library.

Next, the SMF VSAM file or the SMF dump file must be allocated via DYNAM.

Once this is done, a TABLE request can be run with either SMFVSAM or SMFFIX. If using SMFFIX, ensure that MAXLRECL is set to 32767.

For example:

```

SET MSG = ON
SET MAXLRECL = 32767
DYNAM ALLOC FILE MASTER DSNAME prefix.MSO.DATA SHR
DYNAM ALLOC FILE SMFVSAM DSNAME SYS1.SMFDUMP.DATA SHR
-RUN
DEFINE FILE SMFVSAM
CPU/D8.2 = SMFOFCPU / 100 ;
TIME/D9.2 = SMFOFLTM / 360000 ;
HR/I2 = SMFOFTME / 360000 ;
MIN/I2 = (SMFOFTME - (HR*360000)) / 6000 ;
TOD/A5 = EDIT(HR) | ':' | EDIT(MIN) ;
END
TABLE FILE SMFVSAM
HEADING CENTER
"MSO USAGE ACCOUNTING REPORT"
"ALL REGIONS "
" "
SUM CPU SMFOFEXC AS 'EXCP,CNT' TIME AS 'HRS'
BY SMFOFUID AS 'USER,ID'
BY SMFOFDTE AS 'DATE'
BY TOD AS 'TIME'
ON SMFOFUID SUB-TOTAL
IF SMFOFRTY EQ -2
END
-RUN

```

Note:

SMFOFRTY (the SMF number) is -2 in our example, and indicates SMF type 254. This is because FOCUS treats the I1 field in the Master Files signed. Thus, user-defined SMF numbers (128 to 255) are represented as -128 to -1 in reports.

SMF Fields Recorded

FOCUS Master Files are provided for the logoff records only. The logon records would have similar Master Files, except that the RECTYPE would have an alias of "1" and no accounting fields would be present. The source code for the MSSMFXT exit provides detailed information on the format of the records. The following fields are recorded:

Field	Contents
SMFOFFLG	Operating system flag
SMFOFRTY	Record type (SMF record number)
SMFOFTME	Time (hundredths of a second)
SMFOFDTE	Date

Field	Contents
SMFOFSID	System ID
SMFOFSBS	Subsystem (JES2, JES3, etc.)
SMFOFSBT	Record subtype (2 means LOGOFF)
SMFOFMISO	MSO jobname
SMFOFJID	JES job ID ('JOB 1234', 'STC 7658', etc.)
SMFOFASI	MSO ASID
SMFOFUID	Security userid of MSO user
SMFOFLID	Logon userid of MSO user
SMFOFRUL	TSO userid / CICS region / VTAM LUname
SMFOFCTI	CICS termid (for CICS only)
SMFOFSRV	Service name
SMFOFCNT	Connection type (1=TSO, 2=CICS, 4=VTAM)
SMFOFCC	Completion code of FOCUS task; low values if no abend has occurred
SMFOFACT	First 8 bytes of account number
SMFOFCPU	CPU time (TCB time, in hundredths of a second)
SMFOFEXC	EXCP count
SMFOFLTM	Logon duration (hundredths of a second)
SMFOFA40	Full account number

Modifying and Redirecting SMF Records

If desired, the MSSMFXT user exit can be used to:

- Modify the format of the logon or logoff record.
- Write the record to somewhere other than the SMF data stream.

See Chapter 15, *Site-Specific Customizations*, for more information on the MSSMFXT user exit.

CHAPTER 11

MSO Operator Commands

Topics:

- MVS Operator Commands
- Examples

MSO provides MVS system operators commands to stop MSO jobs, cancel users, prevent new users from logging on, allow users to log on, and monitor the CPU and I/O activity of logged on users.

MVS Operator Commands

Two MVS operator commands, Stop (P) and Modify (F), are used by MSO.

The STOP Command

The syntax for the STOP command is:

```
P jobname
```

where:

```
jobname
```

Is the MSO jobname or started task name.

After issuing this command, MSO comes down normally, as if stopped by MSOKILL.

The MODIFY Command

The syntax for the MODIFY command is as follows:

```
F jobname, STOP
```

```
F jobname, {CAN[CEL]|PURGE} {SECID|SID} = {securityid|*}
```

```
F jobname, {CAN[CEL]|PURGE} {LOGONID|LID} = {logon id|*}
```

```
F jobname, {CAN[CEL]|PURGE} TCB = tcb address
```

```
F jobname, {QUIESCE|ENABLE} [server]
```

```
F jobname, USERS
```

```
F jobname, DYNAM {ALLOC|FREE|CONCAT}
```

```
F jobname, RESTART CONSOLE
```

```
F jobname, CHAP TCB = tcbaddress,n
```

```
F jobname, SHUTDOWN [NOW|CANCEL]
```

```
F jobname, START server
```

```
F jobname, IDLELIM nnn
```

```
F jobname, TRACE trace_name [START] [,ACTIVE] [,SERVICE=service] [,COND]  
[STOP] [,DEFER] [,USERID=security]  
[level] [,TCB=address]  
[QUERY [FULL|SHORT]]
```

where:

```
jobname
```

Is the MSO jobname or started task name.

jobname STOP

Brings down MSO the same way that the P (stop) command does. Thus, there are three ways to bring down MSO: using MSOKILL, the P operator command, or the F jobname, STOP operator command.

CANCEL

Can be used to cancel a logged-on MSO user. The CANCEL command accepts as a parameter the security ID or logon ID of the user to be canceled. All duplicate security ids or logon ids will also be canceled. Any processing done by the canceled ids is immediately terminated and the resources used by the ids are released. If an asterisk (*) is used instead of a security or logon ID, then all users are immediately canceled. PURGE and CAN are synonyms for CANCEL.

securityid

Is the MSO security ID or an asterisk (*).

logon id

Is the MSO logon ID or an asterisk (*),

tcb address

Is the address of the user's TCB

QUIESCE

Prevents any new users from logging on to the specified server, but allows all the existing users to continue processing. If a server is not specified, all servers are quiesced.

ENABLE

Allows new users to log on after a QUIESCE command is issued. Thus, ENABLE undoes the QUIESCE command. If a server is not specified, all servers are enabled; otherwise, only the specified service is enabled.

USERS

Displays on the console all the current users logged on to the MSO address space along with CPU and EXCP statistics for each of the users. If MSO usage accounting is not active (see Chapter 11, *MSO Usage Accounting*) then CPU and EXCP statistics will not be present.

DYNAM

May be issued from the MVS system console. However, care should be taken to ensure that a DYNAM command will not adversely affect MSO's operation or FOCUS behavior before it is issued.

ALLOC

May be used to allocate additional files to the MSO region.

FREE

May be used with a translated ddname, to free a file allocated to an individual subtask, or with an untranslated ddname, to free a file allocated to the region.

CONCAT

May be used to produce concatenated allocations.

If a DYNAM command does not fit on a single line, you may end a line with a hyphen "-" to indicate that the command is incomplete. MSO will then issue a WTOR to request the next line of the command. Each line of the command except the final line is ended with a hyphen.

RESTART CONSOLE

Reattaches the MSO Console subtask. If the console was running, it terminates the current task and starts a new one. It is normally used to restart the MSO Console following an abend.

CHAP TCB

Changes the specified subtasks' MRM priority group. n must be 0 through 15. 15 is the highest priority, and 1 is the lowest. If 0 is specified, the subtask will be suspended.

SHUTDOWN

Quiesces all services, and sets MSO to watch for the user count to reach zero. When all users are logged off, MSO will terminate.

NOW

Is equivalent to a STOP command.

CANCEL

Undoes a pending SHUTDOWN.

START

Starts the specified server task. START is not required during normal MSO usage.

IDLELIM

Sets the idlelimit value for the region as a whole, overriding the value specified in the MSO configuration file.

TRACE

Allows activating and managing of the various traces which Information Builders products and options may generate. Specific information on traces to be generated are provided by Information Builders support personnel.

trace_name

Is the name of the trace. Wildcards using * and ? may be used.

START

Starts the specified trace(s).

STOP

Stops the specified trace(s).

level

Specifies the level of the trace. This is dependent on the specific trace that is being used.

ACTive

Specifies that this command is to operate with currently active traces.

DEFer

Specifies that this command is to be operated with traces that are not yet active.

SERVICE

Restricts the scope of the trace command, by screening on any combination of service name, security ID, and TCB address. Wildcards using * and ? may be specified. The default for all three keywords is *. TCB is ignored for deferred traces.

COND

Is only valid with the START option. It defines whether or not the trace is conditional. A conditional trace is only recorded if the appropriate ddname for the trace is already allocated; MSO's trace services do not dynamically allocate the trace file for a conditional trace.

QUERY

Displays information about the trace(s).

FULL

Displays all options used with QUERY.

SHORT

Displays a summary view of the options used with QUERY.

Examples

P SYSMSO

Stops the MSO job SYSMSO.

F SYSMSO, STOP

Also stops the MSO job SYSMSO.

F SYSMSO, CAN SECID=*

Cancels all the users in the SYSMSO job.

F SYSMSO, CANCEL LOGONID=MSOUSR1

Cancels all users with logon ID MSOUSR1 in the SYSMSO MSO job.

F SYSMSO, QUIESCE

Prevents new logons in the SYSMSO MSO job.

F SYSMSO, USERS

Displays all logged on users along with CPU and EXCP information.

F SYSMSO, ENABLE

Allows new users to once again log on after a QUIESCE command was issued.

CHAPTER 12

MSO Security

Topics:

- RACF Security for MSO
- CA-ACF2 Security for MSO
- CA-TOP SECRET Security for MSO

MSO can be used with or without an external security package such as RACF, CA-TOP SECRET, or CA-ACF2. If no security package is being used, all the users in a given MSO region inherit whatever security is in effect for the MSO job or started task. Thus, if the MSO job cannot access any file, neither can any MSO user; if the MSO job can access all files, so too can all the users under MSO.

MSO supports user-level security via IBM's System Authorization Facility, or SAF. MSO issues a SAF macro that creates an ACEE control block for each user task as the user logs onto MSO. The userid of the user is inserted into this ACEE, and it is this userid that dictates which resources the FOCUS task can access. After this ACEE is created, whenever the user's FOCUS task attempts to access system resources (usually data sets), SAF issues a call to whatever security package is installed at the site.

The userid inserted into the ACEE is actually the security ID that results when the MSIDTR userid translation exit is called. A site can thus have different security IDs from logon userids using MSIDTR. If a site does not use MSIDTR, then the logon userid is used for the security ID as well.

For VTAM access into MSO, two security checks may be done: first, a check is made that the logon userid and password are authorized, and once this is done, the security ID (SECID) may be changed via MSIDTR mentioned above. This new SECID must then be checked again. The first check can be bypassed if a site chooses to implement the MSIDVER user exit. More information on the MSIDVER exit is available in Chapter 1, *Features and Components*.

In order for user-level security to work, the following must be true:

- The FOCLIB.LOAD library must be APF authorized.
- The external security package must be aware that SAF calls are being made and must correctly process them. Also, depending on the security package, the MSO address-space-level security must either have no access to any system resources, or must have full access to all resources. What this means is that the MSO job's security must not interfere with user-level security.
- Every userid that accesses MSO must be defined to the security package.

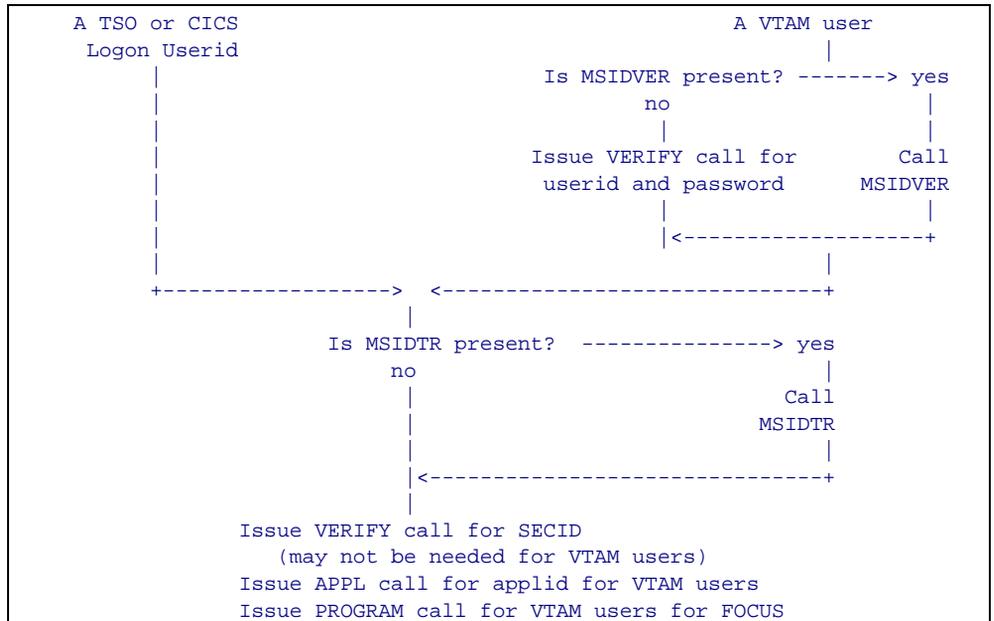
MSO issues three distinct types of SAF calls:

- VERIFY. The verify SAF call occurs when a user logs on from all the MSO terminal environments: CICS, TSO, and VTAM. A verify is done for the SECID and an ACEE is created. For VTAM access, the password is checked along with the userid.
- APPL. When a VTAM user logs on, MSO issues a SAF call for the APPLID (LU2 applid) of MSO along with the SECID. If MSO receives a return code of 0 or 4, the user is permitted to log on.

For VTAM access, the password and userid are both checked, an APPL of "MSO" and the LU2 terminal are passed.

- PROGRAM. When a VTAM MSO user logs on, MSO also issues a program SAF call, where the program name is FOCUS, for the SECID.

The following flow chart shows how these security calls are made for a FOCUS MSO user:



RACF Security for MSO

As RACF is an IBM product, it automatically interfaces with SAF. Therefore, RACF security for MSO is automatic. No additional installation step is required.

- Each user who uses MSO must be defined to RACF.
- User-level security is in effect when MSO is executed from an APF-authorized library and EXTSEC=YES is specified.

CA-ACF2 Security for MSO

Three different SAF classes of calls are issued by MSO: verify, program, and appl. MVS automatically issues data set-class SAF calls as well.

- CA-ACF2 Release 5.2 or higher should be installed.
- CA-ACF2 GLOBAL SYSTEM OPTIONS must be altered to enable SAF validation on the OPTS record. Five SAFPROT records must be added:
 1. `classes (verify,data set) cntlpt(SSFOC) subsys(SSFOC)`
 2. `classes (data set) cntlpt(SSFOC) subsys(SVC019)`
 3. `classes (data set) cntlpt(SSFOC) subsys(SVC022)`
 4. `classes (data set) cntlpt(SSFOC) subsys(SVC026)`
 5. `classes (facility,verify) cntlpt(SSCON) subsys(SSCON)`

If a site would like to verify APPL and PROGRAM SAF calls made by MSO, the first SAFPROT record class can be changed to a dash (-). In this case, the site must code CA-ACF2 rules to allow individual users to access APPL and PROGRAM entities: the MSO LU2 applid, an APPL of MSO, and program name FOCUS, respectively.

- The MSO address space must have access to all system resources that may be needed by any MSO user. CA-ACF2 checks for job-level access as well as user-level access. Thus, the job-level MSO userid must have access to all MSO data sets.
- The LOGONID of the MSO job should have the MUSASS attribute set on.
- Each MSO userid must be defined to CA-ACF2.
- MSO must be executed from an APF-authorized library, and EXTSEC=YES must be specified.

Note:

If there are problems getting CA-ACF2 to process MSO security properly, the CA-ACF2 system administrator should contact Computer Associates ACF2 Technical Services directly.

CA-TOP SECRET Security for MSO

A CA-TOP SECRET facility for the MSO job must be created and made available for CA-TOP SECRET. The INITPGM for the facility must be SSFOC, and the TENV and MULTIUSER attributes in the facility must be on. A sample facility that works at one MSO site follows:

```
FACILITY DISPLAY FOR FOCUSMSO
INITPGM=SSFOC    ID=F  TYPE=07

ATTRIBUTES=IN-USE, ACTIVE, SHRPRF, ASUBM, TENV, NOABEND, MULTIUSER, NOXDEF
ATTRIBUTES=LUMSG, NOSTMSG, SIGN(S), NOPSEUDO, INSTDATA, NORNDPW, AUTHINIT
ATTRIBUTES=NOPROMPT, MENU, NOAUDIT, RES, NOMRO, WARNPW, NOTSOC
ATTRIBUTES=TRACE, NOLAB, NODORMPW, NONPWR, NOIMSXTND

MODE=FAIL  LOGGING=ACCESS

UIDACID=8  LOCKTIME=000  DEFACID=*NONE*  KEY=8
```

- If the MSO Console is being installed, an additional facility must be defined. A sample facility and user definition follows:

```
FACILITY DISPLAY FOR FOCCON
INITPGM=SSCONS  ID=F  TYPE=99

ATTRIBUTES=IN-USE, ACTIVE, SHRPRF, ASUBM, NOABEND, MULTIUSER, NOXDEF
ATTRIBUTES=LUMSG, NOSTMSG, SIGN(S), INSTDATA, RNDPW, AUTHINIT
ATTRIBUTES=NOPROMPT, NOAUDIT, RES, WARNPW, NOTSOC, LCFTRANS
ATTRIBUTES=MSGLC, TRACE, NOEODINIT, NODORMPW, NOPWR, NOIMSXTND

MODE=WARN  DOWN=GLOBAL  LOGGING=INIT, SMF, MSG

UIDACID=8  LOCKTIME=000  DEFACID=*NONE*  KEY=8

MAXUSER=03000  PRFT=003

FACILITY(USER1=NAME=FOCCON)

FACILITY(FOCCON=PGM=SSCONS, ID=F)
FACILITY(FOCCON=ACTIVE, SHRPRF, ASUBM, NOABEND, MULTIUSER, NOXDEF)
FACILITY(FOCCON=LUMSG, NOSTMSG, SIGN(S), NOPSUEDO, INSTDATA, NORDNPW, AUTHINIT)
FACILITY(FOCCON=AUTHINIT, NOPROMPT, MENU, NOAUDIT, RES, MOMRO, WARNPW, NOTSOC)
FACILITY(FOCCON=TRACE, NOLAB, NODORMPW, NONPWRM, NOIMSXTND)
FACILITY(FOCCON=MODE=WARN, LOG=ACCESS, LOCKTIME=000)
```

- The TENV parameter is required until CA-TOP SECRET 4.3, but is no longer required as of that release.
- Each MSO user must be defined to CA-TOP SECRET.
- MSO must be executed from an APF-authorized library, and EXTSEC=YES must be specified.

Note:

If there are problems getting CA-TOP SECRET to process MSO security properly, the CA-TOP SECRET system administrator should contact Computer Associates TOP SECRET Technical Services directly.

CHAPTER 13

The MSO Console

Topics:

- Overview
- Installation
- Operation

The MSO Console is a full-screen VTAM application that enables authorized users to monitor and control an MSO address space.

Overview

When the MSO Console is installed, it provides the ability to:

- Issue any MSO operator command.
- Display all active users in the region.
- Display CPU, storage, and EXCP statistics, as well as other operational data, in real time.
- Assign priorities to users, interacting with the Resource Manager.
- Terminate user sessions.
- Display all files allocated to the address space or to a user.
- Display the last screen written to a user's terminal.
- Display all allocations of a specified data set.
- Display all services in the region.
- Display the region's log file.
- Manage, activate, and deactivate FOCUS trace facilities.
- Browse data sets allocated to the region or to individual user tasks.
- Browse traces produced by FOCUS.

The MSO Console includes a complete security facility for controlling access, with several levels of authorization defined. This security can be implemented via control statements in the MSO configuration file or by using an external security package such as RACF.

This chapter gives an overview of the installation process, including a discussion of the security facility. This is followed by a description of each of the panels and commands available in the MSO Console.

Installation

Activating the MSO Console requires additional installation steps, which are summarized in Chapter 6, *Installing MSO*. The following is an overview of the installation process.

Installation Steps

In order to activate the MSO Console, the following steps must be performed:

1. Install a VTAM applid for the console. This is described in Chapter 6, *Installing MSO*. Assign a VTAM APPLID to MSO. The Console requires a separate LU2-capable applid, distinct from the one used for VTAM access to MSO. Both applids may be defined in the same VTAM major node definition, as shown in the sample in Chapter 6, *Installing MSO*.
2. Create a console configuration file, using the sample in MSO.DATA(SSSCHCNFG) as a template. The local_lu_name record must be updated to reflect the VTAM applid that the console is to use. No other records in the sample should be changed. It is particularly important to ensure the lower-case keywords in the file are not accidentally changed to upper case.
3. Allocate ddname FOCCONS in the MSO JCL; it should point at the console configuration file. If communication traces are needed, in the event of problems activating console communications, then ddname STDOUT must be allocated as well.
4. If you will be using a VTAM session manager with the MSO Console, refer to Chapter 9, *Testing and Logging on to MSO*.
5. Decide whether you will be using internal security, where all authorizations are specified in the MSO configuration file, or external security, where all authorizations are specified in an external security system, such as RACF. MSO Console security is described in detail in Security on page 13-4. If internal security is to be used, specify

```
CONSEC = INTERNAL
```

in the MSO Configuration file, and specify the CONSOPER, CONSDTL, and CONSUSER records as appropriate.

If external security is to be used, specify

```
CONSEC = EXTERNAL
```

in the MSO Configuration file, and specify the CONSCLASS, CONSENTO, CONSENTD, and CONSENTL records if the defaults are inappropriate for your site, as described in Security on page 13-4. You must also supply the appropriate access rules in your security system in order to enable access to MSO Console. If external security is used, MSO must be running with EXTSEC = YES specified in the MSO configuration file.

The above configuration file records are described in detail in Chapter 5, *The MSO Configuration File*.

6. If MSO is running APF-authorized, and the MSO configuration file does not contain the SMFNUM record, add the following record to it:

```
SMFNUM = 0
```

This allows the display of storage, CPU, and I/O details in the Display Users panel.

7. If desired, tailor command authorizations in the SSSCONSEC table.

Security

The MSO Console provides both authorization levels of security and control of access to the MSO Console.

Authorization Levels

The MSO Console provides three levels of security: operator, detail, and logon.

- Operator authority allows a user to issue MSO operator commands, which affect the region's operation. This includes actual MSO operator commands, as well as operator-style commands from the panels.
- Detail authority allows a user to see details on other users, such as their current FOCEXEC, the current screen contents, and which files are allocated to the user.
- Logon authority allows a user to log on to the console. Logon authority enables a user to see I/O and CPU statistics, and to see which userids and services are active. Users with logon authority always have detail and operator authority over users who are logged on to MSO with the same userid as their own.

The actual association of a given function with a given level of authority may be tailored by customizing SSSCONSEC, as described below.

Access Control

Internal security is controlled by listing the userids that have access to the console in the MSO configuration file. Wildcards may be used to simplify administration. The CONSOPER, CONSDTL, and CONSUSER records are used for internal security, and are described in Chapter 5, *The MSO Configuration File*.

Note:

With internal security, CONSUSER is automatically implied when either CONSOPER or CONSDTL is specified for a userid.

External security is controlled by validating access to the MSO Console through SAF. By default, a resource class of "FACILITY" is used, and the following resource names (also known as entities) are used:

Access	Entity
OPERATOR	IBI.CONSOLE.OPERATOR
DETAIL	IBI.CONSOLE.DETAIL
LOGON	IBI.CONSOLE.LOGON

The resource class and/or resource names may be changed by specifying the CONSCCLASS, CONSENTO, CONSENTD, and CONSENTL records in the MSO configuration file, if the defaults are not adequate.

When a user attempts to log on to the MSO Console with external security, three calls are made to SAF to validate access: READ access to the LOGON rule, READ access to the DETAIL rule, and READ access to the OPERATOR rule. A user must have LOGON access in order to use the MSO Console.

Following is an example of RACF rules that might be used if the default class and resource names are in effect:

```
RDEFINE facility ibi.console.logon uacc(read)
RDEFINE facility ibi.console.detail uacc(none)
RDEFINE facility ibi.console.operator uacc(none)
PERMIT ibi.console.operator class(facility) acc(read) id(sys user1)
PERMIT ibi.console.detail class(facility) acc(read) id(sys user2)
```

Tailoring Console Authority

The MSO Console divides functions into different classes: Operator, Detail, and User. It also allows certain commands to be executed against a subtask when the userid associated with the subtask is the same as the userid of the MSO Console operator. If desired, you may change the classes to which functions are assigned. These classes are then assigned to users through the MSO configuration file.

The table SSSCONSEC contains the association between functions and their classes. Source code for SSSCONSEC is found in MSO.DATA(SSCONSEC). As shipped, the table contains the following:

Function	Operator	Detail	User	Same User
Issue operator commands	Yes	No	No	Yes
View user accounting information	Yes	Yes	Yes	Yes
View user's currently-executing command	No	Yes	No	Yes
View user's screen	No	Yes	No	Yes
View user data set allocations	No	Yes	No	Yes
Enter debugger	Yes	Yes	No	n/a
View MSO log	Yes	Yes	Yes	n/a
Data set browser	Yes	Yes	No	Yes
Control traces	Yes	No	No	No

To change these associations, edit SSSCONSEC as appropriate, and then assemble and link-edit it.

Operation

This section describes how to log on to the MSO Console, and gives a description of each command and panel available, as well as descriptions of the fields on each panel.

Logon Screen

In order to access the MSO Console logon screen, specify the MSO Console's applid (as specified in the MSO Console configuration file) in the VTAM LOGON command:

```
LOGON APPLID(consappl)
```

The logon screen displays the region's name and jobid, in order to confirm connection to the proper region.

```

      CCC      OOO      NN      NN      SSS      OOO      LL      EEEEE
      CCCCC    OOOOO    NNN      NN      SSSSS    OOOOO    LL      EEEEEE
      CC  CC   OO   OO   NN  N    NN   SS  SS  OO   OO  LL      EE
      CC      OO   OO   NN  N    NN   SSS      OO   OO  LL      EE
      CC      OO   OO   NN  N    NN   SSSS     OO   OO  LL      EEEEEE
      CC      OO   OO   NN  N    NN   SSS      OO   OO  LL      EE
      CC  CC   OO   OO   NN  N    NN   SS  SS  OO   OO  LL      EE
      CCCCC    OOOOO    NN      NNN      SSSSS    OOOOO    LLLLLLL  EEEEEE
      CCC      OOO      NN      NN      SSS      OOO      LLLLLLL  EEEEEE

                                     Information
                                     Builders,
                                     Incorporated

      Userid  ==>                                REGION MSORGN1
      Password ==>                                JOBID  JOB07626

                                     Press PF3/PF15 to EXIT

```

The user enters a valid, authorized, userid and password on this screen. After MSO validates the logon, the Primary Option Menu appears.

Primary Option Menu

```
----- CONSOLE PRIMARY OPTION MENU -----
OPTION  ===>

Type an option and press ENTER key           Region  MSORGN1
                                              Jobid   JOB07626
                                              Userid  MSOUSR
                                              ARTHUR DENT

DU      - Display all active users in this region
DS      - Display all servers in this region
LOG     - View region log file
TRACE  - Trace handling component

/command - Execute operator command

HELP    - Bring console HELP (you may press PF1 any time)

Press PF3/PF15 to EXIT
```

The Primary Option Menu is displayed upon logon to the MSO Console and when pressing PF3 to exit from any of its sub-panels. The job name and JES jobid of the MSO address space are displayed on the Primary Option Menu. In addition, the userid and user name that were used to log on to MSO Console are displayed.

The Primary Option Menu lists the commands that are available in the MSO Console. All of these commands are accessible from any command line on any panel.

Operator Commands

A user with Operator authority may issue any MSO operator command from any command line in the MSO Console. The command must be preceded by a /, and the prefix of F *jobname*, which would be required from the MVS system console, must be omitted. For example, the MVS operator command

```
F MSO ,PURGE LOGONID=ZAPHOD
```

becomes

```
/PURGE LOGONID=ZAPHOD
```

A complete list of operator commands appears in Chapter 11, *MSO Operator Commands*.

Display User

The Display User panel is displayed after the DU command is entered on the command line. The Display User panel is used to display and control all of the users who are logged on to MSO. It forms the core of the MSO console, and is generally the panel that is used the most. The display shows one line for the region as a whole, and one line for each connected user.

The Display User display spans a width of two screens. PF7/PF8 and PF10/PF11 may be used for scrolling. In addition, the WHOHAS command, described in *WHOHAS* on page 13-12, uses a third screen, to the right of the first two, to display its output.

The Display User panel is refreshed whenever the ENTER key is pressed.

```
PHMSDR1 J080651B----- .CONSOLE DISPLAY USERS PANEL. -----Line:001(002) Col:001
COMMAND ==>

C Logon_ID Server   PRG CPU-TM   CPU% EXCP-T EXCP-S STG-ABOVE STG-BELOW TCADDR
*REGION*
QCSOJK FOCUS       7   0.23   0.00  22   0.00   10239    28 6ACE88
```

The Display user panel contains the following fields:

- C - Command This field may be used to specify a command which is to be issued against a particular user, or, if issued on the first line, for the region as a whole. The commands which are valid are:
- C - Cancel or P - Purge This command cancels a user's MSO session. It may be entered for multiple users on a single display.
- ? - Display screen This command displays the contents of the last screen written to a user's terminal. If the terminal being used has a smaller screen size than the user's terminal, then PF7/PF8 and PF10/PF11 may be used to scroll the display. PF3 is used to return to the Display Users panel.

W - WHOHAS	This command displays all of the files that are allocated to a user's session. When issued on the first line, for the MSO region, it shows the files that are globally allocated to the region. The WHOHAS displays are further discussed on page 13-12.
T - Traces	This command moves you to the Trace panel, with the selection fields filled in to restrict the display to traces associated with this particular user's session.
Logon_ID	This field is the userid under which the user logged on.
Server	This field indicates which service the user logged on to. For normal MSO FOCUS users, it will be "FOCUS".
PSB	The name of the IMS PSB scheduled for each TCB that is accessing IMS. May be up to 8 characters in length.
PRG	This field contains the MRM priority group associated with this user. It may be changed by overtyping it with the desired value. (See Chapter 3, <i>MSO Technical Description</i> .)
CPU-TM	This field contains the total amount of CPU time, in seconds, consumed in the user's current logon session.
CPU%	This field contains the percent of a single processor that the user consumed since the ENTER key was pressed by the Console user.
EXCP-T	This field contains the total number of EXCP operations that were performed in the user's current logon session. Certain EXCP operations, such as those for program fetch, are not included in this number.
EXCP-S	This field contains the average number of EXCP operations per second which have been initiated by this user since the ENTER key was pressed by the Console user.
STG-ABOVE	This field contains the current amount of above-the-line virtual memory that is in use by this user. The number displayed is in Kbytes.
STG-BELOW	This field contains the current amount of below-the-line virtual memory that is in use by this user. The number displayed is in Kbytes.
TCBADR	This field contains the address of the MVS Task Control Block (TCB) for this user's subtask.

```

PRESM001 J0006510----- .CONSOLE DISPLAY USERS PANEL. -----Line:001(002) Col:061
COMMAND ==>
C Logon_ID   TCBADR Secy_ID   Term_adr  HiperStg  Current_statement
*REGION*   PUTLEVEL:   (20011109)  <|
QCS0JK     6ACEBB QCS0JK   VTAK227   SHELPROF 000000 SHELPROF

```

When scrolled to the right by pressing PF11, the Display User panel contains the following fields:

C - Command	This is the same field as on the left panel.
Logon_ID	This is the same field as on the left panel.
TCBADR	This is the same field as on the left panel.
Secy_ID - Security ID	This field contains the ID used by MSO as the SAF security ID. This ID is placed in the ACEE, and is used for all security validation for the user. In most sites, this is identical to Logon_ID. (See Chapter 13, <i>MSO Security</i>).
Term_adr	This field contains the address of the user's terminal.
Current_statement	This field displays the last statement executed by the user. It has three components: the FOCEXEC name, the line number within the FOCEXEC, and the first characters of the actual statement. Sometimes one or more of these fields may not be available for display.

For the first line, which displays the values for the region as a whole, Current_statement contains information on the release date and level of MSO.

WHOHAS

The WHOHAS command may be used to display the global file allocations for the MSO address space, the files that are allocated to a particular user's session, or all users who have a specific data set allocated, as described below. The WHOHAS displays are refreshed whenever the ENTER key is pressed.

```

PMSMSURI JOB07500----- .CONSOLE DISPLAY USERS PANEL. -----Line:001(001) Col:001
COMMAND --->

-

+-----DATA SET ALLOCATIONS-----+
//STCPLID DD DSN=PGMDDP.WDMDAT.LOAD,DISP=SHR
//TASKLIB DD DSN=WIBFDC.R70XFDC.FOCLIB.LOAD,DISP=SHR
//USCLIB DD DSN=WIBFDC.R70XFDC.FUSCLIB.LOAD,DISP=SHR
//ERRORS DD DSN=WIBFDC.R70XFDC.ERRORS.DATA,DISP=SHR
//MSDPROF DD DSN=PMSSCS.MSO.PROFILE,DISP=SHR
//MSGET DD DSN=PMSSCS.MSGET,DISP=SHR
//MSPUT DD DSN=PMSSCS.MSOPUT,DISP=SHR
//MASTER DD DSN=PMSSCS.MASTER.DATA,DISP=SHR
// DD DSN=WIBFDC.R70XFDC.MASTER.DATA,DISP=SHR
//FOCEXEC DD DSN=PMSSCS.FOCEXEC.DATA,DISP=SHR
// DD DSN=WIBFDC.R70XFDC.FOCEXEC.DATA,DISP=SHR
//CAR DD DSN=PMSSCS.CAR.FOCUS,DISP=SHR
//EMPLOYCC DD DSN=PMSSCS.EMPLOYCC.FOCUS,DISP=SHR
//SYSPRINT DD SYSOUT=*
//MSDPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=D
//MSDTRACE DD SYSOUT=*

```

The region-level WHOHAS display is shown when the W prefix command is entered against the *REGION* userid in the Display Users panel. It shows the files that are allocated globally to the MSO region. These files are either used by MSO itself, or files that are available, subject to security rules, to all users of the MSO region.

You may use the region-level WHOHAS display to invoke the data set browser for data sets that may be browsed from the MSO Console. There is an input field to the left of each eligible ddname. You may enter the S command in this field to browse the data set.

```

MSORGN1 JOB07626-- .CONSOLE DISPLAY USERS PANEL. --Line:001(004) Col:001
COMMAND ===>

+-----DATA SET ALLOCATIONS-----+
| //STDOUT DD DUMMY /*TR000001|
| //OFFLINE DD SYSOUT=A /*OF001001|
| //FOCSORT DD SPACE=(TRK,(5,5)),DISP=NEW /*FO001001|
| //FOCEXEC DD DSN=MSOUSR.FOCEXEC.DATA,DISP=SHR /*FO002001|
| //ATT DD DSN=MSOUSR.ATT.DATA,DISP=SHR /*AT001001|
| //MASTER DD DSN=MSOUSR.MASTER.DATA,DISP=SHR /*MA001001|
| |

```

The task-level WHOHAS display is shown when the W prefix command is entered against a userid in the Display Users panel. It shows the files that are allocated to that user's task. The right side of the display shows the real (translated) ddnames that were used to allocate the files. These translated ddnames are required if the DYNAM operator command is to be used against these files. (This does not affect DYNAM commands issued by the user, which continue to use the untranslated ddnames.) See Chapter 15, *Site-Specific Customizations*, for a discussion of ddname translation.

You may use the task-level WHOHAS display to invoke the data set browser for data sets that may be browsed from the MSO Console. There is an input field to the left of each eligible ddname. You may enter the S command in this field to browse the data set.

```
(PMSMSDR1 JOB07560----- .CONSOLE DISPLAY USERS PANEL. -----Line:001(002) Col:153
COMMAND ==>)

C Logon_ID Runcount MVSD0 USER-DD OPN DDs QueryID1 QueryID2 PSB
 *REGION*
 QCS0JK
```

The data set-specific WHOHAS display is generated by entering the command

`WHOHAS data set name`

from the Display User panel. This command displays information on all allocations of a specific file in the MSO region. The display contains the following columns:

C - Command	This field is the same as on the Display Users panel. In addition, the W command takes on a new meaning, as described under MVS DD, below.
Logon_ID	This field is the same as on the left Display Users panel.
Runcount	The number of attempts to connect to the central PDM. Runcount corresponds to the runcount specified in the service block.
MVSD0	This field contains the real, translated, ddname, which MSO used to allocate the file. If the file is allocated more than once by one user, this field will contain *MORE..., and the W command must be entered in the Command field to display detail on the allocations.
USER-DD	This field contains the untranslated ddname, which the user specified to allocate the file.
OPN	This field shows a count of how many open DCBs exist for this file.
DDs	This field shows a count of how many different allocations exist for this file by this user. If greater than 1, W must be entered in the Command field to display the allocations.
QueryID1	This field is blank for normal MSO usage.
QueryID2	This field is blank for normal MSO usage.
PSB	The name of the PSB being accessed (only applies to IMS).

Display Servers

```
MSORGN1  JOB07626-----  CONSOLE Display Servers panel -----Line:001(001)
COMMAND ===>

C Server      Program  PRG  Servinit  Timeout  Idlelim  Max  Runcount  Act
FOCUS        FOCUS    07              008          003
```

The Display Servers panel is displayed by entering the DS command on the command line. The Display Servers panel enables display and control of services, as defined in service blocks in the configuration file, from the MSO Console. Operator authority is required for all actions from this panel, except for the ability to view the list of servers.

All of the columns described below, except for Act, may be changed by overtyping them with the desired value.

The following columns are displayed in this panel:

C - Command

This field may be used to specify a command that is to be issued against a particular service. The commands that are valid are:

Q - Quiesce Server	Used to prohibit new logons to the specified service.
E - Enable Server	Used to allow a QUIESCED service to accept new logons.
S - Start Server	Starts another task under the specified service. It is generally not used for MSO.
D - Delete Inactive Server	Deactivates a service. In doing so, it changes the service name to *OFFLINE.
I - Insert Server	Adds a new service to the list. You must then supply the service's parameters by overtyping the appropriate columns on the display. A maximum of 20 services may be defined. A service added using Insert Server is identical to one defined in a service block in the configuration file.
T - Traces	Moves you to the Trace panel, with the selection fields filled in to restrict the display to traces associated with this particular service.
Server	Contains the name of the service as defined in the service block in the MSO configuration file. Please note that changing this field may prevent access to the service.

Program	Contains the name of the program that is executed by this service. It corresponds to the PROGRAM specification in the service block, and is always FOCUS for MSO.
PRG	Contains the default MRM priority group for users of this service. It corresponds to the PRTYGROUP specification in the service block.
Servinit	This field is not used for MSO. It corresponds to the SERVINIT specification in the service block.
Timeout	This field contains the timeout value for this service. It corresponds to the TIMEOUT specification in the service block.
Idlelim	This field contains the idlelimit value for this service. It corresponds to the IDLELIM specification in the service block.
Max	This field contains the maximum number of tasks that may be running for this service. It corresponds to the MAXIMUM specification in the service block.
Runcount	This field is not used for MSO. It corresponds to the RUNCOUNT specification in the service block.
Act	This field contains the actual number of tasks that are currently running for this service. It is initially equal to the NUMBER_READY specification in the service block, and goes up as the number of logged-on users increases.

The Display Users display is refreshed whenever the ENTER key is pressed.

Log Viewer

The log viewer is entered by issuing the LOG command on the command line. The log viewer allows you to view the MSO log file, which is written to ddname MSOPRINT, from the MSO Console. The most recent 48 lines are available for viewing. PF7/PF8 and PF10/PF11 may be used for scrolling, as needed. Operator authority is required for access to the log viewer.

```

PMSOR1 JOB06510----- .CONSOLE LOG VIEWER. -----
COMMAND ==>

11/05/2001 10.41.06 (MSD13369) MONITORING IS NOT ACTIVE, DEFAULT INTERVAL = 180
11/05/2001 10.41.06 (MSD13184) MSD CONSOLE IS ACTIVE, SECURITY IS INTERNAL
11/05/2001 10.42.10 (MSD13021) LOGON ID: PMSSCS SERVICE: FOCUS TCB:
11/05/2001 10.43.03 (MSD13025) TASK ENDED SERVICE: FOCUS TCB:
11/05/2001 10.43.03 (MSD13024) TASK STARTED SERVICE: FOCUS TCB:
11/05/2001 10.52.32 (MSD13021) LOGON ID: PMSBED SERVICE: FOCUS TCB:
11/05/2001 10.53.07 (MSD13025) TASK ENDED SERVICE: FOCUS TCB:
11/05/2001 10.53.07 (MSD13024) TASK STARTED SERVICE: FOCUS TCB:
11/05/2001 11.10.29 (MSD13021) LOGON ID: PMSBED SERVICE: FOCUS TCB:
11/05/2001 11.11.07 (MSD13025) TASK ENDED SERVICE: FOCUS TCB:
11/05/2001 11.11.07 (MSD13024) TASK STARTED SERVICE: FOCUS TCB:
11/12/2001 10.31.37 (MSD13075) SECURITY FAILED, USERID=QCSBJK RC=0000 RETURN=
11/12/2001 10.31.52 (MSD13101) OPERATOR QCSBJK LOGGED ON TO THE MSD CONSOLE [
    
```

Trace Handling

```

MSORGN1 JOB07626----- CONSOLE TRACE panel - Line:001(013)
COMMAND ==>

TR S TY ALL NAME * SERV * USER * TCB * LEV *
C TRACE SERVICE USERID TCBADR TRACEDD_ DDNAME TYP COND LEVEL
FSTRACE5 FOCUS MSOUSR 7CA680 TR000013 FSTRACE5 ACT 4
STDOUTFX FOCUS MSOUSR 7CA680 TR000004 STDOUTFX ACT 0
HFTRACE FOCUS MSOUSR 7CA680 TR000001 STDOUT ACT 2048
DYNTRACE FOCUS MSOUSR 7CA680 TR000003 STDOUTDY ACT 0
FSTRACE FOCUS MSOUSR 7CA680 TR000002 FSTRACE ACT 0
STDOUTFX FOCUS MSOUSR2 7CC628 TR000008 STDOUTFX ACT 0
HFTRACE FOCUS MSOUSR2 7CC628 TR000005 STDOUT ACT 2048
DYNTRACE FOCUS MSOUSR2 7CC628 TR000007 STDOUTDY ACT 0
FSTRACE FOCUS MSOUSR2 7CC628 TR000006 FSTRACE ACT 0
STDOUTFX FOCUS MSOUSR3 7CC310 TR000012 STDOUTFX ACT 0
HFTRACE FOCUS MSOUSR3 7CC310 TR000009 STDOUT ACT 2048
DYNTRACE FOCUS MSOUSR3 7CC310 TR000011 STDOUTDY ACT 0
FSTRACE FOCUS MSOUSR3 7CC310 TR000010 FSTRACE ACT 0
    
```

The TRACE panel is displayed when the TRACE command is entered on the command line. The TRACE panel allows activating and managing of the various traces which IBI products and options may generate. Specific information on traces to be generated is provided by Information Builders support personnel. Operator authority is required in order to use the trace facility.

Trace Panel Commands

The INDEX command may be entered in the COMMAND field to display the trace index.

Trace Panel Control Fields

Use the trace panel control fields to control the format of the trace display and to select that traces are to be displayed.

TR	TR can be set to S or F. S selects the short, single-line format display. On a 3270 Model 5, this display extends past 80 columns. F selects the full, multi-line format display.
TY	Use TY to restrict the trace types displayed. ACT displays active traces while DEF displays deferred traces. Specifying * causes all trace types to be displayed.
NAME	Use NAME to restrict the display to traces with the specified trace name. Wildcards using * may be used.
SERV	Use SERV to restrict the display to traces in tasks running under the specified service. Wildcards using * may be used.
USER	Use USER to restrict the display to traces in tasks running under the specified userid. Wildcards using * may be used.
TCB	Use TCB to restrict the display to traces in the specified TCB. Wildcards using * may be used.
LEV	Use LEV to restrict the display to traces running with the specified level. Specify * to get all levels.

Trace Detail Lines

The following columns are displayed for each trace:

C	May be used to specify a command to be issued for the specified trace. The commands which are valid are: <ul style="list-style-type: none"> • S - Start Trace This command starts the specified trace. • P - Stop Trace This command stops the specified trace.
TRACE	Contains the name of the trace.
SERVICE	Contains the name of the service under which the task is running.
USERID	Contains the userid associated with the task.
TCBADDR	Contains the address of the task's TCB.
TRACEDD	Contains the real (translated) ddname to which trace records are written.

- DDNAME Contains the untranslated ddname that is used for writing trace records.
- TYP Contains one of the following values:
- ACT - Active
An Active trace is currently running, although it only produces output if it is set to an appropriate level.
 - DEF - Deferred
A Deferred trace is not currently running; rather, it is waiting for a trace matching its selection criteria to be started.
- COND This field contains the value COND if the trace is a conditional trace.
- LEVEL This field contains the current level for the trace.

The Trace Index

The trace index provides a list of all traces that have been activated for the life of the MSO address space. Active traces are highlighted.

```
MSORGN1 JOB07550----- .CONSOLE DATA SET BROWSER. -----Line 00001
COMMAND ==>

..Date.. ..Time..JobName TracName Server User TCADDR DDname RealNam
11/17/94 10.43.17MSORGN1 DYNTRACE FOCUS MSOUSR 7BEE88 STDOUTDY TR000003
11/17/94 10.44.23MSORGN1 DYNTRACE FOCUS MSOUSR2 7BF3F0 STDOUTDY TR000007
```

If traces are being recorded in a PDSE or in sequential data sets, the trace index allows use of the data set browser to display traces. The data set browser may not be used for active traces, only for closed ones, and it may not be used if traces are being recorded in SYSOUT files.

The trace index contains the following columns:

- | | |
|---------------------------|--|
| (unlabeled command field) | You may enter S to the left of the trace information to select the trace for browsing. This field is only available if the trace is eligible for browsing. |
| Date | Date the trace was activated. |
| Time | Time the trace was activated. |
| JobName | Name of the MSO address space. |
| TracName | Name of the trace. |
| Server | Service under which the task is running. |
| User | Contains the userid associated with the task. |

TCBADR	Contains the address of the task's TCB.
Ddname	ddname that FOCUS is using to write the trace.
RealNam	Real (translated) ddname that MSO is using for the trace.

Data Set Browser

The data set browser allows you to view data sets that are allocated to the MSO address space.

You can enter the browser by typing the command S next to the entry for a data set in:

- The region-level WHOHAS display.
- The task-level WHOHAS display.
- The TRACE INDEX display.

Data Set Browser Member List

If you select a PDS, the following screen is displayed, listing the members of the PDS:

```
MSORGN1  JOB04804----- .CONSOLE DATA SET BROWSER. -----Line 00001
COMMAND ==>

DSN=MSOUSR.FOCEXEC.DATA  VOL=SER=UXYZ123 RECFM=FB
  ABEND                01.00 93/02/24  93/02/24 12:28      1 MSOUSR
  ASDF                  01.00 94/02/01  94/02/01 11:33      3 MSOUSR
  ASDFG                 01.00 94/02/01  94/02/01 11:33      3 MSOUSR
  ATT                   01.01 93/03/26  93/03/31 14:42      3 MSOUSR
  ATTDDEL              01.00 93/03/26  93/03/26 14:16     20 MSOUSR
  ATTMOD               01.07 93/03/25  93/03/26 12:18     26 MSOUSR
  ATTMOD1              01.00 93/03/26  93/03/26 12:14     25 MSOUSR
  ATTRPT               01.02 93/03/25  93/03/25 16:41     21 MSOUSR
  A2580010             01.01 93/12/16  93/12/16 12:26      5 MSOUSR
  CC                   01.01 93/11/24  93/11/24 10:19     11 MSOUSR
  CC2                  01.01 93/11/24  93/11/24 10:20      9 MSOUSR
  CONCAT               01.01 93/11/03  93/11/03 15:27      3 MSOUSR
  CURRDATE
  DB2                  01.00 94/01/05  94/01/05 18:04     21 PMSAJG
  DB2AUTH              01.00 94/01/31  94/01/31 11:28      2 PMSAJG
  DEBUG
  DFSORT               01.00 94/01/27  94/01/27 16:14      6 PMSAJG
  DYNTEST              01.09 93/05/05  93/05/05 12:16      3 PMSAJG
  FASTLOAD             01.02 93/05/14  93/09/16 12:21     13 PMSAJG
  FB                   01.00 94/03/11  94/03/11 11:52      2 PMSAJG
```

The member list shows the ISPF statistics for members, if they are available. You may use PF7 and PF8 to scroll through the list, and PF3 to exit. Enter S to the left of a member name to select the member for browsing.

Data Set Browser Display

The data set browser display screen shows the contents of a sequential data set or a PDS member. PF7/PF8 and PF10/PF11 may be used for scrolling, and PF3 may be used to exit from the browser.

```
MSORGN1  JOB04804----- .CONSOLE DATA SET BROWSER. -----Line 000001
COMMAND ===>

DSN=MSOUSR.FOCEXEC.DATA(OFFL)  VOL=SER=XYZ123 RECFM=FB
-*OFFLINE CLOSE
DYNAM FREE FILE OFFLINE
DYNAM ALLOC FILE OFFLINE SYSOUT A DEST SYS2B.MSOUSR
OFFLINE
```

In addition, the following commands are available:

HEX ON

Displays both EBCDIC and hexadecimal representations of the file. HEX may be used as a synonym for HEX ON.

HEX OFF

Displays the file in EBCDIC.

TOP

Go to the top of the file.

Help Facility

Online help information is provided for the MSO Console. This information is accessible by entering the HELP command, or by pressing PF1 at any time. Following are samples of the help screens. PF7/PF8 may be used to scroll through the help screens; PF3 is used to return to the MSO Console.

```
----- CONSOLE PRIMARY OPTION MENU -----
+----- CONSOLE HELP-----+
e:01 of 08 |
|
|          CONSOLE PANELS
|
|  DU      - DISPLAY USER panel      (active users info)
|  DS      - DISPLAY SERVER panel    (servers info   )
|  LOG     - DISPLAY LOG panel       (log file browser)
|  PARMS   - GLOBAL PARAMETERS panel (_in_development)
|  DEBUG   - DEBUGGER panel         (memory browser )
|  TRACE   - TRACE panel             (trace handling )
|  HELP    - HELP                   (help screens  )
|  /cmd    - Issue operator command (eg. /QUIESCE  )
```

PF3 - Exit; PF7/PF19 - Previous; PF8/PF20 - Next


```
----- CONSOLE PRIMARY OPTION MENU -----
+----- CONSOLE HELP-----+
                                         Page:04 of 08
                                         DISPLAY LOG

LOG          DISPLAY LOG panel allows you to browse and
              scroll back and forth the last few log file
              buffers. PF7/PF8/PF10/PF11

PF3 - Exit;   PF7/PF19 - Previous;   PF8/PF20 - Next
```

```
----- CONSOLE PRIMARY OPTION MENU -----
+----- CONSOLE HELP-----+
                                         Page:06 of 08
                                         DISPLAY TRACE

DISPLAY TRACE  DISPLAY TRACE panel makes it possible to con-
                trol traces, switch trace levels, etc.
                Primary command 'INDEX' shows the trace index
                file.
                Line trace commands are:
S              Start the trace (switch level to the default)
P              Stop the trace (switch level to zero)

PF3 - Exit;   PF7/PF19 - Previous;   PF8/PF20 - Next
```


On Demand

These commands can be issued from MSO/MVS console only. To display statistics based on CONFIG file specifications, type:

```
F servername,DISPLAY STATS
```

To display all statistics regardless of CONFIG file specifications, type:

```
F servername,DISPLAY STATS,ALL
```

Short On Storage

```
SOS_PERCENT = {nn|0}
```

where:

nn

Is an integer representing a percentage between 0 and 99 percent. The default is 0.

Short on Storage specifies the percentage of storage utilization, which when exceeded, will cause a warning message to be issued via non-scrollable messages to WTO and MSOPRINT. The message destination can be altered with the Enhanced Message Routing Facility.

Enhanced message routing allows MSO installations to control the printing/listing destination of all MSO messages. MSO installations can choose to route messages to the operator's console (WTO) or MSOPRINT or both.

Enhanced message routing is implemented by adding a control character to the MSO messages found in the members MSOERR1 and CMSO1ERR of the PDS prefix.ERRORS.DATA.

Standard MSO error messages look like this:

```
00000(MSO12345) TEXT
```

Enhanced routing replaces the blank field in position 16 (before text) with:

- b,B to route the message to both the console and MSOPRINT.
- p,P to route the message to MSOPRINT only.
- w,W to route the message to the console only.
- _ Blank defers to the default output destination.

If storage utilization falls below the defined `SOS_PERCENT`, a message is issued stating that the storage constraint has been relieved and the WTO message is deleted (unless suppressed by the Enhanced Message Routing Facility). The `SOS_PERCENT` check, if active, is made at every `MRM_INTERVAL` interval (MSO Resource Manager Interval).

Specifying 0 disables the SOS monitoring facility. SOS monitoring may be enabled or disabled dynamically by setting `SOS_PERCENT` to a non-0 or 0 value, respectively.

APF authorization is required.

Monitoring

These commands can be issued from either the MSO or OS/390 consoles.

Using MSO console:

```
MONITOR = {ON|OFF}
```

where:

`ON`

Turns monitoring on.

`OFF`

Turns monitoring off.

Using OS/390 console:

```
F servname,MONITOR = {ON|OFF}
```

where:

`ON`

Turns monitoring on.

`OFF`

Turns monitoring off.

`MONITOR OFF` resets current setting for `MN_USER`, and for `MN_INTERVAL`, etc. to original values in the `CONFIG` file. APF authorization is required.

Using MSO console:

```
MN_INTERVAL = nnn
```

where:

`nnn`

Is an integer indicating the monitoring interval in seconds. The minimum is 30 and the maximum is 3600.

Using OS/390 console:

```
F servname,MN_INTERVAL = nnn
```

where:

nnn

Specifies the interval at which to perform monitoring (when active). To turn on monitoring, set MONITOR=ON. Monitoring is then performed at integer multiples of the MRM_INTERVAL (MSO Resource Manager Interval).

Default: 1800 (30 minutes).

APF authorization is required.

Using MSO console:

```
MN_STORAGE = {OFF|ON}
```

where:

OFF

Does not include storage statistics if monitoring is active. OFF is the default.

ON

Includes storage statistics if monitoring is active.

Using OS/390 console:

```
F servname,MN_STORAGE = {OFF|ON}
```

where:

OFF

Does not include storage statistics if monitoring is active. OFF is the default.

ON

Includes storage statistics if monitoring is active.

To turn monitoring on, set MONITOR=ON.

APF authorization is required.

Using MSO console:

`MN_REGION = {OFF|ON}`

where:

`OFF`

Specifies that region statistics are not required when monitoring is active. OFF is the default.

`ON`

Specifies that region statistics are required when monitoring is active.

Using OS/390 console:

`F servname,MN_REGION = {OFF|ON}`

where:

`OFF`

Specifies that region statistics are not required when monitoring is active. OFF is the default.

`ON`

Includes storage statistics if monitoring is active.

To turn monitoring on, set MONITOR=ON.

APF authorization is required.

Using MSO console:

`MN_USERS = {OFF|ALL}`

where:

`OFF`

Curtails monitoring of userids.

`ALL`

Produces usage statistics for all logged on userids.

Using OS/390 console:

`F servname,MN_USERS = {OFF|ALL}`

where:

`OFF`

Curtails monitoring of userids.

`ALL`

Produces usage statistics for all logged on userids.

To turn monitoring on, set MONITOR=ON.

APF authorization is required.

Messages for On Demand and Monitoring

On demand messages are issued according to the CONFIG FILE values, unless the user issues

```
display stats,ALL
```

Monitoring messages are issued according to the CONFIG FILE values as (currently) modified by (MSO/MVS) console commands.

```
(MSO13373) MONITORING STATISTICS FOR xxxxxxxx (JOBxxxxxx)
(MSO13373) ON DEMAND STATISTICS FOR xxxxxxxx (JOBxxxxxx)
(MSO13374) STATISTICS FOR STORAGE
(MSO13372) BELOW: REGION =   xxxK  IN USE =   xxxK  USED=xxx%  SOS=xx%
(MSO13372) ABOVE: REGION = xxxxxK  IN USE = xxxxxK  USED=xxx%  SOS=xx%
(MSO13375) STATISTICS FOR REGION
(MSO13383) CPU_T=xxxxx.xx  CPU%=xx.xx  EXCP_T=xxxxxxxxx  EXCP/S=xxxx
(MSO13384) LU2_NAME=xxxxxxxx  ACTIVE USERS=xxxx  APPLGROUP=xxxxxxxx
(MSO13377) STATISTICS FOR USERS
```

The following messages appear one time per logged on user.

```
(MSO13386) STATISTICS FOR LOGON ID=xxxxxx  LOGON: yy.ddd  hh:mm
(MSO13387) SERVER=xxxxxxxx  TCBADR=xxxxxx  TERMADR=xxxxxxxx
(MSO13383) CPU_T=xxxxx.xx  CPU%=xx.xx  EXCP_T=xxxxxxxxx  EXCP/S=xxxx
(MSO13388) STG_BELOW=xxxxxxK  STG_ABOVE=xxxxxxK
(MSO13389) WAIT_TIME=xxxxx SECONDS  (SINCE LAST "ENTER")
(MSO13392) END OF DISPLAY
```

Messages for Short on Storage

Only the address space/storage messages are issued, along with a critical shortage or critical shortage relieve message

```
(MSO13371) CRITICAL STORAGE SITUATION BELOW/ABOVE THE LINE FOR JOB
xxxxxxxx (JOBnnnnn)
```

```
(MSO13378) CRITICAL STORAGE SITUATION BELOW/ABOVE THE LINE RELIEVED FOR
JOB xxxxxxxx (JOBnnnnn)
```

followed by MSO13372 giving details.

Messages for Shutdown

```
(MSO13373) SHUTDOWN STATISTICS FOR xxxxxxxx (JOBxxxxxx)
(MSO13374) STATISTICS FOR STORAGE
(MSO13380) BELOW: MAX_USED = xxxxxK  OUT_OF xxxxxK  USED=xxx%
(MSO13380) ABOVE: MAX_USED = xxxxxK  OUT_OF xxxxxK  USED=xxx%
(MSO13375) STATISTICS FOR REGION
(MSO13381) MAX_CPU%=xx  MAX_EXCP/SEC=xxxx  (PER MRM_INTERVAL)
(MSO13385) USERS FAILED: BAD_PSWD=xxxxxx  BAD_ACCT=xxxxxx  NO_STORAGE=x
(MSO13398) USERS FAILED: NO_SERVICE=xxxxxx
(MSO13382) SERVER WAS ACTIVE FOR xx DAY(S), xx HOUR(S) and xx.xx MINUTE(
(MSO13376) STATISTICS FOR SERVICES
```

The following messages appear one time per service

```
(MSO13390) SERVICE=xxxxxxxx PROGRAM=xxxxxxxx MAXIMUM=xxxx
(MSO13391) MAX_ACTUALLY_USED=xxxx TOTAL_TASKS=xxxxxxxx
(MSO13394) USERS FAILED: EXCEEDED_MAXIMUM=xxxxxx QUIESCED_SERV=xxxxxx
(MSO13395) USERS FAILED: CNCLD=xxxxxx ABENDED=xxxxxx
(MSO13397) USERS FAILED: IDLELIM=xxxxxx TIMEOUT=xxxxxx
(MSO13396) CPU_T=xxxxxx CPU%=xx.xx
(MSO13392) END OF DISPLAY
```

MSO Dynamic VTAM Re-configuration

MSO installations can re-configure the VTAM LU2 applid used for MSO FOCUS access. This allows them to dynamically switch to a newly specified LU2 applid without recycling the MSO region.

Usage

The SET LU2_NAME command is issued at the MSO console or via the modify command at the MVS system console. This conditions the MSO region to use the new LU2 applid when the RESTART LU2_NAME command is issued from MSO console or the OS/390 system console. When the RESTART LU2_NAME command is issued, current MSO FOCUS sessions are terminated. The newly specified LU2_NAME is then activated for MSO FOCUS logons. All sessions connected to the MSO region from TSO or CICS are unaffected.

It is recommended that all MSO console commands be issued while viewing the MSO LOG. The command's effect is then immediately visible by pressing ENTER.

Conditioning MSO to use the VTAM LU2 applid:

MSO Console

```
SET LU2_NAME = applid_name
```

OS/390 System Console

```
F mso_jobname,SET LU2_NAME = applid_name
```

Performing the Dynamic Switch to the New applid:

MSO Console

```
RESTART LU2_NAME
```

OS/390 System Console

```
F mso_jobname,RESTART LU2_NAME
```

Note:

The keyword LU2_APPLID is synonymous with LU2_NAME and the two may be used interchangeably.

Special Considerations

The VTAM LU2 applid name specified in the SET LU2_NAME command must be an active LU2 applid that is defined with the following characteristics in SYS1.VTAMLST:

```
FOCUS  APPL  AUTH=(VPACE,PASS),EAS=10,PARSESS=YES
```

Note:

The PASS specification is only required if the Load Balancing feature of MSO is used.

Error Messages

```
MSO13360  INVALID KEYWORD IN SET COMMAND
```

The keyword in the SET command is unknown. Please check the spelling of the keyword and be sure our documentation matches the version of the program you are using.

```
MSO13361  INVALID OPERAND IN SET COMMAND
```

A valid SET command keyword has an invalid operand. Please check the spelling of the operand and be sure your documentation matches the version of the program you are using.

```
MSO13362  LU2_APPLID IS NOW SET TO %1
```

The specified new value for LU2_APPLID is now in effect.

CHAPTER 14

Common Installation Problems and Solutions

Topics:

- Diagnosing Installation Problems
- Diagnosing Storage Problems
- Diagnosing System Failures
- Applying Maintenance

This chapter describes common problems and suggested solutions you may encounter when installing MSO.

Diagnosing Installation Problems

If the FOCUS Terminal Operator Environment screen does not appear, the MSO installer should follow the steps below to help determine what went wrong:

1. If the MSO job stays up, bring it down (using MSOKILL or by using the operator command P *jobname* or by canceling it). Look at the job output, and make sure that MSO initialization commenced by looking at the MSO log messages. Usually, MSOPRINT is allocated to SYSOUT=* and thus log messages appear on the job output. After the messages appear, the initial FOCUS loads occur. When these are done, the message that the initial loads are complete is put into the log. If these two messages appear, MSO has initialized without error.
2. Bring up MSO without authorization. Copying the FOCLIB library into a non-authorized data set does this. Without authorization, the MSO users inherit the authorization of the MSO job. If MSO now works, there is a problem with the security system at the site.
3. For VTAM sites where the IBI logon screen does not appear, and yet MSO initialized properly, cut a VTAM trace of all messages going to the MSO APPLID and call the Customer Support Hotline at 1-800-736-6130.
4. If TSO is available, bring up MSO again and try logging on as a TSO user. The message that a user is logging on should appear in the log. If it does not, then the communications data sets for the TSO user and the MSO job are probably different.

If the message does appear, but the Terminal Operator Environment screen for the TSO user does not, then both MSO and the user may not have correct access to the communications data sets. Verify that MSO has read/write access to both communications data sets (this can be done by attempting to edit the communications data sets from the MSO ID). All MSO users must have read access to these data sets, as well.

If the TSO Terminal Operator Environment screen appears and the CICS Terminal Operator Environment screen does not, then the problem is associated with the definitions of the CICS programs and transactions. Make sure the CICS user has security access to all the MSO transactions and check the CICS parts of the installation procedure again.

A common error with the installation of the CICS transactions is incorrectly linking the MSO programs with the CICS command level stubs. These errors cause ASRA and AEY8 abends as soon as the MSO transaction is initiated.

5. If the CICS region is not large enough, the MSO transactions may not work properly.

Also, make sure that OSCOR is large enough if applicable. MSO needs at least 165K of OSCOR over existing needs.

6. If a large number of MSO users log on, the 32M OS390 virtual memory default limit (in the IBM IEFUSI exit) may cause an S106 abend. In this case, the default limit must be increased by MVS systems personnel.

Diagnosing Storage Problems

Storage problems typically manifest themselves via the MSO13144/MSO13145 messages or by producing storage-related system abends. The following points should be considered in diagnosing storage problems:

1. Is the address space large enough? In Chapter 4, *MSO Capacity Planning and Performance*, the requirements of virtual storage for MSO are discussed. As a starting point, you might want to allocate all available below-the-line storage and 4 megabytes per user above the 16MB line.
2. Is MVS allocating the requested storage? Some sites restrict virtual storage through the IEFUSI exit in MVS. The MSO13142 message, issued at MSO startup, will confirm the amount of storage available to MSO.
3. Are appropriate values specified for STORAGEABOVE and STORAGEBELOW in the MSO configuration file? In general, you should not use numbers that are lower than the defaults, but might need to increase the values, depending on application storage requirements.
4. Are applications using an unusual amount of storage? If so, you should allocate a larger address space, and should increase STORAGEABOVE and STORAGEBELOW appropriately. You can use the Display Users panel in the MSO Console to determine actual storage utilization by an application.
5. How long has the MSO address space been active? Eventually, due to storage fragmentation, even the largest address space will run out of large contiguous areas of storage. If MSO has been active for more than a few days, it is possible that fragmentation has occurred. In this case, the only solution is to recycle the MSO address space. You should schedule regular, possibly even daily, recycling of MSO in order to prevent this situation.

Diagnosing System Failures

In the event of a system failure, MSO provides the Snapshot Facility (IBISNAP), MSO's own dump snapshot formatter, which enhances problem determination and serviceability by assisting dump interpretation by support personnel. This facility is automatic, to provide a clear record of a problem even when no dump is available.

The IBISNAP Facility is always active. This guarantees that every abend in the MSO region leaves at least some minimal problem determination information behind, thus shortening the problem detection and resolution cycle.

In some cases, the IBISNAP snapshot enables problems to be solved on the spot, and in other cases it makes obtaining a dump unnecessary. However, in other cases dumps and/or traces are required.

By default, each IBISNAP snapshot is written to a separate class A SYSOUT file. The SYSOUT class can be changed by specifying IBISNAP in the MSO configuration file (see Chapter 5, *The MSO Configuration File*). These SYSOUT files are closed and freed as soon as they are written.

If, instead, ddname IBISNAP is allocated in the MSO JCL, then this allocation is used, as is, for each abend. For SYSOUT files, this means that all IBISNAP snapshots produced by an MSO job are contained in a single SYSOUT file. For files allocated to DASD, you must specify DISP=MOD to prevent the snapshots from overwriting one another. Allocating ddname IBISNAP in the MSO JCL overrides the specification of IBISNAP in the MSO configuration file.

An IBISNAP snapshot contains a summary box, describing the system environment, the abend itself, the exact version and level of MSO, and information about the failing task. This is followed by a formatted dump of various MVS, MSO, and FOCUS control blocks and data areas. Finally, an image of the last screen written to the abending user's terminal is printed.

The following is an example of the summary box, taken from an actual abend, with a description of each of the fields in the summary:

Summary of the Environment

```
*****  
* CPU: FF0123459021 0000 *  
* MVS: ESA SP4.1.0 IPO1 *  
* CVTOPTS: HIPER UCBSV COMDS MVPG *  
* DFP: 3.2.0 * * DATE: 93.112 *  
* TIME: 16:20:07 *  
*****
```

where:

CPU

Is the CPUID and processor number.

MVS

Is the operating system level and MVS's name.

CVTOPTS

Describes the hardware features.

DFP

Is the release of DFP.

Date

Is the date the snapshot was produced.

TIME

Is the time the snapshot was produced.

About the Abend

```
*****
* JOB: MSOUSR1 JOB07622 * * ASID: 002B *
* APFAUTH: ON *
* TASK: SSFOC *
* TCB: 009CB308 *
* MODULE: F$ROOT +000002BA *
* PROGRAM: ???????? +03ACF8D2 *
* ABEND: S0C1 U0000 REASON=00000001 *
* PSW: 078D1200 83ACF8D2 *
* CMD: (0000) *
*****
```

where:

JOB

Is the Jobname and JES jobid.

ASID

Is the address space number, in hexadecimal.

APFAUTH

Identifies whether MSO is running APF-authorized.

TASK

Is the name of the failing task.

TCB

Is the failing task's TCB address.

MODULE

Is the abending load module & offset.

PROGRAM

Is the program in module (if known).

ABEND

Is the abend code and reason code.

PSW

Is the PSW at the time of abend.

CMD

Is the failing instruction.

This section of the box above is the first line of attack in troubleshooting an abend. The module, program, offset, and abend code lines make it possible to see whether the abend is a known one or not.

About the MSO Address Space

```
*****  
* MSO VER: 04/16/93 9304 *  
*****
```

where:

MSO VER

Is the release date and put level of MSO.

About the Failing Task

```
*****  
* SERVER: FOCUS *  
* USERID: JSMITH *  
* NAME: JOHN SMITH *  
* ACCOUNT: *  
* LOGGED: 93.112 16:19:19 *  
* PRYGRP: 07 *  
* SECID: JSMITH *  
* COMMID: T37CT17 *  
* ACCESS: VTAM *  
* LU2NAME: FOCUS1 *  
*****
```

where:

SERVER

Is the name of the server the user logged on to.

USERID

Is the logon userid of the failing task.

NAME

Is the name associated with the SECID by the security system.

ACCOUNT

Is the name of the account. There is no account if the field is blank.

LOGGED

Date and time of user logon.

PRYGRP

Is the MSO priority group.

SECID

Is the security userid.

COMMID

If a VTAM session, the user's terminal's luname. If a CICS session, the user's terminal name. If a TSO session, the user's TSO userid.

Applying Maintenance

When a PTF for MSO is to be installed, follow these steps:

1. First, apply the PTF to the appropriate load library or libraries, according to the PTF letter that accompanies the PTF. This usually involves relinking one or more modules. Almost all MSO programs are AMODE=31,RMODE=ANY. Be sure that the resulting program modules after links have the correct AMODE and RMODE. Only explicit directions for AMODE=24,RMODE=24 in the PTF letter should generate 24-bit MSO modules.
2. After the new modules are linked, re-apply any customization that was applied to MSO or FOCUS. This includes any exits that were installed into the affected modules, as well as any zaps that were applied.
3. Copy all the modules in FUSELIB.LOAD into FOCLIB.LOAD, if this was done during the original MSO installation.
4. Re-link the MSO CICS programs with the IBM CICS command-level stubs.
5. Ensure that the MSO JCL and the CICS JCL (if applicable) point to the newly updated libraries with the PTF applied.

CHAPTER 15

Site-Specific Customizations

Topics:

- Exits
- User Written Subroutines
- ZAPs
- Environmental Subroutines and MSOINFO

This chapter contains ways that you can customize your individual site environments.

Exits

The following exits allow you to individually tailor MSO for your site.

MSIDVER: User Authentication Exit

The MSIDVER exit is used to replace MSO's authentication of the userid and password for a VTAM user.

MSIDVER must have entry point MSIDVER, and should be linked into a module with name MSIDVER. It receives control in AMODE 31. MSO must be running with APF authorization in order to use MSIDVER. The MSIDVER module should be placed in FOCLIB.LOAD or in another library that is allocated to STEPLIB.

A sample is provided in MSO.DATA(MSIDVER). The parameters for the exit are documented in the sample.

MSIDTR: Userid Translation Exit

The MSIDTR exit is used to translate logon userids to security userids known by MSO.

MSIDTR must have entry point MSIDTR, and should be linked into a module with name MSIDTR. It receives control in AMODE 31. The MSIDTR module should be placed in FOCLIB.LOAD or in another library that is allocated to STEPLIB.

A sample is provided in MSO.DATA(MSIDTR). The parameters for the exit are documented in the sample. If you use the provided sample, be sure to update the userid table to include your MSO users, and to remove the sample userids.

MSCXUID: Supply CICS Userid

The MSCXUID exit is used to supply a CICS userid when ASSIGN USERID is not supported, or when it is desired to override the userid that it returns. MSCXUID is a stand alone CICS program that the user must supply.

MSCXUID is a regular CICS-defined program that we call as follows:

```
EXEC CICS LINK PROGRAM('MSCXUID') COMMAREA(AREA) LENGTH(8)
```

It returns the userid in the COMMAREA (8 bytes)

MSUSRXT: The MSO Initialization and Termination Exit

MSUSRXT is a user-written module that is called by MSO when MSO is started and ended, or if MSO abends. The parameters sent to MSUSRXT are as follows:

CALLTYPE: I4 -*INPUT*

CALLTYPE specifies the type of call made to MSUSRXT; 0 if MSO is being initialized, 1 if MSO is being terminated, and 2 if MSO is in the process of abending.

USER DOUBLEWORD: *address* -*INPUT*

The user doubleword address is the address of eight bytes of memory that is meant for the use of MSO users. To provide memory that can be shared by the MSO users, the MSUSRXT exit would GETMAIN whatever amount of memory desired, and put the address of this memory wherever the user double word address points. Any MSO user can then, by finding this double word area, access this common user memory. Individual users in MSO can find this doubleword area by using the MSODBLW: Retrieve the User Doubleword Address exit, described on page 15-6.

MAXFOC: I4 -*INPUT*

The maximum number of user tasks which can exist at any one time in the address space. This allows MSUSRXT, if desired, to GETMAIN all needed memory for all possible users in advance of the users logging on.

RETCODE: I4 -*OUTPUT*

MSUSRXT puts its return code at the location of this parm. If RETCODE is set to 4, MSO will immediately terminate.

To implement the MSUSRXT exit, the site must assemble and link their source code with module name MSUSRXT and entry point MSUSRXT. The module is deleted from storage after the initialization call, so the exit must be coded in a way that allows for this, such as keeping all static data in user common memory, described in User Common Memory on page 15-6. Once created, the module should be copied into FOCLIB.LOAD.

DYNALLOC: The DYNAM ALLOC User Exit

FOCUS contains a user exit that allows a site to trap all DYNAM ALLOC commands and, if desired, prevent a user from allocating a file. MSO sites in particular may wish to take advantage of the additional security afforded by this exit.

The exit is invoked every time a DYNAM ALLOC command is executed. The user program can then examine the DSNAME and user ID involved in the DYNAM command, and accept or reject the command. If the exit returns a zero, the DYNAM is accepted. If the exit returns a non-zero code, the DYNAM is rejected with a FOC796 message.

The user exit is a self-contained MVS module with a site-selectable name and entry point. The name and entry point of the module should be the same. This name is zapped into the module FOCUS at entry point DYNOPT+8. Once FOCUS sees that this area is non-blank, it loads the module, and on further calls to DYNAM, calls the module.

A sample of this exit can be found in FOCCTL.DATA(DYNEXIT).

MSVTXT: VTAM Text User Exit

The MSVTXT exit is used to manipulate the VTAM text in the DATA field passed by VTAM when the MSO VTAM logon screen has been bypassed. This occurs when using MSO with a VTAM session manager, as described in Chapter 9, *Testing MSO*.

MSVTXT is used in lieu of MSO's standard parsing of the VTAM text. MSVTXT might be necessary if, for example, the VTAM text is encrypted, which would prevent MSO from being able to parse the data properly.

MSVTXT is called with a single input parameter, which is the string passed to MSO by VTAM at session initiation. The output parameters of MSVTXT correspond to the fields on MSO's VTAM logon screen and have the same meaning as if they were read from that screen.

MSVTXT must have entry point MSVTXT, and should be linked into a module with name MSVTXT. It receives control in AMODE 31. The MSVTXT module should be placed in FOCLIB.LOAD or in another library that is allocated to STEPLIB.

A skeleton MSVTXT is found in MSO.DATA(MSVTXT).

SSUSRACT: VTAM Account Field Validation

The SSUSRACT exit is used to validate and/or alter the account field specified on the VTAM logon screen.

A skeleton SSUSRACT is provided in MSO.DATA(SSUSRACT). To install SSUSRACT, after it is assembled, run the following JCL:

```
//LINKSTEP EXEC PGM=IEWL,PARM='RENT,NCAL,LET,LIST,SIZE=(2048K)'  
//SYSPRINT DD SYSOUT=*  
//SYSUT1 DD UNIT=VIO,SPACE=(CYL,(10,5))  
//OBJ DD DSN=ssusract.assemble.object,DISP=SHR  
//FOCLIB DD DSN=prefix.FOCLIB.LOAD,DISP=SHR  
//MAINTAIN DD DSN=prefix.FOCTL.DATA,DISP=SHR  
//SYSLMOD DD DSN=prefix.FOCLIB.LOAD,DISP=SHR  
//SYSLIN DD *  
INCLUDE OBJ  
INCLUDE FOCLIB(SSFOC)  
INCLUDE MAINTAIN(SSFOC)  
NAME SSFOC(R)  
/*
```

MSSMFXT: Customized SMF Records

The MSSMFXT exit allows you to customize the SMF records that MSO writes for user logons and logoffs, or to write the data to somewhere other than the SMF files. The standard logoff record is described in Chapter 10, *MSO Usage Accounting*.

A sample of MSSMFXT is supplied in MSO.DATA. The sample contains DSECTS that map the SMF records and the parameters for the exit. The supplied sample simply writes the standard record to the SMF data stream.

Sample JCL to assemble and link-edit MSSMFXT is supplied in member MSSMFJCL of MSO.DATA. MSSMFXT must be reentrant, and runs in AMODE 24, RMODE 24.

MSCXWRT: Customized CICS Termination Processing

The MSCXWRT exit allows you to customize or eliminate the message that is written to a CICS user's terminal after terminating a FOCUS session, and to control the processing that is performed when the MSO transaction completes.

A sample of MSCXWRT is supplied in MSO.DATA(MSCXWRTS). Sample JCL to pre-process, assemble, and link-edit MSCXWRT is supplied in MSO.DATA(MSCXWRTJ).

FOCUSID: Setting the FOCUS DBA Password

The FOCUSID exit allows the FOCUS DBA password, normally set by the FOCUS command SET PASS, to be set automatically by an external security system such as RACF, CA-ACF2, or CA-TOP SECRET.

The use of FOCUSID is fully documented in the *FOCUS for S/390 MVS/TSO Installation Guide*.

Note:

When writing a version of FOCUSID to run under MSO, you must use the MSOINFO subroutine to find the userid, if needed; normal methods of retrieving the userid are not valid under MSO.

User Written Subroutines

As multiple FOCUS users function within the same MVS address space, it may be necessary for user-written subroutines to coordinate between themselves the use of MVS services. To this end MSO provides services that MSO sites can use to write user written subroutines that operate properly in a multiple-task address space. MSUSRXT, the MSO initialization and termination exit, and the MSODBLW subroutine exist to provide these services.

User Common Memory

MSO provides eight bytes of user memory that can be used by a site to allow the users in an MSO address space to share memory and coordinate services. The location of the eight bytes of memory can be obtained in a variety of ways, including the MSUSRXT and MSSMFXT user exits. The MSODBLW exit allows any MSO user written subroutine to obtain the address of the user doubleword at any time.

Usually, the user doubleword will be used by a site as the repository of the address of GETMAINed memory that is to be used by all the users in an MSO address space. Once all users can share some common memory, any MVS service that requires coordination between users can be used. For instance, a user-written subroutine for an MSO user can be written that does I/O independently of MSO that does not interfere with other MSO users using the same subroutine. As long as this subroutine enqueues on some resource and dequeues on the resource after the I/O is done, any number of MSO users can use the user-written subroutine without interfering with each other. The name of the resource would be present in the common memory all users share.

MSODBLW: Retrieve the User Doubleword Address

MSODBLW is a program that can be called within an MSO user written subroutine to retrieve the address of the user doubleword. This program is present in object form in MSO.DATA(MSODBLW). To obtain the address of the user doubleword, a user-written subroutine calls MSODBLW with two parameters: first, the address of a 300 byte workspace to be used by MSODBLW, and second, a fullword into which MSODBLW will insert the address of the user doubleword. The 300-byte workspace for MSODBLW *must* be non-sharable. The safest way to provide this space is to GETMAIN it in the user-written subroutine before the exit is called. MSODBLW is fully reentrant, and may be called in AMODE 24 or AMODE 31.

A user-written subroutine that calls MSODBLW simply links the object code MSO.DATA(MSODBLW) into the user-written subroutine module.

DDname Translation: The MSODDX Subroutine

MSODDX is a utility routine intended to be incorporated into user-written subroutines that will be accessing ddnames allocated to a FOCUS/MSO user. Its function is to provide a data adapter to MSO's ddname translation services, so that your program is able to access files under the same ddname MSO uses.

A good example of a routine that requires the use of the MSODDX utility is the FINDMEM user-written subroutine. FINDMEM is passed a ddname, a member name, and a return code field, and opens the specified ddname to determine if a member with the given name exists in the file. Under MSO, which is using ddname translation, FINDMEM must call MSODDX to determine the correct ddname to open.

Why are ddnames handled differently under MSO? Under MSO, each of the many FOCUS users want to allocate files according to their individual needs. Thus there is likely to be a different allocation for FOCEXEC, MASTER, etc., in each FOCUS task. In order to keep each user's allocations separate, MSO uses a scheme called ddname translation to create unique ddnames for every file belonging to each user. When a user allocates a file, MSO generates a unique name to use to perform the allocation, and saves, in a table, the relationship between the ddname the user sees, and the ddname that MSO uses for the actual allocation.

Only files allocated locally, by the individual FOCUS user, participate in this ddname translation scheme. Ddnames that are present in the MSO start-up JCL are not translated, and are referred to as global ddnames, since they are available to all of the FOCUS/MSO tasks. Global ddnames do not require the MSODDX routine, as they are not translated.

Since the allocation performed within the FOCUS session is not done under the same ddname originally specified by the user, when a user-written subroutine is to access one of those files, the program must find out what ddname has actually been used, and use the actual name when opening the file. This can be accomplished by calling the MSODDX routine to perform the translation for you.

MSODDX Calling Environment

The MSODDX routine offers a flexible data adapter. It is fully reentrant, may be called by a program in either AMODE 31 or AMODE 24, and returns valid results under FOCUS environments outside of MSO, including TSO and batch FOCUS. If you call it while executing in AMODE 24 (below the line), you must provide all data areas and addresses below the line as well.

MSODDX may be called from an assembly-language program or any high-level language that supports dynamic specification of the ddname to be opened. Some languages that do this are PL/I and C. Languages that require the ddname to be hard-coded at compilation time are not suitable. Consult the documentation for your particular compiler for further information.

Usage of MSODDX in a Typical Application

The most likely use for MSODDX is simply to be able to code a user-written subroutine that is callable under MSO, and will open and process a file for you. For such a routine, you will want to call MSODDX once for each file that you open, immediately before issuing the OPEN. Use a CALLTYPE of 1 (see parameters below for more information on CALLTYPE=1), and pass the generic ddname, or, in other words, the ddname that the user specified when the file was allocated. The correct ddname is returned to you, and you will either place this into the DCB (using ASSEMBLER), or into a variable (for a high-level language). Then you will issue the OPEN, and the remainder of your application proceeds normally. You do not need to test for MSO in your application. If you call MSODDX when not under MSO, it merely returns the same ddname you have passed it.

MSODDX Parameter List

MSODDX uses the standard OS-calling conventions. Register 1 must point to a list of fullword addresses, which point to the parameters described below; Register 13 must point to a 72-byte register savearea; Register 15 contains the entry point address of MSODDX; and Register 14 contains the return address to your program (this is all done for you in a high-level language call). The parameters are labeled to indicate whether a value is expected on input (I) to MSODDX, or will contain a value upon return from MSODDX (O).

Note:

For assembler-language callers the return code is provided as one of the parameters, and is not given in Register 15 (R15 is returned unchanged).

Input/ Output	Field	Contents
(I)	CALLTYPE	<p>The function to be performed. CALLTYPE is a fullword (4-byte integer) containing one of the following values:</p> <ul style="list-style-type: none"> 0 Check whether we are currently running under MSO. RETCODE is 0 if YES, 4 if NO. WORKAREA is required. GENDD and TRANDD are not referenced. 1 The STANDARD call. You provide the generic ddname (for example, FOCEXEC) in GENDD. The correct ddname to use is returned in TRANDD. WORKAREA is required. The RETCODE is 0 if the ddname is known to MSO, either for this individual FOCUS user, or as a ddname known globally to all MSO users. 4 means the ddname is not known to MSO (TRANDD will be set equal to GENDD, so it is not necessary to check RETCODE before using TRANDD). When called outside of MSO, TRANDD is set equal to GENDD. RETCODE is set to 0 if the ddname exists, or to 4 if the ddname does not exist. 2 Untranslation. You provide a translated ddname in TRANDD, the generic ddname it came from is returned in GENDD. If TRANDD is not found to be a translated ddname, then GENDD is set equal to TRANDD. RETCODE is 0 if TRANDD is a translated ddname, 4 otherwise. WORKAREA is required. Is always set to the same value as GENDD. 3 Create a translated ddname. Use only when your user-written subroutine will be performing dynamic allocation. Provide the generic ddname you intend to use in GENDD. If GENDD is not already in use, a new TRANDD is returned, and RETCODE is 0. If the ddname in GENDD is already in use, TRANDD is set equal to GENDD, and RETCODE is set to 4. WORKAREA is required. When called outside of MSO, TRANDD.

Input/ Output	Field	Contents
	CALLTYPE (continued)	<p>4 Delete a translated ddname previously created using a CALLTYPE 3 call. Use when your user-written subroutine will be dynamically freeing the ddname it had previously allocated. Provide the generic ddname in GENDD. If found and removed, RETCODE is 0. RETCODE is 4 if an error was encountered. WORKAREA is required.</p> <p>When called outside of MSO, this call will do nothing.</p>
(I/O)	GENDD	<p>The generic (untranslated) ddname. GENDD is an eight-byte character field. Its usage is described under the CALLTYPEs above, as it varies based on the CALLTYPE value.</p>
(I/O)	TRANDD	<p>The actual (translated) ddname. TRANDD is an eight-byte character field. Its usage is described under the CALLTYPEs above, as it varies based on the CALLTYPE value.</p>
(I)	WORKAREA	<p>A 300-byte workarea for MSODDX's use. WORKAREA is a 300-byte area used by MSODDX for its workarea. You must initialize this area to low values (hex zeroes) prior to the first call you make to MSODDX, and should not alter its contents from then on.</p>
(O)	RETCODE	<p>The return code from MSODDX. RETCODE is a fullword (4-byte integer) providing an indication of the results of the call made to MSODDX. The only values ever returned will be zero and four; their meanings are described under the CALLTYPEs above, as their meaning varies based on the CALLTYPE performed.</p>

Sample Assembler Call

This is a sample of the calling sequence and data areas that you may set up in your program for calling MSODDX. A program running in AMODE 24 is being assumed here (because the DCB must be below the line).

```

CALL MSODDX, (CALLTYPE, GENDD, TRANDD, WORKAREA,          X
              RETCODE), VL      call MSODDX
LA R3, MYDCB      point to the DCB
USING DCB, R3     use DSECT to reference DCB fields
MVC DCBDDNAM, TRANDD move correct ddname into DCB
DROP R3          drop the dsect's base register
OPEN (MYDCB, (INPUT)) now open the file with real DD etc...
CALLTYPE DC F'1'  return translated ddname
GENDD DC CL8'SAMPLE' the generic one known to FOCUS user
TRANDD DC CL8' '  the name to use
WORKAREA DC 300X'00' workarea for MSODDX
RETCODE DC F'0'  the return code

* sample DCB
MYDCB DCB DDNAME=X, DSORG=PO, MACRF=R a sample DCB for a PDS
* macro to obtain DSECT of the DCB
      DCBD DSORG=PO include this macro for DSECT of the DCB
      etc...

```

Link-Edit Example

To incorporate MSODDX into your program, link-edit it with your program to create the load module for your user-written subroutine. The object code for MSODDX can be found in MSO.DATA, member MSODDX. Below is a sample of a link-edit job that might be used. You will need to make changes to adapt this to your particular needs.

```

//MSODDX JOB
//LINK EXEC PGM=IEWL, PARM='XREF'
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=SYSDA, SPACE=(CYL,(1,1))
//MSODATA DD DSN=MSO.DATA, DISP=SHR
//MYPGM DD DSN=my.object.library, DISP=SHR
//SYSLMOD DD DSN=FOCLIB.LOAD, DISP=SHR
//SYSLIN DD *
INCLUDE MYPGM(program)
INCLUDE MSODATA(MSODDX)
ENTRY program
NAME program(R)
/*

```

ZAPs

The following zaps allow you to individually enable or disable features of MSO at your site.

IBISNAP -- Disable User Screen Display in IBISNAP Dumps

The IBI Snapshot Facility includes a copy of the last screen written to the terminal at the end of the dump. If your site considers this a security risk, apply this zap to prevent inclusion of the screen in the dump.

This IBISNAP zap is supplied in MSO.DATA(IBISNAP).

SUBMITZ2 -- Enable a Site-Specific SUBMIT Exit

Batch jobs that are submitted through either TED SUBMIT or DYNAM SUBMIT may be screened by a site-specific security exit, IKJEFF10.

The SUBMITZ2 zap is supplied in FOCCTL.DATA(SUBMITZ2).

In addition to running this zap, MSO must be running with APF authorization to enable access to IKJEFF10.

DYNZAP -- Enable Allocation of WRITER INTRDR

In order to use WRITER INTRDR in the DYNAM ALLOC command, the DYNAM zap must be run.

The DYNZAP zap is supplied in FOCCTL.DATA(DYNZAP).

DYNEXZAP -- Enable the DYNAM ALLOC User Exit

In order to use the DYNAM ALLOC User Exit (see DYNALLOC: The DYNAM ALLOC User Exit on page 15-3), the DYNEXZAP zap must be run. This zap inserts the name of the module containing the exit into FOCUS.

The DYNEXZAP zap is supplied in FOCCTL.DATA(DYNEXZAP).

Environmental Subroutines and MSOINFO

The MSOINFO subroutine returns information on the MSO environment. The subroutine has two parameters: the information being requested, and a return field where the results are placed. Unless otherwise specified, the return field must be an 8-byte field.

The following items can be queried:

Field	Length	Contents
ACCESS		Returns VTAM, TSO, or CICS.
ACCOUNT	40	The account field (currently implemented only for VTAM).
APPLID		The LU2 applid for the region (available only to VTAM callers).
CICSREG		CICS region name (available only if connected from CICS).
GROUP		Security group.
LOGID		Logonid, same field as from GETLOGID subroutine.
PRFID		Prefix, same field as from GETPRFID subroutine.
RELEASE	12	Release of FOCUS.
SECID		Security id, same field as from GETSECID subroutine.
SERVER		Name of the MSO region (batch job name or started task name).
SERVICE		Name of the SERVICE user is running under (normally FOCUS).
TASKNUM		Unique task number (a count from 1 to MAXSERVERS, as a 4-digit number).
TERMID		CICS terminal id (available only if connected from CICS).
TERMINAL		Luname of terminal (available only if connected via VTAM).
TSOID		TSO userid (available only if connected from TSO or ROSCOE).

The following example requests information about ACCESS, using Dialogue Manager. The value for ACCESS is returned to the variable &ANS:

```
-SET &REQ = 'ACCESS  ' ;  
-SET &ANS = '          ' ;  
-TSO RUN MSOINFO,&REQ,&ANS  
-TYPE You have accessed MSO via &ANS
```

The result is displayed as:

```
You have accessed MSO via VTAM
```

In addition, the FOCUS GETUSER subroutine is supported in MSO. MSO additionally provides subroutines to allow the user to obtain the MSO security ID (GETSECID), the logon ID (GETLOGID), and the prefix (GETPRFID). These subroutines are called using the same calling sequence as GETUSER. An example follows:

```
-SET &USERID='12345678' ;  
-SET &USERID=GETLOGID(&USERID) ;  
-TYPE Your userid is &USERID
```

The result is displayed as:

```
Your userid is THX1138
```

APPENDIX A

MSO/CICS Cooperative Processing

Topics:

- MSO FOCEXEC Cooperative Processing Service
- MSO/CICS Cooperative Processing Services
- CMSORCV Function Codes

CICS transactions and MSO FOCEXECs may now communicate directly with each other in a synchronous mode. A CICS pseudo conversational transaction may start an MSO session on behalf of a CICS user.

Once a cooperative processing session is started, data can be handed back and forth between FOCEXECs and CICS transactions in packets containing up to 256 bytes.

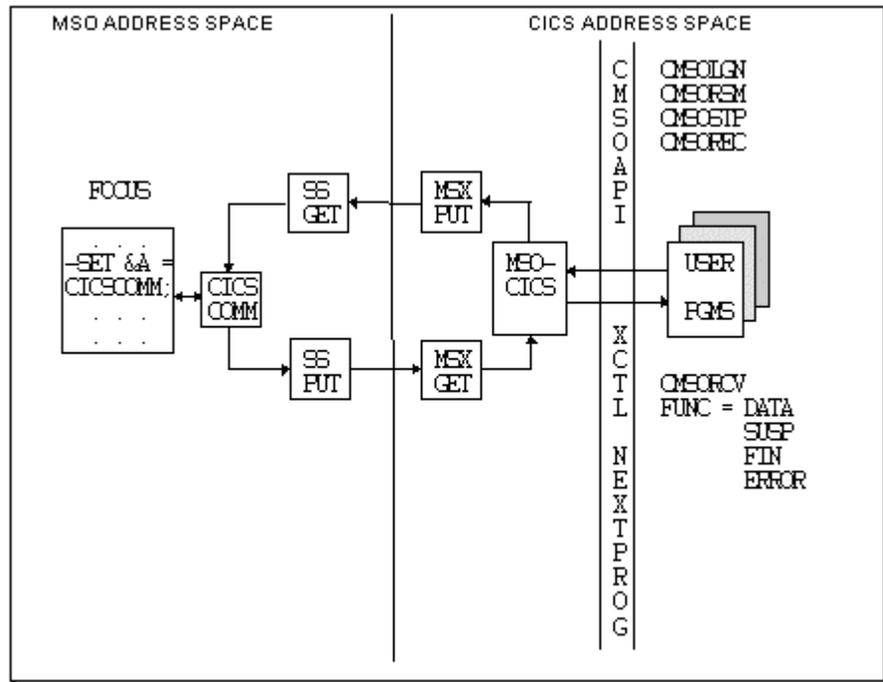
Additionally, a suspend function is available. When this is invoked in MSO, the CICS transaction is given control. This allows a “hot” MSO to be available to the CICS user. The CICS user may go in and out of MSO without terminating their MSO session.

These new facilities allow CICS transactions to get and retrieve data from the MSO region and vice versa. The FOCEXEC interaction is implemented in MSO via a FUSELIB routine, CICS COMM.

The CICS transaction functions are implemented by linking the installation's transaction module with the Information Builders supplied module, CMSOAPI. CMSOAPI provides five function calls that supply the MSO/CICS communications.

Also, a reconnect facility is available for cooperative processing sessions and for standard MSO connections from CICS. This allows the reconnection of an MSO session when the logical connection from a CICS terminal is lost.

The figure below illustrates the relationship between a FOCEXEC running in the MSO address space and user programs running in the CICS address space when a cooperative session has been established.



MSO FOCEXEC Cooperative Processing Service

The CICS_{COM} FUSELIB routine supplies the cooperative processing facility to FOCEXECs executing in the MSO region. It may be called from wherever a FOCUS user-written subroutine is supported. This routine is used to communicate and synchronize activity with the CICS portion of the dialog. When invoked, the FOCEXEC is placed into a wait until action is taken on the CICS side of the conversation. Data specified by the outlen/outbuf parameters is passed to CICS. When this subroutine is called it causes a CICS transaction specified in the CMSLOGN to be started in the CICS region. When CICS_{COM} completes, the inlen/inbuf parameters contain data that was passed back from CICS.

CICS_{COM} supports the following syntax:

```
CICSCOMM(timeout, outlen, outbuf, inlen, inbuf);
```

where:

timeout

Is the number of seconds to wait for response before timing out FOCUS. If this timeout duration is reached before the CICS portion of the session responds, the MSO FOCUS session is terminated. This action is represented as a FIN function code to the CMSRCV call.

outlen

Is length (0-256) of the outbound message to CICS.

outbuf

Is the field containing the outbound message to CICS.

inlen

Is the length (0-256) of the inbound message buffer.

This value is the maximum amount that can be returned by CICS. It is presented to the CICS portion of the conversation in the outlen parameter of the CMSORCV call.

inbuf

Is the field to contain the inbound message from CICS.

MSO/CICS Cooperative Processing Services

The CICS portion of the cooperative processing functions is provided in the IBI supplied module CMSOAPI. This module contains the functional code that supports the individual calls available to CICS transactions. When this module is link edited with a CICS transaction module, five functions become available to the module.

CMSOLGN

Logs a CICS user as an MSO FOCUS user and establishes a session between MSO FOCUS and CICS

CMSORCV

Interrogates MSO/CICS to identify the session that requires servicing, receiving data if sent.

CMSOREC

Reconnects to a session that is in an indeterminate state.

CMSOSTP

Stops or terminates the session. This cancels the users MSO FOCUS task.

CMSORSM

Sends data back to the FOCUS portion of a session or restarts a suspended MSO FOCUS session.

Syntax and descriptions of the functions follow:

How to Use the CMSOLGN Function

This function starts an MSO session for the user by logging on to MSO. It defines the details of the MSO/CICS conversation processing that will take place.

`CALL CMSOLGN(EIB, COMMAREA, MSONAME, USERWD, NEXTPROG, LOGONBUF)`

where:

EIB

Is the CICS Exec data adapter control block.

COMMAREA

Is the CICS Communications Area.

MSONAME

Is the 4 byte name of the MSO CICS transaction. Each MSO transaction name corresponds to a single MSO region that may be connected to.

USERWD

Is the 4-byte user word to associate with an MSO session.

It is an arbitrary 4-byte value that is returned by CMSOLGN. It represents a unique identifier for the conversation that this service just established. It will be returned when the CMSRCV is issued to identify the specific conversation that needs to be serviced

NEXTPROG

Is the 8-byte name of next program to call. It identifies the CICS user program that will be called when the CMSCOMM FUSELIB is invoked on the MSO side of the conversation. NEXTPROG is mutually exclusive with the LOGTRAN entry in the LOGONBUF control block. Only one of these parameters should be specified. They function identically. Cooperative processing support, that is, the ability to be called back at all, is enabled by specifying a NEXTPROG in CMSOLGN. Fill NEXTPROG with blanks or nulls if no callbacks are desired.

LOGONBUF

Is a required control block. It should be completely initialized to blanks (x'40') before individual fields are set. Some of the fields support the MSO Load Balancing feature.

An assembler copy file, CMSOAPIA, is supplied in MSO.DATA. This file maps the LOGONBUF. The fields and their meanings are listed in the table below. All fields are alphabetic.

Field Name	Length	Contents
LOGAPPL	8 bytes	A load balancing parameter that limits MSO region selection to those service groups that specify the same application name.
LOGSERV	8 bytes	Specifies the particular MSO Service Group that this user should be started in. It applies to load balancing as well as a single region MSO.
LOGTRAN	4 bytes	This defines the CICS transaction that will be invoked when the CICSCOMM FUSELIB routine is called in the MSO region. It is mutually exclusive with NEXTPROG in the invocation parameters. Only one should be specified.
LOGBREAK	4 bytes	Break key (PF/PAnn) Identifies the key that will unconditionally terminate the active MSO session.
LOGSUSP	4 bytes	Suspend key (PF/PAnn) Identifies the key that will cause the active MSO session to suspend operation. When this key is pressed, the LOGTRAN or NEXTPROG CICS transaction (whichever was specified) will be invoked in the CICS region.
LOGELVL	4 bytes	Error level (ALL,ERR,NONE) Controls what messages are displayed to the user when the MSO session ends. ALL - All messages are displayed ERR - Only error messages are displayed NONE - No messages are displayed
LOGGROUP	8 bytes	Logon group name The load balancing group name. It controls the MSO load balancing group that the user will be started in.
LOGVALID	1 byte	Security check flag (N/Y) Y must be specified for the reconnect service (CMSOREC call) to work. In addition, UNIQUE=LOGONID must be specified in the MSO configuration file.
LOGINITM	1 byte	Suppress initialization message flag (N/Y) Y suppresses the MSO initialization message.

Field Name	Length	Contents
LOGFLAG1	1 byte	Reserved flag
LOGFLAG2	1 byte	Reserved flag
LOGRESV	20 bytes	Reserved
LOGACCT	40 bytes	Account Specifies the MSO account field. This field will be recorded in the MSO SMF records when that feature is active.
LOGUPRM	256 bytes	Logon parameter The content of this field is available to FOCEXECs via the MSOINFO subroutine. It is generally used to provide control information to the MSO profile exec so that specific FOCUS applications may be invoked in the MSO region.

How to Use the CMSORCV Function

This function is used to interrogate the MSO/CICS control program. The returned parameters identify the particular conversation that had a status change, the current status of the conversation, and any data that may have been received from MSO. It is usually the first MSO/CICS Cooperative Processing service used in the transaction that is triggered by the CICS COMM subroutine (NEXTPROG or LOGTRAN).

The FUNCTION field is set based on either an event in the MSO region or if an event in the CICS region caused the status. The FUNCTION codes that are returned are described in *CMSORCV Function Codes* on page A-9.

```
CALL CMSORCV(EIB, COMMAREA, FUNCTION, CONNID, USERWD, INLEN, OUTLEN,
BUFFER, ERRNUM)
```

where

EIB

Is the CICS Exec data adapter control block.

COMMAREA

Is the CICS Communications Area.

FUNCTION

Is the 4-byte callback function code. See *CMSORCV Function Codes* on page A-9

CONNID

Is the 4-byte connect id of MSO session. This value, together with USERWD, uniquely defines each MSO/CICS session.

USERWD

Is the 4-byte user word to associate with an MSO session. This value, together with CONNID, uniquely defines each MSO/CICS session.

INLEN

Is the 4-byte length of buffer inbound from FOCUS.

OUTLEN

Is the 4-byte length of return buffer expected by FOCUS.

When using the CMSORSM, the LENGTH parameter may not exceed the value of OUTLEN. If it does, the data presented to the MSO FOCEXEC is truncated to the value of OUTLEN.

BUFFER

Is the 256 byte inbound data buffer.

ERRNUM

Is the 4-byte FOCUS ending error number (function FIN) or 4. This field is a binary number and is mapped by the CMSOAPIA member of MSO.DATA. Possible returned values are described in section *CMSORCV Function Codes* on page A-9

How to Use the CMSOREC Function

This function re-establishes a MSO/CICS session based upon the current user's CICS id. The connect id may be supplied if known. Otherwise, the function uses a connect id of 0 and the userid to identify the session. This condition may be caused by the user powering off their terminal while an MSO/CICS session is active and then logging on to CICS again.

Security flag - If the security flag in the logon buffer is set to yes, then the resuming or reconnecting userid is validated against the known userid. The reconnection is rejected if they do not match. If present, userid determination is subject to the MSCXUID exit. The default for the security flag in the logon buffer is no.

```
CALL CMSOREC(EIB, COMMAREA, MSONAME, CONNID, NEXTPROG)
```

where:

EIB

Is the CICS Exec data adapter control block.

COMMAREA

Is the CICS Communications Area.

MSONAME

Is the 4-byte name of MSO transaction.

CONNID

Is the 4-byte connect id of MSO session.

NEXTPROG

Is the 8-byte name of next program to call.

How to Use the CMSOSTP Function

This function stops an MSO/CICS session immediately. MSO must have passed control to the CICSCOMM FUSELIB program before a stop can be issued. A CMSOSTP received while MSO is still in control is treated as a protocol error.

```
CALL CMSOSTP(EIB, COMMAREA, MSONAME, CONNID, NEXTPROG)
```

where:

EIB

Is the CICS Exec data adapter control block.

COMMAREA

Is the CICS Communications Area

MSONAME

Is the 4-byte name of MSO transaction.

CONNID

Is the 4-byte connect id of MSO session.

NEXTPROG

Is the 8 byte name of next program to call. This field specifies a CICS transaction to start if the current invocation fails and cannot be associated with a known session. If a session is identified then the nextprog that was specified in the CMSOLGN service for the identified session is called.

How to Use the CMSORSM Function

This function sends data to an MSO FOCUS session that previously issued the CICSCOMM FUSELIB routine or resumes an existing MSO/CICS session that was suspended by the user with the suspend key. Data may be sent in both cases but will be ignored if the session is in a suspended state. A CMSORSM received while MSO is still in control is a protocol error. If there is data, a buffer is allocated which is freed by MSOCICS.

```
CALL CMSORSM(EIB, COMMAREA, MSONAME, CONNID, NEXTPROG, LENGTH, BUFFER)
```

where:

EIB

Is the CICS Exec data adapter control block.

COMMAREA

Is the CICS Communications Area.

MSONAME

Is the 4-byte name of MSO transaction.

CONNID

Is the 4-byte connect id of MSO session.

NEXTPROG

Is the 8-byte name of next program to call.

LENGTH

Is the 4-byte length of outbound buffer. Data sent to MSO FOCUS is truncated to the original length specified by INLEN on the MSO FOCEXEC call to CICSCOMM.

BUFFER

Is the outbound data buffer (up to 256 bytes).

CMSORCV Function Codes

The possible function codes that may be returned on the CMSORCV call are listed below:

Value name	Value	Meaning	Returned Parameters
IB_DATA	9	Data returned This is the result of CICSCOMM being invoked in MSO.	CONNID USERWD INLEN OUTLEN BUFFER
IB_SUSP	10	suspend key struck The users MSO session is dormant until CMSORSM is issued for it.	CONNID USERWD
IB_FIN	11	FOCUS session ended ERRNUM contains return code from the MSO FOCUS session.	CONNID USERWD ERRNUM
IB_ERROR	12	API protocol error ERRNUM contains the value for the error. See the following table for a list of errors and their meanings.	CONNID USERWD ERRNUM
IB_NOAPI	17	Program not called by MSO\CICS	None

The ERRNUM values that may be associated with a function code of IB_ERROR are:

Error	Value	Description
IBERR_STATE	1	MSO called in invalid state
IBERR_INVFUNC	2	MSO called w/invalid function
IBERR_SECURE	3	id verification failed.
IBERR_NORECON	4	RECON called but not supported
IBERR_NOTFOUND	5	user not found for connect id

Examples

The following sample members are supplied in FOCEXEC.DATA to aid in developing installation applications to use this feature:

CCDEMO	A FOCEXEC that implements a sample MSO menu to illustrate the function supplied by the CICS COMM FUSELIB routine
CCDEMOAS	Assembler source of CICS API program
CCDEMOAJ	JCL to build CCDEMOA module
CCMAPS	Map source for demo program
CCMAPSAJ	JCL to build mapset and MAP DSECT

Reconnection Capability

A CICS connection may be lost by powering off a terminal or closing an emulator session. This feature adds the capability of reconnecting to that session. Previously there was no way to do this and the user would not be allowed to re-logon until the original session had timed out or been canceled by the MSO operator.

A standard MSO session may be reconnected with, by specifying MSO RECON (where MSO is the CICS transaction that invokes the MSOCICS program).

An MSO/CICS cooperative processing session, one that was established via the CMSOLGN call, may be reconnected using the CMSREC call. CMSREC allows the use of conid or userid to be specified. If conid is specified then the userid of the MSO FOCUS session has to match the CICS userid only if the LOGVALID security flag was set to Y in the LOGONBUF. The reconnection is found using the userid associated with the CICS session. UNIQUE=LOGONID must have been specified in the MSO configuration file for this service to be able to reconnect.

Suspend key

If the suspend key is activated in the MSO FOCUS session, it is subject to the MSO configuration setting of IDLELIM. If the session is in the suspended state long enough to set off the IDLELIM limit, the session is terminated. This condition can be avoided by defining a separate service group that has IDLELIM set to a high value and placing the cooperative processing MSO sessions into these service groups by specifying the appropriate LOGSERV parameter.

APPENDIX B

Sample JCL for Installing MSO and CICS

Topics:

- Sample JCL for Installing MSO
- Sample JCL for Installing CICS

This appendix contains sample JCL, source code, and ZAPs used during the installs of MSO and CICS.

Sample JCL for Installing MSO

The following table lists the source code, samples, and JCL which are provided with MSO:

Member	Library	Type	Contents
MSIDVER	MSO.DATA	Source	Sample exit for VTAM user authentication.
MSLNKSTB	MSO.DATA	JCL	Link-edits CICS command-level stubs into MSO/CICS programs.
MSO	MSO.DATA	JCL	Sample MSO JCL.
MSODBLW	MSO.DATA	Object	MSODBLW subroutine.
MSODDX	MSO.DATA	Object	MSODDX subroutine.
MSOEXIT	MSO.DATA	Source	XZCAT global (CICS) user exit, for Attn key support.
MSOINFO	MSO.DATA	Object	MSOINFO subroutine.
MSOKILL	MSO.DATA	JCL	Sample JCL for running MSOKILL.
MSOTABLJ	MSO.DATA	JCL	Assembles & link-edits CMSOTABL.
MSOTORJ	MSO.DATA	JCL	Zap for overriding MSAT transid.
MSOUSER	MSO.DATA	CLIST	Sample CLIST for running MSO/TSO.
MSSMFJCL	MSO.DATA	JCL	Assembles & link-edits MSSMFXT.
MSXZCATJ	MSO.DATA	JCL	Assembles & link-edits MSOEXIT.
SMFEXEC	MSO.DATA	Focexec	Sample TABLE request for SMF report.
SMFFIX	MSO.DATA	Master	MFD for reading SMF dump datasets.
SMFVSAM	MSO.DATA	Master	MFD for direct read of SMF datasets.
SSCHCNFG	MSO.DATA	Config	Sample MSO Console configuration file.

Member	Library	Type	Contents
MSIDVER	MSO.DATA	Source	Sample exit for VTAM user authentication.
MSLNKSTB	MSO.DATA	JCL	Link-edits CICS command-level stubs into MSO/CICS programs.
MSO	MSO.DATA	JCL	Sample MSO JCL.
MSODBLW	MSO.DATA	Object	MSODBLW subroutine.
MSODDX	MSO.DATA	Object	MSODDX subroutine.
MSOEXIT	MSO.DATA	Source	XZCATT global (CICS) user exit, for Attn key support.
MSOINFO	MSO.DATA	Object	MSOINFO subroutine.
MSOKILL	MSO.DATA	JCL	Sample JCL for running MSOKILL.
MSOTABLJ	MSO.DATA	JCL	Assembles & link-edits CMSOTABL.
MSOTORJ	MSO.DATA	JCL	Zap for overriding MSAT transid.
MSOUSER	MSO.DATA	CLIST	Sample CLIST for running MSO/TSO.
MSSMFJCL	MSO.DATA	JCL	Assembles & link-edits MSSMFXT.
MSXZCATJ	MSO.DATA	JCL	Assembles & link-edits MSOEXIT.
SMFEXEC	MSO.DATA	Focexec	Sample TABLE request for SMF report.
SMFFIX	MSO.DATA	Master	MFD for reading SMF dump datasets.
SMFVSAM	MSO.DATA	Master	MFD for direct read of SMF datasets.
SSCHCNFG	MSO.DATA	Config	Sample MSO Console configuration file.
SSENTTAB	MSO.DATA	Source	Table with default MSO Console security resource class and entities.
SSUSRACT	MSO.DATA	Source	Sample exit for VTAM account field validation.

Member	Library	Type	Contents
FOCDNS3S	FOCSQL.DATA	Source	FOCDNS3 source; used with DSN3SATH.
FOCDNS4S	FOCSQL.DATA	Source	FOCDNS4 source; used with DSN3SATH.
SATHJCL	FOCSQL.DATA	JCL	Link-edits DSN3SATH.

Sample JCL to Unload MSO Datasets

```

//*****
//STEP1 EXEC PGM=IEBCOPY
//*****
//I35 DD DSN=MSO.LOAD,DISP=(OLD,PASS),UNIT=TAPE,
//      VOL=SER=tvolid,LABEL=(35,SL)
//I36 DD DSN=MSO.DATA,DISP=(OLD,PASS),UNIT=TAPE,
//      VOL=SER=tvolid,LABEL=(36,SL)
//*****
//O35 DD DSN=prefix.MSO.LOAD,DISP=(,CATLG,DELETE),
//      UNIT=SYSDA,VOL=SER=vvolid,SPACE=(TRK,(15,2,5)),
//      DCB=(BLKSIZE=13030,RECFM=U)
//O36 DD DSN=prefix.MSO.DATA,DISP=(,CATLG,DELETE),
//      UNIT=SYSDA,VOL=SER=vvolid,SPACE=(TRK,(10,2,10)),
//      DCB=(BLKSIZE=1600,RECFM=FB,LRECL=80)
//SYSIN DD *
COPY INDD=I35,OUTDD=O35
COPY INDD=I36,OUTDD=O36
/*
//SYSPRINT DD SYSOUT=*

```

Sample JCL for Installing CICS

The following members of MSO.DATA contain sample JCL, source code, and zaps for your use. They are reproduced here for reference.

Member	Contents
CICSJCL	Sample CICS JCL.
MSLNKSTB	JCL to link-edit CICS command-level stubs into MSO programs.
MSXZCATJ	JCL to assemble and link-edit MSOEXIT (XZCATT exit).
MSCXUIDJ	JCL to assemble and link-edit MSCXUID.
MSCXUIDS	Sample MSCXUID source.
MSCXWRTJ	JCL to assemble and link-edit MSCXWRT.
MSCXWRTS	Sample MSCXWRT source.
CMSOTABL	Table with default next-transaction name, user auxtrace record number, and flag to suppress initialization message.
MSOTABLJ	JCL to assemble and link-edit CMSOTABL.
MSOTORJ	Sample zap of MSOTOR to change MSOATTN's transid.

APPENDIX C

Installing MSO-Supported Data Adapters

Topic:

- MSO-Supported Data Adapters

This appendix lists the data adapters supported by MSO.

MSO-Supported Data Adapters

The following data adapters are currently supported by MSO.

- ADABAS

For details, see the *ADABAS Interface User's Manual and Installation Guide 7.0*.

- DATACOM

For details, see the *CA-Datcom[®] Data Adapter Installation Guide 7.0*.

- DB2 Read/Write

For details, see the *DB2 and SQL/DS[™] Read/Write Interface Users Manual*.

- Oracle

For details, see the *FOCUS Interface to Oracle User's Manual*

- IDMS/R

For details, see the *CA-IDMS[®] Data Adapter Installation Guide 7.0*.

- IMS via the BMP Extension

For details, see the *IMS/DB Interface User's Manual and Installation Guide*.

- SYSTEM 2000

For details, see the *SYSTEM 2000 Interface User's Manual*.

- Teradata[®] Read/Write

For details, see the *Teradata Read/Write Interface Installation Guide*.

- TOTAL[®]

For details, see the *TOTAL Interface User's Manual*.

- VSAM Read/Write

For details, see the *VSAM Write Data Adapter User's Manual 7.0*.

- SUPRA[®]

For details, see the *SUPRA Interface User's Manual and Installation Guide 6.8*.

APPENDIX D

Security Implementation for DB2

Topics:

- Steps for Modifying DSN3SATH
- Modifying DSN3SATH

User-level security for DB2 is implemented via the standard IBM DB2 exit, DSN3SATH. DSN3SATH is present in the IBM DB2 sample library (for example, DSN610.DSNSAMP), which is supplied with DB2. This exit is called by DB2 when any task attempts to connect to DB2.

This sample exit assumes that RACF is your site's primary security package. For sites using the CA-ACF2 security package, the vendor supplies DSN3SATH; for CA-TOP SECRET there is a usermod to the IBM-supplied exit.

The assembler program DSN3SATH must be modified, so that if a task tries to connect to DB2, the primary authorization ID is set to the userid in the MVS ACEE control block, established by the MSO region during the sign on process. If secondary authorization IDs are also in use, they may also need to be set for the MSO region. If the task attempting to connect to DB2 is not the MSO region, then the original DSN3SATH logic is used unchanged.

Information Builders provides the program load library supplied with the DB2 data adapter, which includes a load module called FOCDSN3, which resides in FOCSQL.LOAD. FOCDSN3 is used to set the proper primary authorization ID. (See *Modifying DSN3SATH* on page D-3.

Another program, FOCDSN4, is also provided in the FOCSQL.LOAD library and is used to set the proper secondary authorization ID(s) for RACF and CA-TOP SECRET. FOCDSN4 is not needed with CA-ACF2; the secondary authorization ID(s) will be set correctly without FOCDSN4.

Steps for Modifying DSN3SATH

1. For RACF sites: Modify DSN3SATH as shown in *Modifying DSN3SATH* on page D-3. This section contains a portion of DSN3SATH, illustrating the correct location of the modifications.

For CA-TOP SECRET sites: You must obtain the DSN3SATH usermod from the Optional Materials file of the CA-TOP SECRET product tape. Apply the CA-TOP SECRET usermod. Then modify DSN3SATH as shown in *Modifying DSN3SATH* on page D-3.

Note:

The CA-TOP SECRET modifications will add a new instruction in the same location as the CALL FOCDSN4 line. The CALL FOCDSN4 insertion should be made following this new instruction:

```
L   R6,X'BO' ( ,R6)                @USER
```

As of CA-TOP SECRET Release 4.3 Genlevel 9301, the DSN3SATH usermod is no longer required, and the application of the usermod as described above should be omitted.

For CA-ACF2 sites: CA-ACF2 provides a sample, customized DSN3SATH exit on the CA-ACF2 tape (Release 5.1 or higher) or on PTF78210. Computer Associates, Inc supplies this PTF tape. The sample exit supplied by Computer Associates should be used in this case, rather than the IBM sample exit. FOCDNS4 should not be used with CA-ACF2.

2. Assemble and link-edit the modified DSN3SATH (AMODE31,RENT) into a module with entry point DSN3@ATH and name DSN3@ATH. Sample JCL is provided.
3. Copy the module into the DB2 exit library (usually DSN230.DSNEXIT).

Modifying DSN3SATH

The example shows changes to be made to the IBM sample DSN3SATH exit, which should be used for RACF and CA-TOP SECRET sites. The example after that should be used by CA-ACF2.

The arrows in the code indicate the lines containing Information Builders recommended modification of DSN3SATH, which calls FOCDNS3.

Note:

The positioning of these lines is appropriate assuming that no other changes or additions have been made to DSN3SATH previously. If any previous changes were made, then you should decide where the most appropriate location for this call to FOCDNS3 should be.

Modifying DSN3SATH for RACF and CA-TOP SECRET

```

SATH001 DS    0H
        USING WORKAREA,R11      ESTABLISH DATA AREA ADDRESSABILITY
        ST     R2,FREMFLAG      SAVE FREEMAIN INDICATOR
        XC     SAVEAREA(72),SAVEAREA CLEAR REGISTER SAVE AREA
        LA     R15,SAVEAREA      GET ADDRESS OF CSECT'S SAVE AREA
        ST     R13,FOUR(,R15)    CHAIN THE SAVE AREA BACK POINTER
        ST     R15,EIGHT(,R13)   CHAIN SAVEAREA FORWARD
        LR     R13,R15           ADDRESS OF CSECT'S SAVE AREA
        SPACE
        XC     EXPLARC,EXPLARC    INIT RETURN CODE TO NORMAL RETURN
        XC     SECCOUNT,SECCOUNT  CLEAR GROUP NAME COUNTER FIELD
        L      R8,PSAAOLD-PSA    GET CURRENT ASCB ADDRESS AND
        USING ASCB,R8           SET MAPPING ADDRESSABILITY
        EJECT
    
```

```

*****SECTION 1:  DETERMINE THE PRIMARY AUTHORIZATION ID *****
*
*   IF THE INPUT AUTHID IS NULL OR BLANKS, CHANGE IT TO THE AUTHID
*   IN EITHER THE JCT OR THE FIELD POINTED TO BY ASCBJBNS.
*
*   THE CODE IN THIS SECTION IS AN ASSEMBLER LANGUAGE VERSION OF
*   THE DEFAULT IDENTIFY AUTHORIZATION EXIT.  IT IS EXECUTED ONLY
*   IF THE FIELD ASXBUSER IS NULL UPON RETURN FROM THE RACROUTE
*   SERVICE.  FOR EXAMPLE, IT DETERMINES THE PRIMARY AUTH ID FOR
*   ENVIRONMENTS WITH NO SECURITY SYSTEM INSTALLED AND ACTIVE.
*
*****
      SPACE
==>   LA   R1,AIDLPRIM          LOAD PARM REG1
==>   CALL FOCDSN3             GO GET INFORMATION BUILDERS EXIT
      CLI  AIDLPRIM,BLANK      IS THE INPUT PRIMARY AUTHID NULL
      BH   SATH020             SKIP IF A PRIMARY AUTH ID EXISTS
      L    R7,ASCBCSCB        GET CSCB ADDRESS
      CLI  CHTRKID-CHAIN(R7),CHTSID IS IT TSO FOREGROUND ADDR SPACE
      BNE  SATH010            BRANCH IF NOT
      L    R7,ASCBJBNS        GET ADDRESS OF LOGON ID
      MVC  AIDLPRIM,0(R7)     MAKE IT THE PRIMARY AUTH ID
      B    SATH019            TO END OF THIS ROUTINE
SATH010 DS   0H              NOT TSO, BUT BATCH OR STC SPACE
      L    R6,PSATOLD-PSA     CURRENT TCB ADDRESS
      L    R7,TCBJSCB-TCB(,R6) CURRENT JSCB ADDRESS
      L    R5,JSCBJCT-IEZJSCB(,R7) CURRENT JCT ADDRESS
      LA   R5,X'10'(,R5)      ADJUST FOR CORRECT DSECT MAPPING
      CLI  JCTUSER-INJMJCT(R5),X'4E' IF JCTUSER PLUS SIGN OR LESS
      BNH  SATH019            THEN LEAVE AIDLPRIM BLANK  KEB0026
      MVC  AIDLPRIM(7),JCTUSER-INJMJCT(R5) COPY JOB USER ID
      MVI  AIDLPRIM+7,BLANK   ASSURE BLANK PADDING
SATH019 DS   0H              END OF ROUTINE
      EJECT
*****SECTION 2:  DETERMINE THE LIST OF SECONDARY AUTHORIZATION IDS*****
*
*   THIS SECTION IS WRITTEN SPECIFICALLY FOR THE RACF ENVIRONMENT.
*   IT CAN/SHOULD BE REPLACED FOR OTHER SECURITY PRODUCTS.
*
*****
*   IF RACF IS ACTIVE AND THE LIST OF GROUPS OPTION IS ALSO ACTIVE,
*   USE THE CGRP AREA TO GET THE CONNECTED GROUP NAMES.
*   COPY THEM TO THE SECONDARY ID LIST IN THE AIDL.
*
*****
      SPACE
      CLI  AIDLPRIM,BLANK      IS THE INPUT PRIMARY AUTHID NULL
      BNH  SATH090            EXIT IF PRIMARY AUTH ID NULL
SATH020 DS   0H              BRANCH TO HERE IF PRIMARY EXISTS

```


Modifying DSN3SATH for CA-ACF2

```

*****
*          PRIMARY AUTHORIZATION ID          *
*****
*
--->    LA      R1,AIDLPRIM          POINT TO AUTH FIELD
--->    CALL    FOCDSN3              CALL INFORMATION BUILDERS TASK-
LEVEL-EXIT
        CLI     AIDLPRIM,C'  '      PRIMARY AUTHID THERE?
        BH     PRIMTSO              YES, EVERYTHING OK HERE
        L      R3,PSAAOLD-PSA(0)    CURRENT ASCB ADDRESS

```

Link JCL for DSN3SATH

The following sample link JCL for the IBM exit DSN3SATH can be found in FOCSQL.DATA(SATHJCL).

```

/** Your job card
/**
//LKED EXEC PGM=IEWL,PARM='LIST,XREF,LET,RENT,AMODE=31'
//OBJ DD DSN=DSN610.SDSNSAMP.OBJ,DISP=SHR
//FOCMOD DD DSN=prefix.FOCSQL.LOAD,DISP=SHR
//SYSLMOD DD DSN=DSN610.SDSNEXIT,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=SYSDA,SPACE=(100,(50,50))
//SYSLIN DD *
        INCLUDE OBJ
        INCLUDE FOCMOD(FOCDNS3)
        INCLUDE FOCMOD(FOCDNS4) <--- Omit for CA-ACF2
        ENTRY DSN3@ATH
        NAME DSN3@ATH(R)
/*

```

APPENDIX E

MSO Glossary of Terms

Topic:

- MSO Glossary of Terms

MSO glossary of terms.

MSO Glossary of Terms

MSO	Multi-Session Option is an Information Builders product offering that provides a multi-user MVS address space for executing FOCUS commands and applications.
region	An alternate term for an MVS address space
Load Balancing	An MSO feature that automatically routes an MSO user to one of several MSO regions.
group, Load Balancing	A named collection of MSO regions that comprise a Load Balancing group.
primary or initially contacted MSO region	An MSO user initiates a Logon request to a Load Balancing group by issuing a normal MSO Logon to one of the MSO regions in the group. This first step is referred to as the primary MSO region or the initially contacted region.
destination region	This is the MSO region that the user is routed to by the MSO Load Balancing algorithm.
MSO Configuration File	This file is a text file that contains the startup specifications for the MSO region.
communication data sets	A pair of sequential files that are dedicated to an individual MSO region. These data sets provide the basis for access to MSO for TSO and CICS users.
service block	The MSO configuration file must have at least one service block defined that specifies PROGRAM=FOCUS. This is the basic definition of the MSO FOCUS server. In most MSO environments only one service group is defined per MSO region.
service name	Each service block definition is started by specifying SERVICE=service_name. This name is referred to as the service name.
application name	A new service block keyword to support Load Balancing. It associates a meaningful name such as "PAYROLL" or "ACCT" with one or more MSO Load Balancing regions in an MSO Load Balancing group.
sub-group or target group	A subset of MSO regions in an MSO Load Balancing group. This subset is mapped based upon the application name specified in the Logon request and is treated as the total MSO group of regions eligible for the user.

Index

A

APF authorization, 2-4, 3-9
APFAUTH, 5-5
API, 7-13
APPLICATION, 7-11, 7-17
APPLNAME, 5-16, 7-6
Attention key support, 1-9, 8-11, 8-18

B

batch processing, 1-11, 6-8
buffers, 4-3

C

CA-ACF2, 12-4
CA-TOP SECRET, 12-5
CICS, 3-3
 access to MSO, 3-7, 9-3
 attention key support, 8-14
 Attention key support, 8-10, 8-18
 communication datasets, 8-4
 control blocks, 8-11
 cooperative processing, A-2
 diagnostic information, 8-23
 ending a FOCUS session, 8-13
 initializing the MSO link, 8-16
 initialization and termination of a user session, 8-16
 installation, 8-2
 installation prerequisites, 8-2
 installation requirements, 8-2
 installing access to MSO, 6-11
 JCL, 8-4
 keyboard locking, 8-13
 LOGON, 7-12
 MRO, 8-9, 8-18
 MSCXUID, 8-5, 8-19

 MSCXWRT, 8-5, 8-17
 MSO Master Terminal, 8-12
 MSO setup, 8-6
 MSOCICS, 8-16
 MSOPLT, 8-16
 MSOUTIL, 8-16
 multiple address spaces, 8-5, 8-9
 PLT functions, 8-11
 required datasets, 8-4
 session termination key, 8-14
 setup, 8-7
 shutdown, 8-12
 starting a FOCUS Session, 8-13
 startup, 8-11
 steady-state operation, 8-15
 terminating the MSO link, 8-16
 transaction security, 8-19
 troubleshooting, 8-22
 userid processing, 8-19
CICS setup, 8-6, 8-8
CICSBREAK, 5-8
CMSOLGN, A-4
CMSORCV, A-6, A-9
CMSOREC, A-7
CMSORSRM, A-8
CMSOSTP, A-8
configuration
 connectivity, 5-7
 diagnostic facilities, 5-11
 global, 5-2
 MSO Console security, 5-8
 obsolete parameters, 5-17
 service block, 5-13
Configuration file, 8-4
connectivity options, 5-7
CONSCLASS, 5-10
CONSDTL, 5-9
CONSEC, 5-8

CONSENTD, 5-11
CONSENTL, 5-10
CONSENTO, 5-10
CONSOPER, 5-9
CONSUSER, 5-10
control tasks
 SSCON, 3-4
 SSCTL, 3-3
 SSGET, 3-4
 SSOPER, 3-4
 SSPUT, 3-5
 SSTIME, 3-5
cooperative processing, A-2

D

data adapters
 ADABAS, C-2
 DATACOM, C-2
 DB2, C-2
 IDMS/R, C-2
 IMS, C-2
 Oracle, C-2
 SUPRA, C-2
 SYSTEM 2000, C-2
 Teradata, C-2
 TOTAL, C-2
 VSAM, C-2
DB2
 security, D-2, D-3
diagnostic facilities, 5-11
 IBISNAP, 5-11
 SNAPEXCLUDE, 5-11
 TRACECLASS, 5-12
 TRACELIB, 5-12
 TRACEPREF, 5-12
diagnostic information
 MSOCICS, 8-23
Display Users, 4-5
DSN3SATH, D-3
DU, 4-5

DYNALLOC, 15-3
DYNAM, 1-10, 4-8
 allocations for MSO, 6-10
DYNAM vs. JCL, 6-10
DYNEXZAP, 15-12
DYNZAP, 15-12

E

environmental information, 1-11
ERRORS, 6-9
ERRORS.DATA, 8-4
exits, 15-2
 DYNALLOC, 15-3
 FOCUSID, 15-5
 installing, 6-12
 MSCXUID, 15-2
 MSCXWRT, 15-5
 MSIDTR, 15-2
 MSIDVER, 15-2
 MSSMFXT, 15-4
 MSUSRXT, 15-2
 MSVTXT, 15-4
 SSUSRACT, 15-4
EXTSEC, 5-2, 7-16

F

failure processing, 1-6
FAQS, 2-2
FastLoad, 1-5, 2-3, 3-8, 4-4, 5-5
FASTPDS, 5-6
FOCCONS, 6-9
FOCLIB.LOAD, 6-6, 8-4
FOCMO, 6-9
FOCUSID, 15-5
frequently asked questions, 2-2
FSTRACE, 1-6

FUSELIB.LOAD, 6-6

G

graphics, 1-10

H

HiperFOCUS, 1-11

I

IBI Subsystem, 2-2, 5-15, 7-2, 7-18, 8-2

IBISNAP, 5-11, 6-10, 14-3, 15-12

IBM RACF, 7-17

IDLELIM, 5-4, 5-14

IEFUSI, 4-5

installation

authorizing FOCLIB.LOAD, 6-3

CICS access, 6-11

configuration file, 6-7

datasets and files, 6-2

FUSELIB.LOAD, 6-6

IBI Subsystem, 6-4

installation steps, 6-2

installing FOCUS, 6-3

JES2 banner support, 6-4

MSDEB, 6-6

MSGET, 6-6

MSO address space, 6-3

MSO JCL, 6-8

MSOPROF, 6-7

MSPUT, 6-6

problems, 14-2

PROFILE, 6-7

requirements, 1-2

system security, 6-4

unloading MSO datasets, 6-6

VTAM APPLID, 6-5

VTAM logon screen, 6-12

Internal APF authorization, 3-9

J

JCL

allocations for MSO, 6-10

CICS installation, B-5

MSO installation, B-4

JCL vs. DYNAM, 6-10

JES2, 2-2

K

keyboard locking, 8-13

L

limitations

MSO DDNAME, 4-8

TIOT restrictions, 4-8

Load Balancing, 2-4, 5-16, 7-2, 7-11, 7-16,

benefits, 7-20

CICS, 7-13

defaults, 7-6, 7-15

logon procedures, 7-10

parameters, 5-15, 7-7

requirements, 7-3, 7-15

segmenting, 7-9

troubleshooting, 7-18

LOGON_TIMEOUT, 5-15

LU2_NAME, 5-7

M

MAXIMUM, 5-13, 7-17

MAXSERVERS, 5-4

memory, 1-4

above the line requirements, 4-4

below the line requirements, 4-2

real memory requirements, 4-6

user common memory, 15-6

virtual memory requirements, 4-2

MODIFY, 11-2

- MRM, 3-7, 5-6, 7-2
 - MRM_INTERVAL, 5-7
 - MRM_PERCENT, 5-7
 - MRO, 8-7, 8-18
 - MSCXUID, 8-5, 8-19, 15-2
 - MSCXWRT, 8-5, 15-5
 - MSDEB, 6-6
 - MSGET, 6-6, 6-9
 - MSIDTR, 15-2
 - MSIDVER, 15-2
 - MSMT, 8-12
 - MSO
 - JCL, 8-4
 - multiple regions, 9-4
 - security, 12-2
 - testing, 9-2
 - MSO Console, 1-4, 2-2, 4-5, 4-8, 5-3, 13-2
 - console authority, 13-5
 - dataset browser, 13-19
 - display servers, 13-14
 - display user, 13-9
 - installation, 6-11, 13-2
 - log viewer, 13-15
 - logon screen, 13-7
 - online help, 13-21
 - operation, 13-6
 - operator commands, 13-9
 - primary option menu, 13-8
 - security, 13-4
 - trace handling, 13-16
 - WHOHAS, 13-12
 - MSO console security, 5-8
 - MSO Resource Manager, 3-7, 7-2
 - MSO.DATA, 8-4
 - MSO.LOAD, 8-4
 - MSOCICS, 7-12
 - MSODBLW, 15-6
 - MSODDX, 15-6
 - MSOINFO, 1-11, 7-19, 15-13
 - MSOKILL, 2-2, 6-10
 - MSOPRINT, 6-9
 - MSOPROF, 6-7
 - MSPUT, 6-6, 6-9
 - MSSMFXT, 15-4
 - MSUSER, 7-12
 - MSUSRXT, 15-2
 - MSVTXT, 15-4
 - multiple regions, 1-3
 - MVS, 2-4
 - performance group recommendations, 4-7
- ## N
- NUMBER_READY, 5-13
- ## O
- operator commands, 1-4
 - MODIFY, 11-2
 - STOP, 11-2
- ## P
- PDS allocation, 1-10
 - printer support, 1-10
 - PRIVATEDD, 1-10, 7-16
 - PROFILE FOCEXEC, 1-7
 - PROGRAM, 5-13
 - PRTYGROUP, 5-14
 - PTFs, 14-8
 - installation, 14-8

Q

QUIESCE, 7-18

R

RACF, 12-3

RACSP, 5-5

REBUILD/INDEX, 1-10

regions

specifying size, 4-4

regions

monitoring region requirements, 4-5

Resource Manager, 1-6, 5-3, 5-6

S

SAF, 1-5, 7-17, 12-2

SASLIB, 8-5

SBORDER, 1-9

SDSF, 1-6

security, 1-5, 12-2

DB2, D-2

SAF calls, 12-2

VERIFY, 12-2

SERVICE, 5-13, 5-16, 7-8, 7-11, 8-6

service block, 5-12, 5-16, 7-8, 7-17

IDLELIM, 5-14

LOGON_TIMEOUT, 5-15

MAXIMUM, 5-13

NUMBER_READY, 5-13

PROGRAM, 5-13

PRTYGROUP, 5-14

SERVICE, 5-13

STORAGEABOVE, 5-15

STORAGEBELOW, 5-14

TIMEOUT, 5-14

session termination, 1-9

Simultaneous Update, 2-2

Simultaneous Usage, 1-11

SmartMode, 1-11

SMF

fields recorded, 10-3

modifying records, 10-4

processing records, 10-2

SMF records, 2-3

SMFNUM, 5-3, 5-6, 10-2

SNAPEXCLUDE, 5-11

Snapshot Facility, 1-4, 2-3, 5-11, 14-3

SOS_PERCENT, 5-7

SSCONSEC, 5-8

SSCTL, 3-2, 3-6

SSFOC, 3-5

SSUSRACT, 15-4

STDOUT, 6-9

STEPLIB, 6-9

STOP, 11-2

storage

expanded, 4-6

problems, 14-3

STORAGEABOVE, 5-4, 5-15

STORAGEBELOW, 5-3, 5-14

SU, 1-11, 2-2

SUBMITZ2, 15-12

subroutines

environmental, 15-13

SYSOUT, 2-3

system programmer tasks, 6-3

SZERO, 5-6

T

TED, 1-11

Terminal Operator Environment, 1-7

thrashing
 preventing, 4-7

TIMEOUT, 5-14

TIOT, 4-8

TOE, 1-7, 7-5

trace handling
 MSO Console, 13-16

trace management, 1-6

TRACECLASS, 5-12

TRACELIB, 5-12

TRACEPREF, 5-12

TRANSID, 5-8

TSO
 access to MSO, 9-4
 LOGON, 7-12

U

UDB, 2-4

UNIQUE, 5-3

usage accounting, 1-3, 2-3, 10-2

user exits, 1-3, 8-5

user tasks, 3-5

USERLIB, 6-10

USERPARM, 7-11

user-written subroutines
 MSODBLW, 15-6
 MSODDX, 15-6

V

VTAM, 1-3, 3-4, 6-12
 access to MSO, 9-2
 applid, 2-2
 LOGON, 7-11
 logon screen, 6-12
 session managers, 9-3

W

WHOHAS, 13-12

X

XAI0, 4-3

Z

ZAPs, 2-4
 DYNEXZAP, 15-12
 DYNZAP, 15-12
 IBISNAP, 15-12
 installing, 6-12
 SUBMITZ2, 15-12

Reader Comments

In an ongoing effort to produce effective documentation, the Documentation Services staff at Information Builders welcomes any opinion you can offer regarding this manual.

Please use this form to relay suggestions for improving this publication or to alert us to corrections. Identify specific pages where applicable. You can contact us through the following methods:

Mail: Documentation Services – Customer Support
Information Builders, Inc.
Two Penn Plaza
New York, NY 10121-2898

Fax: or FAX this page to (212) 967-0460, or call **David Kempler** at (212) 736-4433, x**3480**.

E-mail: books_info@ibi.com

Web form: <http://www.informationbuilders.com/bookstore/derf.html>

Name: _____

Company: _____

Address: _____

Telephone: _____ Date: _____

E-mail: _____

Comments:

Reader Comments