
The MSO/CICS Interface in FOCUS Release 7.0**Introduction**

New Features

Preparing for Installation

Prerequisites

Installation Requirements

Planning

Required Datasets

Changes From Previous Releases

Temporary Storage Queues

Attention Key Support

Next-Transaction Support

MSO User Exits

Configuration File Keywords

Considerations for Multiple Address Spaces

Considerations for File Transfer

Installation Steps

MSO Setup

CICS Setup

CICS Setup for MRO

Multiple Address Spaces

Installation Testing

Testing Basic Functionality

Testing Attention Key Support

Testing PLT Functions

Operations

MSO/CICS Startup

Startup for Attention Key Support

MSMT -- MSO Master Terminal

CICS Shutdown

End-User Issues

Starting a FOCUS Session

Ending a FOCUS Session

Keyboard Locking

Attention Handling
Session Termination Key
MSO and Other CICS Programs

MSO/CICS Technical Description

Steady-State Operation
Initialization and Termination of the MSO Link
Initialization of the MSO Link by MSOCICS
Termination of the MSO Link by MSOUTIL
Termination of the MSO Link by MSOPLT
Initialization and Termination of a User Session
MSOCICS Initialization
MSOCICS Termination
The MSCXWRT User Exit
Attention Key Handling
MRO Issues
Userid Propagation Under MRO
Attention Key Support Under MRO
Transaction Security
Userid Processing
The MSCXUID User Exit
Summary of MSO Transactions and Programs

Troubleshooting

Diagnostic information

Error Messages

Messages for MSOCICS (Transaction MSO)
Messages for MSOUTIL (Transaction MSMT)

Sample JCL & Source Code

Sample 1. CICSJCL: Sample CICS JCL
Sample 2. MSLNKSTB: JCL to link-edit CICS command-level stubs into MSO programs
Sample 3. MSXZCATJ: JCL to assemble and link-edit MSOEXIT (XZCATT exit)
Sample 4. MSCXUIDJ: JCL to assemble and link-edit MSCXUID
Sample 5. MSCXUIDS: Sample MSCXUID source
Sample 6. MSCXWRTJ: JCL to assemble and link-edit MSCXWRT
Sample 7. MSCXWRTS: Sample MSCXWRT source
Sample 8. CMSOTABL: Table of site-modifiable defaults
Sample 9. MSOTABLJ: JCL to assemble and link-edit CMSOTABL
Sample 10. MSOTORJ: Sample zap of MSOTOR to change MSOATTN's transid

Introduction

In Release 7.0, the CICS component of the FOCUS Multi-Session Option (MSO) has been substantially rewritten. This rewrite provides keyboard lock, attention key support, simplified installation, enhanced diagnostic facilities, a cleaner user interface, and enhanced error messages.

This document provides information on installation, operations, usage, troubleshooting, and the internal structure of the new CICS interface. It also contains copies of the new error messages and the sample MVS JCL, zaps, and user exits. Except where otherwise indicated, this information supersedes the information provided in the *FOCUS for IBM Mainframe: Multi-Session Option Installation and Technical Reference Guide, Release 6.8*, referred to as the MSO Installation Guide in this document.

New Features

- **Keyboard lock:** Just as with FOCUS on TSO and with MSO/VTAM and MSO/TSO, the 3270 keyboard is now locked during processing.
- **Full Attention key support:** The 3270 Attention key is now fully supported and can be used to interrupt processing whenever the keyboard is locked. When the key is pressed, the user is given the option of returning to FOCUS command level or ending the FOCUS session.
- **Simplified installation:** Redesigned transaction and program structure and cleaner MRO support greatly simplify the installation process.
- **Enhanced diagnostic facilities:** The new program structure simplifies tracing and troubleshooting. In addition, MSO now writes user auxtrace records when running under CICS Version 3 or higher.

Preparing for Installation

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

Prerequisites

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 either FOCUS SVC or 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 MVS/ESA or MVS/XA that is currently supported by IBM, and CICS 2.1 or higher.

The new MSO/CICS interface completely replaces the old version. Both FOCUS and MSO must, as always, be installed from the same tape. No mixing of release or PUT levels is allowed in any interconnected MSO and CICS address spaces.

The old and new releases of MSO can coexist in the same MVS system, but can not be used in the same CICS MRO collection or the same MSO address space.

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 datasets 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 user-id 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 Datasets

Please note that all IBI-supplied datasets must come from the same tape. It is not permissible to mix datasets 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 from PUT level to PUT level, and change significantly from release to release. Most of them have been completely rewritten for the new MSO/CICS support in Release 7.0.

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 which 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 which 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 datasets	As described in the MSO Installation Guide, these datasets 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 which 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.

Changes From Previous Releases

Temporary Storage Queues

As MSO no longer uses CICS Temporary Storage Queues, the MSCREQID exit is obsolete.

Preparing for Installation

Attention Key Support

MSO/CICS now features new Attention Key support. The old feature, which allowed a PA key to terminate a session, is now called the "Session Termination Key" feature. While its use is discouraged, it is enabled by specifying the CICSBREAK parameter in the MSO configuration file. In addition, the AIDDEF zap is no longer available. You must use the MSO configuration file to specify the key to be used.

Next-Transaction Support

The MSOZ transaction is no longer used; MSO simply clears the screen when it is done. You may specify the TRANSID parameter in the MSO configuration file to cause MSO to start another transaction when it is done. You may also update CMSOTABL to change the default, if desired, as described in this document.

MSO User Exits

The following user exits have not changed from Release 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.

Configuration File Keywords

The following CICS-related keywords may be specified in the configuration file:

`CICSBREAK < NONE | PAn | PFn >`

This is a CICS-only parameter. CICSBREAK specifies the 3270 key to be used as the CICS Session Termination Key. When this key is pressed, the user's MSO session will be terminated and all the user's resources freed. If CICSBREAK is not specified, this feature will not be active. 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 which will be used in any FOCUS applications which might be run.

CICSBREAK replaces the ATTNKEY parameter from Release 6.8 and earlier. (If ATTNKEY is specified, message MSO13086 will be issued.)

Any PA key (from 1 to 3) or PF key (from 1 to 24) may be specified.

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

`TRANSID = < CICS transaction ID >`

This is a CICS-only parameter. TRANSID specifies the CICS transaction that is to be executed when an MSO/CICS user returns to CICS. If no transaction is specified, MSO will simply clear the screen when done.

Installation Steps

The specified TRANSID does not take effect until the MSO transaction successfully establishes a connection with MSO. If the connection is not established, then the default TRANSID, if any, will be used.

A default TRANSID may be specified by re-assembling CMSOTABL.

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 and PUT 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.

Considerations for File Transfer

File transfer using the XFER command works in MSO as described in the *FOCUS for IBM Mainframe Users Manual*. For XFER to work properly, the CICS terminals must be configured to accept input and output of at least 2048 bytes each; the usual remote SNA terminal definitions are 1500 and 256 bytes. These values may be changed by updating the RDO SENDSIZE and RECEIVESIZE parameters.

Only XFER is available for file transfer under MSO. IND\$FILE and similar products do not work with MSO.

Installation Steps

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 which will be accessed from CICS.

As mentioned above, either the FOCUS SVC or 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. These files should have been created as described in the MSO Installation Guide. Note that the ddnames are always MSGET and MSPUT in the MSO JCL, regardless of the ddnames which are used in the CICS JCL. There is only one pair of communication datasets per MSO address space; it is shared by all CICS and TSO address spaces which are communicating with that MSO. An example of these DD cards is:

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

Update the MSO configuration file:

Installation Steps

Ensure SVC=YES is specified if the IBI SVC is being used.

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.

If desired, code the TRANSID parameter to cause the MSO transaction to start another transaction when it completes. This specification overrides the default which is set by re-assembling CMSOTABL.

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 document assumes the name "MSO" for the transaction which is used to access MSO, and "MSMT" for the utility transaction, which is used to terminate MSO operations. Any valid transaction ids may be used, but you should keep in mind that the name of the MSO transaction will affect the ddnames which are used in the CICS JCL for the communication datasets.

Change the CICS JCL:

Include the dataset MSO.LOAD to the DFHRPL concatenation. If required by your version of MVS, ensure the block sizes of the datasets in the concatenation are in descending order, or override them if necessary.

Add a DD statement for FOCLIB, pointing to FOCLIB.LOAD.

Add a DD statement for ERRORS, pointing to ERRORS.DATA.

Add the MSOGET/MSOPUT ddnames. These are the communications datasets which were 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 were to use transaction id "FOC", you would use ddnames "FOCGET" and "FOCPUT".

If using a version of CICS which 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.MSPUT,DISP=SHR
//MSOGET DD DSN=prefix.MSO.MSGET,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.

Installation Steps

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.

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
MSAT	MSOATTN

Set up transaction security as required:

MSO must be run by all users.

MSMT is run by personnel authorized to stop MSO communications.

MTOR is started at CICS startup.

MSAT is started by MTOR when required.

Link the IBM Command-Level stubs into the programs in MSO.LOAD. Sample JCL is provided is 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.

If Attention Key support is needed, enable MSOEXIT (an XZCATT exit) at startup.

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 which 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 the "MSO/CICS Technical Description" section of this document.

Installation Steps

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 the "MSO/CICS Technical Description" section of this document.

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 datasets), and once per CICS address space (adding the MSO transaction and xxxxGET and xxxxPUT datasets 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, simply install an extra instance of the MSO transaction (using program MSOCICS) and an extra pair of communication datasets (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.MSPUT,DISP=SHR
//MSOGET DD DSN=prefix.MSO.MSGET,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 datasets:

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

The second MSO would have its communication datasets allocated as follows:

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

Installation Testing

```
//MSGGET 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, then issue FIN to end the session.
4. Run the MSMT transaction to break FOCUS communications. You should get message MSO13300 on your CICS terminal and MSO13028 in the MSO log.
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 again.
8. Bring down CICS and MSO; to ensure both can come down without any problem.

Testing Attention Key Support

As described in the Operations section, enable XZCATT and run transaction MTOR

1. Start an MSO session.
2. Start a long-running TABLE request.
3. Press Attention while the request is running, then press Enter to resume the request and ensure the table completes normally.
4. Run two more TABLE requests, trying the KX and HX options at the Attention prompt.
5. 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 which are 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 which corresponds to the specified transaction. (The first user to run the MSO transaction will encounter a slightly longer wait as communication is established.)

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

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

The value for GALEN is calculated by multiplying the maximum number of termids which 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 will be ignored from termids which do not fit 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)
```

or

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

where the transaction ids specified are the ones by which MSO is invoked.

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 which perform the communication.

End-User Issues

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 will see 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 Dialog 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 will perform 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, 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 will be presented with 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, if desired) may be defined as an MSO session termination key with MSO/CICS. This key, if pressed when the keyboard is unlocked, will cause 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 with the CICS Attention Key feature under 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 the communication datasets 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 you with 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 will be running 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.

MSO/CICS Technical Description

The MSXGET and MSXPUT subtasks communicate with their counterparts in the MSO address space, MSPUT and MSGET. They pass messages in processor storage, using either the IBI SVC or the IBI Subsystem to perform the communications. These communication methods are both described in other IBI documentation, which should be consulted for further details. Please note that the communication datasets which are set up during installation do not contain the actual messages being passed; they are only used for initialization and control of the cross-address-space communications.

The MSOCICS program, which, by default, is installed with transid MSO, is a pseudo-conversational transaction. It:

- 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 being 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 needs to 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 (which are 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 is provided to be run 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

MSOCICS 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 datasets are to be used in communicating with the MSO address space. This processing makes it critical that the transid and the ddnames of the communications datasets correspond (as described in the Installation section of this document). It also makes it crucial that MSOCICS always be run by invoking its associated transaction; CALLing it directly will cause it to read an incorrect transid and will 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.

MSOCICS Termination

When a user's MSO/FOCUS session is ended, MSOCICS 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 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

As discussed below under Transaction Security, 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 which must be active while any MSO sessions are running.

MSO/CICS Technical Description

- 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 will start the MSAT transaction in the AOR. MSAT will then notify 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. Table 1 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, then the userid will not be available to MSO, and user-level security will not be 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 to which they are not authorized. See the MSO Installation Guide for further information on external security issues.

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 will execute 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.

MSO/CICS Technical Description

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

Table 1: Transaction Summary

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.

Troubleshooting

MSXPUT	MVS		AOR		Part of inter-address-space communications; transmits data to MSO address space.
--------	-----	--	-----	--	--

Table 2: Program Summary

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 FOCUS SVC has not been installed.
- The correct SVC number has not been zapped into both the FOCUS and the MSO modules.
- The MSO configuration file does not contain SVC=YES.
- 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 datasets have not been created properly.
- The communication datasets are not allocated to the correct ddname or dsname. (Remember, the ddnames are different in the MSO address space and the CICS address space.)
- 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, the following might be requested, depending on the nature of the problem:

- A CICS auxtrace of the MSO and/or MSMT transaction. In CICS V3, this will include MSO's user-defined records as well as standard CICS records.
- An SVC dump of both the MSO and CICS address spaces.

Error Messages

Messages for MSOCICS (Transaction MSO)

MSO13201 COMMUNICATION WITH MSO: %1 IS NOT AVAILABLE.
 Communication with the specified MSO region is unavailable.

MSO13202 LOGON FAILED DUE TO QUIESCE. PLEASE PRESS ENTER KEY.
 A logon was rejected because the operator has quiesced new logons.

MSO13203 INSUFFICIENT STORAGE FOR NEW LOGON. PLEASE PRESS ENTER KEY.
 A FOCUS logon failed because the storage available was less than the
 threshold value. See messages MSO13144 and MSO13145 in MSOPRINT for more
 information.

MSO13204 LOGON FAILED, MAXIMUM SESSIONS IN USE. PLEASE PRESS ENTER KEY.
 A logon was rejected because the threshold for the number of running FOCUS
 sessions has been reached.

MSO13205 TIMEOUT WAIT LIMIT EXCEEDED. YOUR SESSION WAS CANCELED.
 A session has been canceled because the timeout wait interval was
 exceeded.

MSO13206 FOCUS SESSION HAS ENDED
 A FOCUS session has ended normally.

MSO13207 FOCUS SESSION ABENDED: %1
 A FOCUS session has abnormally terminated with the given completion code.

MSO13208 INVALID LOGON ATTEMPT TO MSO. PLEASE PRESS ENTER KEY.

Error Messages

A FOCUS logon was rejected because of a security violation. Check CICS userid and MSO security options.

MSO13209 SESSION CANCELED: TRANSMISSION EXCEEDED BUFFER SIZE
A datastream larger than expected has been received. Please contact your IBI representative.

MSO13210 SESSION HAS BEEN FORCED OFF BY OPERATOR.
A FOCUS session has been canceled by an operator console command.

MSO13211 SESSION REJECTED BECAUSE OF DUPLICATE LOGON IDS.
A FOCUS logon was rejected because the logon id was already logged on and UNIQUE=LOGID was specified.

MSO13230 INVALID COMMAREA DETECTED
A COMMAREA was detected in a format inconsistent with IBI internal usage.

MSO13231 SCREEN RECEIVE ERROR. RESPONSE: %1.
An unexpected condition occurred as a result of a 3270 read. Please contact your IBI support representative.

MSO13233 GETMAIN FAILURE ESTABLISHING MSO ENVIRONMENT.
An error occurred trying to allocate the storage required to start an MSO. Analyze CICS storage requirements.

MSO13234 GETMAIN FAILURE ESTABLISHING INDIVIDUAL USER.
An error occurred trying to allocate the storage required to start an MSO terminal user. Analyze CICS storage requirements.

MSO13235 LOGON FAILED: TERMID FOUND ALREADY IN USE.
An attempt to start a FOCUS session was received, but the terminal was found to be in use by another FOCUS session. If this is not the case, please contact your IBI support representative.

MSO13236 SESSION TERMINATED: UNRECOGNIZED ERROR CONDITION.
A FOCUS session has ended but the error condition is unrecognized. Please contact your IBI support representative.

MSO13237 SCREEN INPUT EXCEEDS BUFFER SIZE.
The amount of screen input exceeds the available buffer size. Please contact your IBI support representative.

MSO13238 ERROR IN TERMINAL SEND. RESPONSE: %1.
An unexpected condition occurred as a result of a 3270 send. Please contact your IBI support representative with the given response.

MSO13239 PLEASE WAIT FOR INITIALIZATION TO COMPLETE.
This is an informational message displayed before the initial TOE window arrives from FOCUS.

MSO13240 BREAK KEY PRESSED. PRESS %1 AGAIN TO EXIT MSO.

Error Messages

The user break key was pressed. If pressed again the FOCUS session is ended. If another key is pressed the FOCUS session is resumed.

MSO13241 ERROR STARTING MSO: %1. COMMUNICATION DATASET NOT FOUND.
A communications dataset corresponding to the given MSO was not found. See the MSO installation guide for more information.

MSO13242 ERROR STARTING MSO: %1. COMMUNICATIONS SUBTASK ERROR.
An error occurred starting the communications subtasks for the named MSO. Check for possible misallocation of FOCLIB.

MSO13243 LINK TO USERID EXIT (MSCXUID) FAILED. RESPONSE: %1
The presence of an MSCXUID exit was detected but the EXEC CICS LINK to it failed. Consult the appropriate CICS documentation to interpret the response.

MSO13244 XCTL TO TERMINATION EXIT (MSCXWRT) FAILED. RESPONSE: %1
The presence of an MSCXWRT exit was detected but the EXEC CICS XCTL to it failed. Consult the appropriate CICS documentation to interpret the response.

Messages for MSOUTIL (Transaction MSMT)

MSO13300 MSO: %1 WAS SUCCESSFULLY STOPPED
The named MSO was successfully stopped. All users have been logged off and communication with the MSO region has been severed.

MSO13301 MSO: %1 WAS ALREADY IN A STOPPED STATE
An attempt was made to stop the named MSO, which was found already to be in a stopped state.

MSO13302 STOP FAILED. MSO: %1 NOT FOUND
A stop command was issued for an MSO not known to the system.

MSO13303 ARGUMENTS FOR KEYWORD: %1 NOT FOUND OR INCORRECTLY SPECIFIED
The arguments required for the specified command are incorrectly specified. See the MSO installation guide for details.

MSO13304 KEYWORD: %1 NOT A RECOGNIZED COMMAND
The specified keyword is not a known command.

MSO13305 SUPERFLUOUS TEXT AFTER COMMAND: %1 IGNORED
Trailing text for the specified command has been ignored.

MSO13306 NO KEYWORDS FOR TRANSACTION: %1 FOUND
The utility transaction was invoked without a keyword command.

MSO13307 COMMAND: %1 INVALID BEFORE SYSTEM INITIALIZATION
The specified command can not be run before the MSO communications environment has been initialized.

Sample JCL & Source Code

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

Sample 1. CICSJCL: Sample CICS JCL

```

//*** Your JOB card
//*-----*
//*          EXECUTE CICS          *
//*-----*
//CICS      EXEC PGM=DFHSIP,REGION=6144K,TIME=1440,
// PARM=( 'SIT=7$,ISC=YES,XTP=YES,EXEC=YES,APPLID=TESTCICS, ',
//          'OSCOR=307200,ICV=500,SI' )
//SYSIN     DD *
TCT=NO
GRPLIST=EWLIST
EXTSEC=YES
SYSIDNT=TEST
.END
/*
//*-----*
//*          DFHRPL CONTAINS CICS MANAGEMENT PROGRAMS          *
//*          AND TABLES THAT ARE LOADED BY CICS/OS/VS          *
//*          AS WELL AS APPLICATION PROGRAMS                    *

```

Sample JCL & Source Code

```
//*-----*
//DFHRPL DD DSN=prefix.MSO.LOAD,
//      DISP=SHR,DCB=BLKSIZE=23476
//      DD DSN=CICSVS.LOADLIB2,DISP=SHR
//      DD DSN=CICSVS.LOADLIB,
//      DISP=SHR
//*-----*
//*          FOCLIB contains MVS subtasks and programs          *
//*          that are loaded by MSO/CICS                          *
//*-----*
//FOCLIB DD DSN=prefix.FOCLIB.LOAD,DISP=SHR
//*-----*
//*          ERRORS contains error messages                      *
//*          that are issued by MSO/CICS                          *
//*-----*
//ERRORS DD DSN=prefix.ERRORS.DATA,DISP=SHR
//*-----*
//*          CICS STEPLIB - CONTAINS MODULES LOADED BY          *
//*          SUPERVISOR SERVICES                                *
//*-----*
//STEPLIB DD DSN=CICSVS.LOADLIB1,DISP=SHR
//*-----*
//*          TEMP STORAGE DATASET                                *
//*-----*
//DFHTEMP DD DSN=CICSVS.TEST.DFHTEMP,
//      DISP=OLD
//*-----*
//*          INTRA-PARTITION DATA SET (VSAM)                    *
//*-----*
//DFHINTRA DD DSN=CICSVS.TEST.DFHINTRA,
//      DISP=OLD
//*-----*
//*          EXTRAPARTITION DATA SETS - DEFINED IN SAMPLE DFHDCT2$ *
//*-----*
//LOGUSR DD SYSOUT=*
//MSGUSR DD SYSOUT=*
//PLIMSG DD SYSOUT=*
//*-----*
//*          CICS DUMP DATA SET(S) AND SYSUDUMP                *
//*-----*
//DFHSNAP DD DUMMY
```

Sample JCL & Source Code

```
//DFHDMPA DD DSN=CICSVS.DUMPA,DISP=SHR
//DFHDMPB DD DSN=CICSVS.DUMPB,DISP=SHR
//SYSUDUMP DD DUMMY
//*-----*
/*          SYSTEM DEFINITION FILE          *
/*-----*
//DFHCSD DD DSN=CICSVS.DFHCSD,DISP=SHR
//*-----*
/*          SAMPLE PROGRAMS FILE          *
/*-----*
//FILEA DD DSN=CICSVS.FILEA,DISP=SHR
//*-----*
/*          RESTART DATASET          *
/*-----*
//DFHRSD DD DSN=CICSVS.TEST.DFHRSD,DISP=SHR
//*-----*
/*          AUXILIARY TRACE DATA SETS          *
/*-----*
//DFHAUXT DD DSN=CICSVS.TEST.TRACEA,DISP=SHR
//DFHBUXT DD DSN=CICSVS.TEST.TRACEB,DISP=SHR
//*-----*
/*          AUTO STATISTICS DATA SETS - DEFINED IN SAMPLE DFHDCT2$          *
/*-----*
//DFHSTM DD DSN=CICSVS.STATA,DISP=SHR
//DFHSTN DD DSN=CICSVS.STATB,DISP=SHR
//*-----*
/*          MSO/CICS communication data sets          *
/*-----*
//MSOPUT DD DSN=prefix.MSO.MSPUT,DISP=SHR
//MSOGET DD DSN=prefix.MSO.MSGET,DISP=SHR
```

Sample 2. MSLNKSTB: JCL to link-edit CICS command-level stubs into MSO programs

```
// JOB <-- complete jobcard
/*
/* THIS JOB RELINKS MSO/CICS THE MODULES WITH THE CICS COMMAND LEVEL
/* STUBS FOR YOUR SHOP'S RELEASE OF CICS. UPDATE THE DATASET NAMES TO
/* REFLECT THE DESIRED RELEASES OF CICS AND FOCUS.  OLDMOD AND SYSLMOD
/* MAY POINT TO THE SAME LIBRARY.
/*
/******
```

Sample JCL & Source Code

```
/*
//MSLNKSTB EXEC PGM=IEWL,REGION=1024K,PARM='LET,NCAL'
//SYSUT1 DD UNIT=SYSDA,DCB=BLKSIZE=1024,
//      SPACE=(1024,(200,20))
//SYSPRINT DD SYSOUT=*
//CICSLIB DD DISP=SHR,DCB=BLKSIZE=80,DSN=cics.loadlib <-- update
/*                                          dataset
//OLDMOD DD DISP=SHR,DSN=prefix.MSO.LOAD <-- update dataset
//MAINTAIN DD DISP=SHR,DSN=prefix.MSO.DATA <-- update dataset
//SYSLMOD DD DISP=OLD,DSN=prefix.MSO.LOAD <-- update dataset
//SYSLIN DD *
        INCLUDE CICSLIB(DFHEAI)
        INCLUDE CICSLIB(DFHEAI0)
        INCLUDE OLDMOD(MSOCICS)
        INCLUDE MAINTAIN(MSOCICS)
        NAME MSOCICS(R)
        INCLUDE CICSLIB(DFHEAI)
        INCLUDE CICSLIB(DFHEAI0)
        INCLUDE OLDMOD(MSOUTIL)
        INCLUDE MAINTAIN(MSOUTIL)
        NAME MSOUTIL(R)
        INCLUDE CICSLIB(DFHEAI)
        INCLUDE CICSLIB(DFHEAI0)
        INCLUDE OLDMOD(MSOTOR)
        INCLUDE MAINTAIN(MSOTOR)
        NAME MSOTOR(R)
        INCLUDE CICSLIB(DFHEAI)
        INCLUDE CICSLIB(DFHEAI0)
        INCLUDE OLDMOD(MSOATTN)
        INCLUDE MAINTAIN(MSOATTN)
        NAME MSOATTN(R)
        INCLUDE CICSLIB(DFHEAI)
        INCLUDE CICSLIB(DFHEAI0)
        INCLUDE OLDMOD(MSOPLT)
        INCLUDE MAINTAIN(MSOPLT)
        NAME MSOPLT(R)
/*
```

Sample JCL & Source Code

Sample 3. MSXZCATJ: JCL to assemble and link-edit MSOEXIT (XZCATT exit)

```
/** Your JOB card
/*
/* This job assembles and links the MSOEXIT program, which is to be
/* used as an XZCATT global user exit for SNA Attn support. Update
/* the indicated datasets with the proper high-level qualifiers.
/*
/******
/*
/*ASM      EXEC PGM=IEV90,REGION=1024K,PARM=(NODECK,OBJECT,LIST)
//SYSUT1   DD UNIT=SYSDA,SPACE=(1024,(120,120))
//SYSLIB   DD DSN=cics.MACLIB,DISP=SHR      <--- update dataset
//         DD DSN=SYS1.MACLIB,DISP=SHR
//SYSLIN   DD DSN=&&OBJ,SPACE=(3040,(40,40)),
//         UNIT=SYSDA,DISP=(NEW,PASS)
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DSN=prefix.MSO.DATA(MSOEXIT),DISP=SHR <-- update dataset
/*
/*
//LINK     EXEC PGM=IEWL,REGION=1024K,PARM='NCAL'
//SYSUT1   DD UNIT=SYSDA,SPACE=(1024,(120,120))
//SYSPRINT DD SYSOUT=*
//SYSLMOD  DD DISP=SHR,DSN=prefix.MSO.LOAD      <-- update dataset
//SYSLIN   DD DSN=&&OBJ,DISP=(OLD,DELETE)
//         DD DDNAME=SYSIN
//SYSIN    DD *
           MODE AMODE(31),RMODE(ANY)
           ENTRY MSOEXIT
           NAME MSOEXIT(R)
/*
```

Sample 4. MSCXUIDJ: JCL to assemble and link-edit MSCXUID

```
/** Your JOB card
/*
/* This job translates, assembles, and links the MSCXUID user exit.
/* The IBI datasets MSO.LOAD and MSO.DATA are assumed to be present.
/* The source code for the sample MSCXUIDS is supplied in MSO.DATA
/******
//TRN     EXEC PGM=DFHEAP1$,
//         REGION=1024K
```

Sample JCL & Source Code

```
//STEPLIB DD DSN=CICSVS.LOADLIB,DISP=SHR
//SYSPRINT DD SYSOUT=S
//SYSPUNCH DD DSN=&&SYSCIN,
//      DISP=(,PASS),UNIT=SYSDA,
//      DCB=BLKSIZE=400,
//      SPACE=(400,(400,100))
//SYSIN DD DSN=prefix.MSO.DATA(MSCXUIDS),DISP=SHR
//ASM EXEC PGM=IEV90,
//      REGION=1024K,
//      PARM='DECK,NOOBJECT,LIST'
//SYSLIB DD DSN=CICSVS.MACLIB,DISP=SHR
//      DD DSN=SYS1.MACLIB,DISP=SHR
//SYSUT1 DD UNIT=SYSDA,SPACE=(1700,(400,400))
//SYSUT2 DD UNIT=SYSDA,SPACE=(1700,(400,400))
//SYSUT3 DD UNIT=SYSDA,SPACE=(1700,(400,400))
//SYSPUNCH DD DSN=prefix.MSO.DATA(MSCXUID),DISP=SHR
//SYSPRINT DD SYSOUT=S
//SYSIN DD DSN=&&SYSCIN,DISP=(OLD,DELETE)
//LINK EXEC PGM=IEWL,REGION=1024K,
//      PARM='LET,AMODE=31,RMODE=ANY,NCAL'
//SYSUT1 DD UNIT=SYSDA,DCB=BLKSIZE=1024,
//      SPACE=(1024,(200,20))
//SYSPRINT DD SYSOUT=*
//OBJLIB DD DISP=SHR,DSN=PREFIX.MSO.DATA
//CICSLIB DD DISP=SHR,DSN=CICSVS.LOADLIB
//SYSLMOD DD DISP=SHR,DSN=PREFIX.MSO.LOAD
//SYSLIN DD *
        INCLUDE CICSLIB(DFHEAI)
        INCLUDE CICSLIB(DFHEAI0)
        INCLUDE OBJLIB(MSCXUID)
        MODE AMODE(31),RMODE(ANY)
        ORDER DFHEAI,DFHEAI0,MSCXUID
        ENTRY MSCXUID
        NAME MSCXUID(R)
/*
```

Sample 5. MSCXUIDS: Sample MSCXUID source

```
*-----*
*
* This is a sample userid exit for MSO. MSO normally obtains the
* userid under CICS by executing "EXEC CICS ASSIGN USERID". When
* such a request will not provide correct results, this exit may be
* installed to obtain the userid correctly.
*
* This program is called via 'EXEC CICS LINK' from transaction MSO
* and is passed an eight-byte COMMAREA. This exit is responsible
* for placing the correct userid into the COMMAREA (padded with
* blanks). Whatever is returned will be used by MSO as the userid
* for signon.
*
*-----*
          SPACE
*-----*
* WORKING STORAGE AREA.
*-----*
          SPACE
DFHEISTG DSECT
          DS CL4 ANY FIELDS NEEDED BY THE EXIT
MSCXUID CSECT END OF WORKING STORAGE
          SPACE
*-----*
* PROGRAM INTRO. POINT TO THE EIGHT-BYTE COMMAREA FIELD, WHERE THE
* USERID MUST BE RETURNED.
*-----*
          SPACE
          L R6,DFHEICAP GET ADDRESS OF COMMAREA FIELD
          USING XUSRID,R6 COVER DSECT
          SPACE
XUSRID DSECT THE COMMAREA DSECT
MSOUSER DS CL8 RETURN THE USERID HERE
MSCXUID CSECT
          ORG
          SPACE
*-----*
* THIS SAMPLE EXIT RETURNS THE CICS 'OPID' FIELD AS THE ID TO
*-----*
```

Sample JCL & Source Code

```
* BE SUPPLIED TO MSO. THIS FIELD MAY BE FURTHER MANIPULATED BY *
* MSO VIA THE MSIDTR USERID TRANSLATION EXIT FOR MSO. *
*-----*
      SPACE
      EXEC CICS ASSIGN OPID(MSOUSER)
      SPACE
R0     EQU  0
R1     EQU  1
R2     EQU  2
R3     EQU  3
R4     EQU  4
R5     EQU  5
R6     EQU  6
R7     EQU  7
R8     EQU  8
R9     EQU  9
R10    EQU 10
R11    EQU 11
R12    EQU 12
R13    EQU 13
R14    EQU 14
R15    EQU 15
      END
```

Sample 6. MSCXWRTJ: JCL to assemble and link-edit MSCXWRT

```
/** Your JOB card
/**
/** This job translates, assembles, and links the MSCXWRT user exit.
/** The IBI datasets MSO.LOAD and MSO.DATA are assumed to be present.
/** The source code for the sample MSCXWRTS is supplied in MSO.DATA
/**-----**
//TRN    EXEC PGM=DFHEAP1$,
//      REGION=1024K
//STEPLIB DD DSN=CICSVS.LOADLIB,DISP=SHR
//SYSPRINT DD SYSOUT=S
//SYSPUNCH DD DSN=&&SYSCIN,
//      DISP=(,PASS),UNIT=SYSDA,
//      DCB=BLKSIZE=400,
//      SPACE=(400,(400,100))
//SYSIN  DD DSN=prefix.MSO.DATA(MSCXWRTS),DISP=SHR
//ASM    EXEC PGM=IEV90,
```

Sample JCL & Source Code

```
//      REGION=1024K,
//      PARM='DECK,NOOBJECT,LIST'
//SYSLIB DD DSN=CICSVS.MACLIB,DISP=SHR
//      DD DSN=SYS1.MACLIB,DISP=SHR
//SYSUT1 DD UNIT=SYSDA,SPACE=(1700,(400,400))
//SYSUT2 DD UNIT=SYSDA,SPACE=(1700,(400,400))
//SYSUT3 DD UNIT=SYSDA,SPACE=(1700,(400,400))
//SYSPUNCH DD DSN=prefix.MSO.DATA(MSCXWRT),DISP=SHR
//SYSPRINT DD SYSOUT=S
//SYSIN   DD DSN=&&SYSCIN,DISP=(OLD,DELETE)
//LINK    EXEC PGM=IEWL,REGION=1024K,
//      PARM='LET,AMODE=31,RMODE=ANY,NCAL'
//SYSUT1  DD UNIT=SYSDA,DCB=BLKSIZE=1024,
//      SPACE=(1024,(200,20))
//SYSPRINT DD SYSOUT=*
//OBJLIB  DD DISP=SHR,DSN=PREFIX.MSO.DATA
//CICSLIB DD DISP=SHR,DSN=CICSVS.LOADLIB
//SYSLMOD DD DISP=SHR,DSN=PREFIX.MSO.LOAD
//SYSLIN DD *
        INCLUDE CICSLIB(DFHEAI)
        INCLUDE CICSLIB(DFHEAI0)
        INCLUDE OBJLIB(MSCXWRT)
        MODE AMODE(31),RMODE(ANY)
        ORDER DFHEAI,DFHEAI0,MSCXWRT
        ENTRY MSCXWRT
        NAME MSCXWRT(R)
/*
```

Sample 7. MSCXWRTS: Sample MSCXWRT source

```
        TITLE 'SAMPLE MSCXWRT CICS/MSO EXIT FROM MSO'
*-----*
*
* This is a sample exit program which is given control when a user
* is exiting MSO and returning to CICS. This exit may be used in
* place of, and replaces, the "next transid" which MSO will set.
*
* The advantage of this exit is simply to bypass the final message
* which MSO normally displays while it is allowing the "next
* transid" field to be reset within CICS command-level standards.
* It avoids the requirement of having to press Enter to invoke the
* selected "next transid" transaction. However, in cases where an
```

Sample JCL & Source Code

```
* error message is to be presented to the user, MSO will still
* pause to allow the user to see the message, before this program
* is given control.
*
* This program is invoked via EXEC CICS XCTL from the MSO
* transaction. The next-transid field should be assumed to be
* filled in with an unwanted value upon entry.
*
* A sixteen-byte COMMAREA is passed; the first 8 bytes are the
* userid which CICS passed to MSO; the second 8 bytes are the
* security userid used within the MSO session, which may be
* identical to the first userid. ID). When a value is unknown, it
* is passed as blanks; if both are unknown, the whole COMMAREA is
* filled with blanks.
*
*-----*
*          SPACE
*-----*
*  WORKING STORAGE AREA.
*-----*
*
*          SPACE
DFHEISTG DSECT
*          DS    CL4          ANY FIELDS NEEDED BY THE EXIT
MSCXWRT  CSECT          END OF WORKING STORAGE
*          SPACE
*-----*
*  PROGRAM INTRO.  POINT TO THE 16-BYTE COMMAREA FIELD, WHERE THE
*  SIGNON AND SECURITY IDS ARE PASSED TO US.
*-----*
*
*          SPACE
*          L      R6,DFHEICAP    GET ADDRESS OF COMMAREA FIELD
*          USING XUSRID,R6      COVER DSECT
*          SPACE
XUSRID   DSECT          THE COMMAREA DSECT
SIGNON   DS    CL8      USERID USED TO SIGNON TO MSO
SECURE   DS    CL8      SECURITY ID USED UNDER MSO
MSCXWRT  CSECT
*          SPACE
*-----*
*  START THE NEXT TRANSACTION WHICH WILL BEGIN IMMEDIATELY UPON
*  TERMINATION OF FOCUS/MSO.
*-----*
```

Sample JCL & Source Code

```

*-----*
      SPACE
      LA  R7,EIBTRMID          GET CURRENT TERMD TO START TRANS
      EXEC CICS HANDLE CONDITION
                                SYSIDERR(ERR) IOERR(ERR) ISCINVREQ(ERR)
                                INVREQ(ERR) NOTAUTH(ERR) TERMIDERR(ERR)
                                TRANSIDERR(ERR) PGMIDERR(ERR)
      EXEC CICS START TRANSID('ABCD') TERMD(0(R7))
                                FROM(0(R6)) LENGTH(16) NOCHECK
ERR    DS  0H
      EXEC CICS IGNORE CONDITION
                                SYSIDERR      IOERR      ISCINVREQ
                                INVREQ        NOTAUTH     TERMIDERR
                                TRANSIDERR    PGMIDERR
*-----*
*  THERE IS A VALUE IN THE 'NEXT TRANSID' FIELD OF THE TCTTE WHICH
*  WE WON'T WANT TO BE USED.  WE
*  CAN EITHER PASS A NEXT TRANSID IN THE RETURN BELOW, OR, WE CAN
*  DO SOMETHING A LITTLE NON-STANDARD, WHICH WILL WORK IN CICS 1.7
*  WHERE THE TCTTE CAN BE RELIABLY FOUND FROM THE TCA WHICH CAN BE
*  RELIABLY FOUND FROM THE CSA.  THIS LITTLE TRICK IS NOT GUARANTEED
*  TO WORK EVERYWHERE.  IF THIS PIECE OF CODE DOES NOT WORK, THEN
*  THE TRANSID('XXXX') PARAMETER SHOULD BE ADDED TO THE EXEC CICS
*  RETURN CALL BELOW.
*
*  BY THE WAY, THE ADDRESS CSA() PARAMETER CAUSES A RETURN CODE OF 4
*  IN BOTH THE CICS TRANSLATOR STEP AND THE ASSEMBLY STEP.  THIS
*  RETURN CODE CAN BE IGNORED.
*-----*
      EXEC CICS ADDRESS CSA(R6)
      L   R7,CSACDTA-DFHCSADS(,R6)      POINT TO THE TCA
      ICM R7,B'1000',=X'00'             CLEAR HI BYTE
      L   R8,TCAFCAAA-DFHTCADS(,R7)     POINT TO THE TCTTE
      ICM R8,B'1000',=X'00'             CLEAR HI BYTE
      LA  R9,TCTTETC-TCTTETI(,R8)       POINT TO 'NEXT TRANSID'
      XC  0(4,R9),0(R9)                 WIPE IT OUT
      EXEC CICS RETURN
      SPACE
R0     EQU  0
R1     EQU  1
R2     EQU  2

```

Sample JCL & Source Code

```
R3      EQU  3
R4      EQU  4
R5      EQU  5
R6      EQU  6
R7      EQU  7
R8      EQU  8
R9      EQU  9
R10     EQU 10
R11     EQU 11
R12     EQU 12
R13     EQU 13
R14     EQU 14
R15     EQU 15
TCTTEAR EQU R6
        COPY DFHCSADS
        COPY DFHTCADS
        COPY DFHTCTTE
        END
```

Sample 8. CMSOTABL: Table of site-modifiable defaults

```
*-----*
* Copyright (c) 1994 Information Builders, Inc. All rights reserved.
*
* _Name_      ==> CMSOTABL ASSEMBLE   _Opsys_      ==> TSO
* _Release_   ==> 70                 _Product_    ==> SST
* _Description_ ==>
*             ==>
* _Notes_     ==>
*
* _History_:
* Date  Time Who Proj      Project Title
* =====
* 940819 1534 ANH 14088 MSO/CICS: MAKE INITIALIZATION MESSAGE OPTIONAL
* 940803 0914 ANH 13750 MSO/CICS: ADD MAIN CONTROL BLOCK ANCHORING SCHE
* 940624 1626 ANH 13498 MSO/CICS: IMPLEMENT MSCXUID, MSCXWRT AND NEXTRAN
*
* END %&§
*-----*
*-----*
* CMSOTABL - Table of user modifiable defaults. After changes are *
* made, link into modules MSOCICS and MSOUTIL using the JCL provided *
```

Sample JCL & Source Code

```
* In MSO.DATA(MSOTABLJ). Retain a copy of the original for reference.*
*-----*
CMSOTABL CSECT ,          OFFSET
*-----*
NEXTTRAN DC    CL4' '      +0          TRANSACTION TO RUN AFTER MSO
USERAUX# DC    H'32'       +4          USER AUXTRACE NUMBER
INITMSG  DC    C'Y'        +6          DISPLAY INITIALIZATION MSG(Y/N)
         DS    X            +7          RESERVED FOR FUTURE USE
         DS    10F         +8          RESERVED FOR FUTURE USE
END
```

Sample 9. MSOTABLJ: JCL to assemble and link-edit CMSOTABL

```
/** Your JOB card
/**
/** This job assembles and links the CMSOTABL CSECT, which supplies
/** runtime defaults to MSO/CICS.
/**
/**-----**
/**
/**ASM      EXEC PGM=IEV90,REGION=1024K,PARM=(NODECK,OBJECT,LIST)
/**SYSUT1   DD UNIT=SYSDA,SPACE=(1024,(120,120))
/**SYSLIN   DD DSN=&&OBJ,SPACE=(3040,(40,40)),
/**          UNIT=SYSDA,DISP=(NEW,PASS)
/**SYSPRINT DD SYSOUT=*
/**SYSIN    DD DSN=prefix.MSO.DATA(CMSOTABL),DISP=SHR <-- update dataset
/**
/**
/**LINK     EXEC PGM=IEWL,REGION=1024K,PARM='NCAL,RENT,REUS'
/**SYSUT1   DD UNIT=SYSDA,SPACE=(1024,(120,120))
/**SYSPRINT DD SYSOUT=*
/**SYSLMOD  DD DISP=SHR,DSN=prefix.MSO.LOAD          <-- update dataset
/**OBJECT   DD DSN=&&OBJ,DISP=(OLD,DELETE)
/**SYSLIN   DD *
INCLUDE OBJECT
INCLUDE SYSLMOD(MSOCICS)
INCLUDE MAINTAIN(MSOCICS)
NAME      MSOCICS(R)
INCLUDE OBJECT
INCLUDE SYSLMOD(MSOUTIL)
INCLUDE MAINTAIN(MSOUTIL)
NAME      MSOUTIL(R)
```

Sample JCL & Source Code

/*

Sample 10. MSOTORJ: Sample zap of MSOTOR to change MSOATTN's transid

```
/** Your JOB card
/**
/** This job changes the name of the transaction which MSOTOR will
/** start in response to an SNA Attention.
/**
/** The IBI default, MSAT, may be changed to any available transaction
/** name, which must be defined in the CSD to execute program MSOATTN
/** in the AOR running the MSO transactions.
/**
/** Substitute your the appropriate prefix on the SYSLIB card and
/** the hex representation of the desired transaction on the REP
/** statement.
/*******
/**
//ZAP      EXEC PGM=AMASPZAP,REGION=1024K
//SYSLIB   DD DSN=prefix.MSO.LOAD,DISP=SHR  <--- update dataset
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
NAME MSOTOR MSOTOR
VER  0190 D4E2,C1E3  C'MSAT'      <--- IBI supplied default
REP  0190 4040,4040  C'      '    <--- substitute your transaction
/**
```