Mainframe Access Installation Guide

Micro Focus
The Lawn
22-30 Old Bath Road
Newbury, Berkshire RG14 1QN
UK
http://www.microfocus.com

2019-11-06

Contents

Mainframe Access Installation Guide

Introduction

Mainframe Access is the OS/390 and z/OS server for Micro Focus development environments. It is a common component providing access to host resources for environments. One installation of Mainframe Access can support all of these products, providing connectivity to any number of DB2, IMS, and CICS systems located anywhere in an enterprise. It can also provide access to JES facilities, VSAM data sets, non-VSAM data sets, and data controlled by external library management products such as Panvalet, Librarian and Endevor.

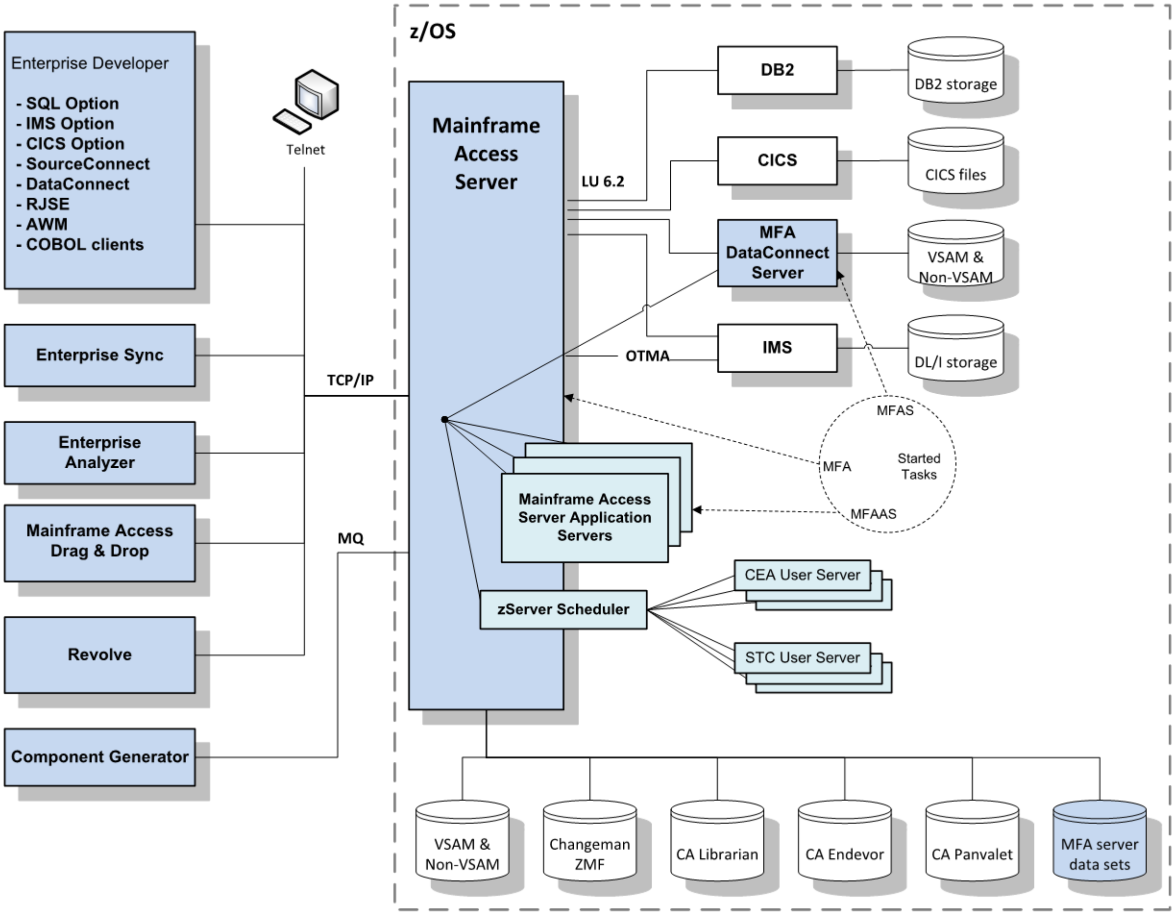
Note: Mainframe Access is supported on z/OS and OS/390 environments. Unless specifically stated otherwise, references to z/OS also apply to OS/390.

Mainframe Access is data and transaction middleware that exploits the best features of workgroup and enterprise architectures. Micro Focus client programs can achieve the same performance levels, security, and data integrity that native mainframe applications deliver. The technology requires no changes to existing mainframe security, database, and transaction processing systems.

Mainframe Access provides access to:

- DB2 for z/OS and OS/390 from clients.
- CICS transactions and data from clients.
- IMS data from clients via the Remote IMS Serve.
- Non-relational data sets (VSAM, sequential, partitioned, etc.) from Mainframe Access Client.
- Mainframe source control systems the Mainframe Access Client.
- JES2 and JES3 from the Mainframe Access Client.

The figure below illustrates the relationships between Mainframe Access, its client systems and the z/OS services it provides.



Communications

Mainframe Access provides comprehensive support for communications between PC clients and mainframe servers. Mainframe Access supports both TCP/IP and LU6.2 protocols. Mainframe Access' ability to bridge client TCP/IP connections to SNA LU6.2 server connections eliminates the need to install, configure and maintain SNA software packages on your client and server workstations. You can even access any DB2 for z/OS and OS/390 database using TCP/IP client connections, without the need for specific levels of DB2 software. Normally, DB2 Version 5.1, or later, is required for TCP/IP access. Mainframe Access, however, dynamically associates TCP/IP client connections with traditional LU6.2 DRDA connections to the DB2 systems.

Details are as follows:

- **TCP/IP Communications**

  Mainframe Access has two TCP/IP socket listeners, one for TCP/IP clients using proprietary protocols and one for HTTP clients. The listeners accept client connection requests and examine the request data to determine the target server or internal destination for the request. When the target server is external, the client connection is then associated with an LU6.2 connection to the target server. For each target server type, Mainframe Access creates an initial number of tasks (z/OS TCBs) to handle the client/server connection association and processing. As each connection request arrives, Mainframe Access scans all available tasks for the target server and assigns the new connection request to the task with the lightest load. Mainframe Access' TCP/IP communications support can be used with IBM's z/OS TCP/IP implementation.
- **LU6.2 Communications**

When the first client request for an LU6.2 target server is received, Mainframe Access establishes the LU6.2 connection to the target server and negotiates session limits for the mode name associated with conversations (user transactions) to the target server. Mainframe Access then prepares an LU6.2 Function Management Header 5 (FMH-5) and allocates a conversation to the target server. The FMH-5 includes any LU6.2 security subfields (user ID and password) provided by the client and an LUW ID (SNA Logical Unit of Work ID) generated by Mainframe Access to manage the unit of work. Packaged user data to be processed by the transaction program follows the FMH-5. Acceptance of the conversation request by the target server starts the execution of the remote transaction program that will process the request. Response data from the remote transaction program is received by Mainframe Access and returned to the requesting client through the TCP/IP connection. If an LU6.2 communications error occurs during the request processing, Mainframe Access creates a special packet containing the LU6.2 error information and returns this to the requesting client.

Security

Mainframe Access provides security features that interact with existing mainframe, operating system, and software security schemes. They are:

- **System Authorization Facility (SAF) interface**

  Mainframe Access' security component uses the z/OS SAF interface to verify the user IDs and passwords and to check the authorization of users to access and update z/OS data sets. The Password Expiration Manager (PEM) feature of Mainframe Access is used by some client products; PEM allows PC end users to change their mainframe password using the SAF interface. Mainframe Access uses the SAF RACROUTE interface, and therefore exploits the existing security definitions in your external security manager, such as IBM's RACF, CA-ACF-2, and CA-Top Secret.
- **Encryption**

  Mainframe Access and its associated products use several methods of encryption and encoding, including 32-bit private key algorithms, to manage the encryption and decryption of passwords and data.
- **Access List Checking**

  You can use this optional feature to restrict client connections based on the IP address of the incoming client. You create an access list that specifies which client IP addresses, or ranges of addresses, are allowed or disallowed. Once you have enabled access list checking, Mainframe Access always validates the incoming client's IP address using your specifications and rejects unauthorized connection attempts.
- **Audit Logging**

  Mainframe Access' optional audit logging facility records client login and logout information to a VSAM file. The information that is recorded includes user ID, login and logout times, target server, security status, and other data.

Data Set Services

Mainframe Access' Data Set Services component uses state-of-the-art z/OS data management interfaces and facilities to provide file services for z/OS data sets. Data Set Services is designed to support the data access and security requirements of VSAM and non-VSAM data sets in a high-performance, multi-user environment.

Administration

Mainframe Access provides functions to monitor and control the processing of work, for example:

- **Client Timeout Management**

  Mainframe Access monitors the activity of clients that have a persistent connection to Mainframe Access for activity and disconnects them from the system after a length of idle time that you specify. It also releases all of the associated client resources on the mainframe, including LU6.2 conversations with server systems.

- **Trace Facility**

  The tracing facility of Mainframe Access traces both the client flow (usually TCP/IP) and server flow (usually LU6.2) simultaneously, recording all control information and data to a single output destination. The trace information is very useful for diagnosing client/server communication problems. You can turn tracing on and off dynamically while Mainframe Access is running.

Mainframe Access provides a set of simple commands that you can use to monitor and control its run-time operations. You can use the commands from an z/OS console or from a Telnet client; for further information see the chapter *Administration Using Console Commands*.

You can use Mainframe Access' administrative interfaces to obtain comprehensive displays of current system activity and internal components of the system. The information provided extends from overall systems operation down to details about individual client/server connections. You can update important operational definitions and functions of the system, such as target server parameters, idle timeout value, access list specifications and trace activity while the system is running.

Dependent Address Space Services

Mainframe Access' Dependent Address Space Services component uses standard z/OS facilities to create and manage auxiliary Application Server address spaces for program execution. This feature is used to automatically start a Mainframe Access Data Connect server address space during Mainframe Access initialization. The Data Connect server is also stopped automatically during Mainframe Access shutdown.

Mainframe Access Services

Mainframe Access Drag & Drop and Source Connect functions are seamlessly integrated into the overall server workload. Mainframe Access' high-performance task, communications, transaction and data set management services are leveraged to increase the speed, reliability and scalability for these functions.

z/Server Feature (deprecated)

Note: This is deprecated, and provided for backward compatibility only.

The z/Server scheduler is no longer required in Enterprise Developer 5.0 as the MFA Server now supports z/Server functionality via the MFA TSO Application Server. If you are installing Mainframe Access Server 5.0 and all of your clients are running Enterprise Developer 5.0 then you do not need to configure the z/Server scheduler, instead you should configure the MFA TSO Application Server. See *Migration information* for more information.

z/Server is now deprecated, and marked as such in the documentation. The only reason to configure the z/Server scheduler is if support for clients earlier than Enterprise Developer version 5.0 is still required. Micro Focus recommends keeping all Enterprise Developer clients up to date and in sync with the Mainframe Access Server version.

If you have a mixed environment of clients running Enterprise Developer 5.0 and clients running Enterprise Developer 4.0 or earlier then you will need to configure both the MFA TSO application Server and the z/Server scheduler.

Migration information

The migration actions listed here are for the changes required when upgrading from Enterprise Developer 4.0 to Enterprise Developer 5.0.

The z/Server scheduler is no longer required in Enterprise Developer 5.0 as the MFA Server now supports z/Server functionality via the MFA TSO Application Server. If you are installing Mainframe Access Server 5.0 and all of your clients are running Enterprise Developer 5.0 then you do not need to configure the z/Server scheduler. Instead, you should configure the MFA TSO Application Server.

z/Server is now deprecated, and marked as such in the documentation. The only reason to configure the z/Server scheduler is if support for clients earlier than Enterprise Developer version 5.0 is still required. Micro Focus recommends that you keep all Enterprise Developer clients up to date and in sync with the Mainframe Access Server version.

Migrating from z/Server Scheduler to MFA TSO Application Server

When migrating from the z/Server Scheduler to the new MFA TSO Application Server the following steps must be applied:

- Configure the new MFA TSO Application Server.
- Any customization made to the z/Server IVPUSRT JCL procedure must be replicated in the hlq.MFA.CNTL(MFATSO) JCL procedure.

  Note: Any data sets added to the STEPLIB in hlq.MFA.CNTL(MFATSO) JCL procedure must be APF Authorized.
- Any customization made to the z/Server IVPINIT1 JCL procedure must be replicated in the hlq.MFA.EXEC(IVPINIT1) JCL procedure.
- Any customization made to the z/Server Master Configuration file (TAUZCAPP) must be replicated in the hlq.MFA.EXEC(TAUZCAPP) member.
- Remove the <Scheduler /> and any relevant z/Server configuration items, such as SVC_NO, from the MFAXML XML configuration.

Installation requirements

Machine requirements

IBM mainframe model zEC12 or later is required for installing and using MFA Server. If your hardware does not meet these requirements please contact Micro Focus support.

Software requirements

An IBM supported operating system z/OS (V2.1 or later) is required. If your operating system version does not meet these requirements please contact Micro Focus support.

Mainframe Access requirements:

- IBM Communications Server
- Two APPLIDs, and one TCP/IP port
- Availability of APF security authorization support personnel
- Access to a network share with acceptable space for source and data, as well as the ability to access the IP address and ports used to access MFA

Mainframe Access TSO Application Server requirements:

- ISPF
- Binder
- High Level Assembler

- Language Environment
- RACF or an equivalent product
- z/OS Communications Server - IP Services
- IBM REXX Library

To run user applications requested via TCP/IP client calls, there may be additional optional software requirements depending on the application. Typical optional requirements are:

- IBM Enterprise COBOL for z/OS
- IBM Enterprise PL/I
- IBM DB2 for z/OS

Support for Error Feedback (displaying compiler error messages in an Enterprise Developer error view) requires the following additional compiler options:

- For COBOL: EXIT(ADEXIT('/N 100 /W 50',TAURTOCX)), ADATA
- For PL/I: XINFO(XML)
- For Assembler: EXIT(ADEXIT(TAURTOAX('/N 100 /W 25'))),ADATA

To compile CICS applications, a compiler supporting the CICS Transaction Server 2.1 or later is required:

- COBOL applications compiled using any of:
  - IBM® COBOL for OS/390® &VM, Version 2 Release 2, program number 5648-A25, with APAR PQ49375 applied.
  - IBM Enterprise COBOL for z/OS and OS/390, Version 3 Release 1, program number 5655-G53, and later COBOL releases.
- PL/I applications compiled using any of:
  - IBM VisualAge PL/I for OS/390 Version 2 Release 2.1, program number 5655-B22, with APAR PQ45562 applied.
  - IBM Enterprise PL/I for z/OS and OS/390, Version 3 Release 1, program number 5655-H31, and later PL/I releases.

Recommended versions of z/Server for Enterprise Developer (deprecated)

Note: This is deprecated, and supported for backward compatibility only.

The following table shows the versions of z/Server delivered with the different Enterprise Developer releases. These are also the recommended version of z/Server to use with a particular Enterprise Developer release.

Enterprise Developer Version	z/Server Version	Date Released
2.1	z/Server V2R0M06 and V2R0M07 (including performance testing)	October 2012
2.1.1	z/Server V2R0M07	April 2013
2.2	z/Server V2R2M00	October 2013
2.2 Update 1	z/Server V2R2M02	May 2014
2.3	z/Server V2R3	September 2015
2.3 Update 1	z/Server V2R3M01	March 2016
3.0	z/Server V3R0	June 2017
4.0	z/Server V4R0	June 2018

Enterprise Developer Version	z/Server Version	Date Released
5.0	z/Server V5R0	June 2019

⚠ **Important:** An Enterprise Developer client can communicate with a version of z/Server from the same delivery or newer, but not vice versa. For example, an Enterprise Developer 2.3 client can communicate with z/Server delivered with Enterprise Developer 2.3, 2.3 Update 1, 3.0, 4.0, or 5.0.

Mainframe Access Configuration Utility

The Mainframe Access configuration utility is a web-based utility that simplifies the process of configuring the XML file that controls how MFA Server operates. The utility guides you through the steps required to create a new XML file, specifying MFA Start Task, Application Servers and Services elements. It validates the specified parameters and values ensuring that the XML file is well-formed. You can modify a basic configuration file adding advanced or custom parameters before uploading it to the mainframe ready for use with MFA Server. You can also download and modify existing XML files directly from the mainframe.

Features

The Mainframe Access configuration utility has the following features:

- Web-based user interface.
- Validation of XML configuration files.
- Parameter and value validation at the point of entry.
- Grouped parameters for holder, scheduler and user server configuration.
- Ability to upload or download configuration files directly to the mainframe.

Limitations

The Mainframe Access configuration utility has the following limitations:

- Supports configuration of XML files created with Enterprise Developer 4.0 or greater.
- Internet Explorer does not support saving the XML configuration file directly to the browser's download folder. The XML file is displayed in a browser popup. You can copy from the popup and paste to a new location.
- The web-based utility is restricted to password lengths that are 8 characters or fewer.

Prerequisites

The Mainframe Access configuration utility has the following software requirements:

- A Web browser (with a minimum required version):
  - Internet Explorer (version 11.0)
  - Firefox (version 39.0)
  - Chrome (version 43.0)

Starting the Mainframe Access Configuration Utility

To start the Mainframe Access configuration utility, perform the following steps:

1. Click **Start > All Programs > Micro Focus Enterprise Developer > Tools > Enterprise Developer Command Prompt**
2. At the command prompt, type `mfaconfig.bat`.
3. Press **Enter**.

This opens the Mainframe Access configuration utility in you default Web browser, with the URL `http://localhost:2345`.

Operation

The Mainframe Access configuration utility has two modes of operation:

- Creating a new XML configuration file. The interface takes you through the steps for specifying the MFA Server parameters as well as Application Server and Services parameters. Once you have specified the basic configuration file you can choose to:
  - Upload the file to a mainframe location or you can save it locally in your browser's download directory.
  - Edit your initial MFA Server XML configuration, found in hlq.MFA.CNTL(MFAXML), and customize it for your needs.
- Downloading an XML configuration file from a mainframe location. Once you have downloaded the configuration file you can choose to:
  - Upload the file to an alternative mainframe location or you can save it locally in your browser's download directory.
  - Review the downloaded configuration file. This is performed in stages:
    1. Review the existing MFA Server parameters.
    2. Review the existing RIMS, MCO, MFA and ES-MTO Services parameters.
    3. Review the existing DataConnect, Endevor, ChangeMan, AMS and z/Server Application Server parameters.
    4. Once you have reviewed the configuration you can upload the file to a mainframe location or you can save it locally in your browser's download directory.

Creating a new XML configuration file

To create a new XML configuration file, perform the following steps:

1. Click **Create**.
2. Review and modify the default MFA Server parameters, and then click **NEXT**.

   See *MFA Server Parameters* for more information.
3. Review and modify that Data Connect Application Server parameters, and then click **NEXT**.

   See *Data Connect Application Server Parameters* for more information.
   - Optionally, you can add more Application Server definitions by clicking **ADD MORE**, and then choosing your Application Server from the list. See *Endevor Application Server Parameters*, *ChangeMan Application Server Parameters*, and *AMS Application Server Parameters* for more information.
4. Review and modify the MFA Data Connect Service parameters, and then click **NEXT**.

   See *MFA Data Connect Service Parameters* for more information.
   - Optionally, review and modify the z/Server feature parameters, and then click **NEXT**.

     See *z/Server Feature Parameters* for more information.
   - Optionally, review and modify the z/Server Scheduler Application Server parameters, and then click **NEXT**.

     See *z/Server Scheduler Application Server Parameters* for more information.

To upload your XML configuration file to a mainframe

To upload your XML configuration file to the mainframe, perform the following steps:

1. In the **NODE** field, type your host name or IP address.
2. In the **PORT** field, type your MFA port number. Alternatively, you can use the up and down arrows to increment or decrement the port number.
3. In the **USER** field, type your username.
4. In the **PASSWORD** field, type your password.
5. In the **DATASET** field, type the fully qualified PDS library name of your XML configuration file.
6. In the **MEMBER** field, type the member name you want your XML configuration file to have.
7. To upload the XML file to the mainframe, click **UPLOAD**.
8. If successful, an **UPLOAD COMPLETE** dialog box is displayed, you can then choose to **SHUTDOWN**, **RESTART** or go **BACK**.

Downloading an existing XML configuration file

To download an existing XML configuration file, perform the following steps:

1. Click **DOWNLOAD**.
2. In the **NODE** field, type your host name or IP address.
3. In the **PORT** field, type your MFA port number. Alternatively, you can use the up and down arrows to increment or decrement the port number, respectively.
4. In the **USER** field, type in your username.
5. In the **PASSWORD** field, type your password.
6. In the **DATASET** field, type the fully qualified PDS library name of your XML configuration file.
7. In the **MEMBER** field, type the member name containing the XML configuration file you wish to download.
8. Click **NEXT**.

   The XML configuration file can now be reviewed. See *Reviewing an XML configuration file* for more information.

Edit Configuration page

At this point you can choose where to save your XML configuration file, start a new configuration or shutdown the utility:

- To upload your XML configuration file to the mainframe:
  - Click **UPLOAD**.

    See *To upload your XML configuration to a mainframe* for more information.
- To save your XML configuration file to your browser's download directory:
  - Click **SAVE**.
- To delete the configuration that you have created and start again:
  - Click **RESTART**.
- To shutdown the configuration utility:
  - Click **SHUTDOWN**.

See *Starting the z/Server Configuration Utility* for more information on how to start the z/Server configuration utility again.

Configuration parameters reference

See *Configuration reference* for more information on the parameters in your XML configuration file.

MFA Server parameters

Standard parameters:

TCP_PORT

The port number to accept connections from Micro Focus clients.

NETWORK_ID

The SSCP network ID used by ACF/VTAM on this z/OS system.

ORGANIZATION

Your company name or other meaningful identifier up to 40 characters in length.

Data Connect Application Server parameters

PROCEDURE

The name of the started task JCL procedure that can be used to start address spaces for this group. Sample JCL procedure MFAAS provides the basic JCL for an application server address space, and MFAAMS for the AMS application server address space. Specify a JCL procedure name of up to eight characters.

Endevor Application Server parameters

PROCEDURE

The name of the started task JCL procedure that can be used to start address spaces for this group. Sample JCL procedure MFAAS provides the basic JCL for an application server address space, and MFAAMS for the AMS application server address space. Specify a JCL procedure name of up to eight characters.

JOBNAME

The jobname prefix to be used for address spaces that are started for this group. For multiple-instance address spaces this is a prefix of 1 to 4 characters and Mainframe Access Server will pad this prefix to a full 8-character jobname by appending a 4 to 7 digit sequence number. For example, JOBNAME="MFAE" will result in jobnames MFAE0001, MFAE0002, etc. The sample prefix MFAE can be changed to meet the needs of your installation. If you do change the suggested prefix you will need to review the security subsystem definition for the Mainframe Access started tasks. The configuration process uses a generic STARTED task definition (the generic name specified during Quick Configuration is MFA*.*) that covers generated jobnames such as MFAExxxx, in addition to the MFA (the MFA Server control region) and MFAS (the MFA Server for Data Connect) started task names. Specify a jobname prefix of up to four characters.

MAXIMUM

Specifies the maximum number of address spaces that Mainframe Access server will start for a multiple-instance group. Additional address spaces beyond the minimum will be started in response to transaction load, up to the maximum allowed by this specification. Mainframe Access server issues messages (to the XDBOUT data set) when a client request must be queued to wait for an available processing address space. Increase the MAXIMUM value (by 1) when you observe frequent queueing of client requests.

MINIMUM

Specifies the maximum number of address spaces that Mainframe Access server will start for a multiple-instance group. Additional address spaces beyond the minimum will be started in response to transaction load, up to the maximum allowed by this specification.

ChangeMan Application Server Parameters

PROCEDURE

The name of the started task JCL procedure that can be used to start address spaces for this group. Sample JCL procedure MFAAS provides the basic JCL for an application server address space, and MFAAMS for the AMS application server address space. Specify a JCL procedure name of up to eight characters.

JOBNAME

The jobname prefix to be used for address spaces that are started for this group. For multiple-instance address spaces this is a prefix of 1 to 4 characters and Mainframe Access Server will pad this prefix to a full 8-character jobname by appending a 4 to 7 digit sequence number. For example, JOBNAME="MFAE" will result in jobnames MFAE0001, MFAE0002, etc. The sample prefix MFAE can be changed to meet the needs of your installation. If you do change the suggested prefix you will need to review the security subsystem definition for the Mainframe Access started tasks. The configuration process uses a generic STARTED task definition (the generic name specified during Quick Configuration is MFA*.*) that covers generated jobnames such as MFAExxxx, in addition to the MFA (the MFA Server control region) and MFAS (the MFA Server for Data Connect) started task names. Specify a jobname prefix of up to four characters.

MAXIMUM

Specifies the maximum number of address spaces that Mainframe Access server will start for a multiple-instance group. Additional address spaces beyond the minimum will be started in response to transaction load, up to the maximum allowed by this specification. Mainframe Access server issues messages (to the XDBOUT data set) when a client request must be queued to wait for an available processing address space. Increase the MAXIMUM value (by 1) when you observe frequent queueing of client requests.

MINIMUM

Specifies the maximum number of address spaces that Mainframe Access server will start for a multiple-instance group. Additional address spaces beyond the minimum will be started in response to transaction load, up to the maximum allowed by this specification.

AMS Application Server Parameters

PROCEDURE

The name of the started task JCL procedure that can be used to start address spaces for this group. Sample JCL procedure MFAAS provides the basic JCL for an application server address space, and MFAAMS for the AMS application server address space. Specify a JCL procedure name of up to eight characters.

JOBNAME

The jobname prefix to be used for address spaces that are started for this group. For multiple-instance address spaces this is a prefix of 1 to 4 characters and Mainframe Access Server will pad this prefix to a full 8-character jobname by appending a 4 to 7 digit sequence number. For example, JOBNAME="MFAE" will result in jobnames MFAE0001, MFAE0002, etc. The sample prefix MFAE can be changed to meet the needs of your installation. If you do change the suggested prefix you will need to review the security subsystem definition for the Mainframe Access started tasks. The configuration process uses a generic STARTED task definition (the generic name specified during Quick Configuration is MFA*.*) that covers generated jobnames such as MFAExxxx, in addition to the MFA (the MFA Server control region) and MFAS (the MFA Server for Data Connect) started task names. Specify a jobname prefix of up to four characters.

MAXIMUM

Specifies the maximum number of address spaces that Mainframe Access server will start for a multiple-instance group. Additional address spaces beyond the minimum will be started in response to transaction load, up to the maximum allowed by this specification. Mainframe Access server issues messages (to the XDBOUT data set) when a client request must be queued to wait for an available processing address space. Increase the MAXIMUM value (by 1) when you observe frequent queueing of client requests.

MINIMUM

Specifies the maximum number of address spaces that Mainframe Access server will start for a multiple-instance group. Additional address spaces beyond the minimum will be started in response to transaction load, up to the maximum allowed by this specification.

MFA Data Connect Service parameters

ID

Data Connect client requests do not specify a target server ID and Mainframe Access always looks for the DEFAULT Mainframe Access Data Connect target server definition.

LUNAME

The LU name of the Mainframe Access Data Connect server (also known as the ACBNAME or VTAM APPLID).

MODENAME

The SNA log mode name that will be used by Mainframe Access to indirectly specify SNA session parameters when Mainframe Access initiates sessions with the Mainframe Access Data Connect server. Specify a log mode name for LU6.2 sessions of up to eight characters. This log mode name must be present in the VTAM log mode table available to Mainframe Access. IBM's default log mode table, ISTINCLM, typically provides several standard log modes that can be used by Mainframe Access, including both IBMRDB and #INTER LU6.2 log modes.

TPNAME

The transaction program name for Mainframe Access Data Connect server requests.

SYNCLEVEL

The SNA LU6.2 sync level option to be used on conversations with the Mainframe Access Data Connect server. Conversations use LU6.2 CONFIRM protocols.

SECURITY

The security level to be indicated in SNA LU6.2 Attach FMH-5 requests sent to the Mainframe Access Data Connect server. Mainframe Access forwards security subfield information as provided by the Data Connect client and sets the FMH-5 fields accordingly.

RIMS Service Parameters

ID

If an IMS Option client request does not specify a target server ID or the specified target server ID does not exist, Mainframe Access selects the DEFAULT IMS target server, if one has been defined. This IMS target server ID name is used in the configuration of the client. When a Remote IMS request is sent this name is sent in the request data and is used to locate the target server definition for the IMS system that will receive the request. Specify DEFAULT or an ID of up to four alphanumeric characters.

LUNAME

The LU name of the IMS server (also known as the ACBNAME or VTAM APPLID). Specify an LU name of up to eight characters. This name must match the ACBNAME defined in an APPC/MVS LU definition for the target IMS system.

MODENAME

The SNA log mode name that will be used by Mainframe Access to indirectly specify SNA session parameters when Mainframe Access initiates sessions with the IMS server. Specify a log mode name for LU6.2 sessions of up to eight characters. This log mode name must be present in the VTAM log mode table available to Mainframe Access. IBM's default log mode table, ISTINCLM, typically provides several standard log modes that can be used by Mainframe Access, including both IBMRDB and #INTER LU6.2 log modes.

TPNAME

The IMS server transaction program name for IMS Option transactions. This is the LU6.2 transaction program name that will be sent to the IMS server in SNA Attach FMH-5 requests to begin an IMS Option transaction. Specify the APPC/MVS transaction program name that was specified in the APPC/MVS definitions during installation of Remote IMS.

SYNCLEVEL

The SNA LU6.2 sync level option to be used on conversations with the IMS server.

SECURITY

The security level to be indicated in SNA LU6.2 Attach FMH-5 requests sent to the IMS server. Mainframe Access forwards security subfield information as provided by the IMS Option client and sets the FMH-5 fields accordingly.

MCO Service Parameters

ID

If a CICS client request does not specify a target server ID or the specified target server ID does not exist, Mainframe Access selects the DEFAULT CICS target server, if one has been defined. This CICS target server ID name is used in the configuration of the client. When a CICS request is sent to Mainframe Access this name is sent in the request data and is used to locate the target server definition for the CICS system that will receive the request.

LUNAME

The LU name of the CICS server (also known as the ACBNAME or VTAM APPLID).

MODENAME

The SNA log mode name that will be used by Mainframe Access to indirectly specify SNA session parameters when Mainframe Access initiates sessions with mainframe CICS. Specify a log mode name for LU6.2 sessions of up to eight characters. This log mode name must be present in the VTAM log mode table available to Mainframe Access. IBM's default log mode table, ISTINCLM, typically provides several standard log modes that can be used by Mainframe Access, including both IBMRDB and #INTER LU6.2 log modes.

TPNAME

The server transaction program name for CICS requests. This is the LU6.2 transaction program name that will be sent to CICS in SNA Attach FMH-5 requests to begin a CICS transaction. Specify a transaction program name of up to eight characters or specify * to cause Mainframe Access to use the transaction program name provided by the CICS client. The CICS client prepares a partial FMH-5 request that specifies the standard CICS transaction program names for function shipping, distributed program linking and so on.

SYNCLEVEL

The SNA LU6.2 sync level option to be used on conversations with CICS.

SECURITY

The security level to be indicated in SNA LU6.2 Attach FMH-5 requests sent to the CICS server. Mainframe Access forwards security subfield information as provided by the CICS client and sets the FMH-5 fields accordingly.

ES-MTO Service Parameters

ID

The ID can be one to four characters in length and must match the SYSID of the ES/MSS server being defined. The initial connection messages exchanged by MFA Server and ES/MSS use this ID.

ACBNAME

The name of the VTAM ACB associated with this ES/MSS server. MFA Server OPENs and initializes this ACBNAME during startup. The z/OS CICS system(s) must be configured to communicate with this ES/MSS server through this VTAM ACB name using CICS CONNECTION and SESSIONS definition statements.

IPADDRESS

The internet host name or IP address of the ES/MSS server. Specify either the host name that resolves to the correct IP address or the actual IP address in standard dotted-decimal format. The combination of IPADDRESS and PORT from this definition provides MFA Server with the information it needs to originate a socket connection to ES/MSS and to verify the authenticity of a socket connection request from ES/MSS.

PORT

The port number where ES/MSS is listening for ISC connections.

SESSIONS

Specifies the number of concurrent conversations MFA Server can initiate to the ES/MSS server over a single socket connection. If ES/MSS has a different definition for the number of concurrent sessions, the session count is negotiated to a common value at the time a socket connection is established. Specify a value that supports the required number of concurrently active conversations for your application.

SOCKETS

Specifies the maximum number of concurrent socket connections between MFA Server and the ES/MSS server. ES/MSS support is currently limited to a single socket connection between MFA Server and an ES/MSS server; therefore, specify a value of 1.

z/Server feature Parameters

IPSTACK

Specifies the name of the IP stack used for processing.

SVC_NO

Specifies the SVC number for the type-3 SVC routine to be dynamically installed at holder address space startup.

DSP_TOKEN

Specifies the token name that each address space for one z/Server uses to address the data space with the user administration control structures.

z/Server Scheduler Application Server Parameters

Scheduler

SCHEDULER_NAME

Specifies the job start of the started task.

LISTENER_PORT

Specifies the IP port that the scheduler listens on for incoming work requests.

FIRST_PORT

Specifies the low end of the port range that the scheduler address space owns and assigns to a user server address space when the user server is started for IP communication.

LAST_PORT

Specifies the high end of the port range that the scheduler address space owns and assigns to a user server address space when the user server is started for IP communication.

CCSID

Specifies the code page used by z/Server (for example, 37 for US, or 1141 for German).

USER_SERVER_JOBNAME

Specifies the name of the user server start procedure used in a z/OS start command.

CEA_ACCOUNT

Specifies the name of the account to be used by all users of this scheduler.

CEA_LOGONPROC

Specifies the name of the TSO/E logon procedure used to start a CEA-launched TSO address space.

User Server CCSID

Specifies the code page used by z/Server (for example, 37 for US, or 1141 for German).

Host Installation

Preparing the Installation

Before you begin

The following list details important configuration data that you need to know in order to complete the quick configuration. Review these items and determine the appropriate values before you proceed with customization:

userid TSO user ID that can update `proclib`, update `vtamlst`, update RACF, and issue z/OS system commands through SDSF. Alternatively, the cooperation of individuals with the appropriate authorities.

hlq High level qualifier that was given to all Mainframe Access data sets when the FRESTORE job was run.

clientport A TCP/IP port number on the mainframe that you will reserve for Mainframe Access and client connections.

stcuserid RACF user ID that you want to associate with the Mainframe Access started tasks.

stcgroup RACF group ID that you want to associate with the Mainframe Access started tasks.

tcpdata Data set name of the IBM TCP/IP "TCPIP.DATA" data set that provides configuration information to programs that use TCP/IP.

proclib Data set name of a system procedure library.

vtamlst Data set name of the active VTAMLST definitions data set.

netid VTAM SNA network id used on your system. See NETID= in the active ATCSTRxx `vtamlst` member.

SVC Routine (deprecated)

Note: SVC routine is deprecated, and supported for backward compatibility only.

MFA z/Server feature supports requires a type-3 SVC routine for all authorized commands. You need to call this routine TAURAUTH and reserve a number for it. 238 is the default specified in the configuration files that come with z/Server.

Installation procedure

Upload the software

In the instructions that follow, the information that you must provide is shown as one of the variable names from the table of information in the previous section. For example, if your high-level qualifier (`hlq`) value is MY.MFA, then substitute MY.MFA for `hlq`.

To load Mainframe Access Server:

1. Download the installation file from the link in your Electronic Product Delivery email and extract its contents to a directory on the PC.
2. On the mainframe, allocate a new partitioned data set named `hlq`.UPLOAD to receive the uploaded files. Use the following data set characteristics for this upload library:

   ```
       DSORG=PO                   <=== PDS (partitioned data set)
       RECFM=FB                   <=== record format fixed and blocked
       LRECL=80                   <=== 80 character record size
       BLKSIZE=3120               <=== 3120 character block size
       SPACE=(3120,(3500,500,50)) <=== allocate blocks (BLKS) size 3120
                                           3500 primary blocks
                                           500 secondary blocks
                                           50 directory blocks
   ```

3. On the PC, issue the following FTP commands. The actual text of the FTP prompts and responses that you see may differ slightly from those shown in this example:

   a. Start FTP:

   ```
   C:\>ftpyour.mainframe.name
   Connected to your.mainframe.name.
   220-FTPD1 IBM FTP CS/390 VxRy at YOUR.MAINFRAME.NAME, hh:mm:ss
   220 Connection will close if idle for more than 5 minutes.
   User (your.mainframe.name:(none)): userid
   331 Send password please.
   Password: pswd
   230 userid is logged on. Working directory is "userid.".
   ```

   b. Change the working directory on the mainframe to be the upload library that you allocated:

   ```
   ftp> cd 'hlq.UPLOAD'
   250 The working directory "hlq.UPLOAD" is a partitioned data set.
   ```

   c. Set file transfer type to binary:

   ```
   ftp> binary
   200 Representation type is Image
   ```

   d. Set FTP prompting off to transfer all files without interruption:

   ```
   ftp> prompt
   Interactive mode Off.
   ```

   e. Transfer all files from the extracted \Upload directory to members in the `hlq`.UPLOAD library:

   ```
   ftp> mput drive:\upload\*
   200 Port request OK.
   125 Storing data set hlq.UPLOAD(Xxxxxxxx)
   250 Transfer completed successfully.
   ftp: xxxx bytes sent in x.xx seconds (xxx.xx Kbytes/sec)
   ```

   f. When `mput` has transferred all files the `ftp>` prompt appears. End the FTP connection:

   ```
   ftp> quit
   221 Quit command received. Goodbye.
   ```

   g. On the mainframe, verify that all files transferred successfully and that for each xxxxxxxx file in the `\Upload` directory there is a corresponding member in the `hlq`.UPLOAD data set. There should be 20 members.

Receive the software

In `hlq`.UPLOAD you now have a partitioned dataset with 20 members. Each member except FRESTORE is a dataset in transmit format. You need to execute TSO RECEIVE commands on the datasets to create the Mainframe Access datasets.

Member	Description	Received to
AUTHLIB	z/Server load library that needs to be APF authorized.	hlq.ZSERVER.AUTHLIB

Member	Description	Received to
CONFIG	z/Server default configuration members and XML schema.	hlq.ZSERVER.CONFIG
DATA	z/Server XML sample documents.	hlq.ZSERVER.DATA
EXEC	z/Server REXX procedures run as ISPF applications called from a client via TCP/IP.	hlq.ZSERVER.EXEC
F1	The load module library for Mainframe Access server that needs to be APF authorized.	hlq.MFA.LOADLIB
F2	Samples for Mainframe Access server. These include JCL for jobs to be run, JCL for started task procedures and parameter files. The members in this data set are referred to frequently in these configuration instructions.	hlq.MFA.CNTL
F3	Translation tables used by Mainframe Access server.	hlq.MFA.TABLES
F4	Product source and object files that you may need during setup of the Remote IMS Server feature.	hlq.MFA.RIMS
F5	Contains REXX execs used for MFA TSO support, and includes AWM functions.	hlq.MFA.EXEC
FRESTORE	JCL to receive all other members.	
JCL	z/Server JCL samples.	hlq.ZSERVER.JCL
LOADLIB	z/Server load library.	hlq.ZSERVER.LOADLIB
MASTER	z/Server master configuration file.	hlq.ZSERVER.MASTER
MSGS	z/Server ISPF message library belonging to the ISPF applications in EXEC.	hlq.ZSERVER.MSGS
OSR	Optimized schema representation used for internal XML validation.	hlq.ZSERVER.OSR
PANELS	z/Server ISPF panel library belonging to the ISPF applications in EXEC.	hlq.ZSERVER.PANELS
REXX	z/Server REXX procedures and REXX samples.	hlq.ZSERVER.REXX
SAMPLIB	z/Server Sample programs.	hlq.ZSERVER.SAMPLIB
SKELS	z/Server ISPF skeleton library belonging to the ISPF applications in EXEC.	hlq.ZSERVER.SKELS
XML	z/Server Sample models (PDS Explorer).	hlq.ZSERVER.XML

On the mainframe, edit member FRESTORE in the upload library, `hlq`.UPLOAD. Follow the instructions in that member to customize the JCL and then submit that job to restore the product libraries from the uploaded files and populate your new product runtime libraries.

Configuration overview

There are two types of configuration tasks that you must complete before you can start and use Mainframe Access. These are:

- Updating the z/OS mainframe system to create an execution environment for Mainframe Access.
- Setting parameters to meet your requirements.

The process of configuring Mainframe Access is separated into two distinct procedures. See *Quick Configuration* and *Advanced Configuration Activities* for more information.

Quick configuration is performed using the UpQuick instructions as indicated during the installation process. These configuration steps must be completed for all installations. Quick configuration takes care of all of the z/OS customization and many basic Mainframe Access customizations. This quick configuration is the only Mainframe Access setup required for Mainframe Access Drag and Drop, SQL Option for DB2, and basic Mainframe Access Data Connect server functions.

Additional setup is required if:

- You are going to be using Mainframe Access Data Connect server and you need to use the file name mapping table and/or alter the processing in the SAF security exits.
- You are going to be using the following features:
  - Remote IMS
  - ES/MSS support
  - Mainframe Access's access list checking
  - Mainframe Access's audit log
  - Mainframe Access's support for external library management systems
  - z/Server support

The section on advanced configuration describes these additional customizations.

If you have completed the installation process, including the UpQuick configuration steps, and if you are not going to use features that require additional setup, your configuration is complete.

If you have uploaded and restored the Mainframe Access data sets but you have not completed the UpQuick configuration, you should begin with the *Quick Configuration* topic.

If you have completed the configuration and testing of your basic Mainframe Access server and you need to customize for an advanced feature, you should continue at the *Advanced Configuration Activities* topic.

Mainframe Access Data Sets

Following the UpQuick instructions from the installation package, you should have already completed the upload and restore of the Mainframe Access data sets. During that process you selected a high level qualifier to be used for all of your Mainframe Access data sets. This guide refers to the qualifier that you selected as <hlq> or `hlq`. The data sets that you restored are listed in the following table:

Data Set	Description
hlq.ZSERVER.AUTHLIB	z/Server load library that needs to be APF authorized.
hlq.ZSERVER.CONFIG	z/Server default configuration members and XML schema.
hlq.ZSERVER.DATA	z/Server XML sample documents.
hlq.ZSERVER.EXEC	z/Server REXX procedures run as ISPF applications called from a client via TCP/IP.
hlq.MFA.LOADLIB	The load module library for Mainframe Access server that needs to be APF authorized.

Data Set	Description
hlq.MFA.CNTL	Samples for Mainframe Access server. These include JCL for jobs to be run, JCL for started task procedures and parameter files. The members in this data set are referred to frequently in these configuration instructions.
hlq.MFA.TABLES	Compression tables used by Mainframe Access server.
hlq.MFA.RIMS	Product source and object files that you may need during setup of the Remote IMS Server feature.
hlq.MFA.EXEC	Contains REXX execs used for MFA TSO support, and includes AWM functions.
hlq.ZSERVER.JCL	z/Server JCL samples.
hlq.ZSERVER.LOADLIB	z/Server load module library which must be APF authorized.
hlq.ZSERVER.MASTER	z/Server master configuration file.
hlq.ZSERVER.MSGS	z/Server ISPF message library belonging to the ISPF applications in EXEC.
hlq.ZSERVER.OSR	Optimized schema representation used for internal XML validation.
hlq.ZSERVER.PANELS	z/Server ISPF panel library belonging to the ISPF applications in EXEC.
hlq.ZSERVER.REXX	z/Server REXX procedures and REXX samples.
hlq.ZSERVER.SAMPLIB	z/Server Sample programs.
hlq.ZSERVER.SKELS	z/Server ISPF skeleton library belonging to the ISPF applications in EXEC.
hlq.ZSERVER.XML	z/Server Sample models (PDS Explorer).

Mainframe Access Samples

The installation procedure places many samples in the <hlq>.MFA.CNTL data set. If a sample is provided for a configuration task, use it as a starting point for your configuration. Usually you need to edit the sample, replacing provided information with information that is unique to your system.

The following table summarizes the samples that are provided for Mainframe Access:

Member	Description
ACCESS	List used to restrict access to Mainframe Access consisting of the IP addresses of connecting clients. The access list also contains the list of user IDs that are authorized to access the Mainframe Access administration functions through the Web browser interface. You should add the IP addresses and user IDs for your system. See *Editing Access List Definitions* for more information.
AUDIT	JCL for creating and initializing a Mainframe Access audit log data set. The audit log is a VSAM key sequenced data set (KSDS). You should edit the sample to include JOB card information, data set names and a volume serial number.
AUDRPT	JCL for running the Mainframe Access audit log report program to list the contents of the audit log data set. You should edit the sample to include JOB card information and data set names.

Member	Description
FRESTORE	A copy of the FRESTORE job for reference purposes. This job was originally transferred from the client to the <hlq>.UPLOAD data set during installation. The job was customized and submitted to allocate the permanent Mainframe Access data sets and restore the data set content from other uploaded files.
GTFCNTL	Input control statements referenced by the sample JCL procedure in member MFAGTF.
INSTALL	A copy of the installation upload instructions for reference purposes. These are the instructions that describe how to allocate the <hlq>.UPLOAD data set and transfer the CD content into that data set.
MFA	JCL for executing Mainframe Access. You should edit the sample to include your data set names. This JCL contains references to the ACCESS, PARMS and SERVERS sample members.
MFAAS	JCL for executing a Mainframe Access Application Server. Application servers are auxiliary address spaces that process specific types of client requests such as Endevor transactions. You should edit the sample to include your data set names. This JCL contains a reference to the PARMSAS sample member.
MFAAMS	JCL for executing a Mainframe Access Application Server for AMS support. Application servers are auxiliary address spaces that process specific types of client requests such as AMS requets. You should edit the sample to include your data set names. This JCL contains a reference to the PARMSAS sample member.
MFAGTF	JCL for running IBM's Generalized Trace Facility (GTF). GTF provides high-performance recording of diagnostic data and can be useful during problem determination. You should edit the sample to include your data set names.
MFANLS	JCL for re-compiling the default code page that may be configured at a client workstation. This is equivalent to the MFLSCTRN load module of Mainframe Access Version 2. During installation the default is set to use code pages E037/A437 for EBCDIC/ASCII translation. The sample JCL contains instructions for changing the default code pages selected by installation.
MFAS	JCL for executing the Mainframe Access Data Connect server. You should edit the sample to include your data set names.
MFAVTAM	Definition of VTAM applications for Mainframe Access, Mainframe Access Data Connect, ES/MSS and Remote IMS Server. You can edit the sample to include your own ACB names and network names. See *VTAMLST Definitions* for more information.
MFAXML	An XML configuration that contains parameters that are read when Mainframe Access and Mainframe Access Application Server is initialized. The sample JCL for executing Mainframe Access refers to this member for parameter input. You will need to edit some of these parameters. See *Editing Mainframe Access Parameters*

Member	Description
	for more information. It also contains definitions that describe the z/OS subsystems that will be providing service in response to client requests. The servers are your IMS and CICS subsystems, the Mainframe Access Data Connect server and ES/MSS servers. You must edit the sample to include information specific to your servers and also coordinate the information with definitions in the client packages.
PVSUFFX	JCL for executing a migration tool which extracts a text definition from the MFLSCPAN load module of Mainframe Access Version 2. See *Panvalet* for more information.
TSOXMIT	Sample job that uses TSO XMIT to package diagnostic data for transmission.
UPQUICK	A copy of the quick configuration instructions for reference purposes. These are the instructions that describe how to quickly configure z/OS and Mainframe Access to create an operational Mainframe Access server system. See *Quick Configuration* for more information.

Quick Configuration

After you have uploaded the mainframe files and successfully run the FRESTORE job, there are some basic customization tasks that must be completed. Quick configuration works through these configuration tasks quickly and gets you to the point where the software can be started and a basic installation verification check can be performed. After the installation verification check, continue with *Advanced Configuration Activities* if there are additional features that you need to set up.

You should be familiar with TSO, ISPF, SDSF, z/OS system commands, JCL procedures and proclibs, VTAMLST definitions and RACF.

Note: These instructions refer to RACF as the security subsystem. Mainframe Access does use the SAF interface to the security subsystem and is compatible with other security products that provide the SAF API. When these instructions refer to RACF-related customizations, please refer to your security product documentation to determine the corresponding security subsystem changes.

If you are using the ISPF editor to make the configuration changes, use the change command once to make the first update to the `hlq` value and then use ISPF's Rchange command (usually PF6) to apply the same change in all the members that you need to update. For example: use "c HLQ MY.HIGH.LEVEL" the first time and then PF6 to apply the same change in other places. Changes to other parameters are easily accomplished by overtyping the preset value.

MFA Started Task JCL Procedure

Customize member MFA in the CNTL data set. This is the started task JCL procedure for Mainframe Access server. Use the ISPF change command to change HLQ to your qualifier (`hlq`).

If you are using IBM's TCP/IP, overtype TCPIP.OS390.TCPDATA with the name of your installation's "TCPIP.DATA" data set (`tcpdata`). This is the data set used by client programs to obtain the name of the TCP/IP started task and other installation-dependent TCP/IP information.

To enable DBCS support, you need to edit the procedure by uncommenting the following four statements:

```
//*-----------------------------------------------------
//* Optional DBCS code pages
//*-----------------------------------------------------
//*
//*EZACHLAT DD  DSN=TCPIP.SEZADBCX(EZACHLAT),DISP=SHR
//*EZAHGLAT DD  DSN=TCPIP.SEZADBCX(EZAHGLAT),DISP=SHR
```

```
//*EZAKJLAT DD  DSN=TCPIP.SEZADBCX(EZAKJLAT),DISP=SHR
//*EZASCLAT DD  DSN=TCPIP.SEZADBCX(EZASCLAT),DISP=SHR
```

To start the z/Server holder under MFA, you need to edit the procedure by uncommenting the following five statements.

Note: You should only do this if told to by Micro Focus support.

```
//*--------------------------------
//* zServer Holder
//*--------------------------------
//*
//*SYSEXEC  DD DISP=SHR,DSN=&TAUQUAL..EXEC
//CONFXML  DD DISP=SHR,DSN=&DSNQUAL..CNTL(MFAXML)
//CONFOSR  DD DISP=SHR,DSN=&TAUQUAL..OSR(MFAOSR)
//*DSPPRT   DD SYSOUT=*,LRECL=255
//*SYSOUT   DD SYSOUT=*
//*SYSTSIN  DD DUMMY
//*SYSTSPRT DD SYSOUT=*
```

MFATSO Started Task JCL Procedure

Customize member MFATSO in the CNTL data set. This is the started task JCL procedure for Mainframe Access TSO Command Server. Use the ISPF change command to change HLQ to your qualifier (hlq).

A MFA STC user server MFATSO is started by the MFA TSO Application Server to execute REXX execs or ISPF applications.

Ensure that the ISPF high level qualifier is ISP, unless local naming conventions are different. Similar to a logon procedure, add all required libraries, such as panels, messages, and REXX execs that are necessary to run the user's required ISPF applications.

The initial REXX procedure MFAREXX must be located in the hlq.MFA.EXEC data set. This then calls the REXX procedure IVPINIT1 which is also located in hlq.MFA.EXEC. This REXX exec should be customized by the installation so that it meets your installation requirements. See *Optional customization* for more information.

Note: All data sets on the STEPLIB concatenation need to be APF. Failure to do so will result in a 0C4 abend when the MFA TSO Application Server is started.

Optional customization

MFA can have more than one user server per TSO user to be run in parallel. Each user server needs its own exclusive ISPF environment. This implies the allocation of an ISPF user profile data set. The allocation of the user profile data set is done in the sample REXX exec hlq.MFA.EXEC(IVPINIT1) and should be customized to adhere to the installation's standards.

ISPF user profile allocation is performed as follows:

- DD statement ISPPROF is allocated to a temporary data set. If a MFA ISPF profile data set named userid.TAUZCISP.PROFILE already exists for the TSO user, the content of this ISPF profile is copied using IEBGENER to a temporary data set allocated under the ISPPROF DD statement.
- When the ISPF session terminates, control returns to REXX exec MFAREXX and the temporary ISPF profile is copied back to MFA profile data set userid.TAUZCISP.PROFILE.
- The temporary ISPF profile data set is deleted in REXX exec IVPTERM1.

  Micro Focus recommends that you allocate the temporary ISPF profile data set to an SMS managed temporary data set pool, which is automatically deleted according to the installation's standards. In this case, the deletion of the temporary ISPF profile data set in the REXX MFAREXX can be omitted.

MFAAS Started Task JCL Procedure

Customize member MFAAS in the CNTL data set. This is the started task JCL procedure for Mainframe Access Application Servers such as Endevor, ChangeMan, and AMS. Use the ISPF change command to change HLQ to your qualifier (`hlq`).

If you are using IBM's TCP/IP, overtype TCPIP.OS390.TCPDATA with the name of your installation's "TCPIP.DATA" data set (`tcpdata`).

MFAS Started Task JCL Procedure

Customize member MFAS in the CNTL data set. This is the started task JCL procedure for Mainframe Access Data Connect server. Use the ISPF Rchange command to change HLQ to your qualifier (`hlq`).

MFAAMS Started Task JCL Procedure

Customize member MFAAMS in the CNTL data set. This is the started task JCL procedure for using Access Method Services, required for dataset renaming using MVS Explorer in the eclipse client. Use the ISPF change command to change HLQ to your qualifier (`hlq`).

If you are using IBM's TCP/IP, overtype TCPIP.OS390.TCPDATA with the name of your installation's "TCPIP.DATA" data set (`tcpdata`).

Mainframe Access MFAXML

Customize member MFAXML in the CNTL data set. These are the Mainframe Access server initialization parameters referenced by the Mainframe Access started task JCL and Mainframe Access Application Server started task JCL.

Change TCP_PORT="2020" to TCP_PORT="`clientport`". This is the port number that Micro Focus client software should be configured to use.

Change NETWORK_ID="DDINET1" to NETWORK_ID="`netid`", the VTAM SNA network id of your system.

Note: Ensure that when editing or uploading the MFAXML configuration file that you use CCSID 1047. Otherwise the XML document might fail validation.

VTAMLST Definitions

Copy member MFAVTAM to `vtamlst`. This application major node definition contains APPL definitions for Mainframe Access server, Mainframe Access Data Connect server and Remote IMS Server. Use the z/OS VARY command to activate the definition:

```
V NET,ACT,ID=MFAVTAM
```

Note:

- If you change any ACBNAME= values or the PRTCT= value in the MFAVTAM sample you must make corresponding changes:
  - in member MFAXML (if ACBNAME=MFM62ACB is changed)
  - in member MFAXML (if PRTCT=MFM62PSW is changed)
  - in member MFAS (if ACBNAME=MFA62ACB is changed)
- You will want to add MFAVTAM into the `vtamlst` (ATCCONxx) member so that this application major node is automatically activated during VTAM initialization.

Add Started Task JCL Procedures to Proclib

Copy members MFA, MFAAS, MFAS, and MFATSO to `proclib`.

Note:

- If you store proc MFAS in `proclib` using a different name, you must also change PROCDURE attribute of DataConnect in member MFAXML to specify that new procedure name.
- If you store proc MFAAS in `proclib` using a different name, you must also change PROCEDURE attribute of the relevant Application Server in member MFAXML to specify that new procedure name.
- If you store proc MFATSO in `proclib` using a different name, you must also change TSO_PROCEDURE attribute of the relevant TSO Application Server in member MFAXML to specify that new procedure name.

Add MFATSOIN to System Link List

The MFA TSO initialization module MFATSOIN must be copied from HLQ.MFA.LOADLIB to the system link list. This must then be refreshed by issuing the following console command:

```
F LLA,REFRESH
```

Note: If you store load module MFATSOIN in the system link list using a different name, you must also change the TSO_INITMOD attribute of the relevant TSO Application Server in member MFAXML to specify the new TSO load module name.

Customize AWM master configuration file

Customize member TAUZCAPP in the hlq.MFA.EXEC data set. This is the master configuration file used by the AWM client. Use the ISPF change command to change HLQ to your qualifier (hlq). See *Master Configuration on z/OS* for more information.

APF-Authorize the Load Module Libraries

Use z/OS SETPROG commands to APF-authorize the load libraries:

```
SETPROG APF,ADD,DSNAME=hlq.MFA.LOADLIB,SMS
SETPROG APF,ADD,DSNAME=hlq.ZSERVER.AUTHLIB,SMS
SETPROG APF,ADD,DSNAME=hlq.ZSERVER.LOADLIB,SMS
```

Note: Add these APF library definitions to a SYS1.PARMLIB(PROGxx) member so that these same authorizations are automatically activated during z/OS system initialization.

Define Mainframe Access to the Security Subsystem

Use RACF TSO commands to define the Mainframe Access started tasks (MFA, MFAAS and MFAS) and assign a userid and groupid to the tasks. The `stcuserid` should have appropriate access to the Mainframe Access data sets and should also have a basic OMVS RACF segment defined. At a minimum, the `stcuserid` OMVS segment must have a UID. The `stcgroup` must also have an OMVS segment with a GID. The following REDEFINE will provide the necessary definition for all of the started tasks (MFA, MFAAS and MFAS).

```
RDEFINE STARTED MFA*.* STDATA(USER(stcuserid) GROUP(stcgroup))
   OWNER(stcuserid)
SETROPTS RACLIST(STARTED) REFRESH
```

If you are using another security product such as CA-ACF2 or CA-Top Secret you will need to make similar updates to your security system definitions. MFA, MFAAS and MFAS will need a started task definition with an OMVS segment and they will need access to the SAF API for security subsystem calls. Please refer to your security product documentation.

Define started tasks to WLM

As a server, the importance of the MFA Server address space should be set below TCPIP but above the Data Connect Mainframe Access Server, and other Mainframe Access Application Servers such as

Endevor, Changeman, IDCAMS, and MFATSO Application Servers. This setting is checked when MFA Server starts up, terminates, or when commands are executed against the MFA Server address space.

The user servers should be classified like any TSO user address space, using the response time goal of TSO. All transactions executing within the user server are TSO transactions.

Note: These address spaces are STCs, so the classification must be done under the STC subsystem.

Starting Mainframe Access

When you have completed all of the tasks described in the preceding sections you are ready to start Mainframe Access to perform some simple installation verification tests. Start Mainframe Access by issuing the following z/OS START command at a system console or using the TSO/ISPF SDSF command facility:

```
S MFA
```

Here is an example that shows the messages displayed during startup. The start command is issued for MFA, the name of the started task JCL sample that has been customized and copied into a system procedure library. The MFAXML sample also defines the Mainframe Access Data Connect server as an application server address space that should be started automatically by Mainframe Access during initialization. Mainframe Access issues an internal start command for MFAS, the name of the started task JCL sample for the Mainframe Access Data Connect server.

```
04:11:52.41 XXXXXXXX 00000290  S
MFA

04:11:52.47 STC03887 00000090  £HASP100 MFA   ON
STCINRDR
04:11:52.55 STC03887 00000290  IEF695I START MFA   WITH JOBNAME MFA   IS
ASSIGNED TO USER XXXXXXXX
                                , GROUP
XXXXXXXX
04:11:52.55 STC03887 00000090  £HASP373 MFA
STARTED
04:11:52.56 STC03887 00000090  IEF403I MFA - STARTED -
TIME=04.11.52
04:11:53.06 STC03887 00000090  MFM0063I: MFA Direct is
active
04:11:53.06 STC03887 00000090  MFM0064I: MFA Direct number of processing
tasks is 10
04:11:53.09 STC03887 00000090  MFM0134I: ES/MTO Outbound feature is
active
04:11:53.10 STC03887 00000090  MFM0127I: ChangeMan subsystem ID is
SERA
04:11:53.20 STC03887 00000090  MFMDS008I DBCS code page support
enabled
04:11:53.22 STC03887 00000090  MFMDS051I CA/LIBrarian
Ready
04:11:53.27 STC03887 00000090  MFMDS052I CA/PanValet
Ready
04:11:53.29 STC03887 00000090  MFMDS053I CA/ENDEVOR API
REL=B1700C,ESI=N,UID=N
04:11:53.46 STC03887 00000090  MFMDS055I MCG/RPC feature
active
04:11:53.46 STC03887 00000090  MFMDS001I DSS Server
Ready
04:11:54.32 STC03888 00000090  £HASP100 MFAS  ON
STCINRDR

04:11:54.39 STC03888 00000090  £HASP373 MFAS
STARTED
04:11:54.40 STC03888 00000090  IEF403I MFAS - STARTED -
TIME=04.11.54
04:11:54.48 STC03888 00000090  MFA302I.MFAS.MFDSTART / ATTACHING VTAM
```

```
PROCESSOR
04:11:54.48 STC03888 00000090  MFA303I.MFAS.MFDSTART / MFA/DATACONNECT V6.00
- 04000000 COPYRIGHT
                               (C) 1987-2018 MICRO FOCUS INTERNATIONAL
LTD.
04:11:54.49 STC03888 00000090  MFA110I.MFAS.MFAFSQ00 / VTAM PROCESSOR ACTIVE
USING VTAMAPPL
MFA62ACB
04:11:59.29 STC03887 00000090  MFM0001I: Mainframe Access V6.00    (04000000)
is active
```

Stopping Mainframe Access

Stop Mainframe Access by issuing the following z/OS STOP command at a system console or using the TSO/ISPF SDSF command facility:

```
P MFA
```

Here is an example that shows the messages displayed during shutdown. The stop command is issued only for Mainframe Access, the Mainframe Access server. Mainframe Access issues an internal stop command for MFAS, the application server address space for Mainframe Access Data Connect that was defined in the SERVERS member and started automatically by Mainframe Access. MFAS ends first and then Mainframe Access completes its shutdown processing.

```
04:50:44.70 XXXXXXXX 00000290  P
MFA
04:50:50.72 STC01318 00000090  MFMDS005I DSS Server
shutdown
04:51:01.02 STC01318 00000290  P
MFAS
04:51:01.03 STC01319 00000090  MFA112I.MFAS.MFAFSQ00 / VTAM PROCESSOR HAS
TERMINATED
04:51:02.04 STC01319 00000090  MFA317I.MFAS.MFDSTART / HAS BEEN
TERMINATED
04:51:02.06 STC01319 00000090  IEF404I MFAS - ENDED -
TIME=04.51.02
04:51:02.06 STC01319 00000090  £HASP395 MFAS  ENDED - RC=0000
04:51:12.54 STC01318 00000090  MFM0011I: Mainframe Access shutdown
completed
04:51:12.59 STC01318 00000090  IEF404I MFA - ENDED -
TIME=04.51.12
04:51:12.60 STC01318 00000090  £HASP395 MFA   ENDED - RC=0000
```

Advanced Configuration Activities

Mainframe Access is a general purpose server supporting a variety of Micro Focus client products. The Quick Configuration process that you have completed enables a large number of Mainframe Access server features that will satisfy most access requirements. This section describes the customizations (beyond the quick configuration) that are needed to enable various additional features of Mainframe Access server.

The following table summarizes the required updates on a per-feature basis. This will help you to select the correct configuration activities for a specific feature.

Feature	Customization
Audit Log	Refer to *Allocating and Initializing an Audit Log Data Set* to prepare the data set. Then see *Editing Mainframe Access Parameters* and update the audit log parameter to **AUDIT_LOG="1"**.
Access List	Refer to *Editing Access List Definitions* for guidance in defining IP addresses and address ranges. Then see *Editing Mainframe Access Parameters* and update the

Feature	Customization
	access list check parameter to **ACCESS_LIST_CHECK="1"**.
CICS support	Using CICS support, clients can interact with z/OS CICS servers for file access, remote program execution, transaction execution, etc. Activate support for this feature by setting **MCOLINK_MAXTASKS="2"**. The number of tasks supporting this feature can be increased in the future as your usage dictates. To complete the setup for this feature you will need to customize your client and the z/OS CICS server.
Remote IMS Option	Using the Remote IMS Option, clients can execute DL/I calls that are processed by a z/OS IMS server. Activate support for this feature by setting **IMSLINK_MAXTASKS="2"**. The number of tasks supporting this feature can be increased in the future as your usage dictates. To complete the setup for this feature you will need to customize your client and the z/OS IMS server. You must also install the Remote IMS server software in the target z/OS IMS server and enable APPC/MVS communications for that IMS server. All of the machine-readable installation materials for your Remote IMS Server can be found in the <hlq>.RIMS data set.
List DB2	Mainframe Access' List DB2 feature dynamically identifies z/OS DB2 and IMS servers that are active on the z/OS system where Mainframe Access server is executing. This information can be helpful when defining these systems to Mainframe Access server and clients. Activate support for this feature by setting **LIST_DB2="1"**.
Data Connect	Quick Configuration installs the Mainframe Access Data Connect server with SAF security checking active and without special customizations for PC to mainframe file name mapping. If you need to alter this basic setup see the appendix *Data Connect*.
Enterprise Server/Mainframe Subsystem Support	Enterprise Server/Mainframe Subsystem Support (ES/MSS) enables Mainframe Access server to accept CICS ISC requests initiated by z/OS CICS and route them to an ES/MSS server for execution. The combination of ES/MSS support and the CICS support creates a bi-directional ISC connection between ES/MSS and z/OS CICS. To complete the setup for this feature you need to customize your ES/MSS server(s) and the z/OS CICS server(s). See your Enterprise Server documentation and IBM z/OS CICS publications for help with these customizations.
ChangeMan interface	Mainframe Access' Source Connect feature works with ChangeMan to provide client access to ChangeMan-managed files. For more information see the section *ChangeMan ZMF* in the appendix *External Library Management Systems*.
Panvalet interface	Mainframe Access' Source Connect feature works with Panvalet to provide client access to Panvalet-managed files. For more information see the section *Panvalet* in the appendix *External Library Management Systems*.

Feature	Customization
Librarian interface	Mainframe Access' Source Connect feature works with Librarian to provide client access to Librarian-managed files. For more information see the section *Librarian* in the appendix *External Library Management Systems*.
Endevor interface	Mainframe Access' Source Connect feature works with Endevor to provide client access to Endevor-managed files. For more information see the section *Endevor* in the appendix *External Library Management Systems*.
z/Server support	Mainframe Access' z/Server feature ….. For more information about configuring z/Server see the section *Configuring Access to z/Server*.

Note: As noted in the preceding customization summaries, you also need to prepare the external servers and clients that Mainframe Access works with.

Editing Mainframe Access Parameters

You edit the Mainframe Access parameter definitions located in member MFAXML to customize them for your system. The sample parameter definitions are provided here as an example. MFAXML is an XML formatted file, so any lines that are surrounded by the XML comment characters <!-- and --> are treated as comments and are skipped in processing:

```
<Configuration
                     TCP_PORT="2020"
                     NETWORK_ID="DDINET1"
                     LU62_APPLID="MFM62ACB"
                     APPLID_PASSWORD="MFM62PSW"
                     TRACING="0"
                     ORGANIZATION="YOUR_COMPANY_NAME"
>

    <!-- Application Server address space definitions  -->
    <ApplicationServers>
        <!-- Mainframe Access Server (Data Connect) address space -->
        <DataConnect PROCEDURE="MFAS"
        />

        <!-- Mainframe Access Application Server(s) for Endevor -->
        <Endevor     PROCEDURE="MFAAS" MAXIMUM="2" JOBNAME="MFAE"
        />

        <!-- IDCAMS address space - needed for data set RENAME support -->
        <AMS         PROCEDURE="MFAMS" MAXIMUM="1" JOBNAME="MFAA"
        />

        <!-- Mainframe Access Application Server for TSO command support -->
        <TSO         PROCEDURE="MFAAS" TSO_PROCEDURE="MFATSO"
                     TSO_JOBCHAR="W"
        />
    </ApplicationServers>

    <!-- Service Configuration -->
    <Services>
        <!-- MFA Data Connect server - only one DEFAULT server is supported --
>
        <MFA         ID="DEFAULT" LUNAME="MFA62ACB" MODENAME="#INTER"
                     TPNAME="FILESHR2" SYNCLEVEL="1" SECURITY="0"
        />

        <!-- Sample IMS servers - multiple servers may be defined -->
        <!--
        <RIMS        ID="IMSA" LUNAME="IMSSYSA" MODENAME="IBMRDB"
                     TPNAME="MFDBTP6" SYNCLEVEL="0" SECURITY="0"
        />

        <RIMS        ID="DEFAULT" LUNAME="IMSTEST" MODENAME="IBMRDB"
                     TPNAME="MFDBTP6" SYNCLEVEL="0" SECURITY="0"
        />
        -->

        <!-- Sample CICS servers - multiple servers may be defined -->
        <!--
        <MCO         ID="CICA" LUNAME="CICSSYSA" MODENAME="#INTER"
                     TPNAME="*" SYNCLEVEL="0" SECURITY="0"
        />

        <MCO         ID="DEFAULT" LUNAME="CICSTEST" MODENAME="#INTER"
                     TPNAME="*" SYNCLEVEL="0" SECURITY="0"
        />
        -->

        <!-- Sample ES/MSS servers - multiple servers may be defined -->
        <!--
        <ES-MTO      ID="MTO1" ACBNAME="MFAMTO1" IPADDRESS="YOUR.MTO.HOSTNAME"
                     PORT="2200" SESSIONS="4" SOCKETS="1"
        />

        <ES-MTO      ID="MTO2" ACBNAME="MFAMTO2" IPADDRESS="101.102.103.104"
                     PORT="3300" SESSIONS="4" SOCKETS="1"
        />
        -->
    </Services>
</Configuration>
```

The section in bold above highlights the area of the XML file that you can edit, add, or remove parameters in order to control Mainframe Access behavior. See *Configuration reference* for more information on the available parameters.

Note: Ensure that when editing or uploading the MFAXML configuration file that you use CCSID 1047. Otherwise the XML document might fail validation.

Enabling Passphrase Support

Mainframe Access Server supports passphrases up to 100 characters in length. To enable this support you need to enable the MFA_GUI_ACCEPT_PASSPHRASES configuration option in your MFAXML configuration file.

See *MFA_GUI_ACCEPT_PASSPHRASES* for more information.

Editing Mainframe Access Application Server Parameters

The MFAXML member also contains definitions that describe both application servers and services. Service definitions such as RIMS, MCO, MFA and ES-MTO provide an ID label and the LU 6.2 parameters needed to contact the server using SNA LU 6.2 protocols. The ID label is used in some client configurations to select a specific target server.

Application server's definitions such as DataConnect, Endevor, ChangeMan and AMS define address spaces that can be started and controlled by Mainframe Access server. The definition provides a JCL procedure name that can be used to start the address space, a server program name, values that govern the number of address spaces that can be started, etc.

If you use CICS Option or IMS Option, you must edit the Mainframe Access sample services definitions to customize them for your system. The application server and service definitions for the Mainframe Access Data Connect server will need to be edited only if you have changed the JCL procedure name or the VTAM APPL ACBNAME, or if the default logmode name cannot be used.

The sample definitions for services that you will not be using can be commented out and they will not affect the operation of Mainframe Access with services that you have customized. For example, if you are not using CICS Option you can leave the sample CICS target server definitions unchanged; or you can delete them altogether.

The sample server definitions are provided here as an example. Lines that begin with an asterisk are treated as comments and are skipped in processing.

```
<Configuration
                     TCP_PORT="2020"
                     NETWORK_ID="DDINET1"
                     LU62_APPLID="MFM62ACB"
                     APPLID_PASSWORD="MFM62PSW"
                     TRACING="0"
                     ORGANIZATION="YOUR_COMPANY_NAME"
>

    <!-- Application Server address space definitions  -->
    <ApplicationServers>
        <!-- Mainframe Access Server (Data Connect) address space -->
        <DataConnect PROCEDURE="MFAS"
        />

        <!-- Mainframe Access Application Server(s) for Endevor -->
        <Endevor     PROCEDURE="MFAAS" MAXIMUM="2" JOBNAME="MFAE"
        />

        <!-- IDCAMS address space - needed for data set RENAME support -->
        <AMS         PROCEDURE="MFAMS" MAXIMUM="1" JOBNAME="MFAA"
        />

        <!-- Mainframe Access Application Server for TSO command support -->
        <TSO         PROCEDURE="MFAAS" TSO_PROCEDURE="MFATSO"
                     TSO_JOBCHAR="W"
        />
    </ApplicationServers>

    <!-- Service Configuration -->
    <Services>
        <!-- MFA Data Connect server - only one DEFAULT server is supported --
>
        <MFA         ID="DEFAULT" LUNAME="MFA62ACB" MODENAME="#INTER"
                     TPNAME="FILESHR2" SYNCLEVEL="1" SECURITY="0"
        />

        <!-- Sample IMS servers - multiple servers may be defined -->
        <!--
        <RIMS        ID="IMSA" LUNAME="IMSSYSA" MODENAME="IBMRDB"
                     TPNAME="MFDBTP6" SYNCLEVEL="0" SECURITY="0"
        />

        <RIMS        ID="DEFAULT" LUNAME="IMSTEST" MODENAME="IBMRDB"
                     TPNAME="MFDBTP6" SYNCLEVEL="0" SECURITY="0"
        />
        -->

        <!-- Sample CICS servers - multiple servers may be defined -->
        <!--
        <MCO         ID="CICA" LUNAME="CICSSYSA" MODENAME="#INTER"
                     TPNAME="*" SYNCLEVEL="0" SECURITY="0"
        />

        <MCO         ID="DEFAULT" LUNAME="CICSTEST" MODENAME="#INTER"
                     TPNAME="*" SYNCLEVEL="0" SECURITY="0"
        />
        -->

        <!-- Sample ES/MSS servers - multiple servers may be defined -->
        <!--
        <ES-MTO      ID="MTO1" ACBNAME="MFAMTO1" IPADDRESS="YOUR.MTO.HOSTNAME"
                     PORT="2200" SESSIONS="4" SOCKETS="1"
        />

        <ES-MTO      ID="MTO2" ACBNAME="MFAMTO2" IPADDRESS="101.102.103.104"
                     PORT="3300" SESSIONS="4" SOCKETS="1"
        />
        -->
    </Services>
</Configuration>
```

The section in bold above highlights the area of the XML file that you can edit, add, or remove parameters in order to control Mainframe Access behavior in regard to *application servers* and *services*. See *Configuration reference* for more information on the available parameters.

Application Servers

Application Server Parameters for the Mainframe Access Data Connect Server

An example of the DataConnect configuration:

```
<!-- Mainframe Access Server (Data Connect) address space -->
<DataConnect PROCEDURE="MFAS"
/>
```

Use the following parameters to configure Data Connect:

PROCEDURE

The JCL procedure name that will be used when the address space is started.

See *Configuration reference* for more information on the parameters available for a Data Connect application server.

Application Server Parameters for Endevor support

An example of the Endevor configuration:

```
<!-- Mainframe Access Server (Data Connect) address space -->
<Endevor PROCEDURE="MFAAS" MAXIMUM="2" JOBNAME="MFAE"
/>
```

Use the following parameters to configure Endevor:

PROCEDURE

Specifies the JCL procedure name that is used when the address space is started.

MAXIMUM

Specifies the maximum number of address spaces that Mainframe Access server starts for Endevor support.

JOBNAME

Specifies the jobname prefix to be used for address spaces that are started for Endevor support.

See *Configuration reference* for more information on the parameters available for an Endevor application server.

Application Server Parameters for AMS/IDCAMS support

An example of the AMS configuration:

```
<!-- IDCAMS address space – needed for data set RENAME support -->
<AMS PROCEDURE="MFAAMS" MAXIMUM="1" JOBNAME="MFAA"
/>
```

Use the following parameters to configure AMS:

PROCEDURE

Specifies the JCL procedure name that is used when the address space is started.

MAXIMUM

Specifies the maximum number of address spaces that Mainframe Access server starts for IDCAMS support.

JOBNAME

Specifies the jobname prefix to be used for address spaces that are started for Endevor support.

See *Configuration reference* for more information on the parameters available for an AMS application server.

Application Server Parameters for TSO Command support

An example of the TSO command support configuration:

```
<!-- Mainframe Access Application Server for TSO command support -->
<TSO         PROCEDURE="MFAAS" TSO_PROCEDURE="MFATSO"
             TSO_JOBCHAR="W"
/>
```

Use the following parameters to configure TSO command support:

PROCEDURE

Specifies the JCL procedure name that is used when the address space is started.

TSO_PROCEDURE

Specifies the JCL procedure name that is used when the individual user server address space is started.

TSO_JOBCHAR

Specifies the character that will be appended to the TSO userid to form the jobname of the running user server.

See *Configuration reference* for more information on the parameters available for a TSO command server.

Application Server Parameters for ChangeMan support

An example of the ChangeMan configuration:

```
<!-- Mainframe Access Server (Data Connect) address space -->
<ChangeMan PROCEDURE="MFAAS"
/>
```

Use the following parameters to configure ChangeMan:

PROCEDURE

Specifise the JCL procedure name that is used when the address space is started.

See *Configuration reference* for more information on the parameters available for a ChangeMan application server.

Application Server Parameters for z/Server Scheduler support (deprecated)

Note: This z/Server scheduler configuration is deprecated, and provided for backward compatibility only.

An example of the z/Server scheduler configuration:

```
<!-- z/Server Scheduler -->
<Scheduler   SCHEDULER_NAME="TAURISPF" LISTENER_PORT="1200"
             FIRST_PORT="1201" LAST_PORT="1249"
             USER_SERVER_JOBNAME="IVPUSRT" CCSID="037"
             DEFAULT="1"
>
    <UserServer CCSID="037" />
</Scheduler>
```

See *Configuring z/Server feature support* for more information on configuring Mainframe Access z/Server feature support.

Services

Service Parameters for ES/MSS

An example of the ES/MSS configuration:

```
<!-- Sample ES/MSS servers - multiple servers may be defined --
>
<ES-MTO      ID="MTO1" ACBNAME="MFAMTO1" IPADDRESS="YOUR.MTO.HOSTNAME"
             PORT="2200" SESSIONS="4" SOCKETS="1"
/>

<ES-MTO      ID="MTO2" ACBNAME="MFAMTO2" IPADDRESS="101.102.103.104"
             PORT="3300" SESSIONS="4" SOCKETS="1"
/>
```

These statements define an ES/MSS server and provide MFA Server with the information needed to establish a TCP/IP socket connection to the ES/MSS server. This definition also creates a VTAM ACB that represents the ES/MSS server to the z/OS CICS system(s). Use the following parameters to configure ES-MTO:

ID

Specifies the SYSID of the ES/MSS server being defined.

ACBNAME

Specifies the name of the VTAM ACB associated with this ES/MSS server.

IPADDRESS

Specifies the internet host name or IP address of the ES/MSS server.

PORT

Specifies the port number where ES/MSS is listening for ISC connections.

SESSIONS

Specifies the number of concurrent conversations MFA Server can initiate to the ES/MSS server over a single socket connection.

SOCKETS

Specifies the maximum number of concurrent socket connections between MFA Server and the ES/MSS server.

See *Configuration reference* for more information on the parameters available for an ES/MSS services.

Service Parameters for Mainframe Access Data Connect Server

An example of the ES/MSS configuration:

```
<!-- MFA Data Connect server - only one DEFAULT server is supported -->
<MFA         ID="DEFAULT" LUNAME="MFA62ACB" MODENAME="#INTER"
             TPNAME="FILESHR2" SYNCLEVEL="1" SECURITY="0"
/>
```

These statements define an ES/MSS server and provide MFA Server with the information needed to establish a TCP/IP socket connection to the ES/MSS server. This definition also creates a VTAM ACB that represents the ES/MSS server to the z/OS CICS system(s). Use the following parameters to configure ES-MTO:

ID

Data Connect client requests do not specify a target server ID and Mainframe Access always looks for the DEFAULT Mainframe Access Data Connect target server definition. Specify DEFAULT.

LUNAME

The LU name of the Mainframe Access Data Connect server (also known as the ACBNAME or VTAM APPLID). Specify an LU name of up to eight characters.

MODENAME

The SNA log mode name that will be used by Mainframe Access to indirectly specify SNA session parameters when Mainframe Access initiates sessions with the Mainframe Access Data Connect server.

TPNAME

The transaction program name for Mainframe Access Data Connect server requests. Specify FILESHR2.

SYNCLEVEL

The SNA LU6.2 sync level option to be used on conversations with the Mainframe Access Data Connect server.

SECURITY

The security level to be indicated in SNA LU6.2 Attach FMH-5 requests sent to the Mainframe Access Data Connect server.

See *Configuration reference* for more information on the parameters available for an MFA Data Connect service.

Service Parameters for IMS Option (Remote IMS)

An example of the RIMS configuration:

```
<!-- Sample IMS servers - multiple servers may be defined --
>
<RIMS        ID="IMSA" LUNAME="IMSSYSA" MODENAME="IBMRDB"
             TPNAME="MFDBTP6" SYNCLEVEL="0" SECURITY="0"
/>

<RIMS        ID="DEFAULT" LUNAME="IMSTEST" MODENAME="IBMRDB"
             TPNAME="MFDBTP6" SYNCLEVEL="0" SECURITY="0"
/>
```

Use the following parameters to configure RIMS:

ID

If an IMS Option client request does not specify a target server ID or the specified target server ID does not exist, Mainframe Access selects the DEFAULT IMS target server, if one has been defined. This IMS target server ID name is used in the configuration of the client. When a Remote IMS request is sent this name is sent in the request data and is used to locate the target server definition for the IMS system that will receive the request. Specify DEFAULT or an ID of up to four alphanumeric characters.

LUNAME

The LU name of the IMS server (also known as the ACBNAME or VTAM APPLID). Specify an LU name of up to eight characters. This name must match the ACBNAME defined in an APPC/MVS LU definition for the target IMS system.

MODENAME

The SNA log mode name that will be used by Mainframe Access to indirectly specify SNA session parameters when Mainframe Access initiates sessions with the IMS server.

TPNAME

The IMS server transaction program name for IMS Option transactions.

SYNCLEVEL

The SNA LU6.2 sync level option to be used on conversations with the IMS server. Specify 0.

SECURITY

The security level to be indicated in SNA LU6.2 Attach FMH-5 requests sent to the IMS server.

See *Configuration reference* for more information on the parameters available for a Remote IMS service.

Service Parameters for CICS

An example of the CICS configuration:

```
<!-- Sample CICS servers - multiple servers may be defined --
>
<MCO         ID="CICA" LUNAME="CICSSYSA" MODENAME="#INTER"
             TPNAME="*" SYNCLEVEL="0" SECURITY="0"
/>

<MCO         ID="DEFAULT" LUNAME="CICSTEST" MODENAME="#INTER"
             TPNAME="*" SYNCLEVEL="0" SECURITY="0"
/>
```

Use the following parameters to configure RIMS:

ID

If a CICS client request does not specify a target server ID or the specified target server ID does not exist, Mainframe Access selects the DEFAULT CICS target server, if one has been defined. This CICS target server ID name is used in the configuration of the client. When a CICS request is sent to Mainframe Access this name is sent in the request data and is used to locate the target server definition for the CICS system that will receive the request. Specify DEFAULT or an ID of up to four alphanumeric characters.

LUNAME

Specifies the LU name of the CICS server (also known as the ACBNAME or VTAM APPLID). Specify a LU name of up to eight characters.

MODENAME

Specifies the SNA log mode name that is used by Mainframe Access to indirectly specify SNA session parameters when Mainframe Access initiates sessions with mainframe CICS.

TPNAME

Specifies the server transaction program name for CICS requests.

SYNCLEVEL

Specifies the SNA LU6.2 sync level option to be used on conversations with CICS. Specify 0.

SECURITY

Specifies the security level to be indicated in SNA LU6.2 Attach FMH-5 requests sent to the CICS server.

Editing Access List Definitions

You must edit the access list definitions located in member ACCESS to customize them for your installation. The definitions in this list are:

- PERMIT and REJECT lines - these specify the IP addresses from which client connections are allowed or rejected. Client connection requests are checked only if the Mainframe Access parameter ACCESS_LIST_CHECK is set to YES. You must customize the PERMIT and REJECT lines with appropriate values for your implementation

The sample access list definitions are provided here as a convenient reference while you read. Lines that begin with an asterisk are treated as comments and are skipped in processing:

```
*---------------------------------------------------------------------*
*                                                                     *
* Micro Focus Mainframe Access - Access List                          *
*                                                                     *
*---------------------------------------------------------------------*
*
*---------------------------------------------------------------------*
* Permission list                                                     *
*---------------------------------------------------------------------*
*
PERMIT=111.111.111
*
*---------------------------------------------------------------------*
* Rejection list                                                      *
*---------------------------------------------------------------------*
*
REJECT=222.222.222
*
END
```

Permission and Rejection Lists

If you specify ACCESS LIST CHECK=YES in the Mainframe Access parameter file and specify the access list filename in an //XDBACC DD statement in the Mainframe Access startup JCL, client connection requests are accepted or rejected according to the values you specify for the PERMIT and REJECT parameters:

```
PERMIT=ipaddress
REJECT=ipaddress
```

The format of the IP address is the familiar dotted decimal notation. A complete Internet address consists of four decimal numbers or address groups, each in the range 0 through 255, with the groups being separated by decimal points.

You can specify complete Internet addresses to permit or reject individual clients.

You can also specify a range of consecutive Internet addresses in a single statement, by omitting one or more trailing address groups. This is known as a masked address specification. To decide whether or not a particular request is to be allowed, Mainframe Access first shortens the client address by removing trailing address groups that correspond to address groups omitted in the specification, then compares the shortened address with the masked address.

If you are using the access list mechanism, client connection requests from all addresses not specified in PERMIT or REJECT parameters are rejected.

Here are two example sets of access list parameters

Example 1

```
PERMIT = 111.111.148
REJECT = 111.111.148.244
```

The effect of these parameter values is that all client connection requests from IP addresses in the range 111.111.148.0 to 111.111.148.255 are accepted with the exception of client address 111.111.148.244. Client connections from all other IP addresses are rejected.

Example 2

```
PERMIT = 11.1.82
REJECT = 11.1.82.4
PERMIT = 111.202.111.18
```

The effect of these parameter values is that all client connection requests from IP addresses in the range 11.1.82.0 through 11.1.82.255 are accepted with the exception of client address 11.1.82.4. Client connections requests from the address 11.202.111.18 are also accepted. Connection requests from all other IP addresses are rejected

Allocating and Initializing an Audit Log Data Set

The Mainframe Access audit log feature uses a VSAM data set to record client login and logout information. The information that is logged includes the user ID, client IP address, login and logout times, target server name and encryption usage. If you want to use the audit log feature you need to prepare the audit log data set and update the Mainframe Access JCL procedure.

Sample member AUDIT contains the JCL to allocate and initialize a Mainframe Access audit log VSAM data set. Edit the JOB card, data set names and volume serial number information, then submit the job and verify that the condition code for the INITLOG step is 0 (zero). Note that a condition code of 8 for the ALLOCLOG step is normal the first time that you run this job.

When editing the data set names you may use a global change command to change all instances of HLQ to the high level qualifier that you selected for Mainframe Access data sets during the installation procedure.

Edit the Mainframe Access startup JCL to make the audit log data set available. Do this by removing the keyword DUMMY and the trailing comma (DUMMY,) from the XDBAUDIT DD statement:

```

    //*
    //*----------------------------
    //* Optional audit log data set
    //*----------------------------
    //*
    //XDBAUDIT DD  DUMMY,DISP=SHR,DSN=&DSNQUAL..AUDIT

```

Adding Mainframe Access to TCP/IP's Autolog List

If you are using IBM's TCP/IP, you can add the Mainframe Access started task name (the name of the Mainframe Access JCL procedure in the system procedure library) to the TCP/IP autolog list. TCP/IP will then automatically start Mainframe Access when it starts and stop Mainframe Access when it stops. Do not perform this update until you have finished configuring Mainframe Access and are satisfied with the way Mainframe Access is running.

Setting Up a Mainframe Access GTF Procedure

Mainframe Access provides sample members MFAGTF and GTFCNTL to assist you when you need to use IBM's Generalized Trace Facility (GTF) to obtain trace data for problem determination. In some problem situations our Product Support will request a VTAM trace of an application LU, usually the Mainframe Access server LU.

Adding the GTF Procedure to a System Procedure Library

Sample JCL to run GTF as a started task is provided in member MFAGTF. Edit this JCL to customize the data set names for your installation and then copy the JCL into a system procedure library that is available when z/OS START commands are processed. Member GTFCNTL contains GTF control statements that are referenced by the sample GTF JCL procedure. The USR=(FF1,FEF,0E9) statement in this member specifies the recording of VTAM buffer trace records (FF1 and FEF) and Mainframe Access server trace records (0E9).

Note: If you change the Mainframe Access EID="00E9" parameter setting, you should update member GTFCNTL with the new value. See *Editing Mainframe Access Parameters* for more information.

Testing the Mainframe Access GTF Procedure

After you customize and install procedure MFAGTF you can test out the procedure by issuing the following z/OS START command at a system console or using the TSO/ISPF SDSF command facility:

```
S MFAGTF.MYGTF
```

GTF will start and read the GTFCNTL parameter file. When the *initialization complete* message is issued GTF is ready to record trace data.

```
S MFAGTF.MYGTF
$HASP100 MFAGTF   ON STCINRDR
IEF695I START MFAGTF WITH JOBNAME MFAGTF IS ASSIGNED TO USER RWITEK, GROUP
CSIDVLP
$HASP373 MFAGTF   STARTED
IEF403I MFAGTF - STARTED - TIME=19.08.44
AHL121I  TRACE OPTION INPUT INDICATED FROM MEMBER GTFCNTL OF PDS
<hlq>.CNTL
TRACE=USRP
USR=(FF1,FEF,0E9)
END
AHL103I  TRACE OPTIONS SELECTED --USR=(FEF,FF1,0E9)
AHL906I THE OUTPUT BLOCK SIZE OF    27998 WILL BE USED FOR OUTPUT 228
        DATA SETS:
          <hlq>.TRACE
AHL080I GTF STORAGE USED FOR GTF DATA: 229
        GTFBLOCK STORAGE       82K BYTES (BLOK=        40K)
        PRIVATE STORAGE      1038K BYTES
        SADMP HISTORY          54K BYTES (SADMP=       40K)
        SDUMP HISTORY          54K BYTES (SDUMP=       40K)
        ABEND DUMP DATA         0K BYTES (ABDUMP=       0K)
AHL031I GTF INITIALIZATION COMPLETE
```

You stop GTF by issuing the following z/OS STOP command at a system console or using the TSO/ISPF SDSF command facility:

```
P MYGTF
```

GTF acknowledges the stop command and closes down, as shown in the following example:

```
P MYGTF
AHL006I GTF ACKNOWLEDGES STOP COMMAND
IEF404I MFAGTF - ENDED - TIME=19.10.31
$HASP395 MFAGTF   ENDED
$HASP250 MFAGTF PURGED -- (JOB KEY WAS B9C101AA)
```

Defining RACF Profiles for JES Spool Access

MFA now uses the RACF classes JESSPOOL and JESJOBS to check a user's authority to read or delete a job on the spool. See *z/OS Security Server RACF Security Administrator's Guide* for more information. When a request to read (IMPORT) a job or SYSOUT is received the following JESSPOOL profile is checked:

```
nodename.userid.jobname.jobid.dsidentifier.name
```

When a CANCEL, HOLD, or RELEASE command is received one of the following JESJOBS profiles are checked:

```
CANCEL.nodename.userid.jobname
HOLD.nodename.userid.jobname
RELEASE.nodename.userid.jobname
```

If no profile is defined then MFA reverts to checking the job's ownership.

Running Multiple Instances of Mainframe Access Server

Mainframe Access server is a robust server that is designed to efficiently handle all of the different client access requirements in high-volume usage. A single instance of Mainframe Access server will most likely meet the needs of your installation. It is, however, possible to run multiple instances of Mainframe Access server on your z/OS system. This might be desirable for establishing test and production servers, isolating specific user groups in their own server, isolating specific feature activations, etc.

Once you have a successful Mainframe Access server installation you can easily create additional instances as follows:

- Create and activate VTAMLST APPL definitions for a new Mainframe Access server and a new Mainframe Access Data Connect server.
- Reserve a unique TCP/IP port for the new Mainframe Access server (a TCP_PORT).
- Create unique MFAXML and ACCESS members for the new instance and customize them appropriately. You will definitely need to create a unique PARMS member and assign unique values to the LU62_APPLID="" and TCP_PORT="" parameters. It is possible to share ACCESS definitions, if that will meet the needs of your configuration. An alternative method would be to create a copy of the CNTL data set and retain the original member names.
- Allocate a unique audit log data set for the new instance if you are using this feature.
- Create a copy of the MFA proc in the proclib data set and update the JCL to reference the new MFAXML and ACCESS members for this instance. Also, update the audit log data set name if you are using this feature.
- Create a copy of the MFAS proc in the proclib data set and update the JCL EXEC statement PARM field to specify the new MFAS ACBNAME in the APPLID="" parameter. Also, update the VSAMCTL DD statement data set name if you want to allocate and manage a different set of mapping members for this instance.
- Update the MFAXML definition for the new instance with the name of the new proc that was created from the MFAS proc (that is, update the PROCEDURE="MFAS" value).

Troubleshooting

This chapter contains information that will help you to diagnose software problems related to Mainframe Access.

Abend Codes

An abend, or abnormal end, is the most obvious way for a software problem to be indicated. The z/OS processing for an abend normally creates a dump of the failing program's address space storage for use in problem determination. An abend can be initiated by Mainframe Access itself or by other software components of the z/OS system. Although a program abend is most often the result of a software error within the failing program, abend conditions can also be caused by other software and hardware components of the system.

System Abend Codes

Many software failures are detected by z/OS and its components. These conditions are normally reported as system abends using a code consisting of the letter S identifying a system abend and three hexadecimal digits, for example, S0C4. Complete descriptions of these conditions can found in the IBM publications for your specific version of z/OS. If a system abend is reported for Mainframe Access, it is important to review the abend description. This will help you to make an initial determination as to whether this is most likely a Mainframe Access program failure or a failure caused by other system conditions that have affected the execution of Mainframe Access.

Mainframe Access User Abend Codes

When Mainframe Access detects a condition that prevents the program from continuing, it requests z/OS to begin abend processing for a user abend. Many of these conditions are detected by Mainframe Access initialization or by the item library format utility. z/OS reports user abends using a code consisting of the letter U identifying a user abend and four decimal digits, for example, U2111. The user abend codes that can be issued by Mainframe Access and recommended actions are listed in the following table.

User Abend Code	Description of Cause	Recommended Action
99	Issued when a Data Set Services ISPI module cannot determine the invocation reason.	Contact our Product Support.
801	Processing to establish or remove a Mainframe Access ESTAEX error recovery routine has failed.	Contact our Product Support.
996	The MFA Server DSS Services component failed to initialize properly. This is a *should not occur* condition.	Contact our Product Support.
997	An Endevor error has occurred in the application server address space. Common causes include a general setup failure, MSGLOG allocation failure, or the Endevor C1DEFLTS module could not be found. Endevor requests cannot be processed.	Examine the joblog and syslog for system messages and server messages that can help identify the specific error. Contact our Product Support if you are unable to resolve the problem
998	There is a difference between the Endevor support modules available to the MFA Server control region and the Endevor support modules available in the MFA Server application server region(s). Different versions of Endevor are being referenced by the regions. Endevor requests cannot be processed.	This is most likely caused by differences in the STEPLIB or JOBLIB concatenations in the JCL used to start the regions. Contact our Product Support if you are unable to resolve the problem.
999	This abend code indicates that the Endevor support modules are not available to MFA Server. Endevor requests cannot be processed.	The Endevor AUTHLIB and CONLIB data sets must be included in the MFA started task JCL (for both the control region and the application server regions) or these data sets must be available in the standard system LNKLST specifications. Contact our Product Support if you are unable to resolve the problem.

User Abend Code	Description of Cause	Recommended Action
2100	Processing to establish or remove a Mainframe Access ESTAEX error recovery routine has failed.	Contact our Product Support.
3100	Processing associated with Mainframe Access' dependent addresss space services has encountered a *should not occur* condition.	Contact our Product Support.
3101	Work order allocation for a dependent address space service has failed. This is a *should not occur* condition.	Contact our Product Support.
3102	Allocation for a dependent address space service program call block has failed. This is a *should not occur* condition.	Contact our Product Support.
3103	Work order queuing for a dependent address space service has failed. This is a *should not occur* condition.	Contact our Product Support.

LU6.2 Diagnostic Information

Mainframe Access uses IBM's VTAM APPCCMD interface for LU6.2 communication with many of the target server systems, such as CICS, IMS and DB2. When problems arise and an error is reported for LU6.2 communication with one of these systems, the exact conditions of the error are indicated to Mainframe Access as return values from the APPCCMD call. These return values are logged to Mainframe Access' XDBOUT SYSOUT data set and can help you to identify and correct the situation. The following tables summarize the return value information. If you need more information, see the z/OS Communications Server SNA Programmer's LU 6.2 Reference manual for your level of z/OS and VTAM.

RTNCD and FDB2 Return Values for LU6.2 APPCCMD Calls

The RTNCD (return code) and FDB2 (feedback two) values are the major return values associated with an APPCCMD call. The following table summarizes these major return values.

RTNCD	FDB2	Explanation
X'00'	X'0B'	An error occurred during the execution of the APPCCMD call. The specific nature of the error is indicated by secondary return values, RCPRI and RCSEC, as documented in the next section
X'04'	X'05'	Symbolic name known by network-qualified name only
X'10'	X'13'	Attempt to start 6.2 session: request rejected
X'10'	X'14'	Attempt to start 6.2 session: pending session terminated
X'10'	X'15'	An APPCCMD must be issued
X'14'	X'7F'	Policing error: non-APPC macro

RCPRI and RCSEC Return Values for LU6.2 APPCCMD Calls

The RPL extension contains two fields in which return code information is passed to the application program at the completion of an LU6.2 APPCCMD macroinstruction execution. The RCPRI field returns a primary return code to the application; the RCSEC field returns a secondary return code to the application.

RCPRI	RCSEC	Explanation
X'0000'	X'0000'	OK; no errors
X'0000'	X'0001'	As specified; CNOS values were accepted
X'0000'	X'0002'	As negotiated; CNOS values changed by negotiation
X'0000'	X'0003'	Receive specific rejected
X'0000'	X'0004'	Partner LU supports single session
X'0000'	X'0005'	Internal VTAM error
X'0000'	X'0006'	Restore unnecessary; no modes to restore
X'0000'	X'0007'	Restore complete; input work area too small
X'0000'	X'0008'	No immediately available information
X'0000'	X'0009'	Request terminated by end of conversation
X'0000'	X'000A'	Sessions will use appl name, generic name requested
X'0000'	X'000B'	Sessions will use generic name, appl name requested
X'0000'	X'000C'	As specified, partner LU known by different name
X'0000'	X'000D'	As negotiated, partner LU known by different name
X'0004'	ALL	Allocation error
X'0004'	X'0000'	Allocation failure no retry
X'0004'	X'0001'	Allocation failure retry
X'0004'	X'0002'	Conversation type mismatch
X'0004'	X'0003'	PIP not allowed
X'0004'	X'0004'	PIP not specified correctly
X'0004'	X'0005'	Security not valid
X'0004'	X'0006'	Sync level not supported by LU
X'0004'	X'0007'	Sync level not supported by program
X'0004'	X'0008'	Transaction program name (TPN) not recognized
X'0004'	X'0009'	Transaction program name (TPN) not available; no retry
X'0004'	X'000A'	Transaction program name (TPN) not available; retry
X'0004'	X'000B'	Cannot reconnect transaction program; no retry
X'0004'	X'000C'	Cannot reconnect transaction program; retry
X'0004'	X'000D'	Reconnect not supported by program
X'0004'	X'000E'	Mode must be restored before using

RCPRI	RCSEC	Explanation
X'0004'	X'000F'	Deallocation requested
X'0004'	X'0010'	Allocation error - sync level not valid for full duplex
X'0004'	X'0011'	Allocation error - LU pair not supporting FDX conversation
X'0008'	ALL	CNOS failure
X'0008'	X'0000'	Allocation failure; retry
X'0008'	X'0001'	Allocation failure; no retry
X'0008'	X'0002'	Transaction program not available; retry
X'0008'	X'0003'	Transaction program not available; no retry
X'0008'	X'0004'	Conversation type mismatch
X'0008'	X'0005'	Security not valid
X'0008'	X'0006'	Mode must be restored before using
X'0008'	X'0007'	Network qualified name mismatch
X'000C'	X'0000'	CNOS resource failure; no retry
X'0010'	X'0000'	Partner granted retry
X'0010'	X'0001'	Control operator for local LU retried
X'0010'	X'0002'	Partner CNOS in progress
X'0010'	X'0003'	LU in pending single state
X'0010'	X'0004'	Partner LU starting session
X'0014'	X'0000'	Deallocate abend program
X'0018'	X'0000'	Deallocate abend service
X'001C'	X'0000'	Deallocate abend timer
X'0020'	X'0000'	CNOS failure; retry
X'0024'	X'0000'	Logical record boundary error
X'0028'	X'0000'	LU mode session limit closed
X'002C'	ALL	Parameter error
X'002C'	X'0000'	Invalid LU name or network identifier
X'002C'	X'0001'	Invalid mode
X'002C'	X'0002'	Invalid conversation
X'002C'	X'0003'	Invalid LL
X'002C'	X'0004'	Invalid values for SNASVCMG mode
X'002C'	X'0005'	Invalid DRAINL change
X'002C'	X'0006'	SNASVCMG mode cannot currently be reset
X'002C'	X'0007'	MINWINL plus MINWINR exceeds SESSLIM
X'002C'	X'0008'	Supplied length insufficient
X'002C'	X'0009'	Incomplete structure supplied

RCPRI	RCSEC	Explanation
X'002C'	X'000A'	Incomplete FMH5 supplied
X'002C'	X'000B'	Incomplete GDS variable supplied
X'002C'	X'000C'	Zero EXIT field
X'002C'	X'000D'	Zero ECB field
X'002C'	X'000E'	Request invalid for address space
X'002C'	X'000F'	Control block invalid
X'002C'	X'0010'	Invalid data address or length
X'002C'	X'0011'	Previous macroinstruction outstanding
X'002C'	X'0012'	Buffer list length invalid
X'002C'	X'0013'	No corresponding mode in LM table
X'002C'	X'0014'	Invalid BIND parameters
X'002C'	X'0015'	Invalid TPN
X'002C'	X'0016'	No corresponding LU in LM table
X'002C'	X'0017'	Invalid mode specified
X'002C'	X'0018'	Invalid limit specified
X'002C'	X'0019'	SNASVCMG mode already initialized
X'002C'	X'001A'	All modes specified on single session LU
X'002C'	X'001B'	SNASVCMG or CPSVCMG mode for single session LU
X'002C'	X'001C'	Single session, mode already initialized
X'002C'	X'001E'	CID invalid
X'002C'	X'001F'	APPCCMD issued for non-APPC
X'002C'	X'0020'	Previous REJECT request outstanding
X'002C'	X'0021'	Abnormal deallocate rejected; retry
X'002C'	X'0022'	Invalid CONTROL or QUALIFY value
X'002C'	X'0023'	Invalid session instance identifier
X'002C'	X'0024'	PS header not supplied
X'002C'	X'0025'	PS header length is insufficient
X'002C'	X'0026'	Session instance identifier and conversation identifier mismatch
X'002C'	X'0027'	Invalid deactivation type code
X'002C'	X'0028'	Cryptography not allowed on mode
X'002C'	X'0029'	Invalid LIST value specified on APPCCMD for restore
X'002C'	X'002A'	Invalid CGID value specified
X'002C'	X'002B'	Network-qualified name required
X'002C'	X'002C'	Parameter error; invalid expedited data length

RCPRI	RCSEC	Explanation
X'002C'	X'002D'	Parameter error; invalid sense code value specified
X'002C'	X'002E'	Vector area not valid
X'002C'	X'002F'	Vector area length insufficient
X'002C'	X'0030'	Parameter error; storage type not valid
X'002C'	X'0031'	Parameter error; SENDRCV specified with OPTCD=BUFFLST/XBUFLST
X'002C'	X'0032'	Parameter error; unexpected vector provided on APPCCMD
X'002C'	X'0033'	Parameter error; required vector not provided or incorrect
X'002C'	X'0034'	Password substitution value set in error
X'0030'	X'0000'	Program error; no truncation
X'0034'	X'0000'	Program error purging
X'0038'	X'0000'	Program error truncating
X'003C'	X'0000'	Service error; no truncation
X'0040'	X'0000'	Service error purging
X'0044'	X'0000'	Service error truncating
X'0048'	X'0000'	Resource failure; no retry
X'004C'	X'0000'	Resource failure; retry
X'0050'	X'0000'	State error
X'0054'	X'0000'	Unrecognized mode name
X'0058'	X'0000'	Unsuccessful; session not available
X'005C'	ALL	User error code received
X'005C'	X'0000'	Following negative response
X'005C'	X'0001'	Without negative response
X'0060'	X'0000'	No FMH5 available
X'0064'	X'0000'	Activation failure
X'0068'	X'0000'	LU mode session limit exceeded
X'006C'	X'0000'	Session not pending
X'0070'	X'0000'	Temporary storage shortage or resource shortage
X'0074'	X'0000'	Halt issued
X'0078'	X'0000'	VTAM inactive for your ACB
X'007C'	X'0000'	Request aborted
X'0080'	X'0000'	Deallocate normal
X'0084'	X'0000'	Storage shortage
X'0088'	X'0000'	Canceled by reject or abnormal deallocate
X'008C'	X'0000'	Partner committed protocol violation

RCPRI	RCSEC	Explanation
X'0090'	X'0000'	Application not APPC capable
X'0094'	X'0000'	Invalid condition for sending data
X'0098'	X'0000'	Temporary storage shortage while sending data
X'009C'	X'0000'	Restore rejected - restore issued before SETLOGON start
X'00A0'	ALL	Request not allowed
X'00A0'	X'0001'	LU pair does not support sending expedited data
X'00A0'	X'0002'	Request blocked
X'00A0'	X'0003'	Execution of request terminated
X'00A0'	X'0004'	CONTROL/QUALIFY value invalid for full-duplex conversation
X'00A0'	X'0005'	Response has not been received for a previous SENDEXPD request
X'00A0'	X'0006'	Program not authorized for requested function
X'00A4'	X'0000'	Mode must be restored before using
X'00A8'	ALL	Environment error
X'00A8'	X'0000'	OS level does not support request function
X'00A8'	X'0001'	Suspend failure
X'00A8'	X'0002'	Resume failure
X'00AC'	ALL	Error indication received
X'00AC'	X'0001'	Deallocate abend program
X'00AC'	X'0002'	Deallocate abend service
X'00AC'	X'0003'	Deallocate abend time
X'00AC'	X'0004'	Allocation error
X'00AC'	X'0005'	Unknown error code
X'00AC'	X'0006'	Resource failure; retry
X'00AC'	X'0007'	Resource failure; no retry
X'00B0'	ALL	Name resolution error
X'00B0'	X'0001'	LU name found in a variant name entry
X'00B0'	X'0002'	Name returned differs from associated name
X'00B0'	X'0003'	Name returned found in variant name entry
X'00B0'	X'0004'	Name returned found in supplied name entry
X'00B0'	X'0005'	Partner network name mismatch
X'00B0'	X'0006'	LU name found in an unusable name entry

RCPRI	RCSEC	Explanation
X'00B0'	X'0007'	Name returned found in an unusable name entry
X'00B0'	X'0008'	LU name found in a disassociated name entry
X'00B4'	ALL	CSM detected error
X'00B4'	X'0001'	CSM detected error - not specified
X'00B4'	X'0002'	CSM detected error - invalid buffer token specified
X'00B4'	X'0003'	CSM detected error - invalid instance id specified

Mainframe Access Host Error Codes

The return code and reason code values documented in the following table may appear in error responses sent from Mainframe Access to client products.

Return Code	Reason Code	Explanation	Action
X'00000008'	X'00000001'	XERROR_ALLOC The LU6.2 session allocation failed.	Check the target DB2 APPLID or MODNAME.
X'00000008	X'00000002'	XERROR_ALLOC The LU6.2 session allocation failed without a conversation block.	Check the APPLID and MODNAME for the target DB2.
X'00000009'	X'00000001'	XERROR_INVALIDPACKET An invalid packet type; the correct packet is an allocation packet.	Check the client program.
X'00000009'	X'00000002'	XERROR_INVALIDPACKET An invalid packet type; the correct packet is a dellocation data packet.	Contact our Product Support.
X'00000010'	X'FFFFFFFF'	XERROR_SEND The maximum number of concurrent users was reached.	Contact a Micro Focus sales representative to purchase more seats for your license.
X'00000011'	X'00000001'	XERROR_RECEIVE The LU6.2 session failed in the receive mode.	Contact our Product Support.
X'00000013'	X'00000001'	XERROR_FUNCTIONNOT ALLOWED The IMS/VSAM access feature is not available at your site.	Contact a Micro Focus sales representative to purchase a license for this feature.
X'00000013'	X'00000002'	XERROR_FUNCTIONNOT ALLOWED	Contact a Micro Focus sales representative to purchase a license for this feature.

Return Code	Reason Code	Explanation	Action
		The PEM feature is not available at your site.	
X'00000014'	X'00000001'	XERROR_INVALIDTRAN The transaction type in the first packet is invalid.	Contact our Product Support.
X'00000015'	X'00000001'	XERROR_CONV The LU6.2 session lost the conversation control block.	Contact our Product Support.
X'00000017'	X'00000rpl'	XERROR_SENDRPL The LU6.2 session failed with the RPL code in the reason code block.	Contact our Product Support.
X'00000018'	X'00000rpl'	XERROR_RECEIVERPL The LU6.2 session failed in receive mode with the RPL code in the reason code block.	Contact our Product Support.
X'00000100'	X'00000100'	XERROR_INTERNAL Unknown request type.	Contact our Product Support.

Mainframe Access Remote IMS Error Codes

Return Code	Reason Code	Description of Cause	Action
400	Bad request type	TYPE=ALLOCERR: Request type is invalid	The first packet's request type must be ALLOC. Resubmit an ALLOC packet.
400	Bad request, ID block is missing	TYPE=ALLOCERR: Mainframe Access does not have the configuration for the IMSID	Check the IMSID and resubmit.
505	Version not supported	TYPE=ALLOCERR: The client is not supported by Mainframe Access	Contact our Product Support.
602	IMS Applid is not valid	TYPE=ALLOCERR: Either IMS is not available or IMS/APPC is not enabled.	Contact our Product Support.
604	IMS Alloc Failure, RC = *rc*	TYPE=ALLOCERR: Routine failed to initiate a conversation with the target IMS system. *rc* is the decimal number of the return code from VTAM.	Contact our Product Support.
606	Receive Error	TYPE=RPL,CODE=*rc*: Conversation to IMS has failed during a receive	Check the entries, correct any errors, and resubmit request.

Return Code	Reason Code	Description of Cause	Action
		operation with RPL code. *rc* is the return code. The first two bytes are the RPL primary code; the next two bytes are the RPL secondary code.	
606	Send Error	TYPE=RPL,CODE=*rc*: Conversation to IMS has failed during a send operation with RPL code. The first two bytes are the RPL primary code; the next two bytes are the RPL secondary code.	Check the entries, correct any errors, and resubmit request.
606	LU6.2 Error	TYPE=RPL,CODE=*rc*: Conversation to IMS has failed with RPL code. The first two bytes are the RPL primary code; the next two bytes are the RPL secondary code.	Check the entries, correct any errors, and resubmit request.
613	Userid/Password Required	TYPE=ALLOCERR: Security is enforced in the configuration and user ID or password in the request packet is missing.	Check and correct user ID and password and resubmit request packet.
699	Mainframe Access HTTP Parsing Error	TYPE=ALLOCERR: The input HTTP packet has invalid record or format.	Check the record or format of the HTTP packet and resubmit.

IBM SAF and RACF Error Codes

The return code and reason code values documented in the following table may appear in error messages associated with z/OS security subsystem request failures.

Error Codes for VERIFY Requests

The return code and reason code values documented in the following table are associated with RACROUTE REQUEST=VERIFY requests that are issued to authenticate a user based on the userid and password that were submitted.

SAF Return Code	RACF Return Code	RACF Reason Code	Description
X'00'	X'00'	X'0000'	Request completed successfully.
X'00'	X'04'	X'000C'	Request completed successfully. TOKNIN was specified but the length was too large.
X'00'	X'04'	X'0010'	Request completed successfully.

SAF Return Code	RACF Return Code	RACF Reason Code	Description
			STOKEN was specified but the length was too large.
X'04'	X'00'	X'0000'	Request could not be completed. No RACF decision was possible. ENVIR=VERIFY was specified without SAF installation exit processing.
X'04'	X'04'	X'0000'	Request could not be completed. No RACF decision was possible. The user profile is not defined to RACF.
X'04'	X'20'	X'0000'	Request could not be completed. No RACF decision was possible. RACF is not active.
X'04'	X'58'	X'0000'	Request could not be completed. No RACF decision was possible. RJE or NJE operator FACILITY class profile not found.
X'08'	X'04'	X'0000'	Request failed. The user profile is not defined to RACF.
X'08'	X'08'	X'0000'	Request failed. The password is not authorized.
X'08'	X'0C'	X'0000'	Request failed. The password has expired.
X'08'	X'10'	X'0000'	Request failed. The new password is not valid.
X'08'	X'14'	X'0000'	Request failed. The user is not defined to the group.
X'08'	X'18'	X'0000'	Request failed. RACROUTE REQUEST=VERIFY was failed by the installation exit routine.
X'08'	X'1C'	X'0000'	Request failed. The user's access has been revoked.
X'08'	X'24'	X'0000'	Request failed.

SAF Return Code	RACF Return Code	RACF Reason Code	Description
			The user's access to the specified group has been revoked.
X'08'	X'28'	X'0000'	Request failed. OIDCARD parameter is required but not supplied.
X'08'	X'2C'	X'0000'	Request failed. OIDCARD parameter is not valid for specified user.
X'08'	X'30'	X'0000'	Request failed. The user is not authorized to the port of entry in the TERMINAL, JESINPUT, or CONSOLE class. Indicates the user is not authorized to the port of entry.
X'08'	X'30'	X'0004'	Request failed. The user is not authorized to the port of entry in the TERMINAL, JESINPUT, or CONSOLE class. Indicates the user is not authorized to access the system on this day, or at this time of day.
X'08'	X'30'	X'0008'	Request failed. The user is not authorized to the port of entry in the TERMINAL, JESINPUT, or CONSOLE class. Indicates the port of entry may not be used on this day, or at this time of day.
X'08'	X'34'	X'0000'	Request failed. The user is not authorized to use the application.
X'08'	X'38'	X'0004'	Request failed. SECLABEL checking failed. MLACTIVE requires a SECLABEL; none was specified.
X'08'	X'38'	X'0008'	Request failed. SECLABEL checking failed. Indicates the user is not authorized to the SECLABEL.
X'08'	X'38'	X'000C'	Request failed. SECLABEL checking failed. The system was in a multilevel secure status,

SAF Return Code	RACF Return Code	RACF Reason Code	Description
			and the dominance check failed.
X'08'	X'38'	X'0010'	Request failed. SECLABEL checking failed. Neither the user's nor the submitter's SECLABELs dominate. They are disjoint.
X'08'	X'44'	X'0000'	Request failed. A default token is used as input token.
X'08'	X'48'	X'0000'	Request failed. Indicates that an unprivileged user issued a RACROUTE REQUEST=VERIFY in a tranquil state (MLQUIET).
X'08'	X'4C'	X'0000'	Request failed. NODES checking failed. Submitter's node is not allowed access to execution node.
X'08'	X'4C'	X'0004'	Request failed. NODES checking failed. NJE failure: UACC of NONE for USERID type of NODES profile.
X'08'	X'4C'	X'0008'	Request failed. NODES checking failed. NJE failure: UACC of NONE for GROUP type of NODES profile.
X'08'	X'4C'	X'000C'	Request failed. NODES checking failed. NJE failure: UACC of NONE for SECLABEL type of NODES profile.
X'08'	X'4C'	X'0010'	Request failed. NODES checking failed. NJE failure: No local submit node specified.
X'08'	X'4C'	X'0014'	Request failed. NODES checking failed. NJE failure: Reverification of translated values failed.
X'08'	X'50'	X'0004'	Request failed. Indicates that a surrogate submit attempt failed. Indicates the SURROGAT class was inactive.

SAF Return Code	RACF Return Code	RACF Reason Code	Description
X'08'	X'50'	X'0008'	Request failed. Indicates that a surrogate submit attempt failed. Indicates the submitter is not permitted by the user's SURROGAT class profile.
X'08'	X'50'	X'000C'	Request failed. Indicates that a surrogate submit attempt failed. Indicates that the submitter is not authorized to the SECLABEL under which the job is to run.
X'08'	X'54'	X'0000'	Request failed. Indicates that a JESJOBS check failed.
X'08'	X'64'	X'0000'	Request failed. Indicates that the CHECK subparameter of the RELEASE keyword was specified on the execute form of the RACROUTE REQUEST=VERIFY macro; however, the list form of the macro does not have the same release parameter. Macro processing terminates.

Error Codes for AUTH Requests

The return code and reason code values documented in the following table are associated with RACROUTE REQUEST=AUTH requests. AUTH requests are issued to check the authorization of an already verified user to access a protected resource. The most common use of AUTH is to check if a user is authorized to access a specific data set.

SAF Return Code	RACF Return Code	RACF Reason Code	Description
X'00'	X'00'	X'0000'	Request completed successfully. The user is authorized by RACF to obtain use of a RACF-protected resource.
X'00'	X'00'	X'0004'	Request completed successfully. The user is authorized by RACF to obtain use of a RACF-protected resource. Indicates one of the following: • STATUS=ERASE was specified and the data set is to be erased when scratched

SAF Return Code	RACF Return Code	RACF Reason Code	Description
			• the warning status of the resource was requested by the RACROUTE REQUEST=AUTH issuer's setting bit X'10' at offset 12 decimal in the request-specific portion of the RACROUTE REQUEST=AUTH parameter list with the resource in warning mode.
X'00'	X'00'	X'0010'	Request completed successfully. The user is authorized by RACF to obtain use of a RACF-protected resource. When CLASS=TAPEVOL, indicates the TAPEVOL profile contains a TVTOC.
X'00'	X'00'	X'0020'	Request completed successfully. The user is authorized by RACF to obtain use of a RACF-protected resource. When CLASS=TAPEVOL, indicates that the TAPEVOL profile can contain a TVTOC, but currently does not (for a scratch pool volume).
X'00'	X'00'	X'0024'	Request completed successfully. The user is authorized by RACF to obtain use of a RACF-protected resource. When CLASS=TAPEVOL, indicates that the TAPEVOL profile does not contain a TVTOC.
X'00'	X'14'	X'00XX'	Request completed successfully. Requested function with STATUS=ACCESS specified has completed successfully. The user's highest access to the specified resource is indicated by one of the following reason codes: • 00 The user has no access. • 04

SAF Return Code	RACF Return Code	RACF Reason Code	Description
			The user has READ authority. - 08 The user has UPDATE authority. - 0C The user has CONTROL authority. - 10 The user has ALTER authority.
X'04'	X'00'	X'0000'	Request could not be completed. No RACF decision was possible. No security decision could be made. RACF is not installed -or- the specified requester, subsystem, or class is not in the RACF router table -or- the specified class is not in the RACF class descriptor table.
X'04'	X'04'	X'0000'	Request could not be completed. No RACF decision was possible. The specified resource is not protected by RACF. **Note:** Note: If PROTECTALL is active, no profile is found, and the user ID whose authority was checked does not have the SPECIAL attribute, RACF returns a return code X'08' instead of a return code X'04' and denies access. One of the following has occurred: - There is no RACF profile protecting the resource - RACF is not active - Specified class is not active - Specified class requires SETROPTS RACLIST

			option to be active and it is not. • CLASS TEMPDSN was active and the dataset is a temporary dataset.
X'04'	X'04'	X'0004'	Request could not be completed. No RACF decision was possible. The specified resource is not protected by RACF. Indicates STATUS=ERASE was specified and the data set is to be erased when scratched.
X'08'	X'08'	X'0000'	Request failed. The user is not authorized by RACF to obtain use of the specified RACF-protected resource. Indicates a normal completion. A possible cause would be PROTECTALL is active, no profile is found, and the user ID whose authority was checked does not have the SPECIAL attribute.
X'08'	X'08'	X'0004'	Request failed. The user is not authorized by RACF to obtain use of the specified RACF-protected resource. Indicates STATUS=ERASE was specified and the data set is to be erased when scratched.
X'08'	X'08'	X'0008'	Request failed. The user is not authorized by RACF to obtain use of the specified RACF-protected resource. Indicates DSTYPE=T or CLASS=TAPEVOL was specified and the user is not authorized to use the specified volume.
X'08'	X'08'	X'000C'	Request failed. The user is not authorized by RACF to obtain use of the specified RACF-protected resource. Indicates the user is not authorized to use the data set.
X'08'	X'08'	X'0010'	Request failed. The user is not authorized by RACF to obtain use of the specified RACF-protected resource.

			Indicates DSTYPE=T or CLASS=TAPEVOL was specified and the user is not authorized to specify TAPELBL=(,BLP).
X'08'	X'08'	X'0014'	Request failed. The user is not authorized by RACF to obtain use of the specified RACF-protected resource. Indicates the user is not authorized to open a non-cataloged data set.
X'08'	X'08'	X'0018'	Request failed. The user is not authorized by RACF to obtain use of the specified RACF-protected resource. Indicates the user is not authorized to issue RACROUTE REQUEST=AUTH when system is in tranquil state (MLQUIET).
X'08'	X'08'	X'001C'	Request failed. The user is not authorized by RACF to obtain use of the specified RACF-protected resource. A user with EXECUTE authority to the data set profile specified ATTR=READ, and RACF failed the access attempt.
X'08'	X'08'	X'0020'	Request failed. The user is not authorized by RACF to obtain use of the specified RACF-protected resource. The user's security label does not dominate that of the resource; it fails SECLABEL authorization checking.
X'08'	X'08'	X'0024'	Request failed. The user is not authorized by RACF to obtain use of the specified RACF-protected resource. The user's security label can never dominate that of the resource.
X'08'	X'08'	X'0028'	Request failed. The user is not authorized by RACF to obtain use of the specified RACF-protected resource. The resource must have a security label, but does not have one.
X'08'	X'0C'	X'0000'	Request failed.

			The OLDVOL specified was not part of the multivolume data set defined by VOLSER, or it was not part of the same tape volume defined by ENTITY.
X'08'	X'10'	X'00XX'	Request failed. RACROUTE REQUEST=VERIFY was issued by a third party, and RACROUTE REQUEST=AUTH failed. The reason code 00XX value is the RACF return code from the RACROUTE REQUEST=VERIFY.
X'08'	X'64'	X'0000'	Request failed. Indicates that the CHECK subparameter of the RELEASE keyword was specified on the execute form of the RACROUTE REQUEST=AUTH macro; however, the list form of the macro does not have the same RELEASE parameter. Macro processing terminates.

Mainframe Access Audit Report Program

Mainframe Access' optional audit log feature records client logon and logoff information to a VSAM data set. For a description of how to allocate the audit log data set and enable the recording see the section *Allocating and Initializing an Audit Log Data Set* in the chapter *Configuration*. Mainframe Access also provides a batch reporting program to list the audit log records. This section describes the use of the audit log report program.

Running the Audit Report Program

Member AUDRPT in the Mainframe Access <hlq>.CNTL data set is a sample job for running the audit report program. Customize this job with a valid JOB card and the correct data set qualifier for your installation. The sample member is shown below.

```
//MFASETUP JOB (MFA),'RUN AUDIT REPORT',CLASS=A,MSGCLASS=A
//*
//AUDITRPT PROC DSNQUAL='HLQ'
//*
//*------------------------------------------------------------------*
//*                                                                  *
//* Micro Focus Mainframe Access - Run the Audit Log Report          *
//*                                                                  *
//* Change the DSNQUAL value on the PROC statement to the correct    *
//* value for your installation.                                     *
//*------------------------------------------------------------------*
//*
//*------------------------------------------------------------------*
//* Step 1: Run the audit report utility to list the records in the  *
//*         audit log dataset.                                       *
```

```
//*-----------------------------------------------------------------*
//*
//AUDITRPT EXEC PGM=AUDITRPT,REGION=0M
//STEPLIB  DD   DSN=&.DSNQUAL..LOADLIB,DISP=SHR
//AUDITIN  DD   DSN=&.DSNQUAL..AUDIT,DISP=SHR
//SYSPRINT DD   SYSOUT=*
//SYSTERM  DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//         PEND
//AUDITRPT EXEC AUDITRPT
```

When you have customized the job, submit it for execution. A report similar to the following is written to a SYSOUT data set.

```
                                                            Audit Log Report

  Report date and time: FRI JUL 25 16:37:57 2003

  Audit data set: <hlq>.AUDIT

  Type     Record    Sess  User      IP Address       Partner    Date
+Time                Type of
  of       Sequence  ID    ID
LU                                  Encryption used (see note)
  Record

  Logoff        1    00000 CSIRLW2   10.24.11.15      MFADIR     FRI JUL 25
16:02:51 2003
  Logoff        3    00000 CSIRLW1   10.24.11.15      MFADIR     FRI JUL 25
16:03:33 2003
  Logoff        5    00000                                       FRI JUL 25
16:04:22 2003 N
  Logoff       10    00000                                       FRI JUL 25
16:35:48 2003 N
  Logon         0    00001 CSIRLW2   10.24.11.15      MFADIR     FRI JUL 25
16:02:27 2003
  Logon         2    00002 CSIRLW1   10.24.11.15      MFADIR     FRI JUL 25
16:03:20 2003
  Logon         4    00003 CSIRLW1   10.24.11.15      LUDB27R    FRI JUL 25
16:03:59 2003 P
  Logon         6    00004           10.24.11.15      CICSTSR7   FRI JUL 25
16:34:32 2003 N
  Logon         7    00005           10.24.11.15      CICSTSR6   FRI JUL 25
16:34:33 2003 N
  Logon         8    00006           10.24.11.15      CICSTSR3   FRI JUL 25
16:34:34 2003 N
  Logon         9    00007 CSIRLW1   10.24.11.15      IMS6PPC    FRI JUL 25
16:35:42 2003 N

  Note: Encryption values   N = None, P = Password, X = Packet encryption, C
= Compression is in use

  Total type 0 records =         11
  Total logon records  =          7
  Total logoff records =          4
```

Audit Report Abend Codes

The following table lists the user abends that may be issued by the audit report program.

User Abend Code	Description of Cause	Recommended Action
1211	This abend is issued when a GETMAIN request fails. A console message is also issued to indicate the failure condition. A program dump is not requested.	Increase the region size for the audit report program. If the problem persists contact our Product Support.

Audit Report Diagnostic Messages

The following messages may be issued by the audit report program when error conditions occur.

AUD0001E: DDNAME xxxxxxxx WAS NOT FOUND

This message appears when either the SYSPRINT or AUDITIN DD statements are missing. Add the missing DD statements to the JCL for the job.

AUD0002E: OPEN FAILED FOR xxxxxxxx FILE

This message appears when either SYSPRINT or AUDITIN fails to open. Examine associated system error messages to determine the reason for the failure.

AUD0003E: VSAM ACB FAILED FOR xxxxxxxx
AUD0003E: RETURN CODE IS xxxxxxxx
AUD0003E: REASON CODE IS xxxxxxxx

These messages appear when a critical error occurs while accessing the AUDITIN VSAM data set. The program terminates with the return code 16. Examine any associated system error messages and the VSAM return code and reason code values to determine the reason for the failure.

Obtaining a VTAM Buffer Trace

During the process of resolving a problem, our Product Support may require you to obtain and submit a VTAM buffer trace (and possibly other VTAM trace data). This section describes how to use IBM's Generalized Trace Facility (GTF) to capture the relevant data.

The procedure for gathering the trace is as follows:

1. Make sure that Mainframe Access, the clients and the target servers are all ready and at a point where you want to start tracing. For example, if the problem is associated with a specific CICS distributed transaction, bring the client up to the point where the transaction will be issued.
2. Start GTF (see the section *Starting GTF*)
3. Activate the VTAM trace (see the section *Activating the VTAM Trace*)
4. Perform the activity that you want to trace (for example, run the transaction that fails)
5. Deactivate the VTAM trace (see the section *Deactivating the VTAM Trace*)
6. Stop GTF (see the section *Stopping GTF*)
7. Package the raw GTF trace data set for transmission to our Product Support.
8. Use the File Transfer Protocol (FTP) utility to send the packaged trace to Micro Focus. Use binary mode for the transfer.

Starting GTF

Mainframe Access provides a sample procedure, MFAGTF, for running GTF; its use is described in the section *Adding the GTF Procedure to a System Procedure Library* in the chapter *Configuration*. You may want to pre-allocate a permanent GTF trace data set on DASD that can be reused each time you start MFAGTF. If you do so, update the MFAGTF procedure to use this data set.

A PDS member is normally used to automate the specification of GTF options when GTF is started and the sample procedure uses parameters from another sample member, GTFCNTL. This sample is set up to start GTF for recording Mainframe Access' activity trace (USR '0E9' records) and VTAM's buffer trace (USR 'FEF' and 'FF1' records).

A sample system log for the startup of GTF is shown below. Remember to have Mainframe Access and other software components that are part of the tracing activity already running, connected and ready to go, if possible.

```
  S MFAGTF.MYGTF
  $HASP100 MFAGTF   ON STCINRDR
  IEF695I START MFAGTF WITH JOBNAME MFAGTF IS ASSIGNED TO USER RWITEK, GROUP
CSIDVLP
  $HASP373 MFAGTF   STARTED
  IEF403I MFAGTF - STARTED - TIME=16.34.12
  AHL121I  TRACE OPTION INPUT INDICATED FROM MEMBER GTFCNTL  OF PDS
  <hlq>.CNTL
  TRACE=USRP
  USR=(FF1,FEF,0E9)
  END
  AHL103I  TRACE OPTIONS SELECTED --USR=(FEF,FF1,0E9)
  AHL906I THE OUTPUT BLOCK SIZE OF    27998 WILL BE USED FOR OUTPUT 997
          DATA SETS:
            <hlq>.TRACE
  AHL080I GTF STORAGE USED FOR GTF DATA: 998
          GTFBLOCK STORAGE       82K BYTES (BLOK=         40K)
          PRIVATE STORAGE      1038K BYTES
          SADMP HISTORY          54K BYTES (SADMP=        40K)
          SDUMP HISTORY          54K BYTES (SDUMP=        40K)
          ABEND DUMP DATA         0K BYTES (ABDUMP=        0K)
  AHL031I GTF INITIALIZATION COMPLETE
```

GTF is now up and running and ready to record type 'FEF' and 'FF1' records from VTAM. z/OS may also force some USR records into the trace data set for system components. GTF wraps trace data when writing trace records to a DASD data set. This means that you can leave GTF tracing on for a long time to catch an intermittent problem, then stop the trace soon after the problem occurs. However, if you do not stop the trace soon after the problem occurs, the trace data that you collected may be overwritten by trace data from other system components.

Activating the VTAM Trace

When you are ready to perform an action that you want to trace, activate the VTAM buffer trace for the Mainframe Access ACB. You do this using a z/OS Modify command to VTAM. In this example the VTAM JCL procedure name is VTAM. The name may be different on your z/OS system. The only other thing you may want to change is the ID= parameter so that it specifies the name of your Mainframe Access VTAM ACB.

```
  F VTAM,TRACE,TYPE=BUF,ID=MFM62ACB,AMOUNT=FULL
  IST097I MODIFY ACCEPTED
  IST1515I BUFFER TRACE ACTIVE
  IST1144I TRACE INITIATED FOR DDINET1.MFM62ACB 316
  IST1045I NODE TYPE = APPL
  IST314I END
```

GTF should now be up and running and VTAM should be creating TYPE=BUF trace records for any activity associated with MFM62ACB. If someone has previously started a TYPE=BUF trace for another VTAM resource and did not bother to stop the trace, these records may also appear in the trace data set and complicate the analysis of what you are tracing.

Deactivating the VTAM Trace

When you have finished performing the activity to be traced, turn off the VTAM TYPE=BUF trace for the Mainframe Access ACB. Use the z/OS Modify command as follows:

```
F VTAM,NOTRACE,TYPE=BUF,ID=MFM62ACB
IST097I MODIFY ACCEPTED
IST1143I TRACE TERMINATED FOR DDINET1.MFM62ACB 349
IST1045I NODE TYPE = APPL
IST314I END
```

Stopping GTF

Stop the GTF process using the z/OS Stop (P) command. In this example GTF was started using "S MFAGTF.MYGTF" where the .MYGTF is assigned as an identifier for the GTF process that is then specified in the Stop command.

```
P MYGTF
AHL006I GTF ACKNOWLEDGES STOP COMMAND
AHL904I THE FOLLOWING TRACE DATASETS CONTAIN TRACE DATA : 353
          <hlq>.TRACE
IEF404I MFAGTF - ENDED - TIME=11.48.15
$HASP395 MFAGTF   ENDED
$HASP250 MFAGTF PURGED -- (JOB KEY WAS B5A444E4)
```

The trace data set, <hlq>.TRACE in this example, now has the raw GTF data.

Formatting VTAM GTF Trace Records

There may be times when you need, or want, to examine the GTF trace data. In this case, the raw data collected by GTF needs to be processed by a formatting program such as IBM's Interactive Problem Control System (IPCS). You can run IPCS either using a TSO command under ISPF or as a batch job. To run IPCS you need an IPCS dump directory VSAM data set. If you do not already have a dump directory, you can allocate one using a job similar to the following.

```
//jobname JOB (accounting),'name',CLASS=?,MSGCLASS=?,NOTIFY=userid
//--------------------------------------------------------------------
//* Allocate an IPCS dump directory VSAM data set
//--------------------------------------------------------------------
//IDCAMS   EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
 DELETE <hlq>.DIR
 DEFINE CLUSTER (NAME(<hlq>.DIR)                -
                SHAREOPTIONS(1)                 -
                RECSZ(256 3072)                 -
                KEYS(128 0)                     -
                VOLUMES(volser)
          INDEX (NAME(<hlq>.DIRINDEX)           -
                TRK(15 5))                      -
          DATA  (NAME(<hlq>.DIRDATA)            -
                CYLINDERS(5 1)                  -
                CISZ(X'1000'))
```

This example JCL runs IPCS to print all 'FEF' and 'FF1' records from the GTF trace data set.

```
//jobname JOB (accounting),'name',CLASS=?,MSGCLASS=?,NOTIFY=userid
//--------------------------------------------------------------------
//* Run IPCS batch to print GTF trace records
//--------------------------------------------------------------------
//IPCSBAT  EXEC PGM=IKJEFT01,DYNAMNBR=20,REGION=0M
```

```
//IPCSDDIR DD DSN=<hlq>.DIR,DISP=SHR
//IPCSPRNT DD SYSOUT=*
//TRACE    DD DSN=<hlq>.TRACE,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 IPCS NOPARM
 DROPDUMP DDNAME(TRACE)
 SETDEF DDNAME(TRACE) NOCONFIRM
 GTFTRACE DDNAME(TRACE),USR(FEF,FF1),TERMINAL,NOPRINT
 END
```

Obtaining an SVC Dump of Mainframe Access

During the process of resolving a problem, our Product Support may require you to obtain and submit a dump of the Mainframe Access address space. This section describes how to use IBM's DUMP command to create the dump data set. Using the following procedure, you can obtain a dump without stopping or canceling the Mainframe Access address space.

The procedure for creating the dump is as follows:

1. Issue the DUMP command
2. Respond to the DUMP command options request
3. Package the dump data set for transmission to our Product Support
4. Use the File Transfer Protocol (FTP) utility to send the packaged dump to Micro Focus. Use binary mode for the transfer.

The following example shows a DUMP command and the response to the request for operands in bold face. You use the COMM parameter of the DUMP command to specify a meaningful title for the dump. The operands specified in the sample reply, SDATA=(RGN,LSQA,PSA,SUM,SWA,TRT,CSA,LPA), are normally needed for any Mainframe Access dump; you should specify them as shown. Technical support may request additional, or different, options for a specific problem. After the options have been entered it will take the system anywhere from a few seconds to a few minutes to complete the dump. The IEA611I COMPLETE DUMP message indicates successful completion of the command and it identifies the dump data set that needs to be saved and packaged up for transmission. In this example, the system has created data set SYS1.CSIA.DMP00032 to contain the dump data.

```
DUMP COMM=(MFA DUMP JULY 28 2003)
*63 IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R 63,JOBNAME=MFA,SDATA=(RGN,LSQA,PSA,SUM,SWA,TRT,CSA,LPA)
 IEE600I REPLY TO 63 IS;JOBNAME=MFA,SDATA=(RGN,LSQA,PSA,SUM,SWA,TRT
 IEA794I SVC DUMP HAS CAPTURED: 138
 DUMPID=032 REQUESTED BY JOB (*MASTER*)
 DUMP TITLE=MFA DUMP JULY 28 2003
 IEF196I IGD100I 0309 ALLOCATED TO DDNAME SYS00036 DATACLAS (        )
 IEF196I IEF285I   SYS1.CSIA.DMP00032                           CATALOGED
 IEF196I IEF285I   VOL SER NOS= DUMP01.
 IEA611I COMPLETE DUMP ON SYS1.CSIA.DMP00032 142
 DUMPID=032 REQUESTED BY JOB (*MASTER*)
 FOR ASID (00A5)
 INCIDENT TOKEN: ROCKPLEX CSIA     07/28/2003 21:56:11
```

Packaging and Transmitting Diagnostic Data

This section provides suggestions and guidelines that will help if you need to send diagnostic data to our Product Support. When you need to send relatively small amounts of text data (for example, short Mainframe Access activity traces, parameter settings, excerpts from the system log, excerpts from a dump listing, etc.), it is often convenient to simply include the text in an email or as a text file attachment to an email. Large amounts of data and non-textual information should be packaged and transmitted as described in the following paragraphs.

After gathering large amounts of diagnostic data (for example, SVC dumps, GTF traces, Mainframe Access activity traces, system log contents, etc.), the most expedient way to send the information is using the TCP/IP File Transfer Protocol (FTP) utility. Before you send the data, it is good practice to package the data for transmission. This will reduce transmission time by compressing the data and it will facilitate the handling of data when it is received.

Our Product Support can accommodate data that is packaged using the TSO Interactive Data Transmission Facility (the TSO XMIT and RECEIVE commands) or data that is packaged using IBM's TRSMAIN program (the terse utility). Mainframe Access sample TSOXMIT provides a job that demonstrates the use of TSO XMIT to package a data set for transmission. If you want to use TRSMAIN, we recommend that you start by asking the z/OS systems programming staff at your company if TRSMAIN is already installed and available for use. If it is not, you can download the IBM TRSMAIN program from the ftp.software.ibm.com Web site using FTP. You can access the download using the anonymous user ID with your email address provided as the password. Use change directory commands to locate the TRSMAIN downloads in /s390/mvs/tools/packlib at this IBM ftp site. You should find a single binary file and both plain text and HTML explanations of the procedure for installing the utility.

When your data is ready for transmission, our Product Support can provide additional instructions for sending your data to the ftp.microfocus.com site. This may include specific directory locations, filenames, etc. that should be used. After you have transferred the data, please send a follow-up email to your support representative. This note should list the files that you have sent, the packaging tool(s) that you have used, and the z/OS data set characteristics of both the packaged files and the data sets that will be created when the packages are opened. The z/OS data set characteristics should include the RECFM, LRECL, BLKSIZE, primary space allocation and secondary space allocation.

Customizing Data Connect

This section describes advanced configuration for the Mainframe Access Data Connect server.

Introduction

At the completion of Mainframe Access installation and quick configuration, the Mainframe Access Data Connect server is ready to use for most Data Connect applications. This chapter will help you with the necessary customization if you need to change the default user verification and data set authorization checking. This chapter also describes how to customize a FSTAB file mapping table and the MFAVCTL file mapping data set.

Samples for Configuration

The installation procedure places several samples in the <hlq>.MFA.MFACNTL data set. If a sample is provided for a configuration task, use it as a starting point for your configuration. Usually you need to edit the sample, replacing provided information with information that is unique to your system.

The following table summarizes the samples that are provided for Mainframe Access Data Connect server.

MEMBER	DESCRIPTION
FSTAB	Assembler language macro used by the MFASFTAB sample.
MFASFTAB	Sample job and source statements for creating your own FSTAB module for file name mapping operations.

JCL Change for Data Connect

In order to supply all MFA and Data Connect function within one authorized library, the Data Connect module names have been renamed, and slightly reorganized as static load modules rather than dynamic overlays. All Data Connect load modules now begin with the prefix MFD.

```
//MFAS    PROC DSNQUAL='HLQ.MFA'
//*
//*------------------------------------------------------------------*
//*                                                                  *
//* Micro Focus Mainframe Access - Data Connect / FILESHARE          *
//*                                                                  *
//* Change the DSNQUAL value on the PROC statement to the correct    *
//* value for your installation.                                     *
//*------------------------------------------------------------------*
//MFAS     EXEC PGM=MFDSTART,PARM='APPLID=MFA62ACB',
//         REGION=0M,TIME=1440
//STEPLIB  DD DISP=SHR,DSN=&DSNQUAL..LOADLIB
//         DD DSNAME=CEE.SCEERUN,DISP=SHR
//         DD DSNAME=CEE.SCEERUN2,DISP=SHR
//CEEOPTS  DD DISP=SHR,DSN=&DSNQUAL..CNTL(CEEOPTS)
//SYSUDUMP DD SYSOUT=*
//FSMSGA   DD SYSOUT=*             for IDCAMS
//FSMSGC   DD SYSOUT=*             for APPC errors
//FSMSGL   DD SYSOUT=*             for FileShare log
//VSAMCTL  DD DISP=SHR,DSN=&DSNQUAL..MFAVCTL
```

Configuring Access to External Library Management Systems

This section describes the Mainframe Access support for external library management systems and change management systems.

About External Library Management

Mainframe Access provides seamless access to objects housed in repositories controlled by the following external management systems:

- Panvalet from Computer Associates (*www.cai.com*)
- Librarian from Computer Associates (*www.cai.com*)
- Endevor from Computer Associates (*www.cai.com*)
- ChangeMan ZMF

Contact the appropriate vendor for further information about these products.

Panvalet

Panvalet is a direct access library system marketed by Computer Associates. It's primary use is as a source control system for software development.

About Panvalet

With Panvalet, all source members are managed as fixed length records. The default size is the historical card image of 80 bytes. Panvalet may be used to manage any fixed-length data up to 4096 characters in length.

Panvalet provides licensed users with programmable access to their repository services via the PAM API. This is a read-only interface which permits multiple concurrent access for Mainframe Access threads to browse source members residing in Panvalet repositories. During Server initialization, Mainframe Access attempts to load this interface into memory. If this API interface is available through the standard search order (STEPLIB, JOBLIB, or SYS1.LINKLIB at the customer site) then Panvalet services are offered. Startup initialization messages indicate if Panvalet is available.

It is a customer responsibility to provide a suitable STEPLIB concatenation which enables selection of the desired release and version of the Panvalet components. At most sites, PANVALET is installed as an MVS Subsystem. Therefore the API is available within SYS1.LINKLIB. No override would be necessary unless a new version is being tested.

```
    //STEPLIB DD DSN=hlq.MFA.LOADLIB,DISP=SHR
    //        DD DSN=CAI.PANV144.CAILIB,DISP=SHR
```

The following messages are typical of the information written to the SYSLOG during Mainframe Access startup:

MFM0063I: MFA Direct is active MFM0064I: MFA Direct number of processing tasks is 5 MFMDS051I CA/LIBrarian Ready MFMDS052I CA/PanValet Ready MFMDS053I CA/ENDEVOR API 39 Ready MFMDS055I MCG/RPC feature active MFMDS001I DSS Server Ready MFM0001I: Mainframe Access V6.00 (04000000) is active

For source additions or updates under Panvalet, Mainframe Access must invoke the standard Panvalet batch utility PAN#1. Access to this utility is serialized, since Panvalet does not support multiple, concurrent access to this service. In some cases, you may have renamed the Batch Update utility PAN#1. If this is the case, the new name at this site must be defined to Mainframe Access as a parameter override.

References

PANVALET System Management Guide
PANVALET Messages Guide
PANVALET Installation Guide
PANVALET User Guide
PANVALET Extended Features Guide

Panvalet Suffix Table

Panvalet recognizes the following programming language or object types:

- BAL
- COBOL
- DATA
- FORTRAN
- JCL
- PL/I
- RPG

Specification of such a native language type allows Panvalet to format the source files, and automatically add or remove the sequence numbers in the appropriate position for that language. In order to automate this language selection process, Mainframe Access has introduced a Panvalet Suffix Table to specify the language type based upon the file suffix as used on the workstation. In addition, the Suffix Table also specifies the default options to be specified on the ADD command of the PAN#1 batch update utility.

SUFFIX	++ADD Options
(default)	DATA,LIST,NOFORMAT
ASM	BAL,LIST,NOFORMAT
ASSEM	BAL,LIST,NOFORMAT

BAL	BAL,LIST,NOFORMAT
BMS	BAL,LIST,NOFORMAT
CBL	COBOL,LIST,NOFORMAT
CNTL	JCL,LIST,NOFORMAT
COB	COBOL,LIST,NOFORMAT
COBOL	COBOL,LIST,NOFORMAT
COP	COBOL,LIST,NOFORMAT
COPY	COBOL,LIST,NOFORMAT
CPY	COBOL,LIST,NOFORMAT
DATA	DATA,LIST,NOFORMAT
DBD	BAL,LIST,NOFORMAT
JCL	JCL,LIST,NOFORMAT
JOB	JCL,LIST,NOFORMAT
MAC	BAL,LIST,NOFORMAT
MFS	BAL,LIST,NOFORMAT
PLI	PL/1,LIST,NOFORMAT
PL1	PL/1,LIST,NOFORMAT
PSB	BAL,LIST,NOFORMAT

The above table describes the default Suffix Table as provided within Mainframe Access. If no changes are desired, no further customization is necessary.

Suffix Table Override

The built-in Panvalet Suffix Table may be replaced by including a special DD card in the Mainframe Access task JCL as follows:

```
//PANSUFX  DD  DSN=hlq.MFA.CNTL(PANVALET),DISP=SHR
```

During initialization, this text file is analyzed and a new Suffix Table is created dynamically. The first entry defines a default language type to be used if no match is found in the table. The following extract illustrates the free-form format of the text file:

```
*
* MFA PANVALET CONFIGURATION CONVERSION TOOL
*      AS ON 06/07/2003 AT 12:40
*
* SUFFIX TYPE     ++ADD OPTIONS
******** ******** ***********************
*
DEFAULT  DATA     LIST,NOFORMAT
ASM      BAL      LIST,NOFORMAT
ASSEM    BAL      LIST,NOFORMAT
BAL      BAL      LIST,NOFORMAT
BMS      BAL      LIST,NOFORMAT
DBD      BAL      LIST,NOFORMAT
MAC      BAL      LIST,NOFORMAT
MFS      BAL      LIST,NOFORMAT
MLC      BAL      LIST,NOFORMAT
PSB      BAL      LIST,NOFORMAT
CBL      COBOL    LIST,NOFORMAT
COB      COBOL    LIST,NOFORMAT
```

```
     COBOL    COBOL    LIST,NOFORMAT
     COP      COBOL    LIST,NOFORMAT
     COPY     COBOL    LIST,NOFORMAT
     CPY      COBOL    LIST,NOFORMAT
     JCL      JCL      LIST,NOFORMAT
     JOB      JCL      LIST,NOFORMAT
     PLI      PL/I     LIST,NOFORMAT
     PL1      PL/I     LIST,NOFORMAT
```

In this way, customers may change the search order, the suffix name, and the Panvalet ADD options to meet their needs. You may have already compiled a custom Suffix Table from Mainframe Access Version 2. A migration tool is provided to convert this configuration back into a text file which may be presented to Mainframe Access Version 3.

```
//jobname  JOB (MFA),'PANVALET SUFFIX',CLASS=A,MSGCLASS=X
//***************************************************************
//*        EXTRACT PANVALET SUFFIX FROM MFA SERVER V2
//***************************************************************
//*
//PVSUFFX  EXEC PGM=MFAUTL01,REGION=1M
//STEPLIB   DD  DISP=SHR,DSN=hlq.MFA.V3.LOADLIB
//MFALIB    DD  DISP=SHR,DSN=hlq.MFA.V2.LOADLIB
//MFAPARM   DD  DISP=SHR,DSN=hlq.MFA.CNTL(PANVALET)
```

Panvalet Security Exit

MFA supports a customized user exit (MFAPVXIT) that enables you to implement member-level security within your Panvalet repositories. This user exit is required because security is not available via RACF, and a proprietary Panvalet exit (PSPILXIT) is only available under TSO dialogs.

MFAPVXIT is automatically invoked to approve access rights for all Panvalet activity: import, export, and directory requests. A sample exit and test harness is provided within the `.CNTL` file shipped with the MFA product libraries:

Member	Description
MFAPVXIT	Sample Panvalet member security exit
PVTESTX	Test harness for MFAPVXIT
ASMPVT	Assemble and link PVTESTX
ASMPVX	Assemble and link MFAPVXIT

MFA provides the user ID, member name, and Panvalet master file name to MFAPVXIT, and interprets a return code of zero to mean that access is permitted. Any other return code results in access being denied. This means that the MFAPVXIT can create a directory filter based on the access rights of the individual or his department. The algorithm is entirely up to the site, and will be based on table-driven data compiled into the exit. The sample exit demonstrates how naming conventions, or tables of authorized users can be used to accomplish this.

Librarian

Librarian is a powerful change management system for source modules and data records.

About Librarian

Under Librarian, all source is stored within proprietary Librarian master files in a highly compressed format. Individual source modules may be stored and retrieved by name. Multiple master files may be defined and shared among the user groups.

Librarian provides licensed users with programmable access to source modules via the File Access Interface Routines (FAIR) API. This is a read-only interface which permits multiple concurrent access for

Mainframe Access threads to browse source members residing in a Librarian master file. During Server initialization, Mainframe Access attempts to load this interface into memory. If this API interface is available through the standard search order (STEPLIB, JOBLIB, or SYS1.LINKLIB at the customer site) then Librarian services are offered. Startup initialization messages are indicated if Librarian is available. It is a customer responsibility to provide a suitable STEPLIB concatenation which enables selection of the desired release and version of the Librarian components.

```
        //STEPLIB DD DSN=hlq.MFA.LOADLIB,DISP=SHR
        //        DD DSN=CAI.LIBR43.CAILIB,DISP=SHR
```

For source additions or updates under Librarian, Mainframe Access invokes the standard Librarian batch utility AFOLIBR. Access to this utility is serialized, since Librarian does not support multiple, concurrent access to this service from the same address space. In some cases, you may have renamed the Batch Update utility AFOLIBR. If this is the case, the new name at this site must be defined to Mainframe Access as a parameter override.

```
        MFA_LIBRARIAN_HISTORY=”1”             - MFA default
        MFA_LIBRARIAN_UPD_MODULE=”LIBRUPD”    - site override
```

References

Librarian Command Reference Guide
Librarian File Access Interface Routines
Librarian Messages Guide
Librarian Installation Guide
Librarian User Guide
Librarian Systems Services Guide
Librarian Security Administration Guide

Librarian Exit Table

Librarian does support user exits under its Batch Update facility. The user exit may be common, or may be assigned to a specific master file. In order to configure this option, MFA references a reserved DD named LIBXIT$. If specified in the JCL, MFA builds a memory table to specify the Output Exit to be used for the given master file.

```
*  Librarian Exit Table (fixed format)
*
*  EXIT columns 1-8 is an exit module name
*       found in the STEPLIB, JOBLIB, LINKLIB concatenation
*  LIBRARIAN MASTER name resides in columns 11-54
*       any master file with this prefix will invoke the exit
*
* EXIT    Librarian Master file prefix or full name
********  ********************************************
LIBREXIT  PROD.LIB
LBXIT02   MY.LIBRARIAN.MASTER
```

Endevor

Endevor is a full function change control system with signout controls and extensive user customization.

About Endevor

Through the use of Exits and Processor Groups, it is possible to automatically re-compile the changed objects, and re-link any components which require those objects. Endevor also has several optional system features such as External Security, and alternate RACF user ID support.

Starting with Release 39, Endevor provides licensed users a powerful Endevor Services API (ENA$NDVR). This interface permits full-function access to read or update source objects residing in an Endevor repository. During Server initialization, Mainframe Access will attempt to load this interface into memory. If this API interface is available through the standard search order (STEPLIB, JOBLIB, or SYS1.LINKLIB at the customer site) then Endevor services will be offered. Startup messages will indicate if Endevor is available. It is a customer responsibility to provide a suitable STEPLIB concatenation for selection of the desired release and version of the Endevor components. Mainframe Access will examine the Endevor C1DEFLTS table to determine the version of Endevor being used. This is necessary because the control structures used by the Endevor API versions are not fully downward-compatible. Mainframe Access must build the API request using the format demanded by the version in effect at your site. If no C1DEFLTS table is found, Endevor support will not be enabled.

```
        //STEPLIB DD DSN=hlq.MFA.LOADLIB,DISP=SHR
        //        DD DSN=CAI.NDVRR16.CSIQAUTU,DISP=SHR
        //        DD DSN=CAI.NDVRR16.CSIQAUTH,DISP=SHR
        //CONLIB  DD DSN=CAI.NDVRR16.CSIQLOAD,DISP=SHR
```

Note that the Endevor API was designed to be a batch-oriented service. It is a file-based interface, not a record-based interface. That is, an object is moved from an Endevor-managed repository, to a specified work file; or vice-versa. Mainframe Access will dynamically create the necessary files as required. A transaction history log is likewise written to a log file. Mainframe Access may optionally be configured to retain a full Endevor transaction log to be used as an audit trail. Access to the Endevor interface is serialized since Endevor does not support concurrent accesses from the same address space. At the same time, all Endevor user exits and packages are operational. This means that site customization will affect the response time for an end-user accessing Endevor data under Mainframe Access. This may require adjustments to the timeout parameters on the clients.

Since all data movement is facilitated through temporary files, Mainframe Access is not able to support the Dsname Validation feature of Endevor. The temporary file used by Mainframe Access is deleted as soon as the transaction (Import or Export) ends. For those sites employing this feature, users could use the Endevor dialog tools under TSO to move their members into their own PDS. Then Mainframe Access could be used to move the PDS member from the mainframe to the workstation, and vice-versa. A subsequent promotion to Endevor would then indeed come from the exact same PDS that received the file at sign-out.

Mainframe Access does maintain the user ID security context for each active thread using the standard IBM security environment and standard SAF calls. Mainframe Access will always invoke the Endevor API interface using the security credentials of the end-user as provided during client logon. To minimize some administrative overhead, Micro Focus recommends the adoption of an Alternate RACF user ID to simplify the administration of RACF access rights for the user population. Access security remains a SITE responsibility. This is no different than standard BATCH or TSO access under Endevor. Mainframe Access will not schedule any Endevor access using the Mainframe Access started-task profile. Also be mindful that some change control "processors" might better be managed by specialist personnel using the Endevor ISPF panels rather than being triggered repeatedly by online Mainframe Access users.

Endevor support is available for R14 and later.

References

Endevor Administration Guide
Endevor API Guide
Endevor Error Codes and Messages
Endevor Exits Guide
Endevor Footprints Guide
Endevor Installation Guide
Endevor Packages Guide
Endevor Security Guide
Endevor User Guide

Endevor Dependent Regions

In order to isolate the Endevor run-time environment from the Mainframe Access address space, and improve the concurrency (and hence scalability) of Endevor accesses, Mainframe Access will schedule the execution of the Endevor API in an independent processing region referred to as an Endevor Dependent Region. This is a special purpose address space that Mainframe Access creates dynamically and shares with all Mainframe Access/Endevor users. It is serially assigned to whatever thread is in need of Endevor API services.

Multiple Endevor dependent regions may be started to meet your throughput requirements. The sample SERVERS member contains a definition for an Endevor address space group that allows a maximum of 2 address spaces to be started for Endevor processing. This value can be increased if the demands of your Endevor usage will benefit from greater concurrency.

You will need to update the started task JCL for dependent regions before you can use the Endevor support. Review the sample JCL member MFAAS and follow the instructions in that member to add your Endevor AUTHLIB and CONLIB program libraries to the STEPLIB concatenation. If you use Processor Groups to compile and relink automatically, specify DYNAMNBR=300 on the EXEC statement and eliminate the pre-allocation of SYSPRINT and SYSTERM which conflict with Endevor procedures to allocate compile and link output files.

For those sites where the logged in users may not have the authority to create transient files using their USERID as the high-level qualifier, there is a configuration parameter to set that high-level qualifier to some other value where all users have READ/WRITE/ALTER access.

```
ENDEVOR_DSNQUALIFIER="prefix"         --- OVERRIDE FOR ENDEVOR ACCESS
```

This must be specified within the configuration parameters specified by DD=XDBIN both in the MFA Control region, and the MFA Endevor Dependant Region.

In addition, the Endevor Dependent Region will automatically shutdown after a pre-defined idle period. The default is 30 minutes. This automatic shutdown feature allows Endevor resource allocation to quiesce with non-use, and thereby permits overnight backups to run without manual intervention. This idle period is defined as two scan intervals as specified in the Dependent Region parameter file:

```
DSS_SCAN_INTERVAL="15"                --- idle timeout for shutdown
```

By default, the Endevor Dependent Region will create a new VIO file (DDNAME=ENDVMSG) to be used as the Endevor transaction log for all subsequent transactions within that dependent region. The DCB attributes are:

```
PS,FBA,LRECL=133,UNIT=VIO,DISP=(NEW,DELETE),DSN=&MSGLOG
```

This file will effectively be a memory resident file, buffered by JES. Note that during initialization there is no end-user profile in effect. That means the VIO file will be owned by the started task. Therefore, the started task security profile must allow any end-user to write to this file. Endevor will OPEN the file as the transaction log. This behavior can be taken as the default, or specified by:

```
VIO_FOR_ENDEVOR_LOG=”1”         default VIO
```

If this causes a security violation at your site, you may configure the log to be disk-resident, provided the high-level qualifier will allow universal READ/WRITE access to the log. To override the default, use:

```
VIO_FOR_ENDEVOR_LOG=”0”          use DISK
```

Now Mainframe Access will supply overrides to disk as follows:

```
UNIT=SYSALLDA,DSNAME=prefix.jobname.ASnnnnn.MSGLOG
```

where the jobname and address space number will ensure uniqueness. The prefix will be taken from the ENDEVOR_ DSNQUALIFIER described above.

Endevor C1DEFLTS File

It may be necessary to modify the C1DEFLTS table to facilitate the Endevor support in a multi-user environment. If you do modify the C1DEFLTS table, the resulting C1DEFLTS load module should be stored in the STEPLIB of the MFA server start-up JCL to allow for a custom usage. Your Endevor administrator will have the current C1DEFLTS source and corresponding JCL.

The following C1DEFLTS table change is required:

- The BATCHID= parameter must be 1. This permits signout and security accesses to be based on the security profile of the user, not the MFA Server started task.

```
C1DEFLTS TYPE=MAIN,
         ACCSTBL=,       ACCESS SECURITY TABLE           X
         APRVFLG=N,      APPROVAL PROCESSING (Y/N)       X
         ASCM=0,         ASCM CONTROL PASSWORD           X
         BATCHID=1,      BATCH UID FROM JOBNAME/USER=    X
         CIPODSN=,       CCID VALIDATION DSNAME          X
          .
          .
          .
          .
C1DEFLTS TYPE=END
END
```

Endevor Setup Errors

Several environmental conditions are validated when the first transaction is processed on an Endevor Dependent Region. Failure to provide a consistent run-time environment will result in the following abends:

Abend	Description
996	MFAS startup error - no DSS environment found
997	Endevor DR setup error - no MSGLOG established
998	Endevor version mismatch between CR and DR
999	No Endevor support found in STEPLIB or JOBLIB

ChangeMan ZMF

Micro Focus's ChangeMan ZMF is a comprehensive mainframe specific solution that provides reliable, streamlined implementation of changes in z/OS environments. Its version, build and release management functions can be used to manage system changes from development to deployment.

About ChangeMan ZMF

ChangeMan ZMF 5.3 introduced the ChangeMan ZMF XML Services, an open interface for interoperability and data interchange with ChangeMan ZMF. This new interface is intended to replace the RPC and API interfaces that were used with earlier releases of ChangeMan. Mainframe Access Server implements the ChangeMan functions using the XML Services interface.

Mainframe Access requires ChangeMan Version 5.5.6 or later.

Customization

The customizations required in Mainframe Access Server are simple and deal primarily with defining the interface to the ChangeMan ZMF subsystem. There are four MFA Server startup parameters for defining this interface and they are described in detail below. The parameters are also documented in the section *Editing Mainframe Access Parameters* in the chapter *Configuration*.

The Server's ChangeMan interface uses Mainframe Access Application Server address spaces (also known as dependent regions) to provide isolation for the ChangeMan processing. Separating the processing from the MFA Server control region avoids potential interference with other client requests being processed in the control region.

Both the control region and the dependent region(s) have startup parameter requirements for the ChangeMan interface definition. The sample control region parameters are found in member PARMS and the sample dependent region parameters are found in member PARMSAS. Review the startup parameter information in the sections that follow and update your parameter definitions with values appropriate for your installation.

Control Region Startup Parameters

CHANGEMAN_COMMON_BUILD	Specifies whether or not Mainframe Access accommodates group builds submitted under AppMaster Builder. Specify 1 to enable group builds or 0 to disable them. The default is 0. When enabled for group builds, MFA modifies the build options for each member of the AMB group to match the component name and type submitted, ensuring that each build request in the group is properly executed.
CHANGEMAN_DSNQUALIFIER	Specifies a high-level qualifier for dynamically allocated data sets created by MFA Server to hold input files during the staging process. MFA Server creates these data sets dynamically as card-image sequential data sets that are deleted when the associated client request completes. The default qualifier is the login ID of the current user. You should set an alternate qualifier if your installation's SERNET does not have RACF authority to read or write data into files stored in data sets named with the current user's login ID as the high-level qualifier. If an alternate qualifier is required, see your ChangeMan administrator or your security administrator for assistance in selecting a proper value. If your ChangeMan functionality is restricted by security subsystem authorizations, you might need to specify your dataset high-level qualifier such that read access to the data sets is always permitted.
CHANGEMAN_INTERFACE	Specifies the name of the ChangeMan ZMF module. During ChangeMan request processing, MFA Server loads and branch enters this module. Valid values are `SERXMLBC` and `SERXMLAC`. The default is `SERXMLAC`. The preferred interface module is SERXMLAC, which is not documented in ChangeMan ZMF; however, it provides better interface performance than SERXMLBC by using virtual storage instead of data sets for XML input and output exchange between MFA Server (the requester) and ChangeMan ZMF. ChangeMan ZMF documents only the SERXMLBC interface, which is the XML services batch client. SERXMLBC is intended for use in a batch job stream and requires XMLIN and XMLOUT DD statements to define its required data sets. The MFAAS JCL sample provides sample DD statements for XMLIN and XMLOUT; however, they are commented out. If you must use SERXMLBC, contact SupportLine for assistance with data set definitions.

CHANGEMAN_SSID	Initializes the Mainframe Access ChangeMan interface by providing the final character of the z/OS subsystem ID used to identify ChangeMan. Omit this parameter to bypass Mainframe Access initialization for the ChangeMan interface. Valid values are any single alphabetic or numeric character. The character specified is appended to the subsystem ID string "SER", completing the four-character ID. For example, if you specify CHANGEMAN_SSID="A", the z/OS subsystem ID becomes "SERA". Your ChangeMan administrator can help you to determine the correct subsystem ID to use.
CHANGEMAN_TEST_OPTION	Specify whether or not MFA uses the XML trace option. Specify 1 to enable XML trace or 0 to disable it. The default is 0. When enabled, SERNET logs information in XML format into the standard SERPRINT DD file. This file can be used to identify problems and to validate MFA services. Enabling XML trace is not recommended for production systems due to the large volume of log data generated.
CHANGEMAN_XMS_SIZE	Specifies the size in megabytes of the cross-memory buffer used by MFA for transmitting XML requests and responses to the ChangeMan dependant Region. A single integer in the range of 1 through 8 is accepted. Note that about 2500 directory entries, or 10,000 lines of source code may be passed in 1 megabyte. If your requirements exceed these limits, then you can configure a more appropriate value. The default value with no configuration is 1 megabyte.

Control Region Started Task JCL

You can customize the sample JCL procedure for the MFA Server started task with required and optional DD statements for the ChangeMan interface.

CMNOPTS

Include and customize this DD to provide MFA with your site-dependent user options and your default build options. MFA reads the XML file you specify, using only the options defined for field names that correspond to build and user options. These options enable MFA to properly check out and stage ChangeMan components. See your ChangeMan ZMF documentation for ZDD Build Options for more information.

```
//CMNOPTS DD  DSN=youroptions.xml,DISP=SHR
```

Where *youroptions* represents a filename prefix of your choice.

The following is an example of an XML file containing user and build options:

```
<?xml version="1.0"?>
<options name="BUILD">
 <profile application="*" language="*" procname="*" libtype="*">
  <field name="Language"       editable="Y" default="CBL"/>
  <field name="BuildProc"      editable="Y" default="CMNCOB2/>
  <field name="Db2PreCompile"  editable="Y" default="N" />
  <field name="Db2Subsystem"   editable="Y" default="DSN7" />
  <field name="CompileOptions" editable="Y"
default="RENT,LIST,XREF(SHORT)" />
  <field name="LinkOptions"    editable="Y"
default="RENT,MAP,XREF" />
  <field name="UserOption01"   editable="Y" default="N"
     tag="Compile only"  required="Y"  validation="YN"   />
  <field name="UserOption02"   editable="Y" default="N"
     tag="IMS DLIT entry"  required="N"  validation="YN"   />
  <field name="UserOption03"   editable="Y" default="N"
     tag="CICS precompile"  required="N"  validation="YN"   />
  <field name="UserOption04"   editable="Y" default="N"
     tag="Drop INCLUDEs"  required="Y"  validation="YN"   />
  <field name="UserOption05"   editable="Y" default="N"
     tag="Easytrieve"  required="Y"  validation="YN"   />
  <field name="UserOption06"   editable="Y" default="N"
     tag="AMB Generation"  required="Y"  validation="YN"   />
 </profile>
</options>
```

The user build options tags <UserOption*nn*> support a special keyword for validation. If the valuation="*text*" corresponds to a reserved keyword as described by ChangeMan ZMF, then a generic test will be made for NUMERIC, ALPHABETIC, or ALPHANUMERIC text. Otherwise, the "*text*" characters themselves will be become the customized set of characters to be used as the validation string. Only a character found in this string will be allowed to act as the User Option value.

CMNLIB$

Optionally include and customize this DD to enable MFA to determine which component types are buildable, depending on the language type and build procedures specified. This not only identifies which Library types Drag & Drop may build, but also provides default language names for new components which have no component history which would otherwise provide this information.

```
//CMNLIB$ DD  DSN=hlq.MFA.CNTL(CMNLIB$),DISP=SHR
```

Where *hlq* is a high level qualifier that was given to all Mainframe Access data sets when the FRESTORE job was run.

The following is the default data:

```
*  ChangeMan Library Table (fixed format)
*          Like Source only
* L
* I BLDPROC  LANGUAGE
* B  NAME     Name
*** ******** ********
ASM CMNASM   ASM
COB CMNCOB2  COBOL
APS DGSAPCOM COBOL
ASC DGSAPSCR COBOL
```

CMNJOB$

Optionally include and customize this DD to define up to four job cards ChangeMan can submit for a build request, which is scheduled as an independent job by SERNET as part of the staging process.

```
//CMNJOB$ DD DSN=hlq.MFA.CNTL(CMNJOB$),DISP=SHR
```

Where *hlq* is a high level qualifier that was given to all Mainframe Access data sets when the FRESTORE job was run.

The default job card contains:

```
//useridB  JOB (MFA),'CMN-BUILD',CLASS=A,MSGCLASS=X
//*
//*
//*
```

Dependent Region Started Task JCL

The MFAAS sample JCL procedure contains JCL statements you must customize and enable for the ChangeMan interface. If you update sample JCL in the MFA Server samples data set, be sure to copy it to the system procedure library.

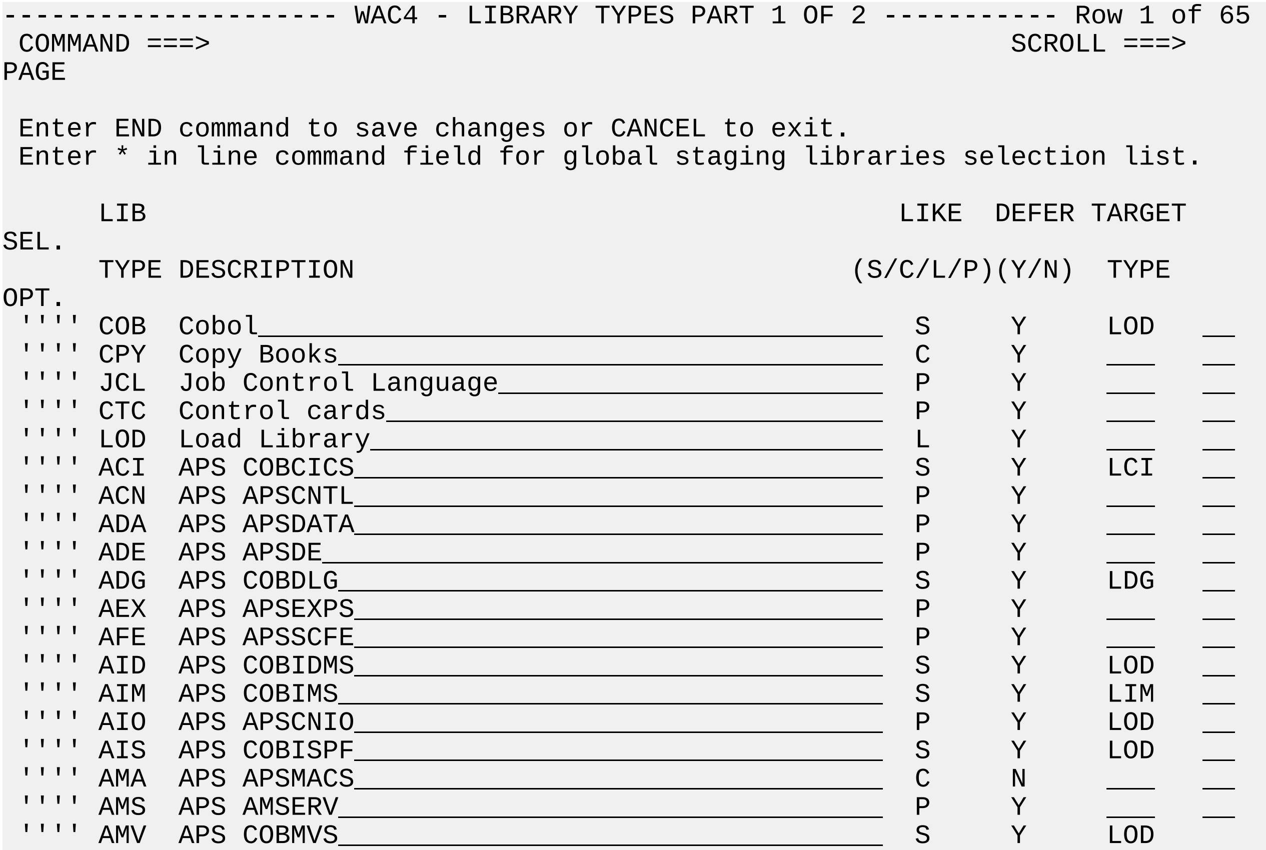
References

ChangeMan ZMF XML Services User Guide
ZMF Administrator Guide
ZMF Messages Guide

AppMaster Builder Support using APS Components

ChangeMan contains an optional feature for managing APS file types in a ChangeMan application. This support is incompatible with the AppMaster Builder support for APS components as provided by Micro Focus as it does not allow you to over-ride the Build procedures to invoke the AppMaster Builder Generation process.

As a result, when configuring ChangeMan for use with applications and packages used by AppMaster Builder, ChangeMan Administration should not configure the file types using the Selectable Option column, as shown in the following screen shot from the ChangeMan Administration Panels under TSO:



```
--------------------- WAC4 - LIBRARY TYPES PART 1 OF 2 ----------- Row 1 of 65
 COMMAND ===>                                                  SCROLL ===>
PAGE

 Enter END command to save changes or CANCEL to exit.
 Enter * in line command field for global staging libraries selection list.

      LIB                                               LIKE  DEFER TARGET
SEL.
      TYPE DESCRIPTION                               (S/C/L/P)(Y/N)  TYPE
OPT.
 '''' COB  Cobol_______________________________________  S     Y     LOD   __
 '''' CPY  Copy Books__________________________________  C     Y     ___   __
 '''' JCL  Job Control Language________________________  P     Y     ___   __
 '''' CTC  Control cards_______________________________  P     Y     ___   __
 '''' LOD  Load Library________________________________  L     Y     ___   __
 '''' ACI  APS COBCICS_________________________________  S     Y     LCI   __
 '''' ACN  APS APSCNTL_________________________________  P     Y     ___   __
 '''' ADA  APS APSDATA_________________________________  P     Y     ___   __
 '''' ADE  APS APSDE___________________________________  P     Y     ___   __
 '''' ADG  APS COBDLG__________________________________  S     Y     LDG   __
 '''' AEX  APS APSEXPS_________________________________  P     Y     ___   __
 '''' AFE  APS APSSCFE_________________________________  P     Y     ___   __
 '''' AID  APS COBIDMS_________________________________  S     Y     LOD   __
 '''' AIM  APS COBIMS__________________________________  S     Y     LIM   __
 '''' AIO  APS APSCNIO_________________________________  P     Y     LOD   __
 '''' AIS  APS COBISPF_________________________________  S     Y     LOD   __
 '''' AMA  APS APSMACS_________________________________  C     N     ___   __
 '''' AMS  APS AMSERV__________________________________  P     Y     ___   __
 '''' AMV  APS COBMVS__________________________________  S     Y     LOD
```

Configuring Access to Enterprise Server Mainframe Subsystem Support

Mainframe Subsystem Support (MSS) is an Enterprise Server feature that provides an execution environment for CICS transaction programs on Windows and UNIX platforms. These programs can be ones migrated from the z/OS platform, or entirely new applications, developed specifically for the Enterprise Server MSS environment.

Mainframe Access runs on z/OS and functions as middleware, providing a transparent bridge between your ES/MSS systems and your z/OS CICS systems and enabling a full range of peer ISC communications between these systems including transaction routing, function shipping, distributed program link and distributed transaction processing. The configuration elements in MFA Server create the relationship between MSS and CICS and enable MFA Server to convert between MSS's TCP/IP-based protocol and CICS's LU6.2-based ISC protocol.

This section presents a step-by-step approach to completing the Mainframe Access Server customizations for ES/MSS support. Use this section to work through the customizations.

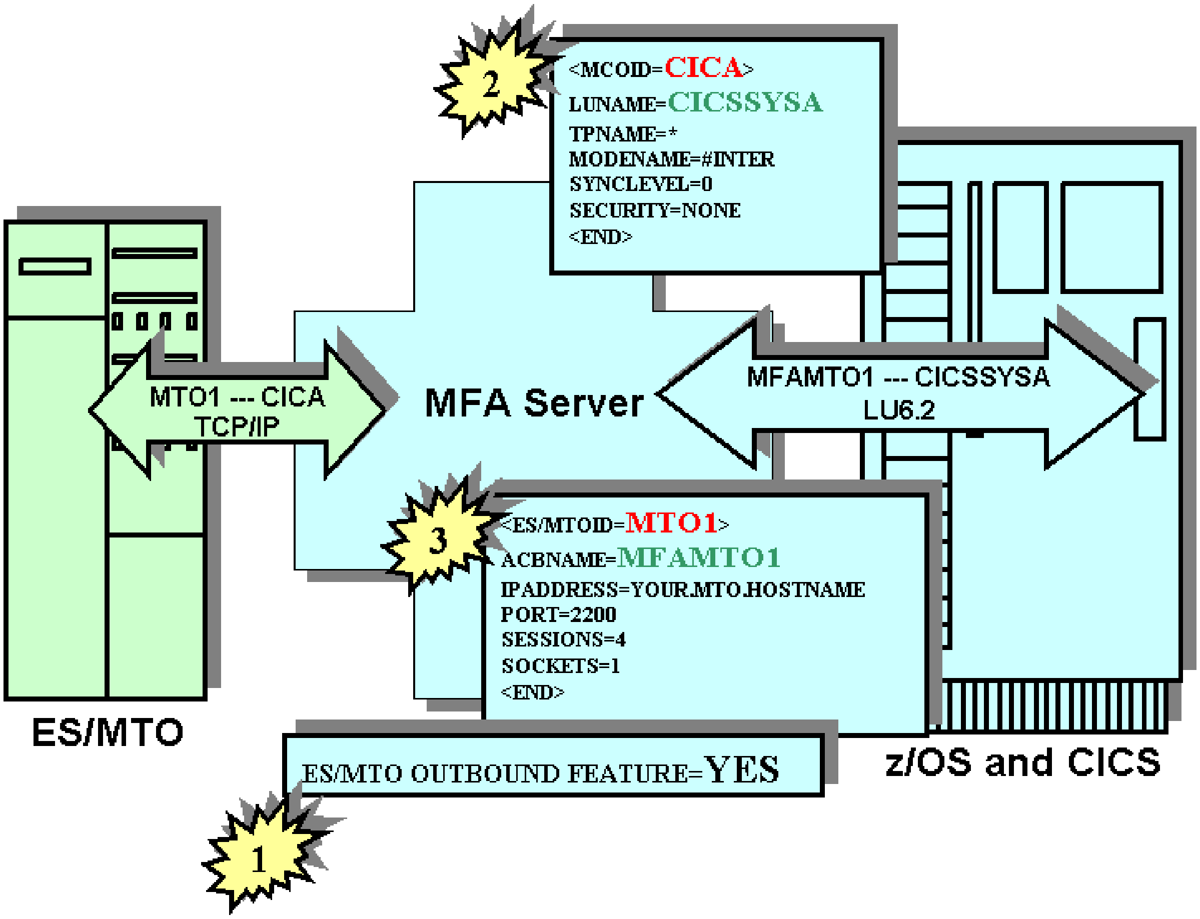
Prerequisites

Be sure to complete the basic configuration and installation testing of your Mainframe Access Server before you start the advanced configuration activities for ES/MSS support. See the section *Quick Configuration* in the *Configuration* chapter for information on the basic configuration. The configuration described in this appendix assumes that you have completed all of the quick configuration steps that establish your parameter file, server definitions, JCL procedures, etc. Also complete the installation verification testing to ensure that you have a running configuration.

MFA Server Configuration for ES/MSS

The diagram below shows a single MSS to CICS connection. The MFA Server configuration statements that enable this bi-directional connection are highlighted in the diagram. Keep in mind that there are corresponding configuration requirements in MSS and CICS, and the entire configuration relies on the communications infrastructure provided by your TCP/IP stack and VTAM SNA server.

ES/MSS runs on a Windows or UNIX platform and is connected to your z/OS system by TCP/IP communications. The other software elements reside on a z/OS platform(s). MFA Server runs on z/OS and functions as middleware, providing a transparent connection between ES/MSS and the CICS system. MFA Server is normally installed on the same z/OS system as CICS, although this is not a requirement. The connection between MFA Server and CICS uses standard CICS ISC LU6.2 protocols and MFA Server and CICS may reside in different systems on your SNA network. ES/MSS and MFA Server participate in the TCP/IP network by starting socket listeners, and by accepting and originating socket connection requests. CICS and MFA Server participate in the VTAM SNA network as Logical Units (LUs) by creating and opening a VTAM Access Method Control Block (ACB), and by allocating and managing LU6.2 conversations.



```
<MCOID=CICA>
LUNAME=CICSSYSA
TPNAME=*
MODENAME=#INTER
SYNCLEVEL=0
SECURITY=NONE
<END>
```

```
<ES/MTOID=MTO1>
ACBNAME=MFAMTO1
IPADDRESS=YOUR.MTO.HOSTNAME
PORT=2200
SESSIONS=4
SOCKETS=1
<END>
```

```
ES/MTO OUTBOUND FEATURE=YES
```

The MFA Server configuration is easy. There are three distinct parts to the configuration, numbered 1, 2, and 3 in the diagram. Part 1 is a simple enabling, or activation, statement. Part 2 defines your z/OS CICS system and provides the information needed for a VTAM SNA connection to this system. Part 3 defines your ES/MSS system and provides the information needed for a TCP/IP connection to this system. All of these statements can be found in the PARMS (Part 1) and SERVERS (Parts 2 and 3) members of your MFA Server <hlq>.CNTL data set. The definitions shown here for ES/MSS and CICS are two of the actual samples from that data set. When you understand the configuration requirements and have determined the correct values for your systems, complete your configuration using a text editor to update these sample definitions with your values. There is no pre-processor or configuration utility used with the MFA Server definitions and there is no configuration information maintained by MFA Server across restarts. So, each time MFA Server is started, your current definitions will be read in during start up, processed and activated. If you need to make changes during your testing, simply update the configuration statements and restart MFA Server.

The following subsections describe the customizations needed for your ES/MSS - CICS connection. The discussion refers to sample definitions that are provided by MFA Server installation and if this is your first ES/MSS configuration it will be easy to update the sample definitions as described.

Enable ES/MSS Support

MFA Server automatically activates the outbound feature and processes other definition statements for the MSS feature when it detects a valid (uncommented) MCO configuration in the MFAXML member.

Define the Target z/OS CICS System

Your MFAXML member in the MFA Server <hlq>.CNTL data set contains two sample CICS server definitions, one being the CICA sample shown here:

```
<MCO          ID="CICA" LUNAME="CICSSYSA" MODENAME="#INTER"
              TPNAME="*" SYNCLEVEL="0" SECURITY="0"
/>
```

ID="CICA"

Identifies the start of a z/OS CICS target server definition and provides the SYSID value for that server. Change this value to the SYSID of your z/OS CICS system. Specify the same SYSID when you define this CICS to your ES/MSS server in a connection definition.

LUNAME="CICSSYSA"

Provides the SNA LU name (also referred to as the VTAM ACB name or VTAM APPL name) of your CICS system. Change the value to the LU name of your CICS system.

TPNAME="*"

Identifies the transaction program name used in LU6.2 requests. The actual transaction program name used in each LU6.2 request depends on the function that is requested, and the value is supplied by the system that originates a request. MFA Server adapts depending on the value supplied. Leave this parameter as-is.

MODENAME="#INTER"

Provides the name of the VTAM SNA logmode used for LU6.2 sessions. If necessary, change the value to the name of the VTAM SNA logmode you want to use for your LU6.2 sessions. This same logmode name should be used in your CICS SESSIONS definition for the ES/MSS server.

SYNCLEVEL="0"

Sets the synchronization level for LU6.2 requests and is supplied by the system that originates a request. MFA adapts depending on the value supplied. Leave this parameter as-is.

SECURITY="0"

Sets the security level for LU6.2 requests and is supplied by the system that originates a request. MFA adapts depending on the value supplied. Leave this parameter as-is.

Define the Target ES/MSS Server

Your MFAXML member in the MFA Server <hlq>.CNTL data set contains two sample ES/MSS server definitions, one being the MSS1 sample shown here:

```
<ES-MTO       ID="MTO1" ACBNAME="MFAMTO1" IPADDRESS="YOUR.MTO.HOSTNAME"
              PORT="2200" SESSIONS="4" SOCKETS="1"
/>
```

ID="MTO1"

Identifies the start of an ES/MSS target server definition and provides the ES/MSS SYSID needed to identify the ES/MSS system to MFA Server. Change this value to the SYSID of your ES/MSS server. Specify the same SYSID for the CONNECTION(name) when you define this ES/MSS server to your z/OS CICS system.

ACBNAME="MFAMTO1"

Assigns a VTAM ACB name (also referred to as the SNA LU name or VTAM APPL name) to the ES/MSS server. You can define more than one ES/MSS server and each server you define must have its own ACBNAME, giving it a unique identity in the z/OS CICS system. MFA Server provides two VTAM APPL definitions for ES/MSS servers - MFAMTO1 and MFAMTO2. These are installed and activated in your VTAM SNA configuration during MFA

Server installation. If you are defining only one ES/MSS server, leave the ACBNAME as MFAMTO1. Use the MFAMTO2 ACB name if you define a second server. To define more than two ES/MSS servers, you must first install additional VTAM APPL definitions.

IPADDRESS="YOUR.MTO.HOSTNAME"

Identifies the internet address of your ES/MSS server machine. Change this value to either the machine name of the server where you're running ES/MSS or the internet IP address of the server.

PORT="2200"

Identifies the TCP/IP port on your ES/MSS server that has been configured as an ISC listener. Change this value to the ISC listener PORT number on your ES/MSS server.

SESSIONS="4"

Identifies the maximum number of transaction processing sessions that will be managed on a single socket connection. Each active LU6.2 conversation requires a session and the session traffic is interleaved on the MSS to MFA socket connection. For testing purposes, leave this value as-is. However, you might need to adjust it upwards to satisfy your installation's demand as usage increases.

SOCKETS="1"

Identifies the number of sockets available for each MSS to CICS connection. Because ES/MSS requires a single socket for each connection to CICS, leave this value as-is.

Restart MFA Server

When you complete the configuration steps, save your updated definitions and restart your MFA Server to activate the definitions. Be sure to complete the related configuration activites in ES/MSS and z/OS CICS before you begin testing.

Technical Details

The more detailed tecnical information in this section will increase your understanding of the configuration requirements. You will also be better prepared to respond to operational issues that may arise during the testing of this feature.

How ES/MSS Understands the Connection

The communication between ES/MSS and MFA Server uses a proprietary TCP/IP protocol that conveys the necessary LU6.2 protocol information in TCP/IP packets. The SYSIDs used in the MFA Server configuration are the key to this communication. In the example definitions, the target z/OS CICS system is identified as CICA and the target ES/MSS server is identified as MTO1. The initial connection messages between ES/MSS and MFA Server use these SYSID values to exactly identify the two target systems.

If ES/MSS initiates the socket connection to MFA Server, ES/MSS will send a "bind" packet for SYSID CICA. The bind packet also contains the SYSID of the originating MSS, MTO1. The initial packet from ES/MSS is identical to the initial packet from a CICS support client, so MFA Server needs to identify the type of requester. If MFA Server finds a definition for an ES/MSS server with a SYSID of MTO1 (the ID parameter in the MCO definition) and the request originated from the IP address specified in that definition, then MFA Server recognizes this as a request from an ES/MSS server. Otherwise, the connection is recognized as a request from a CICS support client.

At this point, it is important to note that support for CICS has been available in earlier versions of MFA Server and this support allows MFA Server clients to originate requests INTO a z/OS CICS system. This same level of support has also been used by ES/MSS systems, before the availability of the ES/MSS outbound feature. This level of support does not require a definition for the ES/MSS server and it does not require the ES/MSS outbound feature support. In this case, the only definition required is the definition of the target z/OS CICS system (number 2 in the diagram). If your only need is for ES/MSS requests into CICS, you may continue to configure your MFA Server in this way. However, our recommendation is that you follow the configuration requirements as outlined in this appendix and define a complete bi-directional connection for the systems that uses MFA Server's ES/MSS outbound feature. The extra effort is minimal and your systems will be ready to support CICS requests into ES/MSS, when they are needed. The complete configuration also results in each ES/MSS having a unique identity from the perspective of the z/OS CICS systems.

Once the initial connection is established, the ES/MSS and CICS definitions have been selected and MFA Server has all of the information it needs to accept a request from ES/MSS and send it into z/OS CICS over an LU6.2 ISC connection. MFA Server will allocate LU6.2 conversations using the MFAMTO1 ACB that has been defined for this ES/MSS. The conversation partner will be the z/OS CICS system that is using the ACB named CICSSYSA and the SNA session for the conversation will use the #INTER SNA logmode. The VTAM SNA logmodes are part of your VTAM SNA configuration and the suggested name can be changed to any other logmode name used for LU6.2 ISC connections in your system. Just be sure that the specified logmode definition is available to both MFA Server and CICS. The MFAMTO1 ACB name used in this sample definition is defined to the VTAM SNA network using a VTAM APPL definition. The APPL definition for MFAMTO1 is provided in another MFA Server sample, the MFAVTAM member in your <hlq>.CNTL data set. This member is installed in your VTAMLST definitions and activated during MFA Server installation.

Each ES/MSS server that you define must have its own ACBNAME and it is this ACBNAME that gives each server a unique identity in the z/OS CICS system(s).

How z/OS CICS Understands the Connection

The communication between z/OS CICS and MFA Server uses standard LU6.2 ISC protocol as implemented by CICS. The SNA logical unit names used in the MFA Server configuration are the key to this communication. In the example, the target z/OS CICS system is identified as CICSSYSA (LUNAME="CICSSYSA") and the target ES/MSS server is identified as MFAMTO1 (ACBNAME= "MFAMTO1").

Our target z/OS CICS initiates work to ES/MSS when a request is directed to an external CICS SYSID that is associated with the MFAMTO1 LU name by CICS definitions. A CICS CONNECTION defines the MFAMTO1 LU name and assigns an external SYSID. A CICS SESSIONS definition for that connection defines the SNA logmode to be used for the ISC conversations. It is important to note that the SYSID values known to z/OS CICS do not have to match the SYSID values known to MFA Server and ES/MSS, however, MFA Server and ES/MSS must agree in their definitions. Although the SYSID values known to CICS can be different from the ES/MSS and MFA Server definitions, we recommend that the same SYSID values be used throughout the definitions.

CICS handles the request to the external SYSID by establishing the LU6.2 sessions that support the ISC exchanges, if they are not already established. CICS then sends the initial FMH-5 Attach request for the conversation to MFAMTO1. MFA Server recognizes this request as a conversation for ES/MSS because CICS used the MFAMTO1 LU name. MFA Server also checks that the request originated in a defined CICS server by locating the definition for the originator's LU name (CICSSYSA) and logmode name (#INTER) combination. Having located the two sets of definitions, MFA Server can forward the request on to the target ES/MSS server. If there is currently no socket connection to ES/MSS, MFA Server has the IP address and port number needed to contact ES/MSS's ISC listener. Now the request can be forwarded to ES/MSS over the socket connection associated with the MTO1-CICA SYSID combination.

Configuring z/Server feature support (deprecated)

Note: These features are deprecated, and provided for backward compatibility only.

Introduction

z/Server V4R0 consists of mainframe components which have to be installed in a z/OS environment.

This document is intended for system programmers installing and configuring the z/Server software on their z/OS host system.

z/Server overview

z/Server supports the execution of mainframe programs called from a client via TCP/IP requests.

z/Server provides:

- A multi-tasking scheduling server (called scheduler)
- An initialized LE environment
- Automatic code page translation
- Task level security
- XML support
- User servers running an ISPF environment
- Command support
- MVS catalog and data set access
- JESx access

The z/Server configuration file

XML configuration file

The configuration for z/Server is now part of the Mainframe Access Server XML configuration file and is used to store all of the configuration and properties that control how Mainframe Access Server and the z/Server feature operates. It's these parameters that you need to review before you can get your z/Server feature setup.

In order to take advantage of the new single Mainframe Access XML configuration file you must have taken all the actions in the *Migration to use the XML configuration* file, be using the unified scheduler and using MFA Server to start the holder task internally (equivalent to having set ZSERVER FEATURE=HOLDER in the old MFA Server configuration file). These migration actions can be found in the Micro Focus Enterprise Developer 3.0 product Help.

See *Migrating old configurations to new XML configuration* for more information on how to migrate your existing configuration files.

Legacy configuration files

Legacy configuration files are still supported. If you want to continue to use them, full details of their use are contained in the version of the *z/Server Installation Guide* that accompanied Enterprise Developer for z Systems 3.0, available from the Micro Focus Product Documentation web pages.

Program materials

In hlq.ZSERVER you will find the z/Server software components (machine-readable material) and the following samples in hlq.ZSERVER.SAMPLIB:

TAUSACOB Demonstrating an ISPF tool written in COBOL. The tool returns the parameters passed to it using the ISPF Shared Pool variable TAUTOPM.

TAUSAPL1 Demonstrating an ISPF tool written in Open PL/I. The tool returns the parameters passed to it using the ISPF Shared Pool variable TAUTOPM.

TAUSACLA Demonstrating an ISPF tool written in CLIST that adds two input values and returns their sum.

TAUSACLI Demonstrating an ISPF tool written in CLIST that returns the input value that is entered.

Configuration overview

To configure the z/Server feature with a minimum of customization, use the following guidelines:

1. Define the necessary started tasks:
    - A z/Server scheduler address space
    - User server address spaces (batch TSO address space)
2. Optional: Define a logon procedure for user servers as CEA-launched TSO address spaces.
3. Before z/Server address spaces can be started, you must customize a minimum configuration that consists of:
    - The MFA Server holder task configuration:
        - Specify the SVC number (default 238)
        - Specify the data space token (default TAURSERV)
        - Specify the name of the IP stack used for processing (default TCPIP)
    - The scheduler configuration:
        - Specify the general listener port (default 1200)
        - Specify the user server port range (default 1201-1249)
        - Specify the name of the user server started tasks (default IVPUSRT)
    - Optional: For CEA scheduler support:
        - Specify the name of the logon procedure (default CEAPROC)
        - Specify the name of the initial command REXX procedure (default ZCEAICMD)
4. Issue a command to MFA to start the z/Server holder, which in turn will start the scheduler address space(s).
5. Run the installation verification procedure *hlq*.ZSERVER.JCL(IVPVERI) for scheduler.
6. Run the TAU REXX procedure to verify user server startup. This is limited to support of port numbers up to a length of four characters.
7. Edit the master model configuration file *hlq*.ZSERVER.MASTER:
    - Enter a unique system name after "System:", for example, LPAR1.
    - Customize the dataset names pointing to the models after "Conf:".
8. Edit the REXX procedure *hlq*.ZSERVER.EXEC(TAUZCAPP) and specify the correct name of the master model configuration file *hlq*.ZSERVER.MASTER after "sysdsn=".

Preparing for configuration

Before You Begin

The following table lists important configuration data that you need to know in order to complete the quick configuration. Review these items and determine the appropriate values before you proceed with customization:

ITEM	DESCRIPTION
hlq	High level qualifier that was given to all Mainframe Access data sets when the FRESTORE job was run.
taurispf	The JCL procedure and jobname of the z/Server STC scheduler started task. The default value is TAURISPF.
ivpusrt	The JCL procedure name of the z/Server STC user server started task.
svc_no	The type-3 supervisor call routine number reserved for TAURAUTH. The default value is 238.

ITEM	DESCRIPTION
listener_port	The TCP port reserved for the z/Server STC scheduler. The default value is 1200.
first_port	The first TCP port that defines the range of ports available for the STC user servers. The default value is 1201.
last_port	The last TCP port that defines the range or ports available for the STC user servers. The default value is 1249.

Optionally, for CEA:

ITEM	DESCRIPTION
ceaproc	The JCL procedure of the CEA user server.
cea_account	The accounting number used by all TSO users who use the CEA scheduler.

SVC Routine (deprecated)

Note: SVC routine is deprecated, and supported for backward compatibility only.

MFA z/Server feature supports requires a type-3 SVC routine for all authorized commands. You need to call this routine TAURAUTH and reserve a number for it. 238 is the default specified in the configuration files that come with z/Server.

Address spaces

There are different types of address spaces that come with z/Server:

- One or more scheduler address spaces (jobname taurispf).
- One or more STC user server address spaces per TSO user/ Eclipse client (STC).

If you want to use the CEA-launched address space you need the following address spaces:

- One or more CEA user server address spaces per TSO user/Eclipse client (TSO address space).

They need an associated user ID:

- Scheduler use the assigned user ID.
- STC user servers start out with the assigned user ID, which is then switched to the user ID of the client that requested execution of some application via TCPIP and started them.

If you want to use the CEA-launched address space you will also need the following:

- CEA user servers are started with the TSO user ID of the client.

The scheduler address space defines a LISTENER_PORT that listens for incoming requests from the client, and a further port range (defined by FIRST_PORT through LAST_PORT) that will be assigned to different user servers as required. The combination of ports specified by the scheduler (both LISTENER_PORT and the range specified by FIRST_PORT through LAST_PORT) must be opened in the firewall protecting the z/OS host system. These ports should not be assigned to the scheduler. Instead, if desired, an appropriate RACF profile in the SERVAUTH needs to be defined. Please refer to IBM's TCPIP documentation. The default port range specified by the configuration file that comes with z/Server is 1200 to 1249.

User servers use the port assigned to them by the scheduler, to listen for incoming requests.

Dataset renaming

To be able to use the rename data set functionality in the AWM Eclipse client you must have configured MFA Server to have an AMS/IDCAMS Application Server. See *MFAAMS Started Task JCL Procedure* and *Application Server Parameters for AMS/IDCAMS support* for more information .

Steps to activate z/Server

A running z/Server consists of at least two started tasks:

- z/Server holder address space (TAURHLD)
- z/Server STC scheduler address space (TAURISPF)
- Zero or more z/Server STC user server address spaces (wtsouid, with "w" a configurable prefix and tsouid the TSO user ID of the client). See *TSOE_JOBCHAR* for more information on this configuration. They are started as IVPUSRT by the scheduler address space
- Optional: z/Server CEA scheduler address space for CEA-launched TSO user address spaces (TAURCEA)
- Optional: Zero or more CEA-launched TSO user address spaces (TSO user id)

The holder address space TAURHLD provides the necessary infrastructure and user administration control structures using a common access data space. It is non-swappable. It is also used to start and stop the scheduler and the user server address spaces.

The scheduler address space provides the requested client services or starts other z/Server user server address spaces to run a user specific ISPF application. There can be more than one scheduler running under the umbrella of the same holder address space.

A user server address space is started first to download the master configuration file to the Eclipse client and whenever the Eclipse client requests a specific REXX exec to be run. Each client can have more than one user server running for that TSO user ID.

A z/Server scheduler address space for CEA-launched TSO user address spaces (CEA scheduler in short) provides the services to start a TSO user address space. It is started when the Eclipse client calls the action “Launch ISPF” from zExplorer’s context menu in Remote Systems View, provided the port specified in the MVS Explorer settings under “ISPF launcher port” maps the parameter PORT specified in the CEA scheduler (CEASCHED) configuration.

The CEA-launched TSO user address space runs like any TSO address space under the user id of the connected user. The client communicates with this address space using the CEA scheduler and a z/OS USS message queue.

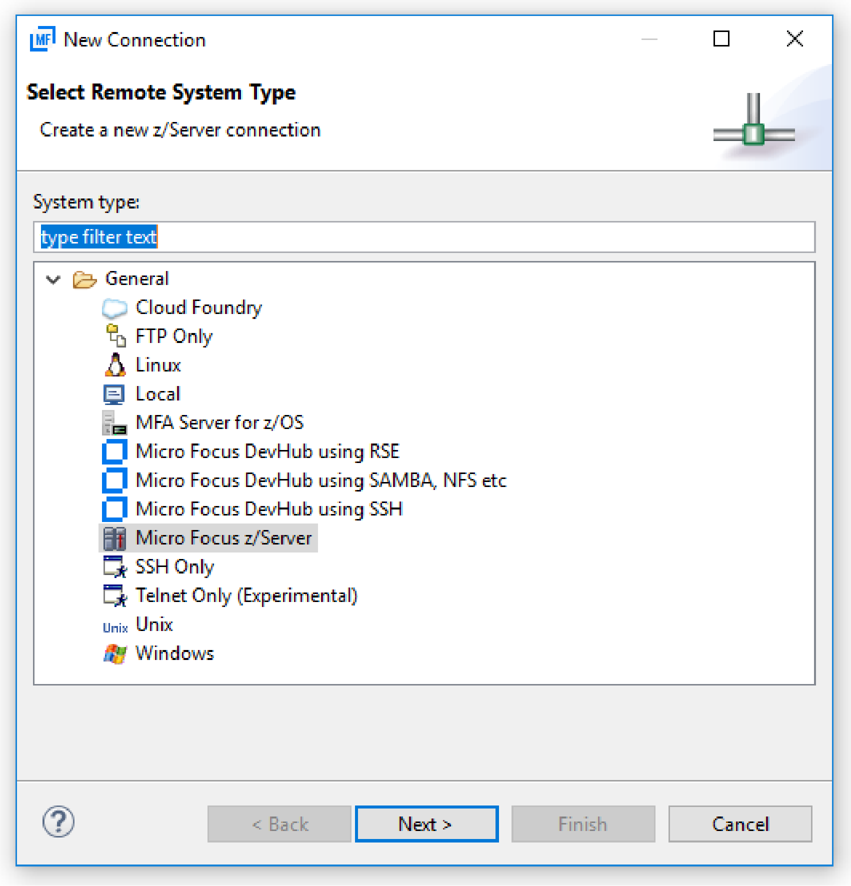
The holder can be started under MFA by specifying the ZSERVER FEATURE mainframe access parameter. Starting the holder under MFA results in improved performance. See *Editing Mainframe Access Parameters* for more information.

Defining a client connection to z/Server

In order to test the connection to z/Server from an EDz client you must define a Micro Focus z/Server connection in the Remote Systems view:

1. Click **Define a connection to remote system**.

   This opens the **New Connection** dialog box.



New Connection

Select Remote System Type

Create a new z/Server connection

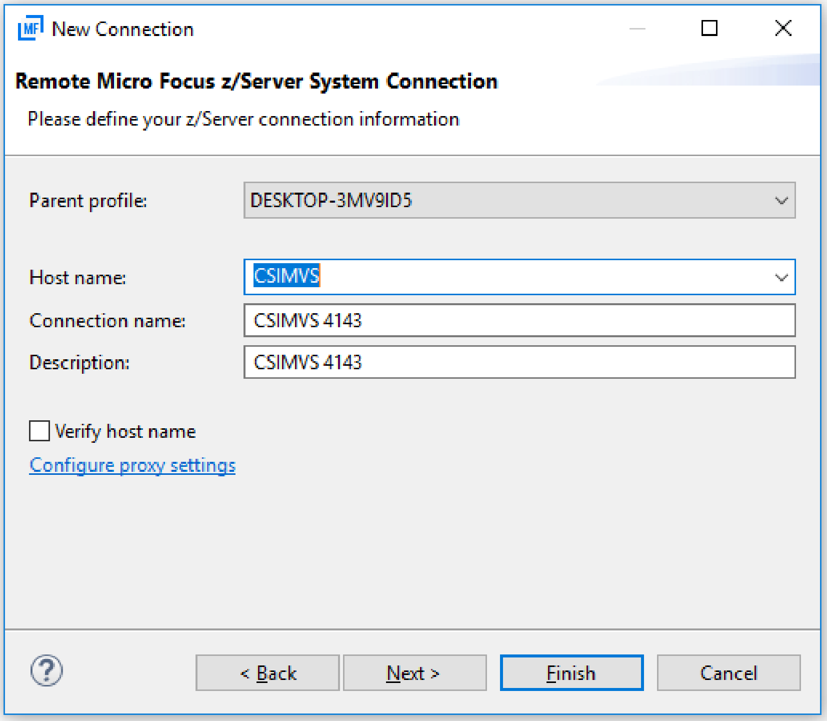
System type:

type filter text

General
Cloud Foundry
FTP Only
Linux
Local
MFA Server for z/OS
Micro Focus DevHub using RSE
Micro Focus DevHub using SAMBA, NFS etc
Micro Focus DevHub using SSH
Micro Focus z/Server
SSH Only
Telnet Only (Experimental)
Unix
Windows
< Back
Next >
Finish
Cancel

2. Click **Micro Focus z/Server**.
3. Click **Next**.

   You need to specify the z/Server connection information:



New Connection

Remote Micro Focus z/Server System Connection

Please define your z/Server connection information

Parent profile: DESKTOP-3MV9ID5

Host name: CSIMVS

Connection name: CSIMVS 4143

Description: CSIMVS 4143

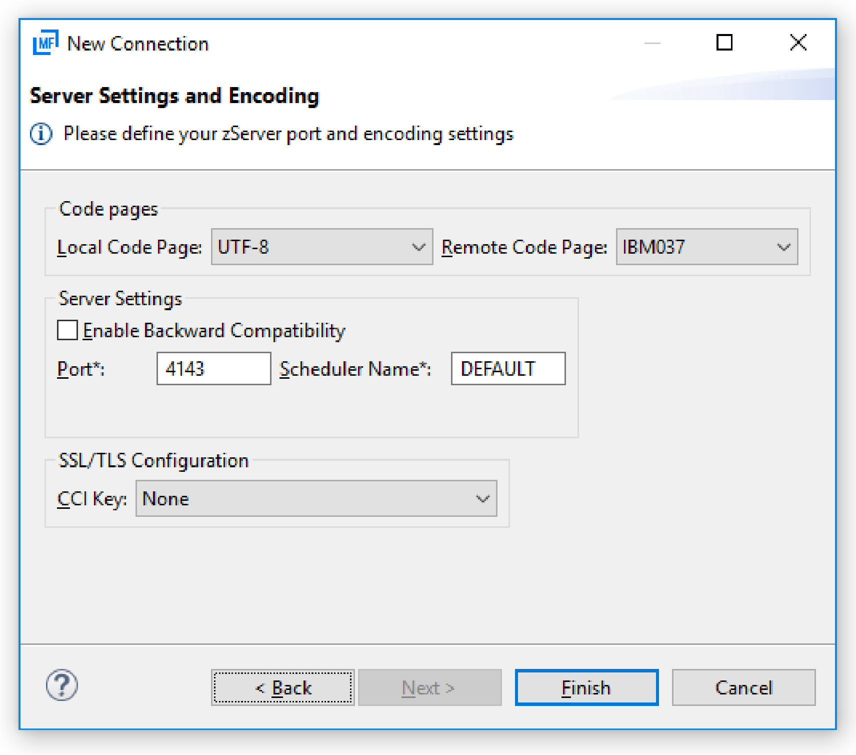
□ Verify host name

Configure proxy settings

4. In the **Host name** field, type the logical name or IP address of the z/OS system you want to connect to.
5. In the **Connection name** field, type the connection name that will be listed in the Remote Systems view.

   Optionally, in the **Description** field, type a description for the connection.
6. Click **Next**.

   This takes you to the Server Settings and Encoding step.



Local Code Page: UTF-8
Remote Code Page: IBM037
□ Enable Backward Compatibility
Port*: 4143
Scheduler Name*: DEFAULT
CCI Key: None

7. In the **Local Code Page** list, select the code page which you want to use on the client side.

   Note: Currently, only UTF-8 is supported as the local code page.

8. In the **Remote Code Page** list, select the code page which you want to use on the server side to represent a source file.

   A source file downloaded from the mainframe is translated using the code page specified by **Remote Code Page** and **Local Code Page**.

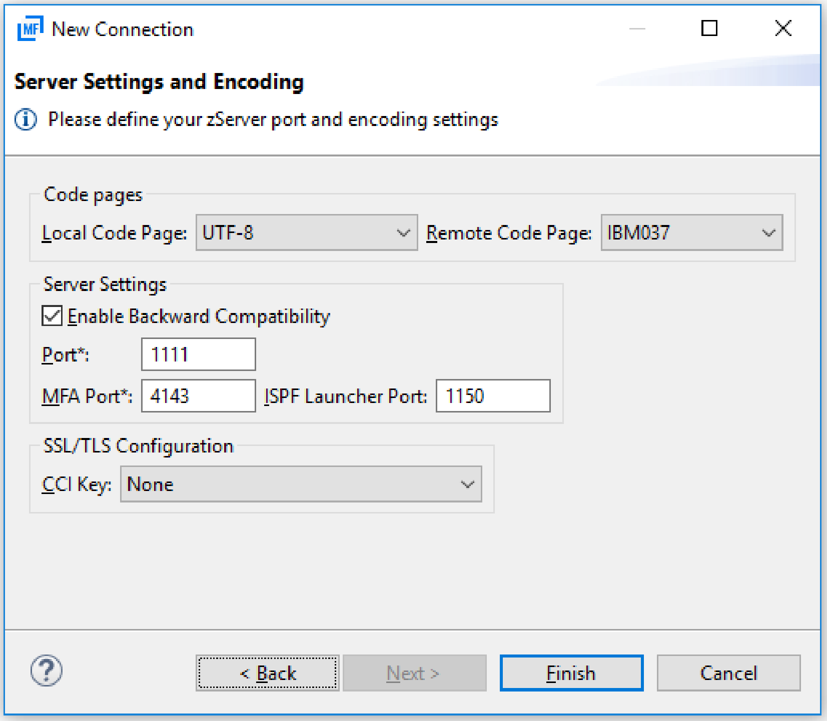
9. In the **Port** field, type the port of the MFA server to which the z/Server installation is mapped by default.

   Note: Do not specify the port on which the z/Server in running.

10. In the **Scheduler Name** field, leave the value as DEFAULT if you want to use the z/Server that your MFA server is mapped to by default.

    You only need to change this field if you want to use z/Server instance other than the one specified in the default mapping of your MFA server. To use another z/Server instance, type the name of the z/Server STC scheduler.

11. Only check the **Enable Backward Compatibility** check box if you want to connect to a z/Server prior to 2.3 Update 1.



New Connection

Server Settings and Encoding

Please define your zServer port and encoding settings

Code pages

Local Code Page: UTF-8

Remote Code Page: IBM037

Server Settings

☑ Enable Backward Compatibility

Port*: 1111

MFA Port*: 4143

ISPF Launcher Port: 1150

SSL/TLS Configuration

CCI Key: None

After checking **Enable Backward Compatibility**:

a. In the **Port** field, type the port number which the z/Server is running on.
b. In the **MFA Port** field, type the port number which the MFA server is running on.
c. In the **ISPF Launcher Port** filed, type the port number which the CEA server scheduler of the z/Server is running on.

12. The **CCI Key** list is only relevant if the z/Server is configured to use SSL/TLS encryption. In this case, the **CCI Key** list contains all the SSL/TLS configurations defined in your local `CCI.ini` file for the port specified by the **Port** field. Select the SSL/TLS connection from the **CCI Key** list that you want to work with.
13. Optionally, if you want input data sets to be displayed for jobs under the JES Explorer for that connection, you must check the option **Show input data sets**. Right-click the connection and click **Properties** in the context menu. Select **Subsystem** on the left of the dialog and check the option **Show input data sets** on the right of the dialog.
14. Click **Apply and Close**.

Define Started Tasks to z/OS

Define started tasks to RACF and ACF2

Each of the started tasks needs to be defined to the security product. We recommend that you run all z/Server STCs using the same ID. Work done for the client is run using the TSO user ID credentials of the client (task level security). All user IDs to be used with z/Server need a valid OMVS segment (required by TCP/IP).

READ access to the z/Server datasets is needed.

Depending on your installation, there may be more authorization needed (see *User authorizations*).

Assuming that RACF is the security product, the definitions could look like this:

```
AU usrname DATA('z/SERVER Userid') NOPASSWORD DFLTGRP(grpname) OWNER(grpname)
OMVS(AUTOUID HOME('/u/usrname'))
```

```
ALU usrname NOPASSPHRASE
RDEF STARTED TAURHLD.* STDATA(USER(usrname)) OWNER(grpname)
RDEF STARTED TAURISPF.* STDATA(USER(usrname)) OWNER(grpname)
RDEF STARTED IVPUSRT.* STDATA(USER(usrname)) OWNER(grpname)
SETR RACLIST(STARTED) REFRESH
ADDSD 'hlq.ZSERVER.**' OWNER(grpname) UACC(NONE)
PE 'hlq.ZSERVER.**' ACCESS(READ) CLASS(DATASET) ID(usrname)
```

The RACF definitions for a CEA scheduler could look like this:

```
RDEFINE TSOPROC CEAPROC OWNER(SYS1) UACC(NONE)
PE CEAPROC CLASS(TSOPROC) ACCESS(READ) ID(all-required-groups)
SETROPTS RACLIST(TSOPROC) REFRESH
```

If ACF2 is the security product, to allow READ access to the z/Server datasets, the equivalent definitions for class STARTED could look like this:

```
SET CONTROL(GSO)
INSERT STC.TAURHLD  LOGONID(usrname)
INSERT STC.TAURISPF LOGONID(usrname)
INSERT STC.IVPUSRT  LOGONID(usrname)  STCID(********)
F ACF2,REFRESH(ALL)
```

The holder address space administers the port range to be used for the scheduler and the user servers, but does not use them. The default port range specified in the configuration files that come with z/Server is 1100 to 1200. This port range needs to be opened in the firewall protecting the z/OS host system.

The scheduler address space listens for incoming requests from clients on the port designated as PORT in the scheduler configuration file. PORT=1111 is set as default. The scheduler assigns ports to different user servers from the port range defined for that scheduler (TSOE_FIRST_PORT .. TSOE_LAST_PORT).

Since the holder address space starts and stops the scheduler and stops user servers, and the scheduler address space starts and stops user servers, the associated user ID needs the appropriate access rights in the OPERCMDS class to these system commands.

APF authorizations

The data set hlq.ZSERVER.AUTHLIB must be APF authorized in the PARMLIB concatenation, for example, SYS1.PARMLIB(PROGxx).

Note: APF authorizations can be set dynamically with the following console command:

```
SETPROG APF,ADD,DSN=hlq.ZSERVER.AUTHLIB,SMS
```

or, if it is not an SMS managed data set:

```
SETPROG APF,ADD,DSN= hlq.ZSERVER.AUTHLIB,VOL=volser
```

Define started tasks to WLM

As a server, the importance of the MFA Server address space should be set below TCPIP but above the scheduler address space TAURISPF. This setting comes into play when MFA Server starts up and terminates or when commands are executed against the MFA Server address space.

The scheduler and the user server address spaces should have the same importance, with the user servers classified like any TSO user address space using the response time goal of TSO. All transactions executing within scheduler and user server are TSO transactions.

Keep in mind that these address spaces are STCs, so the classification must be done under the STC subsystem.

Customizing the MFA Server address space JCL (MFA)

The MFA Server address space MFA is used to install the SVC routine and to set up a common access data space that contains user administration control structures needed for communication.

The type 3 SVC routine is installed whenever the address space starts, and is deleted from the system SVC table when it is stopped.

Customize the sample JCL procedure that is used to start MFA Server. This is located in *hlq*.MFA.CNTL(MFA). You must uncomment the following four lines highlighted below:

```
//*-------------------------------
//* zServer Holder
//*-------------------------------
//*
//*SYSEXEC  DD DISP=SHR,DSN=&TAUQUAL..EXEC
//CONFXML  DD DISP=SHR,DSN=&DSNQUAL..CNTL(MFAXML)
//CONFOSR  DD DISP=SHR,DSN=&TAUQUAL..OSR(MFAOSR)
//*DSPPRT   DD SYSOUT=*,LRECL=255
//*SYSOUT   DD SYSOUT=*
//*SYSTSIN  DD DUMMY
```

Customizing an STC scheduler address space JCL (TAURISPF)

The z/Server scheduler task TAURISPF is responsible for the communication with a client and listens for incoming client requests on the port designated as LISTENER_PORT in the scheduler configuration file. The scheduler also defines a range of ports that are used, as needed, when a user server is started, and routes some of the incoming work to the appropriate user server address space. It is essentially a batch TSO address space. Configuring these ports is covered in *Configuring z/Server*.

Customize the sample hlq.ZSERVER.JCL(IVPISPFJ) and copy it to the PROCLIB concatenation and named as taurispf.

Verify and customize the high level qualifier defined for TAURHLQ, and also TCPDATA to make sure it specifies the correct TCPIP configuration data set:

```
//TAURISPF PROC
//TAURHLQ  SET  TAURHLQ=HLQ.ZSERVER
//TCPDATA  SET  TCPDATA=TCPIP.TCPDATA
//IVPISPF  EXEC PGM=IKJEFT01,PARM='TAURIP',
//       DYNAMNBR=200,REGION=0M,TIME=NOLIMIT
//STEPLIB  DD  DISP=SHR,DSN=&TAURHLQ..LOADLIB
//ZSRVAUTH DD  DISP=SHR,DSN=&TAURHLQ..AUTHLIB
//*SYSABEND DD  SYSOUT=*,DSN=&&SYSUDUMP
//SYSUDUMP DD  SYSOUT=*,DSN=&&SYSUDUMP
//SYSPRINT DD  SYSOUT=*,DSN=&&SYSPRINT
//SYSTSPRT DD  SYSOUT=*,DSN=&&SYSTSPRT
//ZCOTSPRT DD  SYSOUT=*,DSN=&&SYSTSPRT
//SYSTSIN  DD  DUMMY
//ZCOTSIN  DD  DUMMY
//*IPCONFIG DD  DISP=SHR,DSN=&TAURHLQ..CONFIG(IVPISPFJ)
//READER   DD  SYSOUT=(*,INTRDR)
//SYSEXEC  DD  DISP=SHR,DSN=&TAURHLQ..REXX
//SYSTCPD  DD  DISP=SHR,DSN=&TCPDATA
//*MAILHDR  DD  DISP=SHR,DSN=&TAURHLQ..SAMPLIB(IVPMHDR)
```

Note: Notice that the IPCONFIG DD statement is commented out. This must not be enabled if you are using the new XML configuration file.

The SYSTCPD DD statement has been added to provide information about the IP stack on your host to the started task.

DD statement MAILHDR names the sample dataset from which to copy the pattern for email notification addressees if email notification is configured. Email notification is intended to send an email to predefined recipients in case of an error situation. It requires that an SMTP server is available which forwards the generated emails.

Customizing a user server address space JCL (IVPUSRT)

A z/Server STC user server IVPUSRT is started by the scheduler address space to execute REXX execs or ISPF applications. It is essentially a two-step batch TSO address space. The first step executes under the general z/Server user ID and switches the security environment to the user ID of the client that had requested the user server to be started. The second step (the actual batch TSO address space) then uses this specific TSO user ID.

Customize the sample hlq.ZSERVER.JCL(IVPUSRT) and copy it to the PROCLIB concatenation and named as ivpusrt .

Verify and customize the high level qualifier defined for both TAURHLQ and ISPFHLQ, and also TCPDATA to make sure it specifies the correct TCPIP configuration data set. Make sure that the ISPF high level qualifier is ISF. Similar to a logon procedure, add all required libraries for things such as panels, messages, and REXX execs that are necessary to run the user’s required ISPF applications.

The initial REXX procedure IVPINIT1 must be located in the z/Server REXX dataset. This REXX exec should be customized by the installation to fit the installation's needs (see *Optional customization*).

Note: The dataset hlq.ZSERVER.REXX must be allocated using DDNAME SYSEXEC (not SYSPROC).

```
//TAURHLQ  SET TAURHLQ=HLQ.ZSERVER
//ISPFHLQ  SET ISPFHLQ=ISP
//TCPDATA  SET TCPDATA=TCPIP.TCPDATA
//*********************************************************************
//*    START TSO/E-ISPF-SESSION AS STARTED JOB
//*********************************************************************
//*  SET ACEE FOR USER
//*********************************************************************
//TAURSJOB EXEC PGM=TAURSJOB
//STEPLIB  DD DISP=SHR,DSN=&TAURHLQ..LOADLIB
//ZSRVAUTH DD DISP=SHR,DSN=&TAURHLQ..AUTHLIB
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//*IPCONFIG DD DISP=SHR,DSN=&TAURHLQ..CONFIG(IVPACEE)
//*********************************************************************
//*    START TSO/E-ISPF-SESSION BATCH
//*    MOVE TEMP ALLOCATIONS TO IVPINIT1 RPI 611619
//*********************************************************************
//IVPISPF  EXEC PGM=IKJEFT1B,PARM='%IVPINIT1',
//            DYNAMNBR=200,REGION=0M,TIME=NOLIMIT,COND=(4,LT)
//STEPLIB  DD DSN=&TAURHLQ..LOADLIB,DISP=SHR
//ZSRVAUTH DD DISP=SHR,DSN=&TAURHLQ..AUTHLIB
//*IPCONFIG DD DISP=SHR,DSN=&TAURHLQ..CONFIG(IVPUSR)
//SYSTCPD  DD DISP=SHR,DSN=&TCPDATA
//SYSEXEC  DD DSN=&TAURHLQ..EXEC,DISP=SHR
//         DD DSN=&TAURHLQ..REXX,DISP=SHR
//ISPPLIB  DD DSN=&TAURHLQ..PANELS,DISP=SHR
//         DD DSN=&ISPFHLQ..SISPPENU,DISP=SHR
//         DD DSN=ISF.SISFPLIB,DISP=SHR
//ISPSLIB  DD DSN=&TAURHLQ..SKELS,DISP=SHR
//         DD DSN=&ISPFHLQ..SISPSENU,DISP=SHR
//         DD DSN=ISF.SISFSLIB,DISP=SHR
//ISPMLIB  DD DSN=&ISPFHLQ..SISPMENU,DISP=SHR
//         DD DSN=ISF.SISFMLIB,DISP=SHR
//ISPTLIB  DD DSN=&ISPFHLQ..SISPTENU,DISP=SHR
//         DD DSN=ISF.SISFTLIB,DISP=SHR
//ISPLOG   DD SYSOUT=*,DCB=(RECFM=VA,LRECL=125)
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//ZCOTSPRT DD SYSOUT=*
//SYSTSIN  DD DUMMY
//ZCOTSIN  DD DUMMY
//SYSOUT   DD SYSOUT=*
//CEEAPI03 DD SYSOUT=*
//CEEREX30 DD SYSOUT=*
```

```
//ISPDPTRC DD SYSOUT=*
//         PEND
```

Temporary data set references have been removed from the above JCL and can now be found in the IVPINIT1 REXX procedure.

Note: Notice that the IPCONFIG DD statement is commented out. This must not be enabled if you are using the new XML configuration file.

The SYSTCPD DD statement has been added to provide information about the IP stack on your host to the started task.

Optional: Customizing a CEA-launched TSO user address space JCL (CEAPROC)

A z/Server user server started as a CEA-launched TSO user address space CEAPROC is started by the z/Server scheduler address space. It behaves like a foreground TSO address space running with the TSO user ID of the client user who requested the start of the user server. Terminal input and screen output is read from and written to the associated USS message queue.

Customize the sample *hlq*.ZSERVER.JCL(CEAPROC) and copy it to the PROCLIB concatenation and name as ceaproc.

Verify and customize the high level qualifier defined for TAURHLQ, and also TCPDATA to make sure it specifies the correct TCPIP configuration data set. Similar to a logon procedure, add all required libraries for things such as panels, messages, and REXX execs that are necessary to run the user's required ISPF applications.

The initial REXX procedure CEALOGON must be located in the z/Server REXX data set.

Note: The data set *hlq*.ZSERVER.REXX has to be allocated using DDNAME SYSEXEC (not SYSPROC).

```
//CEAPROC  PROC
//CEAPROC  EXEC PGM=IKJEFT01,DYNAMNBR=175,
//             PARM='%CEALOGON',TIME=120
//TAURHLQ  SET  TAURHLQ=hlq.ZSERVER
//TCPDATA  SET  TCPDATA=TCPIP.TCPDATA
//********************************************************************
//* Z/SERVER
//*********************************************************************
//STEPLIB  DD  DISP=SHR,DSN=&TAURHLQ..LOADLIB
//ZSRVAUTH DD  DISP=SHR,DSN=&TAURHLQ..AUTHLIB
//*IPCONFIG DD DISP=SHR,DSN=&TAURHLQ..CONFIG(CEAUSER)
//SYSTCPD  DD  DISP=SHR,DSN=&TCPDATA
//CEEREX30 DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//CEATSPRT DD SYSOUT=*
//ZCOTSPRT DD TERM=TS
//ZCOTSIN  DD TERM=TS
//READER   DD  SYSOUT=(*,INTRDR)
//MAINTASK DD SYSOUT=*
//T0000001 DD SYSOUT=*
//CMDTASK  DD SYSOUT=*
//SRVTASK  DD SYSOUT=*
//CEATASK  DD SYSOUT=*
//CEAISPF  DD SYSOUT=*
//LISTENER DD  SYSOUT=*
//*********************************************************************
//*      ADD YOUR OWN LOGON PROCEDURE HERE
//*      BUT KEEP ALL &TAURHLQ DATA SETS IN THE RIGHT PLACES
//*********************************************************************
//SYSUADS  DD  DISP=SHR,DSN=SYS1.UADS
//SYSLBC   DD  DISP=SHR,DSN=SYS1.BRODCAST
//SYSPROC  DD  DISP=SHR,DSN=ISP.SISPCLIB
//         DD  DISP=SHR,DSN=SYS1.SBLSCLI0
```

```
//SYSEXEC  DD  DISP=SHR,DSN=&TAURHLQ..EXEC
//         DD  DISP=SHR,DSN=&TAURHLQ..REXX
//         DD  DISP=SHR,DSN=ISP.SISPEXEC
//SYSHELP  DD  DISP=SHR,DSN=SYS1.HELP
//         DD  DISP=SHR,DSN=ISP.SISPHELP
//ISPMLIB  DD  DISP=SHR,DSN=ISP.SISPMENU
//         DD  DISP=SHR,DSN=SYS1.SBLSMSG0
//         DD  DISP=SHR,DSN=ISF.SISFMLIB
//         DD  DISP=SHR,DSN=&TAURHLQ..MSGS
//ISPEXEC  DD  DISP=SHR,DSN=ISP.SISPEXEC
//ISPLLIB  DD  DISP=SHR,DSN=&TAURHLQ..LOADLIB
//ISPPLIB  DD  DISP=SHR,DSN=ISP.SISPPENU
//         DD  DISP=SHR,DSN=&TAURHLQ..PANELS
//         DD  DISP=SHR,DSN=SYS1.SBLSPNL0
//         DD  DISP=SHR,DSN=ISF.SISFPLIB
//ISPSLIB  DD  DISP=SHR,DSN=&TAURHLQ..SKELS
//         DD  DISP=SHR,DSN=ISP.SISPSLIB
//         DD  DISP=SHR,DSN=ISP.SISPSENU
//         DD  DISP=SHR,DSN=ISF.SISFSLIB
//         DD  DISP=SHR,DSN=SYS1.SBLSKEL0
//ISPTLIB  DD  DISP=SHR,DSN=ISP.SISPTENU
//         DD  DISP=SHR,DSN=SYS1.SBLSTBL0
//         DD  DISP=SHR,DSN=ISF.SISFTLIB
//ISPCTL1  DD  DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
//             DCB=(LRECL=80,BLKSIZE=800,RECFM=FB)
//ISPCTL2  DD  DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
//             DCB=(LRECL=80,BLKSIZE=800,RECFM=FB)
//ISPLST1  DD  DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
//             DCB=(LRECL=121,BLKSIZE=1210,RECFM=FBA)
//ISPLST2  DD  DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
//             DCB=(LRECL=121,BLKSIZE=1210,RECFM=FBA)
//SDSFMENU DD  DSN=ISF.SISFPLIB,DISP=SHR
//         PEND
```

Parameter P1 have been added to support parameters required for startup to the CEA user server:

- P1 is an internal parameter the CEA user server uses. This must match the CEA_LOGONPROC of the CEA user server.

The SYSTCPD DD statement has been added to provide information about the IP stack on your host to the started task.

The name of the job step in the JCL procedure must match the same value you entered as the CEA_LOGONPROC in the configuration of either the Unified scheduler or the CEA scheduler. Failure to do so results in the User Server starting, but the client is unable to connect to it.

For example, if you have CEA_LOGONPROC="MYPROC" in your configuration then it would look like:

```
//MYPROC   PROC
//MYPROC   EXEC PGM=IKJEFT01,DYNAMNBR=175,
//             PARM='%CEALOGON,TIME=120
//TAURHLQ  SET  TAURHLQ=hlq.ZSERVER
//TCPDATA  SET  TCPDATA=TCPIP.TCPDATA
...
```

Optional: Customizing a CEA-launched TSO user address space JCL (CEAPROC)

A z/Server user server started as a CEA-launched TSO user address space CEAPROC is started by the z/Server scheduler address space. It behaves like a foreground TSO address space running with the TSO user ID of the client user who requested the start of the user server. Terminal input and screen output is read from and written to the associated USS message queue.

Customize the sample *hlq*.ZSERVER.JCL(CEAPROC) and copy it to the PROCLIB concatenation and name as ceaproc.

Verify and customize the high level qualifier defined for TAURHLQ, and also TCPDATA to make sure it specifies the correct TCPIP configuration data set. Similar to a logon procedure, add all required libraries for things such as panels, messages, and REXX execs that are necessary to run the user's required ISPF applications.

The initial REXX procedure CEALOGON must be located in the z/Server REXX data set.

Note: The data set *hlq*.ZSERVER.REXX has to be allocated using DDNAME SYSEXEC (not SYSPROC).

```
//CEAPROC  PROC
//CEAPROC  EXEC PGM=IKJEFT01,DYNAMNBR=175,
//             PARM='%CEALOGON',TIME=120
//TAURHLQ  SET  TAURHLQ=hlq.ZSERVER
//TCPDATA  SET  TCPDATA=TCPIP.TCPDATA
//********************************************************************
//* Z/SERVER
//*********************************************************************
//STEPLIB  DD  DISP=SHR,DSN=&TAURHLQ..LOADLIB
//ZSRVAUTH DD  DISP=SHR,DSN=&TAURHLQ..AUTHLIB
//*IPCONFIG DD DISP=SHR,DSN=&TAURHLQ..CONFIG(CEAUSER)
//SYSTCPD  DD  DISP=SHR,DSN=&TCPDATA
//CEEREX30 DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//CEATSPRT DD SYSOUT=*
//ZCOTSPRT DD TERM=TS
//ZCOTSIN  DD TERM=TS
//READER   DD  SYSOUT=(*,INTRDR)
//MAINTASK DD SYSOUT=*
//T0000001 DD SYSOUT=*
//CMDTASK  DD SYSOUT=*
//SRVTASK  DD SYSOUT=*
//CEATASK  DD SYSOUT=*
//CEAISPF  DD SYSOUT=*
//LISTENER DD  SYSOUT=*
//*********************************************************************
//*      ADD YOUR OWN LOGON PROCEDURE HERE
//*      BUT KEEP ALL &TAURHLQ DATA SETS IN THE RIGHT PLACES
//*********************************************************************
//SYSUADS  DD  DISP=SHR,DSN=SYS1.UADS
//SYSLBC   DD  DISP=SHR,DSN=SYS1.BRODCAST
//SYSPROC  DD  DISP=SHR,DSN=ISP.SISPCLIB
//         DD  DISP=SHR,DSN=SYS1.SBLSCLI0
//SYSEXEC  DD  DISP=SHR,DSN=&TAURHLQ..EXEC
//         DD  DISP=SHR,DSN=&TAURHLQ..REXX
//         DD  DISP=SHR,DSN=ISP.SISPEXEC
//SYSHELP  DD  DISP=SHR,DSN=SYS1.HELP
//         DD  DISP=SHR,DSN=ISP.SISPHELP
//ISPMLIB  DD  DISP=SHR,DSN=ISP.SISPMENU
//         DD  DISP=SHR,DSN=SYS1.SBLSMSG0
//         DD  DISP=SHR,DSN=ISF.SISFMLIB
//         DD  DISP=SHR,DSN=&TAURHLQ..MSGS
//ISPEXEC  DD  DISP=SHR,DSN=ISP.SISPEXEC
//ISPLLIB  DD  DISP=SHR,DSN=&TAURHLQ..LOADLIB
//ISPPLIB  DD  DISP=SHR,DSN=ISP.SISPPENU
//         DD  DISP=SHR,DSN=&TAURHLQ..PANELS
//         DD  DISP=SHR,DSN=SYS1.SBLSPNL0
//         DD  DISP=SHR,DSN=ISF.SISFPLIB
//ISPSLIB  DD  DISP=SHR,DSN=&TAURHLQ..SKELS
//         DD  DISP=SHR,DSN=ISP.SISPSLIB
//         DD  DISP=SHR,DSN=ISP.SISPSENU
//         DD  DISP=SHR,DSN=ISF.SISFSLIB
//         DD  DISP=SHR,DSN=SYS1.SBLSKEL0
//ISPTLIB  DD  DISP=SHR,DSN=ISP.SISPTENU
//         DD  DISP=SHR,DSN=SYS1.SBLSTBL0
//         DD  DISP=SHR,DSN=ISF.SISFTLIB
```

```
//ISPCTL1  DD  DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
//             DCB=(LRECL=80,BLKSIZE=800,RECFM=FB)
//ISPCTL2  DD  DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
//             DCB=(LRECL=80,BLKSIZE=800,RECFM=FB)
//ISPLST1  DD  DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
//             DCB=(LRECL=121,BLKSIZE=1210,RECFM=FBA)
//ISPLST2  DD  DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
//             DCB=(LRECL=121,BLKSIZE=1210,RECFM=FBA)
//SDSFMENU DD  DSN=ISF.SISFPLIB,DISP=SHR
//         PEND
```

Parameter P1 have been added to support parameters required for startup to the CEA user server:

- P1 is an internal parameter the CEA user server uses. This must match the CEA_LOGONPROC of the CEA user server.

The SYSTCPD DD statement has been added to provide information about the IP stack on your host to the started task.

The name of the job step in the JCL procedure must match the same value you entered as the CEA_LOGONPROC in the configuration of either the Unified scheduler or the CEA scheduler. Failure to do so results in the User Server starting, but the client is unable to connect to it.

For example, if you have CEA_LOGONPROC="MYPROC" in your configuration then it would look like:

```
//MYPROC   PROC
//MYPROC   EXEC PGM=IKJEFT01,DYNAMNBR=175,
//             PARM='%CEALOGON,TIME=120
//TAURHLQ  SET  TAURHLQ=hlq.ZSERVER
//TCPDATA  SET  TCPDATA=TCPIP.TCPDATA
...
```

Configuring z/Server

Once you have reviewed and modified all of the JCL you can move on to configuring the z/Server feature itself. The configuration for z/Server is stored in XML format and is in *hlq*.MFA.CNTL(MFAXML).

```
<Configuration
                     TCP_PORT="2020"
                     NETWORK_ID="DDINET1"
                     LU62_APPLID="MFM62ACB"
                     APPLID_PASSWORD="MFM62PSW"
                     TRACING="0"
                     ORGANIZATION="YOUR_COMPANY_NAME"
                     SVC_NO="NNN"
                     DSP_TOKEN="TAURSERV"
                     IPSTACK="TCPIP"
>
...
    <!-- Application Server address space definitions  -->
    <ApplicationServers>
        <!-- z/Server Scheduler -->
        <Scheduler   SCHEDULER_NAME="TAURISPF" LISTENER_PORT="1200"
                     FIRST_PORT="1201" LAST_PORT="1249"
                     USER_SERVER_JOBNAME="IVPUSRT" CCSID="037"
                     DEFAULT="1"
        >
            <UserServer CCSID="037" />
        </Scheduler>
    </ApplicationServers>
...
</Configuration>
```

The aspects of z/Server's behavior that you want to modify determine the parts of the XML that you need to edit.

General configuration

The following general configuration parameters apply to both unified and individual schedulers:

SVC_NO

Change the SVC number from NNN to the required value for your host. The default is 238.

IPSTACK

Specifies the name of the IP stack used for processing.

DSP_TOKEN

If you are running multiple z/Server instances DSP_TOKEN must be different from any other token used by any other MFA Server holder task. Make sure that the MAXCADS definition in the IEASYSxx member of the parmlib concatenation specifies a sufficiently high value to allow for the creation of that common access data space.

Use the z/OS health checker to see how many common access data spaces are in use. If the value is too low, the data space creation will fail. The value in IEASYSxx can only be increased via IPL.

See *Configuration reference* for more information on configuration parameters.

Scheduler configuration

The scheduler combines the functionality of for STC and CEA into one started task.

For example:

```
<Scheduler
        SCHEDULER_NAME="TAURISPF" LISTENER_PORT="1200" FIRST_PORT="1201"
        LAST_PORT="1249" USER_SERVER_JOBNAME="IVPUSRT" CCSID="037"
        DEFAULT="1"
>
       <UserServer CCSID="037" />
</Scheduler>
```

Note: If you have change the name of the STC scheduler JCL procedure (TAURISPF) in the PROCLIB you must change the SCHEDULER_NAME value to match.

If you have change the name of the STC user server JCL procedure (IVPUSRT) in the PROCLIB you must change the USER_SERVER_JOBNAME value to match.

Default Scheduler

By specifying DEFAULT="1" in a scheduler you are instructing the MFA Server that this is the Scheduler to use when the AWM client has entered the value of "DEFAULT" in the **Scheduler Name** field of the **New Connection** dialog box. Only one scheduler should be marked as the default.

You can have multiple Schedulers defined and you can specify their names in the **Scheduler Name** field of the **New Connection** dialog box.

If you type DEFAULT in the **Scheduler Name** field of the **New Connection** dialog box, and there is no default defined you will get the following message saying there is no default scheduler defined:

No default z/Server Scheduler configured.

If you specify a **Scheduler Name** that does not exist you get an error message saying that a matching scheduler could not be found:

Unable to find matching server details for z/Server Scheduler named Scheduler Name.

Legacy Clients

For clients supplied with versions of Enterprise Developer earlier than 3.0 the scheduler is unable to determine what type of user server the client requires. By default, the scheduler attempts to start an STC user server. You can specify the default user server to be started by the scheduler by adding the DEF_USRSRV_MODE parameter with the value of STC or CEA.

For CEA

To configure the scheduler for CEA, you need to add the following to the configuration:

CEA_LOGONPROC

This points to the logon procedure to be used when a CEA-launched TSO user address space is started. See Optional: Customizing a CEA-launched TSO user address space (CEAPROC) for information on how to configure this logon procedure.

CEA_ACCOUNT

This must be set to a valid accounting number that all TSO users that use this scheduler have access to. The accounting number used must be defined in general resource class ACCTNUM. All TSO users that use the CEA Scheduler.

For example:

```
<Scheduler
        SCHEDULER_NAME="TAURISPF" LISTENER_PORT="1200" FIRST_PORT="1201"
        LAST_PORT="1249" USER_SERVER_JOBNAME="IVPUSRT" CCSID="037"
    CEA_LOGONPROC=”CEAPROC” CEA_ACCOUNT=”ACCT#”
>
       <UserServer CCSID="037" />
</Scheduler>
```

Note: If you change the name of the CEA user server JCL procedure (CEAPROC) in the PROCLIB you must change the CEA_LOGONPROC value to match.

Optionally, if you have uncommented MAILHDR in the scheduler procedure, you need to add the MAIL_NOTIFY="1" attribute to the scheduler section.

For example:

```
<Scheduler
        SCHEDULER_NAME="TAURISPF" LISTENER_PORT="1200" FIRST_PORT="1201"
        LAST_PORT="1249" USER_SERVER_JOBNAME="IVPUSRT" CCSID="037"
    MAIL_NOTIFY=”1”
>
       <UserServer CCSID="037" />
</Scheduler>
```

See *Configuration parameters reference* for more information on the available parameters for a scheduler address space.

See *Configuration reference* for more information on the parameters applicable to a CEA scheduler.

Troubleshooting the z/Server configuration

You can use the following diagnostic tools and information to troubleshoot issues with the z/Server configuration:

Mainframe Access Configuration Utility

The z/Server Configuration Utility makes it easier to download, validate and edit a configuration file. This utility is available from the Product Updates section of the *Micro Focus SupportLine* Web site. To download the utility, once you have logged in to the SupportLine site, navigate to the latest version of Enterprise Developer and select z/Server Configuration Utility.

Error messages

Use the information in any error messages to determine what the possible issues might be.

For example, trying to start the MFA Server holder process with an invalid XML results in the MFA Server holder failing to start and you receive a message indicating there is a problem with the XML validation. For example:

```
XCO0021E  10:50:59.846 XML Validation has failed. Reason = 8800. Error offset
into XML document = 00000020.
```

Such messages include a reason code and an error offset:

Reason The `Reason` code indicates a specific problem with the XML code - check the section *Reason codes listed by value* in IBM's "z/OS V2R1.0 XML System Services User's Guide and Reference" (SA38-0681).

Error offset The location in hexadecimal from the start of the XML document where the error occurred.

The error message quoted earlier is caused by the following invalid XML code:

```
<Configuration
                     TCP_PORT="2020"
                     NETWORK_ID="DDINET1"
                     LU62_APPLID="MFM62ACB"
                     APPLID_PASSWORD="MFM62PSW"
                     TRACING="0"
                     ORGANIZATION="YOUR_COMPANY_NAME"
                     SVC_NO="NNN"
                     DSP_TOKEN="TAURSERV"
                     IPSTACK="TCPIP"
>
...
    <!-- Application Server address space definitions  -->
    <ApplicationServers>
        <!-- z/Server Scheduler -->
        <Scheduler   SCHEDULER_NAME="TAURISPF" LISTENER_PORT="1200"
                     FIRST_PORT="1201" LAST_PORT="1249"
                     USER_SERVER_JOBNAME="IVPUSRT" CCSID="037"
                     DEFAULT="1"
        >
            <UserServer CCSID="037" />
        </Scheduler>
    </ApplicationServers>
...
</Configuration>
```

Use the reason code and the offset shown in this message to determine the cause of the error. In this example, the offset points to a location in the first line of the XML configuration file. Here there is an invalid attribute value in the `<Configuration>` section - `SVC_NO="NNN"` is invalid because `SVC_NO` must be a numeric value.

XML validation tools

You can use any third-party XML validation tool to check the XML configuration file against the XML schema file available in `hlq.ZSERVER.CONFIG(MFASCHEM)`.

Customizing GTF tracing

If SupportLine asks for a GTF trace of z/Server and your installation does not have its own customized procedure, then customize the hlq.ZSERVER.JCL(IVPGTF) and copy it to the PROCLIB concatenation. z/Server writes GTF trace user records with number 3E8.

```
//TAUGTF  PROC M=IVPGTF,
//             CYL=100,
//             PROG=AHLGTF
//IEFPROC EXEC PGM=&PROG,
//             PARM='MODE=EXT,DEBUG=NO,TIME=YES,NOPROMPT',
//               REGION=0M
//IEFRDER DD   DSNAME=HLQ.GTFTRACE.&M..D&LYYMMDD..T&LHHMMSS,
//               SPACE=(CYL,(&CYL),,CONTIG),
//               RECFM=VB,
//               DISP=(NEW,CATLG)
//SYSLIB  DD   DSNAME=HLQ.ZSERVER.DATA(&M),DISP=SHR
```

Make sure that the IEFRDER data set can be allocated without RACF errors.

Activation of GTF for user record 3E8 alters the way z/Server writes messages. There is no need to set a specific trace level; all events are now traced. The trace output is written to a GTF trace data set, no longer to the JESx job log. Also, not all messages from trace level 0 will appear in the JESx job log.

Note: If more than one z/Server is running, starting GTF trace for user record 3E8 will automatically set the trace level to full tracing for all servers that are running. To avoid JESx spool problems, the trace level of the unaffected address spaces should be reduced to the desired value x using the operator command

```
F <scheduler|userserver>,TRACE,LEVEL=x
```

User authorizations

As stated above, all users need an OMVS segment for TCPIP to work correctly. In addition, the scheduler task and the holder task must be authorized in class OPERCMDS to issue the following operator commands:

Function/Configuration Parameter	Type	Command
START/STOP/CANCEL	MVS command	START <scheduler> START <userserver> STOP <scheduler> STOP <userserver> CANCEL <scheduler> CANCEL <userserver>
PORTCHECK=1 (Port check function)	MVS command	DISPLAY TCPIP
TSOE_CLEANUP=1	JESx command	JES2: $C JES3: *F

Note: If the configuration parameters listed above are not in use (which means they are set to 0), the appropriate commands are not executed and do not need to be authorized.

If the JESSPOOL RACF class is active, a RACF general resource profile of the following format must be defined for each user:

```
<NODENAME>.<STCUSR>.<USERSERVER>.**
```

where:

<NODENAME>	is the installation’s node name
<STCUSR>	is the started task user ID of the user server
<USERSERVER>	is the job name of the user server (TSO user ID plus TSOE_JOBCHAR)

Every TSO user must have UPDATE access to this profile allowing every TSO user to allocate spool datasets that begin with a high level qualifier equal to the STC user ID.

Example: Assuming the node id is NODE, the user ID of the z/Server started tasks is TAUUSR, TSOE_JOBCHAR is Z, and the user ID is USR123, then the RACF general resource profile should be defined as follows:

```
NODE.TAUUSR.ZUSR123.**
```

If the installation protects the use of Extended MCS consoles, then every scheduler address space must be authorized to activate an EMCS console. The naming convention for these consoles is aaaaxxxx with aaaa the first to fourth character of the system name and xxxx the address space ID of the address space establishing the EMCS console in hexadecimal notation.

Enable model configuration

Using an Eclipse client, ISPF applications can be modeled. For more information, see *Attaching ISPF tools* in the *Workflow Manager Configuration Guide*. To enable modeling, the master configuration data set hlq.ZSERVER.MASTER must be customized:

```
********************************************************
* System and Application Workflow Model Definitions
* Do not change the key words:
*    System:
*    Appl:
*    Conf:
*    Version:
*    INFO:
*    User:
*
* Enter an unique logical name for this mainframe system
*
System: XXXXXXXXXX
*
* MVS PROJECTS EXAMPLE APPLICATION
*
User:
* application name
Appl: MVS Projects Sample Application
* location of the application configuration file
Conf: mvs:'hlq.ZSERVER.XML(MVSCONF)'
* application version number
Version: 2.3.1
* process information
Property:
MVS_PROP_STATIC_SYSLIB_ASM=SYS1.MACLIB;SYS1.MODGEN;&userid..BANKVSAM.MACLIB
Property: MVS_PROP_STATIC_SYSLIB_COBOL=&userid..BANKVSAM.COPYLIB
Property: MVS_PROP_STATIC_SYSLIB_PLI=&userid..STAFF.INCLUDE
INFO:
* end of application definition. do not delete this line
EndAppl:
EndUser:
*
* AWM dialog sample application
*
User:
* application name
Appl: ISPF Dialogs
* location of the application configuration file
Conf: MVS:'hlq.zserver.XML(TAUDIALG)'
* application version number
Version: 1.4
* process information
INFO:
* end of application definition. do not delete this line
EndAppl:
EndUser:
```

Specify a system name after "System:", for example, LPAR1. Replace every occurrence of hlq.zserver after "Conf:" with the high level qualifier chosen for the installation.

There is a REXX exec in hlq.ZSERVER.EXEC named TAUZCAPP which reads the master configuration data set and sends the content to the client in formatted form. Replace hlq.ZSERVER after "sysdsn=" with the high level qualifier chosen for the installation.

When a client logs on using a new connection for the first time, the master configuration is sent from z/Server to the client and shows up in the Application Explorer view. This is done in the background, and a user server address space is started to send this master configuration. On subsequent logons (when the master configuration is already available), no user server is started. A user server address space is only started when a client request requires an ISPF environment, for example, when an AWM ISPF tool is called.

Optional customization

ISPF user profile

z/Server allows more than one user server per TSO user to be run in parallel. Each user server needs its own exclusive ISPF environment. This implies the allocation of an ISPF user profile dataset. The allocation of the user profile dataset is done in the sample REXX exec IVPINIT1 and should be customized to adhere to the installation's standards.

ISPF user profile allocation is done as follows:

- DD statement ISPPROF is allocated to a temporary dataset. If a z/Server ISPF profile dataset named userid.TAUZCISP.PROFILE already exists for the TSO user, the content of this ISPF profile is copied using IEBGENER to a temporary dataset allocated under the ISPPROF DD statement.
- ISPF is started and control is passed to REXX exec IVPINIT2.
- When the ISPF session terminates, control returns to REXX exec IVPINIT1, and the temporary ISPF profile is copied back to z/Server profile dataset userid.TAUZCISP.PROFILE.
- The temporary ISPF profile dataset is deleted.

We recommend that you allocate the temporary ISPF profile dataset to an SMS managed temporary dataset pool, which is automatically deleted according to installations' standards. In that case, the deletion of the temporary ISPF profile dataset in the REXX IVPINIT1 can be omitted.

Usage in a parallel sysplex

z/Server itself does not use any sysplex services. It relies on the ability to connect to an IP address specified using a TCPIP header. This is normally done under the covers by an Eclipse client:

1. The Eclipse client asks for "Connect" to an IP address (or symbolic name) using a port specified in the client. This port corresponds to the PORT parameter that the scheduler address space listens on. As a response, z/Server sends back the IP address of the system that this particular scheduler address space runs on.
2. The next transaction initiated from the client (which includes the TCPIP function connect - different from the function "Connect" in 1.) uses this IP address for further communication and (IP-)connection.
3. As long as the network is set up in such a way that usage of this IP address will guarantee connection to the same host system, the first connection used (from the Eclipse function "Connect") z/Server functions without problems.

Note that it is mandatory to have a one-to-one relation between the IP address and the system name. A configuration where an IP address can resolve to different systems (for load balancing, for instance) is not supported. (See also *VIPA*).

z/Server startup and installation verification

Now you can start z/Server. By issuing the start MFA Server command, MFA Server then begins to start and automatically starts any schedulers defined in the XML configuration file.

Example starting the MFA Server task:

```
 S MFA
```

By default, during MFA Server startup z/Server automatically starts any schedulers defined in the XML configuration file. If you want to disable this and start the schedulers independently of the MFA Server holder, add the AUTOSTART="0" XML attribute to the <Configuration /> section of your XML configuration file.

Verifying z/Server feature support startup

In addition to the regular JESx DD statements, the job log of a running MFA Server address space contains the DD names MAINTASK and SYSTSPRT.

DD MAINTASK shows if the data space was created correctly and the SVC routine installed. The MFA Server address space MFA initializes z/Server feature support correctly when HLD0002I is the last message.

```
 HLD0090I  15:35:24.879 Holder called as subroutine
                        Function.: S
                        Parms....: ,,
 HLD0001I  15:35:24.916 zServer holder task 04000000 startup
 HLD0009I  15:35:25.039 TSO-Environment successful created
                        A(Comand-Processor-Parameter-List) : 009B9CD0
 HLD0078I  15:35:25.040 Read   of NT-Pair E9E2D9E500F864804040404040404040
ended with RC 00000004 (hex).
 ZCF0036I  15:35:25.040 Parameter for z/Server: ,,
 ZCF0038I  15:35:25.044 zServer configuration utility is using
                        Token ......... not specified
                        Config key .... not specified
 ZCF0060I  15:35:25.100 Using configuration style XML
 ZCF0058I
========================================================================
          Completing configuration TAURISPF

========================================================================
 ZCF0024W  15:35:25.100 Server is limited server due to specification of
TSOE_STCID
 ZCF0025W  15:35:25.100 The following configuartion parameters are not used
due to specification od TSOE_STCID
                        TSOE_CLEANUP
                        TSOE_JOBPREFIX
                        TSOE_JOBSEARCH
 ZCF0058I
========================================================================
          Completing configuration IVPUSR

========================================================================
 ZCF0024W  15:35:25.100 Server is limited server due to specification of
TSOE_STCID
 ZCF0025W  15:35:25.100 The following configuartion parameters are not used
due to specification od TSOE_STCID
                        TSOE_CLEANUP
                        TSOE_JOBPREFIX
                        TSOE_JOBSEARCH
 ZCF0058I
========================================================================
          Completing configuration CEAPROC

========================================================================
 ZCF0024W  15:35:25.100 Server is limited server due to specification of
TSOE_STCID
 ZCF0025W  15:35:25.100 The following configuration parameters are not used
due to specification of TSOE_STCID
```

```
                        TSOE_CLEANUP
                        TSOE_JOBPREFIX
                        TSOE_JOBSEARCH
 HLD0069I  15:35:25.100 Holder task configuration
                        Size data space specified....:       262144
Blocks
                        Size data space required.....:          168
Blocks
                        Delay .......................:          500      csecs
                        Trace level..................:            0
                        Modify limit.................:            1
                        Size RESTAB in blocks........:          160
                        Max. # of entries in RESTAB..:         5120
                        Reply........................: NO
 HLD0018I
-----------------------------------------------------------------------
          Installing User-SVC 0238

-----------------------------------------------------------------------
 HLD0019I  15:35:25.103 User-SVC 0238 successfully installed.
                        Pgm : TAURAUTH EpAddr : 8E2CC8A8
 HLD0021I
-----------------------------------------------------------------------
          Creating Common-Dataspace TAURSERV

-----------------------------------------------------------------------
 HLD0003I  15:35:25.103 Creating Dataspace
                        Dataspace-Name    :  TAURSERV
                        Dataspace-Size    :        262144 Blocks
                        Dataspace-StgKey  :  00
 HLD0025I  15:35:25.103 Function Create Dataspace     ended successful
 HLD0025I  15:35:25.103 Function Create Alet          ended successful
 HLD0025I  15:35:25.103 Function Create Name/Token    ended successful
 HLD0025I  15:35:25.103 Function Create Name/Token 2  ended successful
 HLD0033I  15:35:25.103 Server Information
                        Lowest  port over all Servers   :         5151
                        Highest port over all Servers   :         5199
                        Maximum numbers of    Servers   :            1
 HLD0035I  15:35:25.103 Initializing data space for user administration
 HLD0076I  15:35:25.103 Resource table size increased to         5120 entries
 HLD0048I
-----------------------------------------------------------------------
          Starting Server

-----------------------------------------------------------------------
 HLD0049I  15:35:25.116 Starting server TAURISPF
 HLD0002I  15:35:25.125 zServer holder task ready for commands
```

Verifying scheduler startup

If the MFA Server holder has successfully started it will also have started your scheduler. You can now verify that the scheduler has started correctly.

In addition to the regular JESx DD statements and SYSTSPRT/ZCOTSPRT, the job log of a running scheduler address space should contain the following DD names:

DD Name	Function
MAINTASK	(LE) Messages from MAINTASK.
LISTENER	(LE) Listener for client requests.
CMDTASK	(LE) Messages from CMDTASK; contains command output.

DD Name	Function
SRVTASK	(LE) Messages from SRVTASK.
MSGTASK	(LE) Task to queue messages to a USS message queue.
T000000x, x=1..NUMTCB	(LE) Messages from T000000x, the worker tasks.
MAILBOX	If email notification is configured.

A number of messages reference these DD statements as "LE message files".

A scheduler address space has started successfully when the hardcopy log shows:

```
+TAU0067I  zServer startup completed for TAURISPF  for JESx and ASID nnnn
```

To verify that the scheduler is responding to work requests you can customize and submit the verification job hlq.ZSERVER.JCL(IVPVERI):

```
//TAURHLQ  SET TAURHLQ=hlq.ZSERVER
//JOBLIB   DD DISP=SHR,DSN=&TAURHLQ..LOADLIB
//*------------------------------------------------------------------*
//*- Install Verification Job                                       -*
//*-                                                                -*
//*- Valid Selects                                                  -*
//*- ===========================================================    -*
//*- 1                    EXEC REXX IVPREXXE  (echo rexx)           -*
//*------------------------------------------------------------------*
//TAURIVP  EXEC PGM=TAURIVP
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SYSIN    DD *
IVP_SELECT = 1
IVP_Server = 'nnn.nnn.nnn.nnn'
IVP_Port   = 1200;
IVP_COMPAT = 2
IVP_IPNAME = 'TCPIP'
;
/*
```

You need to do the following to customize hlq.ZSERVER.JCL(IVPVERI):

- Add a valid job card.
- Specify the data set prefix defined (TAURHLQ).
- Specify the correct IP address of the host system (IVP_Server).
- Specify the port the scheduler address space is listening on (IVP_Port).
- Specify the name of the IP stack used (IVP_IPNAME).

The IVPVERI output should look like this:

```
IVP0002I Selection code for IVP was :   1
         EXEC IVPREXXI (echo REXX)
IVP0003I call to zServer ended with  RC :       0
===================================================================
Answer from zServer is       203 Bytes long.
===================================================================
00000000   000000CB 4CD4E2C7 40C6C1C3 C9C47E7F   *    <MSG FACID="*
00000010   D3C9C37F 40D4E2C7 D5D67E7F F0F0F1F0   *LIC" MSGNO="0010*
00000020   7F40E2C5 E5C5D9C9 E3E87E7F C57F6E40   *" SEVERITY="E"> *
00000030   F0F97AF4 F87AF3F7 4BF0F8F9 40D38983   *09:48:37.089 Lic*
00000040   85958385 40A58996 9381A389 96957A40   *ence violation: *
00000050   E38889A2 40A28599 A5859940 83819540   *This server can *
00000060   969593A8 40828540 A4A28584 40A38899   *only be used thr*
00000070   96A48788 40819540 E3C1E4D9 E4E24083   *ough a licenced *
00000080   93898595 A34B4040 40404040 40404040   *client.         *
00000090   40404040 40404040 40404040 40404040   *                *
```

```
000000A0   40404040 40404040 40404040 40404040   *                *
000000B0   40404040 40404040 40404040 40404040   *                *
000000C0   40404040 404C61D4 E2C76E00 00000000   *      </MSG>    *
```

If you receive a return code of 49 in message IVP0003I, check if numbering was inadvertently turned on in the member from which this job was submitted. If it was, turn off numbering.

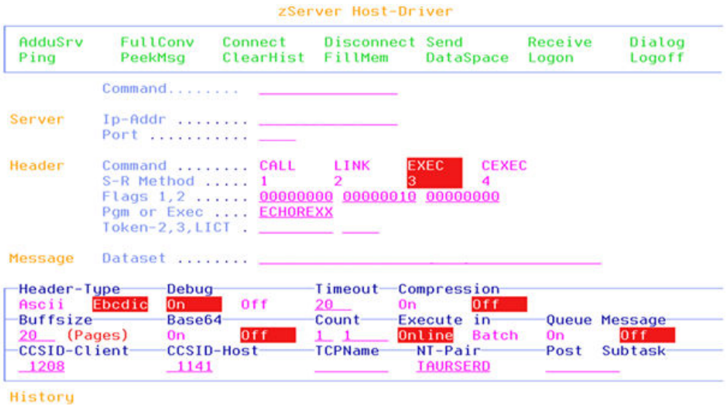
Verifying user server startup

To verify the user server startup you must start a user server, as these are only started on demand. Instructions on starting a user server are below.

We strongly recommended that you execute the following installation verification procedure for a z/Server user server before trying to connect from an Eclipse client. This procedure verifies that the configuration in the z/OS system is correct, without possible added complications from the network configuration.

The start of a z/Server user server can be tested by executing the REXX procedure TAU located in the REXX library. This is limited to support of port numbers up to a length of four characters.

Type EX in front of member TAU to execute the REXX exec. The "zServer Host -Driver" panel should open:



```
                      zServer Host-Driver
 AdduSrv     FullConv    Connect     Disconnect Send        Receive     Dialog
 Ping        PeekMsg     ClearHist   FillMem    DataSpace   Logon       Logoff
             Command........   _________________

Server       Ip-Addr ........  _________________
             Port ...........  ____

Header       Command ........  CALL     LINK     EXEC     CEXEC
             S-R Method .....  1        2        3        4
             Flags 1,2 ......  00000000 00000010 00000000
             Pgm or Exec ....  ECHOREXX
             Token-2,3,LICT .  ________ ____

Message      Dataset ........  ____________________________________________

 Header-Type         Debug                Timeout  Compression
 Ascii    Ebcdic     On         Off       20__     On          Off
 Buffsize            Base64               Count    Execute  in         Queue Message
 20__  (Pages)       On         Off       1_ 1____ Online   Batch      On          Off
 CCSID-Client        CCSID-Host           TCPName     NT-Pair          Post  Subtask
 _1208               _1141                ________    TAURSERD         ________

History
```

Verify that the IP stack used is TCPIP. If it is not, change the name in the last line under TCPName to the correct IP stack name. Press Enter. A message will confirm that another IP stack name is now used.

Set the IP address (IP-Addr) to the "home" IP address of the IP stack used.

Set the field Port to the listener port configured in the configuration dataset of the z/Server scheduler task.

Type "addusrv" (without the quotes) in the line named Command or select the point and shoot field AdduSrv in the panel header. A popup panel is shown prompting for the TSO password of the TSO user ID used. After entering the correct password and submitting the information, a z/Server user server should be started. Address space creation takes some time, so after a while, a response like the following is displayed:

```
Answer from Server for Message   1 is       204 Bytes long.
==================================================================

00000000   000000CC 4CD4E2C7 40C6C1C3 C9C47E7F   *...¦<MSG FACID="*
00000010   E2D3D97F 40D4E2C7 D5D67E7F F0F0F2F3   *SLR" MSGNO="0023*
00000020   7F40E2C5 E5C5D9C9 E3E87E7F C97F6E40   *" SEVERITY="I"> *
00000030   F1F27AF1 F37AF0F1 4BF0F7F8 40E3E2D6   *12:13:01.078 TSO*
00000040   61C540A2 8599A585 9940E9E2 D6C6D1D2   */E server Wxxxxx*
00000050   404040A6 89A38840 D1D6C2C9 C440E2E3   *   with JOBID ST*
00000060   C3F0F8F3 F6F14086 969940A4 A2859940   *C08361 for user *
00000070   E2D6C6D1 D2404040 40A2A482 9489A3A3   *xxxxx    submitt*
00000080   85844B40 40404040 40404040 40404040   *ed.             *
00000090   40404040 40404040 40404040 40404040   *                *
000000A0   40404040 40404040 40404040 40404040   *                *
000000B0   40404040 40404040 40404040 40404040   *                *
000000C0   40404040 40404C61 D4E2C76E            *       </MSG>   *
```

If the response is not as shown above, then something went wrong during the start of the user server and you need to determine what that was:

- Check the JCL of the user server started task (IVPUSRT). There may be a JCL error in this procedure (e.g. a mistyped dataset name).
- Check for RACF problems in the hardcopy log.
- Check the MAINTASK DD statement in the user server joblog for problems and act accordingly.

Now you can confirm that the user server has started correctly. The joblogs can be found under the JOB names defined, by default, as the character W prefixed to your TSO userid. In addition to the regular JESx DD statements and SYSTSPRT/ZCOTSPRT, the job log of a running scheduler address space should contain the following DD names. Note that a user server address space has two MAINTASK JESx DD statements, one for each step in the user server startup process:

DD Name	Function
MAINTASK	(LE) Messages from MAINTASK from step one of user server startup.
MAINTASK	(LE) Messages from MAINTASK from step two of user server startup.
LISTENER	(LE) Listener for client requests.
CMDTASK	(LE) Messages from CMDTASK; contains command output.
SRVTASK	(LE) Messages from SRVTASK.
T0000001	(LE) Messages from T0000001, the worker task.
ISPLOG	(ISPF) Log file.

A number of messages reference these DD statements as "LE message files".

A scheduler address space has started successfully when the hardcopy log shows the following, where XXXX is your TSO user ID:

```
+TAU0067I  zServer startup completed for WXXXX  for JESx and ASID 005E
```

If the user server was started successfully, a "Logoff" command should be sent afterwards to stop the user server again. Type "logoff" on the command line or select the point and shoot field Logoff in the panel header. The password is prompted for again. When the request is submitted with the correct password, a response like this should be displayed:

```
Answer from Server for Message   1 is       124 Bytes long.
======================================================================

00000000   0000007C 4CD4E2C7 40C6C1C3 C9C47E7F   *...§<MSG FACID="*
00000010   E2D3D97F 40D4E2C7 D5D67E7F F0F0F2F6   *SLR" MSGNO="0026*
00000020   7F40E2C5 E5C5D9C9 E3E87E7F C97F6E40   *" SEVERITY="I"> *
00000030   F1F77AF0 F57AF5F3 4BF8F4F1 40E2A396   *17:05:53.841 Sto*
00000040   97408396 94948195 84408696 9940A4A2   *p command for us*
00000050   859940E2 D6C6D1D2 40404040 81838385   *er xxxxx    acce*
00000060   97A38584 4B404040 40404040 40404040   *pted.           *
00000070   40404040 40404C61 D4E2C76E            *      </MSG>    *
```

Verifying CEA user servers (optional)

Make sure you also test the start of a CEA-launched TSO address space by changing the field Port to the listener port of the CEA scheduler and repeating the procedure for verifying a user server. Output similar to the one below in the hardcopy log indicates successful initialization of the CEA-launched TSO user server. Make sure the user server is terminated again using the logoff command as before.

```
$HASP373 userid    STARTED
```

```
IEF125I userid - LOGGED ON - TIME=16.01.44
+ISPWB000  Client requested ISPF session initialization 585
           Userid: userid   ASIDX: 006D
           Message Queue: 0000655407 CCSID: 00037
+TAU0001I  zServer userid    02030000  startup Tuesday   , 11 Mar 2014
16:01:49.04
+TAU0067I  zServer startup completed for userid    for JES2 and ASID 006D
```

If this message does not appear in hardcopy log, problem determination will become very difficult. Use the logon procedure specified by CEA_LOGONPROC and attempt to logon to TSO to make sure there are no JCL or ISPF errors during dataset allocation. The initial REXX exec CEALOGON is not executed, so no ISPF profile dataset is available. At the TSO READY prompt, use the following command to allocate the ISPF profile (with the correct naming conventions for the installation):

```

alloc dd(ISPPROF) dsn(userid.ISPF.ISPPROF)
```

Then call ISPF. If the logon procedure is correct, the ISPF primary option menu will be shown. Otherwise, correct any errors and repeat.

Optional: Adaptations to the IBM Developer for System z RSED installation

Customers who install the Micro Focus AWM feature into an IBM Developer for z Systems (IDz) client have a technology choice between the previously described z/Server and IBM Developer for z Systems RSED. The RSE daemon (RSED) is shipped with an IDz installation.

If RSED is used as server technology, the following changes have to be made to the RSED configuration.

ISPF Client Gateway

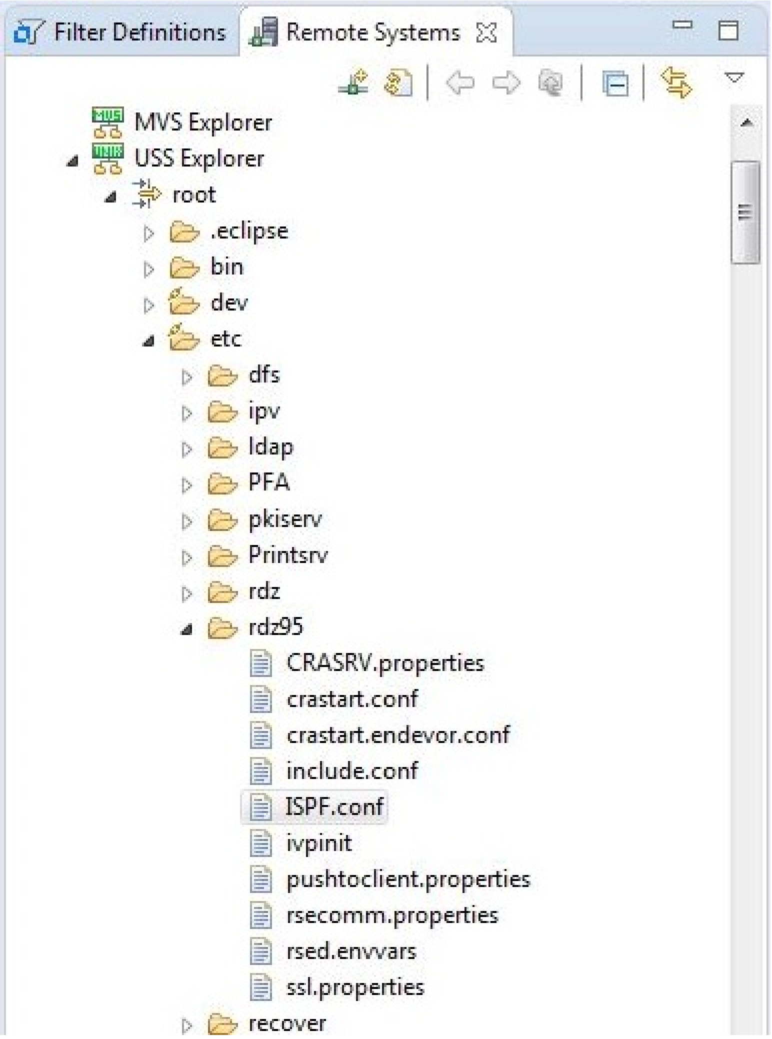
Allocate the REXX data set delivered with z/Server V2R2M01 to the ISPF Client Gateway using DDNAME SYSEXEC or SYSPROC.

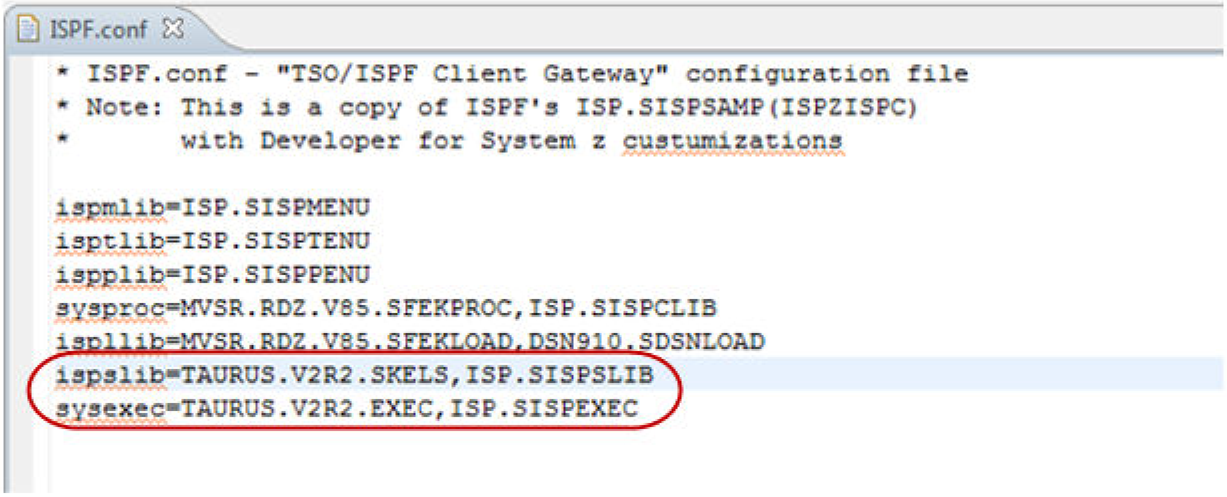
Allocate the SKELS data set delivered with z/Server V2R2M01 to the ISPF Client Gateway using DDNAME ISPSLIB.

Detailed information on how to configure the ISPF.conf file can be found in the IDz Host Configuration Guide.

If new data sets are added to ISPF.conf, make sure that all users of the ISPF Client Gateway (IDz users) have at least READ access to the data sets allocated by the client gateway.

ISPF.conf is a USS file, for example:





```
* ISPF.conf - "TSO/ISPF Client Gateway" configuration file
* Note: This is a copy of ISPF's ISP.SISPSAMP(ISPZISPC)
*       with Developer for System z custumizations

ispmlib=ISP.SISPMENU
isptlib=ISP.SISPTENU
ispplib=ISP.SISPPENU
sysproc=MVSR.RDZ.V85.SFEKPROC,ISP.SISPCLIB
ispllib=MVSR.RDZ.V85.SFEKLOAD,DSN910.SDSNLOAD
ispslib=TAURUS.V2R2.SKELS,ISP.SISPSLIB
sysexec=TAURUS.V2R2.EXEC,ISP.SISPEXEC
```

Master configuration

To work with AWM applications, a master configuration file must exist.

REXX procedure TAULAPPL must be customized to contain the data set name of the master configuration data set. The default name is hlq.SYSTEM.CONFIG.

The REXX TAULAPPL can be found on the installation EXEC library.

Change the following line in TAULAPPL according to the high level qualifier used for the installation:

```
sysdsn= 'hlq.SYSTEM.CONFIG'                 /* master config file */
```

A sample master configuration data set has been delivered as hlq.ZSERVER.MASTER. The contents of this master configuration data set contain a sample application “PDS Explorer”:

```
********************************************************************
*
* TAURUS System and Application Definition
*
* Do not change the key words:
*    System:
*    Appl:
*    Conf:
*    Version:
*    INFO:
*
* Change Activity:
*   date    name    comment
*
********************************************************************
*
* TAURUS SYSTEM
*
* Enter a unique logical name for this mainframe system
*
System: TAURUS_Custom_System
*
* TAURUS PDS EXPLORER EXAMPLE APPLICATION
*
User:
* application name
Appl: PDS Explorer
* location of the application configuration file
Conf: mvs:'hlq.TAURUS.XML(PDSECONF)'
* application version number
Version: 1.4
* process information
INFO:
* end of application definition. do not delete this line
EndAppl:
EndUser:
```

You can customize the reference to the PDS Explorer XML application model and enter a unique system name, for example, your logical LPAR name.

An AWM user needs READ access to the master configuration file and to all XML files referenced; the AWM administrator needs UPDATE access.

Host installation verification

Access to the Master Configuration File

No AWM client installation is needed to verify the AWM host installation.

Call the REXX procedure TAULAPPL from the ISPF command menu:

```
TSO EX ‘hlq.TAURUS.EXEC(TAULAPPL)
```

Or, if the TAURUS REXX library is allocated to SYSPROC or SYSEXEC

```
TSO TAULAPPL
```

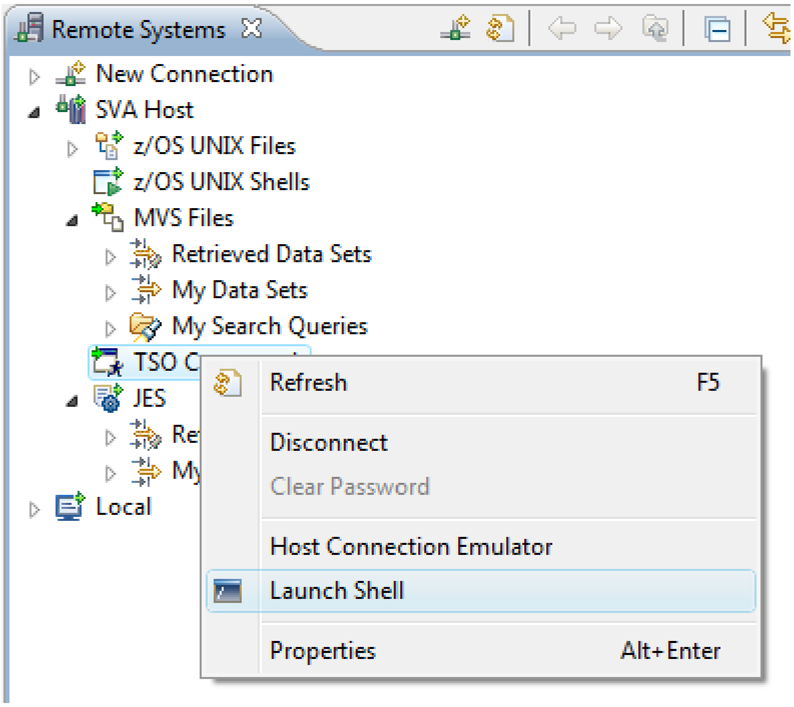
The terminal output looks like this:

```
TAU System: TAURUS_Custom_System
TAU Appl: PDS Explorer
TAU Conf: mvs: 'hlq.TAURUS.XML(PDSECONF)'
TAU Version: 1.4
TAU INFO:
TAU RC: 0
```

Verify ISPF Client Gateway

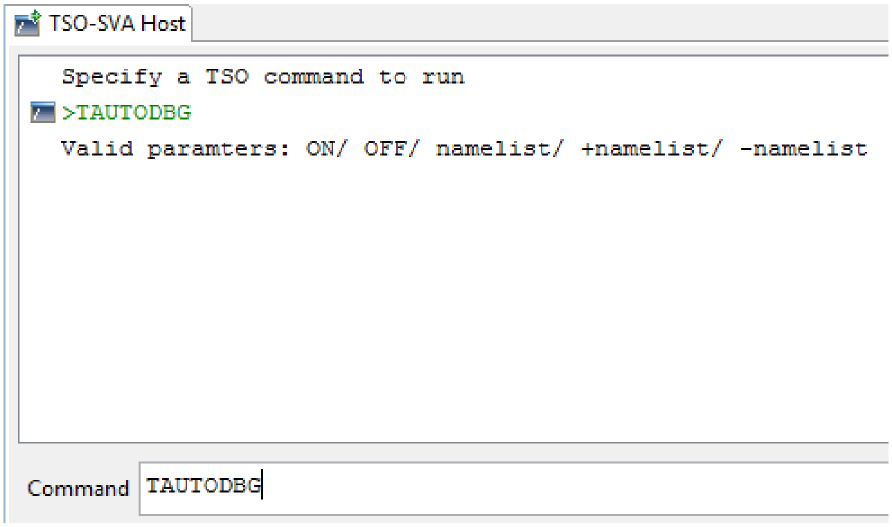
Start IDz and navigate to the “Remote projects perspective”.

Select **Launch TSO** in the context menu for the “MVS Files” entry:



The IDz TSO command window opens.

Enter TAUTODBG and press **Enter**. The following message can be seen:



```
Specify a TSO command to run
>TAUTODBG
Valid paramters: ON/ OFF/ namelist/ +namelist/ -namelist
```

Message `RC=20 invalid command` means that the REXX procedure TAUTODBG was not found. In this case, verify that the AWM REXX procedures are allocated through the ISPF Client Gateway using the SYSPROC or SYSEXEC DDNAME.

Console Commands

Introduction

Mainframe Access provides a number of console commands that you can use to control execution and display the operational status of Mainframe Access. To use these commands, you must be authorized to use z/OS system console commands. Most of the commands are variants of the z/OS Modify (F) command, with the general format:

```
F <your-procedure-name>,<command-text>
```

where:

<your-procedure-name>

The name of your Mainframe Access startup JCL. See *Mainframe Access Data Sets* for more information.

<command-text>

See *Console Commands* for more information on the commands available.

To use the z/OS Start (S) command to start Mainframe Access:

```
S <your-procedure-name>
```

To use the z/OS Stop (P) command to stop Mainframe Access:

```
P <your-procedure-name>
```

You can enter the commands in any of the following ways:

- From an z/OS system console.
- Using an interface that allows the entry of z/OS system commands, such as IBM's TSO/ISPF SDSF (System Display and Search Facility).

Console Commands

General Mainframe Access Server commands

Start

Starts the Mainframe Access task. Mainframe Access automatically starts the Mainframe Access Data Connect server, MFAS, during initialization.

Here is an example of a Start command and the response:

```

15:49:32.45 CSIRLW1  00000290  S MFA
15:49:32.47 STC05579 00000090  $HASP100 MFA ON STCINRDR
15:49:32.55 STC05579 00000290  IEF695I START MFA WITH JOBNAME MFA IS ASSIGNED
TO USER RWITEK, GROUP DEVELOP
15:49:32.55 STC05579 00000090  $HASP373 MFA  STARTED
15:49:32.55 STC05579 00000090  IEF403I MFA - STARTED - TIME=15.49.32
15:49:32.85 STC05579 00000090  MFM0014I: Mainframe Access V3.01 (ga) for
YOUR_COMPANY_NAME
15:49:32.85 STC05579 00000090  MFM0066I: Address space type is CTLRGN
15:49:32.86 STC05579 00000090  MFM0057I: GTF user record id (eid) is 00E9
15:49:32.86 STC05579 00000090  MFM0012I: TCP port number is 1503
```

```
15:49:32.86 STC05579 00000090  MFM0013I: Admin port number is 1504
15:49:32.86 STC05579 00000090  MFM0047I: DSS maximum shared public files is 10
15:49:32.86 STC05579 00000090  MFM0055I: DSS scan interval is 2 minutes
15:49:32.86 STC05579 00000090  MFM0054I: SMF recording is off, record id is
zero or invalid
15:49:32.87 STC05579 00000090  MFM0063I: MFA Direct is active
15:49:32.87 STC05579 00000090  MFM0064I: MFA Direct number of processing
tasks is 5
15:49:33.16 STC05579 00000090  MFMDS052I CA/PanValet Ready
15:49:33.21 STC05579 00000090  MFMDS055I MCG/RPC feature inactive
15:49:33.22 STC05579 00000090  MFMDS001I DSS Server Ready
15:49:36.92 STC05579 00000090  MFM0001I: Mainframe Access V3.01 (ga) is active
15:49:37.00 STC05580 00000090  $HASP100 MFAS ON STCINRDR
15:49:37.06 STC05580 00000090  $HASP373 MFAS  STARTED
15:49:37.06 STC05580 00000090  IEF403I MFAS - STARTED - TIME=15.49.37
15:49:37.23 STC05580 00000090  MFA302I.MFAS.MFARSC00 / ATTACHING VTAM
PROCESSOR
15:49:37.24 STC05580 00000090  MFA303I.MFAS.MFARSC00 / MFA/MVS V3.01 - PTF
2001PTF6 COPYRIGHT (C)
                               1987-2003 MICRO FOCUS INTERNATIONAL LTD.
15:49:37.24 STC05580 00000090  MFA110I.MFAS.MFAFSQ00 / VTAM PROCESSOR ACTIVE
USING VTAMAPPL MFA62ACB
```

Stop

Stops the Mainframe Access task. Mainframe Access automatically stops the Mainframe Access Data Connect server, MFAS, during shutdown. Shutdown may take anywhere from 30 seconds to a few minutes, depending on the level of activity in the system.

Here is an example of a Stop command and the response:

```

16:44:18.52 CSIRLW1  00000290  P MFA
16:44:21.52 STC05579 00000290  P MFAS
16:44:21.53 STC05580 00000090  MFA112I.MFAS.MFAFSQ00 / VTAM PROCESSOR HAS
TERMINATED
16:44:21.68 STC05580 00000090  MFA317I.MFAS.MFARSC00 / HAS BEEN TERMINATED
16:44:21.69 STC05579 00000090  MFMDS005I DSS Server shutdown
16:44:21.70 STC05580 00000090  IEF404I MFAS - ENDED - TIME=16.44.21
16:44:21.70 STC05580 00000090  $HASP395 MFAS  ENDED
16:44:32.77 STC05579 00000090  MFM0011I: Mainframe Access shutdown completed
16:44:32.82 STC05579 00000090  IEF404I MFA - ENDED - TIME=16.44.32
16:44:32.82 STC05579 00000090  $HASP395 MFA  ENDED
```

TRACE ON and TRACE SHORT

Activates detailed tracing of all Mainframe Access activity. Trace output is directed to the destination (SYSPRINT, GTF or CONSOLE) specified by the TRACE parameter during Mainframe Access startup. For more information about these options see the entry for the TRACE parameter in the section *Editing Mainframe Access Parameters* in the chapter *Configuration*. For a tip about viewing trace data sent to SYSPRINT interactively see the *FLUSH XDBOUT*command description

Here is an example of a TRACE ON command and the response:

```

F MFA,TRACE ON
MFM0015I: MFM0025I: Trace set to data level on
```

The TRACE SHORT command activates the same detailed tracing as does the TRACE ON command. The SHORT option limits the display of send data, receive data and storage areas to 256 characters. This can be useful in reducing the amount of trace output, particularly when the message sizes are large.

Here is an example of a TRACE SHORT command and the response:

```
F MFA,TRACE SHORT
MFM0015I: MFM0025I: Trace set to data level on
```

REFRESH ACCESSLIST

Directs Mainframe Access to re-read the access list from the currently active access list data set (the access list identified by the //XDBACC DD statement in the Mainframe Access startup JCL). You can update the access list by adding, removing or updating PERMIT, REJECT and ADMUSER lines, as described in the section *Editing Access List Definitions* in the chapter *Configuration*. When your updates are complete, use REFRESH ACCESSLIST to replace the currently active access list with your updated list without stopping and restarting Mainframe Access.

Here is an example of a REFRESH ACCESSLIST command and the response:

```
F MFA,REFRESH ACCESSLIST
MFM0015I: Refresh Access List has completed
```

DISPLAY LU

Displays information about the target servers which Mainframe Access has successfully connected to by establishing one or more LU6.2 sessions in response to client requests. Target servers are known to Mainframe Access by their SNA LU name or APPLID.

Here is an example of a DISPLAY LU command and the resulting display:

```
F MFA,DISPLAY LU
MFM0015I:  ***************************************************
MFM0015I:  *                   PLU Status                    *
MFM0015I:  ***************************************************
MFM0015I:
MFM0015I:  ******* Partner LU Status
MFM0015I:  LU Name..........LUDB27R
MFM0015I:     Mod Entry........IBMRDB
MFM0015I:     Status...........Active
MFM0015I:  LU Name..........MFA62ACB
MFM0015I:     Mod Entry........#INTER
MFM0015I:     Status...........Active
```

DISPLAY TCPSTATUS

Displays basic TCP/IP configuration information from the Mainframe Access parameters (port number, company name and connection backlog) followed by the information displayed by the *DISPLAY LU* and *DISPLAY CONVERSATION* commands.

Here is an example of a DISPLAY TCPSTATUS command and the resulting display:

```
F MFA,DISPLAY TCPSTATUS
MFM0015I:  ***************************************************
MFM0015I:  *                   TCP Status                    *
MFM0015I:  ***************************************************
MFM0015I:
MFM0015I:  Server Port Number.......1503
MFM0015I:  Web Server Port Number...1504
MFM0015I:  OID......................YOUR_COMPANY_NAME
MFM0015I:  Number of Backlogs.......5
MFM0015I:  ******* Partner LU Status
MFM0015I:  LU Name..........LUDB27R
MFM0015I:     Mod Entry........IBMRDB
MFM0015I:     Status...........Active
MFM0015I:  LU Name..........MFA62ACB
```

```
MFM0015I:     Mod Entry........#INTER
MFM0015I:     Status...........Active
MFM0015I:  ******* Conversation Status
MFM0015I:  Conversation ID..1AA14028
MFM0015I:     Conv Address.....1AC79968
MFM0015I:     Alloc Address....1A928DC0
MFM0015I:     Task Address.....1AA4CE28
MFM0015I:     Conv Status......Active
MFM0015I:     TCP/IP Status....Send
MFM0015I:     User Id..........CSIRLW1
MFM0015I:     Target LU........LUDB27R
MFM0015I:     Peer IP address..10.10.11.130
MFM0015I:     Socket Number....9
MFM0015I:     Start Time.......07/17/03 15:10:32
MFM0015I:     LUW Id...........DDINE.MFM62ACB.B9BBC4F618400001
MFM0015I:     Packets Sent.....14
```

DISPLAY CONVERSATION

Displays detailed information about every active conversation between Mainframe Access and the target servers.

Here is an example of a DISPLAY CONVERSATION command and the resulting display:

```

F MFA,DISPLAY CONVERSATION
MFM0015I:  ***************************************************
MFM0015I:  *               Conversation Status               *
MFM0015I:  ***************************************************
MFM0015I:
MFM0015I:  ******* Conversation Status
MFM0015I:  Conversation ID..1AA14028
MFM0015I:     Conv Address.....1AC79968
MFM0015I:     Alloc Address....1A928DC0
MFM0015I:     Task Address.....1AA4CE28
MFM0015I:     Conv Status......Active
MFM0015I:     TCP/IP Status....Send
MFM0015I:     User Id..........CSIRLW1
MFM0015I:     Target LU........LUDB27R
MFM0015I:     Peer IP address..10.10.11.130
MFM0015I:     Socket Number....9
MFM0015I:     Start Time.......07/17/03 15:10:32
MFM0015I:     LUW Id...........DDINE.MFM62ACB.B9BBC4F618400001
MFM0015I:     Packets Sent.....14
MFM0015I:     Packets Rcv'd....27
MFM0015I:  Conversation ID..1ABFF2E0
MFM0015I:     Conv Address.....1ACD0968
MFM0015I:     Alloc Address....1A928BA0
MFM0015I:     Task Address.....1AA4CA78
MFM0015I:     Conv Status......Active
MFM0015I:     TCP/IP Status....Send
MFM0015I:     User Id..........CSIRLW1
MFM0015I:     Target LU........MFA62ACB
MFM0015I:     Peer IP address..10.10.11.130
MFM0015I:     Socket Number....13
MFM0015I:     Start Time.......07/17/03 15:11:40
MFM0015I:     LUW Id...........DDINET1.MFM62ACB.B9BBC538B9410001
MFM0015I:     Packets Sent.....4
```

C CONV=id

Cancels a specific conversation. Theid parameter is a conversation identifier of eight hexadecimal characters as shown in the information displayed by DISPLAY CONVERSATION and DISPLAY TCPSTATUS commands. Canceling a conversation ends the LU6.2 conversation to the target server and releases the LU6.2 session for use by other conversations. It also disconnects the client by closing the client socket associated with the conversation and cancels the DDF thread if the conversation partner is DB2.

Here is an example of the CANCEL CONVERSATION command that cancels the second conversation shown in the DISPLAY CONVERSATION example:

```
F MFA,C CONV=1ABFF2E0
MFM0015I: Cancel Command accepted
```

FLUSH XDBOUT

Flushes trace data and operational messages to the //XDBOUT DD SYSOUT data set. Operational data is written to XDBOUT during normal operation; trace data is written to XDBOUT only when tracing is switched on and the trace destination is SYSPRINT. The FLUSH command causes any partially filled buffer to be written to the SYSOUT data set, making the output available for browsing. This can be helpful when you are testing and trying to view trace data interactively as the test progresses.

Here is an example of a FLUSH XDBOUT command and the response:

```

F MFA,FLUSH XDBOUT
MFM0015I: Flush XDBOUT completed
```

RESET TIMEOUT=n

Changes the TIME OUT FOR CONNECTION parameter value without you having to stop and restart Mainframe Access. The n parameter is the number of minutes that a client connection can remain idle before the connection will be broken and Mainframe Access resources dedicated to the client will be released. Resources that are released when an idle client is disconnected include allocated storage, TCP/IP resources and LU6.2 conversations to target servers such as DB2, IMS, CICS and Mainframe Access Data Connect. Specify 0 (zero) to disable the client timeout feature and allow unlimited idle time.

If you use the RESET TIMEOUT command to establish a new timeout value, rather than SET TIMEOUT, both the Mainframe Access timeout and any Mainframe Access client timeout remain in effect. The connection partner (Mainframe Access client or Mainframe Access) having the smaller timeout value initiates the disconnection first.

Here is an example of a RESET TIMEOUT command and the response:

```
F MFA,RESET TIMEOUT=30
MFM0015I: Timeout parameter reset to 30
```

SET TIMEOUT=n

Changes the TIME OUT FOR CONNECTION parameter value without you having to stop and restart Mainframe Access. The n parameter is the number of minutes that a client connection can remain idle after which the connection will be broken and Mainframe Access resources dedicated to the client will be released. Resources that are released when an idle client is disconnected include allocated storage, TCP/IP resources and LU6.2 conversations to target servers such as DB2, IMS, CICS and Mainframe Access Data Connect. Specify 0 (zero) to disable the client timeout feature and allow unlimited idle time.

If you use the SET TIMEOUT command to establish a new timeout value, rather than RESET TIMEOUT, then when a Mainframe Access client subsequently starts up, Mainframe Access' connection timeout is disabled, and only the inactivity timeout period specified by the Mainframe Access client is in effect.

Here is an example of a SET TIMEOUT command and the response:

```
F MFA,SET TIMEOUT=20
MFM0015I: Timeout parameter set to 20
```

Mainframe Access z/Server feature commands (deprecated)

Note: These features are deprecated, and provided for backward compatibility only.

Starting and stopping the z/Server feature

To starts the z/Server feature:

```
F MFA,ZSRV:{HSTART|HSTOP}
```

To restart the z/Server feature:

```
F MFA,ZSRV:HRESTART
```

Starting and stopping a z/Server scheduler address space

To start and stop a z/Server scheduler use the following syntax:

```
F MFA,ZSRV:{START|STOP},SCHED=<scheduler>
```

With the keyword START, a scheduler address space is started. It does not need to be defined in the MFAXML configuration member. The character string <*scheduler*> is used as the name of an address space to be started.

The keyword STOP terminates the named scheduler by first terminating all active user servers belonging to that scheduler and then the scheduler task itself.

Stopping a specific user server address space

To stop a specific user server address space, the port that user server listens to must be known. The port can be determined by using the command:

```
F <scheduler>,TSO,DISPLAY,USER=ALL
```

The command response in CMDTASK will have one line with message SLR0036I for every user that is currently using scheduler services. The last column shows the port nnnnn the address space listens on. Then the user server address space can be terminated:

```
F MFA,ZSRV:STOP,PORT=<nnnnn>
```

Stopping all user servers for a Specific user ID

To stop all user servers for a specific user ID use the following syntax:

```
F MFA,ZSRV:STOP,USER=<tsouid>
```

All user servers for that <*tsouid*> are stopped across all schedulers.

Note: The name of the user server address space is prefixed by TSOE_JOBCHAR while this command requires only the TSO user ID to be specified.

Stopping all user servers for a Specific Scheduler address space

To stop all user servers for a specific scheduler address space, use the following syntax:

```
F MFA,ZSRV:STOP,SCHED=<scheduler>,USERONLY
```

All user server address spaces for the named scheduler are terminated.

Display commands for TSO information

To get information about TSO users address the scheduler address space with the keywords TSO and DISPLAY:

```
F <scheduler>,TSO,DISPLAY,{ STATS|VPOOL}
```

The STATS parameter of the DISPLAY command shows statistics of actions done in response to client requests.

The VPOOL parameter of the DISPLAY command shows the variable pool contents.

Display TSO statistics

```
F <scheduler>,TSO,DISPLAY,STATS
```

results in the following output in the CMDTASK DD statement of the scheduler address space:

```
SLR0040I ===============================================
         Server Statistics
         ===============================================
SLR0041I Submit      Count .........             0 times
         Stop        Count .........             7 times
         Communicate Count .........             1 times
         Error       Count .........             0 times
```

The lines in SLR0041I have the following meaning:

Column	Meaning
Submit count	Number of times any user server address space was started.
Stop count	Number of times any user server address space was stopped.
Communicate count	Number of IP messages scheduled to the user server address space for processing (i.e. number of client requests handled in that user server).
Error	Number of errors during processing.

Display VPOOL

```
F <scheduler>,TSO,DISPLAY,VPOOL
```

results in the following output in the CMDTASK DD statement of the scheduler address space:

```
SPR0011I Pool-Address      : 2FB7EF00
         Pool-Size         :         4096
         Pool-Freespace    :         3973 Bytes
         No of Variables   :            5 Bytes
         Variable-Pool-Information
========================== Pool Content =============================
         Var-Typ     Var-Name    Var-Content
=====================================================================
SPR0013I  SYS         CLASS      0
SPR0013I  SYS         JPREFIX    W
SPR0013I  SYS         SCMSPATH   1
SPR0013I  SYS         SCMSTAT    X
SPR0013I  SYS         SCMSREXX   MX0RSC01
```

Diagnostic commands

To get information about special areas in either a scheduler address space or user server address spaces, address the address space concerned with the keyword DUMP:

```
F <scheduler|userserver>,DUMP,{CONFIG|SVCD|AREA=ALL|ADDR=<addr>,LEN=<length>}
```

The CONFIG parameter of the DUMP command enables you to see the current configuration of the address space.

The SVCD parameter of the DUMP produces an SVC dump of the address space titled "zServer SVCDUMP".

Certain or all areas of an address space are dumped using the parameter AREA.

Configuration Dump

```
F <scheduler|userserver>,DUMP,CONFIG
```

writes the current configuration to the CMDTASK DD statement.

SVC Dump

```
F <scheduler|userserver>,DUMP,SVCD
```

writes a standard SVCD dump. Such a dump could also be taken using a regular MVS DUMP command.

Important storage areas

```
F <scheduler|userserver>,DUMP,AREA=ALL
```

writes all information z/Server deems relevant storage areas to DD statement CMDTASK of the address space. These areas include the TAUCA (the main z/Server control block) and control blocks named SUBTASK, one for each of the NUMTCB worker tasks.

Selected storage areas

```
F <scheduler|userserver>,DUMP,ADDR=xxxxxxxx,LEN=yyy
```

writes the storage content starting at address xxxxxxxx in length yyy to DD statement CMDTASK of the address space. Keep in mind that the only areas dumped are those that z/Server owns, in other words, this command cannot be used to show the content of all storage allocated in the address space.

Display data space information

The holder task allocates and manages a common access data space (CADS). The following command displays the information in this data space:

```
F MFA,ZSRV:DISPLAY,DSP,{FORMAT|DUMP}
```

In order to use this command, the holder task must contain a DD statement:

```
//DSPPRT   DD  SYSOUT=*,LRECL=255
```

The output for this display command is written to DSPPRT. If necessary, you might need to scroll to the extreme right to see the full output.

Sub parameter DUMP writes the unformatted storage areas from the data space, when using the keyword FORMAT the information is written in human readable form. Below is an example of a user entry:

```
Userid   JobName  JobId    SYS  Port  L F1 F2 F3 F4 F5 JP  CRP
--------------------------------------------------------------
user1    Vuser1   STCTIMEO syst 01215 M 00 80 80 00 00 00 0006
user2    Vuser2   STC04711 syst 01216 M 00 60 80 00 00 00 0008

Ip-Addr-Client  Ip/Port Scheduler     Creator  ClientId Asid
---------------------------------------------------------------
nnn.nnn.nnn.nnn nnn.nnn.nnn.nnn  1111 TAURISPF 00000003 004C
nnn.nnn.nnn.nnn nnn.nnn.nnn.nnn  1111 TAURISPF 00000006 0037

Start-Time     LastCmd LastCmd-Time   Last Timeout
----------------------------------------------------
mm/dd hh:mm:ss Modify  07/22 hh:mm:ss mm/dd hh:mm:ss
mm/dd hh:mm:ss Modify  07/22 hh:mm:ss mm/dd hh:mm:ss
```

Column JOBID contains a valid JESx job ID when a user server is running and listening on the port specified in column PORT. The (not valid) JESx job ID of STCTIMEO is set either when a user server had terminated (for instance due to inactivity - timeout) or when a client logged on but did not need to start a user server yet. This is called deferred logon. Once the client requests a service that can only run in a user server, that user server is "restarted" using the indicated port.

Setting IPTRACE level

The IPTRACE level is set using the configuration parameter IPTRACE at startup. It is recommended to run at the lowest possible trace level. IPTRACE can be changed dynamically using the following command:

```
F MFA,ZSRV:TRACE,LEVEL=x
F <scheduler|userserver>,TRACE,LEVEL=x
```

where x is a positive integer between 0 and 6. Zero denotes no tracing, 6 is full tracing. Every subsequent trace level includes all lower trace levels.

DD statement CMDTASK (for scheduler and user server) or MAINTASK (for the holder address space) show confirmation:

```
F <scheduler>,TRACE,LEVEL=4
IPC0066I  12:27:46.579 Trace level changed to           4
```

Full tracing in a highly loaded z/Server system can generate a large amount of output data and can lead to JESx spool problems.

Turn Off LE Error Handling

When a problem occurs, z/Server creates a dump in DD statement ZCOMDUMP. When z/Server relies on LE recovery, then LE might write a CEEDUMP. In severe cases an SVC dump is written.

The command

```
F <scheduler|userserver>,RECOVERY,{ON|OFF}
```

can be used to turn of LE error handling. If there is a SYSUDUMP/SYSABEND/SYSMDUMP DD statement in the JCL, such a dump would get written instead of a CEEDUMP.

Note: Switch off recovery only if requested by SupportLine.

REXX commands

When a REXX exec is executing in either a user server or a scheduler address space, the following REXX commands are available, addressing the REXX exec environment (see *SA22-7790-xx: TSO/E REXX Reference*):

```
f <scheduler|userserver>,REXX,{HI|HT|RT|TS},SUBTASK={x|ALL}
```

The abbreviations stand for

- HI = Halt Interpretation
- HT = Halt Typing
- RT = Resume Typing
- TS = Trace Start

Parameter subtask specifies the number of the worker task (T000000x). There are at most NUMTCB worker tasks. Specifying ALL would address all worker tasks simultaneously.

Administrative commands

The following administrative commands are available for day-to-day operations:

```
F <scheduler>,TSO,DELETE,USER=user
F <scheduler|userserver>,SPFDEQ,DSN=<datasetname>[,mem=<member>]
F MFA,ZSRV:TRACE,LEVEL=x
F <scheduler|userserver>,TRACE,LEVEL=x
F <scheduler|userserver>,RECOVERY,{ON|OFF}
F MFA,DISPLAY,DSP,{FORMAT|DUMP}
F <scheduler|userserver>,REFRESH,PGM=nn
```

They are mostly used in error scenarios.

Delete a TSO User in user administration

It is possible that a user server address space terminated without first doing clean-up in the user administration control structures. This can happen if the address space was forced (MVS FORCE command) and no address space related clean-up ran. In this case the command:

```
F <scheduler>,TSO,DELETE,USER=userid
```

enables unconditional removal of the TSO user ID from the user administration control structures to free the entries and the port the user server address space was listening on. The CMDTASK DD statement of the scheduler address space shows:

```
SLR0039W  13:53:19.557 User USERID  deleted.
```

If the specified user is not found in the user administration control structures, the following message is issued:

```
F <scheduler>,TSO,DELETE,USER=ALL
SLR0038W  13:53:08.239 User ALL      not found. Delete command ignored.
```

Note: This command will unconditionally remove the specified user ID from the control structures, even if the client is currently working normally and no problem had occurred before. That means that the client/user server cannot work further without causing other problems. The user server address space also cannot terminate anymore without manual intervention (it must be cancelled):

```
+SLR0092E  14:11:40.508 User servers with Timeout_Action=STOP must be
started with POLICY=1 (Started Jobs). Timeout is ignored.
```

Releasing the SPFEDIT serialization ENQ

When a client starts to edit a member of a partitioned dataset, z/Server locks that member against simultaneous edit from a 3270 TSO user by serializing against the resource SPFEDIT/data_set_name(member_name). That ENQ is held by the command task of the scheduler address space the user server belongs to.

If for some reason the client does not finish the edit process, the member stays locked. In the past another client used the Dequeue function from the context menu for the dataset member to release someone else's serialization. To reduce the integrity exposure of this (client) function, the following command can be used:

```
F <scheduler|userserver>,SPFDEQ,DSN=<datasetname>,MEM=<member>
```

It will release the ENQ that protects against parallel updates of the same dataset / member by different users. The CMDTASK DD statement will contain the following messages acknowledging deletion of the resource:

```
F <scheduler>,SPFDEQ,DSN=DATASET.NAME,MEM=MEMBER
SLR0044I  08:46:21.693 Dequeue for Major-Name SPFEDIT  and Minor-Name
DATASET.NAME                                MEMBER
SLR0046I  08:46:21.694 Dequeue
successful
```

Warning: Only use this command if you have made sure that the original owner will not edit the dataset or member any further. Otherwise you risk the data integrity of your datasets.

Port skip

When a user server address space had allocated a port and does not terminate properly or port ranges were invalidly defined, then it can happen that a port is not available to a newly started user server address space. If that port was the first in the range of the scheduler, this could mean that no user server could get started anymore. With port skip active (PORTCHECK="1" set in the scheduler configuration parameters), z/Server tests the availability of the port. If it is already in use by someone, the message:

```
+SLR0055E  [001] Port [002] is not available.
```

is issued, followed by:

```
+SLR0135S  [001] Port [002] for z/Server scheduler [003] locked because port
was not available.
```

In this context, "port is not available" means that TCPIP returned a response of "the port is in use and has the state LISTEN". There is an entry in the user administration control structures of z/Server for every port defined to the scheduler. The entry that was not available is now marked as LOCKED and will not be used any longer to allocate conversations with a client. This can be displayed by using the command:

```
F MFA,ZSRV:display,dsp,format
```

IN the DD named DSPPRT of the MFA Server task the following output can be found:

```
--------------------------------------
Userid   JobName  JobId    SYS    Port
--------------------------------------
user     jobname  jobid    system nnnnn
```

If a user server has stalled and not terminated and hence still has the port allocated, it needs to get terminated, probably via an operator force arm or force command. That will make the port available for reuse. The port is still in a "locked" state in the user administration part of z/Server, however. It needs to be unlocked manually using the command:

```
F <scheduler>,TSO,UNLOCK,PORT=nnnnn
```

This cleans up the LOCKED state of the port entry, and the port can be used again by the scheduler.

If the port was not allocated to a stalled user server, then the system administrator needs to determine who has the port allocated (for example, using the netstat command). There is probably some sort of definition error in the port ranges for this scheduler/MFA Server holder that needs to get corrected. If ports were defined incorrectly to the scheduler/MFA Server holder, the scheduler/MFA Server holder will need to be restarted with the corrected definitions.

If you only unlock the port in the user administration without first making sure that it really is available to be allocated again, port skip will immediately issue messages SLR0055E and SLR0135S again and relock the port in the user administration control structures.

Refresh a load module

If SupportLine provided a Patch Update that would need a restart of z/Server, and customer support also specified that the load modules affected can be refreshed,

```
F <scheduler|userserver>,REFRESH,PGM=nn
```

with nn a positive integer between 02 and 99 allows reloading the load module and installing the Patch Update without restarting z/Server.

Note: It is not possible to refresh all load modules running in a z/Server address space.

Mainframe Access Messages

Mainframe Access and the z/OS operating system use messages to document both normal and abnormal conditions while Mainframe Access is running. The z/OS system log, the Mainframe Access job log and the Mainframe Access SYSOUT data sets are all possible destinations for important messages about Mainframe Access operation. You should already be familiar with z/OS system messages.

Mainframe Access Server Messages

Some Mainframe Access messages are directed to system consoles as specified by the ROUTE Mainframe Access configuration parameter. Other messages are directed to the job log for the started task or to the XDBOUT file (or both). If you experience problems with Mainframe Access operation, it is important that you review the various message destinations for information that may help you diagnose the problem. This section includes an explanation of each of these messages, ordered by message number.

Mainframe Access Operational Messages Format

Mainframe Access operational messages are of the form:

mm/dd/yy hh:mm:ss.t taskname taskid MFM*nnnna text*

or

mm/dd/yy hh:mm:ss.t taskname taskid MFMDS*nnna-text*

where

mm/dd/yy	indicates the date when the message was issued.
hh:mm:dd.t	indicates the time when the message was issued.
taskname	is the name of the Mainframe Access task that issued the message.
taskid	is a two hexadecimal digit task identifier that helps to uniquely identify the Mainframe Access task for tasks that share a common task name. For instance, task name XDBMFADM processes Mainframe Access Direct client requests and there may be several instances of this task, each managing the work of a different subset of client connections. When examining XDBOUT trace messages for a specific client, trace messages from other clients and their processing tasks may be interleaved with the trace messages for the client that is of interest. The taskid value helps to identify the messages associated with a particular task.
DS	indicates that the message was produced by the Data Set Services (DSS) component of Mainframe Access, which provides all file allocation and record access services.
nnnn	is an integer in the range 0001 through 9999
a	is I if the message is for information only, or E if the message indicates an error condition
text	is the error message text

Messages may be issued that do not have the MFMnnnna or MFMDSnnna message prefix. These messages are diagnostic and/or trace messages that are not documented in this messages section. They are intended for use by Product Support.

Mainframe Access Messages

MFM0001I: Mainframe Access version (service) is active

Issued when Mainframe Access initialization completes. version identifies the Mainframe Access product version and service indicates the current maintenance level that has been applied to Mainframe Access.

MFM0002I: listener socket number is n

Issued during startup by the process that serves client requests arriving on the TCP port, as specified in your Mainframe Access configuration parameter TCP_PORT. This message identifies the socket number n that has been allocated for listening and accepting the client connections.

MFM0003I: getsockopt return code is n

Issued during startup by the process that serves client requests arriving on the TCP port, as specified in your Mainframe Access configuration parameter TCP_PORT. n is the return code from the TCP/IP getsockopt call and a value of zero indicates that startup is proceeding normally.

A non-zero value may indicate a problem. Refer to your TCP/IP provider's socket API return code documentation for an explanation of the non-zero value.

MFM0004I: setsockopt reus return code is n

Issued during startup by the process that serves client requests arriving on the TCP port, as specified in your Mainframe Access configuration parameter TCP_PORT. n is the return code from the TCP/IP setsockopt REUS call and a value of zero indicates that startup is proceeding normally.

A non-zero value may indicate a problem. Refer to your TCP/IP provider's socket API return code documentation for an explanation of the non-zero value.

MFM0005I: setsockopt linger return code is n

Issued during startup by the process that serves client requests arriving on the TCP port, as specified in your Mainframe Access configuration parameter TCP_PORT. n is the return code from the TCP/IP setsockopt call and a value of zero indicates that startup is proceeding normally.

A non-zero value may indicate a problem. Refer to your TCP/IP provider's socket API return code documentation for an explanation of the non-zero value.

MFM0006E: Tcpm socket error, sd=n1, function=name, errno=n2

Issued by the process that serves client requests arriving on the TCP port, as specified in your Mainframe Access configuration parameter TCP_PORT. This process establishes the client connection and then assigns the client to a processing subtask based on the client type (for example, CICS Option client, IMS client, Telnet client). This message indicates that an error has occurred during the processing of a socket API call associated with this connection establishment process. name identifies the specific socket API call, n1 identifies the socket (either the listening socket or a new socket created when a client request is accepted) and n2 is the TCP/IP errno value that was reported for the call.

Refer to your TCP/IP provider's socket API return code documentation for an explanation of the errno value for the specified function.

MFM0007E: Tcpm socket error, sd=n1, function=name, errno=n2

Issued by the process that serves client requests arriving on the TCP port, as specified in your Mainframe Access configuration parameter TCP_PORT. This process establishes the client connection and then assigns the client to a processing subtask based on the client type (for example, CICS Option client, IMS client, Telnet client). This message indicates that an error has occurred during the processing of a socket API call associated with this connection establishment process. name identifies the specific socket API call, n1 identifies the socket (either the listening socket or a new socket created when a client request is accepted) and n2 is the TCP/IP errno value that was reported for the call.

Refer to your TCP/IP provider's socket API return code documentation for an explanation of the errno value for the specified function.

MFM0008E: Parameter initialization has failed

Issued during initialization if parameters processed from the active configuration data set specified by the //XDBIN data definition statement contain errors that make it impossible for Mainframe Access to complete startup. Mainframe Access shuts itself down after issuing this message.

Review the startup messages to identify and correct any configuration errors before restarting Mainframe Access.

MFM0009E: APF authorization has failed

Issued during initialization if Mainframe Access finds that it is not executing as an z/OS authorized program. Mainframe Access must execute as an z/OS authorized program and the z/OS TESTAUTH service is used to verify this during startup. Mainframe Access shuts itself down with a U2199 abend after issuing this message.

Check each program library name in the //STEPLIB data definition statement and make sure that each is identified to z/OS' Authorized Program Facility as an authorized program library. Common mistakes are misspelled data set names and incorrect DASD volume identifications.

MFM0010E: Global data area initialization failed

Issued during initialization if Mainframe Access cannot load required service modules and initialize its global data areas. Mainframe Access shuts itself down with a U2202 abend after issuing this message.

If you see this message contact our Product Support.

MFM0011I: Mainframe Access shutdown completed

This is the last message issued during shutdown to indicate that all processes have been stopped. The Mainframe Access job or started task should end immediately after issuing this message. If the Mainframe Access stop command is issued and you do not see this message within a reasonable amount of time (approximately 30 seconds) there is most likely a problem quiescing one or more of the Mainframe Access processes. This may be due to a Mainframe Access problem or it may be the result of other problems in the system.

If such a condition occurs frequently and no associated system condition can be identified, contact our Product Support for assistance.

MFM0012I: TCP port number is n

Issued during initialization as parameters are processed from the active configuration data set specified by the //XDBIN data definition statement. n is the TCP port number for client connections, as specified in your Mainframe Access configuration parameter TCP_PORT.

MFM0014I: Mainframe Access version (service) for company

Issued during initialization as parameters are processed from the active configuration data set specified by the //XDBIN data definition statement. version identifies the Mainframe Access product version and company is your company name as specified in your Mainframe Access configuration parameter ORGANIZATION.

MFM0015I: message text

Message MFM0015I is a blank template that is used during the creation of Mainframe Access displays in response to administrative commands. message text is a self-explanatory line of display information created by command processing routines. For examples of how this message is used see the sample command output in the chapter Console Commands.

MFM0016E: Tcps socket error, sd=n1, function=name, errno=n2

Issued by the process that serves clients that use DB2 connectivity. This message indicates that an error has occurred during the processing of a socket API call associated with client communication. name identifies the specific socket API call, n1 identifies the client socket and n2 is the TCP/IP errno value that was reported for the call.

Refer to your TCP/IP provider's socket API return code documentation for an explanation of the errno value for the specified function.

MFM0017E Tcps socket error, sd=n1, function=name, errno=n2

Issued by the process that serves clients that use DB2 connectivity. This message indicates that an error has occurred during the processing of a socket API call associated with client communication. name identifies the specific socket API call, n1 identifies the client socket and n2 is the TCP/IP errno value that was reported for the call.

Refer to your TCP/IP provider's socket API return code documentation for an explanation of the errno value for the specified function.

MFM0018I: name-address Task initialization has completed

Issued during startup by Mainframe Access' specialized z/OS processing subtasks as they successfully complete initialization. name identifies the type of subtask and address is the address of Mainframe Access' task control block for the process.

MFM0019I: name-address Task shutdown has completed

Issued during shutdown by Mainframe Access' specialized z/OS processing subtasks as they complete shutdown. name identifies the type of subtask and address is the address of Mainframe Access' task control block for the process. These messages can sometimes help to identify the general nature of a Mainframe Access shutdown delay or failure by identifying the processes that have successfully finished their cleanup procedures.

MFM0020E: Wwwm socket error, sd=n1, function=name, errno=n2

Issued by the process that serves HTTP client requests arriving on the administration port. This process establishes the client connection and then assigns the client to a processing subtask based on the HTTP request (for example, simple resource retrieval of a page or image, or ISPI program request). This message indicates that an error has occurred during the processing of a socket API call associated with this connection establishment process. name identifies the specific socket API call, n1 identifies the socket (either the listening socket or a new socket created when a client request is accepted) and n2 is the TCP/IP errno value that was reported for the call.

Refer to your TCP/IP provider's socket API return code documentation for an explanation of the errno value for the specified function.

MFM0023E: Conversation id returned null for socket blockaddress

Issued by the processes that serve Remote IMS and CICS Option client requests arriving on the TCP port, as specified in your Mainframe Access configuration parameter TCP_PORT. This message indicates that an error occurred when Mainframe Access tried to access the LU6.2 conversation associated with this client's IMS or CICS connection. address identifies the Mainframe Access socket block associated with the requesting client.

If this problem occurs frequently, contact our Product Support for assistance.

MFM0024E: Client data length error, length is 0

Issued by the processes that serve Remote IMS and CICS Option client requests arriving on the TCP port, as specified in your Mainframe Access configuration parameter TCP_PORT. This message indicates that an error occurred when Mainframe Access received a request message from the client.

If this problem occurs frequently, contact our Product Support for assistance.

MFM0025I: Trace set to data level on

Issued in response to the Trace On command to confirm that Mainframe Access activity tracing has been initiated. Trace data is written to the destination specified by the configuration parameter TRACE, or to the default destination, the XDBOUT data set.

MFM0026I: Trace set to data level off

Issued in response to the Trace Off command to confirm that Mainframe Access activity tracing has been stopped.

MFM0027I: Telnet mode on

Issued to confirm that Mainframe Access is accepting Telnet client connection requests.

MFM0028I: Telnet mode off

Issued to confirm that Mainframe Access is not accepting Telnet client connection requests.

MFM0029E: Command is not allowed

Issued to indicate that the previously entered command could not be processed although it was entered correctly.

MFM0030E: name invalid command, please reenter

Issued to indicate that the previously entered command name could not be processed due to an error in the format of the command parameters.

Reenter the command with correct parameters.

MFM0031E: Wtrm task shutdown initiated for unrecoverable error

Issued by the Mainframe Access task that processes operator commands. This message indicates that a severe error occurred during the processing of a command. Commands can originate from the system console (z/OS Modify command) and Telnet clients.

Restart Mainframe Access after shutdown is complete. If this problem occurs frequently, contact our Product Support for assistance.

MFM0032E: Wtrm task is recovered

Issued by the Mainframe Access task that processes operator commands. This message indicates that a severe error occurred during the processing of a command and Mainframe Access was able to resume normal processing. Commands can originate from the system console (z/OS Modify command) and Telnet clients.

If this problem occurs frequently, contact our Product Support for assistance.

MFM0042E: DLL DD name is missing

Issued when a request for an ISPI program is received and Mainframe Access' ISPI program loader cannot find a data definition statement for the ISPI program libraries (the //DLL DD statement). An error response is sent to the requester.

Stop Mainframe Access, add the missing //DLL data definition statements to the Mainframe Access startup JCL and then restart Mainframe Access.

MFM0043E: Open DLL file has failed

Issued when a request for an ISPI program is received and Mainframe Access' ISPI program loader receives an error indication from z/OS when opening the ISPI program libraries identified by the //DLL DD statements. An error response is sent to the requester.

Examine the system log and job log for I/O error messages associated with this failure and use that information to correct the problem. If the problem persists or cannot be identified, contact our Product Support for assistance.

MFM0045E: Unable to allocate LU6.2 conversation with target CICS

Issued by the process that serves CICS Option client requests arriving on the TCP port, as specified in your configuration parameter TCP_PORT. This message indicates that an error has occurred while Mainframe Access was attempting to allocate an LU6.2 conversation to the target CICS region. In some cases the actual error may have occurred during session establishment with the target CICS system and preceding error messages may contain LU6.2 return code information that will be helpful in diagnosing the problem.

Verify that all of the Mainframe Access target server configuration parameters for the CICS region are correct. Also check that the CICS definitions for Mainframe Access are correct and that the VTAM log mode specified in the target server configuration is available to both CICS and Mainframe Access. If this problem occurs frequently, contact Micro Focus Product Support for assistance.

MFM0046E: Function ship error, errno=n, desc=text, data=string

Issued by the process that serves CICS Option client requests arriving on the TCP port, as specified in your configuration parameter TCP_PORT. This message indicates that an error has occurred while Mainframe Access was communicating with the target CICS region. n is an error code assigned by Mainframe Access. text provides a short description of the type of error or the current state of processing. string provides additional error information appropriate for the type of error being reported.

If this problem occurs frequently, contact our Product Support for assistance.

MFM0047I: DSS maximum shared public files is n

Issued during initialization as parameters are processed from the active configuration data set specified by the //XDBIN data definition statement. n is the value specified in the configuration parameter MAX_PUBLIC_FILES. This value indicates the maximum number of shared files that can be concurrently open.

MFM0049I: DLL name scheduled for MFM0049I: DLL name scheduled for refresh

Issued in response to the ISPI module refresh command to confirm that Mainframe Access has accepted the refresh request. A new copy of the requested ISPI load module will be brought into storage using the z/OS LOAD service and used to process subsequent HTTP requests for the program.

MFM0053I: SMF recording is on, record id is n

Issued during initialization as parameters are processed from the active configuration data set specified by the //XDBIN data definition statement. n is the value specified in the configuration parameter SMF_RECORDID. This is a valid user SMF record id and recording is activated.

MFM0054E: SMF recording is off, record id is zero or invalid

Issued during initialization as parameters are processed from the active configuration data set specified by the //XDBIN data definition statement. If the configuration parameter SMF_RECORDID was specified as 0 (zero), you have elected to disable SMF recording. If the configuration parameter SMF_RECORDID was specified as a non-zero value, the value is not valid for SMF user records. In this case, SMF recording is disabled.

If SMF recording is required, stop Mainframe Access, correct the parameter value and then restart Mainframe Access.

MFM0055I: DSS scan interval is n minutes

Issued during initialization as parameters are processed from the active configuration data set specified by the //XDBIN data definition statement. n is the value specified in the configuration parameter DSS_SCAN_INTERVAL.

MFM0056E: DSS scan interval is not valid, defaults to 2 minutes

Issued during initialization as parameters are processed from the active configuration data set specified by the //XDBIN data definition statement. An invalid value was specified for the configuration parameter DSS_SCAN_INTERVAL. The default value of two minutes was substituted for the invalid parameter value.

If two minutes is not an acceptable value, stop Mainframe Access, correct the parameter value and then restart Mainframe Access.

MFM0059E: BLDL for ISPI DLL module name failed

A request to execute ISPI module name was received from an HTTP client. The z/OS BLDL service was unable to locate this load module in the program library concatenation specified by the //DLL data definition statement.

Check that the load module name is correct and that all required program libraries are available.

MFM0060E: LOAD for ISPI DLL module name failed rc=n1 reason=n2

A request to execute ISPI module name was received from an HTTP client. The z/OS LOAD service reported a failure when reading the load module from the program library concatenation specified by the //DLL data definition statement. Return code n1 is the z/OS system completion code that would normally be reported as an abend failure by the z/OS LOAD service and n2 is the additional reason code associated with the abend code. Both values are given as hexadecimal numbers.

Refer to the z/OS system codes manual for your level of z/OS to interpret the return code/reason code pair and take appropriate corrective action.

MFM0061E: ISPI DLL module name unavailable due to rc=n1 reason=n2

A request to execute ISPI module name was received from an HTTP client. This ISPI module has been logically disabled by Mainframe Access due to a previous error condition associated with the module. The previous error may have been an z/OS LOAD service failure or a module execution failure. Return code n1 is the z/OS completion code that was reported for the original failure and n2 is the additional specific reason code associated with the completion code. Both values are given as hexadecimal numbers.

If the completion code value is of the form 00xxx000, xxx is an z/OS system completion code and you should refer to the z/OS system codes manual for your level of z/OS to interpret the return code/reason code pair and take appropriate corrective action. If the completion code value is of the form 00000xxx, the xxx value is a hexadecimal user abend code. User abend codes are normally documented as decimal values and you should convert the xxx hexadecimal value to decimal, if necessary. Refer to the program documentation for the meaning of the user abend code.

MFM0062E: Initialization failed (by name - explanation)

During initialization processing a condition was encountered that would prevent proper execution of Mainframe Access. For example, Mainframe Access' listening process for client connections was unable to allocate and/or open a socket for TCP_PORT. The component that detected the condition has initiated shutdown processing. The name identifies the task that detected the condition and explanation provides a concise statement of the problem.

Examine associated error messages in the system log, joblog and XDBOUT data set to determine the cause of the initialization failure. In most cases, the related error messages can be found just before this message in the XDBLOG data set. Correct the problem and restart Mainframe Access.

MFM0063I: MFA Direct is active

This message is issued during initialization to acknowledge that MFADIRECT=YES was specified in the parameter file.

MFM0064I: MFA Direct number of processing tasks is n

This message is issued during initialization to acknowledge that the MFADIRECT_MAXTASKS=n parameter file specification has been processed. Mainframe Access will start n z/OS tasks for Mainframe Access request processing. During execution

Mainframe Access will create up to n additional tasks for Mainframe Access request processing if the transaction rate and processing load warrant.

MFM0065I: usage port number for client connections is n

This message is issued during initialization to acknowledge that the TCP_PORT parameter file specifications have been processed. The usage indicates that the "Enterprise Server" (for TCP_PORT) will be listening on port number n for client connections.

MFM0066I: Address space type is astype

This message is issued during initialization to acknowledge that the ASTYPE parameter file specification has been processed. The address space type can be "STANDARD", "CTLRGN" or "APPSERVER". The "STANDARD" address space is a stand-alone Mainframe Access server. The "CTLRGN" is a Mainframe Access server capable of starting application server regions such as the Mainframe Access Data Connect server. The "APPSERVER" address space is lightweight subset of the Mainframe Access server that is started and managed by a CTLRGN.

MFM0068I: parmname parameter specification invalid

This message is issued during initialization to indicate that the usage of parameter parmname is incorrect and this parameter was not successfully processed.

Correct the parameter specification and restart Mainframe Access.

MFM0069E: MFADirect not configured - rejecting sd=n ip=addr

Issued by the TCP/IP listener for TCP_PORT when a client connection request for Mainframe Access services is received and MFADirect services have not been configured in the initialization parameters. The client (from IP address addr on socket n) is disconnected.

Check that MFADirect services are enabled by the MFADIRECT and MFADIRECT_MAXTASKS initialization parameter settings. Restart Mainframe Access if parameter settings are changed.

MFM0070E: MFA Data Connect not configured - rejecting sd=n ip=addr

Issued by the TCP/IP listener for TCP_PORT when a client connection request for Mainframe Access Data Connect services is received and the Mainframe Access Data Connect server has not been configured in the initialization parameters. The client (from IP address addr on socket n) is disconnected.

Check that the Mainframe Access Data Connect server is enabled by the Mainframe Access FEATURE, MFALINK_MAXTASKS and ASTYPE initialization parameter settings. Restart Mainframe Access if parameter settings are changed.

MFM0071E: process access list check condition client ipaddr

This message may be issued by the TCP/IP port listeners when access list checking is active. If process is XDBTCPM, this is a connection attempt through the TCP_PORT for Micro Focus clients. Each client's IP address (ipaddr) is checked against the PERMIT and REJECT specifications in the access list. If the condition reported is "rejected", the client IP address was found to match a reject specification in the access list. In this case the client's connection request is immediately rejected. If the condition reported is "warning for", no matching reject or accept specification was found in the access list. In this case the client connection is accepted. You may wish to add an access list specification for this client or an appropriate range of similar client IP addresses.

MFM0072E: Shutdown starting due to failure (in taskname)

This message is issued when one of the Mainframe Access tasks encounters a software failure and ESTAEX processing is unable to fully recover and guarantee the integrity of subsequent processing in the address space. In this case, the ESTAEX recovery routine processing will start an automatic shutdown of Mainframe Access.

Examine the XDBOUT data set, joblog and system log for associated error messages. This may identify a problem that requires you to take a corrective action. Restart Mainframe Access and report the incident to Product Support.

MFM0074E: The following parameter statement is rejected (line n)

This message is issued during startup when an invalid parameter is found in the Mainframe Access configuration parameter file. The statement containing the invalid parameter is displayed in the message line that follows.

Mainframe Access Server completes processing of the parameter file and then shuts down due to the parameter error. Correct the invalid parameter and restart Mainframe Access Server.

MFM0077E: CG production log (ddname) open failed, errno=n _msgno=m

This message is issued during startup when the data set specified by Mainframe Access configuration parameter CGMQOTMA LOGDD cannot be OPENed.

MFM0079I: CG production log closed

This message is issued during shutdown to confirm that the data set specified by the Mainframe Access configuration parameter CGMQOTMA LOGDD has been successfully closed.

MFM0080I: MQ message queue configuration information

MFM0083I: Cross-memory initialization starting (dac=address)

This message is issued during startup when dependent address spaces are configured. Mainframe Access is preparing to create the z/OS cross-memory resources that allow Mainframe Access address spaces to communicate using Program Call. The control region is first to initialize and it establishes the cross-memory environment that allows dependent regions to call back into the control region. As dependent regions initialize (either at startup or on-demand) each dependent creates a cross-memory environment that allows the control region to call into the dependent region. This creates a bi-directional calling mechanism between the control region and each dependent region. The DAC is the primary control table for dependent address space services.

MFM0084I: Cross-memory initialization complete

This message is issued during startup when Mainframe Access has completed the setup of z/OS cross-memory resources that allow Mainframe Access address spaces to communicate using Program Call.

MFM0085I: > Start_Program sent to [rcvd from] stcname (seq=n grp=cccc pgm=cccccccc uid=cccccccc)

This message is issued when Program Call is used to pass a Start Program request to a dependent address space. The control region issues the "sent" form of the message and the dependent region issues the "rcvd" form of the message. seq= is the request sequence number used to correlate the cross-memory activity grp= is the ASGROUP name for the dependent region pgm= is the name of the program to be called uid= is the user id that will provide the program execution security environment

MFM0086I: > Program_Complete sent to [rcvd from] stcname (seq=n ServerRC=n ProgramRC=xxxxxxxx)

This message is issued when Program Call is used to pass a Program Complete indication to the control region. The Program Complete indication returns the result of a Program Call request that was passed to the dependent region. seq= is the request sequence number used to correlate the cross-memory activity ServerRC= is the Mainframe Access return code 0 = the requested program was called and executed succesfully (ProgramRC contains the return code) 1 = the requested program abended (ProgramRC contains the ABEND code) 2 = the dependent region could not load the requested program module 3 = the dependent region Mainframe Access load and call service failed 4 = the call request was not successfully passed to a dependent processing region 5 = the user credentials (userid and password) were invalid ProgramRC= is the return code value from the called program (if the ServerRC is zero)

MFM0087I: Cross-memory environment created

This message is issued during startup to confirm that the z/OS cross-memory Program Call infrastructure was successfully created in this address space.

MFM0088I: Asparm extracted (length=n)

This message is issued during startup of a dependent region to confirm that information needed to establish cross-memory communication was successfully passed from the control region to this dependent region. The asparm data is displayed in dump format following this message.

MFM0089E: Cross-memory start failed (Da_Start rc=n rs=xxxxxxxx)

This message is issued during startup if Mainframe Access is unable to create the z/OS cross-memory Program Call infrastructure.

Mainframe Access ABENDs with code U3100. Contact product support if you cannot resolve the reason for the initialization failure by examining related messages, etc.

MFM0090I: Waiting for server task to initialize

This message is issued during startup when the dependent region has received a request and the required processing task has not yet complete initialization.

This message is issued repeatedly (every second) until the server task completes initialization and is ready to accept the request.

MFM0091I: < Appserver_Started received from stcname (dab=xxxxxxxx)

This message is issued by the control region when a started dependent region has completed its cross-memory initialization and issued a confirming Program Call back to the control region. The dependent's call provides the cross-memory information that the control region needs to Program Call into the dependent region.

MFM0092I: Appserver stcname connected (dab=xxxxxxxx)

This message is issued by the control region when a started dependent region has been successfully contacted by the control region using the dependent's cross-memory Program Call information.

MFM0093E: Appserver stcname connection failed (Da_Conn rc=n rs=xxxxxxxx)

This message is issued by the control region when a started dependent region could not be contacted by the control region using the dependent's cross-memory Program Call information.

Contact product support if this problem persists.

MFM0094I: < Ctlrgn_Acknowledgement received from stcname

This message is issued by the dependent region when the control region contacts the dependent by using the cross-memory Program Call information provided by the dependent.

MFM0095I: < Start_Shutdown received from stcname

This message is issued by the dependent region when the control region uses Program Call to start shutdown processing in the dependent region.

MFM0096I: Cross-memory cleanup complete

This message is issued during shutdown to confirm that all of the z/OS cross-memory resources allocated by this address space have been successfully released.

MFM0097E: Cross-memory cleanup failed (DA_Stop rc=n rs=xxxxxxxx)

This message is issued during shutdown to report that some z/OS cross-memory resources allocated by this address space were not successfully released.

Contact product support if this problem persists.

MFM0098I: Appserver stcname is being shutdown (dab=xxxxxxxx)

This message is issued during shutdown by the control region when an active dependent region is selected to receive a Start_Shutdown Program Call. After the call the PC linkage for the dependent is removed from the control region.

This message is also issued when a dependent region is being shut down in response to a request processing ABEND in that region. When a processing ABEND occurs the control region will shut down the associated dependent region because that address space is considered unstable. For example, storage contents may be corrupted, excessive storage allocations may be accumulating, etc. A new instance of a dependent region will be automatically started when one is needed for subsequent request processing.

MFM0099I: Appserver stcname disconnected (dab=xxxxxxxx)

This message is issued during shutdown by the control region to confirm that the named dependent region has received the Start_Shutdown Program Call and its PC linkage has been removed from the control region.

This message is also issued to confirm the shutdown of an dependent region that has become unstable due to a previous processing ABEND.

MFM0100E: Appserver stcname disconnect failed (Da_Ashut rc=n rs=xxxxxxxx)

This message is issued during shutdown by the control region when the cross-memory linkage to the named dependent region cannot be successfully removed.

Contact product support if this problem persists.

MFM0101I: groupname address space group successfully created (dag=xxxxxxxx)

This message is issued during startup by the control region when the definition for an address space group has been successfully processed. This message confirms that the control structures for managing the group's address spaces have been successfully created. Message MFM0102I follows and displays the parameter values for the group. Address space group definitions are read from the Mainframe Access SERVERS file during initialization.

This message does not confirm that all values in the address space group definition are valid and usable. For example, the JCL procedure name in the definition may not exist in the proclib or the JCL procedure may have syntactical errors that will prevent it from starting.

MFM0102I: groupname - Proc(name) STCprefix(cccc) GroupID(n) Min(n) Max(n) Server(name)

This message is issued during startup by the control region and follows message MFM0101I for a successful address space group creation. This message displays the parameter values from the SERVERS definition for the group.

MFM0103I: groupname is the default address space group

This message is issued during startup by the control region when an address space group definition with GroupID=1 is processed. GroupID=1 specifies that this is the definition for a default group that will be selected if the group name or group id in a request cannot be found in an active address space group definition.

MFM0104I: No default address space group has been defined

This message is issued during startup by the control region when all address space group definitions have been processed and no default group was defined. GroupID=1 is used in a group definition to indicate the default group. There is no requirement to define a default group and system operation can proceed normally without a default group definition. Processing requests that do not specify a valid active group name or group id will be rejected.

MFM0105I: Address space stcname started for group groupname (dab=xxxxxxxx)

This message is issued by the control region when a new dependent address space is started. The stcname is the started task name assigned to the job. The groupname identifies the address space group for which this address space was started.

MFM0106E: Address space for groupname could not be started (Da_Ascre rc=n rs=xxxxxxxx)

This message is issued by the control region when a new dependent address space cannot be started. The return code and reason code from the address space creation routine are reported in the message.

The most common reason, indicated by rc=5, indicates that the maximum number of address spaces defined for the group has been reached. When rc=5 is reported, the request that initiated the address space creation must be queued to wait for one of the existing address spaces to become available.

If rc=5 is frequently reported for your system you should increase the MAXIMUM value for the address space group by 1. This allows an additional address space to be started and reduces the queueing delays for client requests. Contact product support if the problem persists.

MFM0107E: Definition failed for group groupname (Da_Mkgrp rc=n rs=xxxxxxxx)

This message is issued during startup by the control region when the control structures for an address space group definition cannot be successfully created.

Mainframe Access ABENDs with code U3100. Contact product support if you cannot resolve the reason for the initialization failure by examining related messages, etc.

MFM0108I: > Start_Tran sent[rcvd] to groupname (seq=n pri=n len=n1+n2 grp=cccc src=n.n.n.n)

This message is issued when Program Call is used to pass a Start Transaction request to a dependent address space. The control region issues the "sent" form of the message and the dependent region issues the "rcvd" form of the message. seq= is the request sequence number used to correlate the cross-memory activity pri= is the priority assigned to the transaction by the originator len= shows the header length (n1) and the content length (n2) grp= is the ASGROUP name for the dependent region src= is the IP address of the originator

MFM0109I: > Tran_Complete sent to [rcvd from] groupname (seq=n ServerRC=n ProgramRC=xxxxxxxx)

This message is issued when Program Call is used to pass a Transaction Complete indication to the control region. The Transaction Complete indication returns the result of a Start Transaction request that was passed to the dependent region. seq= is the request sequence number used to correlate the cross-memory activity

ServerRC is the Mainframe Access return code 0 -> the requested transaction was called and executed succesfully (ProgramRC contains the transaction return code) 1 -> the requested transaction abended (ProgramRC contains the ABEND code) 2 -> the client socket connection could not be obtained by the dependent region

ProgramRC is the return code value from the called transaction program (if the ServerRC is zero)

MFM0110I: Request queued for group groupname (qcount=n dapc=xxxxxxxx rc=n rs=xxxxxxxx)

This message is issued when a request (Start Program or Start Transaction) is received and no dependent address space is immediately available to process the request. The request is queued to the address space group to wait for an available dependent address space. If the maximum number of address spaces for the group have not been started, a new dependent region will be started to process this request. qcount= is the current number of requests waiting in the queue for this address space group dapc= is the address of the control structure that represents the unit of work rc= and rs= are the return and reason code values from the Da_Qwork service .

If this message occurs frequently, your system may perform better if you update the servers definition file for the indicated group. Increasing the MAXIMUM value of the group by 1 will provide an additional address space for processing client requests. This should reduce the frequency of request queueing which should result in improved response time for requests that would otherwise be queued.

MFM0111I: SAF logon for userid (sd=n (xxxxxxxx) rc=n type=n aaausr=xxxxxxxx acee=xxxxxxxx)

This message is issued when a user at a Mainframe Access client successfully authenticates by providing a userid and password that are accepted by the z/OS SAF authentication routines. Security resources, including a z/OS SAF ACEE control block, are obtained to create a security environment for this user.

MFM0112I: SAF logoff for userid (sd=n (xxxxxxxx) rc=n type=n)

This message is issued when a previously authenticated user at a Mainframe Access client changes user ids or disconnects from the server. This message confirms that the z/OS SAF ACEE and related security resources have been released.

MFM0113E: SAF error for userid (sd=n (xxxxxxxx) SafRC=n RacfRC=n RacfRS=n

This message is issued when the userid and password submitted by a Mainframe Access client are rejected by z/OS SAF authentication routines. The specific SAF and RACF return and reason codes are reported in the message. The SAF and RACF return and reason code definitions are summarized elsewhere in this seciton.

MFM0114E: Timeout for sd=n n.n.n.n at mm/dd/yy hh:mm:ss, last time: mm/dd/yy hh:mm:ss

This message is issued when a Mainframe Access client is idle for the number of minutes specified by the TIME OUT FOR CONNECTION parameter. The idle client is disconnected and associated resources are released. sd= is the socket number of the client connection n.n.n.n is the IP address of the client mm/dd/yy hh:mm:ss is the date and time when the time out occurred The "last time" is the date and time when the last client TCP/IP activity occurred.

MFM0115E: Access list I/O task initialization has failed

This message is issued during initialization if the access list file (DDNAME XDBACC) is not usable.

Disregard this error message for dependent regions. Dependent regions do not use an access list data set. If the error message is for the control region, examine related Mainframe Access and system message to determine the cause of the error. Contact product support if the problem persists.

MFM0116E: Access list DDname is missing

This message is issued during initialization if a data definition statement for DDNAME XDBACC is not found in the started task JCL.

Disregard this error message for dependent regions. Dependent regions do not use an access list data set. If the error message is for the control region, correct the started task JCL for the Mainframe Access control region and restart Mainframe Access Server. Contact product support if the problem persists.

MFM0117I: Mainframe Access ParmBlock is located at xxxxxxxx

This message is issued during initialization to display the address of the Mainframe Access Server primary control table, the ParmBlock.

MFM0118I: type [parameter definition statement]

This message is issued during initialization to display the active parameter definitions when LIST PARMS=YES is specified in the Mainframe Access startup parameters. All statements from the XDBIN parameter file (type "Parms") and the SERVERS definition file (type "Srvrs") are echoed to the XDBOUT sysout data set.

MFM0119I: Mainframe Access jobname is cccccccc

This message is issued during initialization to display the jobname assigned to this Mainframe Access address space.

MFM0120E: Servers definition for MCOID=mcoid not found sd=n (xxxxxxxx)

This message indicates that an error has occurred while Mainframe Access was processing the initial client request to bind a target CICS region to the current socket connection. The mcoid contained in the client request did not match any of the defined CICS servers and no default CICS server was defined.

Verify that the MCOID is specified correctly in the client configuration. Also verify that the Mainframe Access Server definition for that target CICS system is correct. The "Connect id" in the CICS Resource Definition should match the MCOID in the Mainframe Access Server definition. You may also create a default CICS server definition that is to be used when the client-specified MCOID does not exist. If the definitions are correct and this problem persists, contact Micro Focus Product Support for assistance

MFM0121E: MFA startup error: no DSS environment

When a transaction arrives at an Endevor Dependant Region, MFA will attempt to validate that Global Storage has been assigned for Endevor use. This messages indicates that MFA initialization has failed, and no global work areas have been pre-assigned for Endevor.

MFM0122E: Endevor setup error: No MSGLOG established

When a transaction arrives at an Endevor Dependant Region, MFA will attempt to validate that an Endevor transaction message log has been successfully allocated to this job. This is either dynamically created as jobname.CR_jobname.MFA.ENDEVOR.MSGLOG or pre-defined by the site within the DR started task JCL:

```
  //ENDVMSG  DD DISP=NEW,DSN=&&MSGLOG,UNIT=VIO,
//            DCB=(RECFM=FBA,LRECL=133,BLKSIZE=13300),
  //            SPACE=(13300,(10,10))
```

An Endevor transaction cannot be processed without a MSGLOG file.

MFM0123E: Endevor version mismatch between CR and DR

When a transaction arrives at an Endevor Dependant Region, MFA will attempt to validate that the same Endevor API version was used in both address spaces. This is necessary because the Endevor API is not downward compatible. The transaction is aborted if the version does not match. The STEPLIB concatenation must specify the same Endevor load library version in both address spaces.

MFM0124E: No Endevor API modules found

When a transaction arrives at an Endevor Dependant Region, MFA will attempt to load the module ENA$NDVR. This the entry point of the Endevor API. The transaction is aborted if the Endevor API cannot be located.

MFM0125E: name-address Task shutdown forced

Issued during shutdown when one of the Mainframe Access subtasks does end properly. name identifies the type of subtask and address is the address of Mainframe Access' task control block for the process.

This message is normal when the associated task has had a processing failure prior to shutdown. Contact product support if this message appears frequently.

MFM0126E: Tasks (n) forced during shutdown forced

Issued during shutdown when one or more of the Mainframe Access subtasks does not end properly (see message MFM0125E).

This message is normal when the associated tasks have had processing failures prior to shutdown. Contact product support if this message appears frequently.

MFM0128I MCG subtask MCGEXEC cc xxxxxxxx started for XDBMFADM cc xxxxxxxx

This is a normal operational message to log the fact that a new subtask was started for Mainframe Call Generator program execution.

MFM0128E MCG Non-zero return from IGZERRE set call; rc=ddd

The IBM IGZERRE module was called to establish a COBOL run time environment but has indicated there is a problem by returning with a non-zero return code. See the section IGZERRE Entry Conditions and Return Codes in the appendix Mainframe Call Generator for information on the return code values.

MFM0128E MCG Module IGZERRE is not available for set call

This message indicates that the IBM IGZERRE module was not loaded during startup and could therefore not be called to establish a COBOL run time environment.

MFM0128E MCG Non-zero return from IGZERRE reset call; rc=ddd

The IBM IGZERRE module was called to terminate a COBOL runtime environment during Mainframe Call Generator cleanup but has indicated there is a problem by returning with a non-zero return code. See the section IGZERRE Entry Conditions and Return Codes in the appendix Mainframe Call Generator for information on the return code values.

MFM0128E MCG No IGZERRE COBOL environment found for reset

The Mainframe Call Generator cleanup service found that no COBOL environment was previously established for the Mainframe Call Generator session that is ending.

MFM0129E MCG Unable to execute user program cccccccc

An execute request could not be completed for the named program. The preceding error messages will identify the exact cause.

MFM0129E MCG Unable to find or load user program cccccccc

The z/OS BLDL or LOAD operation for the named program failed. The preceding error messages will identify the exact cause.

MFM0129E MCG Request message sequence error

An unexpected Mainframe Call Generator request message was received from the client. The Mainframe Call Generator request is not appropriate for the current state of the remote execution operation. Start the Mainframe Access Server trace facility and recreate the problem to create a log of messages showing the sequence of messages being exchanged.

MFM0129E MCG Request message not recognized

An unexpected request message was received from the client. The request message could not be identified as a Mainframe Call Generator request. Start the Mainframe Access Server trace facility and recreate the problem to log the erroneous message.

MFM0130E MCG cccccccc ABENDed, System=Sxxxx Reason=xxxxxxxx User=Udddd

This message is issued after the Mainframe Call Generator ESTAE extension has trapped and recovered from an ABEND condition. The message provides the system (hexadecimal) or user (decimal) ABEND code. System reason codes are also logged for system ABENDs. If a SYSUDUMP (or SYSMDUMP) DD statement is available to the server, a dump is recorded for the ABEND.

MFM0130E MCG unable to call cccccccc, iSvcRC=dddd iSvcRS=dddd

This message is issued when server preparation for the Mainframe Call Generator call operation encounters an internal logic error. The MD_MCGEXEC_RUN service was called to execute the user program but this service ended with an error indicated by the return code and reason code values. The preceding error messages identify the exact cause.

MFM0130E MCG cleanup failed, iSvcRC=dddd iSvcRS=dddd

This message is issued if the MD_MCGEXEC_CLEANUP service encounters an internal logic error. The preceding error messages identify the exact cause.

MFM0131E Task cccccccc xxxxxxxx TCB xxxxxxxx terminated Udddd; callrtm rc=ddd

This message is issued when a subtask is forcibly terminated by Mainframe Access Server. The Udddd user ABEND code will be U201 for forced terminations of a MCGEXEC subtask. This could occur when the end user logs off or terminates the client while a Mainframe Call Generator session is active.

Data Set Services Messages

MFMDS001I DSS Server Ready

Mainframe Access has successfully initialized the DSS component. Any pre-allocated item libraries are now ready for use.

MFMDS002I DSS Initialization failed

Mainframe Access failed to initialize the DSS component. The services normally provided by DSS are therefore not available.

MFMDS003I Cell pool build failed

The DSS component was unable to acquire enough virtual memory to build a pool of reusable resources.

Adjust the configuration or memory allocation for the Mainframe Access started task.

MFMDS004I Public File non-VSAM

The DSS component was unable to initialize a shared public file because the file specified was not a VSAM key-sequenced data set. DSS discards the request and continues processing. All item libraries are shared public files.

Ensure that all item libraries are created as VSAM key-sequenced data sets.

MFMDS005I DSS Server shutdown

The DSS server has been successfully shut down as part of the Mainframe Access shutdown procedure.

MFMDS006I DSS Server unidentified request

The DSS server detected an invalid request. DSS discards the request and continues processing.

If you see this message contact Micro Focus SupportLine.

MFMDS007I

MFMDS008I Online ITEM LIBRARY=data-set-name

The item library named is now online and available.

MFMDS009I Offline ITEM LIBRARY=data-set-name

The item library named has been taken offline and is no longer available.

Adjust the configuration or memory allocation for the Mainframe Access started task.

Ensure that all item libraries are created as VSAM key-sequenced data sets.

If you see this message contact Micro Focus SupportLine.

MFMDS010I DAIR reas/info DSN=data-set-name

DSS encountered a dynamic allocation error while attempting to allocate the specified data set. Here are some possible reason and information codes:

Reason	Info	Description
02xx		Environmental errors: memory or authorization
0352	0002	Invalid data set name
	0003	Invalid member name
	0004	Invalid disposition
1708	0002	Unable to locate data set name in catalog

All reason codes and information codes are in hexadecimal notation. For a full list of errors and further information on z/OS dynamic allocation errors, see Chapter 26 of the IBM manual MVS Authorized Assembler Services Guide.

MFMDS011I TIOT ENQ INTERFERENCE

A dynamic allocation request has been aborted because another system service request within this task has already locked access to the TIOT (Task Input/Output Table). The request will be re-tried after a 50 millisecond delay. For further information on OS/390 Dynamic Allocation errors, refer to Chapter 26 of the IBM publication (GC27-1763) MVS Authorized System Services Guide.

MFMDS012I M=member ENQ FAILURE RC=nnn

A member of a PDS was to be locked for exclusive access by this operation. However, the PDS member could not be locked at this time due to competion from another user. This request has already been retried several times over a three second period, and now the transaction is being aborted due to BLOCKED status. For further information on OS/390 ENQ errors, refer to the IBM publication MVS Assembler Services Reference (GC28-1910).

RC	Description ENQ RET=USE
04	Resource not available
08	Task already has control of this resource
12	Task has previously failed to acquire this resource

MFMDS013E OPEN FAILED FOR SNAP | READER | SYSOUT

A transient file was not opened properly for subsequent use by MFA resource cleanup (SNAP) or job submission (READER) or logging (SYSOUT). The request is aborted. Additional IBM messages may have been displayed on the SYSLOG to explain the nature of the failure. The MFA Direct server continues to run with degraded services.

MFMDS020E Invalid z/OS file format

The file named in the associated message MFMDS031 has a file format that is not currently supported by MFA. This includes such formats as ISAM, HFS files, direct-access keyed files, and various spanned record formats.

MFMDS021E LRECL too high

DSS detected an attempt to write a logical record which exceeded the maximum logical record size for the file named in the associated message MFMDS031.

MFMDS022E Invalid file handle

DSS detected an invalid parameter list passed by a calling program.

If you see this message contact Micro Focus SupportLine.

MFMDS024E Invalid file name

DSS detected an attempt to create an invalid filename.

If you see this message contact Micro Focus SupportLine.

MFMDS026E LRECL EXCEEDS DEFINITION FOR DSN=dsname

During an update operation, a logical record was too large to be written to the selected file. The operation was aborted.

MFMDS029E RECALL FAILURE=nnnnn FOR DSN=dsname

An internal error occurred during an attempt to recall a file from IBM's HSM (Hierarchical Storage Manager). See the product documentation DFSMShsm Managing Your Own Data (SC35-0420) for details.

RC	Description
100	DFSMShsm not available
400	Invalid request
402	Dataset not found
403	Invalid dataset name
806	Service module not found

MFMDS030E STOW error RC=nn Reason=nnnn M=member

DSS encountered an error while writing to the PDS directory for the file named in the associated message MFMDS031. The first eight characters of the member name are displayed. Here are some possible return code (RC) and reason values:

RC	Reason	Description
4	0	Member name specified already exists
8	0	Member name specified could not be found
12	0	No space remains in the directory
16	01	Permanent physical error
	02	Unable to add EOF marker to directory
	04	Unable to flush system buffers
	3383	SD37 failure: no secondary space available
	3639	SE37 failure: no secondary space available

For a full list of return codes and reasons, see the IBM manual Macro Instructions

MFMDS031I DSN=data-set-name

This information message may accompany other system messages to indicate which data set is affected. The data set name appears as a string of up to 44 characters.

MFMDS032E FIND error RC=nn Reason=nnnM=member

DSS encountered an error while locating a member in the PDS directory for the file named in the associated message MFMDS031. The first eight characters of the member name are displayed. Here are some possible return code (RC) and reason values:

RC	Reason	Description
4	0	Member name specified could not be found
	4	Insufficient RACF authority for this request
	8	Share options not granted on this PDSE
	12	This PDSE already open for output on another member
8	0	Permanent physical error
	4	Insufficient virtual storage available
	8	Invalid DEB (data extent block) or data set owned by another task
	12	Physical error on buffer flush
	16	Invalid DCB

MFMDS033E BLDL error RC=nnReason=nnn M=member

DSS encountered an error while building directory information from the PDS directory for the file named in the associated message MFMDS031. The first eight characters of the member name are displayed. Here are some possible return code (RC) and reason values:

RC	Reason	Description
4	8	Member name specified could not be found
8	0	Permanent physical error

RC	Reason	Description
	4	Insufficient virtual storage available
	8	8 Invalid DEB (data extent block) or data set owned by another task

For a full list of return codes and reasons, see the IBM manual Macro Instructions for Data Sets.

MFMDS034E Directory error RC=nn Reason=nnnn

DSS encountered an error while trying to identify the PDS member for the file named in the associated message MFMDS031. Here are some possible return code (RC) and reason values:

RC	Reason	Description
8	1012	No members found in directory
12	1041	Invalid parameter list
	1054	Invalid DEB (data extent block)
	1057	PDS is not open
	1058	Invalid DCB

For a full list of return codes and reasons, see the topic DESERV (Directory Entry Services) in the IBM manual Macro Instructions for Data Sets.

MFMDS035E QSAM abend Sxxx-yy

DSS has detected a QSAM failure and has directed QSAM to ignore the failure. The failure occurred while DSS was processing the file named in the associated message MFMDS031. Sxxx is the abend or system completion code and yy is the reason code. These codes are presented in hexadecimal format. Possible failures include:

Abend	Reason	Description
C37	4	Invalid extents found on end-of-volume
D37	4	No secondary space specified on this file, and the primary space has been entirely used. Reallocate the file to allow secondary extents or a larger primary extent.
E37	4	No secondary space was available on this volume, or the number of extents has already reached 16 for this file. Reallocate the file to use larger secondary extents or move the file to a different volume where space is available.

MFMDS040E VSAM error DD=ddname RC=nn FB=nnn REQ=verb

DSS has detected a VSAM error probably associated with an input/output operation on an item library. Possible values for verb include:

Verb	Description
GET	Read a record
PUT	Write a record
CHECK	Verify completion of an event
POINT	Position to a specific location
ENDREQ	Cancel an operation
ERASE	Erase a record

Common return code (RC) values and feedback code (FB) values include:

RC	FB	Description
8	8	Attempt to store a duplicate key
	12	Conflicting options specified violate stored sequence
	16	Record not found
	20	Physical locking conflict
	24	Volume not available
	28	Insufficient space on volume to extend the
	32	Invalid relative byte address specified
	36	Key range specified in definition excludes this key
	40	Insufficient virtual storage in the Mainframe Access address space
	48	Conflicting options on VSAM request
	64	No virtual storage to handle this level of concurrency
	68	File not open for type of processing requested
	72	Keyed request made to an ESDS
	88	Invalid switch to sequential processing without positioning
	92	Invalid replace or erase without key
	96	Attempt to modify a key on update
	104	Conflicting RPL options
	108	Invalid record length specified
	112	Invalid key length specified
	4	Read error on data component of file
	8	Read error on index component of file
	16	Write error on data component of file
	20	Write error on index component of file

For a full list of return codes and reason codes see the IBM publication DFSMS/MVS Macro Instructions for Data Sets.

MFMDS041E VSAM open error DD-ddname RC=nnFB=nnn

DSS has detected an error while attempting to open a VSAM data set. This message is accompanied by the z/OS system message IEC161I to describe the problem further. Common return code (RC) values and feedback code (FB) values include:

RC	FB	Description
4	116	Warning: file was not closed properly after previous use and has not been recovered using an internal VSAM verify
	118	Warning: file was not closed properly after previous use but has been recovered using an internal VSAM verify
8	128	No DD statement was found for this file
	136	Not enough virtual storage to complete the open processing
	144	I/O error while reading the catalog
	152	RACF denied open access

RC	FB	Description
	160	Inconsistent open options specified
	168	File already open by another user

For a full list of return codes and reason codes see the IBM publication DFSMS/MVS Macro Instructions for Data Sets. VSAM system messages are documented in the IBM manual z/OS System Messages Volume 4.

For a full list of abend codes and reasons, see the IBM manual MVS System Codes.

MFMDS050E DSS services not available

An attempt has been made to use DSS; however, it is not currently available. This may be due to a failure during initialization or configuration, or it may have been terminated during system shutdown while a transaction is still running.

Review the z/OS system log and the Mainframe Access job log and SYSOUT data sets for messages related to the failure. If you need further assistance contact Micro Focus SupportLine.

MFMDS052I CAPANVALET READY

MFA has initialized full access to CA/Panvalet services.

MFMDS053I CAENDEVOR API REL Brrvvv ESI

MFA has initialized full access to CA/Endevor API services using the release and version specified, and noting whether the External Security (ESI) or Alternate User ID (UID) features are active.

MFMDS054I Librarian Exit Table active

MFA has created a memory table to describe the selected exits to be specified during batch updates against the corresponding Librarian master file.

MFMDS055I MCGRPC feature (in)active

Indicates whether the MCGLIB DD card was specified as the LOAD library for remote calls as used by the Mainframe Call Generator (MCG) interface.

MFMDS056I ChangeMan DD=CMNxxxx Open failure

Access to site overrides specified for DD=CMNLIB$ and DD=CMNOPTS (to define default build procedures and site options repectively) failed during MFA startup.

MFMDS057I CAPANVALET member security exit loaded

The user exit MFAPVXIT was found and loaded to provide member-level security for Import, Export, and Directory access calls agains Panvalet master files.

MFMDS058I Librarian ELIPSGEN not found

This site configuration file is used to validate library types and language names for establishing Librarian run-time parameters. Without this information, new objects will default to TYPE=COB.

MFA Direct Messages

MFA0101I LOGON USER=userid TIME=hhmm DAY=nnn ID=nnnnnnn VER=nn

This information message indicates the time and date that a user session was established with the MFA Direct component. The time is specified using 24-hour clock notation, and the day is the day of the year.

The ID represents the relative session number so as to differentiate when the same user has logged on multiple times.

MFA0102I LOGOFF USER=userid ID=nnnnnnn

A user has logged off the system. This message is added to both the MFALOG and the SYSLOG.

MFA0103I AUTH FAIL USER=userid RC=rc RACF=rr/ss

A user logon failed to pass security authorization. The reason codes are tabulated below. MFA also inserts a logon failure message into the AUDIT log. Similarly, an IBM system diagnostic (RACF) may be added to the SYSLOG

RC	RACF	Description
04	04/00	No decision made
	04/04	No RACF profile found
	04/08	Request failed
08	08/00	Not authorized to logon

MFA0105I {AUTH or FAIL} USER=userid INTENT=action AM=method DSN=dsname

where:

action	is READ, WRITE or ALTER
method	is QSAM, BPAM, VSAM, LIBR, PANV, IMS or DB2
dsname	is the file being accessed

The results of a user access attempt are logged in the MFALOG. This provides an audit trail of the individual file accesses processed by MFA Direct. The information is only accumulated for those systems configured with the parameter: MFA_SAF_HISTORY=YES

MFA0111E MFA INVALID DISPATCHING VERB

MFA Direct has been scheduled with an invalid request. The request will not be serviced. This represents an internal protocol error by MFA. Please contact Micro Focus support if the problem persists.

MFA0112E MFA MCB Environment failure

MFA Direct is unable to initialize a new thread due to an unexpected environmental error. The new thread is rejected. Please contact Micro Focus support to report the problem.

MFA0113E MFA TSA already in use

An MFA thread has been unexpectedly re-dispatched while it is still processing a prior request. The new request is rejected. Please contact Micro Focus support to report the problem.

MFA0801E JES/SAPI RC=xx REASON=yy

A request using the JES Sysout API to retrieve a held file from JES spool has failed. Return codes and reason codes are documented in the IBM publication: Using the Subsystem Inferface (SC28-1879) A partial list is provided below:

RC	Reason	Description
08		Invalid search argument(s)
12		Unable to process now

RC	Reason	Description
16		Duplicate job name
20		Invalid destination
32		Logic error
	32	Conflicting arguments
	36	Invalid destination arguments
	40	Invalid job number
	44	Invalid form
	96	SAPI request header error
36		Invalid CLASS
40		Invalid disposition options

MFA0802E I/O ERROR ON INTERNAL READER

A request to submit a job to the JES internal reader has failed. The action is aborted. There may be an associated IBM system diagnostic on the SYSLOG indicating the nature of the failure.

MFA0803E JES SSI ERROR FUNC=xx RC=yy

A request using the JES Subsystem interface has failed. The only functions issued by MFA are FUNC=79 (SYSOUT API) or FUNC=80 (Extended Status API). Return codes and reason codes are described in the IBM publication: Using the Subsystem Interface (SC28-1879) A partial list is provided below:

RC	Description
04	Function not supported
08	JES not available
12	JES does not exist
16	Fatal error
20	Logic error
24	SSI not initialized

MFA0804E JES STATUS RC=xx REASON=yy

A request using the JES Extended Status API to acquire details of the next held output on JES spool has failed. Return codes and reason codes are documented in the IBM publication: Using the Subsystem Interface (SC28-1879) A partial list is provided below:

RC	Reason	Description
04		Invalid search argument(s)
08		Logic error
	04	Invalid destination
	08	Job ID low invalid
	0C	Job ID high invalid
	10	Job ID high too low
	14	Invalid Job class
	20	Unable to access Job queue

RC	Reason	Description
	24	Invalid control structure
	28	Invalid length
	2C	Invalid job name
	30	Invalid user name
	34	Invalid system name

MFA1034I CSI CATALOG ERROR nn RC=nnn REASON=nnn (ID) filter

An error occurred within the Catalog Search Interface as provided by IBM as part of the DF/SMS z/OS middleware. The ID represents an internal IBM software module ID, and the filter is the Catalog Search argument derived from the client input. The return codes are described in the IBM publication DF/SMS Managing Catalogs (SC26-7401), but a partial list is shown below. Most other return codes and reason codes are documented as part of IBM System Message IDC3009I and can also be found using the IBM Web Service LOOKAT.

RC	REASON	Description
100	04	Access error on catalog
122	04	Invalid filter key

MFA1035I CATALOG ERROR RC=nnn VOL=volid DSN=dsname

An error occurred attempting to OBTAIN a catlog record for the specified dsname on the specified volume ID. The dsname may no longer be catalogued on that volume, and MFA is unable to determine the file attributes. It is possible that the volume is simply not mounted, or is no longer in active service. This file will be ignored and processing will continue. The return codes are described in the IBM publication DFSMSdfp Advanced Services (SC26-7400)

RC	Description
04	Volume not mounted
08	The format-1 DSCB was not found in the VTOC of the specified volume
12	A permanent I/O error was encountered reading the VTOC of the specified volume
16	An invalid workarea pointer was encountered

MFA1036I CSI VSAM ERROR RC=nnn REASON=nnn DSN=dsname

An error occurred attempting to use the CSI interface to identify the VSAM attributes of the specified dataset. This file will be ignored and processing continues. The return codes are as described for message MFA1034I.

MFA1037I CSI GDGB ERROR RC=nnn REASON=nnn DSN=dsname

An error occurred attempting to use the CSI interface to identify the file attributes of the specified generation data group. This file will be ignored and processing continues. The return codes are as described for message MFA1034I.

MFA4001E CA/LIBRARIAN ERROR REQ=aaaa RC=nnn

An internal CA/Librarian error occurred while processing the FAIR API request identified as: aaaa={OPN or LOC or MOD or REC or CLS} The return codes are described in the CA licensed documentation CA/ Librarian File Access Interface Routines Guide. A partial list is provided below:

REQ	RC	Description
OPN	1	Invalid filename or unauthorized
	2	DDname missing
	8	Invalid parameter
	9	Not enough storage to process
LOC	1	Required module not found
	2	Caller not authorized for request
	6	Disk access failure
	7	Format error on access
	8	Invalid parameter
	9	Not enough storage to process
MOD	1	No modules found
	2	Caller not authorized for request
	3	Module not archived
	6	Disk access failure
	7	Format error on access
	8	Invalid parameter
	9	Virtual storage shortfall
REC	1	End of module
	6	Disk access error
	7	Format error on access
	8	Invalid parameter
	9	Virtual storage shortfall
CLS	81	Close failed

MFA4002E OPEN FAILED ON LIBRARIAN UPDATE

The system was unable to open a temporary work file for use with the batch update utility AFOLIBR in an attempted Librarian update operation. Additional information may be available on the SYSLOG if an associated IBM internal system problem was encountered. As well, a small MFA snap dump will be issued to provide additional problem determination data. No batch update will be attempted. If you are unable to resolve the problem, open an incident with Micro Focus product support, and include the Snap Dump and the MFALOG as part of the problem description.

MFA4003E WRITE ERROR ON LIBRARIAN UPDATE

The system was unable to write to a temporary work file for use with the batch update utility AFOLIBR. Additional information may be available on the SYSLOG if an associated IBM data access error was encountered. As well, a small MFA snap dump will be issued to provide additional problem determination data. No batch update will be attempted. The probable cause here is an attempt to write a record (from the workstation) which exceeds the logical record size of the work file.

MFA5001E CA/PANVALET ERROR REQ=aaaa RC=nnn MSG=PVnnn

An internal Panvalet error occurred while processing a PAM API request identified as: aaa={OPEN or CLS or READ or SRCH} The return codes and PanValet message numbers are documented within the CA licensed documentation: CA/Panvalet Messages Guide CA/Panvalet System Management Guide An incomplete summary follows:

Message	Description
PV001	Invalid command
PV002	Member name invalid
PV003	Invalid sequence number
PV004	Excessive parameters
PV005	Required parameter missing
PV006	Numeric parameter too large
PV007	Parameter exceeds 10 characters
PV008	Unsupported language format
PV009	Invalid parameter
PV012	No statement found
PV023	Name not found
PV032	Sequence error
PV033	Library error
PV036	Access error: member locked
PV043	Library storage exceeded
PV046	Panvalet library empty
PV047	Virtual storage shortfall
PV051	Invalid Library file
PV066	Function terminated due to errors
PV070	Statement truncated
PV095	Invalid DCB for Library
PV118	LRECL must be 80 for this language type
PV124	Invalid file

MFA5002E CA/PANVALET LICENSE FAILURE: access quiesced

This site is not licensed to access the PanValet API. Further attempts to access PanValet will be suspended by MFA to avoid repetitive errors. Contact PanValet support to re-establish your license keys. Re-starting MFA will resume PanValet access attempts.

MFA5003E CA/PANVALET PAN#1 error RC=nnnnn

An internal Panvalet error was encountered while processing a batch update request using the PAN#1 utility. The return codes are described within the CA licensed documentation: CA/Panvalet Messages Guide CA/Panvalet User Guide

MFA5004E I/O ERROR on PANVALET Temporary file

A QSAM error was encountered while writing a temporary file needed as a pre-requisite for the PAN#1 batch update utility. Normally, an IBM system error will appear on the SYSLOG to describe the actual reason, The probable cause is an attempt to write a logical record from the workstation which exceeds the logical record size of the host file.

MFA6001E ENDEVOR API ERROR REQ=aaaa RC=nnn/rrrr/msgid

An Endevor internal processing error occurred while processing an API request identified as: aaaa={ERET or LELM or EADD} A copy of the actual Endevor error message is inserted into the MFALOGE listing if

MFA_ENDEVOR_LOGGING is active. The return code, reason code, and Endevor message numbers are described in the licensed CA/Endevor documentation: Endevor for z/OS Messages and Codes Endevor for z.OS API Guide

MFA6002E I/O ERROR ON ENDEVOR TEMPORARY FILE

A QSAM error was encountered while writing a temporary file needed as a pre-requisite for an Endevor API request. Normally, an IBM system error will appear on the SYSLOG to describe the actual reason, The probable cause is an attempt to write a logical record from the workstation which exceeds the logical record size of the temporary file.

MFA6003E SYNTAX ERROR ON ENDEVOR ENVIRONMENT STRING

An invalid environment string was encountered during an Endevor request. The transaction is aborted. Use you client software to re-specify the Endevor prompts and properties correctly.

MFA6004E ENDEVOR INTERFACE ABEND: abend-reason userid session-id

The Mainframe Access ESTAE recovery routines detected an abend while processing an Endevor transaction. The transaction is aborted. The above message appears in the MFALOG showing the diagnostic abend code and reason code.

Data Connect Messages

MFA100E ESTAE ENTERED ABEND CODE Uxxx/Sxxx

Data Connect has detected an abend. The client software is notified and the dump is normally suppressed. See the MVS console log for additional imformation.

MFA101E LIMIT OF 256 PROCESSES EXCEEDED

Logic within Data Connect can manage up to 256 concurrent sessions. It is unable to retain positioning and recovery information for additional sessions.

MFA110I VTAM PROCESSOR ACTIVE USING VTAMAPPL applid

Data Connect has successfully created a VTAM connection and is available to service message traffic using APPLID=applid.

MFA111W VTAM ACB CLOSE FAILURE

Data Connect was unable to close the VTAM connection successfully. This may be a VTAM environmental problem. See the MVS console log for additional information. Shutdown processing continues without VTAM access.

MFA112I VTAM PROCESSOR HAS TERMINATED

Data Connect is no longer servicing VTAM message traffic. Shutdown processing continues normally.

MFA113E USING VTAMAPPL applid ACB OPEN ERROR X"xx"

Data Connect was unable to open the specified APPLID. Therefore, access cannot be provided to network clients. Common errors are identified as described below.

Error	Description
54	APPLID not defined to VTAM
58	APPLID already in use

Error	Description
5A	APPLID inactive to VTAM

MFA114E APPC SESSION REJECTED PRI=xxxxxxxx SEC=xxxxxxxx

Data Connect was unable to complete the logon attempt. See the VTAM APPC documentation for a description of the primary and secondary return codes provided by IBM.

MFA115E MFA DOES NOT SUPPORT 3270 DATASTREAMS

Data Connect does not support LU2 3270 data stream connections. Only APPC VTAM message traffic is supportted.

MFA201W ERROR OPENING DDNAME=VSAMCTL

A file open error was encountered for the VSAM control file. Verify that DDNAME was provided in the Data Connect started task as shown in the sample JCL as MFAS. See the z/OS console log for additional error messages.

MFA202I ERROR READING VSAMCTL FOR membername

A file read error was encountered for the indicated member name. The physical data may be corrupted. See the z/OS console log for additional error information.

MFA299E ADDRESS SPACE NOT AUTHORIZED

The Data Connect modules must reside in an authorized library. The STEPLIB concatenation probably contains at least one load library which is not APF-authorized. Please call your system programmer to authorize your libraries.

MFA301E JCL PARM INVALID OR OMITTED

The parm passed on the // EXEC PGM=MFDSTART,PARM='...' is invalid. The only keyword supported is APPLID=applid. Correct the JCL before resubmitting this job.

MFA302I ATTACHING VTAM PROCESSOR

Data Connect is now initializing APPC access for client messages as relayed through the Host Connectivity interface.

MFA303I MFA/DATACONNECT V4.00 - vers

This message identifies the Data Connect version and copyright information.

MFA310E INVALID OPERATOR REQUEST IGNORED

An invalid operator command was sent to Data Connect. The command is being ignored.

MFA311W PROCESS ALREADY RUNNING

VTAM APPC support is already active. It is not possible to re-cycle comminations while the APPLID is still active.

MFA317I jobname HAS BEEN TERMINATED

The Data Connect task has been successfully terminated.

Drag and Drop Error Messages

Refer to the table below for a list of possible error messages:

MFDAS01: Mainframe access problem.

Return Code

1001

Description

A communication problem has been detected. Refer to the MFAERROR.LOG file for more information.

MFDAS02: Cannot import multiple files to a single file.

Return Code

1002

Description

An illogical request has been made, in that the user is attempting to import more than one file into a single file.

MFDAS03: Cannot export multiple files to a single file.

Return Code

1003

Description

An illogical request has been made, in that the user is attempting to export more than one file into a single file.

MFDAS04: Module 'MFLSC' is missing or inaccessible.

Return Code

1004

Description

The required module MFLSC cannot be found. This is most likely an installation problem.

MFDAS05: File had no length, or does not exist.

Return Code

1005

Description

The file had either a zero length, or had been deleted but the display was not refreshed.

MFDAS06: Call to module 'DFCONV' failed. Unable to convert the VSAM/QSAM file.

Return Code

1006

Description

A bad return code was given by the converter program. Run the `dfconv` program on its own to determine the error.

MFDAS07: Value is not a valid number.

Return Code

1007

Description

A non-numeric value was given when a numeric value was expected.

MFDAS08: For VSAM/QSAM files, extensions of PRO, DAT, or IDX are reserved.

Return Code

1008

Description

You cannot use a restricted extension for a file transfer.

MFDAS09: Rename to file failed.

Return Code

1009

Description

An error was returned when trying to rename a file.

MFDAS10: The allocation of failed with message number.

Return Code

1010

Description

An error message has been returned from the COBOL development environment. See the product documentation for a description of the message.

MFDAS11: The allocation of differs from the mainframe.

Return Code

1011

Description

The DCB information for the COBOL development environment dataset differs from that of the mainframe.

MFDAS12: One or more of the fields have not been properly entered.

Return Code

1012

Description

Insufficient or incorrect information has been provided in the panel.

MFDAS13: The PC file name cannot have more than 4 subdirectory levels.

Return Code

1013

Description

The COBOL development environment is unable to support more than 4 subdirectory levels.

MFDAS14: Mismatch between new and repeat passwords.

Return Code

1014

Description

The new and repeat passwords given do not match.

MFDAS15: Unable to make the directory.

Return Code

1015

Description

An error was returned when trying to create a directory.

MFDAS16: Unable to remove directory.

Return Code

1016

Description

An error was returned when trying to remove a directory.

MFDAS17: Call to module 'PCIMS' failed. Unable to convert the IMS database.

Return Code

1017

Description

A bad return code was given by the PCIMS program. Run the PCIMS program on its own to determine the error.

MFDAS18: A catalog search criteria must be entered.

Return Code

1018

Description

You must enter one or more mainframe catalog search criteria. If specifying more than one, then separate them by commas. This catalog search criteria is similar to that when using ISPF 3.4.

MFDAS19: A real workstation drive must be used.

Return Code

1019

Description

The drive selected for the COBOL development environment dataset does not exist on this workstation.

MFDAS22: Call to module 'SQL-Wizard' failed. Unable to convert the XDB table.

Return Code

1022

Description

A bad return code was given by the SQL wizard program. Run the SQL wizard program on its own to determine the error.

MFDAS23: Cannot establish connection to the mainframe server.

Return Code

1023

Description

Unable to connect to the mainframe server.

MFDAS24: Network communication problems with blocks of size.

Return Code

1024

Description

The user's network is unable to transport a data block of the size specified. This is most likely a router configuration problem. To temporarily circumvent the problem, you can specify the MFAMAXSENDLENGTH environment variable, with the value set to a size below the size in error.

MFDAS25: CCI module was not found.

Return Code

1025

Description

The CCI communications file is not found.

MFDAS26: Module is not a CCI module.

Return Code

1026

Description

The module specified is not a CCI communications file.

MFDAS27: Unable to delete file.

Return Code

1027

Description

An error was returned when trying to delete a file.

MFDAS28: Either recall this dataset or assign a type.

Return Code

1028

Description

This dataset is migrated. Either use the right-mouse context to recall it, or assign a specific type.

MFDAS29: Assign a type to this dataset.

Return Code

1029

Description

Unknown DSORG for this dataset. You need to use the right-mouse context to assign a specific type.

MFDAS30: The workstation dataset is connected to the mainframe

Return Code

1030

Description

This workstation dataset you are tying to access is redirection to a mainframe dataset.

MFDAS31: Copy of file failed.

Return Code

1031

Description

Unable to copy from the Windows clipboard.

MFDAS32: Paste of file failed.

Return Code

1032

Description

Unable to paste the file to the Windows clipboard.

MFDAS33: PCIMS is not available. Unable to convert the IMS database

Return Code

1033

Description

The PCIMS.LBR routines could not be found, or a project was not open.

Status Codes and Error Messages

This section contains a list of status codes that Mainframe Access servers can return to a client program. It also documents the anomalies in file status settings based upon the environment in which the I/O statements are executed.

Note: For additional information on MFA messages, see your Mainframe Access Administrator's Guide.

Status Codes Returned by the MVS Server

Table 1: File Status Codes Returned by the MVS Server

Status Code	Description
0/0	The MVS server returns a status of 0/0 when writing a record to an indexed file which results in two or more records with identical keys. Both IBM VS COBOL II batch processing and Micro Focus COBOL return a status of 0/2 when such a condition arises.
0/5	A file status 0/5 indicates an open operation was successful, but the file is empty. A file status of 0/5 is returned in some cases where IBM VS COBOL II returns a status of 0/0 and in other cases where IBM VS COBOL II returns a status of 3/5. See the appendix *File Status Comparisons* for some examples of the conditions under which this status code is returned during open processing.
4/3	A status code of 4/3 is returned if an attempt is made to delete a VSAM record via an alternate index, if another record in that file contains the same key as the record being deleted.
4/6	If an I/O operation is attempted which requires a previous I/O operation to have set the current record pointer, and the previous I/O operation failed, a file status of 4/6 is returned. It is also issued

Status Code	Description
	if the MVS server loses track of the current record position, probably due to an invalid sequence of I/O statements.
9/100	A file status of 9/100 is returned if the MVS server receives an operation code that it does not support. This status is also returned when the server attempts to execute a command at a time when that command is invalid. In this case, the operation code received by the server was valid, but the I/O operation was invalid given the type of operation requested and the nature of the previous I/O operations executed for the file.
9/125	A status code of 9/125 will be returned when the maximum number of concurrent Users has been exceeded.
9/242	The requested function or component has not been licensed.

File Status Comparisons

This section documents the anomalies in file status settings based upon the environment in which the I/O statements are executed. Expected results, including expected anomalies (for example, the order of records returned when reading an alternate index in reverse order), are not included in this list unless they help clarify the conditions under which the anomalies were detected.

These results were compiled from a set of test programs executed in the following three environments:

- MVS-batch using IBM VS COBOL II accessing VSAM datasets
- OS/2 using Micro Focus COBOL accessing sequential
- Relative and indexed files

The test programs used an extensive combination of file definitions (organization and access mode), open types (INPUT, I/O, OUTPUT, EXTEND, etc.), and I/O verbs for example READ, WRITE, REWRITE) to examine the behavior of execution environment and the results of the file status information returned to the test programs.

In the Cross Reference of File Status Setting table that follows, file status information obtained for the tests in each environment is displayed in the format 'a/b', where 'a' represents the first byte of the file status (file status 1) and 'b' represents the second byte of the file status (file status 2).

For run-time errors, the first byte displayed is '9' and the value for the second byte is the ordinal value of the file status 2 field (e.g. 9/100). When an I/O operation is invalid, a key of 'N/A' is used. When an I/O operation is valid but was not tested because of the results of an earlier I/O operations, a file status of '-/-' is used.

Cross Reference of File Status Setting

READ SEQUENTIAL TEST WITH EMPTY FILES

FILE ORG-ACC	I/O statements	MVS-BATCH	OS/2	CICS server
		COBOL II	MF/COBOL	
SEQ SEQ	OPEN I/O	3/5	0/5	0/5
	READ NEXT	-/-	1/0	1/0
SEQ SEQ	OPEN I/O REVERSED	N/A	0/0	0/5
	READ NEXT	-/-	1/0	1/0
REL SEQ	OPEN I/O	3/5	0/5	0/5
	READ NEXT	-/-	1/0	1/0

FILE ORG-ACC	I/O statements	MVS-BATCH	OS/2	CICS server
		COBOL II	MF/COBOL	
REL SEQ	OPEN I/O	3/5	0/5	0/5
	START RRN=20	-/-	2/3	2/3
REL SEQ	OPEN I/O	3/5	0/0	3/5
	START RRN=0	-/-	2/3	-/-
IDX SEQ	OPEN INPUT	0/0	3/5	0/0
	READ NEXT	1/0	-/-	1/0
IDX SEQ	OPEN I/O	0/0	0/5	0/0
	START KEY < VALUE	N/A	2/3	2/3

When Micro Focus COBOL attempts to open a sequential dataset for I/O REVERSED processing and that dataset does not exist, a status code of 3/5 is returned. If the dataset does exist but is empty, a status code of 0/0 is returned on the open request.

WRITE SEQUENTIAL TEST WITH EMPTY FILES

FILE ORG-ACC	I/O statements	MVS-BATCH	OS/2	CICS server
		COBOL II	MF/COBOL	
IDX SEQ	OPEN OUTPUT	0/0	0/0	0/0
	WRITE (RECORD W/ DUPLICATE ALT KEY)	0/2	0/2	0/0
IDX SEQ	OPEN EXTEND	0/0	0/0	0/0
	WRITE (RECORD W/ DUPLICATE ALT KEY)	0/2	0/2	0/0

The CICS server cannot determine if the result of this write operation would cause any alternate indexes to have two or more records with the same key value.

DELETE SEQUENTIAL TEST

FILE ORG-ACC	I/O statements	MVS-BATCH	OS/2	CICS server
		COBOL II	MF/COBOL	
IDX SEQ	OPEN I/O	0/0	0/0	0/0
	START ALT KEY GTEQ LOW-VALUES	0/0	0/2	0/0
	DELETE	0/0	0/0	1/0

The CICS server does not permit the deletion of a record via the alternate index if another record in the file has the same key for the alternate index. A status code of 4/3 is returned. The current record position is unaffected by the failed request.

RANDOM WRITE TEST

FILE ORG-ACC	I/O statements	MVS-BATCH	OS/2	CICS server
		COBOL II	MF/COBOL	
IDX RAN	OPEN I/O	0/0	0/0	0/0
	WRITE (RECORD W/ DUPLICATE ALT KEY)	0/2	0/2	0/0

The CICS server cannot determine if the result of this write operation would cause any alternate indexes to have two or more records with the same key value.

READ SEQUENTIAL TEST WITH DATA IN FILES

FILE ORG-ACC	I/O statements	MVS-BATCH	OS/2	CICS server
		COBOL II	MF/COBOL	
IDX SEQ	OPEN INPUT	0/0	0/0	0/0
	START KEY GTEQ HIGH-VALUES	2/3	2/3	0/0
	READ NEXT	-/-	-/-	1/0

VSAM treats this as: 'position the current record pointer to the end of the file'; a subsequent READ PREVIOUS request would return the last record in the file.

READ DYNAMIC TEST WITH DATA IN FILES

FILE ORG-ACC	I/O statements	MVS-BATCH	OS/2	CICS server
		COBOL II	MF/COBOL	
IDX DYN	OPEN INPUT	0/0	0/0	0/0
	START KEY GTEQ HIGH-VALUES	2/3	2/3	0/0
	READ NEXT	-/-	-/-	1/0

VSAM treats this as: 'position the current record pointer to the end of the file'; a subsequent READ PREVIOUS request would return the last record in the file.

DELETE DYNAMIC TEST WITH DATA IN FILES

FILE ORG-ACC	I/O statements	MVS-BATCH	OS/2	CICS server
		COBOL II	MF/COBOL	
IDX DYN	OPEN I/O	0/0	0/0	0/0
	START ALT KEY GTEQ LOW-VALUES	0/0	0/0	0/0
	READ NEXT (DUPLICATE KEY)	0/2	0/2	0/2
	READ NEXT (LAST OF DUPLICATE KEY)	0/0	0/0	0/0

FILE ORG-ACC	I/O statements	MVS-BATCH COBOL II	OS/2 MF/COBOL	CICS server
	DELETE (LAST OF DUPLICATE KEY)	0/0	0/0	4/3

The CICS server does not permit the deletion of a record via the alternate index if another record in the file has the same key for the alternate index. A status code of 4/3 is returned. The current record position is unaffected by the failed request.

Configuration reference

You can configure many aspects of z/Server's behavior by using any of z/Server's configuration parameters.

Mainframe Access Server configuration parameters

Configuration parameters - quick reference

The following list provides Mainframe Access Server configuration parameters and for each parameter, which servers and services it applied to:

Parameter	General	Application Servers						Services		
		Data Connect	Endevour	ChangeMan	AMS	TSO	RIMS	MCO	MFA	ES-MTO
ACBNAME	N	N	N	N	N	N	N	N	N	Y
ACCESS_LIST_CHECK	Y	N	Y	Y	Y	Y	N	N	N	N
APPLID_PASSWORD	Y	N	N	N	N	N	N	N	N	N
AUDIT_LOG	Y	N	N	N	N	N	N	N	N	N
BACK_LOG	Y	N	N	N	N	N	N	N	N	N
BIND_NETADDR	Y	N	N	N	N	N	N	N	N	N
BUFFER_SIZE	Y	N	Y	Y	Y	Y	N	N	N	N
CHANGEMAN_COMMON_BUILD	Y	N	N	N	N	N	N	N	N	N
CHANGEMAN_DSNQUALIFIER	Y	N	N	N	N	N	N	N	N	N
CHANGEMAN_INTERFACE	Y	N	N	N	N	N	N	N	N	N
CHANGEMAN_SSID	Y	N	N	N	N	N	N	N	N	N
CHANGEMAN_TEST_OPTION	Y	N	N	N	N	N	N	N	N	N
CHANGEMAN_XMS_SIZE	Y	N	N	N	N	N	N	N	N	N
COMPRESSION	Y	N	Y	Y	Y	Y	N	N	N	N

Parameter	General	Application Servers					Services			
		Data Connect	Endevour	ChangeMan	AMS	TSO	RIMS	MCO	MFA	ES-MTO
DSS_SCAN_INTERVAL	Y	N	Y	Y	Y	Y	N	N	N	N
EID	Y	N	Y	Y	Y	Y	N	N	N	N
ENDEVOR_DSNQUALIFIER	Y	N	N	N	N	N	N	N	N	N
ES-MTO_MAXTASKS	Y	N	N	N	N	N	N	N	N	N
ID	N	N	N	N	N	N	Y	Y	Y	Y
IMSLINK_MAXTASKS	Y	N	N	N	N	N	N	N	N	N
IPADDRESS	N	N	N	N	N	N	N	N	N	Y
JES_BUFFER_TRACE	Y	N	N	N	N	N	N	N	N	N
JES_CONCHAR	Y	N	N	N	N	N	N	N	N	N
JOBNAME	N	N	Y	Y	Y	Y	N	N	N	N
LIST_DB2	Y	N	N	N	N	N	N	N	N	N
LIST_PARAMETERS	Y	N	Y	Y	Y	Y	N	N	N	N
LU62_APPLID	Y	N	N	N	N	N	N	N	N	N
LUNAME	N	N	N	N	N	N	Y	Y	Y	N
MAX_DS_ALLOCATIONS	Y	N	N	N	N	N	N	N	N	N
MAX_PUBLIC_FILES	Y	N	N	N	N	N	N	N	N	N
MAXIMUM	N	N	Y	Y	Y	Y	N	N	N	N
MCOLINK_MAXTASKS	Y	N	N	N	N	N	N	N	N	N
MFA_ACCEPT_EMPTY_FILES	Y	N	N	N	N	N	N	N	N	N
MFA_ENDEVOR_HISTORY	Y	N	N	N	N	N	N	N	N	N
MFA_ENDEVOR_INTERFACE	Y	N	N	N	N	N	N	N	N	N
MFA_GUI_ACCEPT_PASSPHRASES	Y	N	N	N	N	N	N	N	N	N
MFA_GUI_BLANKPASSWORD	Y	N	N	N	N	N	N	N	N	N
MFA_GUI_IGNOREUSERCTO	Y	N	N	N	N	N	N	N	N	N
MFA_JOBNAME_CHECK	Y	N	N	N	N	N	N	N	N	N
MFA_LIBRARIAN_DIR_INFO	Y	N	N	N	N	N	N	N	N	N
MFA_LIBRARIAN_HISTORY	Y	N	N	N	N	N	N	N	N	N

Parameter	General	Application Servers					Services			
		Data Connect	Endevour	ChangeMan	AMS	TSO	RIMS	MCO	MFA	ES-MTO
MFA_LIBRARIAN_INTERFACE	Y	N	N	N	N	N	N	N	N	N
MFA_LIBRARIAN_UPD_MODULE	Y	N	N	N	N	N	N	N	N	N
MFA_LISTING_DATA_CLASS	Y	N	N	N	N	N	N	N	N	N
MFA_MCG_DB2CONNECTION	Y	N	N	N	N	N	N	N	N	N
MFA_PANVALET_HISTORY	Y	N	N	N	N	N	N	N	N	N
MFA_PANVALET_INTERFACE	Y	N	N	N	N	N	N	N	N	N
MFA_PANVALET_UPD_MODULE	Y	N	N	N	N	N	N	N	N	N
MFA_SAF_HISTORY	Y	N	N	N	N	N	N	N	N	N
MFA_SYSOUT_CLASS	Y	N	N	N	N	N	N	N	N	N
MFA_SYSOUT_DEST	Y	N	N	N	N	N	N	N	N	N
MFADIRECT	Y	N	N	N	N	N	N	N	N	N
MFADIRECT_MAXTASKS	Y	N	N	N	N	N	N	N	N	N
MFALINK_MAXTASKS	Y	N	N	N	N	N	N	N	N	N
MINIMUM	N	N	Y	Y	Y	Y	N	N	N	N
MODENAME	N	N	N	N	N	N	Y	Y	Y	N
NETWORK_ID	Y	N	N	N	N	N	N	N	N	N
ORGANIZATION	Y	N	N	N	N	N	N	N	N	N
PEM_FEATURE	Y	N	N	N	N	N	N	N	N	N
PORT	N	N	N	N	N	N	N	N	N	Y
PROCEDURE	N	Y	Y	Y	Y	Y	N	N	N	N
RACF_APPLID	Y	N	N	N	N	N	N	N	N	N
ROUTE	Y	N	Y	Y	Y	Y	N	N	N	N
SECURITY	N	N	N	N	N	N	Y	Y	Y	N
SESSIONS	N	N	N	N	N	N	N	N	N	Y
SMF_RECORDID	Y	N	Y	Y	Y	Y	N	N	N	N
SOCKETS	N	N	N	N	N	N	N	N	N	Y
SYNCLEVEL	N	N	N	N	N	N	Y	Y	Y	N
SYSOUT	Y	N	Y	Y	Y	Y	N	N	N	N
TCP_PORT	Y	N	N	N	N	N	N	N	N	N

Parameter	General	Application Servers					Services			
		Data Connect	Endevour	ChangeMan	AMS	TSO	RIMS	MCO	MFA	ES-MTO
TCPLINK_MAXTASKS	Y	N	N	N	N	N	N	N	N	N
TIMEOUT_FOR_CONNECTION	Y	N	Y	Y	Y	Y	N	N	N	N
TIMEOUT_FOR_INITIAL_RECEIVE	Y	N	N	N	N	N	N	N	N	N
TPNAME	N	N	N	N	N	N	Y	Y	Y	N
TRACE	Y	N	Y	Y	Y	Y	N	N	N	N
TRACING	Y	N	Y	Y	Y	Y	N	N	N	N
TSO_JOBCHAR	N	N	N	N	N	Y	N	N	N	N
TSO_PROCEDURE	N	N	N	N	N	Y	N	N	N	N
VIO_FOR_ENDEVOR_LOG	N	N	Y	N	N	N	N	N	N	N

Configuration parameters - alphabetical list

ACBNAME

Name ACBNAME

Summary The name of the VTAM ACB associated with this ES/MSS server. MFA Server OPENs and initializes this ACBNAME during startup. The z/OS CICS system(s) must be configured to communicate with this ES/MSS server through this VTAM ACB name using CICS CONNECTION and SESSIONS definition statements.

Default NULL

Supported Values

Value	Description
acbname	A valid VTAM ACBNAME.

Example ACBNAME="acbname"

ACCESS_LIST_CHECK

Name ACCESS_LIST_CHECK

Summary Indicates whether or not the access list feature is to be activated for clients. This enables you to restrict access to Mainframe Access according to the connecting client's IP address. See *Editing Access List Definitions* for more information. Specify 0 to accept all client connection requests, regardless of the client's IP address. Specify 1 to permit client connections based on the PERMIT and REJECT rules for IP addresses defined in your access list.

Default 0

Supported Values

Value	Description
0	Allow client access regardless if the clients IP address.

Value	Description
1	Restrict client access based on PERMIT and REJECT rules.

Example ACCESS_LIST_CHECK="1"

APPLID_PASSWORD

Name APPLID_PASSWORD

Summary The ACF/VTAM ACB password to be used when Mainframe Access opens its VTAM ACB. Specify a password consisting of between one and eight characters. This value must match the password value specified by the PRTCT parameter in the Mainframe Access VTAM application program major node definition in VTAMLST.

Default MFM62PSW

Supported Values

Value	Description
applidpswd	A valid VTAM APPLID password.

Example APPLID_PASSWORD="applidpswd"

AUDIT_LOG

Name AUDIT_LOG

Summary Indicates whether or not the audit log is to be activated. Specify 1 to activate or 0 to deactivate. If audit logging is selected, be sure to complete the tasks described in *Allocating and Initializing an Audit Log Data Set*.

Default 0

Supported Values

Value	Description
0	Deactivate audit logging.
1	Activate audit logging.

Example AUDIT_LOG="1"

BACK_LOG

Name BACK_LOG

Summary The number of pending TCP/IP connection requests that the TCP/IP network software should hold while Mainframe Access is processing the current connection request. Connection processing is typically handled very quickly and a small backlog of between 5 and 10 should be adequate for most installations. If you have an unusually high number of concurrent connection requests, some end users might receive a connection reject message. Specify a positive integer value in the range 5 through 50.

Default 5

Minimum Value 5

Maximum Value 50

Example BACK_LOG="10"

BIND_NETADDR

Name BIND_NETADDR

Summary Selects the network address(es) to which the Mainframe Access listening port (TCP_PORT) should be bound. Specify 0.0.0.0 (or omit this parameter) to bind the ports to all network addresses assigned to the host system. Specify a specific network address or a logical host name (64-character maximum) to restrict Mainframe Access to a single, specific host network address.

Default 0.0.0.0

Supported Values

Value	Description
bindnetaddr	A valid IP address or logical host name.

Example BIND_NETADDR="bindnetaddr"

BUFFER_SIZE

Name BUFFER_SIZE

Summary The maximum amount of data that Mainframe Access should send or receive through the TCP/IP network software in one send or receive operation. Specify 0 (zero) to allow Mainframe Access to use the optimum buffer size; this is recommended, as it will reduce the number of send or receive calls made to the TCP/IP network software. Specify an explicit value in the range 512 to 32767 only if you have diagnosed a problem with your TCP/IP network software that is associated with large message sizes.

Default 0

Minimum Value 512

Maximum Value 32767

Example BUFFER_SIZE="600"

CHANGEMAN_COMMON_BUILD

Name CHANGEMAN_COMMON_BUILD

Summary Specifies whether or not Mainframe Access accommodates group builds submitted under AppMaster Builder. Specify 1 to enable group builds or 0 to disable them. The default is 0. When enabled for group builds, MFA modifies the build options for each member of the AMB group to match the component name and type submitted, ensuring that each build request in the group is properly executed.

Default 0

Supported Values

Value	Description
0	Disable group builds.
1	Enable group builds.

Example CHANGEMAN_COMMON_BUILD="1"

CHANGEMAN_DSNQUALIFIER

Name CHANGEMAN_DSNQUALIFIER

Summary Specifies a high-level qualifier for dynamically allocated data sets created by MFA Server to hold input files during the staging process. MFA Server creates these data sets dynamically as card-image sequential data sets that are deleted when the associated client request completes. The default qualifier is the login ID of the current user. You should set an alternate qualifier if your installation's SERNET does not have RACF authority to read or write data into files stored in data sets named with the current user's login ID as the high-level qualifier.

If an alternate qualifier is required, see your ChangeMan administrator or your security administrator for assistance in selecting a proper value.

If your ChangeMan functionality is restricted by security subsystem authorizations, you might need to specify your dataset high-level qualifier such that read access to the data sets is always permitted.

Default NULL

Supported Values

Value	Description
changemands	A valid dataset HLQ.

Example CHANGEMAN_DSNQUALIFIER="changemandsn"

CHANGEMAN_INTERFACE

Name CHANGEMAN_INTERFACE

Summary Specifies the name of the ChangeMan ZMF module. During ChangeMan request processing, MFA Server loads and branch enters this module. Valid values are SERXMLBC and SERXMLAC. The default is SERXMLAC.

The preferred interface module is SERXMLAC, which is not documented in ChangeMan ZMF. It provides better interface performance than SERXMLBC by using virtual storage instead of data sets for XML input and output exchange between MFA Server (the requester) and ChangeMan ZMF.

ChangeMan ZMF documents only the SERXMLBC interface, which is the XML services batch client. SERXMLBC is intended for use in a batch job stream and requires XMLIN and XMLOUT DD statements to define its required data sets. The MFAAS JCL sample provides sample DD statements for XMLIN and XMLOUT which are commented out by default. If you must use SERXMLBC, contact SupportLine for assistance with data set definitions.

Default SERXMLAC

Supported Values

Value	Description
SERXMLAC	Use XML to interface with ChangeMan.
SERXMLBC	Use batch client to interface with ChangeMan.

Example CHANGEMAN_INTERFACE="SERXMLBC"

CHANGEMAN_SSID

Name CHANGEMAN_SSID

Summary Initializes the Mainframe Access ChangeMan interface by providing the final character of the z/OS subsystem ID used to identify ChangeMan. Omit this parameter to bypass Mainframe Access initialization for the ChangeMan interface. Valid values are any single alphabetic or numeric character. The character specified is appended to the subsystem

ID string "SER", completing the four-character ID. For example, if you specify CHANGEMAN_SSID=A, the z/OS subsystem ID becomes "SERA".

This parameter is required in both the PARMS member and the PARMSAS member used by application server address spaces.

Your ChangeMan administrator can help you to determine the correct subsystem ID to use.

Default NULL

Supported Values

Value	Description
changemanssid	Any single alphabetic or numeric character.

Example CHANGEMAN_SSID="changemanssid"

CHANGEMAN_TEST_OPTION

Name CHANGEMAN_TEST_OPTION

Summary Specify whether or not MFA uses the XML trace option. Specify 1 to enable XML trace or 0 to disable it. The default is 0. When enabled, SERNET logs information in XML format into the standard SERPRINT DD file. This file can be used to identify problems and to validate MFA services. Enabling XML trace is not recommended for production systems due to the large volume of log data generated.

Default 0

Supported Values

Value	Description
0	Disable MFA ChangeMan XML tracing.
1	Enable MFA ChangeMan XML tracing.

Example CHANGEMAN_TEST_OPTION="1"

CHANGEMAN_XMS_SIZE

Name CHANGEMAN_XMS_SIZE

Summary The size of the XML buffer used to receive members when using ChangeMan.

Default 1

Minimum Value 1

Maximum Value 8

Example CHANGEMAN_XMS_SIZE="5"

COMPRESSION

Name COMPRESSION

Summary Indicates whether or not Mainframe Access should apply data compression when communicating with SQL Option for DB2 clients. Specify 1 to activate data compression or 0 to deactivate it.

Default 0

Supported Values

Value	Description
0	Deactivate data compression.
1	Activate data compression.

Example COMPRESSION="1"

DSS_SCAN_INTERVAL

Name DSS_SCAN_INTERVAL

Summary The time period in minutes between Data Set Services checks for abandoned buffers or open files. Due to unforeseen environmental errors, transactions may abnormally terminate while holding open files and other I/O resources. These scans locate idle resources and return them to an available status and ensure that no file is held idle for more than two consecutive scan intervals. Specify the number of minutes as an integer value between 30 and 90. The default is 30 minutes.

Default 30

Minimum Value 30

Maximum Value 90

Example DSS_SCAN_INTERVAL="80"

EID

Name ACBNAME

Summary The event ID of Mainframe Access GTF user trace records. The EID must be available for exclusive use by Mainframe Access and must match the USR=(eid) parameter specified to GTF when GTF is started (see the sample member GTFCNTL). This parameter is valid only when TRACE="GTF" is specified. Specify a hexadecimal number of up to four bytes that GTF recognizes as a valid user trace record identifier.

Default 00E9

Supported Values

Value	Description
eid	A valid four byte hexadecimal number that GTF recognizes.

Example EID ="eid"

ENDEVOR_DSNQUALIFIER

Name ENDEVOR_DSNQUALIFIER

Summary Specifies a high-level qualifier for dynamically allocated data sets created by MFA Server to hold input files during the staging process. For those sites where the logged in users may not have the authority to create transient files using their USERID as the high-level qualifier, there is a configuration parameter to set that high-level qualifier to some other value where all users have READ/WRITE/ALTER access.

Default NULL

Supported Values

Value	Description
endevordsn	A valid data set HLQ.

Example ENDEVOR_DSNQUALIFIER="endevordsn"

ES-MTO_MAXTASKS

Name ES-MTO_MAXTASKS

Summary The number of z/OS subtasks (TCBs) to be started and dedicated to processing requests from ES/MTO clients. Specify 0 (zero) or a positive integer value between 1 and 50.

Default 0

Minimum Value 1

Maximum Value 50

Example ES-MTO_MAXTASKS="25"

ID

Name ID

Summary

ES-MTO: The ID can be one to four characters in length and must match the SYSID of the ES/MSS server being defined. The initial connection messages exchanged by MFA Server and ES/MSS use this ID.

IMS: If an IMS Option client request does not specify a target server ID or the specified target server ID does not exist, Mainframe Access selects the DEFAULT IMS target server, if one has been defined. This IMS target server ID name is used in the configuration of the client. When a Remote IMS request is sent this name is sent in the request data and is used to locate the target server definition for the IMS system that will receive the request. Specify DEFAULT or an ID of up to four alphanumeric characters.

MCO: If a CICS client request does not specify a target server ID or the specified target server ID does not exist, Mainframe Access selects the DEFAULT CICS target server, if one has been defined. This CICS target server ID name is used in the configuration of the client. When a CICS request is sent to Mainframe Access this name is sent in the request data and is used to locate the target server definition for the CICS system that will receive the request. Specify DEFAULT or an ID of up to four alphanumeric characters.

MFA: Data Connect client requests do not specify a target server ID and Mainframe Access always looks for the DEFAULT Mainframe Access Data Connect target server definition. Specify DEFAULT.

Supported Values

Value	Description
id	A valid ID according to the rules above.

Example ID ="id"

IMSLINK_MAXTASKS

Name IMSLINK_MAXTASKS

Summary The number of z/OS subtasks (TCBs) to be started and dedicated to processing requests from IMS Option (Remote IMS) clients. Specify 0 (zero) or a positive integer value between 1 and 50.

Default 0

Minimum Value 1

Maximum Value 50

Example IMSLINK_MAXTASKS="25"

IPADDRESS

Name IPADDRESS

Summary The internet host name or IP address of the ES/MSS server. Specify either the host name that resolves to the correct IP address or the actual IP address in standard dotted-decimal format. The combination of IPADDRESS and PORT from this definition provides MFA Server with the information it needs to originate a socket connection to ES/MSS and to verify the authenticity of a socket connection request from ES/MSS.

Default NULL

Supported Values

Value	Description
ipaddress	A valid IP address or logical host name.

Example IPADDRESS="ipaddress"

JES_BUFFER_TRACE

Name JES_BUFFER_TRACE

Summary Enables tracing of interactions between MFA and the JES subsystem. This may be useful for diagnostic purposes.

Default 0

Supported Values

Value	Description
0	Disable MFA JES tracing.
1	Enable MFA JES tracing.

Example JES_BUFFER_TRACE="1"

JES_CONCHAR

Name JES_CONCHAR

Summary Specifies the character that is used to identify JES2 commands from local consoles.

Default $

Supported Values

Value	Description
conchar	A valid CONCHAR character as defined in IBM manual *JES2 Initialization and Tuning Reference SA32-0992*.

Example JES_CONCHAR ="conchar"

JOBNAME

Name JOBNAME

Summary The jobname prefix to be used for address spaces that are started for this group. For multiple-instance address spaces this is a prefix of 1 to 4 characters and Mainframe Access Server will pad this prefix to a full 8-character jobname by appending a 4 to 7 digit sequence number. For example, JOBNAME="MFAE" results in jobnames MFAE0001, MFAE0002, and so forth. The sample prefix MFAE can be changed to meet the needs of your installation. If you do change the suggested prefix you need to review the security subsystem definition for the Mainframe Access started tasks. The configuration process uses a generic STARTED task definition (the generic name specified during Quick Configuration is MFA*.*) that covers generated jobnames such as MFAExxxx, in addition to the MFA (the MFA Server control region) and MFAS (the MFA Server for Data Connect) started task names. Specify a jobname prefix of up to four characters.

Supported Values

Value	Description
jobname	A alphanumeric string up to 4 characters long.

Example JOBNAME="jobname"

LIST_DB2

Name LIST_DB2

Summary Indicates whether or not the Mainframe Access LIST DB2 feature is to be activated. This obtains information about the DB2 and IMS subsystems. Specify 1 to activate LIST DB2 or 0 to deactivate it.

Default 0

Supported Values

Value	Description
0	Deactivate retrieval of DB2 and IMS subsystem information.
1	Activates retrieval of DB2 and IMS subsystem information.

Example LIST_DB2="1"

LIST_PARAMETERS

Name LIST_PARAMETERS

Summary Controls the listing of Mainframe Access parameter definitions on the XDBOUT sysout data set. The parameter listing is helpful when you need to examine the active configuration settings. Specify 1 to echo parameter definitions to the XDBOUT sysout data set or 0 to inhibit the listing.

Default 1

Supported Values

Value	Description
0	Do not echo parameter definitions to XDBOUT data set.
1	Echo parameter definitions to XDBOUT data set.

Example LIST_ PARAMETERS="0"

LU62_APPLID

Name LU62_APPLID

Summary The ACF/VTAM ACBNAME to be used by Mainframe Access. This ACBNAME must match the ACBNAME parameter of an active VTAM application major node. See *Editing Mainframe Access Server Definitions* for more information. Mainframe Access issues a VTAM OPEN request for this ACBNAME during start up and initialization fails if the OPEN is not successful. This ACBNAME (or possibly a different "network name" specified in the VTAM application major node definition) is the SNA LU name that identifies Mainframe Access to other applications in your SNA network. Specify the ACBNAME that has been defined and activated for Mainframe Access.

Default MFM62ACB

Supported Values

Value	Description
lu62applid	A valid VTAM APPLID.

Example LU62_APPLID ="lu62applid"

LUNAME

Name LUNAME

Summary

IMS: The LU name of the IMS server (also known as the ACBNAME or VTAM APPLID). Specify an LU name of up to eight characters. This name must match the ACBNAME defined in an APPC/MVS LU definition for the target IMS system.

MCO: The LU name of the CICS server (also known as the ACBNAME or VTAM APPLID). Specify an LU name of up to eight characters.

MFA: The LU name of the Mainframe Access Data Connect server (also known as the ACBNAME or VTAM APPLID). Specify an LU name of up to eight characters.

Default NULL

Supported Values

Value	Description
luname	A LU name of up to 8 characters.

Example LUNAME="luname"

MAX_DS_ALLOCATIONS

Name MAX_DS_ALLOCATIONS

Summary Controls the maximum number of simultaneous data set allocations (24-bit memory constraint, max 512).

Default 128

Minimum Value 128

Maximum Value 512

Example	MAX_DS_ALLOCATIONS="300"

MAX_PUBLIC_FILES

Name	MAX_PUBLIC_FILES
Summary	Specifies the maximum number of shared public files that Mainframe Access Data Set Services will keep available. When the maximum is reached the next request for a new public file is rejected, and you must close one of the item libraries to make space for the new one. Specify an integer value between 24 and 64. Defaults to 32.
Default	32
Minimum Value	24
Maximum Value	64
Example	MAX_PUBLIC_FILES="40"

MAXIMUM

Name	MAXIMUM
Summary	Specifies the maximum number of address spaces that Mainframe Access server starts for a multiple-instance group. Additional address spaces beyond the minimum is started in response to transaction load, up to the maximum allowed by this specification. Mainframe Access server issues messages (to the XDBOUT data set) when a client request must be queued to wait for an available processing address space. Increase the MAXIMUM value (by 1) when you observe frequent queueing of client requests. Specify a value from 1 to 10.
Default	0
Minimum Value	1
Maximum Value	10
Example	MAXIMUM="5"

MCOLINK_MAXTASKS

Name	MCOLINK_MAXTASKS
Summary	The number of z/OS subtasks (TCBs) to be started and dedicated to processing requests from CICS clients. Specify 0 (zero) or a positive integer value between 1 and 50.
Default	0
Minimum Value	1
Maximum Value	50
Example	MCOLINK_MAXTASKS="25"

MFA_ACCEPT_EMPTY_FILES

Name	MFA_ACCEPT_EMPTY_FILES
Summary	Indicates whether or not empty VSAM or VB data sets can be downloaded. Set to 1 to allow empty files to be downloaded, or set to 0 to restrict this. When the files are downloaded, they are converted to MF format using DFCONV. The default value is 1.

Default 1

Supported Values

Value	Description
0	Disallow downloading empty files.
1	Allow downloading empty files.

Example MFA_ACCEPT_EMPTY_FILES="0"

MFA_ENDEVOR_HISTORY

Name MFA_ENDEVOR_HISTORY

Summary This dynamically creates a cumulative Endevor transaction history file (MFALOGE) to house the access history of all file access attempts using Endevor. The Endevor API interface produces this log for each access requested.

Default 1

Supported Values

Value	Description
0	Do not create Endevor transaction history file MFALOGE.
1	Create Endevor transaction history file MFALOGE.

Example MFA_ENDEVOR_HISTORY="0"

MFA_ENDEVOR_INTERFACE

Name MFA_ENDEVOR_INTERFACE

Summary By default, the MFA Server offers access to Endevor services.

Default 1

Supported Values

Value	Description
0	Prevent MFA Server offering Endevor access.
1	Permit MFA Server offering Endevor access.

Example MFA_ENDEVOR_INTERFACE="0"

MFA_GUI_ACCEPT_PASSPHRASES

Name MFA_GUI_ACCEPT_PASSPHRASES

Summary Enables MFA Server to support 100 character passphrases.

Default 0

Supported Values

Value	Description
0	Disable passphrase support in MFA Server.
1	Enable passphrase support in MFA Server.

Example MFA_GUI_ACCEPT_PASSPHRASES="1"

MFA_GUI_BLANKPASSWORD

Name MFA_GUI_BLANKPASSWORD

Summary Set to 1 to clear the password input field on the Drag and Drop user dialog.

Default 0

Supported Values

Value	Description
0	Do not blank password input field on the Drag and Drop user dialog.
1	Blank password input field on the Drag and Drop user dialog.

Example MFA_GUI_BLANKPASSWORD="1"

MFA_GUI_IGNORUSERCTO

Name MFA_GUI_IGNOREUSERCTO

Summary Set to 1 to disable the inactivity timeout on the Drag and Drop user dialog.

Default 0

Supported Values

Value	Description
0	Enable the inactivity timeout on the Drag and Drop user dialog.
1	Disable the inactivity timeout on the Drag and Drop user dialog.

Example MFA_GUI_IGNOREUSERCTO="1"

MFA_JOBNAME_CHECK

Name MFA_JOBNAME_CHECK

Summary Set to 1 to check that the batch job name starts with the submitter's userid.

Default 0

Supported Values

Value	Description
0	Do not check batch job name start with submitter's userid.
1	Check batch job name start with submitter's userid.

Example MFA_JOBNAME_CHECK="1"

MFA_LIBRARIAN_DIR_INFO

Name MFA_LIBRARIAN_DIR_INFO

Summary LONG is the default. This directs MFA Server to request complete member information from Librarian. The LONG specification significantly increases response time when accessing Librarian master files containing over 1,000 members. Specify SHORT to reduce the amount of member information requested from Librarian.

Default LONG

Supported Values

Value	Description
SHORT	Only retrieve a small amount of member information from Librarian.
LONG	Retrieve a large amount of member information from Librarian.

Example MFA_LIBRARIAN_DIR_INFO="SHORT"

MFA_LIBRARIAN_HISTORY

Name MFA_LIBRARIAN_HISTORY

Summary By default, this will include a summary audit trail of all LIBRARIAN data accesses as part of the MFALOG output log.

Default 1

Supported Values

Value	Description
0	Do not create summary audit trail of Librarian data access in MFALOG.
1	Create summary audit trail of Librarian data access in MFALOG.

Example MFA_LIBRARIAN_HISTORY="0"

MFA_LIBRARIAN_INTERFACE

Name MFA_LIBRARIAN_INTERFACE

Summary By default, MFA Server offers access to Librarian services.

Default 1

Supported Values

Value	Description
0	Prevent MFA Server offering Librarian access.
1	Permit MFA Server offering Librarian access.

Example MFA_LIBRARIAN_INTERFACE="0"

MFA_LIBRARIAN_UPD_MODULE

Name MFA_LIBRARIAN_UPD_MODULE

Summary The name of the Librarian Batch update utility originally shipped as AFOLIBR by Computer Associates. If this module has not been renamed, then this parameter need not be specified.

Default AFOLIBR

Supported Values

Value	Description
updmodule	Name of the Librarian Batch update utility.

Example MFA_LIBRARIAN_UPD_MODULE="updmodule"

MFA_LISTING_DATA_CLASS

Name MFA_LISTING_DATA_CLASS

Summary Specifies an SMS data class used to create the Endevor listing data set. This enables the site to specify how large the data set may extend (allow for 1216 bytes per member transferred in one directory listing). Attributes to consider include: primary and secondary space allocations; number of volumes; the volumes where the data set is created.

Default NULL

Supported Values

Value	Description
mfadcls	A valid SMS data class name.

Example MFA_LISTING_DATA_CLASS="mfadcls"

MFA_MCG_DB2CONNECTION

Name MFA_MCB_DB2CONNECTION

Summary Controls which attachment facility is used by Mainframe Access Server for DB2 connections. Specify CAF (to use the DB2 call attach facility) or RRSAF (to use the DB2 Recoverable Resource Manager Services attachment facility). RRSAF is the default if this parameter is omitted.

Default RRSAF

Supported Values

Value	Description
CAF	Use DB2 Call attach facility.
RRSAF	Use DB2 Recoverable Resource Manager Services attachment facility.

Example MFA_MCB_DB2CONNECTION="CAF "

MFA_PANVALET_HISTORY

Name MFA_PANVALET_HISTORY

Summary By default, this will include a summary audit trail of all user accesses to Panvalet as part of the MFALOG output log.

Default 1

Supported Values

Value	Description
0	Do not create summary audit trail of Panvalet data access in MFALOG.
1	Create summary audit trail of Panvalet data access in MFALOG.

Example MFA_PANVALET_HISTORY="0"

MFA_PANVALET_INTERFACE

Name MFA_PANVALET_INTERFACE

Summary By default, the MFA Server offers access to Panvalet services.

Default 1

Supported Values

Value	Description
0	Prevent MFA Server offering Panvalet access.
1	Permit MFA Server offering Panvalet access.

Example MFA_PANVALET_INTERFACE="0"

MFA_PANVALET_UPD_MODULE

Name MFA_PANVALET_UPD_MODULE

Summary The name of the Panvalet Batch update utility originally shipped as PAN#1 by Computer Associates. If this module has not been renamed, then this parameter need not be specified.

Default PAN#1

Supported Values

Value	Description
updmodule	Name of the Panvalet Batch update utility.

Example MFA_PANVALET_UPD_MODULE="updmodule"

MFA_SAF_HISTORY

Name MFA_SAF_HISTORY

Summary By default, this will include a summary audit trail of all file access authorizations as part of the MFALOG output log.

Default 1

Supported Values

Value	Description
0	Do not create summary audit trail of file access in MFALOG.
1	Create summary audit trail of file access in MFALOG.

Example MFA_SAF_HISTORY="0"

MFA_SYSOUT_CLASS

Name MFA_SYSOUT_CLASS

Summary By default, this sets the SYSOUT class as "A" for any held data produced by Mainframe Access. This includes MFALOG, MFALOGE, and SNAPDUMP message files.

Default A

Supported Values

Value	Description
class	A valid job class character.

Example MFA_SYSOUT_CLASS="class"

MFA_SYSOUT_DEST

Name MFA_SYSOUT_DEST

Summary By default, the value LOCAL is used as the SYSOUT destination for any held data produced by Mainframe Access. This includes MFALOG, MFALOGE, and SNAPDUMP message files.

Default LOCAL

Supported Values

Value	Description
sysoutdest	A valid SYSOUT destination.

Example MFA_SYSOUT_DEST="sysoutdest "

MFADIRECT

Name MFADIRECT

Summary Indicates whether or not Mainframe Access Source Connect and Drag & Drop services are to be activated. Specify 1 to activate these services or 0 to disable these services. The default value is YES (the recommended setting).

Default 1

Supported Values

Value	Description
0	Deactivate Mainframe Access Source Connect and Drag & Drop services.
1	Activate Mainframe Access Source Connect and Drag & Drop services.

Example MFADIRECT="0"

MFADIRECT_MAXTASKS

Name MFADIRECT_MAXTASKS

Summary The number of z/OS subtasks (TCBs) to be started and dedicated to processing requests from Mainframe Access Drag & Drop and Source Connect clients, including Mainframe Access client functions and other development environment clients. Specify 0 (zero) or a positive integer value between 1 and 50. The default value is 5.

Default 5

Minimum Value 1

Maximum Value 50

Example MFADIRECT_MAXTASKS="25"

MFALINK_MAXTASKS

Name	MFALINK_MAXTASKS
Summary	The number of z/OS subtasks (TCBs) to be started and dedicated to processing requests from Mainframe Access Data Connect clients including Mainframe Access client functions originating in COBOL development environment clients. Specify 0 (zero) or a positive integer value between 1 and 50.
Default	0
Minimum Value	1
Maximum Value	50
Example	MFALINK_MAXTASKS="25"

MINIMUM

Name	MINIMUM
Summary	Specifies the maximum number of address spaces that Mainframe Access server will start for a multiple-instance group. Additional address spaces beyond the minimum is started in response to transaction load, up to the maximum allowed by this specification.
Default	0
Minimum Value	1
Maximum Value	10
Example	MINIMUM="5"

MODENAME

Name	MODENAME
Summary	**IMS:** The SNA log mode name that is used by Mainframe Access to indirectly specify SNA session parameters when Mainframe Access initiates sessions with the IMS server. Specify a log mode name for LU6.2 sessions of up to eight characters. This log mode name must be present in the VTAM log mode table available to Mainframe Access. IBM's default log mode table, ISTINCLM, typically provides several standard log modes that can be used by Mainframe Access, including both IBMRDB and #INTER LU6.2 log modes. **MCO:** The SNA log mode name that is used by Mainframe Access to indirectly specify SNA session parameters when Mainframe Access initiates sessions with mainframe CICS. Specify a log mode name for LU6.2 sessions of up to eight characters. This log mode name must be present in the VTAM log mode table available to Mainframe Access. IBM's default log mode table, ISTINCLM, typically provides several standard log modes that can be used by Mainframe Access, including both IBMRDB and #INTER LU6.2 log modes. **MFA:** The SNA log mode name that is used by Mainframe Access to indirectly specify SNA session parameters when Mainframe Access initiates sessions with the Mainframe Access Data Connect server. Specify a log mode name for LU6.2 sessions of up to eight characters. This log mode name must be present in the VTAM log mode table available to Mainframe Access. IBM's default log mode table, ISTINCLM, typically provides several standard log modes that can

be used by Mainframe Access, including both IBMRDB and #INTER LU6.2 log modes.

Default NULL

Supported Values

Value	Description
modename	A valid SNA log mode name that matches the VTAM definition.

Example MODENAME="modename"

NETWORK_ID

Name NETWORK_ID

Summary The SSCP network ID used by ACF/VTAM on this z/OS system. This parameter must match the NETID parameter in the active ATCSTRxx VTAM start parameters member of VTAMLST. This Network ID value is used by Mainframe Access to build an LUWID (SNA Logical Unit of Work ID for LU6.2) when an LU6.2 conversation is allocated to the DB2 DDF. Specify the 1 to 8 character Network ID from ACF/VTAM's startup parameters.

Default DDINET1

Supported Values

Value	Description
networkid	A valid VTAM NETID.

Example NETWORK_ID="networkid"

ORGANIZATION

Name ORGANIZATION

Summary Your company name or other meaningful identifier up to 40 characters in length. Spaces are not permitted. Use underscores or other non-blank characters to separate words if necessary.

Default YOUR_COMPANY_NAME

Supported Values

Value	Description
oraganization	A non spacey alphanumeric string up to 40 characters.

Example ORGANIZATION="organization"

PEM_FEATURE

Name PEM_FEATURE

Summary Indicates whether or not the Mainframe Access Password Expiration Manager (PEM) feature is to be activated.

PEM is used in some Micro Focus client products to enable users of the client software on PCs to change passwords for their z/OS user IDs without logging on directly to z/OS. This is especially useful when the security subsystem indicates that the current password is expired and must be replaced. Specify 1 (this setting is recommended) to activate the PEM feature or 0 to deactivate it.

Default 1

Supported Values

Value	Description
0	Disable clients from changing user passwords.
1	Enable clients to change user password.

Example PEM_FEATURE ="0"

PORT

Name PORT

Summary The port number where ES/MSS is listening for ISC connections.

Default 0

Minimum Value 1

Maximum Value 65525

Example PORT="3030"

PROCEDURE

Name PROCEDURE

Summary The name of the started task JCL procedure that can be used to start address spaces for this group. Sample JCL procedure MFAAS provides the basic JCL for an application server address space, and MFAAMS for the AMS application server address space. Specify a JCL procedure name of up to eight characters.

Supported Values

Value	Description
procedure	A valid JCL procedure name up to eight characters.

Example PROCDURE="procedure"

RACF_APPLID

Name RACF_APPLID

Summary Specifies the use of passtickets with functions that require a password.

Default NULL

Supported Values

Value	Description
NULL	MFA Server uses a userid and password to protect the system.
applname	MFA Server uses a password or passticket to protect the system.

Example RACF_APPLID ="applname"

ROUTE

Name ROUTE

Summary The message route codes to be used by Mainframe Access when console messages are issued using the z/OS WTO (Write to Operator) and WTOR (Write to Operator with Reply) services. See your IBM system documentation for valid values.

Default (2,1)

Supported Values

Value	Description
routcde	A valid ROUTCDE.

Example ROUTE="routcde"

SECURITY

Name SECURITY

Summary

IMS: The security level to be indicated in SNA LU6.2 Attach FMH-5 requests sent to the IMS server. Mainframe Access forwards security subfield information as provided by the IMS Option client and sets the FMH-5 fields accordingly. Specify 0.

MCO: The security level to be indicated in SNA LU6.2 Attach FMH-5 requests sent to the CICS server. Mainframe Access forwards security subfield information as provided by the CICS client and sets the FMH-5 fields accordingly. Specify 0.

MFA: The security level to be indicated in SNA LU6.2 Attach FMH-5 requests sent to the Mainframe Access Data Connect server. Mainframe Access forwards security subfield information as provided by the Data Connect client and sets the FMH-5 fields accordingly. Specify 0.

Default 0

Supported Values

Value	Description
0	Disable security on LU6.2 connections.
1	Enable security on LU6.2 connections.

Example SECURITY="1"

SESSIONS

Name SESSIONS

Summary Specifies the number of concurrent conversations MFA Server can initiate to the ES/MSS server over a single socket connection. If ES/MSS has a different definition for the number of concurrent sessions, the session count is negotiated to a common value at the time a socket connection is established. Specify a value that supports the required number of concurrently active conversations for your application.

Default 4

Minimum Value 1

Maximum Value 8

Example SESSIONS="5"

SMF_RECORDID

Name SMF_RECORDID

Summary The SMF user record ID that Mainframe Access should use when writing SMF (IBM's System Management Facility) records to the z/OS SMF data sets. If an SMF user record ID is specified, that ID value should be assigned to Mainframe Access exclusively. Specify a valid SMF user record ID number to activate the Mainframe Access SMF support. Specify 0 (zero), or omit the parameter altogether, to disable the writing of SMF records.

Default 0

Supported Values

Value	Description
0	Disable writing of SMF records.
smfrecordid	A valid SMF user record ID number.

Example SMF_RECORDID="smfrecordid"

SOCKETS

Name SOCKETS

Summary Specifies the maximum number of concurrent socket connections between MFA Server and the ES/MSS server. ES/MSS support is currently limited to a single socket connection between MFA Server and an ES/MSS server, specify a value of 1.

Default 1

Minimum Value 1

Maximum Value 1024

Example SOCKETS="5"

SYNCLEVEL

Name SYNCLEVEL

Summary

IMS: The SNA LU6.2 sync level option to be used on conversations with the IMS server. Specify 0.

MCO: The SNA LU6.2 sync level option to be used on conversations with CICS. Specify 0.

MFA: The SNA LU6.2 sync level option to be used on conversations with the Mainframe Access Data Connect server. Specify 1. Conversations use LU6.2 CONFIRM protocols.

Default NULL

Minimum Value 0

Maximum Value 2

Example SYNCLEVEL="1"

SYSOUT

Name SYSOUT

Summary The SYSOUT class for the XDBOUT data set if it is dynamically allocated. Mainframe Access operational messages and trace data (when TRACE="SYSPRINT" is specified and tracing is active) are written to this data set. XDBOUT is dynamically allocated to this SYSOUT class if an XDBOUT DD statement is not present in the Mainframe Access JCL. Specify a SYSOUT class designator that is valid for your z/OS system.

Default A

Supported Values

Value	Description
class	A valid SYSOUT class character.

Example SYSOUT="class"

TCP_PORT

Name TCP_PORT

Summary The port number to accept connections from Micro Focus clients.

Default 2020

Minimum Value 1

Maximum Value 65525

Example TCP_PORT="3030"

TCPLINK_MAXTASKS

Name TCPLINK_MAXTASKS

Summary The number of z/OS subtasks (TCBs) to be started and dedicated to processing requests from SQL Option for DB2 clients. Specify a positive integer value between 1 and 50.

Default 2

Minimum Value 1

Maximum Value 50

Example TCPLINK_MAXTASKS="25"

TIMEOUT_FOR_CONNECTION

Name TIMEOUT_FOR_CONNECTION

Summary The number of minutes that a client connection can remain idle, after which the connection will be broken and Mainframe Access resources dedicated to the client will be released. Resources that are released when an idle client is disconnected include allocated storage, TCP/IP resources and connections to server subsystems such as DB2, IMS, CICS and Mainframe Access. Specify 0 (zero) to disable the client timeout feature and allow unlimited idle time, or a positive integer value.

Default 30

Minimum Values 0

Maximum Value 35791394

Example TIMEOUT_FOR_CONNECTION="25"

TIMEOUT_FOR_INITIAL_RECEIVE

Name TIMEOUT_FOR_INITIAL_RECEIVE

Summary The number of seconds that the listener process (Micro Focus client listener or Web Server listener) waits to receive the initial client message after a socket connection has been accepted from a new client. If the time value expires before a complete initial message is received, the client is disconnected and the listening process prepares to accept another client connection request. The default value of 3 seconds is appropriate for all installations. This low value helps keep the listeners running smoothly, even during occasional network and/or client malfunctions. Specify a positive integer value between 1 and 59 (inclusive) to alter the time out value. This may be useful for diagnostic purposes.

Default 10

Minimum Value 1

Maximum Value 59

Example TIMEOUT_FOR_INITIAL_RECEIVE="25"

TPNAME

Name TPNAME

Summary

IMS: The IMS server transaction program name for IMS Option transactions. This is the LU6.2 transaction program name that is sent to the IMS server in SNA Attach FMH-5 requests to begin an IMS Option transaction. Specify the APPC/MVS transaction program name that was specified in the APPC/MVS definitions during installation of Remote IMS.

MCO: The server transaction program name for CICS requests. This is the LU6.2 transaction program name that is sent to CICS in SNA Attach FMH-5 requests to begin a CICS transaction. Specify a transaction program name of up to eight characters or specify * to cause Mainframe Access to use the transaction program name provided by the CICS client. The CICS client prepares a partial FMH-5 request that specifies the standard CICS transaction program names for function shipping, distributed program linking and so on.

MFA: The transaction program name for Mainframe Access Data Connect server requests. Specify FILESHR2.

Default NULL

Supported Values

Value	Description
tpname	A valid LU6.2 transaction program name.

Example TPNAME="tpname"

TRACE

Name TRACE

Summary The destination for trace data when tracing is activated:

- Specify SYSPRINT to send print formatted trace output to the data set identified by the XDBOUT DD statement, normally a SYSOUT data set. Micro Focus recommends this setting during initial product testing and when you are performing controlled problem determination.
- Specify GTF to trace a very active Mainframe Access that is processing a high transaction volume. Trace data collected by GTF is written to external storage (tape or DASD) in a raw data format while tracing is active. You can use IBM's IPCS (Interactive Problem Control System) to format the data for analysis after tracing has been completed. Trace records are sent to the console if GTF is not available when TRACE=GTF is specified.
- Specify CONSOLE to send trace output to the system log and console (not recommended because of the large volume of messages).

Default SYSPRINT

Supported Values

Value	Description
CONSOLE	Send trace to system log.
GTF	Send trace to GTF.
SYSPRINT	Send trace to SYSPRINT identified by XDBOUT DD statement.

Example TRACE="GTF"

TRACING

Name TRACING

Summary Specifies whether or not detailed tracing of Mainframe Access Server program activity should be started during initialization processing. Normally, tracing should be turned off during initialization and during normal product operation. Activity tracing writes detailed information to the XDBOUT sysout data set and this can slow down performance in a busy server. Tracing is normally controlled by the TRACE ON and TRACE OFF commands after initialization is complete. The TRACING parameter makes it possible to trace the initialization activity, before the server is ready to accept TRACE ON/OFF commands. Activity trace started by the TRACING parameter can be stopped later using the TRACE OFF command. Specify 1 to activate tracing during initialization or specify 0 to inhibit activity tracing until a TRACE ON command is issued.

Default 0

Supported Values

Value	Description
0	Disable MFA Server tracing.
1	Enable MFA Server tracing.

Example TRACING="1"

TSO_JOBCHAR

Name TSO_JOBCHAR

Summary Specifies a character which is appended to the user's TSO user id to form a job name for the TSO Command Server task.

Default X

Supported Values

Value	Description
alphanumeric	A character in the range A-Z, 0-9.

Example TSO_JOBCHAR="X"

TSO_PROCEDURE

Name TSO_PROCEDURE

Summary Specifies the name of the JCL procedure used to run the TSO Command Server.

Default MFATSO

Supported Values

Value	Description
Name	A valid JCL procedure name.

Example TSO_PROCEDURE="MFATSO"

VIO_FOR_ENDEVOR_LOG

Name VIO_FOR_ENDEVOR_LOG

Summary By default, the Endevor Dependent Region creates a new VIO file (DDNAME=ENDVMSG) to be used as the Endevor transaction log for all subsequent transactions within that dependent region. The DCB attributes are:

```
PS,FBA,LRECL=133,UNIT=VIO,DISP=(NEW,DELETE),DSN=&MSGLOG
```

This file is effectively be a memory resident file, buffered by JES.

Note: During initialization there is no end-user profile in effect. That means the VIO file is owned by the started task. Therefore, the started task security profile must allow any end-user to write to this file. Endevor will OPEN the file as the transaction log. This is the default behaviour.

If this causes a security violation at your site, you may configure the log to be disk-resident, provided the high-level qualifier allows universal READ/WRITE access to the log. This is allocated as:

```
UNIT=SYSALLDA,DSNAME=prefix.jobname.ASnnnnn.MSGLOG
```

Where the jobname and address space number ensures uniqueness. The prefix is taken from the DSNQUALIFIER_ENDEVOR parameter

Default 1

Supported Values

Value	Description
0	MFA Server allocates a data set for the Endevor transaction log.
1	MFA Server uses the default VIO file for Endevor transaction log stored in ENDVMSG.

Example VIO_FOR_ENDEVOR_LOG="0"

Mainframe Access Server z/Server feature configuration parameters (deprecated)

Note: These features are deprecated, and provided for backward compatibility only.

Configuration parameters - quick reference

The following list shows all z/Server configuration parameters and for each parameter, indicates whether it applies to general, schedulers, or user servers.

Parameter	General	Scheduler	User Server
AUTOSTART	Y		
BUFFERSIZE		Y	Y
CCSID		Y	Y
CEA_ACCOUNT		Y	
CEA_CHARSET		Y	
CEA_CODEPAGE		Y	
CEA_INIT_CMD		Y	
CEA_LOGONPROC		Y	
CEA_REGION_SIZE		Y	
CEA_SCR_SIZE		Y	
DEF_USRSRV_MODE			
DELAY	Y		
DIALOG_TIMEOUT			Y
DSP_PROTECT	Y		
DSP_TOKEN	Y		
ECB_TIMEOUT		Y	Y
FIRST_PORT		Y	
IPSTACK	Y		
IPTRACE	Y	Y	Y
ISPF_STATS		Y	
JOBNAME		Y	
LAST_PORT		Y	
LISTENER_PORT		Y	
LOGON_EXIT		Y	Y
MAIL_CLASS		Y	Y
MAIL_MSGCNT		Y	Y
MAIL_NOTIFY		Y	Y
MAIL_SEVERITY		Y	Y
MAIL_WRITER		Y	Y
MAXUSRV		Y	

Parameter	General	Scheduler	User Server
MSGCLASS		Y	
MULTI_REXX_S		Y	
MULTI_REXX_U			Y
NOFTASK		Y	Y
NUMTCB		Y	Y
PORTCHECK		Y	Y
RACF_APPLID	Y		
RACF_STATS		Y	Y
SCHEDULER_NAME		Y	
SUBMIT_DELAY		Y	
SVC_NO	Y		
TIMEOUT		Y	Y
TSOE_JOBCHAR		Y	
TSOE_NOTIFY		Y	
USER_SERVER_JOBNAME		Y	
USRSRV_TIMEOUT		Y	
USS_DUBPROCESS		Y	Y
VIPA		Y	
VPOOLSIZE		Y	Y

Configuration parameters - alphabetical list

AUTOSTART

Name AUTOSTART

Summary Specifies whether the holder address space will automatically start the scheduler address space(s) associated with it.

Default 1

Supported Values

Value	Description
0	The holder address space does not automatically start the scheduler address space(s) associated with it. Each scheduler address space must be started manually using an MVS START command.
1	The holder address space automatically starts the scheduler address space(s) associated with it.

Example AUTOSTART="0"

BUFFERSIZE

Name	BUFFERSIZE
Summary	Specifies the size of the IP send/receive buffer for each worker task in a scheduler or user server address space.
Default	10M
Minimum Value	2M (or 2048K)
Maximum Value	100M
Supported Values	An integer followed by either the character "K" or "M". There must be no space between the integer and the character.
Additional Information	The size of a message to or from a client cannot exceed the defined buffer size. The buffers are used for dataset content to be edited or JESx data to be browsed and must be large enough to contain the associated data. For example, a 20.000 line source program requires a 1.6MB buffer. The buffers are allocated in LE heap storage (above the line). The CMDTASK DD statement contains the confirmation of the buffer sizes: TAU0025I hh:mm:ss.ttt Inp-Msg-Area for subtask allocated at 29649028 Out-Msg-Area for subtask allocated at 2A04A028 Wrk-Msg-Area for subtask allocated at 2AA4B028 Each area is 10485760 Bytes long If the defined buffers are too small for an incoming or outgoing request message, TAU0050E is written to the LE message file: TAU0050E hh:mm:ss.ttt Input-Msg-Len with 00001030 Bytes is greater than ... Increase Buffer-Size and restart server !
Example	BUFFERSIZE="15M"

CCSID

Name	CCSID
Summary	Specifies the code page used by z/Server (for example, 37 for US, or 1141 for German).
Default	37
Supported Values	Any of the code pages supported by z/Server: 37, 273, 277, 278, 280, 284, 285, 297, 500, 871, 1047, 1140, 1141, 1142, 1143, 1144, 1145, 1146, 1147, 1148, 1149
Example	CCSID="1047"

CEA_ACCOUNT

Name	CEA_ACCOUNT
Summary	Specifies the name of the account to be used by all users of this scheduler.
Default	ACCT#
Supported Values	A character string with maximum length of 40 characters. The string can contain any alphanumeric characters or any special characters supported by the current codeset/codepage.
Additional Information	The account specified by CEA_ACCOUNT must be defined in general resource class TSOACCT, and all potential TSO users must have access to it. The configured default is ACCT#.
Example	CEA_ACCOUNT="ACCT#"

CEA_CHARSET

Name	CEA_CHARSET
Summary	Specifies the character set used for the caller's CEA-launched TSO address space.
Default	697
Supported Values	Any character set supported by z/OS.
Additional Information	The value specified by CEA_CHARSET is used by the applications running in the TSO/E address space to convert messages and responses from UTF-8 to EBCDIC.
Example	CEA_CHARSET="698"

CEA_CODEPAGE

Name	CEA_CODEPAGE
Summary	Specifies the codepage used for the caller's CEA-launched TSO address space.
Default	37
Supported Values	Any codepage supported by z/OS.
Additional Information	The value specified by CEA_CODEPAGE is used by applications running in the TSO/E address space to convert messages and responses from UTF-8 to EBCDIC.
Example	CEA_CODEPAGE="1141"

CEA_INIT_CMD

Name	CEA_INIT_CMD
Summary	Specifies the REXX exec used for a CEA initial request or split screen request.
Default	ZCEAICMD
Supported Values	A character string with maximum length of 8 characters. The first character of the string must be alphabetic. The remaining characters must be alphanumeric.
Additional Information	A CEA initial request should be the first application requested by a client after a CEA-launched TSO user address space is started.
Example	CEA_INIT_CMD="ZCEAIREX"

CEA_LOGONPROC

Name	CEA_LOGONPROC
Summary	Specifies the name of the TSO/E logon procedure used to start a CEA-launched TSO address space.
Default	CEAPROC
Supported Values	A character string with maximum length of 8 characters. The first character of the string must be alphabetic. The remaining characters must be alphanumeric.
Additional Information	The procedure specified by CEA_LOGONPROC must be defined to general resource class TSOPROC and potential TSO users must have access to it. If more than one CEA scheduler is run on a given system, every CEA scheduler needs its own logon

procedure, because the IPCONFIG DD statement contained in this procedure will be different for every scheduler.

The logon procedure should be tested like any TSO logon procedure before using it. When it is used outside of z/Server, the initial REXX exec CEALOGON will not have run, resulting in no ISPF profile, and the TSO user being returned to the TSO READY prompt.

Use the following command to allocate the ISPF profile (with the correct naming conventions for the installation):

```
alloc dd(ISPPROF) dsn(userid.ISPF.ISPPROF)
```

Then call ISPF. If the logon procedure is correct, the ISPF primary option menu will be shown. Otherwise correct any errors.

This is a mandatory parameter for CEA scheduler address spaces.

Example	CEA_LOGONPROC="CEAPROC"

CEA_REGION_SIZE

Name	CEA_REGION_SIZE
Summary	Specifies the region size used to start the CEA-launched TSO user address space.
Default	2047M
Minimum Value	150M
Maximum Value	2047M
Supported Values	An integer followed by the character "M". There must be no space between the integer and the "M".
Additional Information	Any specification provided in the JCL statement will be overwritten by this parameter.
Example	CEA_REGION_SIZE="1000000"

CEA_SCR_SIZE

Name	CEA_SCR_SIZE
Summary	Describes the screen characteristics for the ISPF session.
Default	24x80
Supported Values	24x80, 32x80, 43x80, 27x132, or 62x160
Additional Information	These screen sizes are usually set in a 3270 emulation and queried by VTAM at the start of a 3270 TSO session. The values specified here are used if the Eclipse client did not specify other screen sizes in the preferences section, which would then be used during the start of a CEA-launched TSO user address space.
Example	CEA_SCR_SIZE="62x160"

DEF_USRSRV_MODE

Name	DEF_USRSRV_MODE

Summary	Specifies the default working mode of a user server started by a scheduler. This is used when connecting with legacy clients.
Default	STC
Supported Values	STC, CEA, JOB
Additional Information	
Example	DEF_USRSRV_MODE=CEA

DELAY

Name	DELAY
Summary	Specifies the time, in seconds, that the holder task waits until canceling a user server/scheduler when the stop command didn't terminate the address space and the time between terminating the last scheduler task and the holder task itself.
Default	5
Minimum Value	5
Maximum Value	300
Additional Information	When specifying a value for DELAY, bear in mind that the time necessary to terminate an address space can vary with the CPU load on the system.
Example	DELAY="10"

DIALOG_TIMEOUT

Name	DIALOG_TIMEOUT
Summary	Specifies the timeout, in seconds, for dialog responses.
Default	3600
Minimum Value	20
Maximum Value	86400
Additional Information	A dialog response in this context is the time a user server address space waits for a client response. After the specified time expires, a running ISPF dialog (in a user server address space) receives an error message and must be restarted from the client. z/Server will cancel the user server address space that had encountered the timeout. When a scheduler address space does not get a timely response from a user server, the following message is issued: `SLR0084E hh:mm:ss.ttt User server [002] did not response for [003] seconds. Connection terminated due to timeout!`
Example	DIALOG_TIMEOUT="40"

DSP_PROTECT

Name	DSP_PROTECT
Summary	Specifies whether the data space should reside in protected memory.
Default	Y

Supported Values	Y or N
Additional Information	Micro Focus strongly recommends that you set DSP_PROTECT="Y" in the holder address space configuration.
Example	DSP_PROTECT="Y"

DSP_TOKEN

Name	DSP_TOKEN
Summary	Specifies the token name that each address space for one z/Server uses to address the data space with the user administration control structures.
Default	TAURSERV
Supported Values	A character string with maximum length of 16 characters. The first character must be in the range "J" through "Z". The remaining characters must be alphanumeric. The first three characters of the string must not be "SYS".
Example	DSP_TOKEN="TAURSERV"

ECB_TIMEOUT

Name	ECB_TIMEOUT
Summary	Specifies, in seconds, how long the listener waits for the completion of the command and service task.
Default	30
Minimum Value	10
Maximum Value	120
Additional Information	If a timeout occurs, message TAU0174I is issued and the server startup terminates.

FIRST_PORT

Name	FIRST_PORT
Summary	Specifies the low end of the port range that the scheduler address space owns and assigns to a user server address space when the user server is started for IP communication.
Default	1201
Minimum Value	1
Maximum Value	65535
Additional Information	Specify the high end of the port range using LAST_PORT. The port range (LAST_PORT - FIRST_PORT) specifies the maximum number of user server address spaces that can run in parallel.
Example	FIRST_PORT="1240"

IPSTACK

Name	IPSTACK
Summary	Specifies the name of the IP stack used for processing.

Default TCPIP

Supported Values A character string with maximum length of 8 characters. The first character of the string must be alphabetic. The remaining characters must be alphanumeric.

Additional Information This parameter is mandatory.

Example IPSTACK="TCPIP"

IPTRACE

Name IPTRACE

Summary Specifies the level of tracing to use.

Default 0

Supported Values

Value	Description
0	Normal message traffic for best performance.
1	Warning, error and severe error messages are written.
2	Informational messages are also written.
3	All data sent and received using IP is traced.
4	Calls to EZASOCKET are traced.
5	Full trace.
6	Code page translation and BASE64 encode/decode are traced.

Additional Information The IPTRACE level can be set at startup. The trace level can later be changed via operator command. A higher trace level includes all lower trace levels.

Note: A high trace level may cause JESx spool shortages due to the amount of output written to the JESx job log. Use higher trace levels with caution.

Example IPTRACE="3"

ISPF_STATS

Name ISPF_STATS

Summary Specifies the level of ISPF statistics that z/Server returns.

Default 1

Supported Values

Value	Description
0	z/Server returns basic ISPF statistics on read requests to be displayed in the client.
1	z/Server returns basic ISPF statistics on read requests to be displayed in the client, and writes/updates basic ISPF statistics when writing/updating a member of a partitioned dataset.

Additional Information Extended ISPF statistics are not supported. If an installation uses extended ISPF statistics, they will be replaced with basic ISPF statistics.

	The STATS setting in the ISPF profile of the TSO user is ignored.
Example	ISPF_STATS="0"

JOBNAME

Name	JOBNAME
Summary	Specifies whether the JOBNAME of a user server consists of the user ID prefixed or suffixed with TSOE_JOBCHAR.
Default	PREFIX
Supported Values	PREFIX, SUFFIX

LAST_PORT

Name	LAST_PORT
Summary	Specifies the high end of the port range that the scheduler address space owns and assigns to a user server address space when the user server is started for IP communication.
Default	1249
Minimum Value	1
Maximum Value	65535
Additional Information	Specify the low end of the port range using FIRST_PORT. The port range (LAST_PORT - FIRST_PORT) specifies the maximum number of user server address spaces that can run in parallel.
Example	LAST_PORT="1270"

LISTENER_PORT

Name	LISTENER_PORT
Summary	Specifies the IP port that the scheduler listens on for incoming work requests.
Default	1200
Minimum Value	1
Maximum Value	65535

LOGON_EXIT

Name	LOGON_EXIT
Summary	Specifies the user exit that is invoked every time an ACEE for a user is created or deleted.
Default	SPACE
Supported Values	A character string with maximum length of 8 characters. The first character of the string must be alphabetic. The remaining characters must be alphanumeric.
Additional Information	The exit specified LOGON_EXIT can change or set the userid and password. The exit may also be used to generate a pass ticket for a given user ID.

MAIL_CLASS

Name MAIL_CLASS

Summary Specifies the JESx message class that the emails are forwarded to.

Default F

Supported Values A single-character string. The character must be alphanumeric.

Additional Information The message class specified by MAIL_CLASS must match the message class that is configured as LOCALCLASS for the SMTP server.

Example MAIL_CLASS="G"

MAIL_MSGCNT

Name MAIL_MSGCNT

Summary Specifies how many messages are to be combined into one email.

Default 20

Minimum Value 1

Maximum Value 100

Additional Information At scheduler shutdown, any messages buffered are always sent when the server is shut down.

Example MAIL_MSGCNT="25"

This example results in the 26th message causing the previously accumulated 25 messages to be sent as one email.

MAIL_NOTIFY

Name MAIL_NOTIFY

Summary Specifies whether notification emails are to be sent about errors with a severity greater than or equal to the value specified using the MAIL_SEVERITY parameter.

Default 0

Supported Values

Value	Description
0	Email notifications are deactivated.
1	Email notifications are activated.

Example MAIL_NOTIFY="1"

MAIL_SEVERITY

Name MAIL_SEVERITY

Summary Specifies the severity of the messages that cause an email to be sent.

Default 2

Supported Values

Value	Severity
1	Warnings
2	Errors

Value	Severity
3	Severe errors

Example	MAIL_SEVERITY="3"

MAIL_WRITER

Name	MAIL_WRITER
Summary	Specifies the name of the z/OS task address space that emails are forwarded to.
Default	SMTP
Supported Values	A character string with maximum length of 8 characters. The first character of the string must be alphabetic. The remaining characters must be alphanumeric.
Example	MAIL_WRITER="SMTP"

MAXUSRV

Name	MAXUSRV
Summary	Specifies the maximum number of user server address spaces that can be started for one TSO user or client.
Default	5
Minimum Value	1
Maximum Value	5000
Example	MAXUSRV="5"

MSGCLASS

Name	MSGCLASS
Summary	Specifies the JESx message class to be used for the job log DD statements that are dynamically created.
Default	X
Supported Values	A single-character string. The character must be alphanumeric.
Example	MSGCLASS="K"

MULTI_REXX_S

Name	MULTI_REXX_S
Summary	Specifies if re-entrant REXX environments are to be made available in addition to the non re-entrant REXX environment that is created by default.
Default	1

Supported Values

Value	Description
0	No re-entrant REXX environments are available. REXX execs executing in the default environment write their output to DD statement SYSTSPRT in JESx job log.

Value	Description
1	Each REXX exec is executed in its own REXX environment. Every z/Server subtask has both a non re-entrant and a re-entrant REXX environment available. REXX execs executing in the re-entrant environment write their output to DD statement ZCOTSPRT in JESx job log. This setting is recommended for scheduler address spaces.
2	CMDTASK and SRVTASK execute in a re-entrant REXX environment. All other subtasks share the default non re-entrant environment.

Example MULTI_REXX_S="2"

MULTI_REXX_U

Name MULTI_REXX_U

Summary Specifies if re-entrant REXX environments are to be made available in addition to the non re-entrant REXX environment that is created by default.

Default 2

Supported Values

Value	Description
0	No re-entrant REXX environments are available. REXX execs executing in the default environment write their output to DD statement SYSTSPRT in JESx job log.
1	Each REXX exec is executed in its own REXX environment. Every z/Server subtask has both a non re-entrant and a re-entrant REXX environment available. REXX execs executing in the re-entrant environment write their output to DD statement ZCOTSPRT in JESx job log.
2	CMDTASK and SRVTASK execute in a re-entrant REXX environment. All other subtasks share the default non re-entrant environment. This setting is recommended for a user server address space.

Example MULTI_REXX_U="1"

NOFTASK

Name NOFTASK

Summary Specifies the behavior of LISTENER task when no worker task is available to handle incoming client requests.

Default For CEA schedulers, M.

For STC schedulers, WM.

Supported Values

Value	Description
M	If no worker task is available, LISTENER returns message TAU0171E to the client. It does not wait for a worker task to become available.
W	If no worker task is available, LISTENER waits until one becomes available.
WM	If no worker task is available, LISTENER waits until one becomes available. Additionally, message TAU0009E is written to the hardcopy log.

Example NOFTASK="WM"

NUMTCB

Name NUMTCB

Summary Specifies how many worker tasks (TCBs) are started simultaneously to process incoming connections from a client.

Default For schedulers, 5.

For user servers, 1.

Minimum Value 1

Maximum Value 35

Additional Information Each worker task has its own DD statement Tnnnnnnn in the JESx job log. For every worker task started, the main task writes message TAU0010I to the MAINTASK DD statement:

```
TAU0010I 14:47:07.563 Subtask 1 started A(TCB) : 008B33B0
A(ECB) :
2236C0E0
TAU0010I 14:47:07.563 Subtask 2 started A(TCB) : 008B3120
A(ECB) :
2236C144
TAU0010I 14:47:07.563 Subtask 3 started A(TCB) : 0089EE78
A(ECB) : 2236C1A8
```

This message contains the address of the TCB (Task Control Block) and ECB (Event Control Block) for the worker task. When a worker task is posted for an incoming request, message TAU0011I is written to the MAINTASK DD statement:

```
TAU0011I 15:10:17.883 Subtask 1 posted A(TCB) : 008B33B0
A(ECB) : 2236C0E0
```

When all worker tasks are busy processing client requests and no worker task can be posted for a new client request, TAU0009E is written to the MAINTASK DD statement and the main task waits for one of the worker tasks to become available again:

```
TAU0009E 15:10:20.882 No free ECB available. Waiting for next
free ECB.
```

If this message appears too often, the number of worker tasks NUMTCB should be increased.

Note: Each subtask requires private storage to send and receive data (see *BUFFERSIZE*), and the storage required for the application. Storage is allocated when the worker task is established at scheduler address space startup.

The number of tasks is calculated by the formula: NUMTCB + 4 (CMDTASK + SRVTASK + MSGTASK + LISTENER).

Do not set NUMTCB higher than 33 (when using BUFFERSIZE="10M", as the size of the private region will be insufficient otherwise.

PORTCHECK

Name PORTCHECK

Summary Specifies whether the availability of a port should be tested before a user server address space is started.

Default 1

Supported Values

Value	Description
0	No checking is done.
1	z/Server issues the following command (visible in SYSTSPRT DD statement): `D TCPIP,,N,CONN,PORT` The command response can be found in the hardcopy log. When the port is not available, for instance, when it is in status LISTEN, TIMEWT, or FINWT2, the response to the DISPLAY command is useful for analysis.

Additional Information It is strongly recommended to set PORTCHECK="1" in the scheduler address space configuration.

Example PORTCHECK="1"

RACF_APPLID

Name RACF_APPLID

Summary Specifies the use of passtickets with functions that require a password.

Default NULL

Supported Values

Value	Description
NULL	The holder address space uses a userid and password to protect the system.
applname	The holder address space uses a password or passticket to protect the system.

Additional Information

Permission to generate a passticket is granted to a userid of the scheduler by using IRRPTAUTH profiles in the PTKTDATA class:

```
RDEFINE PTKTDATA applname SSIGNON(KEYMASKED(secure-signon-key)) UACC(NONE)
```

```
RDEFINE PTKTDATA IRRPTAUTH.applname.* UACC(NONE)
```

```
PERMIT IRRPTAUTH.applname.* CLASS(PTKTDATA) ACCESS(UPDATE) ID(userid)
```

Example RACF_APPLID=*applname*

RACF_STATS

Name RACF_STATS

Summary Specifies whether message ICH70001I is suppressed.

Default 0

Supported Values

Value	Description
0	Message ICH70001I is not suppressed.
1	Message ICH70001I is suppressed for RACROUTE and RACINIT REQUEST=VERIFY

Additional Information Message ICH70001I is issued whenever a client connects a scheduler with a request for service.

Example RACF_STATS="0"

SCHEDULER_NAME

Name SCHEDULER_NAME

Summary Specifies the job start of the started task.

Default TAURISPF

Supported Values A character string with maximum length of 8 characters. The first character of the string must be alphabetic. The remaining characters must be alphanumeric.

Example SCHEDULER_NAME="TAURISPF"

SUBMIT_DELAY

Name SUBMIT_DELAY

Summary Specifies the time, in seconds, until a scheduler address space checks if a just-started user server is ready to accept commands.

Default 5

Minimum Value 1

Maximum Value 30

Additional Information

The check is repeated 10 times before an error message similar to the following is issued:

```
SLR0131E hh:mm:ss.ttt User-Server did not signal READY after
a wait period of 10 X [002] milliseconds.
Command was not routed to the user server.
Please wait a moment and retry
```

If address space creation on a system routinely takes longer than 10 seconds, SUBMIT_DELAY should be increased.

Example SUBMIT_DELAY="25"

SVC_NO

Name	SVC_NO
Summary	Specifies the SVC number for the type-3 SVC routine to be dynamically installed at holder address space startup.
Default	238
Minimum Value	200
Maximum Value	255
Additional Information	This parameter is mandatory.
Example	SVC_NO="238"

TIMEOUT

Name	TIMEOUT
Summary	Specifies the length of time, in minutes, that the address space tolerates without IP activity before terminating.
Default	For user servers started as STC, 20. For all other servers, 0 (no timeout).
Minimum Value	0
Maximum Value	1440
Additional Information	This parameter has similar a similar effect to the JCL TIME parameter for CPU consumed.
Example	TIMEOUT="20"

TSOE_JOBCHAR

Name	TSOE_JOBCHAR
Summary	Specifies the job character for JOBNAME.
Default	W
Supported Values	A single-character string. The character must be alphabetic.

TSOE_NOTIFY

Name	TSOE_NOTIFY
Summary	Specifies whether a notify statement is generated in the start command for the STC user server.

Default 0

Supported Values

Value	Description
0	Do not generate a notify statement.
1	Generate a notify statement.

Example TSOE_NOTIFY="1"

USER_SERVER_JOBNAME

Name USER_SERVER_JOBNAME

Summary Specifies the name of the user server start procedure used in a z/OS start command.

Default IVPUSRT

Supported Values A character string with maximum length of 8 characters. The first character of the string must be alphabetic. The remaining characters must be alphanumeric.

Additional Information The PROCLIB concatenation must contain a member with this name in order to start user server address spaces.

Example USER_SERVER_JOBNAME="IVPUSRT"

USRSRV_TIMEOUT

Name USRSRV_TIMEOUT

Summary Specifies the timeout, in seconds, for a server.

Default 45

Minimum Value 45

Maximum Value 86400

Additional Information If no input message is received by the specified idle time, the server will shutdown automatically.

USS_DUBPROCESS

Name USS_DUBPROCESS

Summary Specifies whether subtasks are dubbed as threads in a caller's process or as new processes.

Default 0

Supported Values

Value	Description
0	The subtask of the caller is dubbed as a thread in the caller's process when the subtask issues its first z/OS UNIX service call.
1	The subtask of the caller is dubbed as a new process when the subtask issues its first z/OS UNIX service call.

Additional Information

If USS commands such as FTP are to be executed, USS_DUBPROCESS must be set to 1, otherwise the following error will occur:

```
CEE5101C During initialization the callable service BPX1MSS
failed.
The system return code was 156, the reason code was 0D070200
```

Note: With USS_DUBPROCESS="1", a user server address space does not support ISPF dialogs.

VIPA

Name	VIPA
Summary	Specifies a symbolic name to use instead of z/Server's IP address when connecting to the client.
Default	No default.
Supported Values	Any valid DNS hostname.
Additional Information	By default, during connection z/Server sends its own IP address back to the client, which is then used by the client for further communication. However, when the client connects through NAT'ed addresses, the IP address that the scheduler address space knows about might be completely different from the IP address that the client must use. In this case, the symbolic name provided by the VIPA parameter (and translated using a DNS server) should be specified to ensure communication. **Note:** The VIPA parameter cannot be used when no fixed unique IP address is assigned to the symbolic name, for example, when VIPA is used for load distribution.
Example	VIPA="custom.com"

VPOOLSIZE

Name	VPOOLSIZE
Summary	Specifies the size of the pool of variables processed with VGET, VPUT and VDEL services. This is similar to the ISPF profile pool.
Default	4K
Minimum Value	4K
Maximum Value	40K
Supported Values	An integer followed by the character "K". There must be no space between the integer and the "K".
Additional Information	The value specified for VPOOLSIZE must be a multiple of 4K.
Example	VPOOLSIZE="16K"

Index