



TIBCO® Managed File Transfer Platform Server for z/OS

Installation and Operation Guide

*Version 8.1.0
August 2021*



Contents

Contents	2
Installing TIBCO® Managed File Transfer Platform Server for z/OS	6
Copying Platform Server REXX Execs to an HFS Directory	9
Postinstallation Tasks	10
APF Authorizing the Load Library	10
Defining Platform Server to the z/OS Security System	10
The RACF Security Interface	12
The Top Secret Security Interface	24
The CA-ACF2 Security Interface	31
Defining the VSAM Files	36
Defining the Audit VSAM Dataset	36
Defining the Work Queue VSAM dataset	37
Defining the User Profile Dataspace	38
Defining the DNI Active Queue Dataspace	39
Defining the MSGTEXT Cluster	40
Defining the Sequential Files	40
Creating a TSO Logon Procedure	41
Adding Platform Server ISPF Interface to TSO Logon Procedure	42
(Optional) Dynamically Adding Platform Server Libraries to ISPF	43
(Optional) Adding Configuration Parameters to OMVS User Profile	44
(Optional) Adding Platform Server to an ISPF Selection Panel	45
Defining VTAM Resources for Systems Using SNA	46
Defining an APPLID for the Platform Server	46
Defining APPLIDs for Platform Server Batch Interface and ISPF Panels	49
The Startup JCL	50
EXEC PARM Fields Used by Platform Server	54
STEPLIB	56

Performing an Upgrade	57
Updating the VSAM Datasets	57
The AUDIT Dataset	57
The WORKQ Dataset	59
The MSGTEXT Cluster	59
Platform Server Configuration	60
Defining Local Resources and Initialization Defaults	60
GLOBAL Startup Parameters	60
Defining and Configuring the FUSCFG File	104
FUSCFG Configuration File	104
Methods of Communicating	105
Parameter Syntax Rules	105
FUSCFG Configuration Parameters	106
Sample FUSCFG Configuration	109
Defining Remote Systems in Configuration Library	111
Configuration Member List - Member Name CONFIG	111
Node Definition Parameters	113
Sample Node Definitions	139
Distribution Lists	143
Class of Service (COS) Definitions	144
Translation Table (XLATE) Definitions	150
TCP Translation Table (TCPXLATE) Definitions	153
SSL Authorization File	155
SSL Authorization Parameters	156
Setting Up SSL Authorization File on z/OS	160
User Profiles	160
Using Initiator User Profile Facility	162
Using Responder User Profile Facility	163
User Profile Facility Security	164
Managing User Profiles	164
SUBJCL (Submit JCL) Interface	173

SUBJCL Security	173
SUBJCL Configuration Parameters	174
SUBJCL Parameter Substitution	178
CFACCESS	179
CFACCESS Parameter Configuration	181
CFACCESS Parameters	181
CFACCESS Example	186
CFALIAS	187
CFALIAS Parameter Configuration	188
CFALIAS Parameters	190
CFALIAS Example	192
TRCLASS (Transfer Class)	193
TRCLASS Parameter Configuration	194
Transfer Class Parameters	195
TRCLASS Example	195
SSL Configuration	196
GLOBAL SSL Parameter Definitions	196
NODE SSL Parameter Definitions	201
User Interface TLS Parameter Definitions (Batch/REXX)	201
Defining the Password Associated with the SSL Key Database File	201
Key Database	202
Creating a Key Ring	210
Creating SSL Certificates Using a Certificate Authority	211
Creating SSL Certificates Using RACF as a Certificate Authority	215
Other Useful RACDCERT Commands	220
Authorizing User to Key Ring	221
Operator Commands	223
Command Basics	223
System Operations	223
Node Operations	225
Activities	227

z/OS Console Display Commands	227
File Transfer Activities	229
Platform Server Console Operator Commands	231
Troubleshooting	243
Appendix A. Sample SNA Mode Definition	250
Appendix B. Running Traces	253
Appendix C. Automated Operations	254
Appendix D. File Name Tokens	257
File Name Tokens List	257
Examples of Using the File Name Tokens	260
Rules for Using the File Name Tokens	261
Appendix E. CA-7 Job Scheduler Interface	263
Appendix F. User Exits	264
Invoking Platform Server User Exits	268
Appendix G. WAIT Parameter for IF/THEN/ELSE/ENDIF	271
Appendix H. Overriding JCL SYSIN Parameters	274
Appendix I. XCOM Interface	277
Installing and Configuring Platform Server XCOM Interface	278
Appendix J. PDS and HFS File	282
PDS Support	282
HFS File Support	285
TIBCO Documentation and Support Services	286
Legal and Third-Party Notices	288

Installing TIBCO® Managed File Transfer Platform Server for z/OS

You can install TIBCO Managed File Transfer (MFT) Platform Server for z/OS directly from a workstation by using the TSO/E Transmit and Receive commands.

Procedure

1. Download the package from the TIBCO download site <https://edelivery.tibco.com>.

You are prompted for a user name and password for downloading. If you do not receive a user name and password, contact TIBCO Technical Support.

2. Unzip the package to a local directory.

The unzipped package contains the following two files:

- `MFTvrm.XMT`: This file contains the necessary information to create all installation datasets.
- `readme.TXT`: This file contains information regarding the installation process. Review this file before you begin the installation process.

3. Upload the `MFTvrm.XMT` file.

You can upload the XMT file by issuing the `UPLOAD` or `SEND` command that your emulator provides.

Note: Do not specify any options on the upload, for example, `CRLF` or `ASCII`. The destination dataset must be named in `High_Level_Qualifier.PDS.XMT` format.

When uploading the XMT file to z/OS, you do not have to pre-allocate the z/OS dataset. If you must pre-allocate the dataset, use the following attributes:

Attribute	Value
Organization	PS

Attribute	Value
Record format	FB
Record length	80
Block size	6160
Space unit	TRKS
Directory blocks	0
Primary quantity	150
Secondary quantity	10

4. Receive the XMT file.

- a. Enter the following command at TSO option 6 or at the ready prompt:

```
RECEIVE INDS (Name_of_uploaded_file)
```

For example: `RECEIVE INDS (FUSION.MFT720.XMT)`

- b. Enter the following parameters when you are prompted:

```
DSN (output_dataset_name) VOLUME (destination_volume_name)
```

For example: `DSN (MFT720.INSTALL.PDS) VOLUME (ABC123)`

i Note: The `VOLUME` parameter is optional. You can use it if you want to control the dataset allocation.

5. Read and accept the End User License Agreement (EULA).

You must run the following REXX exec to read and accept the EULA:

```
EXEC 'MFTvrm.INSTALL.PDS(ACCEULA)'
```

6. Execute the @RECEIVE member in the PDS dataset.

i Note: If you are performing an upgrade,

- The LOADLIB library contains sample user exit load modules. If you want to replace the existing datasets, you have to back up any existing customized user exit load modules before running this job, and copy back the customized user exit load modules to the LOADLIB after running this job.
- The SAMPLIB contains the GLOBAL member. Ensure not to overwrite your existing GLOBAL member.

- Edit the @RECEIVE member to meet your installation requirements.
- Submit the @RECEIVE job to receive seven XMT files and create PDS files.

For example: `MFTvrm.INSTALL.PDS(@RECEIVE)`

The following table lists the created PDS files:

PDS File	Description
EXECS	ISPF REXX execs
JCL	Sample JCL
LOADLIB	The load library
MSGs	ISPF messages
MTEXT	Message inquiries
PANELS	ISPF panels
SAMPLIB	The sample library

Copying Platform Server REXX Execs to an HFS Directory

This procedure is optional, and it is only required if you want to execute the Platform Server REXX execs within an OMVS environment.

Procedure

1. Create an HFS directory that is accessible by all users by using the following OMVS command.

```
mkdir /fusexecs
```

2. Copy the execs to the HFS directory by using the following TSO commands.

```
OPUTX 'FUSION.EXECS' '/fusexecs/' LC TEXT MODE(711)  
OPUTX 'FUSION.EXECS' '/fusexecs/' TEXT MODE(711)
```

MODE(711) gives the user all rights to the files and gives the group and all other users execution rights to the files. You can change this parameter as is required by your installation.

LC sets the OMVS file names in lowercase. UC sets the OMVS file names in uppercase.

Postinstallation Tasks

After a new installation of TIBCO MFT Platform Server for z/OS, you must perform some postinstallation tasks before you can start the Platform Server successfully.

APF Authorizing the Load Library

Before starting TIBCO MFT Platform Server for z/OS, the load library must be APF authorized.

TIBCO MFT Platform Server for z/OS requires APF authorization to perform security functions on behalf of multiple users in its single address space.

Procedure

1. Add the following line to your PROGxx member in SYS1.PARMLIB to authorize the library.

```
APF ADD DSNAME(Platform_Server_Load_Library) VOLUME(volser)
```

2. Issue the following z/OS operator command to activate this configuration:

```
SET PROG=xx
```

Where xx is the PROGxx member suffix.

Defining Platform Server to the z/OS Security System

TIBCO MFT Platform Server for z/OS supports different security approaches to adapt to different security architecture you might implement in your environment.

The following security approaches are supported:

- [The RACF Security Interface](#)
- [The Top Secret Security Interface](#)
- [The CA-ACF2 Security Interface](#)

TIBCO MFT Platform Server for z/OS is a multi-user application. From a security perspective, TIBCO MFT Platform Server for z/OS must be defined to your security system as a multi-user address space, and all dataset access must be approved based on the individual user requesting the function. TIBCO MFT Platform Server for z/OS uses the z/OS Security Authorization Facility (SAF) to initialize and check security for each user at the time they initiate.

Because the Platform Server performs functions on behalf of multiple users, it must call the installed z/OS security system to create individual security environments (ACEEs) for each user and to verify that the correct password is entered for that user. All requests for resources, such as datasets, by that user are verified using that user's environment.

The type of security processing that the Platform Server performs depends on whether the local z/OS system initiates the request, in which case the Platform Server for z/OS acts as the initiator, or the Platform Server is processing a request that is started by a remote computer, in which case the Platform Server for z/OS acts as the responder.

- **Initiator security processing:** When a user invokes the client subtask to request that a file be transferred to or from a remote computer, the client software verifies if the user has access to the local dataset. When transmitting to the remote system, security information is transmitted to the remote node, including the remote user ID, the type of access, and an encrypted password.
- **Responder security processing:** When acting as a responder, the Platform Server extracts the necessary security information from the protocol data received from the remote system, including the name of the dataset, the user ID, the password, and the type of access. The Platform Server then uses the z/OS security manager to determine whether there is sufficient authority to perform the requested activity. TIBCO MFT Platform Server for z/OS and the z/OS server are subject to the same security checks as any other z/OS application. Therefore, TIBCO MFT Platform Server for z/OS must have a security profile, to have access privileges to any file that might be transferred or managed. Failing to define proper access can result in a 913 abend error.

The Platform Server identifies which user is trying to access the system depending on whether the file transfer is started by the local z/OS server or by a remote computer such as a Windows server.

- When the Platform Server started task is acting as an initiator, the Platform Server retrieves the initiator's user ID from z/OS control blocks, and places that user ID into the work queue dataspace of the Platform Server. The Platform Server does not support the user ID to run the request under the authority of anyone else.

- When the Platform Server started task is acting as a responder, the Platform Server receives the user ID and encrypted password over the data communications links, and then issues a `RACROUTE TYPE=VERIFY` command to ensure that the user ID is valid, and the user ID and password combination are a match.

The RACF Security Interface

TIBCO MFT Platform Server for z/OS honors existing RACF dataset security on user basis. As defined by user profiles, any user with existing RACF security can access the Platform Server.

If additional security is necessary, or if you want to restrict some users' access to the Platform Server, make the appropriate RACF modifications for those users.

You must choose the RACF profile name (namely the user ID) under whose privileges the Platform Server is run. This can be an existing user ID or a newly created one. Ensure this profile is authorized to every resource and every dataset that the Platform Server might access.

i Note: You must consider if you have to define a new logon procedure for the Platform Server ISPF interface.

Defining TIBCO MFT Platform Server for z/OS as a Multiple-User Address Space

To define TIBCO MFT Platform Server for z/OS to RACF as a multiple-user address space, you must define the RACF user ID under which the Platform Server started task runs and update the RACF started procedures table.

Procedure

1. Define the RACF user ID under which the Platform Server started task runs.

You must choose the RACF profile name (namely the user ID) under whose privileges the Platform Server will run. This must be a new profile name created specifically for TIBCO MFT Platform Server for z/OS.

You can create a new user ID by using the `ADDUSER` command, and then assign the user to a RACF group.

The user defined must have rights to any dataset that the Platform Server might access. The format of the ADDUSER command is as follows:

```
ADDUSER (fususer) DFLTGRP(fusgroup) OMVS(UID(nnnnn))
```

Where:

- *fususer* is the name of the Platform Server user.
- *fusgroup* is the name of the group used as a default name for the Platform Server user. To use TCP/IP connectivity, this group requires a GID of 0.
- *nnnnn* is the OpenMVS UID assigned to the user for TCP/IP connectivity.

i Note:

The user name defined must match the user name added to the ICHRIN03 module. For RACFV2, the user name defined must match the name of the user added to the STARTED class for the Platform Server.

For example, the following command adds a user called FUSUSER and specifies the default group as STASKS. The group STASKS must be previously defined to RACF. An OpenMVS segment is also created for the Platform Server ID with the UID of 100100.

```
ADDUSER (FUSUSER) DFLTGRP(STASKS) OMVS(UID(100100))
```

2. Update the RACF started procedures table to include a new entry for the name of the Platform Server started task.

The RACF started procedure table can be updated by adding the Platform Server to the RACF started resource class. The format of the command is as follows:

```
RDEFINE STARTED (member.jobname)
  STDATA(USER(userid), GROUP(groupid),
  TRUSTED(YES))
```

Where:

- STARTED specifies the RACF resource class.
- *member.jobname* specifies the name of the Platform Server PROC. If a job card is present, jobname specifies the name of the job. Otherwise, jobname must

match the member name.

- STDATA defines the STARTED class data.
- *userid* defines the user ID defined for the Platform Server.
- *groupid* defines the group defined for the Platform Server. If this parameter is not defined, the default group for the Platform Server user is used.
- TRUSTED(YES) defines the user as a trusted user.

i Note: This parameter is optional. If you do not want to provide the trusted user attribute, you must grant the Platform Server user the rights to access any necessary datasets and update any necessary password.

For more information, see these IBM manuals: *RACF Security Administrators Guide* and *RACF Command Language Reference*.

3. Authorize the Platform Server started task to issue operator commands by using the following RACF command:

```
PERMIT MVS.MODIFY.STC.fusionstc.* CLASS(OPERCMDS)
ID(fususer) ACCESS(UPDATE)
```

Normally, the Platform Server can run without this authorization. The Platform Server will detect when TCP/IP is stopped. With this authorization, when TCP is brought up again, the Platform Server can restart the TCP services by internally issuing two Platform Server operator commands to stop and start the TCP Interface.

Creating RACF Facility Classes

The Platform Server uses RACF facility classes to determine whether a user is authorized to perform certain Platform Server maintenance functions.

The Platform Server uses the following two types of facility classes to validate whether a user is authorized for a function:

- REXX/ISPF inquiry and profile authorization
- Command Center functions

Procedure

1. Create the REXX/ISPF inquiry and profile facility under RACF.

The GLOBAL BOSSID parameter, which is defined in [GLOBAL Startup Parameters](#), defines a facility class that is used for authorization checking for the REXX/ISPF interface and the user profile.

For example, BOSSID=\$FUSION:

- To define this facility under RACF, you can use the following RACF command:

```
RDEFINE FACILITY $FUSION UACC(NONE)
```

- To make a user a Platform Server ISPF/REXX administrator, you must give them READ authorization to the profile using the following command:

```
PERMIT $FUSION CLASS(FACILITY) ID(userid) ACCESS(READ)
```

- To make a user a Platform Server profile administrator, you must give them CONTROL authorization to the profile using the following command:

```
PERMIT $FUSION CLASS(FACILITY) ID(userid) ACCESS(CONTROL)
```

For more information on Platform Server user profiles, REXX interface, and ISPF interface, see *TIBCO® Managed File Transfer Platform Server for z/OS User's Guide*.

2. Create a read-only REXX/ISPF inquiry and profile facility.

In addition to the processing defined in [step 1](#), you can define an additional facility class to give ISPF users the authority to view but not update transfer requests. This facility adds the suffix .READ to the GLOBAL BOSSID parameter. When a user has READ access to this facility class, the TSO user can perform the following functions:

- View all transfers.
- Update only transfers where the transfer local user ID matches their TSO user ID.

For example, BOSSID=\$FUSION:

- To define this facility under RACF, you can use the following RACF command:

```
RDEFINE FACILITY $FUSION.READ UACC(NONE)
```

- To make a user a Platform Server ISPF/REXX administrator, you must give them READ authorization to the \$FUSION.READ facility use the following RACF command:

```
PERMIT $FUSION.READ CLASS(FACILITY) ID(userid) ACCESS(READ)
```

At this point, the Platform Server will also check the facility class \$FUSION.READ. If a user has access to this facility class, the user can read but not update transfer requests.

i Note: The .READ facility class is only checked if the user does not have access to the facility class defined by the BOSSID parameter.

3. Create the Command Center facility classes.

Four GLOBAL parameters define the names of the facility classes that are used to determine whether a user is authorized for Command Center functions.

For example, the four parameters are defined as follows in the GLOBAL member of the Platform Server SAMPLIB. For more information, see [GLOBAL Startup Parameters](#).

```
CCC_BROWSE_FACILITY=$CCC.BROWSE
CCC_ALTER_FACILITY=$CCC.ALTER
CCC_ADMIN_FACILITY=$CCC.ADMIN
CCC_TRANSFER_FACILITY=$CCC.TRANSFER
```

- To define these facilities under RACF, you can use the following RACF commands:

```
RDEFINE FACILITY $CCC.BROWSE UACC(NONE)
RDEFINE FACILITY $CCC.ALTER UACC(NONE)
RDEFINE FACILITY $CCC.ADMIN UACC(NONE)
RDEFINE FACILITY $CCC.TRANSFER UACC(NONE)
```

- To make a Command Center user authorized for all Command Center functions, you must give them READ authorization to the facility defined by the CCC_ADMIN_FACILITY parameter by using the following command:

```
PERMIT $CCC.ADMIN CLASS(FACILITY) ID(userid) ACCESS(READ)
```


- To make a Command Center user authorized for inquiry on completed transfers, you must give them READ authorization to the facility defined by the CCC_BROWSE_FACILITY parameter by using the following command:

```
PERMIT $CCC.BROWSE CLASS(FACILITY) ID(userid) ACCESS(READ)
```

- To make a Command Center user authorized for altering transfers on the Platform Server queue, you must give them READ authorization to the facility defined by the CCC_ALTER_FACILITY parameter by using the following command:

```
PERMIT $CCC.ALTER CLASS(FACILITY) ID(userid) ACCESS(READ)
```

- To make a Command Center user authorized for initiating transfers, you must give them READ authorization to the facility defined by the CCC_TRANSFER_FACILITY parameter by using the following command:

```
PERMIT $CCC.TRANSFER CLASS(FACILITY) ID(userid) ACCESS(READ)
```

OMVS Definitions for Access to UNIX System Services (USS) Files under RACF

If you want the Platform Server to access OpenEdition USS files, you must configure the user associated with the Platform Server started task as a superuser.

The Platform Server reads the USS directory structure, and performs authorization checking to ensure that users are authorized to access the USS files. All Platform Server files are opened and accessed under the security environment of the user requesting the transfer.

To configure the Platform Server user ID a superuser, add the following definition for the Platform Server user ID:

```
ALU (FUSUSER) OMVS(UID(0))
```

With this definition, the Platform Server can perform authorization checking on behalf of another user.



Note: The OMVS segment is required only if USS file access is required.

You must define the following facility resources on your system:

- BPX.SUPERUSER
- BPX.DAEMON

If these resources are not defined on your system, you can define them using the following commands:

```
RDEFINE FACILITY BPX.SUPERUSER UACC(NONE)
RDEFINE FACILITY BPX.DAEMON UACC(NONE)
```

The user associated with the Platform Server started task must be given authorization to access these resources. The following command gives authorization to the Platform Server:

```
PERMIT BPX.SUPERUSER CLASS(FACILITY) ID(fususer) ACCESS(READ)
PERMIT BPX.DAEMON CLASS(FACILITY) ID(fususer) ACCESS(READ)
```

Where *fususer* stands for the user associated with the Platform Server started task.

i Note: Based on your RACF definition, you might have to issue the SETROPTS REFRESH command to refresh the RACF storage tables.

Any user that requires OMVS facilities, including the Platform Server started task user, must be defined with the OMVS segment of the RACF profile to indicate that the user is authorized for OMVS.

RACF Surrogate Checking

The Platform Server uses the RACF SURROGAT class to see whether a user is authorized to run a transfer under another user ID without specifying a password.

You can define the local user ID (LUSER) and password (LPASS) that are used to perform a file transfer. When both user ID and password are defined, the Platform Server validates the user ID/password combination, and performs the transfer under that user's credentials.

You can also specify a local user ID without specifying a local password. To avoid security violation, however, the Platform Server checks whether the initiating user ID is authorized to submit jobs under the authorization of the local user ID (LUSER). In RACF terms, this is called surrogate checking.

i Note: Surrogate checking only applies to initiator tasks.

To define the SURROGAT facility under RACF, you can use the following RACF commands.

```
RDEFINE SURROGAT userid1.SUBMIT UACC(NONE)
PERMIT userid1.SUBMIT CLASS(SURROGAT) ID(userid2) ACCESS(READ)
```

Where, *userid2* is the user that submits the file transfer request, and *userid2* wants to run the transfer as *userid1*.

i Note: Depending on how the RACF system is defined, you might have to refresh the SURROGAT class after running the commands.

For example, the following RACF commands define TSO user OPER to run a transfer under the authorization of user ID CA70NL.

```
RDEFINE SURROGAT CA70NL.SUBMIT UACC(NONE)
PERMIT CA70NL.SUBMIT CLASS(SURROGAT) ID(OPER) ACCESS(READ)
```

Extended Security Checking

Based on the GLOBAL EXTENDED_SECURITY_CHECK configuration option, the Platform Server can check whether you are authorized to use the Platform Server or initiate a transfer with a particular Platform Server node.

i Note: Extended security is not available when the Platform Server for z/OS is acting as a responder.

Before performing extended security checking, you must define the RACF resources that the Platform Server checks. The resource that is checked has a variable component that is based on the GLOBAL EXTENDED_SECURITY_RESOURCE parameter and a fixed component that does not change. Therefore, you can specify different resources for different Platform Server started tasks.

The Platform Server does not perform any resource checking if the GLOBAL EXTENDED_SECURITY_CHECK parameter is set to NO. If the parameter is set to WARN, the checking is performed, and the request continues even if the resource check fails. If the parameter is

set to ENFORCE, the checking is performed, and the request is terminated with errors if the resource check fails.

Where Is the RACROUTE Checking Performed

All RACROUTE AUTH requests are run on the same system as the Platform Server started task. If the batch or TSO request is through cross memory services (XMS), the request will be run within the ISPF or batch job. If the batch or TSO request is through SNA or TCP, the request will be run in the Platform Server started task.

In the following examples, the GLOBAL EXTENDED_SECURITY_RESOURCE parameter is set to the default value of \$CFUSION, while the GLOBAL EXTENDED_SECURITY_CHECKING parameter is set to ENFORCE, ENFORCE to check authorization for Platform Server use and node security:

```
EXTENDED_SECURITY_RESOURCE=$CFUSION
EXTENDED_SECURITY_CHECKING=(ENFORCE, ENFORCE)
```

For more information, see [GLOBAL Startup Parameters](#).

Checking Whether the User Is Authorized to Use the Platform Server

The Platform Server performs different checks when running under TSO and when running as a batch job.

- When running under TSO, the Platform Server checks the following RACF resources:
\$CFUSION.TRANSFER.AUTH.TSO
- When running under batch, the Platform Server checks the following RACF resource:
\$CFUSION.TRANSFER.AUTH.BATCH

You can define these resources individually using the following commands:

```
RDEFINE FACILITY $CFUSION.TRANSFER.AUTH.TSO UACC(NONE)
RDEFINE FACILITY $CFUSION.TRANSFER.AUTH.BATCH UACC(NONE)
```

You can give a user authorization to these facility classes using the following commands:

```
PERMIT $CFUSION.TRANSFER.AUTH.TSO CLASS(FACILITY) ID(userid) ACCESS(READ)
PERMIT $CFUSION.TRANSFER.AUTH.BATCH CLASS(FACILITY) ID(userid) ACCESS
(READ)
```

If you want to assign both TSO and batch transfer rights to a user, then you can use the generic facility feature of RACF by specifying the following commands.

- i Note:** RACF uses the most specific profile that matches the resource name. Therefore, if a batch or TSO resource is created, the generic resource is not checked when performing authorization checks for that resource.

```
RDEFINE FACILITY $CFUSION.TRANSFER.AUTH.* UACC(NONE)
PERMIT $CFUSION.TRANSFER.AUTH.* CLASS(FACILITY) ID(userid) ACCESS(READ)
```

Example 1:

In your organization, user ID CFMASTR is responsible for all file transfer activities, user ID CFTSO01 only needs the right to submit transfers from TSO, and batch user ID CFBAT01 is only used for batch transfers.

1. Define the facilities to RACF by using the following commands.

- i Note:** The RDEFINE statements only needs to be performed once.

```
RDEFINE FACILITY $CFUSION.TRANSFER.AUTH.TSO UACC(NONE)
RDEFINE FACILITY $CFUSION.TRANSFER.AUTH.BATCH UACC(NONE)
```

2. Make these facilities be available to the relevant users.

- For CFMASTR, issue the following commands:

```
PERMIT $CFUSION.TRANSFER.AUTH.TSO CLASS(FACILITY) ID(CFMASTR)
ACCESS(READ)
PERMIT $CFUSION.TRANSFER.AUTH.BATCH CLASS(FACILITY) ID
(CFMASTR) ACCESS(READ)
```

Now, CFMASTR can use either batch or TSO to initiate a transfer.

- For CFTSO01 and CFBAT01, issue the following commands:

```
PERMIT $CFUSION.TRANSFER.AUTH.TSO CLASS(FACILITY) ID(CFTSO01)
ACCESS(READ)
PERMIT $CFUSION.TRANSFER.AUTH.BATCH CLASS(FACILITY) ID
(CFBAT01) ACCESS(READ)
```

Example 2:

Use a generic resource to grant access to users CFMASTR and CFUSER to initiate transfers through both batch and TSO.

```
RDEFINE FACILITY $CFUSION.TRANSFER.AUTH.* UACC(NONE)
PERMIT $CFUSION.TRANSFER.AUTH.* CLASS(FACILITY) ID(CFMASTR) ACCESS(READ)
PERMIT $CFUSION.TRANSFER.AUTH.* CLASS(FACILITY) ID(CFUSER) ACCESS(READ)
```

i Note: The RDEFINE statement has only to be performed once.

It is a good practice that before running Platform Server extended security in ENFORCE mode, it is run in WARN mode, in this way any security issues can be corrected before production transfers are cancelled.

NODE Level Security Checking

The Platform Server incorporates the node name into a resource to check if a user is authorized to initiate transfers to a particular node. When validating access to a particular node, you can also differentiate between a send transfer and a receive transfer to the specific node by using the following commands:

```
$CFUSION.TRANSFER.nodename.INIT.SEND
$CFUSION.TRANSFER.nodename.INIT.RECEIVE
```

These resources can be defined to RACF by using the following commands:

```
RDEFINE FACILITY $CFUSION.TRANSFER.nodename.INIT.SEND UACC(NONE)
RDEFINE FACILITY $CFUSION.TRANSFER.nodename.INIT.RECEIVE UACC(NONE)
```

In place of the *nodename*, a reserved IPADDR string can be used to restrict a user from specifying the IP address or IP name on the transfer. This IPADDR string can be used to force users to only use the Platform Server defined nodes when initiating a transfer. The RACF resources used to protect against usage of an IP address or IP name are following:

```
$CFUSION.TRANSFER.IPADDR.INIT.SEND
$CFUSION.TRANSFER.IPADDR.INIT.RECEIVE
```

These resources are defined to RACF by using the following commands:

```
RDEFINE FACILITY $CFUSION.TRANSFER.IPADDR.INIT.SEND UACC(NONE)
RDEFINE FACILITY $CFUSION.TRANSFER.IPADDR.INIT.RECEIVE UACC(NONE)
```

You can use the previous definitions to protect who can perform a send or receive transfer using an IP address or IP name. To protect against all usage of an IP address or IP name, use the following RACF definition:

```
RDEFINE FACILITY $CFUSION.TRANSFER.IPADDR.** UACC(NONE)
```

i Note: RACF uses the most specific profile that matches the resource name. Therefore, if a specific resource is created, the generic resource is not checked when performing authorization checks for that resource.

Example 1:

The following RDEFINE and PERMIT commands protect the use of an IP address or IP name for all users except the CFADMIN ID:

```
RDEFINE FACILITY $CFUSION.TRANSFER.IPADDR.** UACC(NONE)
PERMIT $CFUSION.TRANSFER.IPADDR.** CLASS(FACILITY) ID(CFADMIN) ACCESS
(READ)
```

Example 2:

The following RDEFINE and PERMIT commands protect the use of all nodes except the CFADMIN ID:

```
RDEFINE FACILITY $CFUSION.TRANSFER.** UACC(NONE)
PERMIT $CFUSION.TRANSFER.** CLASS(FACILITY) ID(CFADMIN) ACCESS(READ)
```

Example 3:

The following RDEFINE and PERMIT commands gives the CFOPER ID the ability to initiate a Platform Server send transfer to all nodes:

```
REDEFINE FACILITY $CFUSION.TRANSFER.*.INIT.SEND UACC(NONE)
PERMIT $CFUSION.TRANSFER.*.INIT.SEND CLASS(FACILITY) ID(CFOPER) ACCESS
(READ)
```

Example 4:

The following RDEFINE and PERMIT commands gives the CFNY001 ID the ability to send to and receive from the node CFNY:

```
RDEFINE FACILITY $CFUSION.TRANSFER.CFNY.INIT.* UACC(NONE)
PERMIT $CFUSION.TRANSFER.CFNY.INIT.* CLASS(FACILITY) ID(CFNY001) ACCESS
(READ)
```

Example 5:

The following RDEFINE and PERMIT commands gives the CFLA001 ID the ability to send to and receive from the node CFLA:

```
RDEFINE FACILITY $CFUSION.TRANSFER.CFLA.INIT.* UACC(NONE)
PERMIT $CFUSION.TRANSFER.CFLA.INIT.* CLASS(FACILITY) ID(CFLA001) ACCESS
(READ)
```



Note: It is a good practice that before running Platform Server extended security in ENFORCE mode, you run it in WARN mode. In this way, any security issues can be corrected before production transfers are cancelled.

The Top Secret Security Interface

TIBCO MFT Platform Server for z/OS works with CA-Top Secret to ensure that end users can only access files for which they are authorized. The Platform Server started task must be defined to CA-Top Secret with the authorization to access all files that might be transferred.

The Platform Server makes standard calls to the System Authorization Facility (SAF). It is not possible to suppress the SAF calls made by the Platform Server. Your security administrator can choose whether to grant individual users or groups access to the Platform Server started task. The Platform Server started task must be defined to CA-Top Secret with the authorization to access all files that might be transferred.

Like other applications that can be signed onto and issue the RACROUTE TYPE=VERIFY command, TIBCO MFT Platform Server for z/OS must be defined to CA-Top Secret as a facility.

Before defining the Platform Server facility to CA-Top Secret, you have to decide the attributes that you want the facility to have. These attributes can be modeled after TSO.

Ensure the RES, data checking, and multi-user options are set. The initial program must be specified as FUS (INITPGM(FUS)). You might want to shut off the last message option and the status message option to avoid console clutter.

Defining TIBCO MFT Platform Server for z/OS as a Multi-User Facility

The Platform Server can handle multiple transfers for multiple users simultaneously; therefore, the Platform Server must be defined to CA-Top Secret as a multi-user facility. You can add the Platform Server as a CA-Top Secret facility by modifying the CA-Top Secret facilities matrix table. This table is contained in the CA-Top Secret PARMFILE.

Procedure

1. Add the following entries to your PARMFILE to define the Platform Server as a CA-Top Secret facility.

```
FAC(USER1=NAME=FUSION) * Use the USER1 slot
FAC(FUSION=MULTIUSER) * Facility can have multiple users
FAC(FUSION=AUTHINIT) * Facility can issue RACINIT
FAC(FUSION=PGM=FUS) * RACINIT program name starts with FUS
FAC(FUSION=SIGN(M)) * Allow multiple signons for the same ACID
FAC(FUSION=NOLUMSG) * Do not issue the Last Used message
```

2. Restart CA-Top Secret to activate previous entries.

The CA-Top Secret is restarted typically after an IPL. To make these changes dynamically without an IPL, you can enter the following TSS MODIFY commands:

```
TSS MODIFY(FAC(USER1=NAME=FUSION))
TSS MODIFY(FAC(FUSION=MULTIUSER))
TSS MODIFY(FAC(FUSION=AUTHINIT))
TSS MODIFY(FAC(FUSION=PGM=FUS))
TSS MODIFY(FAC(FUSION=SIGN(M))
TSS MODIFY(FAC(FUSION=NOLUMSG))
```

In the previous example, USER1 slot is used in the facilities matrix table and renamed to FUSION. You can issue TSS MODIFY(FAC(FUSION)) to view the facilities matrix table entry to ensure that the entries are installed correctly:

```
TSS9550I FACILITY DISPLAY FOR FUSION

TSS9551I INITPGM=FUS      ID=2  TYPE=099
```

```

TSS9552I ATTRIBUTES=IN-
USE,ACTIVE,SHRPRF,ASUBM,NOABEND,MULTIUSER,NOXDEF
TSS9552I ATTRIBUTES=NOLUMSG,STMSG,SIGN(M),INSTDATA,RNDPW,AUTHINIT

TSS9552I ATTRIBUTES=NOPROMPT,NOAUDIT,RES,WARNPW,NOTSOC,LCFTRANS

TSS9552I
ATTRIBUTES=MSGLC,NOTRACE,NOEODINIT,IJU,NODORMPW,NONPWR,NOIMSXTND
TSS9553I MODE=WARN DOWN=GLOBAL LOGGING=MSG,SEC9

TSS9554I UIDACID=8 LOCKTIME=000 DEFACID=*NONE* KEY=8

TSS9566I MAXUSER=03000 PRFT=003

TSS0300I MODIFY FUNCTION SUCCESSFUL

```

Defining ACID for TIBCO MFT Platform Server for z/OS Started Task

For the Platform Server to function properly, you must define the Accessor ID (ACID) to CA-Top Secret.

Procedure

1. You can use the following command to define an ACID such as CFUSION that the Platform Server started task runs as:

```

TSS CREATE(CFUSION) NAME('CYBERFUSION FILE TRANSFER') PASSWORD
(NOPW)
DEPARTMENT(SYSTEMS) FACILITY(STC) MASTFAC(FUSION)

```

Where:

- The PASSWORD parameter is set to NOPW because the Platform Server runs as a started task.
- The DEPARTMENT parameter specifies an existing department within your CA-Top Secret database.
- The FACILITY parameter, STC, defines the Platform Server to run as a started task.
- The MASTFAC parameter is the name you assigned for the Platform Server in the

Facilities matrix table in [Defining TIBCO MFT Platform Server for z/OS as a Multi-User Facility](#).

Defining TIBCO MFT Platform Server for z/OS to STC Table

During the installation, the FUSION member of the SAMPLIB is copied to a system procedure library so that the Platform Server can be run as a started task.

Procedure

1. Associate the Platform Server started task with the Platform Server ACID by using the following command:

```
TSS ADD(STC) PROCNAME(FUSION) ACID(CFUSION)
```

Giving Users Access to TIBCO MFT Platform Server for z/OS

After TIBCO MFT Platform Server for z/OS is defined as a facility, you must give users access to that facility to initiate file transfers.

Procedure

1. Use the appropriate command as you want:
 - To give all users access to the Platform Server facility, use the following command:

```
TSS ADD(ALL) FAC(FUSION)
```

This adds the FUSION facility to the CA-Top Secret ALL record, which applies to all users on the system.

- To control access to Platform Server file transfers by adding the Platform Server facility to individual users or groups of users, use the following command:

```
TSS ADD(userid) FAC(FUSION)
```

Creating Top Secret Facility Classes

The Platform Server uses Top Secret facility classes to determine whether a user is authorized to perform certain Platform Server maintenance functions.

The Platform Server uses the following two types of facility classes to validate if a user is authorized for a function:

- REXX/ISPF inquiry and profile authorization
- Command Center functions

Procedure

1. Create the REXX/ISPF inquiry and profile facility under CA-Top Secret.

The GLOBAL BOSSID parameter, which is defined in [GLOBAL Startup Parameters](#), defines a facility class that is used for authorization checking for:

- The REXX/ISPF interface
- The user profile

If BOSSID=\$FUSION:

- To define this facility under Top Secret, you can use the following command:

```
TSS ADD(TSSUSER) IBMFAC($FUSION)
```

- To make a user a Platform Server ISPF/REXX administrator, you must give them READ authorization to the profile by using the following command:

```
TSS PER(userid) IBMFAC($FUSION) ACCESS(READ)
```

- To make a user a Platform Server profile administrator, you must give them CONTROL authorization to the profile by using the following command:

```
TSS PER(userid) IBMFAC($FUSION) ACCESS(CONTROL)
```

For more information, see *TIBCO® Managed File Transfer Platform Server for z/OS User's Guide*.

2. (Optional) Create a Read Only REXX/ISPF inquiry and profile facility.

In addition to the processing defined in [step 1](#), you can define an additional facility class to permit ISPF users to view but not update transfer requests. This facility adds the suffix .READ to the GLOBAL BOSSID parameter. When a TSO user has READ access to this facility class, the user can perform the following functions:

- View all transfers.

- Update only transfers where the transfer local user ID matches their TSO user ID.

If BOSSID=\$FUSION:

- To define this facility under Top Secret, you can use the following command:

```
TSS ADD(TSSUSER) IBMFAC($FUSION.READ)
```

- To make a user a Platform Server ISPF/REXX administrator, you must give them READ authorization to the previous facility by using the following command:

```
TSS PER(userid) IBMFAC($FUSION.READ) ACCESS(READ)
```

At this point, the Platform Server also checks facility class \$FUSION.READ. If a user has access to this facility class, then the user can read but not update transfer requests.

i Note: The .READ facility class is only checked if the user does not have access to the facility class defined by the BOSSID parameter.

3. Create the Command Center facility classes.

Four GLOBAL parameters define the names of the facility classes that are used to determine if a user is authorized for Command Center functions. See the following values defined in the GLOBAL member of the Platform Server SAMPLIB. For more information, see [GLOBAL Startup Parameters](#).

```
CCC_BROWSE_FACILITY=$CCC.BROWSE
CCC_ALTER_FACILITY=$CCC.ALTER
CCC_ADMIN_FACILITY=$CCC.ADMIN
CCC_TRANSFER_FACILITY=$CCC.TRANSFER
```

To define these facilities under Top Secret, you can use the following Top Secret commands:

```
TSS ADD(TSSUSER) IBMFAC($CCC.BROWSE)
TSS ADD(TSSUSER) IBMFAC($CCC.ALTER)
TSS ADD(TSSUSER) IBMFAC($CCC.ADMIN)
TSS ADD(TSSUSER) IBMFAC($CCC.TRANSFER)
```

- To make a Command Center user authorized for all Command Center functions,

you must give them READ authorization to the facility defined by the CCC_ADMIN_FACILITY parameter as follows:

```
TSS PER(userid) IBMFAC($CCC.ADMIN) ACCESS(READ)
```

- To make a Command Center user authorized for inquiry on completed transfers, you must give them READ authorization to the facility defined by the CCC_BROWSE_FACILITY parameter as follows:

```
TSS PER(userid) IBMFAC($CCC.BROWSE) ACCESS(READ)
```

- To make a Command Center user authorized for altering transfers on the Platform Server queue, you must give them READ authorization to the facility defined by the CCC_ALTER_FACILITY parameter as follows:

```
TSS PER(userid) IBMFAC($CCC.ALTER) ACCESS(READ)
```

- To make a Command Center user authorized for initiating transfers, you must give them READ authorization to the facility defined by the CCC_TRANSFER_FACILITY parameter as follows:

```
TSS PER(userid) IBMFAC($CCC.TRANSFER) ACCESS(READ)
```

OMVS Definitions for Access to HFS Files under Top Secret

If you want the Platform Server to access OpenEdition HFS files, you must configure the user associated with the Platform Server started task as a superuser.

This is required because the Platform Server reads the HFS directory structure, and performs authorization checking to ensure that users are authorized to access the HFS files. All Platform Server files are opened and accessed under the security environment of the user requesting the transfer.

You must define the following facility resources on your system:

- BPX.SUPERUSER
- BPS.DAEMON

If these resources are not defined on your system, you can define them by using the following commands:

```
TSS ADD(TSSUSER) IBMFAC(BPX)
```

The user associated with the Platform Server started task must be given authorization to access these resources. You can use the following commands:

```
TSS PERMIT(fususer) IBMFAC(BPX.SUPERUSER) ACCESS(READ)
TSS PERMIT(fususer) IBMFAC(BPX.DAEMON) ACCESS(READ)
```

Where *fususer* stands for the user associated with the Platform Server started task.

Any user that requires OMVS facilities, including the Platform Server started task user, must be defined with the OMVS segment of the RACF profile to indicate that the user is authorized for OMVS.

The CA-ACF2 Security Interface

For the application to perform security access checking for users initiating the interface, TIBCO MFT Platform Server for z/OS must be defined to CA-ACF2 as a multi-user address space.

Creating CA-ACF2 Facility Classes

The Platform Server uses CA-ACF2 facility classes to determine whether a user is authorized to perform certain Platform Server maintenance functions.

The Platform Server uses following two types of facility classes to validate if a user is authorized for a function:

- REXX/ISPF inquiry and profile authorization
- Command Center functions

Procedure

1. Create the REXX/ISPF inquiry and profile facility under CA-ACF2.

The GLOBAL BOSSID parameter, which is defined in [GLOBAL Startup Parameters](#), defines a facility class that is used for authorization checking for:

- The REXX/ISPF interface
- The user profile

If BOSSID=\$FUSION:

- a. To define this facility under CA-ACF2, you can use the following commands:

```
ACF
? SET RULE
? COMPILE STORE
$KEY($FUSION) TYPE(FAC)
```

- b. To make a user a Platform Server ISPF/REXX administrator, give them READ authorization to the profile by using the following command:

```
UID(FUSADM) SERVICE(READ) ALLOW
```

- c. To make a user a Platform Server profile administrator, give them CONTROL authorization to the profile by using the following command:

```
UID(FUSADM) SERVICE(DELETE) ALLOW
```

- d. To complete the resource rule, enter the following command:

```
<ENTER>
? END
```

For more information, see *TIBCO® Managed File Transfer Platform Server for z/OS User's Guide*.

2. (Optional) Create a read only REXX/ISPF inquiry and profile facility.

In addition to the processing defined in [step 1](#), you can define an additional facility class to give ISPF users the authority to view but not update transfer requests. This facility adds the suffix .READ to the GLOBAL BOSSID parameter. When a TSO user has READ access to this facility class, the user can perform the following functions:

- View all transfers.
- Update only transfers where the transfer local user ID matches their TSO user ID.

If BOSSID=\$FUSION:

- a. To define this facility under CA-ACF2, use the following command:


```
ACF
? SET RULE
? COMPILE STORE
$KEY($FUSION.READ) TYPE(FAC)
```

- b. To make a user a Platform Server ISPF/REXX administrator, give them READ authorization to the previous facility by using the following command:

```
UID(FUSADM) SERVICE(READ) ALLOW
```

- c. To complete the resource rule, enter the following command:

```
<ENTER>
? END
```

At this point, the Platform Server will also check facility class \$FUSION.READ. If a user has access to this facility class, the user can read but not update transfer requests.

Note: The .READ facility class is only checked if the user does not have access to the facility class defined by the BOSSID parameter.

3. Create the Command Center facility classes.

Four GLOBAL parameters define the names of the facility classes that are used to determine if a user is authorized for Command Center functions.

See the following parameters defined in the GLOBAL member of the Platform Server SAMPLIB library. For more information, see [GLOBAL Startup Parameters](#).

```
CCC_BROWSE_FACILITY=$CCC.BROWSE
CCC_ALTER_FACILITY=$CCC.ALTER
CCC_ADMIN_FACILITY=$CCC.ADMIN
CCC_TRANSFER_FACILITY=$CCC.TRANSFER
```

- a. To define these Command Center facilities under CA-ACF2, use the following commands:

```
ACF
```

```
? SET RULE
? COMPILE STORE
$KEY($CCC.BROWSE) TYPE(FAC)
$KEY($CCC.ALTER) TYPE(FAC)
$KEY($CCC.ADMIN) TYPE(FAC)
$KEY($CCC.TRANSFER) TYPE(FAC)
```

b. Give the user appropriate rights using any of the following commands:

- To make a Command Center user authorized for all Command Center functions, give them READ authorization to the facility defined by the CCC_ADMIN_FACILITY parameter:

```
$KEY($CCC.ADMIN) TYPE(FAC)
UID(userid) SERVICE(READ) ALLOW
```

- To make a Command Center user authorized for inquiry on completed transfers, give them READ authorization to the facility defined by the CCC_BROWSE_FACILITY parameter:

```
$KEY($CCC.BROWSE) TYPE(FAC)
UID(userid) SERVICE(READ) ALLOW
```

- To make a Command Center user authorized for altering transfers on the Platform Server queue, give them READ authorization to the facility defined by the CCC_ALTER_FACILITY parameter:

```
$KEY($CCC.ALTER) TYPE(FAC)
UID(userid) SERVICE(READ) ALLOW
```

- To make a Command Center user authorized for initiating transfers, give them READ authorization to the facility defined by the CCC_TRANSFER_FACILITY parameter:

```
$KEY($CCC.TRANSFER) TYPE(FAC)
UID(userid) SERVICE(READ) ALLOW
```

LOGONID Definition

TIBCO MFT Platform Server for z/OS must be defined as a full-scope, unrestricted LOGONID with these attributes: MUSASS NO-SMC STC NON-CNCL RESTRICT SUBAUTH

To create the LOGONID, use the following CA-ACF2 SET LID command:

```
INSERT id NAME(Fusion) MUSASS NO-SMC STC NON-CNCL RESTRICT SUBAUTH
```

Where `id` is the started task name, with `Fusion` as the default value. For more information, see *CA-ACF2 Administrator's Guide*.



Note: CA-ACF2 can run either in SAF mode or non-SAF mode. TIBCO MFT Platform Server for z/OS supports only SAF mode.

You must define the Platform Server started task as a multi-user address space, and give access to all datasets that you want to transfer.

The Platform Server opens datasets with the user ID of the initiator, not that of the responder.

OMVS Definitions for Access to HFS Files under ACF2

If you want the Platform Server to access OpenEdition HFS files, you must configure the user associated with the Platform Server started task as a superuser.

The Platform Server reads the HFS directory structure, and performs authorization checking to ensure that users are authorized to access the HFS files. All Platform Server files are opened and accessed under the security environment of the user requesting the transfer.

You must define the following facility resources on your system:

- BPX.SUPERUSER
- BPS.DAEMON

If these resources are not defined on your system, you can define them using the following commands:

```
ACF
?SET RULE
?COMPILE STORE
$(BPX.SUPERUSER) TYPE(FAC)
UID(fususer) SERVICE(READ) ALLOW
$(BPX.DAEMON) TYPE(FAC)
UID(fususer) SERVICE(READ) ALLOW
```

```
<enter>
END
```

Where *fususer* stands for the user associated with the Platform Server started task.

Any user that requires OMVS facilities, including the Platform Server started task user, must be defined with the OMVS segment of the RACF profile to indicate that the user is authorized for OMVS.

Defining the VSAM Files

The Platform Server uses five VSAM clusters during operation. You must run batch jobs to create these VSAM files.

The job control language is supplied in the Platform Server JCL library in the following members:

Member	Description
DEFAUDIT	Defines the VSAM audit file used by the server.
DEFQUEUE	Defines the server work queue dataspace.
DEFPROF	Defines the user profile dataspace.
DEFDNI	Defines the active queue DNI dataspace.
DEFMSGT	Defines the online message help VSAM file.

Defining the Audit VSAM Dataset

You can create a new AUDIT dataset using the DEFAUDIT member located in the Platform Server JCL library.

Procedure

1. Edit the DEFAUDIT member in the JCL library to meet your installation requirements.

Two job steps are included in the DEFAUDIT JCL:

- a. Invoke IEBDG to generate a dummy record.
 - b. Invoke IDCAMS to do the following:
 - Delete the VSAM file if it already exists.
 - Define the VSAM cluster.
 - Repro one record into the VSAM file.
 - Define alternate index 1.
 - Define path 1.
 - Define alternate index 2.
 - Define path 2.
 - Build the index for alternate index 1.
 - Build the index for alternate index 2.
2. Submit the JCL and verify that the linear VSAM file is successfully created.

The GLOBAL parameter MAX_AUDIT_RECORDS defines how many audit records are saved in the Platform Server audit file. The default value is 25,000 records. Each audit record is 4176 bytes. Ensure enough space is allocated to the audit file to contain all of the records defined by the GLOBAL MAX_AUDIT_RECORDS parameter.

When defining the VSAM cluster in step 1, you can set the primary and secondary allocations of the audit file by specifying the RECORDS or Cylinders or Tracks parameter in DEFAUDIT JCL.

i Note: In some cases, it might be necessary to re-create the audit file. The JCL is written to be able to be rerun. Therefore, after deleting the existing audit file, you will get an error code result when the job runs for the first time. If this is the only error code that you receive, you can ignore it.

Defining the Work Queue VSAM dataset

You can create a new QUEUE dataset using the DEFQUEUE member located in the Platform Server JCL library.

Procedure

1. Edit the SYSIN parameters of the DEFQUEUE JCL to specify the volume and file name according to your installation requirements.
IDCAMS is invoked to create a linear VSAM file.
2. Submit the JCL and verify that the linear VSAM file is successfully created.

Note: In some cases, it might be necessary to re-create the work queue (WorkQ) file. The JCL is written to be able to be rerun. Therefore, after deleting the existing WorkQ file, you will get an error code result when the job runs for the first time. If this is the only error code that you receive, you can ignore it.

The amount of disk space that the Platform Server actually uses is determined by the WORKQ parameter on the Platform Server startup JCL. For more information, see [EXEC PARM Fields Used by Platform Server](#).

A work queue entry (4K in size) is created for each Platform Server transaction. Each work queue entry remains in the work queue until completion (successful or unsuccessful). The size of the linear VSAM file depends on the number of transactions anticipated in the work queue at any given instance.

Defining the User Profile Dataspace

To use the Platform Server user profiles, you must define the Platform Server user profile VSAM dataspace.

Procedure

1. Edit the SYSIN parameters of the DEFPROF JCL to specify the volume and file name according to your installation requirements.
IDCAMS is invoked to create a linear VSAM file.
2. Submit the JCL and verify that the linear VSAM file is successfully created.

Note: In some cases, it might be necessary to re-create the profile queue (ProfileQ) file. The JCL is written to be able to be rerun. Therefore, after deleting the existing ProfileQ file, you will get an error code result when the job runs for the first time. If this is the only error code that you receive, you can ignore it.

The amount of disk space that the Platform Server actually uses is determined by the PROFILEQ parameter on the Platform Server startup JCL. For more information, see [EXEC PARM Fields Used by Platform Server](#).

A user profile entry (128 bytes) is created for each user profile definition. Each user profile entry remains in the user profile dataspace until that entry is specifically deleted. The size of the linear VSAM file depends on the user profile definitions anticipated in the user profile dataspace at any given instance.

i Note: This file contains sensitive security information and must be defined to the security subsystem so that only the Platform Server address space can read the file. This file is not updated by any Platform Server batch jobs. Users who delete, define or back up the dataset must also be given access to the dataset.

Defining the DNI Active Queue Dataspace

To use the DNI feature, you must define the Platform Server DNI Active Queue VSAM dataspace.

Procedure

1. Edit the SYSIN parameters of the DEFDNI JCL to specify the volume and file name according to your installation requirements.
IDCAMS is invoked to create a linear VSAM file.
2. Submit the JCL and verify that the linear VSAM file is successfully created.

i Note: In some cases, it might be necessary to re-create the DNI active queue (DNIACTQ) file. The JCL is written to be able to be rerun. Therefore, after deleting the existing DNIACTQ file, you will get an error code result when the job runs for the first time. If this is the only error code that you receive, you can ignore it.

The amount of disk space that Platform Server actually uses is determined by the DNIACTQ parameter on the Platform Server startup JCL. For more information, see [EXEC PARM Fields Used by Platform Server](#).

A DNI Active Queue (512 bytes) is created each time a request is scheduled in the DNI interface. Each entry remains in the DNI Active Queue dataspace until that request is

completed and purged from the queue. The size of the linear VSAM file depends on the number of concurrent DNI requests anticipated in the DNI Active Queue dataspace at any given instance.

The default value of 2000 is large enough to satisfy the requirements of most installations, because DNI requests execute immediately and are purged from the queue when the request is completed.

Defining the MSGTEXT Cluster

To use the online message help, you must define the Platform Server MSGTEXT VSAM dataset.

Procedure

1. Edit the SYSIN parameters of the DEFMSGT JCL to specify the volume and file name according to your installation requirements.
IDCAMS is invoked to create a linear VSAM file.
2. Submit the JCL and verify that the VSAM file is successfully created.

Note: In some cases, it might be necessary to re-create the MSGTEXT cluster. The JCL is written to be able to be rerun. Therefore, after deleting the existing MSGTEXT cluster, you will get an error code result when the job runs for the first time. If this is the only error code that you receive, you can ignore it.

The MSGTEXT file contains the same message text and message explanations as provided in *TIBCO Managed File Transfer Platform Server for z/OS Message Manual*. You can access this information by using the FUSMSG REXX exec or through the primary Fusion ISPF menu.

Defining the Sequential Files

You must execute the CRTCONF and CRTDNI batch jobs in the Platform Server JCL library to create datasets used by the Platform Server.

Procedure

1. Submit CRTCONF to create the CONFIG dataset.

The CONFIG dataset contains node and list definitions. The second step of the CRTCONF job copies the CONFIG member of the Platform Server SAMPLIB to the new CONFIG dataset.

2. Submit CRTDNI to create the DNICFG dataset.

The DNICFG dataset contains the DNI configuration members. The second step of the CRTDNI job copies the DNI samples from the Platform Server SAMPLIB to the new DNICFG dataset.

Creating a TSO Logon Procedure

If you are adding a new logon JCL procedure for the Platform Server, you might have to define this to RACF, CA-ACF2 or CA-Top Secret. Check with your security administrator if you are uncertain.

i Note: The required minimum level of ISPF is version 3 release 2.

See the following primary ISPF window and action bar:

```

Transfer Utilities Manage Node Options Help
MFT Platform Server Primary Menu Version 7.2.0 Maint CZ01963 7.2.0

Command ==> _____

1  Manage File Transfers          Display/Change File Transfers
2  NODE info:  _____        Online Node Inquiry
3  Send   :
   A  File                        Send a file to a file
   B  Job                         Send a file to a job
   C  Print                       Send a file to a printer
   D  Command                     Send a command
4  Receive:
   A  File                        Receive a file to a file

```

```

      B  Job                Receive a file to a job
      C  Print              Receive a file to a printer
5     Messages:  _____ Online Message Inquiry

      6  Script:            Schedule Script to execute
      7  View History:      UNIX, Windows and IBM i history

MFT Platform Server CONFIG ==> _____

```

Procedure

1. Allocate the following datasets in the TSO PROC:

```

//ISPMLIB      DD      DSN=MFT.MSGS,DISP=SHR
//ISPPLIB      DD      DSN=MFT.PANELS,DISP=SHR
//SYSEXEC      DD      DSN=MFT.EXECS,DISP=SHR
//STEPLIB      DD      DSN=MFT.LOADLIB,DISP=SHR
//FUSMSG       DD      DSN=MFT.MSGTEXT,DISP=SHR

```

Optional DD statements:

```

//FUSCFG DD
DSN=MFT.DEVL.EXECS(member),DISP=SHR

```

- By using this FUSCFG DD statement, you can override the standard Platform Server method of detecting the FUSCFG configuration.
- If FUSCFG DD is not defined, the Platform Server scans the SYSEXEC DD for a member called FUSCFG.
- If the FUSCFG DD is defined without a member, the Platform Server scans for member FUSCFG.
- If the FUSCFG DD is defined with a member name, the Platform Server reads the defined member.

For more information on the FUSCFG file configuration, see [Defining and Configuring the FUSCFG File](#).

Adding Platform Server ISPF Interface to TSO Logon Procedure

After creating a TSO logon procedure, you must add the Platform Server ISPF user interface to the procedure.

Procedure

1. Append the previous datasets to the ddnames in a TSO logon procedure to allocate the Platform Server ISPF user interface.

(Optional) Dynamically Adding Platform Server Libraries to ISPF

You can use the FUSCLIST member in the SAMPLIB library to dynamically add Platform Server libraries to the ISPF environment.

This procedure does not dynamically add the Platform Server STEPLIB to the ISPF PROC. You must add the Platform Server STEPLIB to the ISPF PROC in one of the following ways:

- Add the Platform Server LOADLIB to the ISPF PROC STEPLIB.
- Add the Platform Server LOADLIB to the z/OS link list.
- Create a new PROC that has the Platform Server LOADLIB in the STEPLIB.
- Use the following command to activate the STEPLIB. This must be done in TSO before ISPF starting:

```
TSOLIB ACTIVATE DSNAME(FUSION.LOAD)
```

After the Platform Server LOADLIB is added to the ISPF link list or STEPLIB, you can follow the following steps to customize the REXX exec:

Procedure

1. Change the following statements marked in bold text to point to the correct Platform Server libraries.

See the following FUSCLIST member in SAMPLIB:

```
/* REXX EXEC TO INVOKE ISPF INTERFACE          */
/*NOTE: LOADLIB MUST BE IN STEPLIB OR LINKLST */
X=MSG('OFF')
ARG ARG1 ARG2 ARG3                          /* SAVE INPUT PARMS*/
/*****UPDATE THE FOLLOWING STATEMENTS TO POINT TO*****/
/*****THE CORRECT INSTALL FILES*****/
ISPPLIB= 'MFT.PANELS'
ISPMLIB= 'MFT.MSGS'
SYSEXEC= 'MFT.EXECS'
FUSMSG = 'MFT.MSGTEXT'
FUSCFG = 'MFT.EXECS'
/*****CHANGE THE PREVIOUS STATEMENTS*****/
```

```

/*****/
/* ALLOCATE ALL LIBRARIES AND FILES          */
/*****/
"ALTLIB ACTIVATE APPLICATION (EXEC) DATASET('SYSEXEC') UNCOND"
"ISPEXEC LIBDEF SYSEXEC DATASET ID('SYSEXEC') UNCOND"
"ISPEXEC LIBDEF ISPPLIB DATASET ID('ISPPLIB') UNCOND"
"ISPEXEC LIBDEF ISPMLIB DATASET ID('ISPMLIB') UNCOND"
"ALLOC FI(FUSMSG) SHR DA('FUSMSG')"
"ALLOC FI(FUSCFG) SHR DA('FUSCFG')"
/*****/
/* EXECUTE FUSION EXEC                      */
/*****/
"ISPEXEC SELECT CMD(FUSION" ARG1 ARG2 ARG3") NEWAPPL(PROM) PASSLIB"
/*****/
/* FREE ALL LIBRARIES AND FILES            */
/*****/
"ISPEXEC LIBDEF ISPPLIB"
"ISPEXEC LIBDEF ISPMLIB"
"ISPEXEC LIBDEF SYSEXEC"
"ALTLIB DEACTIVATE APPLICATION(CLIST)"
"FREE FI(FUSMSG)"
"FREE FI(FUSCFG)"
X=MSG('ON')
EXIT

```

2. Copy the REXX exec to a library that is currently in the ISPF SYSEXEC or SYSPROC DD statements. Alternatively, you can execute this script using the following syntax:

```
EXEC 'your.library(member)'
```

Where `'your.library(member)'` points to the updated REXX exec.

(Optional) Adding Configuration Parameters to OMVS User Profile

If you want to use the Platform Server REXX execs under OMVS, you must add the configuration parameters to your user profile. Otherwise, the REXX execs do not work.

Procedure

1. Add the following lines to the user profile for any user who wants to use the Platform Server REXX execs under OMVS.

```
export STEPLIB=MFT.LOADLIB
export PATH=$PATH:/fusexec
export FUSEXEC=MFT.EXECS
```

- The second statement adds the Platform Server EXEC directory to the OMVS path. If the EXEC directory is not added to the OMVS path, you must issue the `cd` command to position yourself in the Platform Server EXEC directory before issuing a request.

i Note: The actual directory created in [Copying Platform Server REXX execs to an HFS Directory](#) must replace the characters `/fusexec`.

- The last statement is required and specifies the location of the Platform Server configuration member, `FUSCFG`. This must point to a z/OS PDS containing the `FUSCFG` member.

(Optional) Adding Platform Server to an ISPF Selection Panel

You can update your ISPF panels to add the Platform Server as a menu choice.

Procedure

1. To include the Platform Server as a choice to the selection panel, you can add it as a choice in the `)BODY` and `)PROC` sections of the master selection panel.

See the following example for your reference:

```
)BODY

Existing Panel

SELECT OPTION===>_ZCMD

+0 &PARM
+1 &BROWSE
+2 &EDIT
```

```

Add This
+F1 &MFT Platform Server

Existing Panel

)PROC

&ZSEL=TRANS(&ZQ
0, 'PANEL(ISPOPTA)'
1, 'PGM(ISRBRO)'
2, 'PGM(ISREDIT)'

Add This
F1, 'CMD(FUSION) NEWAPPL(PROM)'
```

Defining VTAM Resources for Systems Using SNA

You are required to configure VTAM resources only if your environment uses SNA. To set up the environment for SNA, you have to define a VTAM APPLID. This enables mainframe to mainframe communications through SNA.

SNA communication is supported for the following cases:

- Platform Server for z/OS to Platform Server for z/OS file transfers
- ISPF communication to the Platform Server for z/OS started task
- Batch communication to the Platform Server for z/OS started task

Defining an APPLID for the Platform Server

You must define an APPLID for use with the Platform Server.

The following is an example of an APPLID definition which can be found in the APPLMFT member of the Platform Server SAMPLIB.

```

*****
*
*  SAMPLE APPLID FOR SNA COMMUNICATIONS
*
*  =====
*
```

```

*****
APPLFUSN VBUILD TYPE=APPL
FUSNAPPL APPL AUTH=(NOPO,ACQ,VPACE), -
          APPC=YES, -
          ACBNAME=FUSNAPPL, -
          MODETAB=USERMODE, -
          SONSCIP=YES, -
          VPACING=5, -
          DLOGMOD=#BATCH

*****
**
* IF YOU WOULD LIKE TO USE PARALLEL SESSIONS WITH CYBERFUSION YOU SHOULD

* ADD THE FOLLOWING LINES TO YOUR APPL DEFINITION. YOU SHOULD
* UNCOMMENT THE LINES BELOW AND MOVE THEM BEFORE THE DLOGMOD ENTRY
* IN ORDER TO KEEP THE CONTINUATIONS IN TACT.
*
*          PARSESS=YES,      THIS IS NEEDED TO DO PARALLEL SESSIONS -
*          DSESLIM=10,      PARALLEL SESSION LIMIT -
*          DMINWNR=5,      MINIMUM CONTENTION LOSERS -
*          DMINWNL=5,      MINIMUM CONTENTION WINNERS -

*****
**

```

i Note: For every major node defined, the VBUILD statement must be placed as the first definition statement.

For more information on defining an APPLID, see the IBM manual: *z/OS Communications Server SNA Resource Definition Reference*.

Operand	Description
ACBNAME	Defines the Access Control Block name. This operand also defines the minor node name for the application. ACB stands for Access Control Block.
APPC	Tells VTAM that the application defined here might use the basic functions of LU6.2.

Operand	Description
	This operand is specific to VTAM.
APPLID	Defines the name of the application program major node.
AUTH	Defines this subsystem to have the ability to acquire an LU (ACQ), and specifies that this subsystem adheres to the methodology of VPACING to LUs (VPACE).
DLOGMOD	<p data-bbox="464 613 1401 682">Defines the name of the default session parameter entry that is used with this application.</p> <p data-bbox="464 716 1401 785">The session parameters are coded in the LOGMODE entry of the MODETAB in VTAM. This mode name must also be defined on the remote system.</p>
FUSNAPPL	<p data-bbox="464 835 1081 863">Defines the minor node name to the application.</p> <p data-bbox="464 896 1328 968">The assigned name must be unique within a network. The default is FUSNAPPL, but you can choose the name as you like.</p>
MODETAB	<p data-bbox="464 1018 1386 1129">Defines the logon mode table to be used to associate each logon mode name with a set of session parameters for the application program if the application program participates as the secondary logic unit (SLU).</p> <p data-bbox="464 1163 1386 1268">VTAM searches the logon mode table for the LOGMODE that is used by the application when it participates as a SLU. This mode table contains entries that define parameters for SLUs.</p>
PARSESS	<p data-bbox="464 1318 1321 1388">Specifies that this subsystem can have multiple sessions with other applications on an LU-LU session.</p> <p data-bbox="464 1421 1401 1488">If you plan to have multiple simultaneous transfers between two LUs, you have to set this parameter to YES.</p>
SONSCIP	Tells VTAM that it can terminate sessions with the SLU on behalf of the application.
TYPE=APPL	<p data-bbox="464 1659 1292 1686">Defines to VTAM that this is an application major node definition.</p> <p data-bbox="464 1719 1401 1791">No other optional operands are applicable to define an application major node. APPL is also used by default if the TYPE operand is not coded.</p>

Operand	Description
	One APPL definition statement is required for each application that is to be identified to VTAM.
VPACING	Specifies the maximum number of normal flow requests that another logical unit can send to this session before waiting to receive a pacing response. If this value is too low, it might result in slower transfers.

Defining APPLIDs for Platform Server Batch Interface and ISPF Panels

If you use the Platform Server for z/OS client software using SNA LU6.2 communications, you must define the Platform Server client APPLIDs on any z/OS instance on which the client runs.

For multiple clients to run simultaneously, the Platform Server requires that you add a suffix from 00 to 99 to a 1-character to 6-character prefix. Typically, ACBNAME is defined the same across all the z/OS systems while APPLID is unique for each system.

See the following sample of Platform Server APPLID definitions:

```
FUSNA00  APPL  AUTH=(NOPO,ACQ,VPACE),DLOGMOD=#BATCH,      -
          APPC=YES,ACBNAME=FUSN00,MODETAB=USERMODE
FUSNA01  APPL  AUTH=(NOPO,ACQ,VPACE),DLOGMOD=#BATCH,      -
          APPC=YES,ACBNAME=FUSN01,MODETAB=USERMODE
FUSNA02  APPL  AUTH=(NOPO,ACQ,VPACE),DLOGMOD=#BATCH,      -
          APPC=YES,ACBNAME=FUSN02,MODETAB=USERMODE
FUSNA03  APPL  AUTH=(NOPO,ACQ,VPACE),DLOGMOD=#BATCH,      -
          APPC=YES,ACBNAME=FUSN03,MODETAB=USERMODE
FUSNA04  APPL  AUTH=(NOPO,ACQ,VPACE),DLOGMOD=#BATCH,      -
          APPC=YES,ACBNAME=FUSN04,MODETAB=USERMODE
FUSNA05  APPL  AUTH=(NOPO,ACQ,VPACE),DLOGMOD=#BATCH,      -
          APPC=YES,ACBNAME=FUSN05,MODETAB=USERMODE
FUSNA06  APPL  AUTH=(NOPO,ACQ,VPACE),DLOGMOD=#BATCH,      -
          APPC=YES,ACBNAME=FUSN06,MODETAB=USERMODE
FUSNA07  APPL  AUTH=(NOPO,ACQ,VPACE),DLOGMOD=#BATCH,      -
          APPC=YES,ACBNAME=FUSN07,MODETAB=USERMODE
FUSNA08  APPL  AUTH=(NOPO,ACQ,VPACE),DLOGMOD=#BATCH,      -
          APPC=YES,ACBNAME=FUSN08,MODETAB=USERMODE
```

```
FUSNA09  APPL  AUTH=(NOPO,ACQ,VPACE),DLOGMOD=#BATCH,
          APPC=YES,ACBNAME=FUSN09,MODETAB=USERMODE
```

For more information on the VTAM APPL operands, see [Defining an APPLID for the Platform Server](#).

The Startup JCL

You can find an example of the Platform Server startup JCL in the MFTSTC member of the Platform Server JCL library.

The Platform Server must be contained in an authorized load library. Before bringing up the Platform Server, you must define your environment.

For more information on all GLOBAL startup parameters, see [Defining Local Resources and Initialization Defaults](#).

The following PROC brings up the Platform Server for z/OS.

```
//MFTSTC  PROC QUAL=MFT,           <<== High Level Qualifier
//          OUTC=X,                <<== Message Output Class
//          DUMPC=X,              <<== Dump Output Class
//          START=WARM,           <<== WARM start option set
//          WORKQ=1000            <<== WORKQ Size
//*
//*****
//* This Procedure starts the MFT Platform Server for z/OS
//*
//* The Started Task JCL requires the TIME=NOLIMIT be coded on the
//* EXEC statement. The parameter PARM='WORKQ=xxxxxx' keyword on
//* the EXEC statement is optional. The WORKQ specification
//* controls the number of entries in the Server Work Queue.
//* Valid Values are:
//* Default = 100
//* Minimum = 100
//* Maximum = 999999
//*****
//*
//MFTSTC  EXEC PGM=OSSERVER,
//          TIME=NOLIMIT,
//          REGION=0M,
//          PARM='WORKQ=&WORKQ,START=&START'
//*
```

```

//*****
//* STEPLIB - Points to the Server Load Library and must be APF      *
//*      Authorized.                                               *
//* * NOTE * The TCP/IP Runtime is only required if you are using *
//*      IBM's TCP/IP and the IBM 'C' Runtime library is not      *
//*      in the LNKLST and you are using IP Names rather than     *
//*      IP Addresses in your node definitions                     *
//*      *
//* * SSL * GSK.SGSKLOAD is only required for SSL                 *
//*      *
//* * MQ * MQSER.SCSQLOAD is only required for MQ                 *
//* * MQ * MQSER.SCSQAUTH is only required for MQ                 *
//*****
//STEPLIB DD DISP=SHR,DSN=&QUAL..LOADLIB
//*      DD DISP=SHR,DSN=TCP.C.RUNTIME.LIBRARY
//* *SSL* DD DISP=SHR,DSN=GSK.SGSKLOAD
//* *MQ* DD DISP=SHR,DSN=MQSER.SCSQLOAD
//* *MQ* DD DISP=SHR,DSN=MQSER.SCSQAUTH
//*
//*****
//* AUDIT - These datasets point to the cluster containing Audit *
//* AUDPATH1 information about transfers completed                 *
//* AUDPATH2                                                         *
//*****
//AUDIT DD DISP=SHR,DSN=&QUAL..AUDIT
//AUDPATH1 DD DISP=SHR,DSN=&QUAL..AUDIT.PATH1
//AUDPATH2 DD DISP=SHR,DSN=&QUAL..AUDIT.PATH2
//*
//*****
//* PROFILE - Points to the Profile dataset that contains user *
//*      profile information                                       *
//*****
//PROFILE DD DISP=SHR,DSN=&QUAL..PROFILE
//*
//*****
//* DIVDDSTM - Points to the Work Queue dataset                    *
//*****
//DIVDDSTM DD DISP=SHR,DSN=&QUAL..QUEUE
//*
//*****
//* DNIACT - Points to the dataset that contains information on *
//*      currently active DNI requests                             *
//*****
//DNIACT DD DISP=SHR,DSN=&QUAL..DNIACT
//*
//*****

```

```

/* DNICFG - Points to the DNI configuration dataset *
/*****
//DNICFG DD DISP=SHR,DSN=&QUAL..DNICFG
//*
/*****
/* GLOBAL - Points to the Configuration parameters *
/*****
//GLOBAL DD DISP=SHR,DSN=&QUAL..SAMPLIB(GLOBAL)
//*
/*****
/* CFACCESS - Points to Access Control Configuration *
/*****
//CFACCESS DD DISP=SHR,DSN=&QUAL..SAMPLIB(CFACCESS)
//*
/*****
/* CFALIAS - Points to the CFALIAS Configuration *
/*****
//CFALIAS DD DISP=SHR,DSN=&QUAL..SAMPLIB(CFALIAS)
//*
/*****
/* SUBJCL - Points to the SUBJCL Configuration *
/*****
//SUBJCL DD DISP=SHR,DSN=&QUAL..SAMPLIB(SUBJCL)
//*
/*****
/* SSLAUTH - Points to the SSL Configuration *
/*****
//SSLAUTH DD DISP=SHR,DSN=&QUAL..SAMPLIB(SSLAUTH)
//*
/*****
/* TRCLASS - Points to the Transfer Class Configuration *
/*****
//TRCLASS DD DISP=SHR,DSN=&QUAL..SAMPLIB(TRCLASS)
//*
/*****
/* CONFIG - Points to the Node Definitions *
/*****
//CONFIG DD DISP=SHR,DSN=&QUAL..CONFIG
//*
/*****
/* OSIMSGS - Points to a SYSOUT dataset that contains messages *
/* issued by started task *
/*****
//OSIMSG DD SYSOUT=&OUTC,DCB=(RECFM=F,LRECL=200)
//*
/*****

```

```

/* INCLUDE - Points to the Installation's Include library.      *
/*          This is required only if INCLUDE stmts will be used *
/*          within scripts run under the CF started task.      *
/*          Uncomment the statement below to use this feature.  *
/******
/*INCLUDE DD DISP=SHR,DSN=&QUAL..INCLUDE
/*
/******
/* SCRIPT - Points to the Installation's SCRIPT libraries.     *
/*          This is required only if scripts are scheduled     *
/*          specifying a member without a DSN.                  *
/*          Uncomment the statements below to use this feature. *
/******
/*SCRIPT DD DISP=SHR,DSN=&QUAL..SCRIPT.LIBRARY1
/*          DD DISP=SHR,DSN=&QUAL..SCRIPT.LIBRARY2
/*          DD DISP=SHR,DSN=&QUAL..SCRIPT.LIBRARYx
/*
/******
/* VTAMTRAC - Points to a SYSOUT dataset that contains trace  *
/*            information for SNA communications                 *
/******
//VTAMTRAC DD SYSOUT=&OUTC
/*
/******
/* TCPTRAC - Points to a SYSOUT dataset that contains trace  *
/*            information for TCP communications                 *
/******
//TCPTRAC DD SYSOUT=&OUTC
/*
/******
/* SYSTCPD - Points to the same dataset as SYSTCPD in your IBM *
/*            TCP/IP Start Up JCL                               *
/* * NOTE * This is only required if you are using IBM's TCP/IP *
/*            and using IP Names rather than IP Addresses      *
/******
//SYSTCPD DD DISP=SHR,DSN=TCP.DATA(TCPDATA)
/*
/******
/* TCPSBCS - Points to the TCPIP SBCS Conversion library      *
/* TCPCHBIN - Points to the TCPIP DBCS Conversion file for Chinese *
/*            characters                                         *
/* TCPSCBIN - Points to the TCPIP DBCS Conversion file for     *
/*            Simplified Chinese characters                     *
/* TCPKJBIN - Points to the TCPIP DBCS Conversion file for KANJI *
/*            characters                                         *
/* TCPHGBIN - Points to the TCPIP DBCS Conversion file for Korean *

```

```

/*          characters                                     *
/* Note that these DD statements are commented out.     *
/* Uncomment them only if you require the SBCS or DBCS conversion *
/* features.                                           *
/******
/**TCP SBCS   DD DISP=SHR,DSN=TCPIP.SEZATCPX
/**TCPCHBIN  DD DISP=SHR,DSN=TCPIP.STANDARD.TCPCHBIN
/**TCPSCBIN  DD DISP=SHR,DSN=TCPIP.STANDARD.TCPSCBIN
/**TCPKJBIN  DD DISP=SHR,DSN=TCPIP.STANDARD.TCPKJBIN
/**TCPHGBIN  DD DISP=SHR,DSN=TCPIP.STANDARD.TCPHGBIN
/**
/*SYSUDUMP   DD SYSOUT=&DUMPC
/*SNAP       DD SYSOUT=&OUTC
/**
/**-----END OF PROC "FUSION"-----*

```

EXEC PARM Fields Used by Platform Server

You can use the PARM field on the z/OS EXEC JCL statement to define parameters for the Platform Server.

See the following table for the parameters you can define.

Parameter	Description
WORKQ	<p>Defines the number of requests that can be on the Platform Server request queue at any one time.</p> <p>The default value of 100 is typically too small for anything but a test system. You must specify a value larger than the total amount of locally initiated and remotely initiated transfers. This includes transfers that are being executed, inactive and in a hold state.</p> <p>For example, the following statement allocates up to 5000 file transfer requests at any one time.</p> <pre>//STEP1 EXEC PGM=OSSERVER,PARM='WORKQ=5000'</pre>
PROFILEQ	<p>Defines the number of user profile entries that can fit into the Platform Server user profile dataspace.</p> <p>The default value of 4000 is sufficient for most systems and fits within a</p>

Parameter	Description
DNIACTQ	<p>single cylinder on most DASD volumes.</p> <p>For example, the following statement allocates dataspace for up to 6000 user profile entries.</p> <pre data-bbox="469 457 1305 485">//STEP1 EXEC PGM=OSSERVER,PARM='WORKQ=5000,PROFILEQ=6000'</pre>
DNIACTQ	<p>Defines the number of DNIACT entries that can fit into the Platform Server dataspace.</p> <p>The default value of 2000 entries is sufficient for most systems.</p> <p>For example, the following statement allocates dataspace for up to 20000 DNIACT entries.</p> <pre data-bbox="469 798 1305 825">//STEP1 EXEC PGM=OSSERVER,PARM='WORKQ=1000,DNIACTQ=10000'</pre>
DNICFGQ	<p>Defines the number of DNI configuration entries that can fit into the internal DNICFG dataspace.</p> <p>The default value of 1000 means that 1000 DNI config members can be activated.</p> <p>For example, the following statement defines 20000 DNI config members can be activated.</p> <pre data-bbox="469 1176 1305 1203">//STEP1 EXEC PGM=OSSERVER,PARM='WORKQ=1000,DNICFGQ=20000'</pre>
SSLDEBUG	<p>Defines whether SSL debugging is turned on at startup.</p> <p>Valid values are:</p> <ul data-bbox="516 1371 1321 1518" style="list-style-type: none"> • NO: SSL debugging is not turned on. This is the default value. • YES: SSL debugging is turned on. • HIGH: detailed SSL debugging is turned on. <p>The SSLDEBUG operator command can be used to turn on or turn off SSL debugging. For example, the following statement turns on SSL HIGH debugging at startup.</p> <pre data-bbox="469 1680 1143 1707">//STEP1 EXEC PGM=OSSERVER,PARM='SSLDEBUG=HIGH'</pre>
START	<p>Defines whether the Platform Server WORKQ is initialized at startup.</p>

Parameter	Description
	<p>Valid values are:</p> <ul style="list-style-type: none"> • WARM: the WORKQ is not initialized at startup. Data in the WORKQ is retained and processed by the Platform Server. • COLD: the WORKQ is initialized at startup. All information in the WORKQ is lost. This option should only be used in severe situations where the Platform Server started task cannot be brought up because the WORKQ has been corrupted. <p>For example, the following statement WARM start the Platform Server WORKQ.</p> <pre>//STEP1 EXEC PGM=OSSERVER,PARM='START=WARM'</pre>

STEPLIB

GSK.SGSKLOAD should be added to the STEPLIB if you are using SSL/TLS and if GSK.SGSKLOAD is not in the MVS LINKLIST.

The MQ libraries are only required when MQ is used. To run MQ, you must have the following MQ libraries authorized and placed in the STEPLIB of the startup JCL: SCSQLOAD and SCSQAUTH.

See the following example for your reference.

```
//STEPLIB DD DISP=SHR,DSN=&QUAL..LOADLIB
//* DD DISP=SHR,DSN=TCP.C.RUNTIME.LIBRARY
//* *SSL* DD DISP=SHR,DSN=GSK.SGSKLOAD
//* *MQ * DD DISP=SHR,DSN=MQSER.SCSQLOAD
//* *MQ * DD DISP=SHR,DSN=MQSER.SCSQAUTH
```


Performing an Upgrade

When performing a Platform Server for z/OS upgrade, the existing configurations and datasets are used. This section describes the datasets that require additional updating.

Updating the VSAM Datasets

You have to update the VSAM datasets after upgrading TIBCO MFT Platform Server for z/OS to the current release.

In the initial Platform Server installation, you have created five VSAM clusters using the JCL supplied in the following members in the Platform Server JCL library.

Member	Description
DEFAUDIT	Defines the audit trail used by the server.
DEFQUEUE	Defines the server disk backup of its work queue .
DEFPROF	Defines the user profile dataspace.
DEFDNI	Defines the active queue for the Platform Server DNI.
DEFMSGT	Defines the online message help.

The AUDIT, WORKQ, and MSGT datasets might have to be updated to work with the current release. The PROFILE and DNI datasets are not changed and do not have to be updated.

The AUDIT Dataset

The AUDIT dataset, pointed to by the AUDIT DD card in the FUSION started task, contains information about transfers that are completed.

i Note: If you are upgrading from version 7.0 and want to continue using your existing AUDIT dataset, then you can do so without any updates and can skip this section.

If you want to start with a new AUDIT dataset or you are running a version older than 6.2, you must run the DEFAUDIT job in the Platform Server JCL dataset to create a new AUDIT dataset. For more information, see [Updating the VSAM datasets](#).

If you are upgrading from Platform Server version 6.2, 6.4, 6.5, or 6.5.1 and want to continue to use your existing AUDIT file, you must run the CPYAUDIT job. For more information, see [Running the CPYAUDIT Job](#).

Running the CPYAUDIT Job

If you are upgrading from Platform Server version 6.2, 6.4, 6.5, or 6.5.1 and want to continue to use your existing AUDIT file with the Platform Server for z/OS, then run the CPYAUDIT job which is provided in the Platform Server JCL dataset.

This CPYAUDIT JCL converts the variable length audit records to fixed length records, reads the variable length VSAM audit file and create a sequential fixed length backup file. This backup file is then used as an input file when you define the audit VSAM dataset using the DEFAUDIT JCL. For more information, see [Defining the Audit VSAM dataset](#).

Updating the Audit VSAM Dataset

You have to define and run the DEFAUDIT JCL, and edit the DEFAUDIT member in the JCL library to meet your installation requirements.

If you are upgrading from an earlier version of Platform Server, you must comment out the following from the DEFAUDIT JCL:

```
/*REPRO          -          */
/*  INDATASET(MFT.AUDIT.BACKUP) -          */
/*  OUTDATASET(MFT.AUDIT)          */
```

If you are creating a new AUDIT dataset, you can simply edit the JCL according to your installation requirements. For more information, see [Defining the Audit VSAM dataset](#).

The WORKQ Dataset

The WORKQ dataset, pointed to by the DIVDDSTM DD card in the FUSION started task, contains information about transfers that are scheduled to run on z/OS and information on any transfers that are in process.

If you are upgrading from the Platform Server version 6.2 or higher and want to continue to use your existing QUEUE dataset, then you can do so without any updates and skip this section.

If you want to start with a new QUEUE dataset or you are running a version earlier than 6.2, you must use the DEFQUEUE job in the Platform Server JCL dataset to create a new QUEUE dataset as described in [Defining the Work Queue VSAM dataset](#).

The MSGTEXT Cluster

The MSGTEXT cluster, pointed to by the FUSMSG DD card in the TSO JCL, contains information about online messages that are displayed. This file is used by the Platform Server ISPF "Online Message Inquiry" function.

If you want to start with a new MSGTEXT cluster or you are running a version earlier than 6.2, you must run the DEFMSGT job in the Platform Server JCL dataset to create a new MSGTEXT cluster as described in [Defining the MSGTEXT Cluster](#).

Platform Server Configuration

To use this product smoothly, you have to configure a new installation of TIBCO MFT Platform Server for z/OS.

Defining Local Resources and Initialization Defaults

Before bringing up the Platform Server, you must define your environment. You must define the local system in the dataset whose ddname is GLOBAL in the Platform Server startup JCL. You have to define each remote system in a separate member of the partitioned dataset whose ddname is CONFIG in the Platform Server startup JCL.

The server requires certain information to initialize and function properly. This information is kept in the GLOBAL file. At initialization time, the server reads the DD statement GLOBAL and goes to the dataset that is referenced in it. The server reads the information that is server specific from that dataset. Before starting the server for the first time, you must specify the appropriate parameters in the GLOBAL dataset.

GLOBAL Startup Parameters

The Platform Server uses some local parameters to initialize and perform functions.

You must specify these parameters in a dataset, with a ddname of GLOBAL member. You can find a sample GLOBAL member in the Platform Server SAMPLIB library.

See the following table for the parameters that are included in the GLOBAL member.

Parameter	Description
ACCEPT_VERIFIED_USER	<p>Defines whether users verified on other z/OS platforms can be used on this system without a password.</p> <p>When a user submits a file transfer request on a z/OS system, by default the local user ID is sent to the remote system.</p> <p>Valid values are YES and NO. The default value is NO.</p>

Parameter	Description
ALLOW_MANAGE_REQUESTS	<p data-bbox="558 296 1409 365">Defines whether transfer management requests initiated through TCP or SNA are accepted.</p> <p data-bbox="558 401 1409 548">This parameter applies to these Platform Server interfaces: batch, script, API, REXX and ISPF, and does not apply to transfers managed by Command Center. Management requests are typically ISPF or REXX transfer inquiries.</p> <p data-bbox="558 583 764 611">Valid values are:</p> <ul data-bbox="607 642 1409 1020" style="list-style-type: none"><li data-bbox="607 642 1409 711">• YES: TCP and SNA transfer requests are accepted. This is the default value.<li data-bbox="607 743 1409 846">• NO: TCP and SNA transfer requests are not accepted. Only requests initiated on the same server and scheduled through cross memory are accepted.<li data-bbox="607 877 1409 1020">• NODE: The node ALLOW_MANAGE_REQUESTS parameter defines whether TCP or SNA requests are accepted. If a node definition is not defined for an incoming transfer management request, the request is denied.
ALLOW_TRANSFER_REQUESTS	<p data-bbox="558 1073 1409 1142">Defines whether transfer schedule requests initiated through TCP or SNA are accepted.</p> <p data-bbox="558 1178 1409 1281">This parameter applies to these Platform Server interfaces: batch, script, API, REXX and ISPF, and does not apply to transfers scheduled by Command Center.</p> <p data-bbox="558 1316 764 1344">Valid values are:</p> <ul data-bbox="607 1375 1409 1759" style="list-style-type: none"><li data-bbox="607 1375 1409 1444">• YES: TCP and SNA transfer requests are accepted. This is the default value.<li data-bbox="607 1476 1409 1579">• NO: TCP and SNA transfer requests are not accepted. Only requests initiated on the same server and scheduled through cross memory are accepted.<li data-bbox="607 1610 1409 1759">• NODE: The node ALLOW_TRANSFER_REQUESTS parameter defines whether TCP or SNA requests are accepted. If a node definition is not defined for an incoming transfer schedule request, the request is denied.

Parameter	Description
APPC-APPLID	<p>Defines the APPLID for VTAM. This operand also defines the minor node name for the application.</p> <p>You must have this parameter defined for the Platform Server to initialize SNA communications. If this parameter is not defined, the Platform Server does not attempt to start Platform Server SNA communications.</p> <p>Valid values are <i>applid</i> and FUSNAPPL.</p>
APPC_GENERIC_NAME	<p>Defines the resource as a VTAM generic resource to enable SYSPLEX support within TIBCO MFT Platform Server for z/OS.</p> <p>If this parameter is defined without the necessary SYSPLEX software and hardware support, a message is displayed indicating that SYSPLEX support is disabled. The valid value is an 8-character VTAM generic name.</p>
AUDIT_TEMP_ERRORS	<p>Defines whether the Platform Server writes audit records for temporary network of file errors.</p> <p>This parameter only applies when an error is retried. When the retries are exhausted, the Platform Server always writes an audit record when the request is purged from the active queue.</p> <p>Valid values are:</p> <ul style="list-style-type: none">• NO: Temporary errors are not written to the audit log or to SMF.• YES: Temporary records are written to the audit log and to SMF.• BOTH: Temporary records are written to the audit log and to SMF.• AUDIT: Temporary records are written to the audit log but not to SMF.• SMF: Temporary records are written to SMF but not to the audit log.

Parameter	Description
AUTOENABLE	<p>Defines how members of the CONFIG DD are enabled at startup.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • NO: Only members defined in the CONFIG member or overridden by the GLOBAL CONFIG member are enabled. This is the default value. • YES: All members of the CONFIG DD statement are enabled unless the member is defined with the AUTOENABLE=NO parameter. If you want to use the Command Center function that creates node definitions, you must specify AUTOENABLE=YES.
BOSSID	<p>Specifies the name of a FACILITY resource used to determine if a user is authorized to administer the work queue or maintain profile information.</p> <p>Valid values are ANY and a facility name. This parameter supports the following functions:</p> <ul style="list-style-type: none"> • Restricts the people that are able to administer the Platform Server work queue. <p>By specifying a facility class name in this field, the Platform Server gives rights to administer the Platform Server work queue only to those users that have READ access to this facility class.</p> <p>By specifying ANY or by leaving this field blank, the Platform Server gives anybody rights to access the work queue: all valid users have the same rights and privileges. Any valid user can browse the work queue and display or put on hold or purge activities waiting to execute. You might have to consult your mainframe security specialist for a facility that can be used to limit access to the Platform Server work queue.</p> • Restricts a user to view but not to update access to the Platform Server work queue. <p>By creating a facility class with a prefix of the BOSSID and a</p>

Parameter	Description
CA7	<p>suffix of .READ, you can assign users that can view all Platform Server transfers but can only update transfers where the local user ID matches their TSO user ID.</p> <ul style="list-style-type: none"> • Determines if a user is a Platform Server administrator with ADD/DELETE/LIST capability of Platform Server profiles for all users. <p>By specifying a facility class name in this field, the Platform Server assigns Platform Server administrator rights only to those users that have CONTROL access to this facility class.</p> <p>By specifying ANY or by leaving this field blank, the Platform Server gives anybody rights to perform the Platform Server administrator functions for all users. This means that any user is able to ADD/DELETE/LIST user profiles for all users.</p> <hr/> <p>Defines when the CA7 interface program (U7SVC) is called.</p> <p>Three values are defined:</p> <ul style="list-style-type: none"> • YES: U7SVC is called for tasks only when a file is created. • ALL: U7SVC is called for tasks when a file is created or replaced. • NO: U7SVC cannot be called. <p>This parameter contains two subparameters. The first subparameter defines the conditions that U7SVC is called for initiator requests, while the second subparameter defines the conditions that U7SVC is called for responder tasks. The default value is NO,NO.</p> <p>For example, CA7=YES,ALL indicates that U7SVC is called when an initiator creates a file and when a responder creates or replaces a file.</p>
CCC_ADMIN_FACILITY	<p>Defines the facility name that is checked to determine if a Command Center user has administrative capabilities.</p> <p>Valid values are a 32-byte facility name and \$CCC.ADMIN. This must</p>

Parameter	Description
	<p>be an RACF or ACF2 or Top Secret facility name. If the facility name is not defined, then no user has Command Center administrative capabilities.</p> <p>The Command Center administrative capabilities include:</p> <ul style="list-style-type: none"> • List or update node definitions. • List or update profile definitions. • Execute Platform Server transfers. • Browse Platform Server completed transfers. • Alter Platform Server transfers waiting to run.
CCC_ALTER_FACILITY	<p>Defines the facility name that is checked to determine if a Command Center user has authorization to alter transfers.</p> <p>Valid values are a 32-byte facility name and \$CCC.ALTER. This must be an RACF or ACF2 or Top Secret facility name. If the facility name is not defined, then users are not authorized to alter transfers.</p>
CCC_BROWSE_FACILITY	<p>Defines the facility name that is checked to determine if a Command Center user has authorization to view transfers.</p> <p>Valid values are a 32-byte facility name and \$CCC.BROWSE. This must be a RACF or ACF2 or Top Secret facility name.</p> <p>If the facility name is not defined, then all validated users are authorized to view transfers. If this facility is not defined to RACF or ACF2 or Top Secret, then all validated users are authorized to view transfers. If a validated Command Center user is not authorized for this facility, then that user can only view transfers initiated by his own user ID.</p>
CCC_TRANSFER_FACILITY	<p>Defines the facility name that is checked to determine if a Command Center user has authorization to initiate transfers.</p> <p>Valid values are a 32-byte facility name and \$CCC.TRANSFER. This must be a RACF or ACF2 or Top Secret facility name.</p> <p>If the facility name is not defined, then all validated users are</p>

Parameter	Description
CONFIG	<p>authorized to initiate transfers. If this facility is not defined to RACF or ACF2 or Top Secret, then all validated users are authorized to initiate transfers.</p> <hr/> <p>Defines the CONFIG DD member that specifies the node members to activate.</p> <p>Valid values are a member name and CONFIG. The default member name is CONFIG. For more information, see Defining Remote Systems in Configuration Library.</p>
CRCCHECK	<p>Defines whether CRC checking is turned on by default. CRC checking is performed against the data that is sent over the network to ensure that the data has not been corrupted. The CRC is not performed against the file itself. ASCII to EBCDIC conversion, translation, and LF/CRLF change the contents of the file between the sender and receiver so the CRC is not performed against the file contents.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • YES: Performs CRC checking by default. • NO: Does not perform CRC checking by default. NO is the default value. <p>The CRCCHECK parameter on the Node and batch parameters override the Global CRCCHECK settings.</p> <div style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p>Note: If the partner Platform Server does not support CRC checking, CRC will still be computed but will not be checked against the partner's CRC value computed.</p> </div>
DATACLASS_DEFAULT_VOLCT	<p>Defines the volume count that is set when a new dataset is created and the DATACLASS parameter is specified.</p> <p>Valid values are any number from 0 to 255. The default value 0 indicates that Platform Server does not set the MVS volume count and therefore uses the volume count defined by the DATACLASS.</p>

Parameter	Description
DEFAULT_COSNAME	<p>When a non-zero value is defined, the Platform Server sets the volume count when the following prerequisites are met:</p> <ul style="list-style-type: none"> • An MVS file is being created. • The DATACLASS parameter is defined. • The GLOBAL DATACLASS_DEFAULT_VOLCT parameter is set between 1 and 255. <p>Defines the default COS name that is used for all file transfers.</p> <p>This name must match a COS entry that is enabled at startup or through the ENABLE operator command.</p> <p>This parameter has no default value. As such, if this parameter is not entered, no default COS can be used. Therefore, if you want to use the Platform Server COS facility, you must define the node DEFAULT_COSNAME or the batch COSNAME parameter.</p>
DEFAULT_RECEIVE_DISP	<p>Defines the action to be taken when a RECEIVE request is executed, and the RECEIVE DISP parameter is not specified.</p> <p>This parameter is also known as EFFECT.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • CREATE_REPLACE: This is the default value. If the file exists on the system, replaces the file. If the file does not exist, creates the file. • CREATE: If the file does not exist, creates the file. If the file exists on the system, displays an error message and terminates the transfer request with errors. • REPLACE: If the file exists on the system, replaces the file. If the file does not exist, displays an error message and terminates the transfer request with errors. • APPEND: If the file exists on the system, appends to the end of the file. If the file does not exist, displays an error message and terminates the transfer request with errors.

Parameter	Description
DEFAULT_TRY_COUNT	<ul style="list-style-type: none"> • CREATE_APPEND: If the file exists on the system, appends to the end of the file. If the file does not exist, creates the file. <p>Defines the total number of times that a file transfer can be tried when temporary errors occur.</p> <p>A request that ends with a network or temporary file error can be retried. This parameter is only used when the RETRY/TRY count defined by the batch interface is set to 0.</p> <p>The default setting is to try a request 7 times before terminating the transfer with a permanent error. If the try count is exceeded for transfers that receive recoverable file errors, transfers are terminated regardless of the FILE_ERROR_TRY_COUNT setting.</p> <p>Valid values are from 1 to 9999.</p>
DNI_INTERVAL	<p>Defines the interval in minutes that the Platform Server waits before scanning the catalog for new datasets.</p> <p>The default value 1 indicates the Platform Server waits for one minute between scanning the catalog for new datasets. Valid values are from 1 to 1440.</p>
DNI_USERID	<p>Defines whether DNI requests are run under the ACEE of the started task or under another user ID.</p> <p>By using the default value of STC, all DNI functions are run under the ACEE of the started task.</p> <p>By specifying a valid z/OS user ID, all DNI functions are run under this user ID's authorization, including scanning the catalog for requests, processing the request, and performing postprocessing actions on the file.</p>
EMAIL_FAIL	<p>Defines either a single email address or multiple email addresses separated by a semicolon (;) to send a notification message if a Platform Server file transfer request is unsuccessful.</p> <p>The maximum field length is 64 characters including any</p>

Parameter	Description
EMAIL_GOOD	<p>semicolons. As an alternative, an email distribution list defined in the organization email system can be used to specify multiple email addresses to receive this notification.</p> <p>This parameter has no default value, which means that no email notification takes place when a request fails.</p> <p>Note: EMAIL_NOTIFY must be selected in the NODE definition as something other than NONE if EMAIL_FAIL or EMAIL_GOOD is selected in the NODE definition.</p>
EMAIL_GOOD	<p>Defines either a single email address or multiple email addresses separated by a semicolon (;) to send a notification message if a Platform Server file transfer request is successful.</p> <p>The maximum field length is 64 characters including any semicolons. As an alternative, an email distribution list defined in the organization's email system can be used to specify multiple email addresses to receive this notification.</p> <p>This parameter has no default for this parameter, which means that no email notification takes place when a request is successful.</p> <p>Note: EMAIL_NOTIFY must be selected in the NODE definition as something other than NONE if EMAIL_FAIL or EMAIL_GOOD is selected in the NODE definition.</p>
EMAIL_NOTIFY	<p>Defines when the Platform Server sends email notifications.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • NONE: No email notification is sent. This is the default value. • INITIATOR: Email notifications are sent on initiator transfers. • RESPONDER: Email notifications are sent on responder transfers. • BOTH: Email notifications are sent on initiator and responder transfers.

Parameter	Description
EMAILSUBFAIL	<p data-bbox="578 310 1354 380">Note: GLOBAL EMAIL_GOOD and EMAIL_FAIL are also required before notifications take place.</p> <p data-bbox="558 443 1328 512">Defines the email subject that is used for unsuccessful email notifications.</p> <p data-bbox="558 548 1360 657">You can enter 1 to 58 characters in this field. In addition to the fixed information, tokens can be defined to add dynamic information to the subject.</p> <p data-bbox="558 688 1357 758">See the following list of the tokens in the long and short forms that are supported:</p> <ul data-bbox="607 789 1409 1360" style="list-style-type: none"> • #(STATUS) or #(ST): Failure or success • #(NODE) or #(ND): 1-byte to 64-byte node name, IP address, or IP name • #(SERVER) or #(STC): 1-character to 8-character started task name • #(PROCESS) or #(PR): 1-character to 8-character transfer process name • #(TRANSID) or #(TID): 10-character transfer transaction ID • #(EASTERTIDE) or #(UD): 1-character to 25-character transfer description • #(LFILE) or #(LF): 1-character to 255-character transfer local file name <p data-bbox="558 1392 1414 1461">The maximum length of the subject field after token substitution is 230 bytes.</p> <p data-bbox="558 1493 1414 1562">The default value is Transfer Request #(ST) Activity Number #(TR).</p> <p data-bbox="578 1612 1365 1682">Note: Columns 1 to 71 are read. Do not put any comments at the end of this parameter.</p>
EMAILSUBGOOD	Defines the email subject that is used for successful email

Parameter	Description
	<p>notifications.</p> <p>You can enter 1 to 58 characters in this field. In addition to the fixed information, tokens can be defined to add dynamic information to the subject.</p> <p>See the following list of the tokens that are supported in the long and short forms:</p> <ul style="list-style-type: none"> • #(STATUS) or #(ST): Failure or success • #(NODE) or #(ND): 1-byte to 64-byte node name, IP address, or IP name • #(SERVER) or #(STC): 1-character to 8-character started task name • #(PROCESS) or #(PR): 1-character to 8-character transfer process name • #(TRANSID) or #(TID): 10-character transfer transaction ID • #(EASTERTIDE) or #(UD): 1-character to 25-character transfer description • #(LFILE) or #(LF): 1-character to 255-character transfer local file name <p>The maximum length of the subject field after token substitution is 230 bytes.</p> <p>The default value is Transfer Request #(ST) Activity Number #(TR).</p> <div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p>Note: Columns 1 to 71 are read. Do not put any comments at the end of this parameter.</p> </div>
ENCRYPT	<p>Defines the level of encryption that is used by default in your system.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • NONE: No encryption. This is the default value.

Parameter	Description
	<ul style="list-style-type: none"> • AES: AES 256-bit encryption • AES128: AES 128-bit encryption • DES: DES encryption • 3DES: Triple DES encryption • BF: Blowfish 56-bit encryption • BLOWFISH: Blowfish 56-bit encryption • BFL: Blowfish 448-bit encryption • BLOWFISH_LONG: Blowfish 448-bit encryption <p>Note: RJ and RIJNDAEL are accepted, but the compatible AES encryption is used instead.</p> <p>If PASSONLY operand is used, then only the password is encrypted using the encryption algorithm specified. The data is not encrypted. PASSONLY is valid only for z/OS to z/OS transfers.</p>
ENFORCE_SECURITY_POLICY	<p>Defines which security policy is enforced.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • FIPS140: Uses FIPS 140 security policy. • TLSFIPS: Uses FIPS 140 protocols and ciphers to execute all SSL/TLS connections, but does not require the entire STC to execute in FIPS 140 mode. • HIPAA: Uses HIPAA security policy. This requires that all files are transferred using encryption, and that encryption must be 128 bits or greater. This checking takes place for both initiator and responder requests. • NO: Uses no security policy. This is the default value.
ENQ_ENTIRE_TRANSFER	<p>Defines whether the entire transfer is enqueued, including Postprocessing Actions (PPA). Valid values are:</p> <p>NO: Uses SYSDSN ENQ to limit access to files while the files are</p>

Parameter	Description
ERRDESC	<p>allocated. This is the default value.</p> <p>YES: Uses ENQ for the file being transferred for the life of the transfer, including Postprocessing Actions.</p>
ERRDESC	<p>Defines the default descriptor code to use for Platform Server error messages.</p> <p>The default value is 4. You can specify as many as you want. It is a good practice that you leave this parameter as default.</p>
ERRROUTE	<p>Defines the default console route to use for Platform Server error messages.</p> <p>The default value is 15. You can specify as many as you want. It is a good practice that you leave this parameter as default.</p>
EXECPOSTPROC	<p>Defines whether directory transfer postprocessing actions are executed on the parent transfer or the child transfers.</p> <p>Valid values are:</p> <p>PARENT: Executes postprocessing on parent transfer.</p> <p>CHILD: Executes postprocessing on child transfers.</p> <p>Default: CHILD</p>
EXECPREPROC	<p>Defines whether directory transfer preprocessing actions are executed on the parent transfer or the child transfers.</p> <p>Valid values are:</p> <p>PARENT: Executes preprocessing on parent transfer.</p> <p>CHILD: Executes preprocessing on child transfers.</p> <p>Default: CHILD</p>
EXIT_MIGRATE_VOLUME	<p>Defines the 6 character VOLSER that informs the Platform Server when a dataset is migrated.</p> <p>This parameter is used when a storage manager other than HSM is</p>

Parameter	Description
EXIT_MIGRATE_WAIT_TIME	<p>used in an installation. This parameter has no default value.</p> <p>When detects this VOLSER, the Platform Server calls EXIT03 to perform the dataset recall. When this parameter is specified, EXIT03 must also be specified as YES, and program FUSEX03 must be assembled and link-edited to the Platform Server STEPLIB.</p>
EXIT00	<p>Defines whether or not user exit is driven at the end of file transfer.</p> <p>Valid values are NO and YES. The default value is NO. With this exit, you can perform an action based on the result of a file transfer. This exit can be invoked on transfer completion (successful or unsuccessful) or on temporary errors, such as network errors, and also for each transfer retry.</p> <p>For more information on how to code a user exit, see Appendix F. User Exits.</p> <div data-bbox="558 1314 1398 1455" style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p>Note: The module name of the user exit is FUSEX00 and must be included in the load library at Platform Server startup if YES is specified for this parameter.</p> </div>
EXIT01	<p>Defines additional security validation to ensure that the user is authorized to use Platform Server or to access the dataset in question.</p> <p>Valid values are NO and YES. The default value is NO. For more information, see Appendix F. User Exits.</p>

Parameter	Description
EXIT03	<p>Defines if the storage manager user exit can be invoked.</p> <p>To invoke this exit properly, the EXIT_MIGRATE_VOLUME must be set to the volume used by the storage management system. Valid values are NO and YES. The default value is NO. For more information, see Appendix F. User Exits.</p>
EXIT04	<p>Defines whether the preallocation user exit FUSEX04 can be called.</p> <p>Valid values are NO and YES. The default value is NO. For more information, see Appendix F. User Exits.</p>
EXIT05	<p>Defines whether or not the purge-transfer-from-queue user exit is driven.</p> <p>With this exit, you can perform an action based on the result of a file transfer. This exit is invoked when a transfer request is completed successfully or unsuccessfully and the transfer is purged from the work queue. The exit is not invoked on temporary errors, such as network errors, until the batch interface RETRY parameter is reached and the transfer is purged. Valid values are NO and YES. The default value is NO.</p> <p>For more information, see Appendix F. User Exits.</p> <div data-bbox="558 1192 1411 1333" style="background-color: #f0f0f0; padding: 10px;"> <p>Note: If YES is specified for this parameter, the module name of the user exit is FUSEX05 and must be included in the load library at Platform Server startup.</p> </div>
EXIT06	<p>Defines whether the add transfer to queue user exit is driven.</p> <p>With this exit, you can check the specified transfer parameters, override parameters specified, fail the transfer or return with no changes. This exit is invoked when a transfer request is queued.</p> <p>Valid values are NO and YES. The default value is NO. For more information, see the Appendix F. User Exits.</p>

Parameter	Description
EXPIRE	<p data-bbox="578 310 1393 415">Note: If YES is specified for this parameter, the module name of the user exit is FUSEX06 and must be included in the load library at Platform Server startup.</p> <p data-bbox="558 480 1403 590">Defines the default number of days that an unfinished file transfer remains on the work queue before being purged by the Platform Server.</p> <p data-bbox="558 625 1378 695">It is a good practice that you leave this parameter to the default value of 30 days.</p>
EXTENDED_SECURITY_CHECK	<p data-bbox="558 743 1386 812">Defines whether extended resource checking is performed for file transfer requests.</p> <p data-bbox="578 858 1341 890">Note: This checking is only performed for initiator requests.</p> <p data-bbox="558 942 1396 1012">This parameter contains two sub-parameters. The default value is NO,NO.</p> <p data-bbox="558 1045 1375 1155">The first subparameter defines whether extended resource checking is performed to check if a user is authorized to use the Platform Server.</p> <p data-bbox="558 1188 1349 1297">The second subparameter defines whether extended resource checking is performed to check if a user is authorized to use a node within the Platform Server.</p> <p data-bbox="558 1304 1084 1335">For each sub-parameter, valid values are:</p> <ul data-bbox="607 1362 1398 1707" style="list-style-type: none"> <li data-bbox="607 1362 1357 1472">• ENFORCE: Performs extended resource checking. If the extended security checking fails, displays a message and terminates the request with errors. <li data-bbox="607 1499 1398 1650">• WARN: Performs extended resource checking. If the extended security checking fails, displays a message, but the request can continue. This is useful in a migration phase where you do not want to terminate transfer requests. <li data-bbox="607 1677 1276 1707">• NO: Does not perform extended resource checking. <p data-bbox="558 1734 1338 1766">For more information, see Define Platform Server to the z/OS</p>

Parameter	Description
EXTENDED_SECURITY_RESOURCE	<p data-bbox="558 296 769 321">Security System.</p> <p data-bbox="558 373 1295 445">Defines the resource that is used when extended resource checking is turned on.</p> <p data-bbox="558 476 1304 548">The value defined by this parameter is a prefix to the fixed component of the extended resource check.</p> <p data-bbox="558 579 1338 646">For more information, see Define Platform Server to the z/OS Security System.</p>
FILE_ERROR_TRY_COUNT	<p data-bbox="558 695 1386 766">Defines the maximum number of times that a file transfer is tried when temporary file errors occur.</p> <p data-bbox="558 798 1386 869">Many file transfer requests are terminated because of retry-able file errors. Valid values are from 1 to 9999. The default value is 5.</p> <p data-bbox="558 900 1414 1087">FILE_ERROR_TRY_COUNT cannot be overridden by the batch interface. Temporary file errors are only retried up to the lower number of the value defined by FILE_ERROR_TRY_COUNT and the try count defined by the transfer or by the GLOBAL_DEFAULT_TRY_COUNT.</p> <p data-bbox="558 1119 1398 1268">If the FILE_ERROR_TRY_COUNT exceeds the try count, file errors are terminated when the try count is reached. The default is to try a request 5 times before terminating the transfer with a permanent error.</p>
FILE_ERROR_TRY_INTERVAL	<p data-bbox="558 1318 1406 1390">Defines the amount of time in seconds that a request waits after a retry-able file error and before the request is retried.</p> <p data-bbox="558 1421 1276 1446">The default value is 60. Valid values are from 10 to 9999.</p> <div data-bbox="558 1478 1414 1619" style="background-color: #f0f0f0; padding: 5px;"> <p data-bbox="578 1497 1393 1602">Note: The actual interval can be up to 60 seconds or more than the value specified, depending on when the error occurs and when the dispatcher runs.</p> </div>
FIND_NODE_COMPATIBILITY	<p data-bbox="558 1669 1409 1778">Defines the mechanism for finding a node definition for incoming requests. In MFT 7.2 and later, the code uses a reverse DNS lookup to match the incoming request to a node definition; prior releases</p>

Parameter	Description
	<p>used a one-time DNS request to match the incoming request to a node definition. Valid values are:</p> <ul style="list-style-type: none"> • Yes: Uses both methods to match an incoming request to a node definition. Both the reverse DNS lookup and the DNS lookup are used. • No: Uses only the reverse DNS look will be used to match an incoming request to a node definition. This is the default value.
HOLD_TEMPORARY_FILE_ERRORS	<p>Defines what happens when a locally initiated file transfer is completed with a temporary file error, and the Platform Server retries until the GLOBAL FILE_ERROR_TRY_COUNT is exceeded.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • NO: Purges the transfer from the active queue. This is the default value. • YES: Puts the transfer on the hold queue. The operator can then release the transfer when the condition that caused the error is resolved.
HOLD_TEMPORARY_NETWORK_ERRORS	<p>This parameter contains two sub-parameters.</p> <p>The first sub-parameter defines what happens when a locally initiated file transfer is completed with a temporary file or network error, and the Platform Server retries until the batch try count or GLOBAL DEFAULT_TRY_COUNT is exceeded.</p> <p>For the first sub-parameter, valid values are:</p> <ul style="list-style-type: none"> • NO: Purges the transfer from the active queue. This is the default value. • YES: Puts the transfer on the hold queue. The operator can then release the transfer when the condition that caused the error is resolved. <p>The second sub-parameter defines if the file can be released automatically.</p>

Parameter	Description
HSM_MIGRATE_VOLUME	<p>For the second sub-parameter, valid values are:</p> <ul style="list-style-type: none"> • NORELEASE: The transfer cannot be released automatically. The transfer must be released by a Platform Server operator. • RELEASE: The Platform Server can automatically release the file transfer request. <p>The transfer request can be released in the following two ways:</p> <ul style="list-style-type: none"> • The operator can release the transfer. • The transfer can also be released if another transfer request is successfully performed with the same node. The request that causes the transfer to be released can be initiated locally or remotely. <p>Defines the 6 character volser that informs the Platform Server when a dataset is migrated.</p> <p>The default value is MIGRAT. When the Platform Server detects a migrated dataset, the dataset is recalled before the transfer is started.</p>
IGNORE_EBCDIC_LF	<p>Defines whether EBCDIC line feed (0x25) and new line (0x15) characters are checked when the transfer DELIMITER is set to LF or CRLF. ASCII line feeds (0x0A) are always checked. The default value of NO indicates that both EBCDIC and ASCII line feeds are checked. When set to YES, only ASCII line feeds are checked.</p> <ul style="list-style-type: none"> • When CRLF is set to LF, the following delimiters are checked: 0x0a, 0x15, 0x25 • When CRLF is set to CRLF, the following delimiters are checked: 0x0d0a, 0x0d15, 0x0d25 <p>Note: This parameter is only used when z/OS is receiving a file and the delimiter is set to LF or CRLF.</p>
INFODESC	<p>Defines the default descriptor code for information messages from the Platform Server.</p>

Parameter	Description
INFORROUTE	<p>You can specify as many descriptor codes as you want. Separate descriptor codes by commas. It is a good practice that you leave this parameter as the default value 4.</p> <p>Defines the default console route code for information messages from the Platform Server.</p> <p>You can specify as many route codes as you want. Separate route codes by commas. It is a good practice that you leave this parameter as the default value 15.</p>
INITIATORSTOP	<p>Defines whether the user can start the Platform Server in a quiesced mode.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • NO: Initiator tasks run normally when the Platform Server is started. This is the default value. • YES: The Platform Server can be started, but cannot initiate any file transfer requests. You can use the INITIATORSTOP operator command to stop initiator requests while the Platform Server is running, and use the INITIATORSTART operator command to start initiator tasks again.
JOB_SUBMIT_DSN	<p>Defines the PDS dataset name used for the SUBMIT operand.</p> <p>When you use TYPE=COMMAND, SUBMIT=<i>membername</i> operand of the Platform Server batch interface to submit a job, if the JCL is located on the remote system, you can define the JCL to be submitted through one of the following two ways:</p> <ul style="list-style-type: none"> • Specify the entire dataset name in the SUBMIT operand. • Specify just a member in the SUBMIT operand. <p>When just a member name is specified in the SUBMIT operand, the Platform Server uses the dataset name defined in the GLOBAL JOB_SUBMIT_DSN parameter.</p>

Parameter	Description
MANAGE_INTERFACE_PROTOCOL	<p data-bbox="578 310 1360 415">Note: This field must be from 1 to 44 characters and must specify a cataloged PDS dataset. Member names must not be specified on this parameter.</p> <p data-bbox="558 478 1333 510">Defines the protocols that are accepted to manage transfers.</p> <p data-bbox="558 541 1352 695">Platform server transfers can be managed in three ways: cross memory, SNA communication and TCP communication. Cross memory requests are always accepted. Only TCP and SNA protocols can be restricted.</p> <p data-bbox="558 726 764 758">Valid values are:</p> <ul data-bbox="607 789 1393 1031" style="list-style-type: none"> • ALL: TCP, SNA and cross memory requests are all accepted. This is the default value. • NONE: Only cross memory requests are accepted. • TCP: TCP and cross memory requests are accepted. • SNA: SNA and cross memory requests are accepted. <p data-bbox="578 1077 1357 1213">Note: This parameter describes how transfers can be managed and does not regulate how transfers actually execute. Only the following programs that manage transfers are referred to:</p> <ul data-bbox="626 1241 1300 1318" style="list-style-type: none"> • Batch: OSIUB000 and OSIUC000 OPERATE function • ISPF: REXX and ISPF panels
MAX_AUDIT_RECORDS	<p data-bbox="558 1388 1357 1461">Defines the number of audit records that can be written to the Platform Server audit file.</p> <p data-bbox="558 1493 1398 1640">When that number is exceeded or the file runs out of space, the Platform Server begins rewriting over the oldest records in the audit files. The default value is 25,000 and the valid range is from 100 to 99,999,999.</p>

Parameter	Description
MAXINITS	<p data-bbox="578 310 1393 445">Note: This parameter is dependent on the size allocated for the audit VSAM dataset. The <code>MAX_AUDIT_RECORDS</code> must not be greater than the maximum number of records that can fit into the audit VSAM dataset.</p> <p data-bbox="568 478 1325 506">For more information, see Defining the Audit VSAM dataset.</p> <p data-bbox="558 554 1357 625">Defines the maximum number of initiator requests that can be active on this system at any one time.</p> <p data-bbox="558 659 1333 686">The default value is 100. The valid range is from 1 to 99,999.</p>
MAXRESP	<p data-bbox="558 737 1377 808">Defines the number of responder transfers that can be active on this system at any one time.</p> <p data-bbox="558 842 1317 869">The default value is 70. The valid range is from 1 to 99,999.</p>
MAXSCRIPTS	<p data-bbox="558 919 1409 991">Defines the maximum number of script requests that can be active on this system at any one time.</p> <p data-bbox="558 1024 1386 1167">The default value is 50. The valid range is from 1 to 99,999. This parameter is used to limit the number of concurrent scripts that can be executed in the started task. This value must be less than or equal to the <code>MAXINITS</code> value.</p>
MAXTOTAL	<p data-bbox="558 1220 1393 1291">Defines the number of transfers in total that can be active on this system at any one time.</p> <p data-bbox="558 1325 1386 1388">The default value is 100. The valid range is from 1 to 99,999. The default value 100 is adequate for most users.</p>
MIXED_CASE_PASSWORDS	<p data-bbox="558 1440 1386 1512">Defines whether password validation uses mixed case passwords or translates all passwords to upper case.</p> <p data-bbox="558 1545 764 1572">Valid values are:</p> <ul data-bbox="610 1606 1333 1724" style="list-style-type: none"> <li data-bbox="610 1606 1260 1633">• Yes: Passwords are not translated to upper case. <li data-bbox="610 1667 1333 1724">• No: Passwords are translated to upper case. This is the default value.

Parameter	Description
MODENAME	<p>Defines the SNA mode name.</p> <p>It is a good practice that you leave it as the default value. If you do not use the default value, you must define a VTAM mode entry. Valid values must be from 1 to 8 characters and must correspond to a VTAM MODEENT. The default value is #BATCH.</p> <p>You can refer to the Platform Server sample library for an example.</p>
MQ_FAIL	<p>Defines the message queue where the Platform Server sends a notification message when a file transfer request is completed unsuccessfully.</p> <p>Valid values are <code>\$MQ:queue_manager_name:message_queue_name</code>.</p> <p><code>\$MQ</code> is a literal and must be the first three characters. The message queue name must be from 1 to 55 characters. This parameter has no default value, which means that no MQ notification takes place when a request fails.</p>
MQ_GOOD	<p>Defines the message queue where the Platform Server sends a notification message when a file transfer request completes successfully.</p> <p>Valid values are <code>\$MQ:queue_manager_name:message_queue_name</code>.</p> <p><code>\$MQ</code> is literal and must be the first three characters. The message queue name must be from 1 to 55 characters. This parameter has no default value, which means that no MQ notification takes place when a request is successful.</p>
MQ_NOTIFY	<p>Defines when the Platform Server sends MQ notifications.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • NONE: No MQ notification is sent. This is the default value. • INITIATOR: MQ notification is sent for initiator tasks. • RESPONDER: MQ notification is sent for responder tasks. • BOTH: MQ notification is sent for both initiator and responder tasks.

Parameter	Description
MQ_WAIT_MSEC	<p data-bbox="578 310 1321 380">Note: The GLOBAL MQ_GOOD and MQ_FAIL are also required before notifications take place.</p> <p data-bbox="558 443 1370 512">Defines the interval in milliseconds that MQ waits for additional data before returning EOF.</p> <p data-bbox="558 548 1406 617">The default value is 5000. This means MQ waits for additional data for 5 seconds before returning EOF.</p> <p data-bbox="558 653 1370 758">Valid values are from 0 to 99999999. A value of 0 tells MQ not to wait for additional data and return EOF after reading the last record.</p>
OUTC	<p data-bbox="558 806 1317 875">Defines the output class to which the Platform Server log is written.</p> <p data-bbox="558 911 1406 1058">Use a held SYSOUT class so the log can be browsed online. Valid values are from A to Z, or 0 to 9. The default value is blank, which implies that the SYSOUT log goes to the system wide default SYSOUT class.</p>
OUTDEST	<p data-bbox="558 1106 1360 1134">Defines the JES destination of the Platform Server SYSOUT log.</p> <p data-bbox="558 1169 1406 1199">Valid values are from 1 to 8 characters. The default value is LOCAL.</p>
PDS_SPACE_RELEASE	<p data-bbox="558 1247 1406 1316">Defines whether unused disk space is automatically released when a PDS file is created.</p> <p data-bbox="558 1352 768 1379">Valid values are:</p> <ul data-bbox="607 1415 1406 1577" style="list-style-type: none"> <li data-bbox="607 1415 1406 1484">• YES: Releases unused disk space when a PDS file is created. This is the default value. <li data-bbox="607 1512 1406 1577">• NO: Does not release unused disk space when a PDS file is created. <p data-bbox="578 1625 1370 1730">Note: This parameter applies only to PDS and PDSE files. It is ignored when a user specifically tells the Platform Server to release or not release space for a particular file.</p>

Parameter	Description
PDS_WRITE_DISP	<p>Defines how the Platform Server allocates PDS files when the Platform Server is updating or writing a PDS file.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • OLD: The Platform Server requires exclusive access (DISP=OLD) to the dataset. This is the default value. • SHR: The Platform Server requires shared access (DISP=SHR) to the dataset.
PRIMARY_SPACE	<p>Defines the number of cylinders or tracks to allocate for the primary allocation when creating a new z/OS dataset if the Platform Server is acting as a responder.</p> <p>When you first install the product, you can leave this parameter to the default value 3.</p> <div data-bbox="558 911 1406 1136" style="background-color: #f0f0f0; padding: 10px;"> <p>Note: The Platform Server allocates new datasets with the release (RLSE) option to avoid wasting space. If that is not acceptable, then you can preallocate the file. This parameter is only used when the initiator of the transfer is not running TIBCO MFT Platform Server for z/OS and has not specified the allocation information.</p> </div>
REMOVE_TRAILING_CR	<p>Defines whether to remove the CR delimiter when z/OS is receiving a file with the CRLF=LF parameter and the file being received is CRLF delimited rather than just LF delimited.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • YES: The Platform Server automatically removes the trailing CR delimiter. • NO: The Platform Server does not remove the trailing CR delimiter. This is the default value.
REQUIRE_NODE_DEFINITION	<p>Defines whether node definition must be enabled before a transfer request can take place.</p> <p>This parameter has two subparameters. The first subparameter is</p>

Parameter	Description
	<p>for initiator requests and the second subparameter is for responder requests. The default values are NO,NO.</p> <ul style="list-style-type: none"> • If YES is specified for an initiator request, the IPADDR and IPNAME parameters cannot be used. Requests can be scheduled only through the use of the NODE parameter, and the node specified in the NODE parameter must be enabled. If IPADDR or IPNAME parameters are used or if the node specified on the NODE parameter is not enabled, the request is terminated with an error. • If YES is specified for a responder request, the Platform Server checks the node definitions when a request is received. If the LUNAME or IP address from which the request is received is defined on an enabled node definition, the request is accepted. Otherwise, the request is rejected with an error.
RESETINTERVAL	<p>Defines the time in minutes before the Platform Server reset all retry-able errors and retry file transfers.</p> <p>Valid values are 1 to 60. The default value is 7.</p>
RESPONDER_PROFILE	<p>Defines if the system requires responder profiles when a request comes in.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • YES: Requires responder profiles. This can only be overridden by a node definition that specifies the RESPONDER_PROFILE parameter. • NO: does not require responder profiles. This is the default value. • DUAL: The responder profiles are checked. If no matches are found in the responder profile, the request can continue to be processed, and the user ID and password are checked against the RACF or ACF2 or Top Secret security system. <p>For more information, see User Profiles.</p>

Parameter	Description
RESPONDER_PROFILE_LPASS	<p>Defines whether a local password is required when defining a responder profile for a local user that is different than the user initiating the request.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • YES: Local password is required to create a responder profile when the user initiating the request is different than the user defined by the LUSER parameter. • NO: Local password is not required to create a responder profile when the user initiating the request is different than the user defined by the LUSER parameter. This is the default value.
RESPONDERSTOP	<p>Defines whether a user can stop responder requests while the Platform Server is running.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • YES: The Platform Server can be started, but cannot accept any responder file transfer requests. You can use the RESPONDERSTART operator command to start responder tasks again. • NO: Responder tasks run normally when the Platform Server is started. This is the default value.
RPROFILE_MIN_LENGTH	<p>Defines the minimum length of the responder profile password.</p> <p>Valid values are 4 - 10. The default value is 6.</p>
RPROFILE_MIN_LETTERS	<p>Defines the minimum number of letters.</p> <div data-bbox="558 1486 1409 1665" style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p>Note: When computing the minimum number of letters, the upper and lower case versions of the same character are considered the same character. So a password of AaBb is computed as having two letters.</p> </div> <p>Valid values are 3 - 8. The default value is 3.</p>

Parameter	Description
RPROFILE_MIN_NUMBER	<p>Defines the minimum number of numbers.</p> <p>Valid values are 0 - 8. The default value is 1.</p>
RPROFILE_MIN_SPECIAL	<p>Defines the minimum number of special characters. The valid special characters are !@#\$%() ; :&<></p> <p>Valid values are 0 - 8. The default value is 1.</p>
RPROFILE_MIN_UNIQUE	<p>Defines the minimum number of unique characters.</p> <p>Valid values are 3 - 8. The default value is 4.</p>
RPROFILE_PASSWORD_VALIDATION	<p>Defines whether the responder profile password validation is performed.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • YES: Validates the responder profile password strength. • NO: Does not validate the responder profile passwords. <p>Default: NO</p>
RPROFILE_REQUIRE_UPPER_LOWER	<p>Defines whether the upper and lower case characters are required:</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • YES: Requires upper and lower case characters. The default value is YES. • NO: Upper and lower case characters are not required.
RSHOST	<p>Defines the IP address or host name of the RocketStream server.</p> <p>By defining a host on the command line or in a transfer template, you can override the RSHost value configured in the config.txt.</p> <p>When this parameter is not defined, if you have RSAccelerator set to Yes, the value configured for RSHost in the config.txt is used.</p>
RSPORT	<p>Defines the port number the RocketStream server is listening on for transfers using the RocketStream technology.</p>

Parameter	Description
SAPI_INTERFACE	<p>By defining a port number on the command line or in a transfer template, you can override the RSPort value configured in the config.txt.</p> <p>The default value is 9099. When this parameter is not defined, if you have RSAccelerator set to Yes, then the value configured for RSPort in the config.txt is used.</p>
SAPI_INTERFACE	<p>Defines whether the Platform Server Sysout API (SAPI) interface program is activated.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • YES: Activates the SAPI interface. • NO: Does not activate the SAPI interface. This is the default value. <p>The SAPI interface extracts data off the JES spools and transmits it to a remote system either as a file or a report.</p>
SAPI_SUBSYSTEM	<p>Defines the name of the SAPI subsystem.</p> <p>Valid values must be 4 bytes. Using this parameter, you can override the name of the subsystem that is used for all SAPI requests.</p> <p>This parameter is not required because the default value is the name of the primary subsystem, which is either JES2 or JES3. If you must use a secondary subsystem, you can use this parameter to override the subsystem name.</p>
SAPI_USERID	<p>Defines whether SAPI requests are run under that ACEE of the started task or under another user ID.</p> <p>By specifying or using the default value of STC, all SAPI functions run under the ACEE of the started task.</p> <p>By specifying a valid z/OS user ID, all SAPI functions, including scanning the JES queue for output, processing the request, and performing postprocessing actions on output, run under this user ID's authorization.</p>

Parameter	Description
SECONDARY_SPACE	<p>Defines the number of units that are allocated on disk for the secondary space allocation for new datasets created by the Platform Server.</p> <p>The type of units is determined by the SPACE= parameter. The default value is 0.</p>
SEND_EMAIL_ON_CANCEL	<p>Defines whether a failure notification email will be sent when an inactive or held transfer is canceled:</p> <p>Valid values are:</p> <p>YES: Sends an email when an inactive or held transfer is canceled.</p> <p>NO: Does not send an email when an inactive or held transfer is canceled.</p> <p>Default: YES</p>
SENDER_FAILURE	<p>Defines the sender email address that will be used when a Failure Notification email is sent. When defined, this overrides the GLOBAL SMTP_SENDER_HOSTNAME and the hard-coded TRANSFER_FAIL value.</p> <p>There is no default for this parameter.</p>
SENDER_SUCCESS	<p>Defines the sender email address that will be used when a Success Notification email is sent. When defined, this overrides the GLOBAL SMTP_SENDER_HOSTNAME and the hard-coded TRANSFER_GOOD value.</p> <p>There is no default for this parameter.</p>
SMF	<p>Defines whether SMF records are written.</p> <p>Valid values are NO or an SMF record number ranging from 192 to 255.</p> <p>For more information on SMF data format, see <i>TIBCO® Managed File Transfer Platform Server for z/OS User's Guide</i>.</p>

Parameter	Description
SMS_STORAGE_CLASS	<p>Defines the name of the SMS storage class to be used when the Platform Server creates new datasets.</p> <p>The valid value is a 1-character to 8-character storage class name. Do not specify this parameter unless you use SMS and your storage or DASD administrator specifically advises it for incoming files.</p>
SMTP_DEST	<p>Defines the JES destination that is used when an email is sent by the Platform Server.</p> <p>The valid value is an 8 character JES SYSOUT destination. If this parameter is not defined, the email is sent to the default JES destination. Using this parameter, the Platform Server can send the data to an SMTP server running on a different system.</p>
SMTP_JOBNAME	<p>Defines the name of the SMTP started task.</p> <p>This name is required for the Platform Server email feature and it must match the name of the SMTP started task.</p>
SMTP_LOCALCLASS	<p>Defines the SYSOUT class to which the Platform Server writes SMTP requests.</p> <p>This class must match the LOCALCLASS parameter of the SMTP configuration dataset.</p>
SMTP_SENDER_HOSTNAME	<p>Defines the name of the TCP host.</p> <p>When sending an email, the Platform Server uses this name when setting up the SMTP MAIL FROM command. This name shows up as the sender of the email when the email is read. Valid values can be up to 40 bytes of data.</p>
SPACE	<p>Defines the default space allocation in cylinders or tracks.</p> <p>This is the default unit type for space allocations when a new dataset must be created on this system to satisfy a Platform Server activity.</p>

Parameter	Description
SSL_CLIENT_DNLABEL	<p>Defines the label name of the certificate that is used by the SSL client.</p> <p>This certificate is used for the client. If you want to use the default certificate, you must specify this parameter as NULL in upper case.</p> <p>If this parameter is not specified, the SSL client uses the certificate label defined by the SSL_DNLABEL parameter.</p> <p>If you are using the gskkyman utility, the label name is entered when issuing the certificate request. When using the RADCERT command, the label name is specified on the ADD request by the WITHLABEL parameter.</p>
SSL_DNLABEL	<p>Defines the label name of the certificate that is used by the SSL server.</p> <p>This certificate is used for the server. If you want to use the default certificate, you must specify this parameter as NULL in upper case.</p> <p>If the SSL_CLIENT_DNLABEL parameter is not specified, the SSL client also uses this certificate defined for the server.</p> <p>If you are using the gskkyman utility, the label name is entered when issuing the certificate request. When using the RADCERT command, the label name is specified on the ADD request by the WITHLABEL parameter.</p>
SSL_ENCRYPT	<p>Defines the level of encryption that is used by default for SSL requests in your system. This parameter is ignored for TLS Tunnel requests because all data is encrypted within the TLS Tunnel.</p> <p>Valid values are:</p> <ul style="list-style-type: none">• NONE: No encryption• AES: AES 256-bit encryption• AES128: AES 128-bit encryption• DES: DES encryption• 3DES: Triple DES encryption

Parameter	Description
SSL_KEY_DBNAME	<ul style="list-style-type: none"> • BLOWFISH BF: Blowfish 56-bit encryption. This is the default value. • BLOWFISH_LONG BFL: Blowfish 448-bit encryption <p>Note: RJ and RIJNDAEL are also accepted, but the compatible AES encryption is used instead.</p>
SSL_KEY_DBSTASH	<p>Defines the name of the key database or the ring name that contains the certificates used by the Platform Server.</p> <p>If you are using the z/OS shell bases utility <code>gskkyman</code>, then you must define the name of the HFS key database file created by that utility.</p> <p>If you are using the RACF <code>RACDCERT</code> command, then you must define the name of the key ring associated with the Platform Server started task.</p> <p>Defines the name of the HFS key database stash file.</p> <p>If you are not using a stash file, you must omit this parameter or specify this parameter as <code>NULL</code>.</p> <p>If you are using a key ring, you must omit this parameter or specify this parameter as <code>NULL</code>.</p> <p>If possible, do not use the stash file. If you are using the <code>gskkyman</code> utility, it is a good practice that you use the <code>\$\$SSLDB</code> user profile instead.</p>
SSL_NETWORK_IPADDR	<p>Defines the IP address of the local system.</p> <p>The default value is the IP address of the local system. This parameter is used to define whether a request must be an SSL request.</p> <p>The Platform Server takes the IP address of the local system and the IP address of the target system. The Platform Server determines the subnet of these two addresses by using the <code>SSL_NETWORK_SUBNET</code> parameter. The Platform Server then compares</p>

Parameter	Description
SSL_NETWORK_IPADDR_IPV6	<p>the two values to determine if a request is within the subnet or outside the subnet. If inside the subnet, then the request does not have to be an SSL request. If outside the subnet, then the request must be an SSL request.</p>
SSL_NETWORK_IPADDR_IPV6	<p>Defines the IPV6 address of the local system.</p> <p>This parameter is used to decide whether a request must be an SSL request.</p> <p>The Platform Server takes the IPV6 address of the local system and the IP address of the target system, and determines the subnet of these two addresses by using the SSL_NETWORK_SUBNET_IPV6 parameter, and then compares the two values to determine if a request is within the subnet or outside the subnet.</p> <p>If inside the subnet, then the request does not have to be an SSL request. If outside the subnet, then the request must be an SSL request.</p>
SSL_NETWORK_SUBNET	<p>Defines the subnet of the SSL_NETWORK_IPADDRESS that is used to check if a request must use SSL.</p> <p>This is specified in dotted decimal format. Any target system that is outside of the defined subnet requires SSL. This parameter has no default value.</p>
SSL_NETWORK_SUBNET_IPV6	<p>Defines the subnet of the SSL_NETWORK_IPADDR_IPV6 that is used to check if a request must use SSL.</p> <p>You must specify a number ranging from 0 to 128 and divisible by 8. Any target system that is outside of the defined subnet requires SSL. This parameter has no default value.</p>
SSL_REQUEST	<p>Defines when or whether SSL must be used.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • NO: SSL is not required but can be used. This is the default value.

Parameter	Description
SSL_REQUEST_IPV6	<ul style="list-style-type: none"> • YES: SSL must be used for both initiator and responder requests. • OUTSIDE_NETWORK: SSL usage depends on the IP address of the target system and how it compares with the address of the local system. <p>If the target system address is outside the subnet defined by the SSL_NETWORK_IPADDRESS and SSL_NETWORK_SUBNET parameters, the request must use SSL.</p>
SSL_REQUEST_IPV6	<p>Defines when or whether SSL must be used on IPV6 networks.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • NO: SSL is not required but can be used. This is the default value. • YES: SSL must be used for both initiator and responder requests. • OUTSIDE_NETWORK: SSL usage depends on the IPV6 address of the target system and how it compares with the address of the local system. <p>If the target system address is outside the subnet defined by the SSL_NETWORK_IPADDR_IPV6 and SSL_NETWORK_SUBNET_IPV6 parameters, then the request must use SSL.</p>
SSL_VERSION	<p>This parameter is typically not used, because the Platform Server automatically detects the operating system level.</p> <p>Valid values are OS390 and ZOS. This parameter has no default value.</p> <p>Ensure that you only use this parameter under TIBCO Support instructions.</p>
TAPE_SECURITY_EXIT	<p>Defines the processing performed when the SMS tape security exit is invoked for a tape mount that is not expired.</p> <p>Valid values are:</p>

Parameter	Description
	<ul style="list-style-type: none"> • ERROR: Rejects the tape mount. Terminates the transfer with retry-able abend 913-34. • IGNORE: Performs the default SMS action. This is the default value.
TCP_CONNECT_TIMEOUT	<p>Defines the amount of time in seconds that TCP waits for a TCP connect request to be completed before terminating the request with an error.</p> <p>The default is to wait until TCP times out the request. Valid values are from 10 to 180.</p>
TCP_KEEPALIVE	<p>Defines the seconds between keepalive, that is, how periodically a packet is sent on an otherwise idle connection for a stream socket. By default, this option is disabled.</p> <p>Valid values are either 0 or a number from 30 to 7200. The default value is 0 and it means no keepalive.</p>
TCP_TIMEOUT	<p>Defines the interval in minutes after which the Platform Server terminates all TCP read and write requests that are not completed.</p> <p>If you leave this parameter to the default value 30, the Platform Server terminates a request only when TCP detects that a connection is terminated. The maximum value is 1440, which is the number of minutes in one day.</p> <p>You have to set this parameter high enough to ensure that network re-transmissions and tape mounts do not cause a request to fail.</p>
TCPCONNECT_ADAPTER_IPADDR	<p>Defines the IP address of the TCP network interface to which your Platform Server started task binds for outgoing connections.</p> <p>The default value ALL means to bind to all TCP network interfaces. If you want to bind to only a single network interface, specify the dotted decimal address of the network interface, so the Platform Server only binds to that network interface for outgoing requests.</p>

Parameter	Description
TCPCONNECT_ADAPTER_IPADDR_IPV6	<p data-bbox="578 310 1386 380">Note: This parameter is used only for outgoing requests, and is ignored for incoming requests.</p> <p data-bbox="558 443 1341 554">Defines the IP address of the TCP network interface that your Platform Server started task binds to for outgoing IPv6 connections.</p> <p data-bbox="558 590 1409 737">The default value ALL means to bind to all TCP network interfaces. If you want to bind to only a single network interface, specify the IPv6 address of the network interface, so the Platform Server only binds to that network interface for outgoing requests.</p> <p data-bbox="578 783 1386 852">Note: This parameter is used only for outgoing requests, and is ignored for incoming requests.</p>
TCPIPJOBNAME	<p data-bbox="558 915 1341 989">Defines the name of the TCP/IP started task on the local z/OS system.</p> <p data-bbox="558 1020 1138 1052">The job name can contain up to 8 characters.</p>
TCPIPPORT	<p data-bbox="558 1098 1414 1171">Defines the IP port that the Platform Server listens on for incoming transfer requests.</p> <p data-bbox="558 1203 1370 1314">Valid values are either 0 or a number from 1025 to 65535. The default value is 46464. If 0 is specified, then the Platform Server does not listen for incoming requests.</p> <p data-bbox="558 1346 1333 1451">This parameter is also used as the default IP port on a TCP connect request unless the IPPORT on the NODE or PROCESS statements is specified.</p>
TCPIPPORT_IPV6	<p data-bbox="558 1497 1317 1570">Defines the IPv6 port that the Platform Server listens on for incoming transfer requests.</p> <p data-bbox="558 1602 1382 1713">Valid values are either 0 or a number from 1025 to 65535. The default value 0 indicates that the Platform Server does not listen for incoming IPv6 requests.</p>
TCPLISTEN_ADAPTER_	<p data-bbox="558 1759 1341 1791">Defines the IP address of the TCP network interface that your</p>

Parameter	Description
IPADDR	<p>Platform Server started task listens for incoming connections.</p> <p>The default value ALL means to listen to all TCP network interfaces. If you want to listen to only a single network interface, specify the dotted decimal address of the network interface, so the Platform Server only listen to that network interface for incoming requests.</p> <p>Note: This parameter is used only for incoming requests, and is ignored for outgoing requests.</p>
TCPLISTEN_ADAPTER_IPADDR_IPV6	<p>Defines the IP address of the TCP network interface that your Platform Server started task listens for incoming IPV6 connections.</p> <p>The default value ALL means to listen to all TCP network interfaces. If you want to listen to only a single network interface, specify the IPV6 address of the network interface, so the Platform Server then only listen to that network interface for incoming requests.</p> <p>This parameter is used only for incoming IPV6 requests, and it is ignored for outgoing requests.</p>
TEXTEOFTERM	<p>Defines whether the Platform Server for z/OS adds a LF or CRLF delimiter to the last record of a text transfer to a UNIX or Windows server.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • NO: Does not append LF or CRLF to the last record. This is the default value. • YES: Appends LF or CRLF to the last record. <p>Note: This parameter can be overridden by the batch TEXTEOFTERM parameter.</p>
TLSCIPHERS	<p>Defines the TLS ciphers that are supported by MFT. The ciphers must be defined as 4 alphanumeric digits. The ciphers are</p>

Parameter	Description
	<p>documented in Appendix C of the IBM manual: <i>z/OS Cryptographic Services System Secure Sockets Layer Programming</i>. If not defined, MFT uses the default SSL ciphers.</p> <p>If FIPS140 is specified, only FIPS approved ciphers are used.</p> <p>Ciphers that meet the following criteria are specified in the sample GLOBAL member:</p> <ul style="list-style-type: none"> • FIPS approved • AES256 • SHA or higher message digest <p>Multiple TLSCIPHERS parameters can be defined. One TLS cipher can be defined for each TLSCIPHERS parameter. The text after the 4 alphanumeric digits is used for documentation only and is ignored.</p>
<p>TLSENABLEDPROTOCOLS</p>	<p>Defines the TLS protocols that are supported when running in SSL Mode. Multiple TLS parameters can be entered separated by a comma.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • TLSV1: TLSV1 is supported. • TLSV1_1: TLSV1_1 is supported. • TLSV1_2: TLSV1_2 is supported. • ALL <p>Note: SSLV2 and SSLV3 are not supported.</p> <p>Example: TLSENABLEDPROTOCOLS=TLSV1_1,TLSV1_2</p> <p>If this parameter is not entered, the default is ALL.</p>
<p>TLSIPPORT (SSLIPPORT)</p>	<p>Defines the IP port that the Platform Server listens on for TLS requests.</p> <p>If a non-TLS request is received on this IP port, then an error message is sent to the initiator and the request is terminated.</p>

Parameter	Description
TLSIIPORT_IPV6 (SSLIIPORT_IPV6)	<p>This field must be different than the IIPORT parameter, and unique on the z/OS system. This parameter has no default value. If this parameter is not defined, then TLS processing is disabled.</p> <hr/> <p>Defines the IPV6 port that the Platform Server listens on for TLS requests.</p> <p>Valid values are either 0 or a number from 1025 to 65535. The default value 0 indicates that the Platform Server does not listen for incoming IPV6 TLS requests.</p> <p>If a non-TLS request is received on this port, then an error message is sent to the initiator and the request is terminated.</p> <p>This field must be different than the IIPORT parameter, and unique on the z/OS system.</p>
TLSLISTEN_ADAPTER_IPADDR (SLLLISTEN_ADAPTER_IPADDR)	<p>Defines the adapter that the Platform Server must bind to before issuing a TCP listen request for the TLS IP port.</p> <p>This indicates that the Platform Server accepts only requests that come in through the interface defined with this specified IP address. The default value ALL means to listen to all interfaces for incoming requests.</p> <p>The address specified must be in dotted decimal format and match an IP address defined for a network interface in the TCP/IP configuration.</p>
TLSLISTEN_ADAPTER_IPADDR_IPV6 (SLLLISTEN_ADAPTER_IPADDR_IPV6)	<p>Defines the IPv6 address of the TCP network interface that your Platform Server started task listens for incoming TLS connections.</p> <p>The default value ALL means to listen to all TCP network interfaces. If you want to listen to only a single network interface, specify the IPv6 address of the network interface, and the Platform Server only listens to that network interface for incoming requests.</p> <div data-bbox="558 1633 1409 1728" style="background-color: #f0f0f0; padding: 10px;"> <p>Note: This parameter is used only for incoming SSL requests, and is ignored for outgoing requests.</p> </div>

Parameter	Description
TLSTUNNELIPPORT	<p>Defines the IPPORT that MFT Platform Server listens on for IPV4 TLS tunnel requests. Only TLS tunnel requests are received on this port. If a non-SSL or an SSL request is received on this port, an error is displayed and the request fails. Because a transfer has not been initiated, no audit record is written. This field must be unique on the z/OS system. There is no default for this parameter. If this parameter is not defined, then IPV4 TLS tunnel processing is disabled.</p>
TLSTUNNELIPPORT_IPV6	<p>Defines the IPPORT that MFT Platform Server listens on for IPV6 TLS tunnel requests. Only TLS tunnel requests are received on this port. If a non-SSL or an SSL request is received on this port, an error is displayed and the request fails. Because a transfer has not been initiated, no audit record is written. This field must be unique on the z/OS system. There is no default for this parameter. If this parameter is not defined, then IPV6 TLS tunnel processing is disabled.</p>
TRANSFER_INTERFACE_PROTOCOL	<p>Defines the protocols that are accepted to initiate transfers.</p> <p>You can initiate transfers in three ways: cross memory, SNA communication and TCP communication. Cross memory requests are always accepted. Only TCP and SNA protocols can be restricted.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • ALL: TCP, SNA and cross memory requests are all accepted. This is the default value. • NONE: Only cross memory requests are accepted. • TCP: TCP and cross memory requests are accepted. • SNA: SNA and cross memory requests are accepted. <p>This parameter describes how transfers can be scheduled, but does not regulate how transfers are actually executed. Only the following programs that initiate transfers are referred to:</p> <ul style="list-style-type: none"> • Batch: OSIUB000 and OSIUC000

Parameter	Description
	<ul style="list-style-type: none"> • ISPF: REXX and ISPF panels
TRAP_COMMUNITY	<p>Defines the community name for the SNMP trap.</p> <p>Note: This parameter is case sensitive. Most SNMP communities are defined in lower case letters.</p>
TRAP_IPADDR	<p>Defines the IP address of the SNMP trap receiver.</p> <p>If the parameter is not defined, SNMP traps are not sent. If this parameter is defined, SNMP traps are sent to the address defined by this parameter.</p>
TRAP_IPPORT	<p>Defines the IP port to which the traps are sent.</p> <p>On most SNMP computers, the default port number is 162.</p>
TRAP_TRANSFER_REQUESTS	<p>Defines whether SNMP traps are sent for file transfers.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • YES: Trap requests are sent for all transfer requests that are completed either successfully or unsuccessfully. • NO: Trap requests are not sent. This is the default value. • ERRORS: Trap requests are sent when a request is unsuccessful and no more retries can be made.
UNEXPIRED_DISK_FILE_DELETE	<p>Defines whether MFT allows Receive Transfers defined with a disposition of DNEW to delete a file that has not expired: Valid values are:</p> <ul style="list-style-type: none"> • YES: Allows MFT to delete unexpired files when DISP=DNEW • NO: Does not allow MFT to delete unexpired files when DISP=DNEW. NO is the default value.
UNEXPIRED_DISK_FILE_UPDATE	<p>Defines whether MFT allows Receive Transfers to update a file that has not expired:</p> <ul style="list-style-type: none"> • YES: Allows MFT to update unexpired files.

Parameter	Description
	<ul style="list-style-type: none"> • NO: Does not allow MFT to update unexpired files. NO is the default value. <div data-bbox="578 411 1370 590" style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>Note: If this parameter is set to YES, message IEC507D is displayed and an operator response is awaited. Until the operator responds, subsequent dynamic allocations will hang. We suggest not setting this parameter to YES unless you have defined an automatic response to the IEC507D message.</p> </div>
UNIT	<p>Defines the unit name used when allocating new datasets created by the Platform Server responder, if the file availability is Immediate.</p> <p>This is the unit type for incoming disk datasets. You can adjust this to your installation requirements.</p>
VOLUME	<p>Defines the default volume serial used for new datasets created by the Platform Server responder.</p> <p>Do not leave this as the default value. You can adjust it to meet your installation requirements. If this parameter is omitted, the volume is assigned from the pool of devices for the unit type defined by the UNIT parameter.</p>
ZEDCCOMPRESSION	<p>Defines whether to use ZEDC data compression. Valid values are:</p> <ul style="list-style-type: none"> • Yes: This is the default value. When the ZEDC (zEnterprise Data Compression) hardware is installed and enabled and when the correct RACF definitions are made, MFT performs ZEDC ZLIB compression using the hardware compression accelerator card. When this option is set, all ZLIB compression will be ZLIB2. When the ZEDC compression is not available, MFT will fallback to software compression. • No: Does not use the ZEDC ZLIB compression. Uses software compression.

Parameter	Description
<p>Note: To use ZLIB hardware compression, you must grant READ access to Facility Class FPZ.ACCELERATOR.COMPRESSION to all transfer users that can perform hardware compression. Refer to the following manual for more details: <i>z/OS MVS Programming: Callable Services for High-Level Languages</i></p>	

Defining and Configuring the FUSCFG File

You must define and configure the FUSCFG file for TIBCO MFT Platform Server for z/OS.

FUSCFG Configuration File

To communicate with the Platform Server started task, the Platform Server clients interface must retrieve some information for the communication methods from the FUSCFG configuration file.

The following Platform Server clients use the FUSCFG configuration for communication with the Platform Server started task:

- (Required) REXX Interface
- (Required) ISPF Interface
- (Optional) Batch Interface
- (Optional) Script Interface

A configuration member called FUSCFG is located in the Platform Server EXECs library. You can define multiple configuration parameters in FUSCFG member to configure communications to multiple Platform Server started tasks.

Note: Do not change the name of the configuration member. FUSCFG must be the name of the configuration member.

For more information on how to use the CONFIG parameter to select a configuration entry, see the client interface chapters in *TIBCO® Managed File Transfer Platform Server for z/OS User's Guide*.

Methods of Communicating

Different communicating methods suit different scenarios.

The Platform Server clients interface communicates with the Platform Server started task in one of the following three ways.

- Through cross memory communication using PC calls
- Through SNA communications
- Through TCP communications using IP address or IP name

The most efficient method of communicating with the Platform Server address space is through cross memory calls. If you are running on the same mainframe as the Platform Server, it is good practice that you use cross memory communication method.

The least efficient method is communicating with the Platform Server through TCP using an IP name, because each connection request requires us to issue a DNS call to get the IP address for the Platform Server. If you are running on a different mainframe than the Platform Server, you can use either SNA or TCP to communicate with the Platform Server.

i Note: Use the IP name option only if the IP address of the Platform Server is dynamically defined and changes.

Parameter Syntax Rules

To configure the FUSCFG interface, you must define the parameters required to communicate with the started task.

When defining these required parameters, you have to comply with the following syntax rules:

- Parameters must be defined in upper case.
- An asterisk (*) in column 1 indicates a comment.
- If there is no data defined in the first 32 columns, the records are skipped.
- The records can start in any column from 1 to 32.
- The parameter must be completed on one card without continuation.
- One parameter can be entered on each line.

- The CONFIG parameter is required for each different configuration. Each configuration ends at the end of the member or when the next CONFIG parameter is identified.
- Do not embed spaces between the parameter, the equal sign (=) and the value.

FUSCFG Configuration Parameters

To configure the FUSCFG interface, you have to specify the required parameters.

See the following tables for the required parameters.

Common Configuration Parameter

The CONFIG parameter is required for all FUSCFG configurations.

Parameter	Description
CONFIG	<p>Defines the name of the Platform Server configuration entry.</p> <p>This parameter is required. Valid values can be up to 24 characters. This parameter must match the CONFIG= or CFG= parameter which can be included on the various interfaces.</p> <p>The special configuration name called DEFAULT must be defined, which is used when no CONFIG or CFG parameter is entered on the Platform Server command. This is only supported for REXX and ISPF. When the CONFIG option is used on the batch and script interfaces, no default value is available.</p>

Cross Memory Configuration Parameters

Three parameters are required to configure the Platform Server clients for cross memory communications to the Platform Server started task.

Parameter	Description
STCNAME	<p>Defines the name of the Platform Server started task.</p> <p>Valid values can be up to 8 characters. The Platform Server started task must be running on the same server as the client.</p>
AUDPATH1	Defines the name of the Platform Server AUDPATH1 VSAM file.

Parameter	Description
	<p>This file is created in Defining the Audit VSAM dataset. This name is also defined in the Platform Server started task JCL on the AUDPATH1 DD statement.</p> <p>If this parameter is not defined, the Platform Server does not attempt to allocate AUDPATH1, and uses an existing allocation if there is one.</p>
AUDPATH2	<p>Defines the name of the Platform Server AUDPATH2 VSAM file.</p> <p>This file is created in Defining the Audit VSAM dataset. This name is also defined in the Platform Server started task JCL on the AUDPATH2 DD statement.</p> <p>If this parameter is not defined, the Platform Server does not attempt to allocate AUDPATH2, and uses an existing allocation if there is one.</p>

SNA Configuration Parameters

Two parameters are required to configure the Platform Server clients for SNA communications to the Platform Server started task. Only one parameter is optional.

i Note: By using this method of communicating with the Platform Server started task, the client can run on any computer that is connected to the Platform Server mainframe by SNA. This can be the same mainframe, or a mainframe located at another location.

Parameter	Description
LUNAME	<p>Defines the name by which the Platform Server started task is known within an SNA network.</p> <p>This is a required parameter. This name is typically the APPLID that is defined to VTAM for the Platform Server started task. This information is defined in Defining VTAM Resources for Systems Using SNA.</p>

Parameter	Description
ACB	<p>Defines the name of the ACB that the Platform Server client uses to communicate with the started task.</p> <p>This is a required parameter. This name is typically 6 characters or less, and has a suffix of a two-digit number from 00 to 99. Therefore, multiple clients can communicate with the Platform Server started task using the same ACB prefix. This information is defined in Defining VTAM Resources for Systems Using SNA.</p> <p>Note: The name defined by the ACB parameter must not include the two digit suffix.</p>
MODENAME	<p>Defines the SNA mode name that is used when setting up SNA sessions.</p> <p>This is an optional parameter. Typically, the default values are defined when creating the Platform Server started task, and Platform Server ACB definitions are adequate.</p>

TCP Configuration Parameters

Two parameters are required to configure the Platform Server clients for TCP communications to the Platform Server started task. Only one parameter is optional.

i Note: By using this method of communicating with the Platform Server started task, the client can run on any computer that is connected to the Platform Server mainframe through a TCP network. This can be the same mainframe, or a mainframe located at another location.

Parameter	Description
IPADDR	<p>Defines the IP address of the mainframe in dotted decimal format.</p> <p>This is a required parameter. This is typically the IP address of TCP running on the z/OS LPAR that the Platform Server started task is running on.</p>

Parameter	Description
	The IPNAME parameter can be used in place of this parameter.
IPNAME	<p>Defines the IP name of the mainframe.</p> <p>This is a required parameter. This name must be defined in a host configuration file, or by a DNS server. This is typically the IP name of TCP running on the z/OS LPAR that the Platform Server started task is running on.</p> <p>The IPADDR parameter can be used in place of this parameter.</p>
TCPIPJOBNAME	<p>Defines the name of the TCP/IP started task that is running on the same mainframe as the Platform Server REXX exec, ISPF user, or batch job.</p> <p>This is a required parameter. To communicate with TCP/IP, the client has to know the TCP/IP job name. Typically, the value is TCP/IP.</p>
IPPORT	<p>Defines the IPPORT on which the Platform Server started task is listening for connection requests.</p> <p>This is an optional parameter. The default value is 46464. This value must match the number defined on the Platform Server started task GLOBAL TCPIPSPORT parameter.</p>

Sample FUSCFG Configuration

Member FUSCFG, which is a sample shipped with this product, exists in the Platform Server EXECs library.

See the following sample FUSCFG configuration for your reference.

```
*   DEFAULT CONFIGURATION USED IF CONFIG=(OR CFG=) IS NOT ENTERED
CONFIG=DEFAULT
STCNAME=FUSION
AUDPATH1=FUSION.AUDIT.PATH1
AUDPATH2=FUSION.AUDIT.PATH2

*   SAMPLE CONFIG FOR STARTED TASK COMMUNICATION WITH FUSION SERVER
```

```

CONFIG=FUSION_STC
STCNAME=FUSION
AUDPATH1=FUSION.AUDIT.PATH1
AUDPATH2=FUSION.AUDIT.PATH2

*   SAMPLE CONFIG FOR SNA COMMUNICATION WITH FUSION SERVER USING SNA
CONFIG=FUSION_SNA
LUNAME=FUSNAPPL
ACB=FUSN
*MODENAME=#BATCH

*   SAMPLE CONFIG FOR TCP COMMUNICATION WITH FUSION SERVER USING IPADDR
CONFIG=FUSION_IPADDR
IPADDR=14.0.0.0
TCPIPJOBNAME=TCPIP
*IPPORT=46464

*   SAMPLE CONFIG FOR TCP COMMUNICATION WITH FUSION SERVER USING IPNAME
CONFIG=FUSION_IPNAME
IPNAME=NYES9000
TCPIPJOBNAME=TCPIP
*IPPORT=46464

```

The FUSCFG configuration member contains the following 5 configurations:

- **Default configuration:** Defines the default configuration that is set by specifying CONFIG=DEFAULT. This configuration is used if the user does not enter the CONFIG (CFG) parameter on the Platform Server REXX or ISPF Interface. In this case, the default configuration is to use cross memory calls to the Platform Server started task because the STCNAME parameter is defined.
- **Sample started task configuration:** Defines the configuration parameters for communicating with the Platform Server started task through cross memory calls. To use this configuration, you must specify CONFIG=FUSION_STC on the command line when you execute a Platform Server client request.
- **Sample SNA configuration:** Defines the configuration parameters required to communicate with the Platform Server started task through VTAM/SNA services. To use this configuration, you must specify CONFIG=FUSION_SNA on the command line when you execute a Platform Server client request.
- **Sample TCP IPADDR configuration:** Defines the configuration parameters required to communicate with the Platform Server started task through TCP/IP. Using this configuration, the Platform Server REXX attempts to make a connection to the Platform Server at IP address 14.0.0.0 (loopback address). To use this configuration,

you must specify `CONFIG=FUSION_IPADDR` on the command line when you execute a Platform Server REXX exec.

- **Sample TCP IPNAME configuration:** Defines the configuration parameters required to communicate with the Platform Server started task through TCP/IP. Using this configuration, the Platform Server REXX attempts to make a connection to the Platform Server defined to the DNS as NYES9000. To use this configuration, you must specify `CONFIG=FUSION_IPNAME` on the command line when you execute a Platform Server REXX exec.

Defining Remote Systems in Configuration Library

By creating a member in the configuration library for each remote system, you can define each remote system to which you want to initiate activities.

The member name in the configuration library is the name by which you can refer to that remote node when initiating transactions.

Configuration Member List - Member Name CONFIG

The configuration library is specified in the `CONFIG DD` statement in the JCL that brings up the Platform Server started task.

Three types of members are included in the configuration library. See the following table for the member descriptions.

Member Name	Description
CONFIG member	<p>This member sorts through a list of members to activate when the Platform Server is initialized.</p> <p>The CONFIG member of the Platform Server definition library is a list of other members in the library to activate.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>Note: This member is used only when the <code>GLOBAL AUTOENABLE</code> parameter is set to <code>NO</code>, which is the default value.</p> </div>
Node member	This member defines addressing information about the remote system.

Member Name	Description
List of remote systems member	This member defines a list of remote systems.

Members within the CONFIG DD statement can be enabled through the following two ways:

- Through the CONFIG member of the CONFIG DD statement. This method is used when the GLOBAL AUTOENABLE parameter is set to NO, which is the default value.
For more information, see [Enabling Members When GLOBAL AUTOENABLE=NO](#).
- By reading the members within the CONFIG DD statement and enabling each member that specifies AUTOENABLE=YES. This method is used when the GLOBAL AUTOENABLE parameter is set to YES. This method must be used when the Command Center features are used to define and enable node definitions.
For more information, see [Enabling Members When GLOBAL AUTOENABLE=YES](#).

Enabling Members When GLOBAL AUTOENABLE=NO

When the GLOBAL AUTOENABLE parameter is set to NO, the members to be enabled are defined by the CONFIG member within the CONFIG DD statement.

The syntax for the CONFIG member is as follows:

```
member1,member2
member3,
|
|
member n
```

The first member name must start in column 1. Use a comma between member names. If you want to continue to more than one line, you can use more lines. Columns 73 to 80 are ignored. An asterisk (*) in column 1 denotes a comment.

In the following example, three members are enabled at Platform Server initialization:

```
*****
* THIS IS WHERE YOU WILL SPECIFY THE OTHER MEMBERS THAT CONTAIN *
* CONFIGURATION INFORMATION THAT WILL BE NEEDED TO COMMUNICATE TO *
* THEM. THE SAMPLES BELOW SHOULD BE REMOVED AND YOU SHOULD *
* SPECIFY THE NAMES THAT YOU HAVE CONFIGURED FOR THE REMOTE *
* SYSTEMS IN YOUR NETWORK. *
```



```

*****
SNANODE,           CONTAINS A SAMPLE OF A REMOTE SNA NODE
TCPNODE,           CONTAINS A SAMPLE OF A REMOTE TCP/IP NODE
DISTRIB            CONTAINS A SAMPLE OF A LIST OF NODES

```

Enabling Members When GLOBAL AUTOENABLE=YES

When the GLOBAL AUTOENABLE parameter is set to YES, the Platform Server reads all of the members in the CONFIG DD.

All of the members are enabled unless the AUTOENABLE=NO parameter is set within the CONFIG member. This method is required when using the Command Center functions to add or delete node definitions.

Node Definition Parameters

Each remote system is defined in a separate member of a partitioned dataset, which is called the Platform Server definition library.

These members have each parameter in a separate record. An asterisk (*) in column 1 denotes a comment.

Parameter	Description
ACCEPT _ VERIFIED_USER	<p>Defines whether verified users on other z/OS platforms can be used on this system without a password.</p> <p>Valid values are YES and NO. The default value is to let the GLOBAL ACCEPT_VERIFIED_USER parameter determine if verified users are allowed.</p> <p>If a user submits a file transfer request on a z/OS system, by default the local user ID is sent to the remote system. When an incoming request is received, the Platform Server searches through the enabled nodes to find a match on a node.</p> <p>The following logic is performed:</p> <p>If this is a TCP request, issues a socket call to get the IP name associated with the IP address. Scans through the node definition looking for a match on the IP address, IP name or LU name.</p>

Parameter	Description
ALLOW_TRANSFER_REQUESTS	<ul style="list-style-type: none"> • If a match is found, then the node associated with the incoming request is found. • If no match is found, scans through the node definition looking for a match on the backup IP address, backup IP name or backup LU name. <ul style="list-style-type: none"> ◦ If there is a match, then the node associated with the incoming request has been found. ◦ If there is no match on these fields, then a node is not associated with the incoming request. If the GLOBAL configuration is set to require a node on incoming requests, the request fails.
ALLOW_TRANSFER_REQUESTS	<p>Defines whether transfer schedule requests initiated through TCP or SNA are accepted when the GLOBAL ALLOW_TRANSFER_REQUESTS parameter is set to NODE.</p> <p>This parameter applies to these Platform Server interfaces: batch, script, API, REXX and ISPF.</p> <div style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p>Note: This parameter applies only to transfers scheduled through SNA or TCP. It does not apply to transfers managed by Command Center.</p> </div> <p>Valid values are:</p> <ul style="list-style-type: none"> • YES: TCP and SNA transfer requests are accepted. This is the default value. • NO: TCP and SNA transfer requests are not accepted. Only requests initiated on the same server and scheduled through cross memory are accepted. • LOGON: Requests are only accepted when local credentials are included with the transfer schedule request.
ALLOW_MANAGE_REQUESTS	<p>Defines whether transfer management requests initiated through TCP or SNA are accepted.</p>

Parameter	Description
AUTOENABLE	<p>This parameter applies to these Platform Server interfaces: batch, script, API, REXX and ISPF. Management requests are typically ISPF or REXX transfer inquiry.</p> <p>Note: This parameter applies only to transfers managed through SNA or TCP. It does not apply to transfers managed by Command Center.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • YES: TCP and SNA management requests are accepted. This is the default value. • NO: TCP and SNA management requests are not accepted. Only requests initiated on the same server and scheduled through cross memory are accepted.
BACKUPLUNAME	<p>Defines whether the member is enabled at the Platform Server startup when the GLOBAL AUTOENABLE parameter is set to YES.</p> <p>This parameter is ignored when the GLOBAL AUTOENABLE parameter is set to NO.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • YES: The member is enabled at startup. This is the default value. • NO: The member is not enabled at startup. <p>Defines the SNA LUNAME that is used as a backup in case the transfer request to the primary node definition or previously defined backup definitions fails with a network error.</p> <p>Up to 5 backup LU name, backup IP address or backup IP name definitions can be defined for a single node definition. You can switch between SNA and IP backup definitions.</p> <p>For an example of BACKUPLUNAME, see the SNANODE and TCPNODE members in the SAMPLIB dataset.</p>

Parameter	Description
<p>Note: This parameter is only used for initiator requests.</p>	
BACKUPMODENAME	<p>Defines the SNA mode name used as a backup in case the transfer request to the primary node definition or previously defined backup definitions fails with a network error.</p> <p>This parameter overrides the SNA mode name for the previous BACKUPLUNAME definition. This mode name becomes the default mode name for all future BACKUPLUNAME definitions within this node definition.</p> <p>For an example of BACKUPMODENAME, see the SNANODE and TCPNODE members in the SAMPLIB dataset.</p>
BACKUPIPADDR	<p>Defines the IP address that is used as a backup in case the transfer request to the primary node definition or previously defined backup definitions fails with a network error.</p> <p>Up to 5 backup LU name, backup IP address or backup IP name definitions can be defined for a single node definition. You can switch between SNA and IP backup definitions.</p> <p>For an example of BACKUPIPADDR, see the SNANODE and TCPNODE members in the SAMPLIB dataset.</p> <p>Note: This parameter is only used for initiator requests.</p>
BACKUPIPNAME	<p>Defines the IP name that is used as a backup in case the transfer request to the primary node definition or previously defined backup definitions fails with a network error.</p> <p>Up to 5 backup LU name, backup IP address or backup IP name definitions can be made for a single node definition. You can switch between SNA and IP backup definitions.</p> <p>For an example of BACKUPIPNAME, see the SNANODE and TCPNODE members in the SAMPLIB dataset.</p>

Parameter	Description
	<p>Note: This parameter is only used for initiator requests.</p>
BACKUPIPPORT	<p>Defines the IP port that is used as a backup in case the transfer request to the primary node definition or previously defined backup definitions fails with a network error.</p> <p>This parameter overrides the IP port number for the previous BACKUPIPNAME or BACKUPIPADDR definition. This IP port becomes the default IP port for all future BACKUPIPADDR and BACKUPIPNAME definitions within this node definition.</p> <p>For an example of BACKUPIPPORT, see the SNANODE and TCPNODE members in the SAMPLIB dataset.</p>
BPS_INIT	<p>Defines the total number of TCP bytes that can be transferred per second for a node in initiator requests.</p> <p>Note: This parameter applies to both send and receive transfers.</p> <p>The parameter must be specified in kilobytes or megabytes. Therefore, the letter K or M must be the last character in this argument. Valid values can be up to 999999999.</p> <p>For example, to specify the value of 2 million bytes, you can specify 2M or 2000K.</p>
BPS_RECV	<p>Defines the total number of TCP bytes that can be received per second from a node.</p> <p>This parameter must be specified in kilobytes or megabytes. Therefore, the letter K or M must be the last character in this argument. Valid values can be up to 999999999.</p> <p>For example, to specify the value of 2 million, you can specify 2M or 2000K.</p>
BPS_RESP	<p>Defines the total number of TCP bytes that can transferred per second for a node in responder requests.</p>

Parameter	Description
BPS_SEND	<p>Note: This parameter applies to both send and receive transfers.</p> <p>This parameter must be specified in kilobytes or megabytes. Therefore, the letter K or M must be the last character in this argument. Valid values can be up to 999999999.</p> <p>For example, to specify the value of 2 million bytes, you can specify 2M or 2000K.</p>
BPS_TOTAL	<p>Defines the total number of TCP bytes that can be sent per second to a node.</p> <p>The parameter must be specified in kilobytes or megabytes. Therefore, the letter K or M must be the last character in this argument. Valid values can be up to 999999999.</p> <p>For example, to specify the value of 2 million, you can specify 2M or 2000K.</p>
BPS_TOTAL	<p>Defines the total number of TCP bytes that can be transferred per second for a node for both send and receive requests.</p> <p>The parameter must be specified in kilobytes or megabytes. Therefore, the letter K or M must be the last character in this argument. Valid values can be up to 999999999.</p> <p>For example, to specify the value of 2 million, you can specify 2M or 2000K.</p>
COMMAND_CENTER_SUPPORT	<p>Defines the support level of the Command Center functions on this node.</p> <p>The Command Center user must still be validated before they can use the Command Center functions.</p> <p>Note: Multiple values can be specified on this parameter.</p> <p>Valid values are:</p>

Parameter	Description
	<ul style="list-style-type: none"> • ALL: NODE, PROFILE, AUDIT, PING, and TRANSFER functions are supported on this node. • NONE: No Command Center function is supported on this node. This is the default value. • NODE: Node list and update functions are supported on this node. • PROFILE: Profile list and update functions are supported on this node. • AUDIT: Requests that inquire on the Platform Server audit file are supported on this node. • ALTER: Requests that alter transfers in the Platform Server inactive queue are supported on this node. • PING: Platform Server PING requests are supported on this node. • TRANSFER: The Command Center transfer function that initiates file transfers is supported on this node.
CONTENTION_LOSERS	<p>Defines the limit number of simultaneous responder transfers from a partner in TCP communications.</p> <p>In SNA communication, this parameter limits the number of incoming sessions that can be created by the remote system.</p> <p>For TCP Communication, this parameter defines the maximum number of Responder requests that will be accepted from the source system. If the number of concurrent requests exceeds this number, the request will be rejected. The source system will treat this as a retry-able error and will attempt a retry if the number of tries has not been reached.</p> <p>Valid values are from 0 to 256. The default value 0 means unlimited.</p>
CONTENTION_WINNERS	<p>Defines the limit number of simultaneous transfers to a partner in both SNA and TCP communications.</p>

Parameter	Description
CRCCHECK	<p>Likewise, this parameter can limit the number of simultaneous scripts that can be executed. Separate counters are maintained for transfers and scripts. Therefore, you can have a number of scripts and transfers executing at the same time for a defined node.</p> <p>Valid values are from 0 to 256. The default value is 5.</p> <p>Defines whether CRC checking is turned on by default. CRC checking is performed against the data that is sent over the network to ensure that the data is not corrupted. The CRC is not performed against the file itself. ASCII to EBCDIC conversion, translation, and LF/CRLF change the contents of the file between the sender and receiver so the CRC is not performed against the file contents. Valid values are:</p> <ul style="list-style-type: none"> • YES: Performs CRC checking by default • NO: Does not perform CRC checking by default. The default value is NO. <p>The CRCCHECK parameter on the Node and batch parameters overrides the Global CRCCHECK settings.</p> <p>Note: If the partner Platform Server does not support CRC checking, CRC is still computed but is not checked against the partner's CRC value computed.</p>
DEFAULT_CHECKPOINT	<p>Defines the checkpoint interval in minutes for transfers.</p> <p>By specifying this parameter, all requests can take a checkpoint restart. Valid values are NO or a number from 1 to 999. The default value NO means not to take checkpoints. Transfers that do not use checkpoint restart start from the beginning of the file.</p>

Parameter	Description
DEFAULT_COMPRESS or COMPRESS	<p data-bbox="505 310 1273 485">Note: Using checkpoint restart when writing to variable length blocked records might result in a D37 abend if insufficient secondary extent is defined. Ensure sufficient secondary extent is allocated or checkpoint restart is turned off for variable length blocked records.</p> <p data-bbox="485 558 1127 585">Defines the default compression for SAPI requests.</p> <p data-bbox="485 617 691 644">Valid values are:</p> <ul data-bbox="534 676 1300 1031" style="list-style-type: none"> • YES: Uses RLE compression for SAPI requests. • NO: Does not use compression for SAPI requests. This is the default value. • RLE: Uses RLE compression for SAPI requests. • LZ: Uses LZ compression for SAPI requests. • ZLIB1 to ZLIB9: Uses ZLIB Compression for SAPI requests. • NEVER: Never uses compression.
DEFAULT_COSNAME	<p data-bbox="485 1079 1305 1150">Defines the default COS name that is used when communicating with this node.</p> <p data-bbox="485 1182 1305 1373">This parameter can be overridden by entering the COSNAME parameter on the Platform Server batch interface. This name must match a COS entry that is enabled at startup or by the ENABLE operator command. The default value for this parameter is the COS entry called COSDFLT.</p>
DEFAULT_CRLF	<p data-bbox="485 1421 894 1449">Defines the default CRLF option.</p> <p data-bbox="485 1480 1279 1551">This parameter is only used for initiator requests when you do not enter the CRLF parameter.</p> <p data-bbox="485 1583 691 1610">Valid values are:</p> <ul data-bbox="534 1642 1159 1715" style="list-style-type: none"> • YES CRLF: Records are delimited by Carriage Return/Line Feed (typically PC based systems).

Parameter	Description
DEFAULT_ENCRYPT or ENCRYPT	<ul style="list-style-type: none"> • LF: Records are delimited by Line Feeds (typically UNIX based systems). • NO: No record delimiters are used in the file. This is the default value. <p>Defines the level of encryption that is used by this node.</p> <p>If specified, this parameter overrides the ENCRYPT parameter specified in the GLOBAL member.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • NONE: No encryption • AES: AES 256-bit encryption • AES128: AES 128-bit encryption • DES: DES encryption • 3DES: Triple DES encryption • BF BLOWFISH: Blowfish 56-bit encryption • BFL BLOWFISH_LONG: Blowfish 448-bit encryption • RJ RIJNDAEL: Rijndael 256-bit encryption • NEVER: No encryption. Even if you override the ENCRYPT option in the batch interface, encryption is turned off for this node. <p>If the PASSONLY operand is used, then only the password is encrypted using the encryption algorithm specified. The data is not to be encrypted.</p> <div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p>Note: PASSONLY is only valid on z/OS nodes.</p> </div> <p>For more information, see the Authentication, Encryption, Compression and Email sections of <i>TIBCO® Managed File Transfer Platform Server for z/OS User's Guide</i>.</p>

Parameter	Description
DEFAULT_LOCALCTFILE	<p data-bbox="505 310 1224 380">Note: This parameter is ignored for TLS Tunnel requests because the TLS Tunnel encrypts all data.</p> <p data-bbox="488 447 1198 516">Defines the default local translation table entry used for communications with this node.</p> <p data-bbox="488 548 1271 617">Translation tables are defined in the CONFIG DD. They can be enabled at startup or through the ENABLE operator command.</p> <p data-bbox="488 648 691 676">Valid values are:</p> <ul data-bbox="537 707 1292 873" style="list-style-type: none"> <li data-bbox="537 707 1255 777">• <i>local_translation_table_name</i>: Translation table name must be from 1 to 16 characters. <li data-bbox="537 808 1292 873">• NONE NULL: No conversion takes place. This is the default value. <p data-bbox="488 884 1295 953">This parameter is used when TYPE=TEXT is defined and ignored when TYPE=BINARY is specified.</p> <p data-bbox="488 984 1192 1054">The LOCALCTFILE parameter on the file transfer request overrides the DEFAULT_LOCALCTFILE parameter.</p> <p data-bbox="488 1085 1287 1155">To use z/OS Unicode Conversion Services, the following format of this parameter is required.</p> <p data-bbox="488 1186 867 1213">LOCALCTFILE=CC:xxxxx:yyyyy</p> <p data-bbox="488 1245 574 1272">Where:</p> <p data-bbox="488 1304 1274 1373">xxxxx defines the CCSID of the Local data - usually the EBCDIC CCSID.</p> <p data-bbox="488 1404 1279 1474">yyyyy defines the CCSID of the Remote data - usually the ASCII CCSID</p> <p data-bbox="488 1505 1308 1575">For information on the CCSID values to use, see the IBM manual: <i>z/OS Unicode Services User's Guide and Reference</i>.</p> <p data-bbox="488 1606 992 1633">Example: LOCALCTFILE=CC:01140:01208</p> <ul data-bbox="537 1665 1265 1734" style="list-style-type: none"> <li data-bbox="537 1665 1265 1734">• For a Send Request, this parameter converts data from EBCDIC CCSID 1140 to ASCII UTF-8 CCSID 01208.

Parameter	Description
	<ul style="list-style-type: none"> For a Receive Request, this parameter converts data from ASCII UTF-8 CCSID 01208 to EBCDIC CCSID 1140. <p>When using this parameter, set the following parameter so that the partner does NOT perform any conversion:</p> <p>REMOTECTFILE=NULL</p>
DEFAULT_LOGON_DOMAIN	<p>Defines a 16-byte Windows domain name.</p> <p>This parameter is required when communicating with a Windows system, and defines the domain where the user ID validated by Windows is located. It has no default value.</p>
DEFAULT_NODECLASS	<p>Defines the default node class for this node definition.</p> <p>Valid values are from 0 to 255. The default value 0 means that node class processing is not turned on for this node. By setting this parameter to a non-zero value, node class processing is turned on for this node.</p> <p>If the transfer request has a node class of 0, the DEFAULT_NODECLASS value replaces the NODECLASS defined for the request. DEFAULT_NODECLASS can never be set to a value higher than the CONTENTION_WINNERS parameter.</p> <p>When the Platform Server NODECLASS facility is turned on, the Platform Server assigns a NODECLASS value to every possible connection. Each of the node classes represents one session.</p> <p>If you define 6 winners, then the Platform Server creates 6 node classes: 1, 2, 3, 4, 5, and 6. A transfer can run at the assigned node class or a node class with a higher number. For example, a transfer defined with NODECLASS=3 can run in class 3, 4, 5 or 6.</p>
DEFAULT_REMOTECTFILE	<p>Defines the default remote translation table entry used for communications with this node.</p> <p>Remote translation tables must be defined on the remote Platform Server node. Valid values are NONE or a translation</p>

Parameter	Description
	<p>table name of 1 to 16 characters. The default value NONE indicates that no conversion takes place.</p> <p>This parameter is used when TYPE=TEXT is defined and is ignored when TYPE=BINARY is specified. The</p> <p>The REMOTEFILE parameter on the file transfer request overrides the DEFAULT_REMOTEFILE parameter.</p> <p>To use z/OS Unicode Conversion Services, the following format of this parameter is required.</p> <pre>REMOTEFILE=CC:xxxxx:yyyyy</pre> <p>Where:</p> <p>xxxxx defines the CCSID of the Local data - that is, z/OS data read or written to disk.</p> <p>yyyyy defines the CCSID of the Remote data - that is, data sent to or received from a partner</p> <p>For information on the CCSID values to use, see the IBM manual: <i>z/OS Unicode Services User's Guide and Reference</i>.</p> <ul style="list-style-type: none"> • For a Send Request (that is, Read), xxxxx is the Source CCSID and yyyyy is the target CCSID. • For a Receive Request, (that is, Write), yyyyy is the Source CCSID and xxxxx is the target CCSID. <p>When using this parameter, set the following parameter so that the local system does NOT perform any conversion:</p> <pre>LOCALFILE=NULL</pre>
DEFAULT_PASSWORD	<p>Defines a password from 1 to 20 bytes that is used as the remote password for SAPI requests.</p> <p>If this parameter is not defined, then a user profile must be defined for this node and the user associated with the started task. This parameter is typically used together with the DEFAULT_USERID parameter. It has no default value.</p>

Parameter	Description
DEFAULT_TYPE	<p>Defines the default file transfer type.</p> <p>This parameter is only used for initiator requests when the file transfer type is not entered.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • TEXT: ASCII/EBCDIC conversion takes place. • BINARY: No conversion takes place. • DATA: The file to be created must be a data file, as opposed to a source file. This option must be used only to communicate with AS/400 systems.
DEFAULT_USERID	<p>Defines a user ID from 1 to 20 bytes that is used as the remote user ID for SAPI requests.</p> <p>If this parameter is not defined, then a user profile must be defined for this node and the user associated with the started task. This parameter is typically used together with the DEFAULT_PASSWORD parameter. It has no default value.</p>
DESCRIPTION	<p>Defines a 32-byte description for a node definition.</p> <p>This parameter is meant for documentation purposes only, and does not change how file transfers run. This description is displayed on the ISPF node detailed display menu.</p>
DISCONNECT	<p>Defines whether the host disconnects the session with the remote side if there are no more activities for this remote system.</p> <p>Valid values are YES and NO. The default value is NO.</p> <p>This parameter is used in conjunction with the WAIT_FOR_SESSION parameter, so the users on mainframe can queue up transactions for remote systems that are not connected and have the Platform Server wait for that system to connect before dispatching the transfers.</p>

Parameter	Description
EMAIL_FAIL	<p data-bbox="505 310 1159 380">Note: This parameter cannot be used for TCP node definitions.</p> <p data-bbox="485 443 1265 554">Defines a single email address or multiple email addresses to send a notification message if a file transfer request with this node is unsuccessful.</p> <p data-bbox="485 590 1279 701">You have to separate multiple email addresses by a semicolon (;). The maximum field length is 64 characters including any semicolons.</p> <p data-bbox="485 730 1292 919">As an alternative, an email distribution list defined in the organization's email system might be used to specify multiple email addresses to receive this notification. This parameter has no default value, which means that no email notification takes place when a request fails.</p>
EMAIL_GOOD	<p data-bbox="485 963 1265 1075">Defines a single email address or multiple email addresses to send a notification message if a file transfer request with this node is successful.</p> <p data-bbox="485 1108 1279 1220">You have to separate multiple email addresses by a semicolon (;). The maximum field length is 64 characters including any semicolons.</p> <p data-bbox="485 1249 1292 1440">As an alternative, an email distribution list defined in the organization's email system might be used to specify multiple email addresses to receive this notification. This parameter has no default value, which means that no email notification takes place when a request succeeds.</p>
EMAIL_NOTIFY	<p data-bbox="485 1484 1284 1554">Defines when the Platform Server sends email notifications for transfers with this node.</p> <p data-bbox="485 1587 695 1614">Valid values are:</p> <ul data-bbox="534 1648 1243 1717" style="list-style-type: none"> <li data-bbox="534 1648 1243 1717">• NONE: No email notification is sent. This is the default value.

Parameter	Description
ENFORCE_SECURITY_POLICY	<ul style="list-style-type: none"> • INITIATOR: An email notification is sent for initiator requests. • RESPONDER: An email notification is sent for responder requests. • BOTH: An email notification is sent for initiator and responder requests. <p>Note: The node EMAIL_GOOD and EMAIL_FAIL parameters are also required before notifications take place.</p>
ENFORCE_SECURITY_POLICY	<p>Defines which Platform Server security policy is enforced.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • FIPS140: FIPS 140 security policy. • HIPAA: HIAA standards. The standards require all file transfers to use encryption key length that is 128 bits or greater. This checking takes place for both initiator and responder requests. • NO: No security policy. This is the default value.
FILE_ERROR_TRY_COUNT	<p>Defines the maximum number of times that a file transfer is tried when temporary file errors occur. This parameter overrides the Global FILE_ERROR_TRY_COUNT parameter for transfers initiated to this node. Valid values are from 1 to 9999.</p> <p>The default value is 0. Using the default value means that the Global value is used.</p>
FILE_ERROR_TRY_INTERVAL	<p>Defines the amount of time in seconds that a request waits after a retry-able file error and before the request is retried. This parameter overrides the Global FILE_ERROR_TRY_INTERVAL parameter for transfers initiated to this node.</p> <p>The default value is 0. Valid values are from 10 to 9999. Using the default value means that the Global value is used.</p>

Parameter	Description
INITHOLD	<p>The actual interval can be up to 60 seconds or more than the value specified, depending on when the error occurs and when the dispatcher runs.</p> <p>Defines whether initiator requests for this node definition are held when the node is enabled.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • YES: Initiator requests for this node are not executed. This parameter can be reset through the ISPF NODE panel, the FUSNODE INITREL REXX exec or the INITREL operator command. • NO: Initiator requests for this node are executed. Nodes can be placed in an INITHOLD state through the ISPF NODE panel, the FUSNODE INITHOLD REXX exec or the INITHOLD operator command. This is the default value.
INITIATOR_ERROR_HOLD	<p>Defines the number of consecutive remote errors that cause the node to be placed in an INITHOLD state.</p> <p>Note: This parameter applies only to initiator requests.</p> <p>Valid values are from 0 to 9999.</p> <p>Each time an initiator request fails with a remote nonrecoverable error, the count is increased by one. When the count exceeds the value defined for this parameter, the node is placed in an INITHOLD state.</p> <p>Each time a successful initiator request is performed with this node, the counter is reset. The default value of 0 indicates that the node is not placed into an INITHOLD state due to remote errors.</p>
IPADDR	<p>Defines the IP address of the remote destination. You can enter either an IPV4 IP Address or an IPV6 IP Address.</p> <p>This parameter is mutually exclusive with IPNAME. In TCP/IP</p>

Parameter	Description
IPNAME	<p data-bbox="485 296 1247 365">transfers, either IPNAME or IPADDR must be defined. In VTAM transfers, LU_NAME must be defined.</p> <p data-bbox="485 415 1081 443">Defines the IP name of the remote destination.</p> <p data-bbox="485 478 1208 548">This name can contain up to 64 characters. For example, JohnD.XYZ.COM.</p> <p data-bbox="485 579 1247 688">This parameter is mutually exclusive with IPADDR. In TCP/IP transfers, either IPNAME or IPADDR must be defined. In VTAM transfers, LU_NAME must be defined.</p> <div data-bbox="488 716 1308 961" style="border: 1px solid #ccc; padding: 10px; background-color: #f9f9f9;"> <p data-bbox="505 737 1284 947">Note: For IBM TCP version 3.1 and version 3.2, if you want to use the DNS lookup feature (IPNAME=), you must concatenate the IBM C Runtime libraries in the STEPLIB if these libraries are not defined to the z/OS link list. Also, the SYSTCPD dd name must be in the Platform Server JCL and must point to the TCP/IP data configuration file.</p> </div>
IPPORT	<p data-bbox="485 1014 1062 1041">Defines the IP port of the remote destination.</p> <p data-bbox="485 1077 1284 1188">Valid values are from 0 to 65535. The default value is 46464. A value of 0 indicates that the Platform Server for z/OS must use the value defined in the GLOBAL TCIPPORT parameter.</p>
LU_NAME	<p data-bbox="485 1234 1276 1304">Defines the SNA server LU name that the Platform Server uses to communicate.</p> <p data-bbox="485 1339 1300 1444">The logical unit handles and enforces the protocols required for user-to-user communications. This parameter is mutually exclusive with the IPNAME and IPADDR parameters.</p>
MODENAME	<p data-bbox="485 1491 1300 1560">Defines a 1-character to 8-character value that corresponds to a VTAM MODEENT.</p> <p data-bbox="485 1596 1268 1665">You must define a VTAM mode entry. For an example, see the Platform Server sample library.</p>
MQ_FAIL	<p data-bbox="485 1711 1279 1738">Defines the message queue where the Platform Server sends a</p>

Parameter	Description
MQ_GOOD	<p>notification message when a file transfer request is completed unsuccessfully.</p> <p>Valid values are \$MQ:MQ_Message_Queue_Name:Message_Queue_Name. \$MQ is a literal and must be the first three characters. The message queue name must be from 1 to 55 characters.</p> <p>This parameter has no default value, which means that no MQ notification takes place when a request fails.</p>
MQ_GOOD	<p>Defines the message queue where the Platform Server sends a notification message when a file transfer request is completed successfully.</p> <p>Valid values are \$MQ:MQ_Message_Queue_Name:Message_Queue_Name. \$MQ is a literal and must be the first three characters. The message queue name must be from 1 to 55 characters.</p> <p>This parameter has no default value, which means that no MQ notification takes place when a request is successful.</p>
MQ_NOTIFY	<p>Defines when the Platform Server sends MQ notifications.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • NONE: No MQ notification is sent. This is the default value. • INITIATOR: MQ notification is sent for initiator tasks. • RESPONDER: MQ notification is sent for responder tasks. • BOTH: MQ notification is sent for both initiator and responder tasks. <p>Note: The GLOBAL MQ_GOOD and MQ_FAIL parameters are also required before notifications take place.</p>
NODENAME	<p>Defines the node name associated with this node definition.</p> <p>Valid values can be from 1 to 32 characters and cannot have any embedded spaces.</p>

Parameter	Description
PARALLEL	<p>If this parameter is defined, then transfers must use this node name in transfers defined to use the NODE parameter.</p> <p>If this parameter is not defined, the 1 to 8 character member name is used as the node name.</p>
PARALLEL	<p>Defines whether multiple concurrent sessions can run between a pair of LU6.2 logical units (LUs), which means multiple operations can be performed simultaneously.</p> <p>Valid values are NO and YES.</p> <ul style="list-style-type: none"> • If this LU is used with an independent local LU, and parallel sessions are used, you can keep the default value YES. • If this LU is used with a dependent local APPC LU, you can specify NO. <p>This parameter can also be specified along with CONTENTION_WINNERS for TCP nodes to limit the amount of requests that can be initiated by a TCP node.</p>
PLATFORM	<p>Defines the platform under which the node is executing.</p> <p>This parameter is for information purposes only. It has no default value. Valid values are WINDOWS, AIX, HPUX, SUN, ZOS, AS400, LINUX, or MFT Platform ServerCC.</p>
RESPHOLD	<p>Defines whether responder requests for this node definition are held when the node is enabled.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • YES: Responder requests for this node are rejected with a recoverable error. The RESPHOLD can be reset through the ISPF NODE panel, the FUSNODE RESPREL REXX exec or the RESPREL operator command. • NO: Responder requests for this node are accepted. Nodes can be placed in RESPHOLD state through the ISPF NODE

Parameter	Description
	panel, the FUSNODE RESPHOLD REXX exec or the RESPHOLD operator command. This is the default value.
RESPONDER_PROFILE	<p>Defines whether this node requires responder profiles or not when a request comes in.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • YES: Requires responder profiles. • NO: Does not require responder profiles. This is the default value. • DUAL: Responder profiles are checked. If no matches are found within the responder profile, the request can continue to be processed, and the user ID and password are checked against the RACF or ACF2 or Top Secret security system. <p>The default value for this parameter is to let the GLOBAL RESPONDER_PROFILE parameter determine if responder profiles are used. For more information, see User Profiles.</p>
SAPI_CLASS	<p>Defines one of the SAPI selection criteria.</p> <p>Valid values can be up to 8 JES SYSOUT classes.</p> <p>If this parameter is not defined, SYSOUT class is not included as a selection filter on SAPI requests. If this parameter is defined, only jobs written to one of the specified SYSOUT classes are selected by the Platform Server.</p> <p>It is very important that CLASS be the only selection criteria defined, because all output for that class are selected for transmission to a remote node. This parameter can be combined with the other SAPI selection filters to limit the output that is selected.</p>
SAPI_DEST	<p>Defines the 8-byte JES destination that is used as a filter for SAPI requests.</p>

Parameter	Description
SAPI_DISP	<p>If this parameter is not defined, destination is not used as a filter for SAPI requests. If this parameter is defined, only jobs written to the specified JES destination are selected by the Platform Server.</p> <p>This parameter can be combined with the other SAPI selection filters to limit the output that is selected.</p> <hr/> <p>Defines the disposition of SYSOUT requests that are completed by the Platform Server.</p> <p>Three subparameters are supported. The three subparameters are used to define the actions that can be taken when a transfer request is completed and is about to be removed from the request queue. The three subparameters define respectively the following scenarios:</p> <ul style="list-style-type: none"> • Successful file transfers • Permanent network errors • Transfer errors <p>For each subparameter, you can specify the following values:</p> <ul style="list-style-type: none"> • HOLD: Output is kept and put on the JES HOLD queue. • KEEP: Output is kept and can be selected by the Platform Server later. • DELETE: Output is deleted from the JES queue. <p>The default value for this parameter is DELETE,KEEP,HOLD indicating that SYSOUT data is deleted after a successful transfer, is kept if a transfer ends due to a permanent network error, or is put on hold when a permanent error occurs.</p> <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <p>Note: If a transfer ends with a temporary network error and retries can be made, the SYSOUT data is always kept.</p> </div>
SAPI_FORM	<p>Defines the 4-byte JES form name that is used as a filter for</p>

Parameter	Description
SAPI_REMOTE_PRINTER	<p>SAPI requests.</p> <p>If this field is not defined, form is not used as a filter for SAPI requests. If SAPI_FORM is defined, only jobs specifying the JES form are selected by the Platform Server.</p> <p>This parameter can be combined with the other SAPI selection filters to limit the output that is selected.</p>
SAPI_REMOTE_FILE	<p>Defines a field from 1 to 64 bytes where the remote printer name is defined for all SAPI requests for this node.</p> <p>The same substitutable parameters are supported as are supported by the SAPI_REMOTE_FILE parameter. However, typically the printer name is hard coded and does not use substitutable parameters.</p> <p>For more information on usage of the substitutable parameters, see SAPI_REMOTE_FILE.</p> <p>Defines a field from 1 to 64 bytes where the remote file name is defined for all SAPI requests for this node.</p> <p>This means that the SAPI transfer request is sent to a file on the remote system, instead of to a printer.</p> <p>To make the remote file name unique, the Platform Server supports a variety of substitutable parameters that can be specified within the remote file name. The substitutable parameters all begin with a percent sign (%) and ends with one of these characters: period (.), forward slash (/), back slash (\), brackets(()), or apostrophe (').</p> <p>When a substitutable parameter is followed by a period (.), the period is removed. If you want a period to follow the parameter, you must specify two periods.</p> <p>Substitutable parameters are also terminated at the end of the remote file name. You can specify the start and length of a substitutable parameter by following the parameter with</p>

Parameter	Description
	<p>(SS.LL). SS defines the start byte relative to 1, and LL defines the length of the parameter.</p> <p>The following are the supported substitutable parameters:</p> <ul style="list-style-type: none"> • %JOBN: Name of the SAPI job • %JOBID: Job ID of the SAPI job • %JDATE: Current Julian date in the format YYDDD • %JDATEC: Current Julian date in the format CCYYDDD • %GDATE: Current Gregorian date in the format YYMMDD • %GDATEC: Current Gregorian date in the format CCYYMMDD • %TIME: Current time in the format HHMMSS • %DEST: JES destination of a SAPI request. If a destination is not defined for a request, then default value LOCAL or DEST is used. <p>See the following two examples for your reference:</p> <ul style="list-style-type: none"> • SAPI_REMOTE_ FILE=C:\PRINT\%JOBN..%JOBID..%GDATEC..T%TIME can be translated to SAPI_REMOTE_ FILE=C:\PRINT\ACCTJOB.JOB01000.20011231.T1205001. • SAPI_REMOTE_FILE=C:\PRINT\report%TRN(9.2).txt can be translated to SAPI_REMOTE_FILE=C:\PRINT\report01.txt.
SAPI_TRY	<p>Defines the number of times a SAPI request is tried when a temporary network error occurs.</p> <p>This parameter has no default value. If this parameter is not defined, then the GLOBAL_DEFAULT_TRY_COUNT is used to define the number of times the request can be tried.</p>
SAPI_WRITER	<p>Defines the 8-byte JES writer name that is used as a filter for</p>

Parameter	Description
	<p>SAPI requests.</p> <p>If this parameter is not defined, writer is not used as a filter for SAPI requests. If this parameter is defined, only jobs specifying the JES writer are selected.</p> <p>This parameter can be combined with the other SAPI selection filters to limit the output that is selected.</p>
TLS	<p>Defines whether SSL/TLS is used when communicating with this NODE definition. There are three values for this parameter:</p> <ul style="list-style-type: none"> • YES: TLS is used. A TLS Connection will be established to verify the client and server. After creating a secure TLS connection, a symmetric encryption key will be transmitted from the client to the server and the SS/TLS connection will be broken. • NO: No SSL/TLS connection is created. • TUNNEL: All data transmitted is sent over an encrypted TLS Tunnel. <p>This parameter is used for initiator (outgoing) requests only. TLS/SSL usage on responder (incoming) requests depends on the IP port on which the request is received. This parameter overrides any default values set by the GLOBAL definitions.</p>
TCPCONNECT_ ADAPTER_IPADDR_ IPV4	<p>Defines an IPv4 address that is used in the TCP bind.</p> <p>When initiating a connection request, the Platform Server can bind the connection to a particular IP address. This parameter overrides the adapter IP address.</p> <p>This parameter applies only when the connection is for an IPv4 address or an IP name.</p> <ul style="list-style-type: none"> • If this parameter is defined, TCP binds to the IP address defined by this parameter. • If this parameter is not defined, TCP binds to the adapter

Parameter	Description
	<p data-bbox="565 296 1284 365">IP address specified in the GLOBAL TCPCONNECT_IPADDR_IPV4 parameter.</p> <ul data-bbox="532 394 1279 506" style="list-style-type: none"> <li data-bbox="532 394 1279 506">• If this parameter is not defined and the GLOBAL TCPCONNECT_IPADDR_IPV4 parameter is not defined, TCP binds to all adapters.
TCPCONNECT_ADAPTER_IPADDR_IPV6	<p data-bbox="485 552 1157 579">Defines an IPv6 address that is used in the TCP bind.</p> <p data-bbox="485 615 1310 800">When initiating a connection request, the Platform Server can bind the connection to a particular IPv6 address. This parameter can be used to override the adapter IPv6 address. This parameter applies only when the connection is for an IPv6 address or an IP name.</p> <ul data-bbox="532 835 1310 1178" style="list-style-type: none"> <li data-bbox="532 835 1310 905">• If this parameter is defined, TCP binds to the IPv6 address defined by this parameter. <li data-bbox="532 932 1310 1043">• If this parameter is not defined, TCP binds to the adapter IP address specified in the GLOBAL TCPCONNECT_IPADDR_IPV6 parameter. <li data-bbox="532 1071 1310 1178">• If this parameter is not defined and the GLOBAL TCPCONNECT_IPADDR_IPV6 parameter is not defined, TCP binds to all adapters.
TRAP_TRANSFER_REQUESTS	<p data-bbox="485 1224 1263 1293">Defines whether SNMP traps are sent for file transfers on this node.</p> <p data-bbox="485 1329 695 1356">Valid values are:</p> <ul data-bbox="532 1392 1310 1608" style="list-style-type: none"> <li data-bbox="532 1392 1310 1461">• YES: Traps are sent for all transfers which are completed either successfully or unsuccessfully. <li data-bbox="532 1488 1310 1516">• NO: Traps are not sent. This is the default value. <li data-bbox="532 1543 1310 1608">• ERRORS: Traps are sent when a request is unsuccessful and cannot be retried. <p data-bbox="485 1623 1310 1730">This parameter overrides the GLOBAL TRAP_TRANSFER_REQUESTS parameter. If this parameter is not defined, the GLOBAL TRAP_TRANSFER_REQUESTS is used by default.</p>

Parameter	Description
WAIT_FOR_SESSION	<p>Defines whether users on the mainframe can queue up transactions for remote systems that are unconnected and have Platform Server for z/OS wait for that system to connect before dispatching the transfers.</p> <p>Valid values are YES and NO. The default value is NO.</p> <p>This parameter works together with the DISCONNECT parameter to define whether the user can disconnect the session with the remote system after all the transaction queued for that system are completed.</p> <p>Note: This parameter cannot be used for TCP node definitions.</p>

Sample Node Definitions

The Platform Server provides sample node definitions in the product CONFIG library.

See the following sample of TCP node definition:

```

*****
* Sample Connection Parameters for TCP/IP : TCPNODE *
*****
IPADDR=127.127.127.1
*IPNAME=PRIMARY.SERVER.COM
IIPORT=46464
*
*****
* Node Description *
*****
DESCRIPTION=Sample TCP Node from TIBCO
*****
* Transfer Protocol Information *
*****
APPLICATION=FUSION
*****
* Backup Connection Information (Maximum 5 Backup Definitions allowed)*
*****
BACKUPIPNAME=BACKUP.SERVER.COM
BACKUIIPORT=46464

```

```

*
BACKUIPADDR=127.127.127.2
BACKUIPPORT=47000
*
BACKUPLUNAME=NTLU0001
BACKUPMODENAME=FUSN32K
*
*****
* Security Information *
*****

RESPONDER_PROFILE=NO
ACCEPT_VERIFIED_USER=NO
SSL=NO
COMMAND_CENTER_SUPPORT=NONE
DEFAULT_CHECKPOINT=NO
DEFAULT_COMPRESS=NO
DEFAULT_CRLF=NO
ENFORCE_SECURITY_POLICY=NO
DEFAULT_ENCRYPT=NONE
DEFAULT_TYPE=BINARY
*****
* The parameters below are only needed if you wish to limit the *
* number of concurrent transactions *
*****
PARALLEL=YES
CONTENTION_WINNERS=5

```

See the following sample of SNA node definition:

```

*****
* Sample Connection Parameters for SNA : SNANODE *
*****
LU_NAME=RMTL1I3
MODENAME=#BATCH
*****
* Node Description *
*****
DESCRIPTION=Sample SNA Node from TIBCO
*****
* Transfer Protocol Information *
*****
APPLICATION=FUSION
*****
* Backup Connection Information (Maximum 5 Backup Definitions allowed)*
*****

```

```

BACKUPIPNAME=PRIMARY.SERVER.COM
BACKUIPPORT=46464
BACKUIPPORT=46464
*
BACKUIPADDR=127.127.127.2
BACKUIPPORT=47000
*
BACKUPLUNAME=NTLU0001
BACKUPMODENAME=FUSN32K
*
*****
* Security Information *
*****
RESPONDER_PROFILE=NO
ACCEPT_VERIFIED_USER=NO
SSL=NO
COMMAND_CENTER_SUPPORT=NONE
DEFAULT_CHECKPOINT=NO
DEFAULT_COMPRESS=NO
DEFAULT_CRLF=NO
DEFAULT_COMPRESS=NO
DEFAULT_CRLF=NO
ENFORCE_SECURITY_POLICY=NO
DEFAULT_ENCRYPT=NONE
DEFAULT_TYPE=BINARY
*DEFAULT_LOCALCTFILE=NONE
*DEFAULT_REMOTECTFILE=NONE
*DEFAULT_LOGON_DOMAIN=
TRAP_TRANSFER_REQUESTS=NO
*****
* Concurrent Session Information *
*****
DEFAULT_NODECLASS=0
PARALLEL=YES
CONTENTION_LOSERS=5
CONTENTION_WINNERS=5
DEFAULT_COSNAME=COSDFLT

```

See the following sample of SAPI node definition:

```

*****
* SAPINODE *
* ~~~~~ *
* Sample Connection Parameters for TCP/IP using the Sysout Spool *
* Interface *
*****

```

```

IPADDR=127.127.127.1
*IPNAME=PRIMARY.SERVER.COM
IPPORT=46464
*****
* Transfer Protocol Information *
*****
APPLICATION=FUSION
*****
* MFT Platform Server Sysout Spool Interface (SAPI) specific information
  *
*****
SAPI_CLASS=up to 8 classes
SAPI_DEST=jes destination
SAPI_FORM=sysout form name
SAPI_WRITER=jes writer
*****
* MFT Platform Server SAPI Disposition
  *
*****
SAPI_DISP=(DELETE,KEEP,HOLD)
*****
* MFT Platform Server SAPI Remote File/Printer Names
  *
*****
*SAPI_REMOTE_PRINTER=
SAPI_REMOTE_FILE=c:\print\%TRN..txt
*****
* MFT Platform Server SAPI Transfer parameters
  *
*****
SAPI_TRY=0
DEFAULT_CHECKPOINT=NO
DEFAULT_COMPRESS=NO
DEFAULT_CRLF=NO
ENFORCE_SECURITY_POLICY=NO
DEFAULT_ENCRYPT=NONE
DEFAULT_TYPE=BINARY
TRAP_TRANSFER_REQUESTS=NO
*****
* MFT Platform Server SAPI Security parameters
  *
*****
DEFAULT_USERID=remoteuser
DEFAULT_PASSWORD=remotepass
DEFAULT_LOGON_DOMAIN=NTDOMAINNAME
*****

```

```

* Backup Connection Information (Maximum 5 Backup Definitions allowed)*
*****
BACKUIPNAME=BACKUP.SERVER.COM
BACKUIPPORT=46464
*
BACKUIPADDR=127.127.127.2
BACKUIPPORT=47000
*
BACKUPLUNAME=NTLU0001
BACKUPMODENAME=FUSN32K
*****
* Security Information *
*****
RESPONDER_PROFILE=NO
ACCEPT_VERIFIED_USER=NO
SSL=NO
COMMAND_CENTER_SUPPORT=NONE
*****
* Concurrent Session Information *
*****
PARALLEL=YES
CONTENTION_WINNERS=5

```

Distribution Lists

You can use the Platform Server batch client to specify a list of remote nodes to which you want to send a dataset.

The list capability works similarly to the way the remote nodes are defined. Each remote system list is defined in the CONFIG member the same way that an individual remote node is defined. A separate member must be created to include a list of remote systems.

By defining a list, a user of the batch client can reference it and the transfer can be sent to all of the remote systems that are included in the list.

To define a list, you can use the following parameter. This parameter specifies that this member is a list and not an individual node.

```
TYPE=LIST
```

You can specify up to 2048 NODE, IPADDR or IPNAME definitions. You can also override IPPORT at any time in the list by specifying the IPPORT statement.

See the following list definition for your reference:

```

TYPE=LIST
NODE=CHICAGO1,NEWYORK2,DENVER4, *Points to node definition.*
IPADDR=192.192.1.200,192.192.1.201,192.192.1.202 *Uses GLOBAL
TCIIPPORT*
IPNAME=acct.companya.com,payroll.companya.com, *Uses GLOBAL TCIIPPORT*
IPPORT=2500
IPADDR=192.200.1.200,192.200.1.201,192.200.1.202 *Uses 2500 as port*
IPNAME=acct.companyb.com,payroll.companyb.com, *Uses 2500 as port*
IPPORT=46464
IPADDR=192.220.1.200,192.220.1.201,192.220.1.202 *Uses 46464 as port*
IPNAME=acct.companyc.com,payroll.companyc.com, *Uses 46464 as port*
NODE=BOSTON,WASHDC,ORLANDO *Points to node definition*

```

See the following sample of a remote nodes list in member DISTRIB:

```

*****
* THIS IS WHERE YOU WILL SPECIFY A LIST OF NODES THAT DATA WILL *
* BE DISTRIBUTED TO. *
*****
TYPE=LIST
NODE=NTLU001,WIN95LU7
NODE=NTLU002
IPADDR=192.192.1.200,192.192.1.201,192.192.1.202
IPNAME=acct.companya.com,payroll.companya.com
IPPORT=2500
IPADDR=192.200.1.200,192.200.1.201,192.200.1.202
IPNAME=acct.companyb.com,payroll.companyb.com

```

Class of Service (COS) Definitions

COS entries are defined in the CONFIG DD statement in the same library that defines the node entries.

In the Platform Server COS definitions, you can define the following functions:

- The buffer sizes that are passed between the Platform Server and VTAM/TCP.
- The time of day that transfers can be run.
- For TCP requests, you can throttle transfer of data to or from a node. In other words, you can set a maximum number of bytes per second that can be sent to or received from a node.

Note: This is only useful for TCP transfers because SNA already has a Class of Service feature that you can use to prioritize data traffic.

COS entries can be enabled at startup if they are contained in the node startup list. Additionally, they can be enabled through the `ENABLE` operator command and disabled through the `DISABLE` operator command.

If a COS entry is defined for a file transfer, and the COS entry is not disabled, the transfer can run as if no COS entry is defined.

Note: If transfers are assigned a COS entry, they are not assigned a priority. The priority defined within the COS entry changes when the time of the transfer changes. A transfer can be assigned a priority of 9 when running early in the morning. If the transfer runs long enough, the priority might be changed to another value based on the parameters in the COS entry.

Additional node parameters are required to perform TCP throttling of data sent and received. For more information, see [BPS_INIT](#), [BPS_RECEV](#), [BPS_RESP](#), [BPS_SEND](#), [BPS_TOTAL](#), [DEFAULT_COSNAME](#).

The COS definitions can be divided to two sections. The first section describes the required parameters and the parameters that define the buffer sizes that are passed between the Platform Server and TCP/VTAM.

See the following table for the parameters in the first section:

Parameter	Description
TYPE	Distinguishes between NODE entries and COS entries. This is a required parameter. You must set the value as COS.
SBUFSIZE	Defines the buffers size in bytes that the Platform Server uses when sending data to VTAM or TCP. This is an optional parameter. Valid values are from 1490 to 9999999. The default value is 131072, which is a proper value for most transfers. By specifying a larger value, you save CPU cycles at the cost of using

Parameter	Description
RBUFSIZE	<p>more storage. You might want to specify a value lower than 32000 if you want to throttle transfers to a very slow level. You will not get much benefit from a buffer size greater than 1024000. Make sure the value defined is less than the system TCP max buff size; otherwise we may use the system default.</p> <p>Defines the buffers size in bytes that the Platform Server uses when receiving data from VTAM or TCP.</p> <p>This is an optional parameter. Valid values are from 1490 to 9999999. The default value is 131072, which is a proper value for most transfers.</p> <p>By specifying a larger value, you save CPU cycles at the cost of using more storage. You might want to specify a value lower than 32000 if you want to throttle transfers to a very slow level. You will not get much benefit from a buffer size greater than 1024000. Make sure the value defined is less than the system TCP max buff size; otherwise we may use the system default.</p>

The second section describes the parameters that define the time period when a file transfer can run, and throttle the transfer of data to and from TCP. These parameters must be on the same line in the documented order. You must define all of the parameters.

You can define up to nine lines of the following parameters for a single COS definition. If you attempt to have more than nine entries, an error message is displayed and the COS definition is not enabled. If you do not want to use some of the parameters, you can use the default values as shown in the following sample:

```
DAYS=YYYYYYYY, STIME=0000, ETIME=2400, PRIORITY=9
```

This field is broken into four parameters. The first three parameters DAYS, STIME and ETIME are filter parameters, while the fourth parameter PRIORITY defines the priority that is used when the current time passes the filters.

See the following table for the descriptions of the four parameters:

Parameter	Description
DAYS	<p>Defines the days of the week.</p> <p>This parameter must have 7 digits that are either Y or N. These digits stand for the days of the week through Sunday to Saturday.</p> <p>When specified as Y, it means that this entry is active for that day of the week, and the Platform Server continues checking the STIME and ETIME parameter.</p>
STIME	<p>Defines the start time to check.</p> <p>It is defined in military time from 0000 to 2400.</p>
ETIME	<p>Defines the end time to check.</p> <p>It is defined in military time from 0000 to 2400 and must be greater than the start time.</p>
PRIORITY	<p>Defines the priority associated with a file transfer if the current day and time passes all the filter parameters: DAYS, STIME and ETIME.</p> <p>Valid values are the following 10 priorities:</p> <ul style="list-style-type: none"> • 0: No priority. The transfer cannot run or be started. If a transfer is running and the priority changes to 0, then terminates the transfer and starts again later when the priority is greater than 1. • 1: Low priority. The transfer cannot be started. If a transfer is running and the priority changes to 1, then the transfer can continue to run. When throttling transfers, sends the data to the node at 25% of the total number of TCP bytes that can be transferred per second. • 2: Low priority. Transfers can be scheduled. When throttling transfers, sends the data to the node at 25% of the total number of TCP bytes that can be transferred per second. • 3: Low priority. Transfers can be scheduled. When throttling transfers, sends the data to the node at 50% of the total

Parameter	Description
	<p>number of TCP bytes that can be transferred per second.</p> <ul style="list-style-type: none"> • 4: Low priority. Transfers can be scheduled. When throttling transfers, sends the data to the node at 75% of the total number of TCP bytes that can be transferred per second. • 5: Medium priority. Transfers can be scheduled. When throttling transfers, sends the data to the node at 100% of the total number of TCP bytes that can be transferred per second. • 6: Medium priority. Transfers can be scheduled. When throttling transfers, sends the data to the node at 133% of the total number of TCP bytes that can be transferred per second. • 7: Medium priority. Transfers can be scheduled. When throttling transfers, sends the data to the node at 166% of the total number of TCP bytes that can be transferred per second. • 8: High priority. Transfers can be scheduled. When throttling transfers, sends the data to the node at 200% of the total number of TCP bytes that can be transferred per second. • 9: No maximum priority. Transfers can be scheduled. No throttling of transfers is performed.

See the following copy of the COSDFLT entry that is provided in the Platform Server SAMPLIB library:

```

*****
* SAMPLE COS (CLASS OF SERVICE) ENTRY: COSDFLT
*****
TYPE=COS                               Defines that this is a COS entry
SBUFSIZE=32000                          Sets the TCP SEND Buffer Size
RBUFSIZE=32000                          Sets the TCP RECEIVE Buffer Size
*****
* Parameters that limit file transfer speed
* Up to 9 lines can be defined per class of service
* Parameters must be on the same line in the defined order:
* DAYS=xxxxxxx (Y or N)                 Days of the week starting with Sunday
* STIME=xxxx (hhmm)                     Start time (0000-2359)
* ETIME=xxxx (hhmm)                     End time (0000-2359)
* PRIORITY=x (0-9)                      Priority
*****

```

```

*   The following line sets high priority 7 days a week 24 hours/day
DAYS=YNNNNNY,STIME=0000,ETIME=2359,PRIORITY=9    High priority 7x24
*   The following lines sets high priority all day Sunday/Saturday
*   They are currently commented out
*DAYS=YNNNNNY,STIME=0000,ETIME=2359,PRIORITY=9    High Sun/Sat 7x24
*   The following lines sets different priorities for Monday-Friday
DAYS=NYYYYYN,STIME=0000,ETIME=0700,PRIORITY=9    High Mid-7AM
DAYS=NYYYYYN,STIME=0700,ETIME=0900,PRIORITY=5    Medium 7-9AM
DAYS=NYYYYYN,STIME=0900,ETIME=1700,PRIORITY=0    Do not Run 9-5pm
DAYS=NYYYYYN,STIME=1700,ETIME=1900,PRIORITY=5    Medium 5-7pm
DAYS=NYYYYYN,STIME=1900,ETIME=2359,PRIORITY=9    High 7-12mid

```

This COS definition sets the send and receive buffer sizes to 32000. The priority is set to 9 (high) for all transfers running on Saturday and Sunday. The last five lines set different priorities for transfers running on Monday through Friday.

- For transfers running between midnight and 7 a.m., the priority is set to 9.
- For transfers running between 7 a.m. and 9 a.m., the priority is set to 5.
- For transfers running between 9 a.m. and 5 p.m., the priority is set to 0.
- For transfers running between 5 p.m. and 7 p.m., the priority is set to 5.
- For transfers running between 7 p.m. and midnight, the priority is set to 9.

Data Throttling

By specifying a priority between 1 and 8, you can throttle the speed at which data is sent to the remote node.



Note:

- This feature is only supported for TCP transfers. It is not supported for SNA transfers.
- This feature is available only when a node definition is used and that node definition defines one of these BPS parameters: [BPS_INIT](#), [BPS_RECEV](#), [BPS_RESP](#), [BPS_SEND](#), [BPS_TOTAL](#), [DEFAULT_COSNAME](#).

Priorities from 1 to 8 perform data throttling of all send and receive requests. The Platform Server keeps track of the number of concurrent requests with a node, and extracts all of the defined BPS parameters. Based on the defined priority, the Platform Server multiplies the BPS parameters by the priority, and then divides the product by the number of concurrent tasks to get the speed at which the Platform Server transmits data.

For example, if you use the [sample COS entry](#), and have a node entry called NODEABC with the following BPS information:

```
BPS_TOTAL=56000
BPS_SEND=32000
BPS_RECEIVE=40000
```

If a transfer is running and sending data to node NODEABC. The node definition indicates that you can send 32000 bytes per second, and receive 40000 bytes per second, but you can send and receive only a total of 56000 bytes per second.

If there are currently 5 total tasks running: 3 send and 2 receive tasks. The day is Monday and the time is 6:55 p.m. (18:55 in military time). As you can see from the [sample COS entry](#), when running on Monday at 1855, the priority is set at 5. Therefore, you take the BPS fields in the node definition and multiply them by 100%. Then divide the BPS values by the number of transfers, and get the maximum number of bytes that you can transmit.

```
BPS_TOTAL=56000*100%/5=11200BPS
BPS_SEND=32000*100%/3=10666BPS
BPS_RECV=48000*100%/2=24000BPS
```

Because you are sending data, the smaller of the BPS_SEND and BPS_TOTAL numbers is used. In this case, you can send a maximum of 10666BPS. If the number of transfers changes, the Platform Server recomputes the maximum BPS on each packet sent.

If the transfer runs long enough and that the current time becomes 7 p.m. (1900 in military time). As the [sample COS entry](#) shows, the priority becomes 9 for this entry. A priority of 9 indicates an unlimited transfer speed. Therefore, the Platform Server transmits data as fast as possible to node NODEABC.

Translation Table (XLATE) Definitions

With the XLATE feature, you can define and use translation tables.

The translation tables can be used to translate data between EBCDIC systems such as z/OS and AS/400, or between EBCDIC and ASCII systems.

Two samples are included in the Platform Server CONFIG DD: ASCIIXL and EBCDICXL.

Translation is only performed when TYPE=TEXT is defined. They must be enabled at startup or through the ENABLE operator command. You can refresh translation tables by using the REFRESH command, but you cannot disable them.

The XLATE feature is enabled in one of the following ways:

- By specifying the LOCALCTFILE parameter on the batch interface.
- By specifying the DEFAULT_LOCALCTFILE parameter on the node definition.
- If the remote system initiates a transfer and specifies the REMOTECTFILE parameter.

i Note: If you specify that you want to use a translation table that is not enabled, the transfer terminates with errors.

See the following list of parameters that are required for the translation table configuration. Comments are defined with an asterisk (*) in column 1.

- TYPE=XLATE is a required parameter. It is used to distinguish between NODE, COS, and XLATE entries. This must be the first parameter defined.
- Following the TYPE=XLATE parameter, you must add 32 lines of translation tables. Each line must contain 32 hex digits from 0-F. Therefore, each line contains 16 bytes of translation table. The first 16 lines are used when you are sending data, and lines 17 through 32 are used when you are receiving data.

See the following sample EBCDIC to ASCII translation table:

```

TYPE=XLATE
* SEND Table (EBCDIC to ASCII)          -----Displacements-----
* Translate table for SEND Transfers    -HEX Bytes--      Decimal Bytes
002E2E2E2E2E2E2E2E0A2E2E0D2E2E      * 0x00 to 0x0F      0   to 15
2E2E2E2E2E0A2E2E2E2E2E2E2E2E2E2E    * 0x10 to 0x1F      16  to 31
2E2E2E2E2E2E2E2E2E2E2E2E2E2E2E2E    * 0x20 to 0x2F      32  to 47
2E2E2E2E2E2E2E2E2E2E2E2E2E2E2E2E    * 0x30 to 0x3F      48  to 63
202E2E2E2E2E2E2E2E2E2E632E3C282B7C   * 0x40 to 0x4F      64  to 79
262E2E2E2E2E2E2E2E2E2E2E21242A293B5E  * 0x50 to 0x5F      80  to 95
2D2F2E2E2E2E2E2E2E2E2E7C2C255F3E3F   * 0x60 to 0x6F      96  to 111
2E2E2E2E2E2E2E2E2E2E603A2340273D22    * 0x70 to 0x7F     112  to 127
2E6162636465666768692E2E2E2E2E2E     * 0x80 to 0x8F     128  to 143
2E6A6B6C6D6E6F7071722E2E2E2E2E2E     * 0x90 to 0x9F     144  to 159
2E7E737475767778797A2E2E2E5B2E2E     * 0xA0 to 0xAF     160  to 175
2E2E2E2E2E2E2E2E2E2E2E2E2E5D2E2E     * 0xB0 to 0xBF     176  to 191
7B41424344445464748492E2E2E2E2E2E     * 0xC0 to 0xCF     192  to 207
7D4A4B4C4D4E4F5051522E2E2E2E2E2E     * 0xD0 to 0xDF     208  to 223
5C00535455565758595A2E2E2E2E2E2E     * 0xE0 to 0xEF     224  to 239
303132333435363738392E2E2E2E2E2E     * 0xF0 to 0xFF     240  to 255
* RECEIVE Table (ASCII to EBCDIC)      -----Displacements-----
* Translate table for RECEIVE Transfers -HEX Bytes--      Decimal Bytes
00010203372D2E2F16050A0B0C0D0E0F      * 0x00 to 0x0F      0   to 15

```

101112133C3D322618193F27221D351F	* 0x10 to 0x1F	16 to 31
405A7F7B5B6C507D4D5D5C4E6B604B61	* 0x20 to 0x2F	32 to 47
F0F1F2F3F4F5F6F7F8F97A5E4C7E6E6F	* 0x30 to 0x3F	48 to 63
7CC1C2C3C4C5C6C7C8C9D1D2D3D4D5D6	* 0x40 to 0x4F	64 to 79
D7D8D9E2E3E4E5E6E7E8E9ADE0BD5F6D	* 0x50 to 0x5F	80 to 95
79818283848586878889919293949596	* 0x60 to 0x6F	96 to 111
979899A2A3A4A5A6A7A8A9C06AD0A107	* 0x70 to 0x7F	112 to 127
00000000000000000000000000000000	* 0x80 to 0x8F	128 to 143
00000000000000000000000000000000	* 0x90 to 0x9F	144 to 159
00000000000000000000000000000000	* 0xA0 to 0xAF	160 to 175
00000000000000000000000000000000	* 0xB0 to 0xBF	176 to 191
00000000000000000000000000000000	* 0xC0 to 0xCF	192 to 207
00000000000000000000000000000000	* 0xD0 to 0xDF	208 to 223
00000000000000000000000000000000	* 0xE0 to 0xEF	224 to 239
00000000000000000000000000000000	* 0xF0 to 0xFF	240 to 255

Using the SEND Table for EBCDIC to ASCII Translation

To convert a character from EBCDIC to ASCII, you can simply get the HEX representation of that character in EBCDIC and ASCII.

Procedure

1. Take the HEX representation of the EBCDIC character.
2. Look that many bytes into the SEND table, and place the HEX representation of the ASCII character in the correct entry in the table.

i Note: The displacements in the table start at 0, and there are two characters in the table for each actual byte of data.

For example: number 1 in EBCDIC is X'F1', and in ASCII is X'31'.

In the SEND table, if you go to displacement F1, you can go to the last line in the SEND table, and the second pair in the line, you can get the character 31, which is the ASCII representation of the EBCDIC character 1.

Using the RECEIVE Table for ASCII to EBCDIC Translation

To convert a character from ASCII to EBCDIC, you can simply get the HEX representation of that character in ASCII and EBCDIC.

Procedure

1. Take the ASCII HEX representation of the ASCII character.
2. Look that many bytes into the RECEIVE table, and place the HEX representation of the EBCDIC character in the correct entry in the table.

Note: The displacements in the table start at 0, and there are two characters in the table for each actual byte of data.

For example: number 1 in ASCII is X'31', and in EBCDIC is X'F1'.

In the RECEIVE table, if you go to displacement 31, you can go to the fourth line in the RECEIVE table, and the second pair in the line, you can get the character F1, which is the EBCDIC representation of the ASCII character 1.

TCP Translation Table (TCPXLATE) Definitions

In addition to supporting its own internal translation tables, the Platform Server supports the Single Byte Character Set (SBCS) and Double Byte Character Set (DBCS) translation tables supplied by IBM TCPIP.

To activate these translation tables, you must define and enable a TCPXLATE member within the CONFIG DD statement. Translation tables must be enabled at startup or through the ENABLE operator command. You can refresh translation tables by using the REFRESH command, but you cannot disable them.

Translation is defined by the LOCALCTFILE parameter on the batch interface, or by the DEFAULT_LOCALCTFILE parameter on the node definition. Translation is only performed when TYPE=TEXT is defined.

See the following table for parameters that are required by the translation table configuration. Comments are defined with an asterisk (*) in column 1.


Parameter	Description
TYPE	Distinguishes between NODE, COS, XLATE, and TCPXLATE entries. This is a required parameter. You must set the value as TCPXLATE to indicate that you want to configure the Platform Server to use the TCP SBCS and DBCS translation tables. It must be the first parameter defined.

Parameter	Description
SBCS	<p>Defines the SBCS member that is used when performing translation.</p> <p>The member defined must be a valid member of the TCPIP.SEZATCPX library defined by the TCPSBCS DD statement. This is a required parameter and it has no default value.</p>
DBCS	<p>Defines the DBCS ddname and the relative number of the DBCS translation tables.</p> <div data-bbox="526 617 1409 716" style="background-color: #f0f0f0; padding: 5px;"> <p>Note: The DBCS translation files can contain more than one DBCS translation table.</p> </div> <p>This is a required parameter and it has no default value. You must define a valid value in this format: DBCS <i>ddname</i>,<i>relative_number</i>.</p> <p>See the following list of the translation tables that are supplied by TCPIP and can be set using this parameter.</p> <ul style="list-style-type: none"> • TCPCHBIN,1 Traditional Chinese • TCPCHBIN,2 Big5 Chinese • TCPSCBIN,1 Simplified Chinese • TCPHGBIN,1 Korean • TCPHGBIN,2 Korean Hangeul • TCPKJBIN,1 JIS Japanese • TCPKJBIN,4 EUK Japanese
SOSI	<p>Defines the type of ShiftOutShiftIn characters that are used by the remote system.</p> <p>This is a required parameter. It has no default value. The parameter applies only when DBCS has been coded. It is not valid when SBCS has been coded.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • SPACE: An ASCII space character (0x20) is the SOSI character. Whether data is converted depends on the character following 0x20. If the following character is a valid DBCS character, then

Parameter	Description
	<p data-bbox="607 296 1406 405">DBCS translation is performed. If the following character is not a valid DBCS character, then standard SBCS conversion is performed on this character.</p> <ul data-bbox="574 434 1395 737" style="list-style-type: none"> <li data-bbox="574 434 1395 464">• ASCII: The SOSI characters are the hex values 0x1e and 0x1f. <li data-bbox="574 491 1395 520">• EBCDIC: The SOSI characters are the hex values 0x0e and 0x0f. <li data-bbox="574 548 1395 737">• NONE: No SOSI characters. Whether DBCS conversion is performed depends on the characters. If the characters are valid DBCS characters, then DBCS conversion is performed. If the characters are not valid DBCS characters, then SBCS conversion is performed. <div data-bbox="526 764 1414 1194" style="background-color: #f0f0f0; padding: 10px; margin-top: 10px;"> <p data-bbox="545 785 618 810">Note:</p> <ul data-bbox="594 840 1395 1173" style="list-style-type: none"> <li data-bbox="594 840 1395 1014">• The local SOSI characters are always the EBCDIC SOSI characters. When data is sent to the remote system, only data within the EBCDIC SOSI characters (0x0e and 0x0f) are converted. The EBCDIC SOSI characters are converted to the format specified in the SOSI parameter. <li data-bbox="594 1041 1395 1173">• When DBCS conversion is being performed, the Platform Server turns checkpoint restart off, because checkpoint restart is not supported when DBCS conversion is being performed. </div>

SSL Authorization File

The Platform Server supports an extension to the standard SSL processing. With an SSLAUTH file, the system administrator can determine which certificates are accepted and which are rejected. SSL Authorization checking is performed only on Responder request to validate the identity of the Initiator (that is, Client). SSL Authorization checking is not performed on Initiator requests.

 **Note:** This is supported on all platforms.

SSL Authorization Parameters

The format of the SSLAUTH file is the same on all platforms, but the way that the file is defined is dependent on each platform.

See the following table for the SSL authorization file on each platform.

Platform	File Name	Default Location
z/OS	SSLAUTH	SAMPLIB
Windows	SslAuth	C:\Program Files\TIBCO\FTMSSVR or C:\Program Files\TIBCO\Fusion SslAuth
UNIX	SSLAUTH	/MFT Platform Server/samples

Note: Only if the certificate is accepted after the SSL authorization checking, the authorization file processing is performed.

The authorization file is compared against the certificate that is received by the Platform Server. The authorization file is not used on Platform Server clients. The components of the certificate Distinguished Name (DN) are compared to the parameter in the authorization file to determine if a certificate is accepted or rejected.

On many of the parameters, a generic character is supported. A generic character is defined in a parameter by an asterisk (*). When a generic character is defined, all characters from that point on are assumed to be a match.

If no authorization file is defined, or a match is not found in the authorization file, the request is accepted. If you want to reject all requests unless defined by the authorization file, then you must insert the following statement as the last entry in the authorization file:

REVOKE

The following two request types are supported within the authorization file:

- ACCEPT: Accepts an SSL request.
- REVOKE | REJECT: Does not accept an SSL request.

All of these requests accept a variety of parameters. If a parameter is not defined, then it is assumed that the parameter is a match. Parameters can be defined on a single line and can be continued over multiple lines. If the input record ends with a comma (,), then the input record is continued on the next record. All parameter data is case sensitive. You have to be careful when entering the values using mixed case fields.

See the following table for parameters you can define in the authorization file.



Note: These parameters must be defined in upper case.

Parameter	Description
/CN	Defines the common name defined in the certificate. This is usually the name of the person who is requesting the certificate. Generic entries are supported.
/OU	Defines the organization unit defined in the certificate. This is also known as the department. Generic entries are supported.
/O	Defines the organization defined in the certificate. This is also known as the company. Generic entries are supported.
/L	Defines the locality defined in the certificate. This is also known as the city. Generic entries are supported.
/ST	Defines the state/province defined in the certificate. Generic entries are supported.
/C	Defines the country defined in the certificate. Generic entries are supported.
/SN	Defines the serial number defined in the certificate. Generic entries are not supported.

Parameter	Description
/SDATE	<p>Defines the start date for the certificate in the format <i>ccyymmdd</i>.</p> <p>Generic entries are not supported.</p> <p>The start date is compared against the date when the transfer request is received by the Platform Server. If the start date is before the current date, then SSLAUTH processing checks the next parameter. If the start date is after the current date, then the transfer request is terminated and an error is sent to the remote system.</p>
/STIME	<p>Defines the start time for the certificate in the format <i>hhmm</i>.</p> <p>Generic entries are not supported.</p> <p>The start time is only checked if the SDATE parameter exactly matches the current date. The start time is compared against the time when the transfer request is received by the Platform Server. If the start time is before the current time, then SSLAUTH processing checks the next parameter. If the start time is after the current time, then the transfer request is terminated and an error is sent to the remote system.</p>
/EDATE	<p>Defines the end date for the certificate in the format <i>ccyymmdd</i>.</p> <p>Generic entries are not supported.</p> <p>The end date is compared against the date when the transfer request is received by the Platform Server. If the end date is after the current date, then SSLAUTH processing checks the next parameter. If the end date is before the current date, then the transfer request is terminated and an error is sent to the remote system.</p>
/ETIME	<p>Defines the end time for the certificate in the format <i>hhmm</i>.</p> <p>Generic entries are not supported.</p> <p>The end time is only checked if the EDATE parameter exactly matches the current date. The end time is compared against the time when the transfer request is received by the Platform Server. If the end time is after the current time, then SSLAUTH processing checks the next parameter. If the end time is before the current time, then the</p>

Parameter	Description
	transfer request is terminated and an error is sent to the remote system.
/USER	<p>Defines a user ID that is used when an SSL certificate is accepted.</p> <p>This parameter is only supported by the z/OS system, and it should only be used by the system administrator.</p> <p>This user ID overrides the user ID associated with the file transfer. Using this option, the remote user does not have to have any knowledge of a user ID and a password on the z/OS system.</p>

See the following examples of authorization file processing:

```
Accept /OU=Marketing/O=TIBCO
revoke
```

- The Platform Server accepts all certificates defined with an organization of TIBCO and an organization unit of Marketing, while rejects all other certificates.

```
REVOKE /SN=987654
REVOKE /SN=12:34:56
ACCEPT
```

- The Platform Server rejects any certificates with a serial number of 987654 or 123456, and accepts all other certificates.

```
Accept /OU=ACCT*/O=ACME
revoke
```

- The Platform Server accepts all certificates defined with an organization of ACME and an organization Unit starting with ACCT, and rejects all other certificates.

```
Accept /CN=Joe*,
      /L=New York,
      /ST=NY,
      /C=US,
      /OU=Dept1,
      /O=ACME,
```

```

        /SDATE=20081201,
        /EDATE=20091130
revoke

```

The Platform Server accepts all certificates that match the information defined by the /CN, /L, /ST, /C, /OU, and /O parameters. The certificate is valid from December 1, 2008 until November 30, 2009. If the certificate is received before December 1, 2008 or after November 30, 2009, the request is rejected. All other certificates not matching these criteria are rejected.

Setting Up SSL Authorization File on z/OS

By adding the SSLAUTH DD statement to the Platform Server startup JCL, you can set up the SSL authorization file to be used with the z/OS platform.

The dataset must point to a file containing the SSLAUTH parameters.

On a z/OS system, you can define an SSLAUTH file in the following two ways:

- If the LRECL of the SSLAUTH file is 80, only columns 1 through 72 are validated.
- If the LRECL of the SSLAUTH file is not 80, all columns of the input record are validated.

User Profiles

To address the security issue in implementing a file transfer environment, the Platform Server has implemented a secure environment where both user IDs and passwords are required for every file transfer.

By implementing the user profile facility, the user must submit jobs to the internal reader with both user ID and password defined. In some cases, this control information must be stored in parameter files on a disk, which can present a security exposure in some organizations. The Platform Server user profile facility resolves these issues by giving a user the ability to store a user ID and encrypted password on a z/OS dataspace, and using that user ID and password in all future transfer requests.

Another requirement for users is to remotely access information on the z/OS mainframe. The users might be located in remote locations, subsidiaries, or completely different companies. These users typically initiate a file transfer to send or receive data to or from the Platform Server on z/OS.

Because the Platform Server uses the security implemented by the z/OS operating system, a user has to supply a user ID and password. The Platform Server then passes this information to RACF. Therefore, every user who uses the Platform Server requires a RACF user ID and a password. This means that thousands of users require RACF user IDs when the only goal they have for access is file transfers.

The Platform Server can alleviate this requirement with the responder profile feature. The Platform Server administrator can configure the user profiles in such a way that when receiving a request, the Platform Server compares the user ID and password against its profile database. If a match is made on user ID, password and node name, the Platform Server uses the RACF user ID associated with the profile. This feature can greatly reduce the number of RACF user IDs required by the system.

i Note: Only the Platform Server administrator can control this feature. By default, the feature is disabled. But, it can be enabled on a node-by-node basis or on the entire system.

To use the Platform Server user profile facility, you must define the Platform Server user profile VSAM file as defined in the [Defining the User Profile VSAM dataset](#), and define the PROFILE DD statement in the Platform Server JCL as described in [The Startup JCL](#). If the PROFILE DD statement is not defined, a message is displayed and the Platform Server user profile facility is disabled.

The Platform Server user profile facility has the following three main functions:

- A user can define a user ID/password combination for a particular node. This information is stored in the Platform Server user profile dataspace. Whenever the user sends a transfer to that node, the Platform Server substitutes the predefined user ID/password. Therefore, the user ID/password does not have to be defined in the file transfer JCL.
- The Platform Server administrator can define user ID/password combinations for any user who might transfer data to a particular node. This provides all of the advantages of the previous function as well as another important feature: if an administrator predefines a user ID/password combination for a user to transfer data to a node, then the user that submits the file transfer request does not have to know the remote password or even the remote user ID of the target computer. The mainframe users can perform file transfer functions to a computer without knowing the user IDs and passwords of that remote system. They can send files to the system but cannot logon to the system because they do not know the user ID and password of the remote system.

i Note: The mainframe administrator still has control over who can submit jobs to a remote node, and the security administrator still has control over what files and directories the mainframe user can access.

- The number of RACF user IDs that are required on the system can be reduced. Additionally, users can perform file transfers without being given access to a RACF user ID that might give them other access into the system.

i Note: The user profile facility is optional. You can continue to use the batch, REXX and ISPF interface parameters to define the user ID/password for each individual file transfer. You can even use a combination of the user ID/password and user profile.

The Platform Server user profiles have two types of usage:

- Initiator profiles: When a request is initiated by the local Platform Server. Only initiator profiles are used by the initiator profile facility.
- Responder profiles: When a request is initiated by a remote Platform Server and is processed on the local Platform Server. Only responder profiles are used by the responder profile facility.

A profile is considered to be an initiator profile unless the RESPONDER=YES parameter is defined.

Using Initiator User Profile Facility

Using the Platform Server user profile facility requires only two steps.

Procedure

1. Define a remote user ID/password combination for a particular node.
For more information, see [Managing User Profiles](#).
2. In the Platform Server batch interface, specify the remote user ID as *PROFILE using the following command.

```
REMOTE_USER=*PROFILE
```

Optionally, in the Platform Server ISPF panels, specify the remote user ID as

*PROFILE.

When the Platform Server started task detects a remote user ID defined as *PROFILE, it scans the user profile dataspace for a match on the following information:

- Local user ID of the user that submitted the transfer request
- Node name, IP address, IP name, or list name

If a match is found, the Platform Server substitutes the user ID/password defined in the user profile for the user ID/password defined in file transfer queue record.

If no match is found, the Platform Server terminates the queue record with a return code, and a message is returned to you that submitted the file transfer request.



Note: *PROFILE is the default profile for the REXX interface when the REMOTE_USER (RUSER) parameter is not defined.

Using Responder User Profile Facility

Using the Platform Server user profile responder facility requires only two steps.

Procedure

1. Define a remote user ID/password combination for a particular node.

You must also define the local RACF user ID that is used when a match is found. For more information, see [Managing User Profiles](#).

2. Specify in the Platform Server GLOBAL or node parameter: RESPONDER_PROFILE=YES or RESPONDER_PROFILE=DUAL.

When the Platform Server accepts a request, it first determines if there is an enabled node definition for the system that initiated the request. Then it scans the user profile dataspace for a match on the following information:

- User ID of the remote user that submitted the transfer request.
- Password entered by the remote user that submitted the request (this field is optional).
- Node name, IP address, or IP name.

If a match is found, the Platform Server substitutes the local user defined in the user profile for the user ID/password defined by the user that initiated the request. If a

match is not found, the Platform Server terminates the request with a return code, and a message is returned to the user that submitted the file transfer request.

i **Note:** If `RESPONDER_PROFILE=DUAL` is specified, the request is not terminated even if a match is not found. The processing continues and the request is checked for trusted user ID (`ACCEPT_VERIFIED_USER`), and if necessary, the RACF or ACF2 or Top Secret user ID is validated.

When a user on a remote system submits a request to the Platform Server, the user enters a remote user ID and a remote password. This user ID/password is compared against the profile `REMOTE_USER` and `REMOTE_PASS` parameters for a match. When a match is found, and the node name matches as well, the profile `LOCAL_USER` becomes the RACF user that is associated with the transfer request.

User Profile Facility Security

The Platform Server user profile facility is secured in two ways. See the following two ways to secure the user profile facility:

- Typical users can add, delete, or list user profile records only for themselves.
- Only administrators can add user profile records for other user IDs. A Platform Server user profile administrator is a user that has RACF or ACF2 or Top Secret control privilege for the facility defined by the `BOSSID` parameter. If the `BOSSID` parameter is set to `ANY`, then all users can add, delete, or list user profile records for any user ID in the system.

i **Note:** To give a Platform Server profile administrator rights to a user, you must give them `CONTROL` access for RACF and CA-Top Secret as well as `DELETE` access for CA-ACF2.

For more information on defining the facility class for the user profiles, see [The RACF Security Interface](#), [The CA-ACF2 Security Interface](#), and [The Top Secret Security Interface](#).

Managing User Profiles

You can execute the Platform Server user profiles program `FUSPROF` to add, delete, or list user profiles.

User Profile Sample JCL

The USERPROF member of the Platform Server JCL library shows the JCL required to execute the Platform Server user profile program to add, delete, or list user profiles.

See the following sample user profile JCL:

```
//USERPROF JOB 555,'JOHN USER',MSGCLASS=X,REGION=1M
//*
//*****
//*      This is a sample of how the MFT Platform Server administrator
can
//*      ADD, DELETE, and LIST MFT Platform Server User Profiles.

//*****
//*
//FUSPROF  EXEC PGM=FUSPROF,
//          PARM='SERVER=FUSION'    <<== Point to MFT Platform Server

//*
//STEPLIB DD  DISP=SHR,DSN=FUSION.LOADLIB
//SYSUDUMP DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
*****
TYPE=ADD                                ADD profile for the local user
      REMOTE_USER=RUSER1
      REMOTE_PASS=password
      IPNAME=NTSERV.YOURCOMPANY.COM
*
TYPE=DELETE                             DELETE profile for a remote user
      NODE=NODE1
      REMOTE_USER=RMTUSER1
      RESPONDER=YES
*
TYPE=LIST                                LIST all profiles for IPNAME
      IPNAME=NTSERV.YOURCOMPANY.COM
```

See the following table for the required JCL statements.

Statement	Description
EXEC PGM=FUSPROF,PARM=SERVER=XXXXX	Defines the program that must be executed to run the Platform Server user profile

Statement	Description
	<p>interface.</p> <p>The program name must be FUSPROF. The parameter field is used to define the name of the Platform Server started task.</p> <p>The only required parameter is SERVER=, which defines the name of the Platform Server STC.</p>
STEPLIB	<p>Defines the library that contains the Platform Server load modules.</p> <p>This statement must be included to identify the Platform Server load library.</p>
SYSPRINT	<p>Defines the output report that shows what parameters are used for file transfer.</p> <p>If the file is successfully queued, then this report reveals what transaction number is assigned to the job.</p>
SYSIN	<p>Defines the input file that shows the user what file to transfer and where to send any other parameter that governs a file transfer activity.</p>

User Profile Statement Parameters

The user profile statement parameters are defined to the SYSIN DD statement.

See the following table for the valid parameters:

Parameter	Description
TYPE	<p>Defines the type of action you want to perform with the user profile.</p>

Parameter	Description
REMOTE_USER	<p data-bbox="558 296 1247 323">Valid values are ADD, CREATE,DELETE, LIST, or REPLACE.</p> <p data-bbox="558 373 1091 401">Defines the user ID on the remote system.</p> <p data-bbox="558 436 766 464">Valid values are:</p> <ul data-bbox="607 499 1409 764" style="list-style-type: none"> <li data-bbox="607 499 1409 764">• *ALLUSER: an administrator can specify a default user ID for all requests from users that are not specifically defined on the system. An administrator can define specific user ID/passwords for a node for users with high security authorization, while set a default user ID/password to be used for all other users with less security authorization defined. <div data-bbox="643 793 1414 863" style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p data-bbox="659 814 1328 842">Note: This option is only used by responder profiles.</p> </div> <ul data-bbox="607 890 1409 957" style="list-style-type: none"> <li data-bbox="607 890 1409 957">• <i>userid</i>: the ID must be specified in the format of <i>domain\user ID</i> or <i>domain/user ID</i>.
REMOTE_PASSWORD	<p data-bbox="558 1003 1409 1073">Defines the password on remote system for an initiator request, or the password used on the local system for a responder request.</p> <p data-bbox="558 1108 1351 1255">If this parameter is not defined, no password checking is performed on incoming requests. When responder profiles are defined, you can specify a value of *VER for this parameter to indicate a trusted user on the remote system.</p>
NODE	<p data-bbox="558 1304 1140 1331">Defines the node name of the remote system.</p> <p data-bbox="558 1367 1065 1394">Valid values are <i>nodename</i> or *ALLNODE.</p> <div data-bbox="561 1423 1414 1675" style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p data-bbox="578 1444 1398 1654">Note: *ALLNODE indicates that you can specify the same user ID/password combination for all node definitions. If a user knows that the user ID/password combinations are the same on all systems, the user can use a single user profile definition to define a single user ID/password that can be used for all file transfer requests for that user ID.</p> </div>
IPADDR	Defines the IP address of the remote system in dotted decimal

Parameter	Description
	notation.
IPNAME	Defines the IP name of the remote system.
LIST	Defines the list name of the remote system.
RESPONDER	Defines whether a request requires a responder profile. The default value is YES.
LOCAL_PASSWORD (LPASS LOCAL_PASS)	<p>Defines the password for a user ID on the local system.</p> <p>This parameter is used only when:</p> <ul style="list-style-type: none"> RESPONDER=YES is defined for the profile. The LUSER or LOCAL_USER parameter is defined. The user defined by the LUSER parameter is different than the user initiating the request. The GLOBAL RESPONDER_PROFILE_LPASS parameter is set to YES. <p>Note: Only the Platform Server administrator can use this parameter.</p>
LOCAL_USER	<p>Defines the user ID on local system.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> *ALLUSER: an administrator can specify a default user ID for all users that are not specifically defined on the system. An administrator can define specific user ID/passwords for a node for users with high security authorization, while set a default user ID/password to be used for all other users with less security authorization defined. <p>Note: This option is only used by the initiator profiles.</p> <ul style="list-style-type: none"> <i>userid</i>: a user ID on the local system.

Parameter	Description
	<p>Note: Only the Platform Server administrator can use this parameter.</p> <p>A user is determined to be a Platform Server administrator only one of the following condition is met:</p> <ul style="list-style-type: none"> • BOSSID=ANY. • The user has CONTROL authorization for the facility defined in the BOSSID GLOBAL parameter.

Required and Optional User Profile Parameters

Different types of user profile function require different parameters.

See the following table for required and optional parameters for each type of user profile function.

Type	Required Parameter	Optional Parameter
TYPE=ADD	<ul style="list-style-type: none"> • REMOTE_USER • REMOTE_PASSWORD • NODE, IPADDR, IPNAME or LIST: only one of the four parameters is required. 	RESPONDER: Only required for responder profiles.
TYPE=CREATE	<ul style="list-style-type: none"> • REMOTE_USER • REMOTE_PASSWORD • NODE, IPADDR, IPNAME or LIST: only one of the four parameters is required. 	RESPONDER: Only required for responder profiles.
TYPE=DELETE	NODE, IPADDR, IPNAME or LIST: only one of the four parameters is required.	<ul style="list-style-type: none"> • LOCAL_USER: Only can be used by the Platform Server administrator.

Type	Required Parameter	Optional Parameter
		<ul style="list-style-type: none"> • REMOTE_USER: Only required for responder profiles. • RESPONDER: Only required for responder profiles.
TYPE=LIST	None	<ul style="list-style-type: none"> • NODE, IPADDR, IPNAME, or LIST: Only one of the four parameters is optional. • LOCAL_USER: Only can be used by the Platform Server administrator.
TYPE=REPLACE	<ul style="list-style-type: none"> • REMOTE_USER • REMOTE_PASSWORD • NODE, IPADDR, IPNAME or LIST: only one of the four parameters is required. 	RESPONDER: Only required for responder profiles.

User Profile JCL Examples

You can define user profile JCL to execute the Platform Server user profile utility FUSPROF.

See the following examples for your reference.

- To add a user profile entry:

```
TYPE=ADD
IPADDR=127.240.240.1
REMOTE_USER=UNIXUSER
REMOTE_PASSWORD=UNIXPASS
```

- To create a user profile entry:

i Note: If the profile already exists, the create action fails.

```
TYPE=CREATE
IPADDR=127.240.240.1
REMOTE_USER=UNIXUSER
REMOTE_PASSWORD=UNIXPASS
```

- To replace a user profile entry:

Note: If the profile already exists, the replace action fails.

```
TYPE=REPLACE
IPADDR=127.240.240.1
REMOTE_USER=UNIXUSER
REMOTE_PASSWORD=UNIXPASS
```

- To delete a user profile entry:

```
TYPE=DELETE
IPNAME=ACCTUSER.YOUR.COMPANY.COM
REMOTE_USER=
```

- To list user profile entries:

```
TYPE=LIST
NODE=NTNODE
```

- To add a default profile for all nodes for a user:

```
TYPE=ADD
NODE=*ALLNODE
REMOTE_USER=UNIXUSER1
REMOTE_PASSWORD=UNIXPASS1
```

- To add a default profile for all users for a node:

i Note: Only the Platform Server administrator can issue this request.

```
TYPE=ADD
LOCAL_USER=*ALLUSER
NODE=NTNODE
REMOTE_USER=UNIXUSER1
REMOTE_PASSWORD=UNIXPASS1
```

- To add a responder user profile entry:

```
TYPE=ADD
IPADDR=127.240.240.1
LOCAL_USER=USER123
REMOTE_USER=UNIXUSER
REMOTE_PASSWORD=UNIXPASS
RESPONDER=YES
```

If a request comes in from IP address 127.240.240.1 and the user ID is UNIXUSER and the password is UNIXPASS, then Platform Server uses the user ID USER123 for the transfer request.

- To add a default responder user profile for all users of a node:

```
TYPE=ADD
IPADDR=127.240.240.1
LOCAL_USER=USER123
REMOTE_USER=*ALLUSER
REMOTE_PASSWORD=UNIXPASS
RESPONDER=YES
```

If a request comes in from IP address 127.240.240.1 for any user ID, and the password is UNIXPASS, then the Platform Server uses the user ID USER123 for the transfer request.

- To add a default responder user profile for all users of a node without password checking:

```
TYPE=ADD
IPADDR=127.240.240.1
LOCAL_USER=USER123
REMOTE_USER=*ALLUSER
```

```
RESPONDER=YES
```

If a request comes in from IP address 127.240.240.1 for any user ID, then the Platform Server uses the user ID `USER123` for the transfer request. No password checking is done, because the `REMOTE_PASS` operand is not configured.

SUBJCL (Submit JCL) Interface

In many cases, sending a file to or receiving data from a partner is only the first step in the overall picture. The data must be processed by the application.

For a job initiated locally, it is simple to perform some processing after a file transfer is completed. However, when a remote system sends a file, it can be difficult to process the data. Many systems use user exits or job schedulers to process the received data. With the Platform Server SUBJCL facility, you can customize jobs to be submitted to the JES internal reader when a file transfer is completed.

You can customize a set of parameters that are compared against a file transfer request that are completed either successfully or unsuccessfully. When a match is found, the Platform Server submits a job to the internal reader to execute.

i Note: SUBJCL processing is only performed when a transfer is purged from the active queue. If a transfer is in retry state, SUBJCL processing is not performed.

For a successful file transfer request, when a file is received, the Platform Server can submit a job to process the data. When a file is sent to a remote system, the Platform Server can submit a job to archive or delete the dataset.

For an unsuccessful file transfer request, you might want to notify a user or a system that a request failed, update an error log, or notify an operator that a transfer failed. You can perform anything as you want in the job that is submitted.

SUBJCL Security

When a job is submitted into the internal reader, the user ID associated with the job must have enough security authorization to process the data but must not have too much authorization that it opens up a security breach. The Platform Server handles the user ID associated with the submitted job in three ways.

i Note: If the JOB card of the submitted job contains USER and PASSWORD parameters, they override the user ID under whose authority the job is submitted.

The Platform Server supplies the following three ways to define the user ID that submits the job:

- Use the user ID and password associated with the file transfer. This is the default method.
- Use the user ID of the Platform Server started task.
- Use a predefined user ID that can be defined for each SUBJCL entry.

The SUBJCL processing can fail when you specify all the following situations:

- A job must be submitted on a file transfer failure.
- Use the user ID and password associated with the file transfer.
- The transfer fails because the user ID and password are invalid.

In this case, the SUBJCL processing cannot be performed because the Platform Server cannot log on with the file transfer user ID and password. As a result, a message is displayed indicating why the job submission cannot be completed.

When a job is submitted, the Platform Server reads the JCL cards from one of the following two datasets:

- From the DSN defined in the JCL parameter if a fully qualified DSN is specified.
- From the GLOBAL JOB_SUBMIT_DSN parameter if a member name is specified by the JCL parameter.

i Note: The user associated with the job must have READ access to these datasets. If the user does not have access to the dataset, then the job submission request fails and an error message is displayed.

SUBJCL Configuration Parameters

The SUBJCL DD statement of the Platform Server started task contains the configuration parameters that are used to control the SUBJCL processing.

These parameters are separated by commas. The last parameter must be followed by a space to indicate the end of the entry. If you want to continue a configuration entry, simply place a comma after the last entry on a line, and start the next parameter on the next configuration card. Comments are defined by placing an asterisk (*) in column 1. If a parameter is not defined, then it is assumed to be a match.

See the following table for the configuration parameters you can define.

Parameter	Description
SUBMIT	<p>Defines the start of a SUBJCL parameter entry.</p> <p>This parameter is required. This must be the first parameter entry.</p>
JCL	<p>Defines the JCL that must be submitted.</p> <p>This parameter can be defined in one of two ways:</p> <ul style="list-style-type: none"> • Member name: The Platform Server uses the DSN defined by the GLOBAL JOB_SUBMIT_DSN parameter. • Fully qualified dataset name: This can be a sequential file, or it can point to a member within a PDS. <p>This parameter is required.</p>
SUBUSER	<p>Defines the user ID under whose authorization the job is submitted.</p> <p>This parameter can be defined in three ways:</p> <ul style="list-style-type: none"> • STC: Uses the user ID associated with the started task. • TRANSFER: Uses the user ID and password associated with the file transfer request. • Uses the user ID defined by the SUBUSER parameter. <p>The default value is TRANSFER.</p>
TYPE	<p>Defines the type of file transfer request.</p> <p>Valid values are SEND, RECEIVE, or BOTH.</p> <p>BOTH indicates that both send and receive requests are considered as a match.</p>

Parameter	Description
SOURCE	<p>Defines the source of file transfer request.</p> <p>Valid values are INITIATOR, RESPONDER, or BOTH.</p> <p>BOTH indicates that both initiator and responder requests are considered as a match.</p>
STATUS	<p>Defines whether a transfer request is successful or unsuccessful.</p> <p>Valid values are SUCCESS, FAILURE, or BOTH.</p> <p>BOTH indicates that both successful and unsuccessful requests are considered as a match.</p>
DSN	<p>Defines the fully qualified dataset name.</p> <p>This field is compared against the local dataset name in the file transfer request. If a member is transferred, the member name must be included in this parameter.</p>
PROCESS	<p>Defines the process name associated with the transfer request.</p>
IPADDR	<p>Defines the IP address of the node that is communicating with the Platform Server for z/OS.</p> <p>If a request is not from an IP address, then this parameter does not match.</p>
NODE	<p>Defines the node name of a transfer request.</p> <p>For initiator requests, this parameter is used when the NODE parameter is used on a request.</p> <p>For responder requests, the Platform Server scans the NODE table for matches on the LUNAME or IP address. These entries are then matched against the value specified in the NODE parameter.</p>

See the following examples of SUBJCL configuration entries.

Example 1:

```
SUBMIT, JCL=ACCTJOB1,  
        SUBUSER=TRANSFER,  
        STATUS=SUCCESS,  
        SOURCE=RESPONDER,  
        TYPE=RECEIVE,  
        DSN=PROD.ACCT.PAYROLL,  
        NODE=NYACCT
```

A match on this entry occurs when all the following criteria are met.

- A responder request is detected.
- The node associated with the request is NYACCT.
- The request is a receive transfer.
- The file is successfully received.
- The dataset received is PROD.ACCT.PAYROLL.

When all of these fields match, the Platform Server logs on as the user associated with the file transfer, extracts the GLOBAL JOB_SUBMIT_DSN dataset name, allocates that dataset, and then submits member ACCTJOB1 to the internal reader.

Example 2:

```
SUBMIT, JCL=TECHSUP.JCL.CNTL(FAILED),  
        SUBUSER=TECHSUP, STATUS=FAILED,  
        SOURCE=BOTH, TYPE=BOTH
```

A match on this entry occurs when all the following criteria are met.

- An initiator or responder request is detected.
- The request is a send or receive transfer.
- The request is unsuccessful.

When all of these fields match, the Platform Server logs on as user TECHSUP and submits the data in TECHSUP.JCL.CNTL(FAILED) to the internal reader.

SUBJCL Parameter Substitution

When submitting a job into the internal reader, the Platform Server can pass specific data associated with the file transfer request to the submitted job through substitutable parameters.

See the following table for supported substitutable parameters.

Parameter	Substituted Data
&TYPE	SEND or RECEIVE
&SOURCE	INITIATOR or RESPONDER
&STATUS	SUCCESS or FAILURE
&RC	Numeric return code (0 if successful)
&DSN	Local dataset name
&PROCESS	Process name
&NODE	Node name (or NODE if no node is found.)
&IPADDR	IP address (or IPADDR if not IP)
&TRN	Transaction number

When a substitutable parameter is used, the parameter must terminate with one of these characters: period, space, null, comma, open parenthesis, close parenthesis, or single quotation mark. If the substitutable parameter is terminated by a period, the period is removed as a part of the parameter substitution.

See the following Platform Server script program as an example of how the parameter substitution works. The JCL defined is located in the Platform Server JCL library with a member name of QSUBJCL.

```
//JOB CARD JOB , 'CFUSION', MSGCLASS=X, REGION=5M, CLASS=A
//S1 EXEC PGM=OSIUC000, PARM='SERVER=FUSION'
//STEPLIB DD DSN=FUSION.LOADLIB, DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSTRACE DD SYSOUT=*
```

```
//SYSIN DD *
WTO "TYPE=&TYPE."
WTO "SOURCE=&SOURCE."
WTO "STATUS=&STATUS."
WTO "RC=&RC."
WTO "TRN=&TRN."
WTO "DSN=&DSN."
WTO "NODE=&NODE."
WTO "IPADDR=&IPADDR."
WTO "PROCESS=&PROCESS."
```

If you have the following configuration entry:

```
SUBMIT, JCL=FUSION.JCL(QSUBJCL),
STATUS=SUCCESS,
SOURCE=RESPONDER,
TYPE=SEND,
DSN=TECHSUP.TEST.DATA
```

When a send request for dataset TECHSUP.TEST.DATA is started by a remote system, the JCL contained in the QSUBJCL member of the Platform Server JCL library is submitted to the JES internal reader. When the job is executed, the following information is displayed on the z/OS console.

```
TYPE=SEND
SOURCE=RESPONDER
STATUS=SUCCESS
RC=0
TRN=R125200000
DSN=TECHSUP.TEST.DATA
NODE=SUPPORT
IPADDR=192.100.100.1
PROCESS=CYBERFUS
```

CFACCESS

With the Platform Server CFACCESS (Access Control) function, the administrator can control file transfer capabilities for a user or node.

For entry into the z/OS system, the Platform Server requires a valid user ID and password. To ensure only authorized users can transfer data successfully, the Platform Server

validates this information with RACF or other security system and verifies if the user is authorized to access the transfer dataset.

But under certain conditions, the Platform Server administrator must have additional control over the functions that users can perform and the datasets that they can access. The Access Control function provides this capability.

Using Access Control, the administrator can control the file transfer capabilities for:

- A user
- A node or IP address
- A combination of user and node/IP address

The administrator can restrict the following transfer functions:

- Sending a file
- Receiving a file
- Submitting a job into the internal reader
- Executing a command
- The High Level Qualifier (HLQ) for a file send transfer
- The HLQ for a file receive transfer

Additionally, the administrator can restrict the following postprocessing actions (PPA):

- Executing a command
- Submitting a job into the internal reader
- The DSN for JCL to be submitted into the internal reader

i Note: CFACCESS checking is only performed for responder transfers.

The file transfer type is dependent on the Platform Server for z/OS that processes the request. For example, a send request on Platform Server for Windows is processed as a receive request on Platform Server for z/OS and the receive parameters are checked against the CFACCESS configuration.

The CFACCESS configuration parameters can be activated through one of following ways:

- When Platform Server starts up
- When the Platform Server CFACCESSREFRESH operator command is entered

CFACCESS Parameter Configuration

The CFACCESS configuration parameters are defined in the Platform Server JCL DD statement CFACCESS. The Platform Server SAMPLIB member CFACCESS shows a sample of the Platform Server CFACCESS configuration.

The CFACCESS configuration file is processed until a match is found, and only one entry is processed even if there are multiple entries that match. The CFACCESS processing is performed after the Platform Server responder profile processing. Therefore, the user ID defined is the actual z/OS user ID used by the Platform Server.

When defining CFACCESS configuration parameters, you have to comply with the following rules for continuation and comments.

- Parameters can be entered on a single line or on multiple lines and are delimited by a comma.
- If a space follows the comma, the parameter continues on the next line. If there is a space in the parameter, you must enclose the space in double quotation marks.
- Comments are defined by placing an asterisk (*) in column 1.

In the following example, each parameter except the last one ends with a comma. By not putting a comma in the last parameter, the entry is completed.

```
USERID="DEFAULT",
NODE=NODEA,
SEND_HLQ="c:\temp\",
SEND_OPTION=ROOT,
RECEIVE_OPTION=NEVER
```

Is the same as:

```
USERID=DEFAULT,NODE=NODEA,SEND_DIR=c:\temp\,SEND_OPTION=ROOT,RECEIVE_
OPTION=NEVER
```

CFACCESS Parameters

You have to define some CFACCESS parameters to control file transfer authorization for a user or node.

See the following table for available CFACCESS parameters.

Parameter	Description
USERID	<p>Defines the user ID that is associated with the file transfer.</p> <p>This parameter can be used along with the NODE or IPADDR parameters to further qualify the CFACCESS matching scan.</p> <p>Note: If the NODE and IPADDR parameters are not defined, then this parameter must be defined.</p> <p>You can specify a special user ID called DEFAULT. With this setting, CFACCESS scanning detects a match for the user ID in this CFACCESS configuration entry. If this parameter is not entered, no checking is performed on the user ID for this entry.</p>
NODE	<p>Defines the node that is associated with the file transfer.</p> <p>Nodes are defined within the Platform Server CONFIG DD statement. This parameter can be used along with the USERID parameter to further qualify the CFACCESS matching scan.</p> <p>Note:</p> <ul style="list-style-type: none">• If the USERID and IPADDR parameters are not defined, then this parameter must be defined.• This parameter is mutually exclusive with the IPADDR parameter. <p>You can specify a special node called DEFAULT. With this setting, CFACCESS scanning detects a match for the node in this CFACCESS configuration entry. If this parameter is not entered, no checking is performed on the node for this entry.</p>
IPADDR	<p>Defines the IP address that is associated with the file transfer.</p> <p>This parameter can be used along with the USERID parameter to further qualify the CFACCESS matching scan.</p>

Parameter	Description
SEND_OPTION	<p data-bbox="548 310 623 338">Note:</p> <ul data-bbox="597 369 1338 520" style="list-style-type: none"> <li data-bbox="597 369 1338 432">• If the USERID and NODE parameters are not defined, this parameter must be defined. <li data-bbox="597 457 1273 520">• This parameter is mutually exclusive with the NODE parameter. <p data-bbox="532 573 1390 642">If this parameter is not entered, no checking is performed on the IP address for this entry.</p> <p data-bbox="532 695 997 722">Defines the options for sending files.</p> <p data-bbox="532 753 1305 823">This parameter has no default value, which indicates that no restrictions are placed on a user/node.</p> <p data-bbox="532 854 737 882">Valid values are:</p> <ul data-bbox="581 913 1414 1598" style="list-style-type: none"> <li data-bbox="581 913 1414 1087">• ROOT: Any file names defined in the file transfer request are appended to the file name defined by the SEND_HLQ parameter. If ROOT is specified, then the SEND_HLQ parameter must also be defined. <li data-bbox="581 1119 1414 1402">• FORCE: If the transfer file name starts with the HLQ defined in the SEND_HLQ parameter, then no changes are made to the transfer file name. Otherwise, any file names defined in the file transfer request are appended to the HLQ defined by the SEND_HLQ parameter. If FORCE is specified, then the SEND_HLQ parameter must also be defined. <li data-bbox="581 1434 1382 1503">• NEVER: The user/node defined in this entry cannot send files. All send requests are terminated with errors. <li data-bbox="581 1535 1414 1598">• USE: The user is authorized to send files to this system. The file name is not changed when USE is specified.
RECEIVE_OPTION	<p data-bbox="532 1650 1013 1677">Defines the options for receiving files.</p> <p data-bbox="532 1709 1305 1736">This parameter has no default value, which indicates that no</p>

Parameter	Description
	<p>restrictions are placed on a user/node.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • ROOT: Any file names defined in the file transfer request are appended to the file name defined by the <code>RECEIVE_HLQ</code> parameter. <p>If <code>ROOT</code> is specified, then the <code>RECEIVE_HLQ</code> parameter must also be defined.</p> <ul style="list-style-type: none"> • FORCE: If the transfer file name starts with the HLQ defined in the <code>RECEIVE_HLQ</code> parameter, then no changes are made to the transfer file name. Otherwise, any file names defined in the file transfer request are appended to the HLQ defined by the <code>RECEIVE_HLQ</code> parameter. <p>If <code>FORCE</code> is specified, then the <code>RECEIVE_HLQ</code> parameter must also be defined.</p> <ul style="list-style-type: none"> • NEVER: The user/node defined in this entry cannot receive files. All send requests are terminated with errors. • USE: The user is authorized to receive files to this system. The file name is not changed when <code>USE</code> is specified.
<code>COMMAND_OPTION</code>	<p>Defines the options for executing commands.</p> <p>This parameter has no default value, which indicates that no restrictions are placed on a user/node. The command options are checked under the following two circumstances:</p> <ul style="list-style-type: none"> • When a file transfer <code>TYPE=COMMAND</code> request is received. • When Post Processing Actions (PPA) are executed. <p>Valid values are:</p> <ul style="list-style-type: none"> • ALLOW: Users can execute commands on this system. • NEVER: The user/node defined in this entry cannot execute commands. Any command requests are terminated with errors.

Parameter	Description
SUBMIT_OPTION	<p>Defines the options for submitting jobs to the z/OS internal reader.</p> <p>This parameter has no default value, which indicates that no restrictions are placed on a user/node.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • ROOT: The file name defined in the file transfer request is actually a member name. The dataset name is defined by the SUBMIT_HLQ parameter. Parentheses are placed around the member name, and the member name is appended to the DSN defined by the SUBMIT_HLQ parameter. <p>This processing is performed only on PPA submit requests. When ROOT is specified, file name cannot be changed in file transfer submit requests.</p> <ul style="list-style-type: none"> • NEVER: The user/node defined in this entry cannot submit jobs into the internal reader. Any submit requests including PPA submit requests are terminated with errors. • ALLOW: The user is authorized to submit jobs into the internal reader. The file name is not changed when ALLOW is specified.
SEND_HLQ	<p>Defines the HLQ to which the file name of the sent file is appended to create the new file transfer file name.</p> <p>This parameter is required when SEND_OPTION is configured as ROOT or FORCE. It is ignored otherwise.</p> <p>You can restrict the file names that users can use.</p>
RECEIVE_HLQ	<p>Defines the HLQ to which the file name of the received file is appended to create the new file transfer file name.</p> <p>This parameter is required when RECEIVE_OPTION is configured as ROOT or FORCE. It is ignored otherwise.</p> <p>You can restrict the file names that users can use.</p>
SUBMIT_HLQ	<p>Defines the HLQ to which the file name of the transfer request is appended to create the new file transfer file name.</p>

Parameter	Description
	<p>This parameter is required when <code>SUBMIT_OPTION</code> is configured as <code>ROOT</code>. It is ignored otherwise.</p> <p>You can restrict the file names that users are allowed to use. The new file transfer name is read and the data records are written to the z/OS internal reader.</p> <p>Note: This parameter is only used for PPA submit requests.</p>

CFACCESS Example

This section provides you a sample of CFACCESS configuration.

For example, take the following CFACCESS configuration:

```

USERID=ACCTUSER,
NODE=NYNODE,
  SEND_HLQ="ACCT.SEND",
  RECEIVE_HLQ="ACCT.RECEIVE",
  SUBMIT_HLQ="ACCT.SUBMIT.JCL",
  SEND_OPTION=ROOT,
  RECEIVE_OPTION=FORCE,
  COMMAND_OPTION=NEVER,
  SUBMIT_OPTION=ROOT

```

When a file transfer request comes in from user `ACCTUSER` on node `NYNODE`, a match is found on this entry. Based on the following file transfer file names, the actual file name used is shown:

- File receive is detected. The transfer file name is `ACCT.RECEIVE.TAX.DATA`. Because `RECEIVE_OPTION=FORCE` is specified, and the HLQ defined in the `RECEIVE_HLQ` matched the data in the file name, no changes are made to the file name. The file name used is `ACCT.RECEIVE.TAX.DATA`.
- File receive is detected. The transfer file name is `TAX.DATA.Y2002`. Because `RECEIVE_OPTION=FORCE` is specified, and the HLQ defined in the `RECEIVE_HLQ` does not match the data in the file name, the transfer file name is appended to the `RECEIVE_HLQ` file name. The file name used is `ACCT.RECEIVE.TAX.DATA.Y2002`.

- File send is detected. The transfer file name is `AUDIT.NY`. Because `SEND_OPTION=ROOT` is specified, the transfer file name is appended to the `SEND_HLQ` file name. The file name used is `ACCT.SEND.AUDIT.NY`.
- Command execution is detected. Because `COMMAND_OPTION=NEVER` is defined, the request is terminated with errors.
- File submit is detected. The `SUBMIT_HLQ` parameter is ignored, because the remote user transmits the JCL to z/OS where it is written directly into the internal reader.
- PPA submit is detected. The PPA data is `JOB1`. The PPA data becomes the member name, and the `SUBMIT_HLQ` is the dataset name. The file name used to submit the JCL is `ACCT.SUBMIT.JCL(JOB1)`.

CFALIAS

With the Platform Server CFALIAS (File Alias Control) function, the administrator can provide an alias for a file based on the information about the initiator.

When transmitting a file through the Platform Server, a user must know the file name on both the local system and the destination systems. Giving an outside user the permission to know and specify file names on your mainframe opens up a small security hole. If they can specify a file name, then they can change the file name. If RACF gives them authorization, then they can update the file.

With the Platform Server CFALIAS, you can limit the ability of a user to know and define a file that exists on the mainframe. For example, you can tell the user to define the file name as `DOG`, and Platform Server CFALIAS can change that file name to an actual file name.

Using File Alias Control, you can define the following criteria to permit a user to supply aliases on a file:

- A user
- A node or IP address
- A combination of user and node/IP address

You can also use the following additional criteria:

- Send or receive
- File name as it exists on the mainframe
- Alias file name as entered by the user

i Note: CFALIAS checking is only performed for responder transfers. A responder transfer is a transaction that is started by a different Platform Server.

The file transfer type is dependent on the Platform Server for z/OS that processes the request. For example, a send request on the Platform Server for Windows is processed as a receive request on the Platform Server for z/OS and the receive parameters are validated against the Platform Server CFALIAS configuration.

You can activate the CFALIAS configuration parameters through one of the following ways:

- When Platform Server starts up
- When the Platform Server CFALIASREFRESH operator command is entered

CFALIAS Parameter Configuration

The CFALIAS configuration parameters are defined in the Platform Server JCL DD statement CFALIAS. The Platform Server SAMPLIB member CFALIAS shows a sample of the Platform Server CFALIAS configuration.

The CFALIAS configuration file is processed from the beginning until a match is found, and that only one entry is processed even if there are multiple entries that match. The CFALIAS processing is performed after the Platform Server responder profile processing. Therefore, the user ID defined is the actual z/OS user ID used by the Platform Server.

Continuation and Comments

Parameters can be entered on a single line or multiple lines. Parameters are delimited by a comma. If a space follows the comma, the parameter continues on the next line. If there is a space in the parameter, you must enclose the space in double quotation marks.

Comments are defined by placing an asterisk (*) in column 1.

See the following table for the valid parameters.

Parameter	Description
%ACB	VTAM ACB name (z/OS only)
%GDATE	Gregorian date in YYMMDD format

Parameter	Description
%GDATEC	Gregorian date in CCYYMMDD format
%JDATE	Julian date in YYDDD format
%JDATEC	Julian date in CCYYDDD format
%JOBN	Job name (z/OS only)
%NODE	Node name (If no node is defined, use the value NODE.)
%SYSID	System name
%TIME	Time in HHMMSS format
%TIMET	Time in HHMMSST format
%TRN	Transaction number
%USER	User name

In the following example, each parameter except the last one ends with a comma. By not putting a comma in the last parameter, you can indicate that the entry is completed.

Example 1:

```
FILE=PROD.ACCT. %NODE . D%GDATE . T%TIMET
```

Can be changed to:

```
FILE=PROD.ACCT. NYNODE . D051230 . T1601029
```

Example 2:

```
USERID="DEFAULT",
NODE=NODEA,
TYPE=RECEIVE,
FILE=MY.ZOS.FILE,
```

```
ALIAS=FILE123
```

Is the same as:

```
USERID=DEFAULT ,NODE=NODEA ,TYPE=RECEIVE ,FILE=MY.ZOS.FILE ,ALIAS=FILE123
```

CFALIAS Parameters

You have to define some CFALIAS parameters to restrict a user's ability to define a file on your mainframe.

See the following table for the available CFALIAS parameters.

Parameter	Description
USERID	<p>Defines the user ID that is associated with the file transfer.</p> <p>This parameter can be used along with the NODE or IPADDR parameters to further qualify the CFALIAS matching scan.</p> <p>Note: If the NODE and IPADDR parameters are not defined, then this parameter must be defined.</p> <p>You can specify a special user ID called DEFAULT. With this setting, CFALIAS scanning detects a match for the user ID in this CFALIAS configuration entry. If this parameter is not entered, no checking is performed on the user ID for this entry.</p>
NODE	<p>Defines the node that is associated with the file transfer.</p> <p>Nodes are defined within the Platform Server CONFIG DD statement. This parameter can be used along with the USERID parameter to further qualify the CFALIAS matching scan.</p>

Parameter	Description
IPADDR	<p data-bbox="529 310 604 338">Note:</p> <ul data-bbox="578 369 1357 520" style="list-style-type: none"> <li data-bbox="578 369 1357 432">• If the USERID and IPADDR parameters are not defined, then this parameter must be defined. <li data-bbox="578 457 1284 520">• This parameter is mutually exclusive with the IPADDR parameter. <p data-bbox="509 575 1365 722">You can specify a special node called DEFAULT. With this setting, CFALIAS scanning detects a match for the node in this CFALIAS configuration entry. If this parameter is not entered, no checking is performed on the node for this entry.</p>
IPADDR	<p data-bbox="509 770 1297 798">Defines the IP address that is associated with the file transfer.</p> <p data-bbox="509 833 1330 903">This parameter can be used along with the USERID parameter to further qualify the CFALIAS matching scan.</p> <p data-bbox="529 951 604 978">Note:</p> <ul data-bbox="578 1010 1382 1161" style="list-style-type: none"> <li data-bbox="578 1010 1382 1073">• If the USERID and NODE parameters are not defined, then this parameter must be defined. <li data-bbox="578 1098 1252 1161">• This parameter is mutually exclusive with the NODE parameter. <p data-bbox="509 1211 1370 1278">If this parameter is not entered, no checking is performed on the IP address for this entry.</p>
TYPE	<p data-bbox="509 1329 1305 1356">Defines whether the processing is for send or receive requests.</p> <p data-bbox="509 1392 1243 1419">This parameter is relative to the Platform Server for z/OS.</p> <p data-bbox="509 1455 716 1482">Valid values are:</p> <ul data-bbox="561 1514 1398 1776" style="list-style-type: none"> <li data-bbox="561 1514 1398 1583">• SEND: An entry is only matched if the Platform Server is sending a file. <li data-bbox="561 1608 1338 1677">• RECEIVE: An entry is only matched if the Platform Server is receiving a file. <li data-bbox="561 1703 1370 1772">• BOTH: An entry is matched if the Platform Server is sending or receiving a file.

Parameter	Description
FILE	<p>Defines the actual file name on the z/OS system.</p> <p>The file name is translated to upper case unless the file is an HFS file. When specified, both FILE and ALIAS must be defined.</p> <p>Note: FILE and ALIAS are mutually exclusive with the ALLOW parameter.</p>
ALIAS	<p>Defines the file name that the user sends.</p> <p>This file name is translated to upper case, and is therefore not case sensitive. When specified, both FILE and ALIAS must be defined.</p> <p>Note: FILE and ALIAS are mutually exclusive with the ALLOW parameter.</p>
ALLOW	<p>Defines whether the user is authorized to execute certain functions.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • YES: The user is authorized to use an actual file name in a file transfer request. CFALIAS processing is not performed for this USERID and NODE/IPADDR combination. • NO: The user is not authorized to perform file transfer operations. The file transfer request for this USERID and NODE/IPADDR combination is rejected. <p>Note: This parameter is mutually exclusive with the FILE and ALIAS parameters.</p>

CFALIAS Example

This section provides you a sample of CFALIAS configuration.

For example, take the following CFALIAS configuration:


```

USERID=ACCTUSER,
  NODE=NYNODE,
  TYPE=SEND,
  ALIAS=MONTHLY,
  FILE=NYUSER.ACCT.MONTHLY.DATA

```

```

USERID=ACCTUSER,
  NODE=NYNODE,
  ALLOW=NO

```

```

USERID=DEFAULT,
  ALLOW=YES

```

When a file transfer request comes in from user ACCTUSER on node NYNODE for a send request and the file name entered is MONTHLY, a match is found on the first entry. The file name MONTHLY is changed to the actual file name NYUSER.ACCT.MONTHLY.DATA.

The second entry indicates that if any other requests come in from user ACCTUSER on node NYNODE, the request is rejected.

The third entry allows all other users to enter actual file names. This means that no CFALIAS checking is performed on any file transfer except for user ACCTUSER and node NYNODE.

TRCLASS (Transfer Class)

By assigning a transfer class to a file transfer, the Platform Server ensures that the maximum number of concurrent transfers for that transfer class is not exceeded.

The Platform Server has a variety of ways to limit the number of concurrent initiator transfers:

- GLOBAL parameters: MAXINIT, MAXRESP, and MAXTOTAL
- Node parameters: CONTENTION_WINNERS and CONTENTION_LOSERS

Transfer classes give you an additional way to limit the number of concurrent file transfers.

i Note:

- Like the GLOBAL MAXINIT and the node CONTENTION_WINNERS parameters, transfer class processing is only performed for initiator transfers. An initiator transfer is a transaction that is started by the local Platform Server and to be sent to a remote Platform Server
- Transfer class processing is only performed for file transfers. Scripts that run in the started task are not limited by the transfer classes.

You can use transfer classes through the following process:

1. Define the transfers classes to the started task.

Equate a transfer class name with a number and a max number of transfers.

2. Define the transfer class that is to be used by the transfer.

Transfer classes work within a single node definition and also work between different nodes. If you have a transfer class defined with a maximum of two concurrent transfers, then only two transfers can be active at any time. These transfers can be for the same node or different nodes.

When a transfer is completed, the Platform Server dispatcher immediately checks if transfers can be executed for the node that meets the transfer class limits. The dispatcher does not dispatch transfers for another node using the same transfer class until the next dispatch cycle.

You can activate the TRCLASS configuration parameters through one of the following two ways:

- At the Platform Server startup.
- When the Platform Server TRCLASSREFRESH operator command is entered.

TRCLASS Parameter Configuration

The TRCLASS configuration parameters are defined in the Platform Server JCL DD statement TRCLASS. The Platform Server SAMPLIB member TRCLASS shows a sample of the TRCLASS configuration.

Continuation and Comments

Transfer class parameters for each individual transfer class must be entered on a single line and must start in column 1. Comments are defined by placing an asterisk (*) in column 1.

Transfer Class Parameters

You must specify the transfer class parameters to define a transfer class.

See the following table for the available TRCLASS parameters.

Parameter	Description
TRCLASS	<p>Defines the name of the transfer class.</p> <p>This parameter is required. The transfer class name can be from 1 to 12 characters long. Transfer class names must be unique.</p> <p>Note: Transfer class names are case insensitive and are displayed in upper case.</p>
NUM	<p>Defines the transfer class number.</p> <p>This parameter is required. The transfer class number must be a numeric value between 1 and 255. Transfer class numbers must be unique.</p>
MAX	<p>Defines the maximum number of concurrent transfers for the transfer class.</p> <p>This parameter is optional. Valid values are from 0 to 255. The value 0 means an unlimited number of concurrent transfers. If this parameter is not entered, an unlimited number of transfers can be performed using the transfer class.</p>

TRCLASS Example

This section provides you a sample of TRCLASS configuration.

See the following example of the TRCLASS from the Platform Server SAMPLIB library:

```
TRCLASS=SINGLETHREAD,NUM=100,MAX=1
TRCLASS=MANYTRANSFER,NUM=200,MAX=255
TRCLASS=UNLIMITED,NUM=201,MAX=0
TRCLASS=TRCLASS1,NUM=1,MAX=1
```

```

TRCLASS=TRCLASS2,NUM=2,MAX=2
TRCLASS=TRCLASS3,NUM=3,MAX=3
TRCLASS=TRCLASS4,NUM=4,MAX=4
TRCLASS=TRCLASS5,NUM=5,MAX=5
TRCLASS=TRCLASS6,NUM=6,MAX=6
TRCLASS=TRCLASS7,NUM=7,MAX=7
TRCLASS=TRCLASS8,NUM=8,MAX=8
TRCLASS=TRCLASS9,NUM=9,MAX=9
TRCLASS=TRCLASS10,NUM=10,MAX=10
TRCLASS=TRCLASS11,NUM=11,MAX=11
TRCLASS=TRCLASS12,NUM=12,MAX=12
TRCLASS=TRCLASS13,NUM=13,MAX=13
TRCLASS=TRCLASS14,NUM=14,MAX=14

```

In this sample, the following transfer classes are defined:

- **SINGLETHREAD:** Uses transfer class number 100, and only one transfer can run concurrently.
- **MANYTRANSFER:** Uses transfer class number 200 and up to 255 transfers can run concurrently.
- **UNLIMITED:** Uses transfer class number 201 and an unlimited number of transfers can run concurrently.
- **TRCLASS14:** Uses transfer class number 14 and only up to 14 transfers can run concurrently.

When an initiator transfer is queued, the internal code converts the transfer class name to the corresponding transfer class number. So if the transfer class configuration is refreshed, the changes are not reflected in transfers that are already scheduled.

SSL Configuration

To configure SSL for TIBCO MFT Platform Server for z/OS, you must specify the required parameters.

GLOBAL SSL Parameter Definitions

You must define some GLOBAL parameters when using SSL.

The following table lists the parameters for SSL usage.

Parameter	Description
SSL_CLIENT_DNLABEL	<p>Defines the label name of certificate that is used for client connections (for example, Initiator).</p> <p>If this parameter is not specified, the certificate defined by the SSL_DNLABEL is used.</p>
SSL_DNLABEL	<p>Defines the label name of certificate that is used.</p> <p>If you want to use the default certificate, you must specify this parameter as NULL in upper case. This certificate is used for both the server and client unless the SSL_CLIENT_DNLABEL parameter is specified.</p>
SSL_ENCRYPT	<p>Defines the default encryption type that is used for SSL requests.</p>
SSL_KEY_DBNAME	<p>Defines the name of the key database created by the gskkyman utility, or the ring file name created by the RACF RACDCERT command.</p>
SSL_NETWORK_IPADDR	<p>Defines the IP address of the local system used to decide whether a request must be an SSL request.</p> <p>The default value is the IP address of the local system.</p>
SSL_NETWORK_IPADDR_IPV6	<p>Defines the IPv6 address used to define whether a request must be an SSL request.</p> <p>The Platform Server takes the IPv6 address of the local system and the IP address of the target system, and determines the subnet of these two addresses by using the SSL_NETWORK_SUBNET_IPV6 parameter.</p> <p>The Platform Server then compares the two values to determine if a request is within the subnet, or outside the subnet. If inside the subnet, then the request does not have to be an SSL request. If outside the subnet, then the request must be an SSL request.</p>

Parameter	Description
SSL_NETWORK_SUBNET	<p>Defines the subnet of the SSL_NETWORK_IPADDR that is used when checking if a request must use SSL.</p>
SSL_REQUEST	<p>Defines whether SSL must be used.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • NO: SSL is not required but can be used. This is the default value. • YES: SSL must be used for both initiator and responder requests. • OUTSIDE_NETWORK: SSL usage depends on the IP address of the target system and how it compares with the address of the local system. <p>If the target system address is outside the subnet defined by the SSL_NETWORK_IPADDRESS and SSL_NETWORK_SUBNET parameters, then the request must use SSL.</p>
SSL_REQUEST_IPV6	<p>Defines when or whether SSL must be used on IPv6 networks.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • NO: SSL is not required but can be used. This is the default value. • YES: SSL must be used for both initiator and responder requests. • OUTSIDE_NETWORK: SSL usage depends on the IPv6 address of the target system and how it compares with the address of the local system. <p>If the target system address is outside the subnet defined by the SSL_NETWORK_IPADDR_IPV6 and SSL_NETWORK_SUBNET_IPV6 parameters, then the request must use SSL.</p>
SSLIPPORT_IPV6	<p>Defines the IPv6 port that the Platform Server listens on for SSL requests.</p> <p>If non-SSL requests are received on this port, then an error</p>

Parameter	Description
SLLISTEN_ADAPTER_IPADDR	<p>message is sent to the initiator and the request is terminated.</p> <p>This field must be different than the IPPORT parameter, and unique on the z/OS system. It has no default value. If this parameter is not defined, then responder IPv6 SSL processing is disabled.</p>
SLLISTEN_ADAPTER_IPADDR	<p>Defines the IP address of the TCP network interface that the Platform Server started task listens for incoming connections.</p> <p>The default is to listen to all TCP network interfaces.</p>
SLLISTEN_ADAPTER_IPADDR_IPV6	<p>Defines the IPv6 address of the TCP network interface that the Platform Server started task listens to for incoming SSL connections.</p> <p>By default, the Platform Server started task listens to all TCP network interfaces. If you want to listen to only a single network interface, specify the IPv6 address of the network interface. Then the Platform Server only listens to that network interface for incoming requests.</p> <p>This parameter is used only for incoming (responder) SSL requests. It is ignored for outgoing (initiator) requests.</p>
SSLIPPORT	<p>Defines the IP port that the Platform Server listens on for SSL requests.</p> <p>If non-SSL requests are received on this IP port, then an error message is sent to the initiator and the request is terminated. This field must be different than the IPPORT parameter, and unique on the z/OS system.</p>
TLSCIPHERS	<p>Defines the TLS ciphers that are supported by MFT. The ciphers must be defined as 4 alphanumeric digits. The ciphers are documented in Appendix C of the IBM manual: <i>z/OS Cryptographic Services System Secure Sockets Layer Programming</i>. If not defined, MFT uses the default SSL ciphers.</p> <p>If FIPS140 is specified, only FIPS approved ciphers are used.</p>

Parameter	Description
	<p>Ciphers that meet the following criteria are specified in the sample GLOBAL member:</p> <ul style="list-style-type: none"> • FIPS approved • AES256 • SHA or higher message digest <p>Multiple TLSCIPHERS parameters can be defined. One TLS Cipher can be defined for each TLSCIPHERS parameter. The text after the 4 alphanumeric digits is used for documentation only and is ignored.</p>
<p>TLSENABLEDPROTOCOLS</p>	<p>Defines the TLS protocols that are supported when running in SSL Mode. Multiple TLS parameters can be entered separated by a comma.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • TLSV1: TLSV1 is supported • TLSV1_1: TLSV1_1 is supported • TLSV1_2: TLSV1_2 is supported • ALL <p>Note: SSLV2 and SSLV3 are not supported.</p> <p>Example: TLSENABLEDPROTOCOLS=TLSV1_1,TLSV1_2</p> <p>If this parameter is not entered, the default is ALL.</p>
<p>TLSTUNNELIPPORT</p>	<p>Defines the IPPORT that MFT Platform Server listens on for IPV4 TLS tunnel requests. Only TLS tunnel requests are received on this port. If a non-SSL or an SSL request is received on this port, an error is displayed and the request fails. Because a transfer has not been initiated, no audit record is written. This field must be unique on the z/OS system. There is no default for this parameter. If this parameter is not defined, then IPV4 TLS tunnel processing is disabled.</p>

Parameter	Description
TLSTUNNELIPPORT_IPV6	Defines the IPPORT that MFT Platform Server listens on for IPV6 TLS tunnel requests. Only TLS tunnel requests are received on this port. If a non-SSL or an SSL request is received on this port, an error is displayed and the request fails. Because a transfer has not been initiated, no audit record is written. This field must be unique on the z/OS system. There is no default for this parameter. If this parameter is not defined, then IPV6 TLS tunnel processing is disabled.

NODE SSL Parameter Definitions

You must define node SSL parameter when using SSL for node requests.

For more information, see [SSL](#) in [Node Definition Parameters](#).

User Interface TLS Parameter Definitions (Batch/REXX)

You must define the `TLS` (or `SSL`) parameter when using TLS on user interface.

See the following table for description on the `TLS` parameter.

Parameter	Description
TLS	<p>Defines whether SSL must be used for this request.</p> <p>Valid values are NO, YES, and TUNNEL. The default value is NO.</p> <p>This parameter overrides the definitions specified by the GLOBAL or NODE definitions.</p>

Defining the Password Associated with the SSL Key Database File

To define a password associated with a key database, you must use the Platform Server user profile utility, `PROFSSL`, located in the Platform Server JCL library.

When using the PROFSSL utility:

- The local user ID must be defined as \$SSLDB, while the remote user ID can be any user ID.
- Any IP address of any node name can be used in this definition. This IPADDR defined here is not used, but must be defined for the job to run successfully. NODE can be used in place of IPADDR, and it is also not used.
- If you are using a RACF RACDCERT command to create a key ring file, the REMOTE_PASS parameter must be defined as NULL in upper case. Otherwise, you must specify the password for the key database.

i Note: You must be a Platform Server administrator to add user ID \$SSLDB to the Platform Server user profile database.

See the following example of JCL that can be used to add the \$SSLDB user profile:

```
//jobcard JOB  , 'CFUSION',MSGCLASS=X,REGION=5M,CLASS=A
//STEP1 EXEC PGM=FUSPROF,PARM='SERVER=CFUSION'
//STEPLIB DD DSN=FUSION.LOADLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  TYPE=ADD
  IPADDR=127.0.0.1           This address is not used
  LOCAL_USER=$SSLDB
  REMOTE_USER=$SSLDB
  REMOTE_PASS=NULL         Specify NULL for RACF KEY RING
  **REMOTE_PASS=sslldbpassword Specify password for key database
//
```

Key Database

This section is required if you are running Top Secret or ACF2 and must create a key database.

i Note: If you are running RACF, you must use the RACF RACDCERT command to create your certificate ring file.

For more information on using the gskkyman utility, you can review this IBM document: *SC24-5877 OS/390 Cryptographic Services System Secure Sockets Layer Programming Guide and Reference*.

To make it easier to execute the `gskkyman` utility, you can copy the `KEYMAN` exec in the Platform Server EXEC library to the HFS system.

The following command can be used to copy the member to your HFS home directory:

```
OPUTX 'FUSION.EXEC(KEYMAN)' /u/userid LC MODE(733)
```

The `/u/userid` field must be changed to the directory where you want to add the exec. The `MODE(733)` assigns read/write/execute privileges to your user ID, and read/execute privileges to all other users. You can set this field as you want.

See the following copy of the exec that can be used to execute the `KEYMAN` utility:

```
# Run SSL Key Manager Program
export LIBPATH=$LIBPATH:./usr/lpp/gskssl/lib
export PATH=$PATH:./usr/lpp/gskssl/bin
export STEPLIB=$STEPLIB:CDS.SGSKLOAD
gskkyman $1 $2 $3
exit
```

The library specified is `CDS.SGSKLOAD`. This version is valid for OS/390 version 2.7. The dataset name for other versions might be different. If this dataset is not found, try using `GSK.SGSKLOAD`.

Creating the Key Database

With the `gskkyman` utility, you can create a key database.

Procedure

1. Execute the `KEYMAN` command in a z/OS shell to start the `gskkyman` utility.

The following screen is displayed for you to choose one of the three functions.

```
IBM Key Management Utility

Choose one of the following options to proceed.

1 - Create new key database
2 - Open key database
3 - Change database password
```

```

0 - Exit program

Enter your option number:

```

2. Enter option 1 to create a new key database.
3. Enter the key database name or press ENTER to use the default name key.kdb.

```

Enter key database name or press ENTER for "key.kdb":
===>

```

By default, the key database is created in the current working directory with a name of key.kdb.

4. Enter a password for the key database.

```

Enter password for the key database.....>
===>

```

i Note: You must remember this password because this is the password used by the Platform Server in the \$SSLDB user profile.

5. Enter the password again for validation.
6. Choose whether the password expires.

It is a good practice to use the default value 0 by pressing Enter, which indicates that the password does not expire.

Result

The key database menu is displayed as follows:

```

Key database menu

Current key database is /u/ibmuser/key.kdb

1 - List/Manage keys and certificates
2 - List/Manage request keys
3 - Create new key pair and certificate request

```

- 4 - Receive a certificate issued for your request
- 5 - Create a self-signed certificate
- 6 - Store a CA certificate
- 7 - Show the default key
- 8 - Import keys
- 9 - Export keys
- 10 - List all trusted CAs
- 11 - Store encrypted database password
- 0 - Exit program

Using the Key Database

You can perform these functions with a key database: storing a CA certificate, creating a new key pair and certificate request, and receiving a certificate issued for your request. If you want to perform any other functions, review the IBM manual for more information.

Storing a CA Certificate

Before you can work with certificates, you must store a CA certificate. If you attempt to store a certificate before the CA certificate is stored, you will receive the following error:

Error: The issuer of the key is not found.

Certificate authorities typically have Base64 encoded files to represent their certificates. This certificate must be saved to a file that is accessible by the `gskkyman` utility before any certificates are loaded into the system. Typically, this means you have to save the CA certificate as an HFS file.

See the following example of a CA certificate.

```
-----BEGIN CERTIFICATE-----
MIICmTCCAgKgAwIBAgIBADANBgkqhkiG9w0BAQQFADCBhzELMAkGA1UEBhMCWkEx
IjAgBgNVBAGTGUZPUiBURVNUSU5HIFBVULBPU0VTIE90TFkxHTAbBgNVBAoTFFRo
YXd0ZSBDZXJ0aWZpY2F0aW9uMRcwFQYDVQQLew5URVNUIFRFU1QgVEVTVDEcMBoG
A1UEAxMTVGhhd3RlIFRlc3QgQ0EgUm9vdAeFw05NjA4MDEwMDAwMDBaFw0yMDEy
MzEyMTU5NTlaMIGHMQswCQYDVQQGEwJaQTEiMCAGA1UECBMZRk9SIFRFU1RJTkcg
UFVSUE9TRVMgT05MWTEDMBsGA1UEChMUUVGhhd3RlIENlc3RlcnRlYXRpb24xZzAV
BgNVBAsTDlRFU1QgVEVTVCBURVNURwwGgYDVQQDExNUaGF3dGUgVGVzdCBDQSBS
b290MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC1fZBvjr0sfwzoZvrSLEH8
1TFhorPebBZhLZDDE19mYuJ+ougb86EXieZ487dSxXKruBFJPSYttHoCin5qkc5k
BSz+/tZ4knXyRFB03CmONEKCPfdu9D06y4yXmjHApfgGJfpA/kS+QbbiilNz7q2H
LArK3umk74zHKqUyThnkjwIDAQABoxMwETAPBgNVHRMBAf8EBTADAQH/MA0GCSqG
```

```

SIb3DQEBAUAA4GBAIKM4+wZA/TvLIItldL/hGf7exH8/ywvMupg+yAVM4h8uf+d8
phgBi7coVx71/lCB0lFmx66NyKlZK5mObgvd2dlnsAP+nnStyhVHFIPky3nsD04J
qrIgEhCsdpikSpbtdo18jUubV6z1kQ71CrRQtbi/WtdqxQEEtgZCJ02lPoIW
-----END CERTIFICATE-----

```

Procedure

1. From the Key database menu, enter option 6 to store a CA certificate.

```
Key database menu
```

```
Current key database is /u/ibmuser/key.kdb
```

```

1 - List/Manage keys and certificates
2 - List/Manage request keys
3 - Create new key pair and certificate request
4 - Receive a certificate issued for your request
5 - Create a self-signed certificate
6 - Store a CA certificate
7 - Show the default key
8 - Import keys
9 - Export keys
10 - List all trusted CAs

11 - Store encrypted database password
0 - Exit program

```

2. Enter the file name where the certificate request is stored.

```
Enter certificate file name or press ENTER for "cert.arm":
```

If you do not enter a file name, the certificate is stored in the current working directory under the name `cert.arm`.

3. Enter a label that describes the certificate. This label is not used within the Platform Server.

A message is displayed indicating that the key manager is processing the request.

```
Please wait while certificate is stored...
```

When the request is completed, the following message is displayed. You can continue processing by entering 0, or terminate by entering 1.

```
Your request has completed successfully, exit gskkyman? (1=yes,
0=no):
```

Result

At this point, you have successfully received the CA certificate and you can now receive certificates for this CA.

Creating New Key Pair and Certificate Request

You can use option 3 from the Key database menu to create a new key pair and a certificate request.

Procedure

1. From the Key database menu, enter option 3 to create a new key pair and a certificate request.

2. Enter the file name where the certificate request is stored.

If you do not enter a file name, the certificate is stored in the current working directory under the name `certreq.arm`.

3. Enter a label for this key.

i Note: The label name is case sensitive. The label is important because it is used in the Platform Server GLOBAL `SSL_DNLABEL` parameter. If you want to use this parameter, you must enter it without any embedded spaces.

4. Enter the desired key strength.

If you do not enter a key strength, the default value 512 is used as the key strength.

```
Select desired key size from the following options (512):
```

```
1: 512
```

```
2: 1024
```

```
Enter the number corresponding to the key size you want:
```

5. Enter the following certificate subject name fields.

- Common Name: Typically, the name of the user or machine where the key is used.

This field is required.

- **Organization:** Typically, the name of the company or organization. This field is required.
- **Organization Unit:** The name of the department. This field is optional.
- **City/Locality:** The city where you are located. This field is optional.
- **State/Province:** The state where you are located. This field is optional.
- **Country Name:** 2 characters of the country where you are located. This field is required.

The following message is then displayed indicating that the request is being created.

```
Please wait while key pair is created...
```

When the key is created, the following message is displayed. You can enter 0 to exit gskkyman.

```
Your request has completed successfully, exit gskkyman? (1=yes,
0=no):
```

Result

At this point, you can edit the file where the certificate request is created. You can use the TSO OEDIT command to edit the dataset. See the following example of a certificate request file:

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBKzCB1gIBADBxMQswCQYDVQQGEwJVUzELMAkGA1UECBMCTlkxFDASBgNVBAcT
C0dhcmRlbjBDbXR5MREwDwYDVQQKEWhQcm9naW5ldDEUMBIGIA1UECxMLRGV2ZWxv
cG1lbnQxZjAUBgNVBAMTDVByb2dpbmV0IFVzZXIwXDANBgkqhkiG9w0BAQEFAANL
ADBIAkEA5g/R9hXIWYe8MJBgNIpn2qB2B1ZT23URKBClWd3+yQ0a++tZpYXqmGfN
ojfDYjgLM8WJazMd49nP8r1Fy6fFpQIDAQABoAAwDQYJKoZIhvcNAQEEBQADQQAW
nEnXjl8zstNnZCSFakfJzNdprLBSTtvyBpH30ML0YjS4yIhMQ+hw2f+CMXYgWQWs
4aDWVBZHRvaXzvKFDTlp
-----END NEW CERTIFICATE REQUEST-----
```

You can send the certificate request to the certificate authority, or you can paste it into an email or into a web interface. When the certificate authority has processed the certificate request and has created a certificate, the next step can be processed.

Receiving a Requested Certificate

After the certificate authority has created a certificate, you must save this certificate to a file that is accessible by the `gskkyman` utility. Typically, this means you have to save the CA certificate as an HFS file.

See the following copy of a certificate:

```
-----BEGIN CERTIFICATE-----
MIICUjCCAbuGAWIBAgIDTHQ1MA0GCSqGSIb3DQEBAUAMIGHMQswCQYDVQQGEwJa
QTEiMCAGA1UECBMZrk9SIFRFU1RJTkcGUFVSUE9TRVMgT05MWTEdMBsGA1UEChMU
VGhhd3RlIENlcnRpZmLjYXRpb24xZzAVBgNVBAsTDlRFU1QgVEVTVCBURVNUMRww
GgYDVQQDExNUaGF3dGUgVGZzdCBDQSBSb290MB4XDTAxMDgyODE2MzQyM1oXDTAx
MDkxODE2MzQyM1owcTELMakGA1UEBhMCMVVMxCzAJBgNVBAGTAk5ZMRQwEgYDVQQH
EwTHYXJkZW4gQ2l0eTERMA8GA1UEChMIUHJvZ2luZXQxZDASBgNVBAsTC0RldmVs
b3BtZW50MRYwFAYDVQQDEw1Qcm9naW5ldCBVc2VyMFwwDQYJKoZIhvcNAQEBBQAD
SwAwSAJBA0YP0fYVyFmHvDCQYDSKZ9qgdgdWU9t1ESgQpVnd/skNGvvrWaWF6phn
zaI3w2I4CzPFiWszHePZz/K9RcunxaUCAwEAAMlMCMwEwYDVR0lBAwwCgYIKwYB
BQUHAWewDAYDVR0TAQH/BAIwADANBgkqhkiG9w0BAQQFAA0BgQCKI5hxH+QdzWsG
YzkiWcs0IORwxLMR6BWN7ILSHCU30h2puofn7yGzjb3Z2Jxg/ebryFQeJOISWjWY
d9mrXO9GfU/TPM1Aimc7wbqVvcGEIKwQL0Ayxb25X4BFdb9Rs1wHwqqyxihgTIV9
WQdiuQH+eL8ncP5DfiRGJOGvJoED0w==
-----END CERTIFICATE-----
```

Procedure

1. Enter the key manager program `gskkyman` and select the key database that you want to use.
2. From the Key database menu, you can select option 4 to receive a certificate issued for your request. You are then prompted for information about the certificate.
3. Enter the name of the file where you created the certificate.

```
Enter certificate file name or press ENTER for "cert.arm":
```

If you do not enter a file name, the system uses the default file name `cert.arm`.

4. Respond to the prompt as to whether to set the key as the default in your key database.

```
Do you want to set the key as the default in your key database?
(1=yes, 0=no):
```

If you select yes, this certificate becomes the default certificate for this key database.

You can then specify the GLOBAL SSL_DNLABEL parameter as NULL and the Platform Server uses the default certificate for the key database.

The following message is displayed indicating that the key manager is processing the request.

```
Please wait while certificate is stored...
```

When the request is completed, the following message is displayed. You can continue processing by entering 0 or terminate by entering 1.

```
Your request has completed successfully, exit gskkyman? (1=yes,  
0=no):
```

Result

At this point, you have successfully applied the certificate.

Creating a Key Ring

You must create a key ring if you are running OS/390 2.8 or higher versions.

For more information on creating the RACF key ring using the RACDCERT command, you can review the following IBM documents:

- SC28-1915 *OS/390 Security Server (RACF) Security Administrator's Guide*
- SC28-1919 *OS/390 Security Server (RACF) Command Language Reference*

With most of the steps, you must supply an ID parameter. The ID parameter defines the user ID that is associated with the key ring or certificate.

i Note: The following steps do not address any security or authorization issues regarding the key ring.

Procedure

1. Create a key ring.
2. Generate a self-signed certificate for the user.
3. Add the CA certificate.

4. Connect the CA certificate to the key ring.
5. Generate a certificate request.
6. Add a user certificate.
7. Connect the user certificate to the key ring.

Creating SSL Certificates Using a Certificate Authority

You can use a certificate authority to create SSL certificates.

In the following steps, a user ID of `mftps` is used. If the user ID associated with the Platform Server started task is different from this, you must make the changes to the sample commands.

For more information on the exact format of the RACF commands and keywords, see *z/OS Security Server (RACF) Command Language Reference*.

Procedure

1. Create a key ring.

A key ring is the equivalent of a key database. The key ring name must be specified on the `GLOBAL SSL_KEY_DBNAME` parameter. Therefore, you cannot embed any spaces in the key ring name.

See the following example of RACF command used to create a key ring:

```
RACDCERT ID(mftps) ADDRING(FusionRing)
```

This command adds a key ring called `FusionRing`. This name is case sensitive. When you supply this name to the Platform Server `SSL_KEY_DBNAME` parameter, you must use exactly the same name as is used on the `ADDRING` parameter.

2. Add the certificate authority certificate.

This certificate usually can be found on a web site, or your security administrator can send it to you. This certificate contains the CA public key and is used to verify the certificates that are signed with the private key of CA. This certificate is typically the same for all users that use CA.

See the following example of a certificate authority certificate.

```
-----BEGIN CERTIFICATE-----
MIICmTCCAgKgAwIBAgIBADANBgkqhkiG9w0BAQQFADCBhzELMAkGA1UEBhMCWkEx
IjAgBgNVBAGTGUZPUiBURVNUSU5HIFBVU1BPU0VTIE90TFkxHTAbBgNVBAoTFFRo
YXd0ZSBDZXJ0aWZpY2F0aW9uMRcwFQYDVQQLLEw5URVNUIFRFRU1QgVEVTVDEcMBoG
A1UEAxMTVGhhd3RlIFRlc3QgQ0EgUm9vdDAeFw05NjA4MDEwMDAwMDBaFw0yMDEy
MzEyMTU5NTlaMIGHMQswCQYDVQQGEwJaQTEiMCAGA1UECBMZRk9SIFRFRU1RJTkcG
UFVSUE9TRVMgT05MWTEdMBsGA1UEChMUUVGhhd3RlIENlcnRpZmZlYXRpb24xZzAV
BgNVBAsTDlRFU1QgVEVTVCBURVNUMRwwGgYDVQQDExNUaGF3dGUgVGVzdCBDQSBS
b290MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC1fZBvjr0sfwzoZvrSLEH8
1TFHoRpebBZhLZDDE19mYuJ+ougb86EXieZ487dSxXKruBFJPSYttHoCin5qkc5k
BSz+/tZ4knXyRFB03CmONEKCPfdu9D06y4yXmjHApfgGJfpA/kS+QbbiilNz7q2H
LArK3umk74zHKqUyThnkjwIDAQABoxMwETAPBgNVHRMBAf8EBTADAQH/MA0GCSqG
SIb3DQEBAUAA4GBA1KM4+wZA/TvLItdL/hGf7exH8/ywMupg+yAVM4h8uf+d8
phgBi7coVx71/LCB0lFmx66NyKlZK5m0bgvd2dlnsAP+nnStyhVHFIPky3nsD04J
qrIgeHcspdikSpbtDo18jUubV6z1kQ71CrRQtbi/WtdqxQEetgZCJ02lPoIW
```

i Note: You must save the certificate in a dataset with a RECFM of V or VB. If you specify any other RECFM, the certificate is invalid.

See the following example of RACF command used to add a certificate to the server. The command is listed on multiple lines for clarity purpose.

```
RACDCERT CERTAUTH
ADD(my.certca.dsn)
WITH(LABEL('CertAuth'))
```

At this point, the CA certificate has been added to the system.

3. Generate a self-signed certificate for the server.

Before you can generate a certificate request, you must create a self-signed certificate for the server. This certificate is internal to RACF and contains all the information required to create a certificate request.

See the following example of RACF command used to create a self-signed certificate for the server. The command is listed on multiple lines for clarity purpose.

```
RACDCERT ID(mftps)
GENCERT
SUBJECTSDN(CN('domain.name')
            OU('organization unit'))
```

```

O('Organization')
SP('State/Province')
L('City or Locality')
C('xx')                xx = 2 byte country name
WITHLABEL('fusionCert')

```

This command has no output. [step 4](#) uses the certificate created in this step when generating a certificate request.

4. Generate a certificate request.

To create a certificate, you must present a certificate request to the certificate authority. This step produces a certificate request based on the self-signed certificate that you created in [step 3](#).

See the following example of RACF command used to create a certificate request for the server. The command is listed on multiple lines for clarity purpose.

```

RACDCERT ID(mftps)
GENREQ(LABEL('fusionCert'))
DSN(my.cert.req.dsn)

```

The output of this certificate request is a dataset with a certificate with Base64 encoding. See the following example of a certificate request.

```

-----BEGIN NEW CERTIFICATE REQUEST-----
MIIB4TCCAuoCAQAwcTElMAkGA1UEBhMCVVMxETAPBgNVBAGTCE5ldyBZb3JrMRQw
EgYDVQQHEwtHYXJkZW4gQ2l0eTERMA8GA1UEChMIUHJvZ2luZXQxZDASBgNVBASt
C0RldmVsb3BtZW50MRAwDgYDVQQDEwdQU1NHUkVHMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQDhiVQR+uODpNC9wPUk8bzBiGumyQfJPEUxVeU9p+nVZVvkFOuG
V9A0HRqXEkrk8TPG7/iVQfmLT4M6nuBD6kJbtxcVzG5YCuFyhF+xJ7wGkbgxnW
7YCxQLLgH8l3Ao0R4f/4GjEJYqHfnL8X0+K00cAyIzQ2k2o5VP7U9XaVwwIDAQAB
oDAwLgYJKoZIhvcNAQkOMSEwHzAdBgNVHQ4EFgQUFpgJNAq12G5fT+TJ3sLuY2hq
MYyWdQYJKoZIhvcNAQEFBQADgYEApQ0fypfL9nASF8Qe3x9jaiAeAc7zKc/jyIq+
lvPABJf8pEFPH02XjYwPKgEqw9cFlskOpGVm05FRUs+tWGl09Fa6WLhv/uVnBseQ
I3Ytf7vbG/zqPV5aAQnSOq4HangZdiBIw3jtw0SNA4KeMtaVdv+sQ5YTYsBvUUxg
3W1Re3R=
-----END NEW CERTIFICATE REQUEST-----

```

At this point, you must send the certificate file to the certificate authority. The exact procedure depends on your organization and the certificate authority that you use. In some cases, you email the request, and in other cases, you cut and paste the

certificate request into a web page.

5. Add the certificate request.

After the certificate authority processes the certificate, a certificate is created for you. You must save the certificate in a dataset with a RECFM of V or VB. If you specify any other RECFM, the certificate is invalid.

See the following example of a certificate.

```
-----BEGIN CERTIFICATE-----
MIICljCCAf+gAwIBAgIDbhBfMA0GCSqGSIb3DQEBAUAMIGHMQswCQYDVQQGEwJa
QTEiMCAGA1UECBMZrk9SIFRFU1RJTkcglUFVSUE9TRVMgT05MWTEdMBsGA1UEChMU
VGhhd3RlIENlcnRpZmlyYXRpb24xZzAVBgNVBAsTDlRFU1QgVEVTVCBURVNUMRww
GgYDVQQDEwNUaGF3dGUgVGVzdCBDQSBz290MB4XDTAxMDgyOTE1Mzg1N1oXDTAx
MDkxOTE1Mzg1N1owcTElMAkGA1UEBhMCMVVMxETAPBgNVBAGTCE5ldyBZb3JrMRQw
EgYDVQQHEwTHYXJkZW4gQ2l0eTERMA8GA1UEChMIUHJvZ2luZXQxZDASBgNVBAsT
C0RldmVsb3BtZW50MRAwDgYDVQQDEwdQU1NHUkVHMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQDhiVQR+u0DpNC9wPuk8bzBiGumyQfJPEUxVeU9p+nVZVvkFOuG
V9A0HRqXEkrk8TPG7/iVQfmLT4M6nuBD6kJbtxcVzG5YCuFyhF+xJ7wGkbgenXnW
7YCxQLLgH8l3Ao0R4f/4GjEJYqHfnL8X0+K00cAyIzQ2k2o5VP7U9XaVwwIDAQAB
oyUwIzATBgNVHSUEDDAKBggrBgEFBQcDATAMBgNVHRMBAf8EAjAAMA0GCSqGSIb3
DQEBAUAA4GBADu1bS4s7f6v6Yfp8DUA7iiTG8X10/iPQBCOUdg8hT2J/MhM2Uq1
C1pnnONIS1xFWzTH9fBwd5TzDFbUdqQTbwyBex/BsxQkrfSLs0Nz6rzDNoraMvdR
DMcwxatAT6YqxIs8KBosSlPLIoBNS510KqD6R91/qKSxru58kG63j0e1
-----END CERTIFICATE-----
```

See the following example of RACF command used to add certificate for the server. The command is listed on multiple lines for clarity purpose.

```
RACDCERT ID(mftps)
ADD(my.cert.req.dsn)
WITH(LABEL('fusionCert'))
```

At this point, the certificate has been added to the system under user ID mftps.

6. Connect the certificate to the key ring.

To use the certificate, you must connect it to the key ring.

See the following example of RACF command used to connect the certificate to the key ring. The command is listed on multiple lines for clarity purpose.

```
RACDCERT ID(mftps)
```

```
CONNECT(ID(mftps)
LABEL('fusionCert')
RING(FusionRing)
DEFAULT)
```

Note: The DEFAULT parameter makes this certificate the default certificate for a key ring. You can override this parameter by using the GLOBAL SSL_DNLABEL and SSL_CLIENT_DNLABEL parameters.

7. Connect the CA certificate to the key ring.

The CA certificate is created in [step 2](#). You must connect it to the key ring if you want certificates to be authenticated with that CA.

See the following example of RACF command used to connect the CA certificate to the key ring. The command is listed on multiple lines for clarity purpose.

```
RACDCERT ID(mftps)
CONNECT(certauth
LABEL('CertAuth')
RING(FusionRing))
```

Result

At this point, you have a certificate that can be used to connect to other systems. You also have a certificate authority certificate that can be used to authenticate certificates received from other systems. The certificate processing has been completed, and you can start to use the Platform Server SSL on z/OS.

Creating SSL Certificates Using RACF as a Certificate Authority

You can use RACF as the certificate authority to create certificates.

In the following steps, a user ID of mftps is used. If the user ID associated with the Platform Server started task is different from this, you must make the changes to the sample commands.

For more information on the exact format of the RACF commands and keywords, see *z/OS Security Server (RACF) Command Language Reference*.

Procedure

1. Create a key ring.

A key ring is the equivalent of a key database. The key ring name must be specified on the GLOBAL SSL_KEY_DBNAME parameter. You cannot embed any spaces in the key ring name.

See the following example of RACF command used to create a key ring:

```
RACDCERT ID(mftps) ADDRING(FusionRing)
```

This command adds a key ring called FusionRing. This name is case sensitive. When you supply this name to the Platform Server SSL_KEY_DBNAME parameter, you must use exactly the same name (case sensitive) as is used on the ADDRING parameter.

2. Generate a locally-signed CA certificate for the server.

This request generates a self-signed certificate that represents the local RACF certificate authority.

See the following example of RACF command used to create a self-signed CA certificate for the server. The command is listed on multiple lines for clarity purpose.

```
RACDCERT
CERTAUTH
GENCERT
SUBJECTSDN(CN('domain.name')
            OU('organization unit')
            O('Organization')
            SP('State/Province')
            L('City or Locality')
            C('xx'))          xx = 2 byte country name
WITHLABEL('ProgCertAuth')
```

This command has no output. The RACDCERT EXPORT command can be used to create a file containing the certificate authority certificate. The RACDCERT GENCERT can be used to generate a certificate.

3. Generate a locally-signed certificate for the server.

This request generates a self-signed certificate for a user.

See the following example of RACF command used to create a self-signed certificate for the server. The command is listed on multiple lines for clarity purpose.


```

RACDCERT ID(mftps)
GENCERT
SUBJECTSDN(CN('user name')
            OU('organization unit')
            O('Organization')
            SP('State/Province')
            L('City or Locality')
            C('xx'))          xx = 2 byte country name
WITHLABEL('CFusionLocalCert')
SIGNWITH(CERTAUTH LABEL('ProgCertAuth'))

```

This step actually creates a certificate for user mftps.

i Note:

- The label name defined in the SIGNWITH parameter must match the label name created by the WITHLABEL parameter on the step that created the CA certificate.
- The label defined in this step by the WITHLABEL parameter must exactly match the label specified in the GLOBAL SSL_DNLABEL or SSL_CLIENT_DNLABEL parameters.

4. Connect the certificate to the key ring.

To use the certificate, you must connect it to the key ring.

See the following example of RACF command used to connect the self-signed certificate to the key ring. The command is listed on multiple lines for clarity purpose.

```

RACDCERT ID(mftps)
CONNECT(ID(mftps)
LABEL('CFusionLocalCert'
RING(FusionRing)
DEFAULT)

```

- i Note:** The DEFAULT parameter makes this certificate the default certificate for a key ring. You can override this parameter by using the GLOBAL SSL_DNLABEL and SSL_CLIENT_DNLABEL parameters.

5. Connect the self-signed CA certificate to the key ring.

The self-signed CA certificate is created in [step 2](#). You must connect it to the key ring if you want certificates to be authenticated with that CA.

See the following example of RACF command used to connect the CA certificate to the key ring. The command is listed on multiple lines for clarity purpose.

```
RACDCERT ID(mftps)
CONNECT(certauth
LABEL('ProgCertAuth')
RING(FusionRing))
```

At this point, you have a certificate that can be used to connect to other systems. You also have a certificate authority certificate that can be used to authenticate certificates received from other systems.

6. Export the CA certificate to a file.

The self-signed CA certificate is created in [step 2](#). For other systems to accept any certificates signed by this CA, they must have a copy of the CA certificate. This step creates a dataset that contains a Base64 encoded certificate file.

See the following example of RACF command used to export the CA certificate to a dataset. The command is listed on multiple lines for clarity purpose.

```
RACDCERT CERTAUTH
EXPORT(LABEL('ProgCertAuth'))
DSN(z/OS dataset name)
FORMAT(CERTB64)
```

The output of this command is a dataset that contains the CA certificate. This certificate must be provided to any system that has to authenticate certificates created by this CA. See the following example of a self-signed CA certificate file.

```
-----BEGIN CERTIFICATE-----
MIICizCCAfSgAwIBAgIBADANBgkqhkiG9w0BAQUFADBCMqswCQYDVQQGEwJVUzER
MA8GA1UEChMIUHJvZ2luZXQxIDAeBgNVBAsTF1Byb2dpbmV0IExvY2FsIENlcuRB
dXR0eW00aXDTAxMDgyOTIzMDMyMVoXDTAyMDgzMDIzMDMyMFowQjELMAkGA1UEBhMC
VVMxETAPBgNVBAoTCFByb2dpbmV0MSAwHgYDVQQLExdQcm9naW5ldCBMb2NhbCBD
ZXJ0QXV0aDCBnzANBgkqhkiG9w0BAQEFAAOBjQAwGyKCGYEA4m3owLs86h2URKAL
3W4iU5yR55RKDB3PRUjeQkXmRkZtgTRam3Gfr8ygFCr4UEHViE/HjUZeoLU2YhMi
/SuoePpFHM8LJVrsYRNxkmJ3ImhBJVlh/yvUniYiuNjuqdQCz6zVMbVOC5lqUnc6
M028tzsPvcvUwcNMh2QTXfMrGzsCAwEAAaOBkDCBjTBLBglVHQ8BhvhCAQ0EPhM8
```

```
R2VuZXJhdGVkIGJ5IHRoZSBTZWN1cmVXYXkgU2VjdXJpdHkgU2VydmVyIGZvc iBP
Uy8zOTAgKGFJBQ0YpMA4GA1UdDwEB/wQEAwIABjAPBgNVHRMBAf8EBTADAQH/MB0G
A1UdDgQWBRCd6LHrj2lR1ww/09RuM/trNGcOjANBgkqhkiG9w0BAQUFAA0BgQCI
HviGIZgcs8QVPuWqnLTXUE0CNnKrTRZKrIs19XF1mh0/Mj6PtCLPANK5Lyq9tEE
eedn3tjrB8qn72aZAame9q0t7LiShDcqIqIt9Em2/0PiW98IEPFOS0/YsuJpfv7X
c05z8hvKyYCCMwnG6PoLyLLh4TRYX3IfLTsYuqARxU==
-----END CERTIFICATE-----
```

At this point, you have created a certificate authority that can create certificates.

7. Create a user certificate based on a certificate request.

If a system requires a certificate, then you must create a certificate request. RACF then takes the certificate request, and creates a certificate for the user. For this step to be completed, you must have created a certificate request in a z/OS file with a RECFM of VB.

See the following example of RACF command used to create a certificate for a user. The command is listed on multiple lines for clarity purpose.

```
RACDCERT ID(userid)
GENCERT(z/OS cert request DSN)
WITHLABEL('CFusionLocalCert')
SIGNWITH(CERTAUTH LABEL('ProgCertAuth'))
```

This step actually creates a certificate for user mftps.

i Note: The label name defined in the SIGNWITH parameter must match the label name created by the WITHLABEL parameter that created the CA certificate in [step 2](#). However, the certificate must be exported to a file before it can be sent to a user.

8. Export the user certificate to a file.

The self-signed user certificate is created in [step 7](#). For the system to use this certificate, you must have a copy of the certificate. This step creates a dataset that contains a Base64 encoded certificate file.

See the following example of RACF command used to export the user certificate to a dataset. The command is listed on multiple lines for clarity purpose.

```
RACDCERT id(mftps)
```

```
EXPORT(LABEL('CFusionLocalCert'))
DSN(z/OS dataset name)
FORMAT(CERTB64)
```

The output of this command is a dataset that contains the CA certificate. This certificate must be provided to any system that has to authenticate certificates created by this CA. See the following example of a self-signed user certificate file.

```
-----BEGIN CERTIFICATE-----
MIICujCCAi0gAwIBAgIBAzANBgkqhkiG9w0BAQUFADBCMqswCQYDVQQGEwJVUzER
MA8GA1UEChMIUHJvZ2luZXQxIDAeBgNVBAsTF1Byb2dpbmV0IEExvY2FsIENlcuRB
dXR0MB4XDTAxMDgzMTAwMDAwMFOxDTAyMDgzMTIzNTk1OVowcTElMAkGA1UEBhMC
VVMxETAPBgNVBAGTCE5ldyBZb3JrMRQwEgYDVQQHEwTHYXJkZW4gQ2l0eTERMA8G
A1UEChMIUHJvZ2luZXQxFDASBgNVBAsTC0RldmVsb3BtZW50MRAwDgYDVQQDEwdQ
U1NHUkVHMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDhiVQR+uODpNC9wPUK
8bzBiGumyQfJPEUxVeU9p+nVZVvkFOuGV9A0HRqXEkRk8TPG7/iVQfmLT4M6nuBD
6kJbtxcVzG5YCuFyhF+xJ7wGkbgxnW7YCxQLLgH8l3Ao0R4f/4GjEJYqHfnL8X
0+K00cAyIzQ2k2o5VP7U9XaVwwIDAQABo4GQMIGNMESGCWCGSAGG+EIBDQQ+EzxH
ZW5lcmF0ZWQgYnkgdGhlfIFNLY3VyZVdheSBTZWN1cm10eSBTZXJ2ZXIgzM9yIE9T
LzM5MCAoUkFDRikwHQYDVR0OBBYEFBaYCTQKtdhuX0/kyd7C7mNoajGGMB8GA1Ud
IwQYMBaAFEJ3oseuPaVHXDD/T1G4z+2s0Zw6MA0GCSqGSIb3DQEBBQUAA4GBAAb6
JP75hQ0ssZrvZxXrCqbr0/m1bK7JzBXY26MgE7qB7QqFO+Eo84s0yD8QjIKMa4fT
gpfBibGYU4aJqHXdcWG1xt/gWU18TgVMkIgsnIDMKtgHr5u8t0AJsIHxY3pHEequ
gL/3+hHqI1G+WYhBNCcHpBdBWhrg04hGNjrIYeFn
-----END CERTIFICATE-----
```

Result

At this point, you have created a certificate authority and certificate that can be used to connect to other systems. You also have created a certificate authority certificate that can be used to create certificates for other systems. The certificate processing is completed, and you can now start to use the Platform Server SSL on z/OS.

Other Useful RACDCERT Commands

This section explains some additional RACDCERT commands that you might find useful.

For more information on the exact format of the RACF commands and keywords, see *z/OS Security Server (RACF) Command Language Reference*.

Checking Validity of Certificate

You can use the CHECKCERT command to determine if a certificate is valid. The CHECKCERT command can only check certificates. The z/OS file that contains the certificate must be defined as RECFM=VB. Only certificates that contain the following data in the first line can be checked:

```
-----BEGIN CERTIFICATE-----
```

See the following example of RACF command to check a certificate:

```
RACDCERT CHECKCERT(z/OS dataset)
```

Listing Contents of Key Ring

You can use the LISTRING command to list the contents of a key ring. Information about the key ring as well as certificates defined in the key ring can be displayed.

See the following example of RACF command to list a key ring:

```
RACDCERT ID(userid) LISTRING(*)           Displays info on ALL key rings
RACDCERT ID(userid) LISTRING(FusionRing)   Displays info on FusionRing
```

Listing Contents of Certificate

You can use the LIST command to list the contents of a certificate.

See the following example of RACF command to list a key ring:

```
RACDCERT ID(userid) LIST Displays info on ALL certificates for the user
RACDCERT ID(userid) LIST(label('fusionCert')) Displays info on fusionCert
certificate
```

Authorizing User to Key Ring

Some versions of z/OS require that the Platform Server user ID be granted access to a RACF facility class before using the key ring.

See the following example on how the required RACF command grant access to the RACF facility class IRR.DIGTCERT.LISTRING:

```
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(userid) ACCESS(READ)
```

i **Note:** After executing this RACF command, you must refresh the RACF facility class through the SETROPTS REFRESH command.

Operator Commands

To ensure that the Platform Server is operating with maximum efficiency, MIS or data center staff must become familiar with the operator console commands.

z/OS operator console commands include several different ways to start, stop, modify, and display file transfer operations using the Platform Server.

Command Basics

z/OS console operators must be able to start, stop, modify, and display the Platform Server application, the optional JES SPOOL interface, the initiator, the responder, remote nodes, and all file transfer applications.

In addition, a user can query the status of all resources on the network. The z/OS console is also used to halt specific file transfer work in progress and/or to purge work from the work queue.

In the following topics, you can find information on different operations:

- **System Operations:** Covers the actions console operators can take when working with the Platform Server, including bringing the server up and down and displaying the Platform Server file server contents.
- **Node Operations:** Describes how to enable and disable nodes that are part of the Platform Server file transfer network.
- **Activities:** Consists of other information on activities, such as modifying and displaying files.

System Operations

z/OS console operators must be familiar with the Platform Server operator console commands.

The primary commands used on the operator's console are `start`, `stop`, and `modify`. These commands can be entered in all upper cases or in abbreviated form.

```
START=S
STOP=P
MODIFY=F
```

Console operators must use a space between these commands and the procedure name. After that, a comma is used to separate each component of the command.

For example:

```
F MFTSTC,STOP
```

Starting the Platform Server

To start the Platform Server from the z/OS console, you can use the following z/OS start command:

```
S procedure_name
```

This works when the procedure name is a member of SYS1.PROCLIB that contains the Platform Server startup JCL.

For example, if the startup JCL MFTSTC is in SYS1.PROCLIB:

```
S MFTSTC
```



Note: To start and override a startup PARM, use upper case. For example:

```
S MFTSTC,START='WARM,SSLDEBUG=HIGH'
```

Stopping the Platform Server

You can use several ways to halt file transfer operations and stop a Platform Server. Each of those commands has a slightly different impact on the file transfers currently in progress.

From the following list of ways to shut down the Platform Server, console operators must choose the method that best suits the situation at hand.

- To perform a shutdown, you can type one of the following commands:


```
P MFTSTC or F MFTSTC,STOP
```

These commands cause the Platform Server to reject any new associations or connections and terminate all existing ones. All VTAM and TCP connections are terminated immediately.

- To perform a force stop to interrupt all current file transfers, you can type the following command:

```
F MFTSTC,STOP
```

i Note: You have to type this command twice to complete the command.

- To perform a quiesce shutdown, you can force a graceful shutdown of the Platform Server started task. The syntax of the modify command is:

```
F MFTSTC,QUIESCE
```

When using the QUIESCE modify command, any jobs queued after issuing this command are placed on the work queue but are not released until the started task is brought back up. Any incoming responder requests are rejected and the Platform Server waits until all active transmissions are completed. During this waiting period, the started task continues to accept modify commands. After the active transfers are completed, the Platform Server continues with the normal shutdown. If an active transfer is taking too long, you can issue a modify STOP command.

Displaying Encryption Available for the Platform Server

To display the hardware encryption that the Platform Server can use on this processor, you can use the following ENCRYPT command. Both AES128 or AES256 are supported, though AES 256 is suggested.

```
F MFTSTC,ENCRYPT
```

Node Operations

z/OS console operators have to be familiar with the commands that are used to enable, disable or display the remote systems that are connected to the Platform Server.

See the following table for the commands that you can use:

Command	Description
NODE	<p>Displays a single node that has connections to this server.</p> <p>F MFTSTC,NODE,<i>nodename</i></p>
NODES	<p>Displays all nodes that are currently enabled.</p> <p>F MFTSTC,NODES</p>
ENABLE	<p>Enables a node.</p> <p>F MFTSTC,ENAB,<i>node_member_name</i></p> <div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <p>Note: The <i>node_member_name</i> must be set to the actual name of the node member in the <code>config</code> library. This cannot be the value found in the <code>NODENAME</code> parameter that is defined within a node member. Otherwise, this action fails.</p> </div>
DISABLE	<p>Disables a node.</p> <p>F MFTSTC,DISA,<i>node_member_name</i></p> <p>The <i>node_member_name</i> can be either the name of the node member in the <code>config</code> library or the value found in the <code>NODENAME</code> parameter that is defined within a node member.</p>
INITHOLD	<p>Modifies the defined node definition to turn on the initiator hold flag.</p> <p>This indicates that initiator requests to this node are not dispatched. The <code>NODENAME</code> parameter is required.</p> <p>F MFTSTC,INITHOLD,<i>nodename</i></p>
INITREL	<p>Modifies the node definition to reset the initiator hold flag.</p> <p>This indicates that initiator requests to this node are dispatched. The <code>NODENAME</code> parameter is required.</p> <p>F MFTSTC,INITREL,<i>nodename</i></p>

Command	Description
REFRESH	Refreshes an enabled node definition. F MFTSTC,REFRESH, <i>node_member_name</i>
RESPHOLD	Modifies the node definition to turn on the responder hold flag. This indicates that responder requests from this node are rejected with a recoverable error. The NODENAME parameter is required. F MFTSTC,RESPHOLD, <i>nodename</i>
RESPREL	Modifies the node definition to reset the responder hold flag. This indicates that responder requests from this node are accepted. The NODENAME parameter is required. F MFTSTC,RESPREL, <i>nodename</i>

Activities

z/OS console operators must be able to perform the z/OS display commands and manage file transfer activities.

z/OS Console Display Commands

The z/OS console display command is used to get the information about an executing address space.

Using the console display commands, you can get the following information:

- The started task or job name and the step name
- Whether the job is swapped in, swapped out, or non-swappable
- The user ID assigned to the job
- The started task or job number
- Numbers of the performance group, address space, and domain

To check if the Platform Server is up and running, the console operator must type `D A,jobname`.

See the following sample for the console display:

```

JOBS      M/S      TS USERS      SYSAS      INITS      ACTIVE/MAX VTAM      OAS
00002    00015    00005      00022      00022      00005/00010  00000
  MFTSTC  MFTSTC   STEP1      NSW S      A=005A    PER=NO      SMC=000
          PGN=005  DMN=006    AFF=NONE
          CT=011.304S  ET=02.34.54
          WUID=STC08893  USERID=MFTSTC
          ADDR SPACE  ASTE=07763680
          DSPNAME=OSIQUEUE  ASTE=07791100
          DSPNAME=OSIQLLOOK  ASTE=01C0A900
          DSPNAME=OSINODES  ASTE=01C0A980
          DSPNAME=OSILOGDS  ASTE=01C0A800
          DSPNAME=OSIWMSGs  ASTE=01C0A780

```

Platform Server VTAM Tracing

z/OS console operators can turn the VTAM tracing facility on and off by using modify commands. For more information, see [Appendix B. Running Traces](#).

The following DD statement must be included in the startup JCL to enable the tracing facility:

```
//VTAMTRAC DD SYSOUT=X
```

z/OS console operators can turn the SNA packet tracing facility on and off by using the following modify commands:

- TRON: To start the SNA packet tracing facility

```
F MFTSTC,TRON
```

- TROF: To stop the SNA packet tracing facility

```
F MFTSTC,TROF
```

Platform Server TCP/IP Tracing

z/OS console operators can turn the TCP/IP tracing facility on and off by using modify commands. The following DD statement must be included in the startup JCL to enable the tracing facility:

```
//TCPTRAC DD SYSOUT=X
```

z/OS console operators can turn the TCP/IP tracing facility on and off by using the following modify commands:

- TCPTRON: To start the TCP/IP packet tracing facility

```
F MFTSTC, TCPTRON
```

- TCPTR0F: to stop the TCP/IP packet tracing facility

```
F MFTSTC, TCPTR0F
```

Platform Server Node Tracing

z/OS console operators can turn the TCP/IP tracing facility on and off by using modify commands. The following DD statement must be included in the startup JCL to enable the tracing facility:

```
//TCPTRAC DD SYSOUT=X
```

z/OS console operators can turn the node tracing on for a particular node defined in the CONFIG by using the following modify commands:

- TCPTRON: To start the node tracing facility

```
F MFTSTC, TCPTRON, NODENAME
```

- TCPTR0F: To stop the node tracing facility

```
F MFTSTC, TCPTR0F, NODENAME
```

File Transfer Activities

z/OS console operators can perform a variety of tasks by issuing specific commands when running the Platform Server.

For example, the console operator can display information about file transfer activities, end those activities, or interrupt them.

See the following table for the commands required to perform these activities:

Command	Description
ABORT	<p>Terminates file transfers in progress and purges them from the work queue.</p> <p>F MFTSTC,ABORT,<i>activitynumber</i></p>
DISPLAY	<p>Shows all queued activities.</p> <p>F MFTSTC,DISPLAY</p>
HOLD	<p>Makes a particular file eligible for file transfer at a specific time.</p> <p>The file transfer does not begin unless you first release it. For more information, see RELEASE</p> <p>F MFTSTC,HOLD,<i>activitynumber</i></p>
PURGE	<p>Purges work from the work queue.</p> <p>Used only for work not yet started. Otherwise, this is treated as a cancellation.</p> <p>F MFTSTC,PURGE,<i>activitynumber</i></p>
RELEASE	<p>Releases an activity that is in hold state on the work queue.</p> <div style="background-color: #f0f0f0; padding: 5px; margin: 5px 0;"> <p>Note: Activities in any other state cannot be released.</p> </div> <p>F MFTSTC,RELEASE,<i>activitynumber</i></p> <p>The RELEASE command can also be used to print the Platform Server OSIMSGS log file.</p> <p>F MFTSTC,RELEASE,LOG</p>
SUSPEND	<p>Terminates an active transfer and notifies the remote LU that the transfer is suspended.</p> <p>The state of the transfer on the Platform Server that issued the SUSPEND command depends on the type of transfer:</p>

Command	Description
	<ul style="list-style-type: none"> • Initiator transfer: Places the transfer on hold. You must execute a RELEASE command to remove it from a hold state. • Responder transfer: Places the transfer in a communications error state which is retried at the next retry interval.
	F MFTSTC,SUSPEND, <i>activitynumber</i>

Platform Server Console Operator Commands

You can find a complete list of the Platform Server console operator commands in this section.

See the following table for the available console operator commands.

Command (Minimum Specification)	Description
ABORT (ABO)	<p>Terminates file transfers in progress and purges them from the Platform Server work queue.</p> <p>F MFTSTC,ABORT or F MFTSTC,ABORT,<i>activitynumber</i></p>
AUTHDISP	<p>Displays the 16 hex digit local authorization key that is defined for this system.</p>
CFACCESSREFRESH	<p>Refreshes the CFACCESS configuration file.</p> <p>This command is automatically executed at startup.</p> <p>F MFTSTC,CFACCESSREFRESH</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>Note: If you make a change to the configuration file after the Platform Server is started, you must execute this command to refresh the CFACCESS configuration file.</p> </div>
CFALIASREFRESH	<p>Refreshes the CFALIAS configuration file.</p>

Command (Minimum Specification)	Description
	<p>This command is automatically executed at startup.</p> <p>F MFTSTC,CFALIASREFRESH</p>
	<p>Note: If you make a change to the configuration file after the Platform Server is started, you must execute this command to refresh the CFALIAS configuration file.</p>
CFPING	<p>Connects to a remote Platform Server node to check if the node is active, and also displays information about the responder license key expiration as well as the Platform Server version information.</p> <p>If you specify the node name in this command, the Platform Server extracts the IP address/IP name and IP port information from the node definition.</p> <p>If you specify the IPADDR or IPNAME parameter, the IPPORT parameter is optional. If not specified, the default Platform Server port 46464 is used.</p> <p>Note: This command only works through TCP and when the remote system is running TIBCO MFT Platform Server with version 6.0 or higher.</p> <p>F MFTSTC,CFPING,<i>nodename</i></p> <p>F MFTSTC,CFPING,<i>ipaddr:ipport</i></p> <p>F MFTSTC,CFPING,<i>ipname:ipport</i></p> <p>CFPING functions can also be called through z/OS JCL. The following are two samples of how to execute the FUSPING job. The FUSPING job is located in the Platform Server library MFT.JCL (PINGJCL).</p>

Command (Minimum Specification)	Description
	<pre data-bbox="532 394 1122 995"> //S1 EXEC PGM=FUSPING,PARM='HOST=127.0.0.1:46464' //STEPLIB DD DSN=MFT.LOADLIB,DISP=SHR // /* The following DD statements are optional. /* To use them, remove the // card after the STEPLIB //SYSPRINT DD SYSOUT=* //SYSIN DD * HOST=127.0.0.1:46464 // </pre> <ul data-bbox="548 1058 1200 1598" style="list-style-type: none"> • The SYSPRINT and SYSIN DD statements are not required. • If SYSIN is not defined, the input parameters are read from the PARM statement. If SYSIN is defined, the input parameters are read from the SYSIN DD and they override parameters defined on the PARM statement. • If SYSPRINT is not defined, the output is displayed on the console; otherwise, the output is displayed in the SYSPRINT DD statement. If SYSPRINT points to a dataset, the dataset must be defined as RECFM=FB or RECFM=F WITH LRECL=133. <p data-bbox="500 1633 1094 1703">See the following sample output of the CFPING command:</p>

Command (Minimum Specification)	Description
	<pre> PGTS2459I FUSPING Host: 127.0.0.1 PGTS2459I FUSPING Port: 46464 PGTS2459I FUSPING System Name: Name=PROD,STC=MFT , CPUType=1234,CPUID=ABCD PGTS2459I FUSPING Key Expiration: 20401231 PGTS2459I FUSPING Version: MFT Platform Server z/OS,Version=720 ,PTFLevel=CZ01963 </pre>
CFSTATS	<p>Prints statistics to the Platform Server log file.</p> <p>Statistics are also printed when the Platform Server STC is terminated.</p> <p>F MFTSTC,CFSTATS</p>
COS	<p>Displays the Platform Server Class of Service (COS) entries.</p> <p>When entered without any additional parameters, a list of enabled COS entries is displayed. When entered with an enabled COS entry, more detailed information about the COS entry is displayed.</p> <p>F MFTSTC,COS: displays enabled COS entries.</p> <p>F MFTSTC,COS,COSHIGH: displays detailed information on COSHIGH entry.</p>
DNIDISA	<p>Disables a DNI entry that is enabled at startup or through the DNIENA command.</p> <p>For more information, see the DNI chapter of <i>TIBCO® Managed File Transfer Platform Server for z/OS User's</i></p>

Command (Minimum Specification)	Description
	<i>Guide.</i>
DNIENA	<p>Enables a DNI entry.</p> <p>For more information, see the DNI chapter of <i>TIBCO Managed File Transfer Platform Server for z/OS User's Guide</i>.</p>
DNIDISP	<p>Displays the DNI entries on the DNI active queue.</p> <p>F MFTSTC, DNIDISP: for summary information on all requests in the DNI active queue</p> <p>F MFTSTC, DNIDISP, dnitrn: for summary information on a single request in the DNI active queue</p>
DNINODE	<p>Displays the information about DNI definitions.</p> <p>If no parameters are specified, a summary display lists all DNI entries that are currently enabled.</p> <p>If a parameter is specified, detailed information about that DNI entry is displayed on the console. When a detailed display is requested, the DNI node entered on the command must be enabled.</p> <p>F MFTSTC, DNINODE: for a summary display</p> <p>F MFTSTC, DNINODE, <i>nodename</i>: for a detailed display on the DNI node</p>
DNIPURGE	<p>Purges a request from the DNI active queue.</p> <p>F MFTSTC, DNIPURGE, dnitrn</p> <p>For more information, see the DNI chapter of <i>TIBCO® Managed File Transfer Platform Server for z/OS User's Guide</i>.</p>

Command (Minimum Specification)	Description
DISABLE (DISA)	<p>Disables the ability to queue transactions to specific remote nodes through the Platform Server or z/OS.</p> <p>F MFTSTC,DISABLE,<i>nodename</i></p>
DISPLAY (DISP)	<p>Shows all activities that are on the work queue.</p> <p>F MFTSTC,DISPLAY</p>
ENABLE (ENA)	<p>Allows the queuing of transactions to a specific remote node.</p> <p>F MFTSTC,ENABLE,<i>nodename</i></p>
FREEDSN	<p>Deallocates a dataset that has not been deallocated successfully.</p> <p>This might be used in case a request terminates and a file is not deallocated correctly.</p> <p>F MFTSTC,FREEDSN,<i>data_set_name</i></p>
HOLD	<p>Makes a particular file eligible for file transfer at a specific time.</p> <p>That file transfer is not completed unless you first release it.</p> <p>F MFTSTC,HOLD,<i>activitynumber</i></p> <p>For more information, see the RELEASE parameter.</p>
INITIATORSTART	<p>Starts initiator tasks after INITIATORSTOP has been specified on the GLOBAL file or through the INITIATORSTOP operator command.</p> <p>F MFTSTC,INITIATORSTART: starts processing initiator requests.</p>
INITIATORSTOP	<p>Stops initiating any file transfer requests.</p>

Command (Minimum Specification)	Description
	<p>F MFTSTC,INITIATORSTART: stops processing initiator requests.</p> <p>You can use the INITIATORSTART operator command to start initiator tasks again.</p>
MODVER	<p>Displays the module date and time compiled along with the hotfix level for a particular module.</p> <p>F MFTSTC,MODVER,FUSASNA</p> <div data-bbox="505 747 1192 926" style="background-color: #f0f0f0; padding: 10px;"> <p>Note: This information is not available for all modules. MFT Technical Support may ask you to enter this command. Otherwise, you generally do not need to execute it.</p> </div>
PURGE (PUR)	<p>Purges work from the work queue.</p> <p>This command is primarily used for work not yet in the process of file transfer.</p> <p>F MFTSTC,PURGE,<i>activitynumber</i></p>
REFRESH (REF)	<p>Refreshes a node definition that is currently enabled.</p> <p>This command performs the same action that the DISABLE/ENABLE commands perform except that the refresh is done in place. The node is never actually disabled. If the node specified on the REFRESH command is not enabled, the request fails.</p> <p>F MFTSTC,REF,<i>nodename</i></p>
RELEASE (REL)	<p>Releases a file for transfer purpose or to print the Platform Server OSIMSG log file.</p> <p>F MFTSTC,RELEASE,<i>activitynumber</i> or F MFTSTC,RELEASE,LOG</p>

Command (Minimum Specification)	Description
RESET	<p>Resets errors on the Platform Server system, so that the request can be retried.</p> <p>If no parameters are specified, all nodes and transfer requests that are in error are reset.</p> <p>If a parameter is specified, errors for that node and all transfer requests that are using that node are reset.</p> <p>F MFTSTC,RESET: resets all errors.</p> <p>F MFTSTC,RESET,<i>nodename</i>: resets errors for this node and for all transfer requests using this node.</p>
RESPONDERSTART	<p>Starts responder tasks after RESPONDERSTOP has been specified on the GLOBAL file or through the RESPONDERSTOP operator command.</p> <p>F MFTSTC,RESPONDERSTART: starts processing responder requests.</p>
RESPONDERSTOP	<p>Stops processing responder requests.</p> <p>You can use the RESPONDERSTART operator command to start responder tasks again.</p> <p>F MFTSTC,RESPONDERSTOP: stops processing responder requests.</p>
SNAP	<p>Produces a SNAP dump copy of the Platform Server private storage and dataspace.</p> <p>F MFTSTC,SNAP</p>
SSLAUTHREFRESH	<p>Refreshes the SSLAUTH configuration file.</p> <p>This command is automatically executed at startup.</p> <p>If you make a change to the configuration file after the Platform Server is started, you must execute this</p>

Command (Minimum Specification)	Description
	<p>command to refresh the SSLAUTH configuration file.</p> <p>F MFTSTC, SSLAUTHREFRESH</p>
SSLDEBUG	<p>Determines the type of SSL debugging that is performed.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • NO: SSL debugging is not turned on. • YES: SSL debugging is turned on. • HIGH: Detailed SSL debugging is turned on. <p>F MFTSTC, SSLDEBUG=YES</p>
SUBJCLREFRESH	<p>Refreshes the SUBJCL configuration file.</p> <p>This command is automatically executed at startup.</p> <p>If you make a change to the configuration file after the Platform Server is started, you must execute this command to refresh the SUBJCL configuration file.</p> <p>F MFTSTC, SUBJCLREFRESH</p>
STARTSNA	<p>Starts the Platform Server SNA interface.</p> <p>This command is only entered if the Platform Server SNA interface does not start properly, or is ended by the TERMSNA operator command.</p> <p>Before using this command, you must verify that the VTAM APPLID defined by the GLOBAL APPC-APPLID parameter has a state of CONCT.</p> <p>If this command is executed when the SNA interface is active, a VTAM error message is displayed along with an operator prompt to wait, end or continue. You must be cautious in using this command.</p>

Command (Minimum Specification)	Description
STARTTCP	<p data-bbox="500 373 753 401">F MFTSTC, STARTSNA</p> <hr/> <p data-bbox="500 447 1016 474">Starts the Platform Server TCP interface.</p> <p data-bbox="500 510 1159 617">This command must only be entered if the Platform Server TCP interface does not start properly, or is ended by the TERMTCP operator command.</p> <p data-bbox="500 653 1192 760">If this command is executed when the Platform Server TCP interface is active, an error message is generated. You must be cautious in using this command.</p> <p data-bbox="500 787 753 814">F MFTSTC, STARTTCP</p>
SUSPEND (SUS)	<p data-bbox="500 867 1149 932">Terminates an active transfer but does not purge it from the queue.</p> <p data-bbox="500 961 972 989">F MFTSTC, SUSPEND, <i>activitynumber</i></p>
TCPTRUF	<p data-bbox="500 1041 1036 1068">Disables the TCP/IP packet tracing facility.</p> <p data-bbox="500 1098 737 1125">F MFTSTC, TCPTRUF</p> <p data-bbox="500 1155 1089 1182">Trace by node name is available. For example:</p> <p data-bbox="500 1211 886 1239">F FUSION, TCPTRUF, NODENAME</p>
TCPTRON	<p data-bbox="500 1293 1036 1320">Initiates the TCP/IP packet tracing facility.</p> <p data-bbox="500 1350 737 1377">F MFTSTC, TCPTRON</p> <p data-bbox="500 1407 1089 1434">Trace by node name is available. For example:</p> <p data-bbox="500 1463 870 1491">F FUSION, TCPTRON, NODENAME</p>
TERMSNA	<p data-bbox="500 1545 1016 1572">Stops the Platform Server SNA interface.</p> <p data-bbox="500 1602 1179 1749">This command must only be entered if the VTAM systems programmer wants to refresh the Platform Server VTAM major node definition without bringing Platform Server down. You must be cautious in using</p>

Command (Minimum Specification)	Description
	<p>this command.</p> <p>Generally, the STARTSNA operator command must be executed after the SNA APPLID major node is activated, to re-initialize the Platform Server SNA interface.</p> <p>F MFTSTC,TERMSNA</p>
TLSSTART	<p>Starts the TLS connection. This can only be executed if TLS was started successfully, but is in a stopped state.</p> <p>F MFTSTC,TLSSTART</p>
TLSSTOP	<p>Stops the TLS connection. This can only be executed if TLS was started successfully and is currently active.</p> <p>F MFTSTC,TLSSTOP</p>
TRCLASSREFRESH	<p>Refreshes the transfer class configuration file defined by the TRCLASS DD statement.</p> <p>This command is automatically executed at startup.</p> <p>If you make a change to the configuration file after the Platform Server is started, you must execute this command to refresh the CFALIAS configuration file.</p> <p>F MFTSTC,TRCLASSREFRESH</p>
TRCLASSDISP	<p>Displays the transfer class definitions.</p> <p>You can display the transfer class definition by entering a transfer class, or you can display all transfer classes by not entering a transfer class.</p> <p>F MFTSTC,TRCLASSDISP,<i>trclassname</i></p> <ul style="list-style-type: none"> To display all transfer class definitions:

Command (Minimum Specification)	Description
	<p>F MFTSTC,TRCLASSDISP</p> <ul style="list-style-type: none"> To display transfer class definitions for transfer class SINGLETHREAD: <p>F MFTSTC,TRCLASSDISP,SINGLETHREAD</p>
TROF	<p>Disables the SNA packet tracing facility.</p> <p>F MFTSTC,TROF</p>
TRON	<p>Initiates the SNA packet tracing facility.</p> <p>F MFTSTC,TRON</p>
VERSION (VER)	<p>Displays the exact version of the Platform Server installed.</p> <p>F MFTSTC,VERSION</p>

Troubleshooting

This section provides possible solutions to some issues that you might encounter when using the Platform Server.

Restore Not Possible

Problem: On the RECEIVE command, the following error occurs when the installation dataset has been transformed to the mainframe with an LRECL that is not 80:

```
Restore not possible. Input data is not complete.
The first record is:
Received file appears not to be an Interactive Data Transmission
Facility file.
X'00FF0000'
dataset X.RECEIVE.INVALID.FILE from USERID on N1
Enter copy parameters or 'DELETE' or 'END' +
```

Solution: Make record length fixed block 80.

Allocation Failure

Problem: The following error occurs on the RECEIVE command if the dataset received is already allocated.

```
RECEIVE INDS(prkevin.test.delete.loadlib)

dataset PROD.FT710.LOAD from PRKEVIN on AL
Enter restore parameters or 'DELETE' or 'END' +

RECEIVE command terminated. Output dataset unusable. +
Allocation failure for dataset 'PROD.STABLE.IND$FILE.LOADLIB'
dataset PROD.FT710.LOAD ALREADY IN USE, TRY LATER+
dataset IS ALLOCATED TO ANOTHER JOB OR USER
```

Solution: Use a new dataset for the RECEIVE command.

Rebuilding the Work Queue

Problem: Errors occur when you delete and redefine the work queue file using the DEFQUEUE command.

Solution: The server must be down before you run the DEFQUEUE command.

Trouble with VSAM Files

Problem: Decoding VSAM errors

The Platform Server keeps a machine readable log to keep a record of all past activities. This log is a key sequenced dataset. If there is a problem with the audit file, the server prints a diagnostic field called the RPLFDBWD.

Solution: You must know the following logic when analyzing the VSAM errors:

- The first byte identifies the specific IBM VSAM module (part of z/OS data facility product) which is reporting the error. This does not have meaning to you, but IBM technical staff can tell you what VSAM module this represents and what it is doing.
- The second byte identifies the type of error. The most common user caused error type is X'08' which means logic error.
- The third byte is not used.
- The fourth byte is the error reason code as documented in chapter 4 of the IBM manual: *OS/390/DFP Macro Instructions for datasets*.

See the following examples on how to decode the RPL feedback word.

Example 1: RPLFDBWD 47080048

This is the VSAM request parameter list from the IBM supplied RPL mapping DSECT. It is interpreted as following:

- First byte: Identifies the specific IBM VSAM module that reports the error. The X'47' identifies IDA019R8 which is a VSAM record management routine responsible for keyed I/O as per IBM technical support.
- Second byte: X'08' means that this is a logic error. If an LERAD routine has been specified in the VSAM ACB, it can be driven; however, VSAM LERADs is not used in the Platform Server for z/OS.
- Third byte: Reserved.
- Fourth byte: '48' is the logical error reason code as documented in chapter 4 of the IBM manual: *OS/390 DFP 3.3 Macro Instructions For datasets (SC26-4747)*. Essentially, a return code of X'48' means that the Platform Server tries to do indexed I/O on a non-indexed dataset. For example, the Platform Server treats the dataset as KSDS, but it is actually ESDS or RRDS. This can be the result of an installation error.

Example 2: RPLFDBWD 0A080008

- First byte: The X'0A' identifies the IBM VSAM Module IDA019RA, which detected the error; the module does direct record locate.
- Second byte: The value X'08" indicates a logic error.
- Third byte: Reserved.
- Fourth byte: X'08' means the Platform Server tries to store a record with a duplicate key. This can be caused by the audit file and the work queue falling out of synchronization with each other.

Activating the VTAM Resources

Problem: VTAM resource does not become active.

Solution: Go through the [Defining the VSAM Files](#) and verify that the VTAM resources have been defined properly.

Remote Nodes Are Not Coming Active When the Product Is Started

Problem: You receive message PGTS1131E indicating that MEMBER IS NOT ACTIVATED DUE TO ERRORS.

Solution: Check the remote node configurations and verify that you have specified all of the correct information.

Transaction Does Not Start after It Is Queued.

Problem: Transaction is placed on the queue as inactive.

Solution: Check if the remote node became active when you start up the Platform Server. If not, check the remote node configurations and verify that you have specified all of the correct information. You have to check the eligible time and date to see if it is the current time and date.

Repeated Error Message for the Same Transaction

Problem: You continually receive the same error message on the system console.

Solution: Check if retry limit is zero. If so, the Platform Server is having difficulty performing this transaction and is continually retrying. To avoid this, you can set the retry limit to a non-zero value.

Receiving a SENSE=087D0001

Problem: VTAM cannot find the LU.

```
IST663I BFINIT REQUEST FROM NCP066 FAILED, SENSE=087D0001
IST664I REAL OLU=TIBCO.DANL1I2 REAL DLU=TIBCO.FUSNAPPL
IST889I SID=F2CF2303704E1476
IST894I ADJSSCPS TRIED FAILURE SENSE ADJSSCPS TRIED FAILURE SENSE
IST895I ISTAPNCP 08400007
IST314I END
```

Solution: The most likely cause of this is that the LU is not active. Check if the LUs are all active.

Receiving a SENSE=08570003

Problem: The session is not active.

```
IST663I BFINIT REQUEST FROM NCP066 FAILED, SENSE=08570003
IST664I REAL OLU=TIBCO.DANL1I2 REAL DLU=TIBCO.FUSNAPPL
IST889I SID=F2CF2303704E1708
IST1138I REQUIRED RESOURCE TIBCO.FUSNAPPL NOT ACTIVE
IST314I END
```

Solution: The probable cause is that the Platform Server is not up and running, therefore the APPL LU is not active. Check if the Platform Server address space is up and that the APPL LU is active.

Batch Job Has Been Submitted However It Has Not Been Placed on the Queue

Look in the output from the JCL for any error messages. The following table lists some of the messages that you might see:

Key Number	Error Message	Cause	User Response
PGTB4100E	Parameter not recognized in next record	Most likely the cause is a typo.	Verify that the specified in the line after this message is correct.
PGTB4102E	Invalid parameter value in next record	The value specified for this parameter is not valid.	Check the valid range of values for this parameter.

Key Number	Error Message	Cause	User Response
PGTB4103E	Dataset name not cataloged	The dataset that is specified for this transaction is not cataloged.	Verify that the dataset name is specified correctly.
PGTB4105E	Operand read before a PROCESS card	The parsing program found parameters before your process card.	Verify that you specify the process card correctly and that all parameters follow the process card.
PGTB4106E	Invalid number of operands in next record	The number of operands is outside the valid number.	Verify that you specify this number correctly.
PGTB4107E	No more room left in the server work Queue	The maximum number of transactions are currently on the work queue.	You can either wait for some of the transactions that are on the queue to complete and be removed, or you can re-create the work specifying a larger VSAM file.
PGTB4108E	Fusion server not available	The server to which you specified this transaction to be queued is not currently up and running.	Start the server and re-queue.
PGTB4109E	Security denied access	You do not have authorization to one of the resources that you specified.	Verify that you have authorization to the resources specified in the batch job.

Key Number	Error Message	Cause	User Response
PGTB4115E	No process card in the SYSIN file	The required process card is not found by the parsing routine.	Verify that you correctly specified a process card for this batch job.
PGTB4116E	Prior process not queued - Parameter error	There is an error in one of the parameters specified for the previous process.	Verify that the parameters specified are correct.
PGTB4117E	Required parameter missing in next record	The required parameter in the next record is missing.	Check that you have specified a value in the next parameter field.
PGTB4118E	Required parameter invalid in next record	The required parameter in the next record is invalid.	Check that the parameter is within the valid range.
PGTB4119I	Invalid/Missing Server parm - Default used	The server is not specified or is invalid.	The default server is used to queue this transaction. To queue this transaction to a different server, you can specify the server parameter with the correct server.
PGTB4120I	Invalid Queue effect for activity	Queue effect is specified was invalid for this transaction type.	Check that you specified a valid value.

i **Note:** If you want further technical assistance, you can contact MFT Technical Support through the following contact information.

Within the United States:

TIBCO Technical Support, 200 Garden City Plaza, Garden City, NY 11530, USA

Phone: 516-535-3636

Email: support@tibco.com

Outside of the United States:

Please contact your local TIBCO distributor.

Appendix A. Sample SNA Mode Definition

You can find a sample SNA mode definition that is used when working with the Platform Server.

See the following sample SNA mode definition for your reference.

i Note: These definitions are not required if you are using the Platform Server TCP interface.

```

USERMODE MODETAB
*****
* MODE ENTRIES FOR CYBERFUSION PERFORMANCE BENCHMARKS
*****
FUSN256 MODEENT LOGMODE=FUSN256, LOGON MODE TABLE ENTRY NAME      -
      COS=FAST,                                                    -
      FMPROF=X'13',          FUNCTION MANAGER PROFILE              -
      TSPROF=X'07',          TRANSMISSION SERVICES PROFILE         -
      PRIPROT=X'B0',         PRIMARY LOGICAL UNIT PROTOCOL        -
      SECPRROT=X'B0',        SECONDARY LOGICAL UNIT PROTOCOL -
      COMPROT=X'50B1',       COMMON LOGICAL UNIT PROTOCOL   -
      RUSIZES=X'8585',       RUSIZE SEC/PRI 256/256      -
      PSNDPAC=5,             PRIMARY SEND PACING COUNT   -
      SRCVPAC=5,             SECONDARY RECEIVE PACING COUNT -
      SSNDPAC=5,             SECONDARY SEND PACING COUNT  -
      PSERVIC=X'060200000000000000002C00' PRESENTATION SRVCS
*****
FUSN512 MODEENT LOGMODE=FUSN512, LOGON MODE TABLE ENTRY NAME      -
      COS=FAST,                                                    -
      FMPROF=X'13',          FUNCTION MANAGER PROFILE              -
      TSPROF=X'07',          TRANSMISSION SERVICES PROFILE         -
      PRIPROT=X'B0',         PRIMARY LOGICAL UNIT PROTOCOL        -
      SECPRROT=X'B0',        SECONDARY LOGICAL UNIT PROTOCOL -
      COMPROT=X'50B1',       COMMON LOGICAL UNIT PROTOCOL   -
      RUSIZES=X'8686',       RUSIZE SEC/PRI 512/512      -
      PSNDPAC=5,             PRIMARY SEND PACING COUNT   -
      SRCVPAC=5,             SECONDARY RECEIVE PACING COUNT -
      SSNDPAC=5,             SECONDARY SEND PACING COUNT  -
      PSERVIC=X'060200000000000000002C00' PRESENTATION SRVCS
*****
FUSN1K  MODEENT LOGMODE=FUSN1K, LOGON MODE TABLE ENTRY NAME      -

```

```

COS=FAST,
FMPROF=X'13',
TSPROF=X'07',
PRIPROT=X'B0',
SECPROT=X'B0',
COMPROT=X'50B1',
RUSIZES=X'8787',
PSNDPAC=5,
SRCVPAC=5,
SSNDPAC=5,
PSERVIC=X'060200000000000000002C00' PRESENTATION SRVCS
*****
FUSN2K MODEENT LOGMODE=FUSN2K, LOGON MODE TABLE ENTRY NAME -
COS=FAST,
FMPROF=X'13',
TSPROF=X'07',
PRIPROT=X'B0',
SECPROT=X'B0',
COMPROT=X'50B1',
RUSIZES=X'8888',
PSNDPAC=5,
SRCVPAC=5,
SSNDPAC=5,
PSERVIC=X'060200000000000000002C00' PRESENTATION SRVCS
*****
FUSN4K MODEENT LOGMODE=FUSN4K, LOGON MODE TABLE ENTRY NAME -
COS=FAST,
FMPROF=X'13',
TSPROF=X'07',
PRIPROT=X'B0',
SECPROT=X'B0',
COMPROT=X'50B1',
RUSIZES=X'8989',
PSNDPAC=5,
SRCVPAC=5,
SSNDPAC=5,
PSERVIC=X'060200000000000000002C00' PRESENTATION SRVCS
*****
FUSN8K MODEENT LOGMODE=FUSN8K, LOGON MODE TABLE ENTRY NAME -
COS=FAST,
FMPROF=X'13',
TSPROF=X'07',
PRIPROT=X'B0',
SECPROT=X'B0',
COMPROT=X'50B1',

```

```

RUSIZES=X'8A8A',          RUSIZE SEC/PRI 8192/8192      -
PSNDPAC=5,                PRIMARY SEND PACING COUNT     -
SRCVPAC=5,                SECONDARY RECEIVE PACING COUNT -
SSNDPAC=5,                SECONDARY SEND PACING COUNT   -
PSERVIC=X'060200000000000000002C00' PRESENTATION SRVCS
*
FUSN16K MODEENT LOGMODE=FUSN16K, LOGON MODE TABLE ENTRY NAME -
COS=FAST,                  -
FMPROF=X'13',             FUNCTION MANAGER PROFILE      -
TSPROF=X'07',            TRANSMISSION SERVICES PROFILE -
PRIPROT=X'B0',           PRIMARY LOGICAL UNIT PROTOCOL -
SECPROT=X'B0',           SECONDARY LOGICAL UNIT PROTOCOL -
COMPROT=X'50B1',         COMMON LOGICAL UNIT PROTOCOL  -
RUSIZES=X'8B8B',         RUSIZE SEC/PRI 16K (8 X 2**11) -
PSNDPAC=5,                PRIMARY SEND PACING COUNT     -
SRCVPAC=5,                SECONDARY RECEIVE PACING COUNT -
SSNDPAC=5,                SECONDARY SEND PACING COUNT   -
PSERVIC=X'060200000000000000002C00' PRESENTATION SRVCS
*
*
MODEEND
END

```

Appendix B. Running Traces

To solve a communication problem, TIBCO Support might ask you to take a trace of the activity in error. This appendix describes the two types of traces that you might be asked to take: Platform Server internal trace and VTAM buffer trace.

Running the Platform Server Internal Trace

The Platform Server can trace either VTAM or TCP communications. The following operator commands activate the respective traces:

- VTAM trace: `F FUSION,TRON`
- TCP trace: `F FUSION,TCPTRON`

Traces are turned off by the following operator commands:

- VTAM trace: `F FUSION,TROFF`
- TCP trace: `F FUSION, TCPTROFF`

For more information, see [Operator Commands](#).

Example for Taking a TCP Trace

The following steps illustrate how to take a TCP trace:

1. Turn on the TCP trace.

```
F FUSION,TCPTRON
```

2. Perform the activity that you want to trace.
3. Turn off the TCP trace.

```
F FUSION,TCPTROFF
```

Locations for Trace Data

VTAM trace data is sent to DD statement VTAMTRAC defined in the Platform Server JCL. TCP trace data is sent to DD statement TCPTRAC defined in the Platform Server JCL.

For more information, see [The Startup JCL](#).

Appendix C. Automated Operations

Many organizations have to automate file transfer processes. In some cases, the data that is transferred must be further processed at the remote side.

Many software packages for the host can scan for certain Write To Operator (WTO) messages and perform different transactions based on the contents of the WTO message.

To automate file transfer processes, certain z/OS messages are required to be standardized. To interface with these automation packages, the Platform Server has standardized several messages that are issued for every transaction.

The following four messages are common to all transactions. The information denoted by the brackets ({ }) are replaceable parameters.

Start of Transfer

The following message is issued at the start of every transfer.

```
PGTF3100 {Send/Receive} Activity {Transaction_Number} Started to Remote Node  
{Remote_Node}
```

Where:

- **Send/Receive:** The direction of the dataset on the host system.
For example, if you are sending a file from a Windows system to the host, then the direction that must be specified is *receive*.
- **Transaction_Number:** The number that is assigned to the transfer.
- **Remote_Node:** The destination to which you are sending or from which you are receiving files.

Completion of Transfer

The message PGTF3108I is issued at the completion of all file transfer activity. This message can be used for automated processing or scheduling.

Successful Completion

```
PGTF3108I Transfer {Activity#} successfully transferred {Record_Count} records  
with node:
```

```
{Remote_Node}{Byte_Count}(Bytes) OR {Remote_Node}{Member_Count} {Members}
```

Where:

- **Activity#:** The number that is assigned to the transfer.
- **Record_Count:** The number of records sent or received by the transaction.
- **Remote_Node:** The destination to which you are sending or from which you are receiving (This can be displayed as an LU name, IP name, or IP address.)
- **Byte_Count:** The number of bytes transferred (uncompressed) for a sequential file.
- **Member_Count:** The number of members processed for PDS/Library transfers.

```
PGTF3108I {Activity#} RC={Return_Code} {Remote_Node} {Local_DSN} {Process}
Record_Count={Record_Count} {S/R} {Userdata}
```

Where:

- **Activity#:** The number that is assigned to the transfer.
- **Return_Code:** The number that is returned at the completion of a file transfer. A successful transfer has a return code of all zeroes. An unsuccessful file transfer issues a return code that helps the user determine the specific nature of the problem.
- **Remote_Node:** The destination to which you are sending or from which you are receiving. This can be displayed as an LU name, IP name, or IP address.
- **Local_DSN:** The name of the local dataset for the transaction. The dataset name must be a valid z/OS dataset and cannot exceed 54 characters in length. If the dataset provided is invalid, the interface prompts the user for a valid name.
- **Process:** An 8-character field intended to provide a description of the transaction being processed.
- **Record_Count:** The number of records sent or received by the transaction.
- **S/R:** The direction of the dataset on the host system. For example, if you are sending a file from the Windows system to the host, then the direction that must be specified is receive.
- **Userdata:** The data entered by the user on the process DESCRIPTION parameter.

Unsuccessful Completion

```
PGTF3109I Activity {Activity#} unsuccessful with Node {Remote_Node} Return
Code={Return_Code} error type ERROR
```

Where:

- **Activity#:** The number that is assigned to the transfer.
- **Remote_Node:** The destination to which you are sending or from which you are receiving files.
- **Return_Code:** The number that is returned at the completion of a file transfer. A successful transfer has a return code of all zeroes. An unsuccessful file transfer issues a return code that helps the user determine the specific nature of the problem.
- **Error_Type:** The type of error that caused the transfer to fail.

```
PGTF3108I {Activity#} RC={Return_Code} {Remote_Node} {Local_DSN} {Process}
Record Count={Record_Count} {S/R} {Userdata}
```

Where:

- **Activity#:** The number that is assigned to the transfer.
- **Return_Code:** The number that is returned at the completion of a file transfer. A successful transfer has a return code of all zeroes. An unsuccessful file transfer issues a return code that helps the user determine the specific nature of the problem.
- **Remote_Node:** The destination to which you are sending or from which you are receiving.
- **Local_DSN:** The name of the local dataset for the transaction. The dataset name must be a valid z/OS dataset and cannot exceed 54 characters in length. If the dataset provided is invalid, the interface prompts the user for a valid name.
- **Process:** An 8-character field intended to provide a description of the transaction being processed.
- **Record_Count:** The number of records sent or received by the transaction.
- **S/R:** The direction of the dataset on the host system. For example, if you are sending a file from the Windows system to the host, then the direction that must be specified is receive.
- **Userdata:** The data entered by the user on the process DESCRIPTION parameter.

Appendix D. File Name Tokens

File name tokens are a feature of the Platform Server. You can use this feature when communicating between any combination of z/OS, Windows and UNIX Platform Servers.

Given a string of tokens, which are characters containing a mixture of literal and substitution parameters, z/OS, Windows and UNIX Platform Servers can generate formatted file names that you can use to create or read file names based upon such tokens as date, time, and user.

Instead of entering a standard file name, you enter a name that consists of tokens. See the following topics for more information.

- [File Name Tokens List](#): lists the available tokens when a Windows or UNIX system is sending files that z/OS supports.

i Note: The Platform Server for z/OS performs token substitution to its local file name on both initiator and responder requests.

- [Examples of Using the File Name Tokens](#): Provides examples that demonstrate how to use file name tokens.
- [Rules for Using the File Name Tokens](#): Lists the rules that you must follow when using the file name tokens.

Tokens are specified by a dollar sign (\$) followed by the token name in parentheses. For example, \$(LocalFileName).

Additional tokens are available for remote file names when a z/OS Platform Server is initiating a request to a Platform Server for Windows or Platform Server for UNIX. For more information, see *TIBCO® Managed File Transfer Platform Server for UNIX User's Guide* or *TIBCO® Managed File Transfer Platform Server for Windows User's Guide*.

File Name Tokens List

To use the file name tokens, you have to be familiar with their respective definitions, and their generated values.

See the following table for a list of tokens that are supported:

Token	Definition	Generated Value (Examples)
CYear	CCYY	2009
Date	YYYYMMDD	20090625
Date1	YYMMDD	090625
Date2	MMDDYY	062509
Date3	DDMMYY	250609
DateUS	MMDDYYYY	12312014
Day	DD	01 - 31
Hour	HH	00 - 23
JDate	YYYYDDD	2009176
LocalUserId	z/OS user ID associated with the transfer.	Local file: \$(LocalUserId).\$(RemoteFileBase) Remote file: ACCT.TAX.Y2009 User ID: PRUSER1 Token resolves to: PRUSER1.ACCT.TAX
Member	Member name associated with the transfer.	Local file: \$(RemoteFileBase).\$(Member) Remote file: PROD.TEST.ACCT.TAX(Y2009) Token resolves to: PROD.TEST.ACCT.Y2009

Token	Definition	Generated Value (Examples)
Minute	MM	00 - 59
Month	MM	MM
Second	SS	00 - 59
Time	HHMMSMSS	084513126
Time1	HHMMSS	084513
Time2	HHMMSST	0845138
Year	YY	00 - 99
RemoteFileBase	The remote file name only.	Local file: \$(RemoteFileBase) Remote file: PROD.TEST.ACCT.TAX.Y2009 Token resolves to: PROD.TEST.ACCT.TAX
RemoteFileExt	All file name data after the first qualifier.	Local file: \$(RemoteFileExt) Remote file: PROD.TEST.ACCT.TAX.Y2009 Token resolves to: TEST.ACCT.TAX.Y2009
RemoteFileName	The remote file name including the extension to be used.	Local file: \$(RemoteFileName) Remote file: c:\source\directory\tstfile1.txt Token resolves to:

Token	Definition	Generated Value (Examples)
		TSTFILE1.TXT
RemoteFileNameFull	The remote file name including the member name to be used.	Local file: \$(RemoteFileNameFull) Remote file: PROD.TEST.ACCT.TAX(Y2009) Token resolves to: PROD.TEST.ACCT.TAX.(Y2009)
RemoteTransactionNumber	Remote transaction number used in the file transfer.	Local file: /tmp/\$(RemoteTransactionNumber).txt Remote file: c:\source\directory\testfile1.txt Token resolves to: d:\fn\I925600005
RemoteUserId	Remote user ID used in the file transfer.	Remote user ID: TEST\cfuser1 Remote file: PROD.\$(RemoteUserId).DATA Token resolves to: PROD.CFUSER1.DATA

Examples of Using the File Name Tokens

When transferring a file, you can create the file name using file name tokens instead of a regular file name.

See the following examples for your reference. These examples use this sample system date/time: Wednesday, April 25, 2009 5:03:45.061 PM.

- Example 1: The user has entered a date token preceded by the letter D when sending a file to a Platform Server for z/OS. The Platform Server on the z/OS system

resolves the string into the Date1YYMMDD format.

File name: PROD.TEST.DATA.SAMPLE.D\$(Date1)

Resolved file name: PROD.TEST.DATA.SAMPLE.D090425

- Example 2: When the Platform Server for z/OS receives files, the following substitution is performed for either token RemoteFileBase and RemoteFileExt.

File name: PROD.TEST.ACCT.TAX.Y2009

RemoteFileBase is all data before the last qualifier: PROD.TEST.ACCT.TAX

RemoteFileExt is all data after the first qualifier: TEST.ACCT.TAX.Y2009

The following two examples can be used when z/OS is sending a file to a Platform Server for Windows:

- Example 3: The user has entered a string of file name tokens when sending a file to a Platform Server for Windows. The Platform Server on the Windows system resolves the string into the directory name and file name.

File name: C:\directory\\$\$(SDD)\$\$(SMON)\$\$(SYYYYY)\\$(SHH24)\$\$(SMI)\$\$(SSS).dat

Resolved file name: C:\directory\25APR2009\170345.dat

- Example 4: The user has used the file name tokens to generate a resolved file name that has dashes between the date and time fields when sending a file to a Platform Server for Windows.

File name: C:\directory\\$\$(SDD)-\$(SMON)-\$(SYYYYY)\\$(SHH24)-\$(SMI)-\$(SSS).dat

Resolved file name: C:\directory\25-APR-2009\17-03-45.dat

Rules for Using the File Name Tokens

When creating a file name that uses file name tokens, you must follow certain rules.

See the following list for the rules you have to comply with:

- Substitution parameters are enclosed in $\$(tokenname)$. A dollar sign, followed by an open parenthesis, followed by the token name, followed by a close parenthesis.
- Each $\$(tokenname)$ can contain one token name only.
- Any text in the local file name that is not a substitution parameter is embedded, as is, into the generated name.

- Tokens might be used anywhere within the local file name.
- Space permitting, any number of substitution parameters might be embedded within the file name.
- If the resolved file name length is greater than the maximum file name, which is 255 characters, it is truncated.
- Tokens are case sensitive and will affect the output of the file name, so you must use them exactly as they are documented in [File Name Tokens List](#).
- If a formatted name contains an invalid substitution code, the transfer fails with an error stating that a substitution code is bad.

Appendix E. CA-7 Job Scheduler Interface

The Platform Server can trigger the execution of jobs on the z/OS system through the CA-7 Job Scheduler. This is achieved by utilizing the CA-7 interface module called U7SVC that is distributed as part of the CA-7 software product.

Module U7SVC is automatically loaded at the Platform Server initialization time. It is invoked after a file transfer is successfully completed. For the Platform Server to load U7SVC, it must either be in the Platform Server load library or concatenated in the STEPLIB DD statement of the startup PROC. When a dataset is successfully created, the name of the dataset and the volume serial on which it is created are passed to the U7SVC module. The z/OS installation must use dataset triggering within CA-7 to schedule the execution of jobs.


Appendix F. User Exits

The Platform Server currently supports six types of user exits.

EXIT00: End of Transfer Exit

At the successful or unsuccessful completion of a file transfer, the Platform Server for z/OS can branch to a user exit called FUSEX00.

TIBCO MFT Platform Server for z/OS provides a copy of its internal queue record. The exit program can analyze the record, and perform any tasks such as displaying additional messages or updating a user file. Any updates to the queue record are ignored by the Platform Server because this exit is intended as an informational record only.

 **Note:** This exit is invoked for temporary network errors and each transfer retry.

EXIT01: Transfer Authorization Checking Exit

Before performing its dataset authorization checking, the Platform Server for z/OS can branch to a user exit called FUSEX01.

As input to its exit, the Platform Server provides a copy of its internal queue records. At the time of the call to the user exit, the SAF ACEE points to the user ID associated with the transfer. FUSEX01 can then perform additional security validation to ensure that the user is authorized to use the Platform Server or to access the dataset in question. This user exit is called for:

- File, job, and report transfers
- Initiator transfer and responder transfer

This exit is invoked at transfer execution time. It is therefore possible for a user to queue a transfer successfully, yet with the transfer failing at execution time.

EXIT03: HSM Migration Exit

With the HSM migration exit, you can perform actions when an HSM migration volume is detected by the Platform Server.

In addition to the GLOBAL EXIT03 parameter, the following two GLOBAL parameters define when and how EXIT03 is invoked.

- EXIT_MIGRATE_VOLUME
- EXIT_MIGRATE_WAIT_TIME

At the start of a transfer, the Platform Server checks the catalog for the volume associated with the dataset. User EXIT03 is called only if the following conditions are met:

- GLOBAL EXIT03=YES.
- The volume matches the GLOBAL EXIT_MIGRATE_VOLUME.
- The volume does not match the GLOBAL HSM_MIGRATE_VOLUME.

DMS (CA Disk™ Backup and Restore Utility) uses a pseudo volume ARCIIVE for archived datasets. However, the actual volume that the dataset resides on is MIGRAT. In this case, to call EXIT03 you must ensure HSM_MIGRATE_VOLUME parameter does not equal MIGRAT and the EXIT_MIGRATE_VOLUME parameter is equal to MIGRAT.

As input to this exit, the Platform Server provides a copy of its internal queue records. Based on the return code, the Platform Server performs the following functions:

Return Code	Server Response
0	Continue processing.
1	Wait for the interval defined by the GLOBAL EXIT_MIGRATE_WAIT_TIME. If the catalog volume changes within this interval, the Platform Server processes the request. If the catalog volume does not change within this interval, the request is terminated with an error.
4	The request is treated as a temporary error and retried at the next retry interval.
> 4	The request is treated as permanent error and will be terminated.

EXIT04: Responder Preallocation Exit

The preallocation exit is called for responder transfers, just before the Platform Server allocating a file. It is called before the user ID and password is verified, and just after CFALIAS processing is performed.

With this exit, you can do the following:

- Terminate the transfer.

- Change these properties of a file transfer: local file name (dataset name), or file status and disposition.

As input to this exit, the Platform Server provides a copy of its internal queue record. Based on the return code, the Platform Server performs the following functions:

Return Code	Server Response
0	Continue processing with no changes.
1	The user exit has changed the local file name or the file status and disposition. The Platform Server will use these changes in the current transfer.
8	The request is treated as permanent error and will be terminated.
other	The request is treated as permanent error and will be terminated.

EXIT05: Purge from Queue Exit

At the successful or unsuccessful completion of a file transfer when the transfer is purged from the queue, the Platform Server can branch to a user exit called FUSEX05.

The Platform Server provides a copy of its internal queue record. The exit program can analyze the record, and perform any tasks such as displaying additional messages or updating a user file. Any updates to the queue record are ignored by the Platform Server because this exit is intended as an informational record only.

i Note: This exit is not invoked for temporary network errors until the RETRY parameter is exceeded and the transfer request is purged with completion status of unsuccessful.

EXIT06: Add to Queue Exit

This exit is called at the time a transfer request is queued to the server.

With this exit, you can do the following:

- Terminate the transfer.
- Check the specified transfer parameters.

- Change the following properties of a file transfer.

Effect (create, replace, append, create/replace, create/append, create/replace/new)

Encryption

Compression

Local file name (dataset name)

Remote file name

Description

Local user

Remote user

Remote password

Return code

Last message

Process name

Date eligible

Time eligible

Execution priority

Hold transfer

Remote system

As input to this exit, the Platform Server provides a copy of its internal queue record. Based on the return code, the Platform Server performs the following functions:

Return Code	Server Response
0	Continue processing with no changes.
1	The user exit has changed transfer parameters. The Platform Server will use these

Return Code	Server Response
	changes in the current transfer.
8	The request is treated as permanent error and will be terminated.
other	The request is treated as permanent error and will be terminated.

Invoking Platform Server User Exits

You must perform certain actions to invoke a Platform Server user exit.

See the following list for the steps you have to go through.

1. Assemble and link edit the exit program into the LOADLIB referenced by the STEPLIB of the Platform Server started task.
2. Turn on the corresponding GLOBAL parameter.
For example: EXIT00=YES
3. Start the Platform Server.

Programming Considerations

See the following list of the programming considerations:

- Programs must be reentrant and reusable. Otherwise unpredictable results will occur.
- Registers at entry

On entry to all FTMS exits, the following registers are set:

Registers	Value
R0	N/A
R1	Address of the EXIT_DSECT control block

Registers	Value
R2 - R12	N/A
R13	Address of 18 word save area
R14	Return address
R15	Entry point of the exit program

- EXIT_DSECT control block

This DSECT is in the FUSION SAMPLIB member EXIDSECT. The assembler version of the queue entry is in FUSION SAMPLIB member OSIQUEUE.

- Registers at exit

R0 - R14 are restored to the value upon entry.

Error codes are set in EXIDSECT field EXIT_RETURN_CODE. The contents of R15 are ignored. See following the return codes that can be set in EXIT_RETURN_CODE for each exit.

Exit Name	EXIT_RETURN_CODE & Action
EXIT00	Continues processing; return codes are ignored.
EXIT01	0: Continues processing. !=0: Terminates transfer. Displays message indicating processing terminated by user exit.
EXIT03	0: Continues processing. 1: Waits for GLOBAL EXIT_MIGRATE_WAIT_TIME to ensure that the 4 - volume has changed. 4: Temporary error. >4: Permanent error.

Exit Name	EXIT_RETURN_CODE & Action
EXIT04	<p>0: Continues processing.</p> <p>1: The user exit has changed the local file name or the file status and disposition. The Platform Server will use these changes in the current transfer.</p> <p>8: The request is treated as permanent error and will be terminated.</p> <p>Other: The request is treated as permanent error and will be terminated.</p>
EXIT05	Continues processing; return codes are ignored.
EXIT06	<p>0: continues processing.</p> <p>1: The user exit has changed the local file name or the file status and disposition. The Platform Server will use these changes in the current transfer.</p> <p>8: The request is treated as permanent error and will be terminated.</p> <p>Other: The request is treated as permanent error and will be terminated.</p>

- Sample program

Sample exits are in the following FUSION.SAMPLIB members:

FUSEX00, FUSEX01, FUSEX03, FUSEX04, FUSEX05, FUSEX06

Appendix G. WAIT Parameter for IF/THEN/ELSE/ENDIF

Using Platform Server WAIT parameter in conjunction with IBM IF/THEN/ELSE/ENDIF statement construct, you can conditionally execute additional job steps based on the outcome of the Platform Server file transfer.

You can choose one of the following two ways to use IF/THEN/ELSE logic when performing Platform Server transfers:

- Using the REXX interface described in the REXX interface chapter of *TIBCO® Managed File Transfer Platform Server for z/OS User's Guide*.
- Using the batch interface described in the batch interface chapter of *TIBCO® Managed File Transfer Platform Server for z/OS User's Guide* along with the JCL IF/THEN/ELSE parameters.

The following sample illustrates the syntax that can be used:

```
//SYSIN DD *
  PROCESS,IFTHEN,TRANSFER,RECEIVE
  .
  .
// [name] IF [(relational-expression)] THEN .
  .                               action when expression is
true
  .
// [name] ELSE
  .
  .                               action when relation-
expression
  .
//[name] ENDIF
```

By using the previous syntax, you can execute job steps within a job based on certain conditions of the outcome of the Platform Server transfer.

The IF statement is always followed by a relation-expression followed by a THEN clause. These three things are required when using the IF/THEN/ELSE/ENDIF statement construct. The THEN clause specifies those steps that the system processes upon the confirmation that the relational-expression, which is evaluated at startup time, is a true condition.

If the relational expression is a false condition, the ENDIF statement will follow. The ENDIF statement indicates the end of the IF/THEN/ELSE/ENDIF statement construct. It is

mandatory that ENDIF be coded at the end of every construct. However, before the ENDIF statement, the user can specify the ELSE clause. This ELSE clause specifies those steps that the system processes if the relational expression is false. This optional statement is followed by the ENDIF statement.

See the following example of how to use IF/THEN/ELSE/ENDIF within the Platform Server batch JCL. For more information on how to use Modal Logic, see the JCL reference manual.

```
//IFTHEN JOB 555,'IF THEN JCL',MSGCLASS=X,MSGLEVEL=1,
00001004
//      REGION=4000K,TIME=(2),NOTIFY=&USERID
00002004
//*****
00030000
//STEP1 EXEC PGM=OSIUB000,PARM='SERVER=FUSION'
00030104
//STEPLIB DD DISP=SHR,DSN=FUSION.LOADLIB
00030203
//SYSUDUMP DD SYSOUT=*
00030303
//SYSPRINT DD SYSOUT=*
00030403
//SYSIN DD *
00030503
    PROCESS,TEST1,TRANSFER,RECEIVE
00030603
    DSN=INPUT.DATASET.NAME
00030703
    REMOTE_FILE=C:\TEMP\TEST.TXT
00030803
    NODE=NODE1
00030903
    TYPE=TEXT
00031003
    CRLF=YES
00031103
    EFFECT=CR
00031203
    HOLD=NO
00031303
    REMOTE_USER=USER1
00031403
    RPASS=USER1
00031503
    NOTIFY=USER1
00031603
```



```
    NOTIFY_TYPE=TSO
00031703
    WAIT=YES
00031804
    RETRY=2
00031904
//NZERO IF (RC>5) THEN
00032003
//STEP2 EXEC PGM=IEBGENER
00033004
//SYSPRINT DD SYSOUT=X
00040000
//SYSUT1 DD DISP=SHR,DSN=FUSION.SAMPLIB(MEMBER1)
00050000
//SYSUT2 DD SYSOUT=X
00060000
//SYSIN DD DUMMY
00070000
//  ELSE
00081003
//STEP3 EXEC PGM=IEBGENER
00090000
//SYSPRINT DD SYSOUT=X
00100000
//SYSUT1 DD DISP=SHR,DSN=FUSION.SAMPLIB(MEMBER2)
00110000
//SYSUT2 DD SYSOUT=X
00120000
//SYSIN DD DUMMY
00130000
//ENDSTEP ENDIF
00420001
/*
00430001
```

Appendix H. Overriding JCL SYSIN Parameters

You can specify SYSIN parameters by using the PARM= parameter of the JCL EXEC statement.

The parameters describing the server (SERVER, SERVLUNAME, SERVIPADDR, SERVIPNAME) and their values must be the first group of items specified. For more information on the required PARM fields, see the batch interface chapter of *TIBCO® Managed File Transfer Platform Server for z/OS User's Guide*.

Following the server name, any valid SYSIN parameters can be specified in the parameter string. For more information on valid SYSIN parameters, see the batch interface chapter of *TIBCO® Managed File Transfer Platform Server for z/OS User's Guide*.

Each parameter and its corresponding value must be separated by an equal sign (=). A comma must separate each parameter/value pair. The PARM statement might span multiple lines by typing up to column 71 and placing a continuation mark in column 72. The next line must begin in column 16.

In the following example, the interpreted value of DSN is DATASET.NAME. Long file names with spaces are also supported by enclosing the remote file name in double quotation marks.

```
//STEPNAME EXEC PGM=OSIUB000,PARM='SERVER=FUSION,NODE=RemNode,DSN=DATAS
//          ET.NAME,REMOTE_FILE="c:\test file.tst"'
```

The parameters specified by the PARM string are processed after the SYSIN parameters and override any duplicate parameter values specified in the SYSIN. The parameters in the PARM string are also applied to every PROCESS statement (every transaction) queued in that particular step.

See the following batch example for your reference. The SYSIN parameters are overwritten.

```
//BATCH1     JOB 555,MSGCLASS=X,REGION=4M
//STEP1     EXEC PGM=OSIUB000,PARM='SERVER=FUSION,REMOTE_
FILE=C:\SalesRe-
//          port.DOC'
//STEPLIB   DD DISP=SHR,DSN=PROD.LOADLIB
//SYSPRINT  DD SYSOUT=*
//SYSUDUMP  DD SYSOUT=X
//SYSIN     DD *
```

```

PROCESS,SALES1,TRANSFER,RECEIVE
  NODE=REMNODE1
  DSN=SALES.REPORT1
  REMOTE_FILE=C:\OVERRIDE
  EFFECT=CR
  ISSUER=USERID
  RPASS=PASSWD
PROCESS,SALES2,TRANSFER,RECEIVE
  NODE=REMNODE2
  DSN=SALES.REPORT2
  EFFECT=CR
  ISSUER=USERID
  RPASS=PASSWD
//STEP2      EXEC PGM=OSIUB000,PARM='SERVER=FUSION,DSN=DISTRIB.ANNUAL.RE-
//           PORT,REMOTE_FILE=C:\AnnualReport.Doc'
//STEPLIB   DD DISP=SHR,DSN=PROD.LOADLIB
//SYSPRINT  DD SYSOUT=*
//SYSUDUMP  DD SYSOUT=X
//SYSIN     DD *
PROCESS,DISTRIB1,TRANSFER,SEND
  NODE=REMNODE1
  DSN=DATASET.OVERRIDE
  EFFECT=CR
  ISSUER=USERID
  RPASS=PASSWD
PROCESS,DISTRIB2,TRANSFER,SEND
  NODE=REMNODE2
  EFFECT=CR
  ISSUER=USERID
  RPASS=PASSWD

```

Bold type SYSIN parameters in the previous example will be overridden and are therefore unnecessary to code. The process names SALES1 and SALES2 in the first step will receive a remote file called C:\SalesReport.DOC whether or not the parameter is specified in the SYSIN. In the second step, processes DISTRIB1 and DISTRIB2 will send the dataset named DISTRIB.ANNUAL.REPORT to the remote file name C:\AnnualReport.Doc. Both the DSN and the REMOTE_FILE parameters are ignored if specified in SYSIN.

When a parameter is overridden, a warning message is written to the batch output file following the SYSIN parameters. The message contains the overridden parameter name and is displayed for every parameter specified in the PARM string.

See the following example of the warning message issued when the SYSIN parameter has been overwritten.

PGTB4124W PARM string parameter DSN will over write SYSIN value

Appendix I. XCOM Interface

If you want to migrate from CA-XCOM to the Platform Server, you can use the Platform Server XCOM interface program that can ease this migration.

The Platform Server has a new user interface that accepts XCOM batch parameters, and creates a Platform Server request instead of an XCOM request.

With this program:

- Users who are familiar with XCOM batch parameters can get immediately productive using the Platform Server.
- It eases the transition from XCOM to Platform Server. Both products can be run simultaneously.

i Note: The Platform Server and XCOM are different products and, as such, perform the same tasks differently. The Platform Server for z/OS XCOM interface does not perform all of the functions that XCOM performs, just like XCOM does not perform all of the Platform Server functions.

The Platform Server XCOM interface is designed to support enough XCOM functionality so that 90 to 95 percent of a user's request can execute without any changes. When the Platform Server XCOM interface determines that it cannot perform a particular function, the step will terminate with a return code 0x28 (decimal 40).

The XCOM batch interface program is called XCOMJOB. The following components of XCOMJOB are supported by the Platform Server XCOM interface:

- TYPE=EXECUTE: Executes a request and waits for completion.
- TYPE=SCHEDULE: Schedules a request with the started task.
- TYPE=INQUIRE: Inquires on one or more scheduled requests.

Installing and Configuring Platform Server XCOM Interface

The Platform Server XCOM interface is installed when TIBCO MFT Platform Server for z/OS is installed.

You must perform a single step to assemble the XCOMDFLT options table. This is the same XCOMDFLT table that is assembled when XCOM is installed at an installation.

Procedure

1. Copy the XCOMDFLT member into the Platform Server SAMPLIB.

If you want, you can use the Platform Server default XCOMDFLT table that already exists in the Platform Server SAMPLIB.

i Note: Because most XCOM parameters are ignored by the Platform Server, it might be easier to update the supplied XCOMDFLT member.

2. Make any changes to the XCOMDFLT table to configure it for the Platform Server.

The following parameters require changes:

- JOBACB: Points to the name of the VTAM ACBs defined to the Platform Server.
- ACBNAME: Points to ACBNAME of the Platform Server started task.

Add the following new Platform Server parameters to #DFLTAB:

```
FUSION_STCOVER={ YES/NO}
```

- If this parameter is specified as YES, then the ACBNAME name overrides the STCAPPL parameter normally defined in the PARM JCL statement. The STCAPPL statement is ignored when FUSION.STCOVER=YES is defined. This is done so that a user can submit the same JCL, except that the STEPLIB will point to the Platform Server library, and the Platform Server request is routed to the Platform Server started task, not to the XCOM started task.
- If this parameter is specified as NO, the STCAPPL statement on the PARM JCL statement overrides the ACBNAME defined to the XCOMDFLT table.

3. Assemble the XCOMDFLT table.

You can use any assembly JCL. The following changes must be made to the JCL to assemble the Platform Server XCOMDFLT module:

- Ensure that the SYSIN DD statement points to the XCOMDFLT in the Platform

Server SAMPLIB.

- Ensure the SYSLIB DD statement of the Assembler step points to the Platform Server SAMPLIB dataset.
- Ensure that the XCOMDFLT module is linked into the Platform Server LOADLIB dataset with the name XCOMDFLT.

At this point, the Platform Server XCOM interface is ready to use. The Platform Server XCOM interface program is called FUSXJOB. This program has an alias of XCOMJOB.

To run the Platform Server XCOM interface, you must change the STEPLIB of the program to use the Platform Server loadlib. That way, when program XCOMJOB is executed, the Platform Server program is executed instead of the XCOM program.

As stated before, the Platform Server XCOM interface accepts XCOM parameters and creates a Platform Server transfer request. After the transfer parameters are read, only Platform Server protocols are used, which means the following things:

- The remote computer must have a version of Platform Server installed. The Platform Server does not communicate with the XCOM started task.
- The messages displayed are Platform Server messages, not XCOM messages.

The PARM fields that are ignored by the Platform Server XCOM interface are COMPNEG, DISPALG, DUMPCL, EDESC, EROUT, GROUP, IDESC, IROUT, LOG, LOGCLASS, LOGDEST, LOGMODE, and LU.

The SYSIN01 fields that are ignored by the Platform Server XCOM interface are DISP, DROPSSESS, EPRTY, LPASS, LUSER, REPORT, REPTHOLD, RESTART, RNOTIFY, RNOTIFYNAME, SPRTY, TRUNCATE, USER, and PACK.

The SYSIN01 fields that will cause the Platform Server XCOM interface to terminate with a return code 40 (0x28) are DEN, EXPDT, LABEL, LLABEL, LUNITCT, LVOLCT, LVOLSQ, RETPD, UNITCT, VOLCT, VOLSQ, CHARS, SPOOL, HOLDCOUNT, XTCERRDECR, XTCERRINCR, XTCERRREL, XTCERRPURGE, XTCGOODDECR, XTCGOODINCR, XTCGOODREL, XTCGOODPURGE, XTCJOB, and XTCNET.

See the following example of the JCL required to run the Platform Server XCOM interface:

```
//jobname JOB    , 'C-Fusion', MSGCLASS=X, REGION=5M, CLASS=A
//S1SCH      EXEC PGM=XCOMJOB,
```

```
// PARM='TYPE=SCHEDULE,ACBNAME=FUSN,GROUP=NYACCT'

//STEPLIB DD DISP=SHR,DSN=FUSION.LOADLIB
//XCOMLOG DD SYSOUT=*

//XCOMINQ DD DSN=yourdsn,DISP=SHR
//SYSIN01 DD *

TYPE=SEND

CODE=TEXT

RECSEP=YES

LFILE=LOCAL.TEST.DATA
FILE=REMOTE.TEST.DATA
USERID=ABC123
PASSWORD=XYZ
//
```

Though you are using the Platform Server XCOM interface, you can still use some Platform Server features. For example, instead of the `USERID` and `PASSWORD` parameters defined in the previous example, you can use the Platform Server user profile feature. The `USERID` and `PASSWORD` parameters are replaced by the following parameter:

```
USERID=*PROFILE
```

That way, the clear text password does not have to be displayed in the JCL. See the following output that is displayed after executing the previous job:

```
TYPE=SEND

CODE=TEXT

RECSEP=YES

LFILE=LOCAL.TEST.DATA
FILE=REMOTE.TEST.DATA
```



```

USERID=ABC123
PASSWORD=XXXXXXXXXXXXXXXXXXXXXXXXXXXXX

PGTX4512I Activity I414000000 Queued to MFT Platform Server started
task
PGTX4510I Successful Requests=1 Failed Requests=0 Timeout
Requests=0
PGTX4511I FUSXJOB ending with Return code 0

```

If TYPE=EXECUTE is requested instead of TYPE=SCHEDULE, the following messages are displayed in the XCOMLOG dataset:

```

TYPE=SEND
CODE=TEXT

RECSEP=YES

LFILE=LOCAL.TEST.DATA
FILE=REMOTE.TEST.DATA

USERID=ABC123
PASSWORD=XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PGTX4512I Activity I414000002 Queued to MFT Platform Server started
task
PGTX4521E Activity=I414000002 Status=ACTIVE Record Count=0
Byte Count=0
PGTX4521E Activity=I414000002 Status=COMPLETE Record Count=3200
Byte Count=259200
PGTX4510I Successful Requests=1 Failed Requests=0 Timeout
Requests=0
PGTX4511I FUSXJOB ending with Return code 0

```

Appendix J. PDS and HFS File

The Platform Server supports transfers of PDS and HFS files.

PDS Support

The Platform Server supports transfers of libraries between z/OS systems.

The following functions for library transfers are supported:

- Transfer of individual members.
- Transfer of complete libraries.
- Transfer of members with wildcard.
- Renaming of members that are transferred.
- Transfer of members between PDS and PDS/E datasets.
- Transfer of LOADLIBs between PDS datasets.
- All member directory information is sent along with the member.
- Alias members are fully supported.
- Dataset allocation attributes are sent to the receiving system.
- Dataset allocation attributes can be overridden.

Restrictions to PDS Support

See the following two restrictions to Platform Server PDS support:

- The Platform Server will not write data to a PDS/E program library.
- The Platform Server will not send an alias alone without the actual member.

Additional Batch Parameters for Library Support

The following two additional batch parameters are added for library support:

- DATASET_TYPE=LIBRARY

Informs the receiver of the file that if a dataset is to be created, a PDS/E must be created. If this parameter is not set, a PDS is created at the remote end.

- `ALLOC_DIR=directory_block_number`

Defines the number of directory blocks that will be allocated when a file is created. This parameter overrides the local dataset attributes that are sent to the system which creates the dataset.

Generic/Wildcard Support for Members

- The asterisk (*) denotes a generic member.
- If a PDS is specified without a member name, all members are sent.
- `FILE=MY.PDS` is the same as `MY.PDS(*)`.
- Question mark (?) is the wildcard character. It can be used with characters of a member and with the generic character (*).

See the following examples of generic member or wildcard support:

Example 1:

```
FILE=MY.PDS(abc*)
REMOTE_FILE=YOUR.PDS(WXYZ*)
```

All files starting with ABC will be sent to the remote system. When the files are written, the first 4 characters are replaced by WXYZ.

Example 2:

```
FILE=MY.PDS(???TEST?)
REMOTE_FILE=YOUR.PDS
```

Members in which characters 4 to 7 are TEST will be sent. The member names on the remote system are the same as on the local system.

Alias Support

- When a full library, or partial library transfer is specified, all members and aliases that match the selection criteria will be sent to the remote system.
- When a specific member is sent, that member, and all aliases of that member will be sent to the remote system.

Examples of Sending PDS Files

```
PROCESS SEND
      DSN=MY.LOCAL.PDSS(ABC)
      REMOTE_FILE=YOUR.PDSS
      NODE=REMOTE
```

- Sends member ABC to node REMOTE member ABC.

```
PROCESS SEND
      DSN=MY.LOCAL.PDS(ABC)
      REMOTE_FILE=YOUR.PDS(XYZ)
      NODE=REMOTE
```

- Sends member ABC to node REMOTE member XYZ.

```
PROCESS SEND
      DSN=MY.LOCAL..PDS
      REMOTE_FILE=YOUR..PDS
      DATASET_TYPE=LIBRARY
      NODE=REMOTE
```

- Sends all members of a PDS to a PDSE on node REMOTE.

```
PROCESS SEND
      DSN=MY.LOCAL.PDS(???PROD*)
      REMOTE_FILE=YOUR.PDS(???TEST)
      DATASET_TYPE=LIBRARY
      NODE=REMOTE
```

- Sends selected members of a PDS to a PDSE on node REMOTE. Changes characters 3 to 6 from PROD to TEST when writing the members.

i Note: When you send libraries to a non-z/OS system, or an earlier release of TIBCO MFT Platform Server for z/OS, if you attempt to send or receive a library, and the remote computer does not support library transfers, the Platform Server will end the transfer and display a message on both the local and remote computer which indicates that the remote computer does not support library transfers. The transfer request is purged from the request queue afterwards.

HFS File Support

The Platform Server supports the transfer of HFS files. Both send and receive operations are supported.

See the following restrictions for this support:

- The Platform Server started task user must be defined as a superuser.
- Users initiating a request for an HFS file must have the OMVS segment defined so that they are recognized as OMVS users.
- For a file create operation, the HFS directory must exist. The Platform Server for z/OS will not create a directory.
- The user associated with a file transfer request must have the necessary authority to read or update an existing file depending on whether a send or receive transfer is performed.
- For a file create operation, the user must have write authorization for the directory that the file is written to.

The Platform Server supports file names of up to 256 characters long. The Platform Server creates the security environment for a transfer request by validating the user ID and password through RACF. After the security environment is established, standard OMVS security validation is performed.

During text record processing, line feed (LF) is inserted at the end of each record. LF is not inserted on binary transfers. All text transfers store data in EBCDIC.

TIBCO Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join TIBCO Community.

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the [TIBCO Product Documentation](#) website, mainly in HTML and PDF formats.

The [TIBCO Product Documentation](#) website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The following documentation for TIBCO® Managed File Transfer Platform Server for z/OS is available on the [TIBCO® Managed File Transfer Platform Server for z/OS Product Documentation](#) page.

- *TIBCO® Managed File Transfer Platform Server for z/OS Release Notes*
- *TIBCO® Managed File Transfer Platform Server for z/OS Managed File Transfer Overview*
- *TIBCO® Managed File Transfer Platform Server for z/OS Installation and Operation Guide*
- *TIBCO® Managed File Transfer Platform Server for z/OS Security Guide*
- *TIBCO® Managed File Transfer Platform Server for z/OS User's Guide*
- *TIBCO® Managed File Transfer Platform Server for z/OS Message Manual*

How to Contact TIBCO Support

Get an overview of [TIBCO Support](#). You can contact TIBCO Support in the following ways:

- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the [TIBCO Support](#) website.
- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to [TIBCO Support](#) website. If you do not have a user name, you can request one by clicking **Register** on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the [TIBCO Ideas Portal](#). For a free registration, go to [TIBCO Community](#).

Legal and Third-Party Notices

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO, the TIBCO logo, the TIBCO O logo, TIBCO Managed File Transfer, TIBCO Managed File Transfer Command Center, TIBCO Managed File Transfer Internet Server, and TIBCO Managed File Transfer Platform Server are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle Corporation and/or its affiliates.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: <https://scripts.sil.org/OFL>

Copyright (c) Paul D. Hunt, with Reserved Font Name Source Sans Pro and Source Code Pro.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the readme file for the availability of this software version on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

This and other products of TIBCO Software Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (<https://www.tibco.com/patents>) for details.

Copyright © 2003-2021. TIBCO Software Inc. All Rights Reserved.