CICS Transaction Server for z/OS

IBM

Migration from CICS TS for z/OS 2.2

Version 3 Release 1

CICS Transaction Server for z/OS

Migration from CICS TS for z/OS 2.2

Version 3 Release 1

This edition applies to Version 3 Release 1 of CICS Transaction Server for z/OS, program number 5655-M15, and to all subsequent versions, releases, and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Contents

(left margin)
\#

Preface

What this book is about

This book is about migration to CICS® Transaction Server for z/OS®
Version 3 Release 1. It provides information for users who plan to migrate from
CICS Transaction Server for z/OS Version 2 Release 2. For the purposes of this
book, "migration" is generally taken to mean running existing applications at the
equivalent level of function provided by the existing release.

Note: If you are migrating from a release of CICS earlier than CICS TS Version 1
Release 3, you are recommended to read the *Release Guide* and the
Migration Guide (where applicable) for the intervening releases.

Note: If you are migrating from a release of CICS earlier than CICS TS Version 2
Release 3, you are recommended to read the *Release Guide* and the
Migration Guide (where applicable) for the intervening releases.

Who should read this book

This book is for those responsible for planning the migration to CICS Transaction
Server for z/OS Version 3 Release 1.

It describes external interfaces, such as system definitions, resource definitions, and
programming interfaces, that have changed or are new, and which may require you
to make changes to your existing CICS and CICSPlex® SM setup.

What you need to know to understand this book

This book assumes that you are familiar with CICS and CICSPlex SM, either as a
systems administrator, or as a system or application programmer.

You should also have read about the new function in CICS TS Version 3 as
described in the the *CICS Transaction Server for z/OS Release Guide*.

Notes on terminology

CICS refers to the CICS element of the CICS Transaction Server for z/OS.

CICS TS, unless stated otherwise, refers to Version 2 Release 2 of CICS
Transaction Server for z/OS.

CICSPlex SM refers to the CICSPlex System Manager element of the CICS
Transaction Server for z/OS.

CICS/MVS is used for Customer Information Control System/Multiple Virtual
Storage.

CICS/ESA is used for Customer Information Control System/Enterprise System
Architecture.

MVS™ is used for the operating system, the Base Control Program (BCP) element
of z/OS.

ix

Part 1. Changes to CICS externals

This part of the book deals with all the changes that affect CICS externals, such as system and resource definitions and programming interfaces. The topics covered are as follows:

Chapter 1. Installation changes

This chapter summarizes the changes to CICS installation.The topics covered are:
- "Installation process changes"
- "Installation changes for Language Environment"
- "Installation changes for data conversion"

Installation process changes

This release of CICS Transaction Server is installed using the SMP/E RECEIVE, APPLY, and ACCEPT commands. The SMP/E dialogs may be used to accomplish the SMP/E installation steps.

The process is described in the *CICS TS 3.1 Program Directory*. It is in line with IBM® Corporate Standards, and may be familiar to those who have installed other z/OS products.

The traditional method, DFHISTAR, of installing CICS Transaction Server is still available. The *Program Directory* indicates where information about DFHISTAR may be found in the *CICS Transaction Server for z/OS Installation Guide*.

Authorization routines

In z/OS, do not install SVCs or PC routines that return control to their caller in any
authorized mode: that is, in supervisor state, system PSW key, or APF-authorized.
Doing so is contrary to the z/OS Statement of Integrity (http://www.ibm.com/
systems/z/os/zos/features/racf/zos_integrity_statement.html).

If you invoke such services from CICS, you might compromise your system
integrity, and any resultant problems will not be resolved by IBM Service.

Installation changes for Language Environment

With support for the XPLINK compiler option for C and C++ programs, there are changes to the way that the SCEERUN and SCEERUN2 libraries must be defined.
- The library SCEERUN2 must be defined in both the STEPLIB and DFHRPL concatenations, in addition to SCEERUN.
- Both the libraries, SCEERUN and SCEERUN2, must be APF-authorized.

For more information, see in "Installing CICS support for Language Environment®" in the *CICS Transaction Server for z/OS Installation Guide*

Installation changes for data conversion

The CICS installation process does not alter if you have data conversion requirements.

However, to get the benefits of z/OS conversion services, if perhaps your system requires support for the conversion of UTF-8 or UTF-16 data to EBCDIC, you must enable the z/OS conversion services and install a conversion image which specifies
the conversions that you want CICS to perform. It is quite likely, for example, that
outbound SOAP applications using the SOAP for CICS feature will try to use z/OS
conversion services.

Refer to the instructions in the *z/OS Support for Unicode: Using Conversion Services* manual SA22-7649 to find out the steps needed to set up and configure conversions supported though the operating system services.

- If z/OS conversion services are not enabled, a message is issued by CICS to indicate this. That message can be suppressed if you do not need these services. If the message is encountered when starting a CICS region that is expected to make use of these services, an IPL is necessary to enable the z/OS conversion services.

To discover the status of z/OS conversion services after an IPL, use one of these commands from an MVS console:

/D UNI
To show whether z/OS conversion services were enabled.

/D UNI,ALL
To show whether z/OS conversion services were enabled, and which conversions are supported by the system.

For details of this, see the *z/OS Support for Unicode: Using Conversion Services* manual SA22-7649

Chapter 2. System initialization parameters

This chapter summarizes the changes to CICS system initialization parameters. The topics covered are:

- "Obsolete system initialization parameters"
- "Changed system initialization parameters"
- "New system initialization parameters" on page 6
- "Getting started with new and changed system initialization parameters" on page 9

Obsolete system initialization parameters

Table 1 shows those system initialization parameters that are obsolete.

Remove any of these obsolete parameters from your system initialization table, or from your CICS startup JCL (for example, the SYSIN data set) before migrating.

Table 1. Obsolete system initialization parameters

Obsolete keywords	Explanation
MAXHPTCBS	Run-time support for Java™ program objects and hot-pooling (HPJ) has been removed. The system initialization parameter MAXHPTCBS is not required, and is removed. The open TCB mode H8, which was used for hot-pooling Java program objects and was controlled by MAXHPTCBS, no longer exists.
SSLTCBS	This parameter is now obsolete and is only kept for compatibility. If it is specified, it is rejected with a message and MAXSSLTCBS is assumed.
TCAM	This parameter is now obsolete and is only kept for compatibility. If it is specified, it is rejected with a message and TCAM=NO is assumed. CICS Transaction Server for z/OS, Version 3 Release 1 does not support the TCAM/ACB interface. It supports the TCAM/DCB interface indirectly.

Changed system initialization parameters

Table 2 shows those system initialization parameters that have changed in some way.

Table 2. Changed system initialization parameters

Keywords	Operands	Explanation
EDSALIM	{**30M**|number}	The default value is now 30M
ENCRYPTION	{**STRONG** |*WEAK* |*MEDIUM*}	Specifies the cipher suites that CICS uses for secure TCP/IP connections. For compatibility with previous releases, ENCRYPTION=NORMAL is accepted as an equivalent to ENCRYPTION=MEDIUM. For more information, see the *CICS System Definition Guide*.

Table 2. Changed system initialization parameters (continued)

Keywords	Operands	Explanation
MAXJVMTCBS	{**5**|number}	specifies the maximum number of open TCBs that CICS can create in the pool of J8- and J9-mode TCBs for use by Java programs that run in a JVM (the JVM pool). Within this limit, there are no constraints on how many of the TCBs in the JVM pool are J9 TCBs, and how many are J8 TCBs. For more information, see the *CICS System Definition Guide*.
STNTR, STNTRxx, SPCTR and SPCTRxx	{level numbers}	The SJ component (JVM domain) now has trace levels 29–32, that are reserved to indicate the JVM trace levels 0, 1, and 2, plus a user-definable JVM trace level. You are recommended to use only the SPCTRSJ system initialization parameter to activate JVM tracing, so that it is only activated for *special* transactions. Selecting tracing levels 29, 30, 31, 32 or ALL for *standard* tracing for the JVM domain (SJ) component (using the STNTR or STNTRSJ system initialization parameters) is not recommended, because JVM trace can produce a large amount of output. For more information, see the *CICS System Definition Guide*.

For more information on the changed parameters, see the *CICS System Definition Guide*.

New system initialization parameters

The default values for these parameters are designed to have minimal impact when you are migrating from an earlier release of CICS.

Table 3. New system initialization parameters

Keywords	Operands	Explanation
CLINTCP	{**437**|codepage}	Specifies the default client code page to be used by the DFHCNV data conversion table but only if the CLINTCP parameter in the DFHCNV macro is set to SYSDEF. For more information, see the *CICS System Definition Guide*.
CRLSERVER	*servername: portnumber*	Specifies the name of the LDAP server and the port number that contains the certification revocation lists (CRLs). Specifying this parameter means that CICS checks each client certificate during the SSL negotiation for a revoked status. If the certificate is revoked, CICS closes the connection immediately. For more information, see the *CICS System Definition Guide*.
DEBUGTOOL	{**NO**|YES}	Specifies whether debugging profiles will be used to select programs that will run under the control of a debugging tool. For more information, see the *CICS System Definition Guide*
INFOCENTER	{infocenter_url }	Specifies the Universal Resource Locator (URL) of the root of the CICS Information Center directory structure. For more information, see the *CICS System Definition Guide*
JVMCCPROFILE	{**DFHJVMCC**|*profile*}	Specifies the JVM profile to be used for the master JVM that initializes the shared class cache. For more information, see the *CICS System Definition Guide*.
JVMCCSIZE	{**24M**|number}	Specifies the size of the shared class cache on an initial or cold start of CICS. For more information, see the *CICS System Definition Guide*.

Table 3. New system initialization parameters (continued)

Keywords	Operands	Explanation
JVMCCSTART	{**AUTO**|YES|NO}	Determines whether or not the shared class cache is started during CICS initialization, and sets the status of autostart for the shared class cache. For more information, see the *CICS System Definition Guide*.
JVMLEVEL0TRACE, JVMLEVEL1TRACE, JVMLEVEL2TRACE, JVMUSERTRACE	{*option*}	Specify the default options for the JVM trace levels. For more information, see the *CICS System Definition Guide*.
JVMPROFILEDIR	{**/usr/lpp/cicsts/ cicsts31/ JVMProfiles**|*directory*} (Take note that the operand above is entered without spaces, or line breaks.)	Specifies the name of an HFS directory that contains the JVM profiles for CICS. For more information, see the *CICS System Definition Guide*.
LOCALCCSID	{**037**|CCSID}	Specifies the default CCSID for the local region. The CCSID is a value of up to 8 characters. If CCSID value is not specified, the default LOCALCCSID is set to 037. For more information, see the *CICS System Definition Guide*.
MAXSSLTCBS	{**8**|*number*}	Specifies the maximum number of S8 TCBs that can run in the SSL pool. The default is 8, but you can specify up to 1024 TCBs. For more information, see the *CICS System Definition Guide*.
MAXXPTCBS	{**5**|*number*}	Specifies the maximum number, in the range 1 through 999, of open X8 and X9 TCBs that can exist concurrently in the CICS region. For more information, see the *CICS System Definition Guide*.
MNRES	{**OFF**|ON}	Specifies whether transaction resource monitoring is to be made active during CICS initialization.

Table 3. New system initialization parameters (continued)

Keywords	Operands	Explanation
SRVERCP	{**037**|codepage}	Specifies the default server code page to be used by the DFHCNV data conversion table but only if the SRVERCP parameter in the DFHCNV macro is set to SYSDEF. For more information, see the *CICS System Definition Guide*.
SSLCACHE	{**CICS**| SYSPLEX}	Specifies whether SSL is to use the local or sysplex caching of session ids. For more information, see the *CICS System Definition Guide*.

For information about the new function relating to these new system initialization parameters, see the *CICS Transaction Server for z/OS Release Guide*.

Getting started with new and changed system initialization parameters

Here is a simple way of migrating with the changes to system initialization parameters described above:

Use the default system initialization table
The unsuffixed default system initialization table (DFHSIT) is supplied in the CICS SDFHAUTH library. You can use this to start a CICS region using the default values — CICS loads DFHSIT by default if there is not a SIT parameter in your JCL.

Override defaults using the SYSIN data set
To override default values, specify system initialization parameters in a permanent member of a SYSIN data set.

You can vary these easily during testing, avoiding the need to reassemble suffixed system initialization tables. Nearly all system initialization parameters entered at run time are used even on a warm start (the exceptions are the FCT and CSD parameters).

Chapter 3. CICS-supplied transactions

This chapter summarizes the changes to CICS-supplied transactions.

The topics covered are:
* "New transaction CCRL"
* "New transaction CWXU" on page 12
* "Changes to CEBR" on page 12
* "Changes to CEMT" on page 12
* "Changes to CETR" on page 15
* "Changes to transaction CWXN" on page 16
* "Additions to CICS RACF category 1 transactions" on page 17

New transaction CCRL

Use the CCRL, the certificate revocation lists transaction, to create and update the certificate revocation lists (CRLs) that are stored in an LDAP server. You only need to use CCRL if you are implementing SSL in your CICS regions and want each connection checked for a revoked certificate during the SSL handshake.

The CCRL transaction specifies the location of CRL repositories on the world wide web. CICS downloads the lists from the CRL repository at the specified URL and stores it in the LDAP server. You can specify more than one URL if you need to access multiple CRL repositories.

Before you run the CCRL transaction, you must have the following set up in CICS:
* An LDAP server that is set up and configured to store the certificate revocation lists.
* The CRLPROFILE system initialization parameter is defined with the name of the profile that authorizes CICS to access the certificate revocation lists on the LDAP server.

You can run the CCRL transaction from a terminal or from a START command. If you want to schedule regular updates, use the START command option.

To run the transaction from a terminal, enter the following command: `CCRL url-list` where *url-list* is a space-delimited list of URLs that contain the certificate revocation lists that you want to download.

To run the transaction from a START command, using the following syntax:
```
EXEC CICS START TRANSID(CCRL) FROM (url-list)
LENGTH (url-list-length) [INTERVAL(hhmmss)|TIME(hhmmss)]
```

where *url-list* is a space-delimited list of URLs that contain the certificate revocation lists that you want to download, *url-list-length* is the length of the URL list, and *hhmmss* is the interval or expiration time at which the CCRL transaction is scheduled to run.

If you enter an invalid URL, you will receive an error message.

New transaction CWXU

In CICS Transaction Server for z/OS, Version 3 Release 1, processing for HTTP requests and processing for non-HTTP requests are kept separate. This ensures that CICS can perform basic acceptance checks on HTTP requests and responses, and that non-HTTP requests are not subjected to these checks. Processing for non-HTTP requests must now be carried out under the user-defined (USER) protocol, which is specified on the TCPIPSERVICE definition for the port that receives the requests.

The new CICS-supplied transaction CWXU, the CICS Web user-defined protocol attach transaction, is the default when the protocol is defined as USER. CWXU executes the CICS program DFHWBXN. The DFHCURDI sample includes a sample definition for CWXU. An alternative transaction that executes DFHWBXN may be used, with the exception of the other default transactions that are defined for protocols on the TCPIPSERVICE resource definition.

CWXU is a RACF® Category 1 transaction.

Changes to CEBR

The CEBR transaction is changed to permit the entry of lower case and mixed case queue names. This applies equally to the queue name used on the CEBR command, and to overtyping a queue name on the displayed CEBR panel. To enter a queue name which includes lower case characters, first ensure that upper case translation is suppressed for the terminal you are using, and then enter the case-sensitive queue name.

Changes to CEMT

New CEMT commands

Table 4 shows new CICS transactions.

Table 4. New CEMT commands

CEMT command	Explanation
DISCARD PIPELINE	Use the command to remove a PIPELINE from the CICS system and the CICS catalog.
DISCARD URIMAP	Use the command to remove a URIMAP from the CICS system and the CICS catalog.
DISCARD WEBSERVICE	Use the command to remove a WEBSERVICE from the CICS system and the CICS catalog.
INQUIRE CLASSCACHE	The INQUIRE CLASSCACHE command is added to give you information about the active shared class cache in the CICS region, and report the presence of any old shared class caches that are awaiting deletion.
INQUIRE HOST	Use the command to retrieve information about a particular virtual HOST in the local CICS region.
INQUIRE JVM	The INQUIRE JVM command is added to enable you to identify JVMs in a CICS region and get information about their status.
INQUIRE PIPELINE	Use the command to retrieve information about an installed PIPELINE.
INQUIRE URIMAP	Use the command to retrieve information about URIMAP resource definitions.
INQUIRE WEBSERVICE	Use the command to retrieve information about an installed WEBSERVICE.

Table 4. New CEMT commands (continued)

CEMT command	Explanation
INQUIRE WORKREQUEST	The INQUIRE WORKREQUEST command is added to enable you to track EJB tasks. You can: • determine which transactions are associated with a single request • correlate all transactions associated with a single request (for example, for accounting purposes)
PERFORM CLASSCACHE	The PERFORM CLASSCACHE command is added to enable you to start and reload the shared class cache, or to phase out, purge or forcepurge the shared class cache and the worker JVMs associated with it. While you are performing one of these operations, you can also change the size of the shared class cache, the JVM profile that is used for the master JVM, or the autostart status of the shared class cache.
PERFORM PIPELINE	Use the command to initiate a scan of the Web service binding directory that is specified in the WSBIND attribute of the PIPELINE definition.
SET CLASSCACHE	The SET CLASSCACHE command is added to enable you to set the status of autostart for the shared class cache.
SET HOST	Use the command to Enable or disable a virtual HOST.
SET PIPELINE	Use the command to enable or disable a PIPELINE.
SET URIMAP	Use the command to enable or disable a URIMAP definition, and apply or remove redirection for a URIMAP definition.
SET WEBSERVICE	Use the command to set the validation status of a WEBSERVICE.
SET WORKREQUEST	The SET WORKREQUEST command is added to enable you to track EJB tasks. You can: • determine which transactions are associated with a single request • correlate all transactions associated with a single request (for example, for accounting purposes) • purge selected work requests .

For detailed information on all the new and changed CEMT transactions and options, see the *CICS Supplied Transactions* manual.

Changed CEMT commands

Table 5 shows those CEMT commands that have changed in some way.

Table 5. Changed CEMT commands

CEMT command	Option	Explanation
INQUIRE CORBASERVER	ENABLESTATUS	These options display: • ENABLESTATUS displays a value indicating the current state of the CorbaServer (DISABLED, DISABLING, DISCARDING, ENABLED, or ENABLING). ENABLESTATUS replaces the STATE option, which is now obsolete.
INQUIRE DISPATCHER	ACTSSLTCBS ACTXPTCBS MAXSSLTCBS and MAXXPTCBS are added	ACTSSLTCBS displays the number of S8 mode open TCBs that are active, and MAXSSLTCBS displays the number that CICS is allowed to attach. ACTXPTCBS displays the number of X8 and X9 mode open TCBs that are active, and MAXXPTCBS displays the number that CICS is allowed to attach.

Table 5. Changed CEMT commands (continued)

CEMT command	Option	Explanation
INQUIRE DOCTEMPLATE	HFSFILE is added	HFSFILE returns the fully-qualified name of the z/OS UNIX System Services HFS file where the template resides.
INQUIRE PROGRAM	APIST is added RUNTIME has a new value	APIST displays the API attribute of the installed program definition. The values are: CICSAPI OPENAPI The XPLINK value for the RUNTIME option means that the program is a C or C++ program which has been compiled using the XPLINK option.
INQUIRE SYSTEM	DEBUGTOOL	Displays a value (DEBUG or NODEBUG) indicating whether debugging profiles will be used to select programs that will run under the control of a debugging tool.
INQUIRE SYSTEM	FORCEQR has a revised description	The description of FORCEQR, and its value FORCE are altered to limit its relevance to CICSAPI programs, because it does not apply to OPENAPI programs.
INQUIRE TCPIP	CRLPROFILE and SSLCACHE are added	CRLPROFILE displays the name of the profile that authorizes CICS to use the certificate revocation lists on an LDAP server. SSLCACHE .displays whether CICS is using local or sysplex caching of session ids. The values are: CICS SYSPLEX .
INQUIRE TCPIPSERVICE	MAXDATALEN is added	MAXDATALEN displays the maximum length of data that may be received by CICS as an HTTP server.
INQUIRE WORKREQUEST	WORKTYPE has a new value.	There is a new value SOAP for the WORKTYPE option.
PERFORM STATISTICS	BEAN, JVMPROFILE, JVMPROGRAM	New resource types for which statistics can be recorded.
PERFORM STATISTICS	PIPELINE and WEBSERVICE are added.	
SET CORBASERVER	DISABLED ENABLED	Specifies whether to enable or disable the CorbaServer.
SET DISPATCHER	MAXSSLTCBS MAXXPTCBS are added	MAXSSLTCBS displays the maximum number of S8 mode open TCBs that CICS is allowed to attach. MAXXPTCBS specifies the maximum number of X8 and X9 mode open TCBs that CICS is allowed to attach.
SET JVMPOOL	TERMINATE	Deletes the shared class cache as well as the JVMs in the JVM pool.

Table 5. Changed CEMT commands (continued)

CEMT command	Option	Explanation
SET PROGRAM	JVMPROFILE	For Java programs, this option specifies the 8–character name of a JVM profile that is to be used for the JVM in which the program runs. Any instances of this program that are currently running in a JVM with the old JVM profile are unaffected, and are allowed to finish running.
SET SYSTEM	DEBUG NODEBUG	Specifies whether debugging profiles will be used to select programs that will run under the control of a debugging tool.
SET SYSTEM	FORCEQR has a revised description	The description of FORCEQR, and its value FORCE are altered to limit its relevance to CICSAPI programs, because it does not apply to OPENAPI programs.
SET TCPIPSERVICE	MAXDATALEN is added	MAXDATALEN specifies the maximum length of data that may be received by CICS as an HTTP server.
SET WORKREQUEST	WORKTYPE	There is a new value SOAP for the WORKTYPE option.

Changes to CETR

The changes to CETR are:
- "Additions to CETR"
- "Controlling tracing for JVMs"

Additions to CETR

The CETR transaction is enhanced to enable you to set special tracing for the following new components:

DP   Debug Tool Interface domain
PI   Pipeline Manager domain

Controlling tracing for JVMs

CETR has new option screens to display and update trace settings for JVMs. Press PF6 on the main screen to access the JVM trace options screens. (Although the JVM trace options are part of the SJ component, they are controlled using the JVM trace options screens, rather than the component trace options screen.) You can use these screens to specify the JVM trace options, using the "free-form" 240–character field, and to specify trace settings for JVMs using the Standard and Special flags. You can then use the Transaction and Terminal Trace screen to switch on these flags for particular transactions. JVM trace can produce a large amount of output, so you should normally activate JVM tracing for special transactions, rather than turning it on globally for all transactions.

The default JVM trace options that are provided in CICS use the JVM trace point level specifications. The default settings for JVM Level 0 trace, JVM Level 1 trace, and JVM Level 2 trace specify LEVEL0, LEVEL1, and LEVEL2 respectively, so they map to the Level 0, Level 1 and Level 2 trace point levels for JVMs. A Level 0 trace point is very important, and this classification is reserved for extraordinary events and errors. Note that unlike CICS exception trace, which cannot be switched off, the JVM Level 0 trace is normally switched off unless JVM tracing is required. The Level 1 trace points and Level 2 trace points provide deeper levels of tracing. The JVM trace point levels go up to Level 9, which provide in-depth component detail. It is suggested that you keep the CICS-supplied level specifications, but if you find that another JVM trace point level is more useful for your purposes than one of the

default levels, you could change the level specification to map to your preferred JVM trace point level (for example, you could specify LEVEL5 instead of LEVEL2 for the JVMLEVEL2TRACE option). The default values for JVM trace options can be overridden using the CICS system initialization parameters JVMLEVEL0TRACE, JVMLEVEL1TRACE, JVMLEVEL2TRACE and JVMUSERTRACE.

You can add further parameters to the basic level specifications for JVM Level 0 trace, JVM Level 1 trace, and JVM Level 2 trace, if you want to include or exclude particular components or trace point types at the selected trace levels. If you want to create more complex specifications for JVM tracing which use multiple trace point levels, or if you do not want to use trace point levels at all in your specification, use the JVMUSERTRACE option to create a trace option string that includes the parameters of your choice. "Defining tracing for JVMs" in the *CICS Problem Determination Guide* has information about the JVM trace options that you can set using the JVM Level 0 trace, JVM Level 1 trace, JVM Level 2 trace, and JVM User trace levels. There is further information about JVM trace and about problem determination for JVMs in the *IBM Developer Kit and Runtime Environment, Java 2 Technology Edition, Version 1.4.2 Diagnostics Guide*, SC34-6358, which is available to download from www.ibm.com/developerworks/java/jdk/diagnosis/.

Changes to transaction CWXN

There are several changes to the processing carried out by the CICS-supplied transaction CWXN, the Web attach transaction. The most significant of these are:

- If a matching URIMAP definition is found for an HTTP request, CWXN now invokes the analyzer program only if instructed to do so by the URIMAP definition.
- Where the HTTP version of the request is HTTP/1.1, CWXN carries out some of the responsibilities of an HTTP server by performing some basic acceptance checks on the request. In response to these checks, CWXN might take action to return a response to the request without involving a user-written application program.
- CWXN pre-processes chunked and pipelined messages received from a Web client, so that user-written applications do not have to perform this processing.
  - Chunked messages are single messages split up and sent as a series of smaller messages (chunks). CWXN receives and assembles the chunks of the message to create a single HTTP request. CWXN checks that the message is complete before passing it to the user application. The user application can then process the request like any other HTTP request.
  - Pipelined messages are multiple messages sent in sequence, where the sender does not wait for a response after each message sent. A server must respond to these messages in the order that they are received. To ensure this, CWXN holds pipelined requests and releases them one at a time to the user application. The user application must send a response to the first request before receiving the next request from CWXN.
- Persistent connections are now the default behavior. The connection is only closed if the Web client requests closure, or if the timeout period is reached, or if the Web client is an HTTP/1.0 client that does not send a Keep-Alive header.
- Before CICS Transaction Server for z/OS, Version 3 Release 1, if a Web client and CICS had a persistent connection, the CWXN transaction would remain in the system for the duration of the persistent connection. Now, the CWXN transaction terminates after each request from the Web client has been passed to the alias transaction (CWBA or another transaction), or after the static response has been delivered. The Sockets listener task monitors the socket and

initiates a new instance of CWXN for each request on the persistent connection. This behavior, known as an asynchronous receive, avoids the possibility of a deadlock in a situation where the maximum task specification (MXT) has been reached, when a CWXN transaction remaining in the system would not be able to attach alias transactions to process further requests. It also means that the maximum number of concurrent connections between CICS and Web clients is no longer limited by the MXT value, but can in theory be up to 64000. In terms of system activity, if you used persistent connections before CICS Transaction Server for z/OS, Version 3 Release 1, you should now see an increased transaction rate, but a decrease in the number of concurrent tasks.

Additions to CICS RACF category 1 transactions

There are some new CICS internal system transactions added to the list of category one transactions. These are the transactions that need to be defined to RACF, and to which the CICS region user ID must be authorized, to enable CICS to initialize successfully when you are running CICS with security enabled (SEC=YES). The new transactions are:

- CJMJ—CICS master JVM transaction
- CPIR
- CPIS
- CRTP
- CWXU

For a full list of all the CICS category 1 transactions, see the *CICS RACF Security Guide*. Also see the DFH$CAT1 CLIST, supplied in the SDFHSAMP library.

Chapter 4. Resource definition (online) changes

This chapter summarizes the changes to CICS resource definition parameters for resources defined in the CICS system definition data set (DFHCSD).It covers the following topics:
- "Changed resource definition attributes"
- "New resource definition types and new attributes" on page 20
- "Other resource definition changes" on page 21
- "Upgrading the CSD" on page 23
- "Sharing the CSD between different releases of CICS" on page 24

For more information about all the new and changed resource definitions, see the *CICS Resource Definition Guide.*

Changed resource definition attributes

Table 6 shows changes to resource definition attributes.

Table 6. Changed resource definition attributes

Resource type	Affected attributes	Explanation
PROFILE	RTIMOUT	Now, in addition to specifying the terminal read time-out feature as in earlier releases, this also specifies the time-out value for IIOP request processor tasks that are waiting for method requests.
		1. The value of RTIMOUT is given as minutes and seconds (mmss) in the range 1--7000. The rounding of this value that was performed by earlier releases of CICS no longer happens. CICS acts upon the RTIMOUT value exactly as it appears in the PROFILE definition. See 1 on page 20 for more information.
		2. The READ TIMEOUT value on the transaction profile definition for MRO sessions is now observed by CICS. The field in the profile already exists. CICS honours it for ISC but used to ignore it for MRO. See 2 on page 20 for more information.
PROGRAM	EXECKEY JVMPROFILE	• The EXECKEY attribute now applies to programs that run in a JVM. You can use the same JVM profile to invoke a JVM in either of the keys.
		• The JVM profiles that you specify using the JVMPROFILE attribute are now files in the HFS directory that is specified by the system initialization parameter JVMPROFILEDIR, and you need to specify the name using the same combination of upper and lower case characters that is present in the HFS file name.

Table 6. Changed resource definition attributes (continued)

Resource type	Affected attributes	Explanation
TCPIPSERVICE	PORTNUMBER TRANSACTION	The description of the PORTNUMBER attribute has been extended, with information regarding the use of well-known IIOP port numbers and port sharing within an MVS™ image. CIEP, for an ECI over TCP/IP TCPIPSERVICE definition, is added to the CICS transactions you can specify on the TRANSACTION attribute.
TRANSACTION	SHUTDOWN	The meaning of this parameter is changed. It now applies to ALL transactions whether the transaction is associated with a terminal or not. Therefore, SHUTDOWN(ENABLED) should be specified on all TRANSACTIONS that you want to run during shutdown. Alternatively, specify them on the XLT used at shutdown.

Notes:

1. In earlier releases, CICS rounded this value up to an exact multiple of 16.78 seconds, for all communications methods. This rounding is now removed, CICS acts upon the RTIMOUT value exactly as it appears in the PROFILE definition. If you have set particular RTIMOUT values, to take account of the rounding up to a multiple of 16.78 seconds, in order to achieve a particular behavior in your CICS region, you may need to reconsider the values that you have set.

2. In earlier Releases of CICS, read time-out is ignored for MRO sessions, tasks waiting on an MRO connection can hang indefinitely. If, for example, transactions in an AOR stall or deadlock for any reason the corresponding relay transactions in an MRO-connected TOR are left hanging. Eventually a problem in an AOR can cause the TOR to stall completely. Although you can prevent new transactions from being routed to, or queued for, the offending AOR, there has not been, until now, an easy way to purge every task that is in flight. This change to CICS enables the READ TIMEOUT value on the transaction profile definition for MRO sessions. The field in the profile already exists. CICS honours it for ISC but used to ignore it for MRO. CICS now checks this RTIMOUT value for MRO sessions. A new abend code is issued if the RTIMOUT value is exceeded for MRO sessions. By enabling the READ TIMEOUT value on the transaction profile definition for MRO sessions, a cause of intersystem queuing (sometimes referred to as "sympathy sickness") is eliminated. The situations which may benefit from this change include some which, because they result from stalls or deadlocks elsewhere, are unlikely to resolve themselves unaided.

New resource definition types and new attributes

Table 7 on page 21 shows new resource definition types and new attributes.

If you have existing resource definitions that were created before the new attributes for the resources were available, check those resource definitions after you upgrade to this CICS release, to ensure that the default values are suitable for your situation. In some situations, CICS enforces certain values for options if these are required for compatibility with existing options in your resource definition.

Table 7. New resource definition attributes

Resource type	New keywords	Explanation
CORBASERVER	STATUS({**Enabled**|Disabled})	Specifies whether the CorbaServer is to be installed in enabled or disabled state. The default is enabled.
PIPELINE		For details, see the *CICS Resource Definition Guide*.
URIMAP		For details, see the *CICS Resource Definition Guide*.
WEBSERVICE		For details, see the *CICS Resource Definition Guide*.

Other resource definition changes

This section describes some other CSD changes affecting IBM-supplied resource definitions. The topics covered are:
- "Additions to IBM-supplied resource definitions"
- "Changes to IBM-supplied resource definitions"
- "Obsolete IBM-supplied resource definitions" on page 22
- "Updating user-modified IBM-supplied definitions" on page 22
- "Migrating copies of CICS-supplied resource definitions" on page 23

Additions to IBM-supplied resource definitions

There are new groups of resource definitions added to your CSD when you run the UPGRADE command:
- DFHDP
- DFHDPWB
- DFHSO

DFHDP
IBM-supplied group DFHDP contains the resource definitions for the new Application debugging profile manager 3270 interface (the CADP transaction), and for the Inactivate debugging profiles utility. The group contains:

PROGRAM definitions
DFHDPLU, DFHDPIN and DFHDPCP
TRANSACTION definitions
CADP and CIDP
MAPSET definitions
DFHDPMS

DFHDPWB
IBM-supplied group DFHDPWB contians the resource definitions for the new application debugging profile manager web interface. The group contains:

PROGRAM definitions
DFHDPWB, DFHDPWM0, DFHDPWM1, DFHDPWM2, DFHDPWM3, DFHDPWM4, DFHDPWM5, DFHDPWM6, DFHDPWT0, and DFHDPWF0

DFHSO
IBM-supplied group DFHSO contains the resource definitions for External sockets support. The group contains:

PROGRAM definitions
DFHSOCI and DFHSOLI

Changes to IBM-supplied resource definitions

Some IBM-supplied resource definitions are changed or obsolete, and are moved to new compatibility groups.

Following the upgrade of your CSD, the resource definitions listed below no longer exist in their old groups, which are removed from DFHLIST. If you plan to share the upgraded CSD with earlier releases of CICS, you must remove the obsolete group names from the group lists you use on the earlier releases, and add groups DFHCOMP5, DFHCOMP6, DFHCOMP7, DFHCOMP8, DFHCOMP9, and DFHCOMPA, (or some of these) in their place. See Chapter 6, "The application programming interface (API)," on page 29.

Moved to group DFHCOMPA

Group DFHJIIRP is obsolete. The definition that was in this group is now in the new compatibility group, DFHCOMPA.

When you upgrade the CSD using DFHCSDUP, DFHCOMPA is locked.

Obsolete IBM-supplied resource definitions

The following groups are obsolete:

- DFH$JAVA
- DFHAUGRP
- DFH$AFFY

DFH$JAVA

IBM-supplied sample application program group DFH$JAVA is removed. This group contained the resource definitions needed for the sample applications for Java support using VisualAge® for Java, Enterprise Edition for OS/390®. The same sample applications are defined for use with a JVM by the DFH$JVM group.

DFHAUGRP

IBM-supplied group DFHAUGRP is removed. This group contained the resource definitions for the CICS transaction affinities utility.

DFH$AFFY

IBM-supplied sample group DFH$AFFY is removed. This group contained sample resource definitions for the CICS transaction affinities utility that you could modify to suit your requirements.

Removing obsolete definition groups from startup group lists.

Obsolete definition groups have been removed from the CICS-supplied default startup group list, DFHLIST.

If you use customized startup group lists, you must remove any obsolete definition groups from them.

Updating user-modified IBM-supplied definitions

When you run the UPGRADE function of the CSD utility program (DFHCSDUP), ensure that you manually upgrade any IBM-supplied definitions that you may have modified on earlier releases. The safest way to do this is to copy the upgraded IBM-supplied definitions and reapply your modifications. This action is required because the UPGRADE command does not operate on your own groups, or on IBM groups that you have copied.

It is important to upgrade these modified definitions to ensure that they are defined correctly with non-default values for attributes that are new. If you fail to upgrade modified definitions, CICS assigns default values to any new attributes, and these may be inappropriate for IBM-supplied resource definitions.

If you are not sure whether your CSD contains any modified IBM definitions, use the DFHCSDUP SCAN function to compare the IBM-supplied resource definitions with any user-modified versions.

The SCAN function searches for the IBM-supplied version of a specified resource name of a specific resource type and compares it with any other resource definition of the same name and type. DFHCSDUP reports any differences it finds between the IBM-supplied definition and a user-modified version. If you have copied and changed the name of an IBM-supplied definition, the SCAN command enables you to specify the changed name as an alias.

See the the *CICS Operations and Utilities Guide* for details of the DFHCSDUP SCAN command.

Migrating copies of CICS-supplied resource definitions

If you have made copies of CICS-supplied resource definitions, you may need to change your copies to match the changes which have been made to the supplied definitions for this release.

To help you, member DFH$CSDU in library SDFHSAMP contains ALTER commands that you can apply using the CSD utility program (DFHCSDUP).

1. Review your resource definitions to determine if you have copied any CICS-supplied definitions.
2. Review DFH$CSDU to determine if the changes which it contains should apply to your resource definitions.
3. Make any necessary changes to DFH$CSDU. It is advisable to make a copy of DFH$CSDU, and apply any changes to the copy.
4. Run DFHCSDUP using your modified version of DFH$CSDU as input.

   Note: As supplied, the ALTER commands in DFH$CSDU specify `GROUP(*)`, which means that DFHCSDUP will attempt to change resources in the CICS-supplied groups. This is not permitted, and will result in message DFH5151. You can ignore this message.

Example

In CICS TS 2.3, JVMPROFILE(DFHJVMCD) was added to the definition of program DFHADJR. Therefore, DFH$CSDU contains the following command:

```
ALTER PROGRAM(DFHADJR) GROUP(*) JVMPROFILE(DFHJVMCD)
```

When you you run DFHCSDUP, the attribute is added to the definitions of program DFHADJR in all groups. Other attributes are unchanged.

Upgrading the CSD

Upgrade the IBM supplied definitions using the CSD utility UPGRADE command "Running the DFHCSDUP UPGRADE job" on page 24.

When you have successfully upgraded your CSD, you can review the topic "Sharing the CSD between different releases of CICS" on page 24 and plan what you need to do to share your CSD.

Running the DFHCSDUP UPGRADE job

When you have redefined your CSD with the correct record size, run the DFHCSDUP utility program, specifying the UPGRADE command, to upgrade the IBM-supplied definitions in your CSD to the latest CICS TS level. You can create a new CSD using the DFHCSDUP INITIALIZE command. For information about running DFHCSDUP with the UPGRADE command, see the the *CICS Operations and Utilities Guide*.

Upgrading other IBM-supplied resource definitions

If you have resource definitions in your CSD that support other IBM products, you may need to upgrade these also. For example, if your Language Environment resource definitions are not at the **z/OS Version 1 Release 4** level, you are recommended to delete and replace the CSD group containing these.

You can find the Language Environment resource definitions in the SCEESAMP library in member CEECCSD. The following job is an example of how to upgrade the Language Environment resource definitions in your CSD:

```
//CSDUPGRD JOB 1,WALSH,MSGCLASS=A,MSGLEVEL=(1,1),
//         CLASS=A,NOTIFY=BELL
/*JOBPARM SYSAFF=MV26
//*  Remove Old Language Environment group
//CSDUP1   EXEC PGM=DFHCSDUP,REGION=2M,PARM='CSD(READWRITE)'
//STEPLIB  DD DSN=CICSTS31.CICS.SDFHLOAD,DISP=SHR
//DFHCSD   DD DSN=CICSTS31.CICSHURS.DFHCSD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
 DELETE GROUP(CEE)
/*
//*
//CSDUP2   EXEC PGM=DFHCSDUP,REGION=2M,PARM='CSD(READWRITE)'
//STEPLIB  DD DSN=CICSTS31.CICS.SDFHLOAD,DISP=SHR
//DFHCSD   DD DSN=CICSTS31.CICSHURS.DFHCSD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD DSN=SYS1.ZOS140.SCEESAMP(CEECCSD),DISP=SHR
/*
//
```

Figure 1. Upgrading Language Environment resource definitions

The Language Environment group CEE contains mostly the program resource definitions needed for all high-level language support, but also contains the mapset and transaction definition for the Language Environment CLER transaction

Sharing the CSD between different releases of CICS

Most releases of CICS make changes to the IBM-supplied groups of resource definitions that are included in the DFHLIST group list. In all such cases, the old versions of the CICS resource definitions are retained in compatibility groups, which are needed to support earlier releases if you share the CSD between different levels of CICS.

If, after upgrading a CSD, you plan to share the CSD with earlier releases of CICS, include the appropriate DFHCOMP*x* compatibility groups in your start-up group list

to provide the required support for earlier releases. Table 8 shows you which DFHCOMP groups you need to include for the earlier releases. Do not attempt to share a CSD with a CICS region running at a higher level than the CSD.

It is important that you install the compatibility groups in the correct order, as shown in Table 8. For example, to run a CICS/ESA 4.1, with the CSD upgraded to CICS TS 3.1, append the compatibility group DFHCOMP9 followed by DFHCOMP8, DFHCOMP7, DFHCOMP6, and DFHCOMP5 at the end of your group list.

Table 8. Required compatibility groups for earlier releases of CICS

CICS release the CSD is shared with	The CICS release level of the CSD						
	CICS TS 3.1	**CICS TS 2.3**	**CICS TS 2.2**	**CICS TS 1.3**	**CICS TS 1.2**	**CICS TS 1.1**	**4.1**
CICS TS 2.3	None	None	Do not share	Do not share	Do not share	Do not share	Do not share
CICS TS 2.2	DFHCOMPA	DFHCOMPA	None	Do not share	Do not share	Do not share	Do not share
CICS TS 1.3	DFHCOMP9 DFHCOMP8	DFHCOMP9 DFHCOMP8	DFHCOMP8	None	Do not share	Do not share	Do not share
CICS TS 1.2	DFHCOMP9 DFHCOMP8 DFHCOMP7	DFHCOMP9 DFHCOMP8 DFHCOMP7	DFHCOMP8 DFHCOMP7	DFHCOMP7	None	Do not share	Do not share
CICS TS 1.1	DFHCOMP9 DFHCOMP8 DFHCOMP7 DFHCOMP6	DFHCOMP9 DFHCOMP8 DFHCOMP7 DFHCOMP6	DFHCOMP8 DFHCOMP7 DFHCOMP6	DFHCOMP7 DFHCOMP6	DFHCOMP6	None	Do not share
4.1	DFHCOMP9 DFHCOMP8 DFHCOMP7 DFHCOMP6 DFHCOMP5	DFHCOMP9 DFHCOMP8 DFHCOMP7 DFHCOMP6 DFHCOMP5	DFHCOMP8 DFHCOMP7 DFHCOMP6 DFHCOMP5	DFHCOMP7 DFHCOMP6 DFHCOMP5	DFHCOMP6 DFHCOMP5	DFHCOMP5	None

Chapter 5. Resource definition (macro) changes

This chapter summarizes the changes to the CICS resource definition macros for CICS control tables. It discusses the following topics:
- "Changed control table parameters"
- "Monitoring control table, DFHMCT"
- "Reassembling control tables" on page 28

Changed control table parameters

Table 9 shows changes to control table parameters.

Table 9. Changed control table parameters

Resource type	Affected attributes	Explanation
DFHTCT	TYPE=SDSCI TYPE=LINE TYPE=TERMINAL	CICS no longer supports local TCAM terminals. These resource definition macro can no longer be used to define local TCAM terminals. See 27 for more detail.
DFHMCT	TYPE=RECORD	Because of the removal of run-time support for Java program objects and hot-pooling (HPJ), DFHTASK field 278, CICS MAXHPTCBS delay time, is no longer relevant. and is removed.

Note: It is still possible to define remote TCAM terminals. You can do this using either of the following methods:
1. A single DFHTCT TYPE=REMOTE macro.
2. A DFHTCT TYPE=REGION macro, followed by a DFHTCT TYPE=LINE and a DFHTCT TYPE=TERMINAL macro. CICS uses only the "remote" attributes of the DFHTCT TYPE=LINE and DFHTCT TYPE=TERMINAL macros.

CICS no longer supports BTAM terminals, even indirectly. You can no longer define BTAM terminals, even as remote resources.

Monitoring control table, DFHMCT

You can specify some new values on the INCLUDE and EXCLUDE operands of the DFHMCT TYPE=RECORD macro. These values allow you to include or exclude specific fields from performance-class monitoring records. The new values are shown in Table 10.

Table 10. Fields that can be included or excluded from performance-class monitoring records

Group Name	Field Id	Description
DFHEJBS	311	CorbaServer for which the request processor instance is handling requests
DFHEJBS	312	Number of enterprise bean activations that have occurred in this request processor

Table 10. Fields that can be included or excluded from performance-class monitoring records (continued)

Group Name	Field Id	Description
DFHEJBS	313	Number of enterprise bean passivations that have occurred in this request processor
DFHEJBS	314	Number of enterprise bean creation calls that have occurred in this request processor
DFHEJBS	315	Number of enterprise bean removal calls that have occurred in this request processor
DFHEJBS	316	Number of enterprise bean method calls executed in this request processor
DFHEJBS	317	Total for this request processor of fields 312–316
DFHTASK	252	User-task peak open TCB count
DFHTASK	264	User-task key 9 TCB dispatch time
DFHTASK	265	User-task key 9 TCB CPU time
DFHTASK	267	User-task J9 TCB CPU Time
DFHTASK	268	User-task TCB mismatch wait time
DFHTASK	279	MVS storage constraint delay time

Reassembling control tables

Reassemble *all* CICS control tables using the CICS TS 3.1` macro libraries, even if there are no changes to the macro externals. This applies also to tables that you are reassembling only to migrate them to the CSD.

DFHCNV: The requirement to reassemble your control tables applies also to any DFHCNV data conversion tables that you use. This is because CICS initialization fails when trying to load DFHCNV tables assembled using macros from an earlier release.

Chapter 6. The application programming interface (API)

This chapter summarizes the changes affecting the CICS application programming interface (API).

> **Program compatibility**
>
> Except for the specific cases described in this chapter, CICS TS provides compatibility with future releases, at source and object level, for all CICS application programs that are written to the CICS application programming interface, and which execute correctly under the previous release.
>
> For information about CICS support for application programming languages, see the *CICS Transaction Server for z/OS Release Guide*.

Although CICS provides API compatibility from release to release, functional changes to some CICS components can affect some of the CICS API commands. The effects of these functional changes are discussed in the following topics:
- "Obsolete commands and options" on page 30
- "New commands and options" on page 30
- "Changed commands and options" on page 31
- "Member DFHEILID has moved" on page 32
- "Changes to RESP2 values" on page 32

Note: In addition to the changes affecting the API, there are extensions to some EXEC CICS commands, and these are described in the the *CICS Transaction Server for z/OS Release Guide*.

Obsolete commands and options

Table 11 shows the CICS application programming interface commands and options that are now obsolete.

Table 11. Obsolete API commands and options

Command	Options	Comments
• CONVERSE (SYSTEM/3) • CONVERSE (SYSTEM/7) • CONVERSE (2741) • CONVERSE (2770) • CONVERSE (2780) • CONVERSE (3600 BTAM) • CONVERSE (3735) • CONVERSE (3740) • ISSUE COPY (3270 display) • RECEIVE (SYSTEM/3) • RECEIVE (SYSTEM/7) • RECEIVE (2741) • RECEIVE (3600 BTAM) • RECEIVE (3735) • RECEIVE (3740) • SEND (SYSTEM/3) • SEND (SYSTEM/7) • SEND (2741) • SEND (3600 BTAM) • SEND (3735) • SEND (3740)		All these BTAM-related, EXEC CICS API commands are obsolete.

New commands and options

Table 12 shows the CICS application programming interface commands and options that are new.

Table 12. New API commands and options

Command	Options	Comments
• EXEC CICS DELETE CONTAINER (CHANNEL) • EXEC CICS GET CONTAINER (CHANNEL) • EXEC CICS MOVE CONTAINER (CHANNEL) • EXEC CICS PUT CONTAINER (CHANNEL) • EXEC CICS START TRANSID CHANNEL		
• EXEC CICS INVOKE WEBSERVICE • EXEC CICS SOAPFAULT ADD • EXEC CICS SOAPFAULT CREATE • EXEC CICS SOAPFAULT DELETE		

Table 12. New API commands and options (continued)

Command	Options	Comments
• EXEC CICS WEB OPEN • EXEC CICS WEB CONVERSE • EXEC CICS WEB CLOSE • EXEC CICS WEB SEND (CLIENT) • EXEC CICS WEB RECEIVE (CLIENT) • EXEC CICS WEB PARSE URL • EXEC CICS WEB CONVERTTIME • EXEC CICS WEB WRITE HTTPHEADER • EXEC CICS WEB READ HTTPHEADER • EXEC CICS WEB STARTBROWSE HTTPHEADER • EXEC CICS WEB READNEXT HTTPHEADER • EXEC CICS WEB ENDBROWSE HTTPHEADER • EXEC CICS WEB EXTRACT • EXEC CICS FORMATTIME • EXEC CICS EXTRACT TCPIP	SESSTOKEN is added to the EXEC CICS WEB xxx HTTPHEADER commands. MAXDATALEN	EXEC CICS WEB SEND and EXEC CICS WEB RECEIVE have a new range of options when used for CICS as an HTTP client: There are changes to the options available on most of the remaining EXEC CICS WEB commands. The changed commands can be used for both CICS as an HTTP client, and CICS as an HTTP server.

Changed commands and options

Table 13 shows the CICS application programming interface commands and options that are changed.

Table 13. Changed API commands and options

Command	Options	Comments
• EXEC CICS VERIFY PASSWORD		See "Changes to EXEC CICS VERIFY PASSWORD."

Changes to EXEC CICS VERIFY PASSWORD

When you issue the EXEC CICS VERIFY PASSWORD command, CICS now
enforces the revoked status of a user ID or a user's group connection. The new
method CICS uses to verify the password is more efficient, but you might notice
changes to the output that is produced when verification takes place.

CICS now attempts to verify a password using a RACROUTE
REQUEST=EXTRACT request to the external security manager. If the password
cannot be verified using this method, CICS uses a RACROUTE
REQUEST=VERIFYX call. Before CICS TS for z/OS, Version 3.1, CICS always
used the RACROUTE REQUEST=VERIFYX call, which is more expensive.

There are some differences in the output produced by the external security
manager for the old and new methods of verifying a password. If your application
programs relied on the output produced by the old method, you need to change
them so that they do not depend on this output. The differences are:
• ESMRESP and ESMREASON codes are not supplied by the external security
manager for the new method of verifying a password using a RACROUTE
REQUEST=EXTRACT call. These are only produced if CICS needs to use the
old method with a RACROUTE REQUEST=VERIFYX call. Your application

programs should always check the EIBRESP and EIBRESP2 values returned by
the EXEC CICS VERIFY PASSWORD command, and not rely on the ESMRESP
and ESMREASON codes.
• Message ICH70002I is not produced by the external security manager for the
new method of verifying a password. The message is only produced if CICS
needs to use the old method with a RACROUTE REQUEST=VERIFYX call. (The
SETR PASSWORD(WARN(nn)) option must also be active in the external
security manager for the message to be produced.) You should therefore not rely
on receiving this message.

Member DFHEILID has moved

Member DFHEILID has moved from the SDFHC370 library to the SDFHSAMP
library. The change affects the COPYLINK step of the sample jobs supplied by
CICS and CICSPlex SM to translate, compile, link-edit and install application
programs. The sample jobs supplied by CICS have names in the format DFHwxTyL
(for example, DFHYITDL, DFHZITCL) and are supplied in the SDFHPROC library. If
you are using an older version of these supplied sample jobs, or if you have
created your own job based on the supplied samples, ensure that you apply the
change for DFHEILID to the COPYLINK step in the job.

Changes to RESP2 values

There are some new RESP2 values for:

• "Program control RESP2 values"

These changes are described below.

Program control RESP2 values

There is a new RESP2 value to qualify the INVREQ response to EXEC CICS LINK
commands.

The new INVREQ RESP2 values are:

49   The shared class cache is STOPPED and autostart is disabled, so a Java
program requesting use of the shared class cache cannot be executed.

The RESP2 value 41 that qualified the INVREQ response to EXEC CICS LINK
requests is removed, because it is now possible to link to more than one JVM
program in the same CICS task:

41 A LINK has been attempted to a JVM program but there is already a JVM
program on the link stack (only one JVM program is allowed on the program
stack).

The new PGMIDERR RESP2 value is:

42   An attempt has been made to LOAD or RELEASE a JVM program. This is
not allowed, because Java byte codes programs are not managed by the
CICS loader domain.

Chapter 7. The system programming interface (SPI)

This chapter summarizes the changes affecting the CICS system programming interface (SPI). It covers the following topics:

- "New commands and options"
- "Changed commands and options" on page 35
- "Obsolete commands and options" on page 36
- "Obsolete CVDA values" on page 37
- "Release levels on INQUIRE SYSTEM command" on page 37

Program compatibility

The system programming commands operate on CICS system resources, such as control blocks and tables of resource definitions (and not on user resources, such as data, on which the API operates).

The SPI is also sensitive to the underlying environment in which it is implemented, and as a consequence compatibility with future releases of CICS cannot be guaranteed.

This chapter describes the effect on the SPI of the functional changes in CICS TS, explaining where incompatibilities exist, to enable you to make programming changes where necessary.

Except for the instances given in this chapter, CICS continues to provide compatibility with future releases , at source and object level, for application programs that use the unaffected SPI commands.

New commands and options

Table 14 shows the new SPI commands and options.

Table 14. New commands and options

Commands	Explanation
CREATE PIPELINE	The CREATE PIPELINE command defines a PIPELINE in the local CICS region.
CREATE URIMAP	The CREATE URIMAP command defines a URIMAP in the local CICS region.
CREATE WEBSERVICE	The CREATE WEBSERVICE command defines a WEBSERVICE in the local CICS region.
DISCARD PIPELINE	The DISCARD PIPELINE command removes a PIPELINE from the CICS system and the CICS catalog.
DISCARD URIMAP	The DISCARD URIMAP command removes a URIMAP from the CICS system and the CICS catalog.
DISCARD WEBSERVICE	The DISCARD WEBSERVICE command removes a WEBSERVICE from the CICS system and the CICS catalog.

Table 14. New commands and options (continued)

Commands	Explanation
EXTRACT STATISTICS	The EXTRACT STATISTICS command is added to provide statistics about new resource types, because the design of COLLECT STATISTICS prevented its further expansion. The syntax of EXTRACT STATISTICS is not like COLLECT STATISTICS, but the results of using EXTRACT STATISTICS are equivalent to those produced by COLLECT STATISTICS. For a full description of EXTRACT STATISTICS see the *CICS System Programming Reference*
INQUIRE CLASSCACHE	The INQUIRE CLASSCACHE command is added to give you information about the active shared class cache in the CICS region, and report the presence of any old shared class caches that are awaiting deletion.
INQUIRE JVM	The INQUIRE JVM command is added to enable you to identify JVMs in a CICS region and get information about their status. You can inquire on the status of a particular JVM, or browse through all of the JVMs in a CICS region by using the browse options (START, NEXT, and END) on the command.
INQUIRE JVMPROFILE	The INQUIRE JVMPROFILE command is added to retrieve the full path name of the HFS file for a JVM profile, and tell you whether or not a JVM with this profile uses the shared class cache. The command only finds JVM profiles that have been used during the lifetime of this CICS region, for JVMs that can run applications. You can inquire on a particular JVM profile, or browse through all of the JVMPROFILEs that have been used in a CICS region by using the browse options (START, NEXT, and END) on the command.
INQUIRE PIPELINE	The INQUIRE PIPELINE command retrieve information about an installed PIPELINE.
INQUIRE URIMAP	The INQUIRE URIMAP command retrieves information about URIMAP resource definitions.
INQUIRE WEBSERVICE	The INQUIRE WEBSERVICE command retrieves information about an installed WEBSERVICE.
INQUIRE WORKREQUEST	The INQUIRE WORKREQUEST command is added to enable you to track EJB tasks. You can: • determine which transactions are associated with a single request • correlate all transactions associated with a single request (for example, for accounting purposes)
PERFORM PIPELINE	The PERFORM PIPELINE command initiates a scan of the Web service binding directory that is specified in the WSBIND attribute of the PIPELINE definition.
SET HOST	The SET HOST command enables or disables a virtual HOST.
SET PIPELINE	The SET PIPELINE command enables or disables a PIPELINE.
SET URIMAP	The SET URIMAP command enables or disables a URIMAP definition, and applies or removes redirection for a URIMAP definition.
SET WEBSERVICE	The SET WEBSERVICE command sets the validation status of a WEBSERVICE.

See the *CICS System Programming Reference* for more information on the changed and new commands and options.

Changed commands and options

Table 15 shows the system programming interface commands and options that are changed.

Table 15. Changed system programming commands

Command	Option	Description of change
COLLECT STATISTICS	BEAN JVMPROFILE JVMPROGRAM	Options are added to retrieve statistics for these resource types.
CREATE CORBASERVER	STATUS	These options define: The STATUS option specifies whether the CorbaServer is to be installed in enabled or disabled state. The default is enabled.
INQUIRE CORBASERVER	ENABLESTATUS	Options are added to return: • The ENABLESTATUS option returns a CVDA value indicating the current state of the CorbaServer (DISABLED, DISABLING, DISCARDING, ENABLED, or ENABLING). ENABLESTATUS replaces the STATE option, which is now obsolete.
INQUIRE JVMPOOL	JVMLEVEL0TRACE, JVMLEVEL1TRACE, JVMLEVEL2TRACE, JVMUSERTRACE	Return a 240–character data value giving the current options for JVM tracing, which are used when the associated level of tracing has been activated using the CICS-supplied transaction CETR, the SET TRACETYPE command, or the system initialization parameter SPCTRSJ.
INQUIRE SYSTEM	DEBUGTOOL	Options are added to return: • a CVDA value (DEBUG or NODEBUG) indicating whether debugging profiles will be used to select programs that will run under the control of a debugging tool.
INQUIRE WORKREQUEST	WORKTYPE	A new value, SOAP, is added
PERFORM STATISTICS RECORD	BEAN JVMPROFILE JVMPROGRAM PIPELINE	Options added to write statistics for these new resource types.
SET PROGRAM	JVMPROFILE	• The JVMPROFILE option specifies the 8–character name of a JVM profile that is to be used for the JVM in which the program runs. Any instances of this program that are currently running in a JVM with the old JVM profile are unaffected, and are allowed to finish running.

Table 15. Changed system programming commands (continued)

Command	Option	Description of change
SET SYSTEM	DEBUGTOOL(*cvda*)	The DEBUGTOOL option specifies whether debugging profiles will be used to select programs that will run under the control of a debugging tool.
SET WORKREQUEST	WORKTYPE	A new value, SOAP, is added

See the *CICS System Programming Reference* manual for information on the changed and new commands and options.

Obsolete commands and options

Table 16 shows the system programming interface commands and options that are now obsolete.

Table 16. Obsolete system programming commands and options

Command	Options	Comments / Explanation
CREATE PROGRAM	HOTPOOL	The CVDA values YES and NO were used with this option to specify whether or not the Java program object was to be run in a preinitialized Language Environment enclave.
INQUIRE CORBASERVER	STATE	The option STATE is replaced by ENABLESTATUS.
INQUIRE DISPATCHER	ACTHPTCBS MAXHPTCBS	ACTHPTCBS displayed the number of H8 mode open TCBs that were active, and MAXHPTCBS displayed the number that CICS was allowed to attach. H8 mode open TCBs no longer exist.
INQUIRE PROGRAM	HOTPOOLING HOTPOOL NOTHOTPOOL	The values Hotpool and Nothotpool were used to show whether or not the Java program object was to be run in a preinitialized Language Environment enclave.
SET DISPATCHER	MAXHPTCBS	This option used to specify the maximum number of H8 mode open TCBs that CICS was allowed to attach. H8 mode open TCBs no longer exist.
SET PROGRAM	HOTPOOL NOTHOTPOOL	The values Hotpool and Nothotpool were used to show whether or not the Java program object was to be run in a preinitialized Language Environment enclave.

Signon retention with XRF and VTAM persistent sessions

In earlier releases of CICS that support VTAM® persistent sessions, CICS recovers only the terminal session, and not the user's signon status. With signon retention support, CICS catalogs the signon status of every user who signs on, enabling CICS to retain a terminal's signon in the event of either a CICS or VTAM failure. Thus, CICS regions using VTAM persistent sessions have the same signon retention capability as CICS regions using XRF. However, XRF and VTAM persistent sessions are mutually exclusive, and rather than have two parameters to control signon and signoff status, RSTSIGNOFF operates for both functions.

If you have application programs that specify the XRFSIGNOFF(FORCE) attribute on an EXEC CICS CREATE TYPETERM command, this attribute is ignored in a CICS TS 2.2 region, which does not recognize the XRFSIGNOFF attribute. To

ensure such application programs have the same effect as before in CICS regions running with XRF support, modify the EXEC CICS CREATE command to specify RSTSIGNOFF(FORCE).

Obsolete CVDA values

Table 17 shows the CVDA values that are now obsolete.

Table 17. Obsolete CVDA values

CVDA	value
HOTPOOL	1065
NOTHOTPOOL	1066

Release levels on INQUIRE SYSTEM command

You are recommended to use the EXEC CICS INQUIRE SYSTEM CICSTSLEVEL(data_area) command to determine the Version and Release number, and hence the function level, of CICS. CICS returns 030100 for CICS TS for z/OS Version 3 Release 1. Use the EXEC CICS INQUIRE SYSTEM OSLEVEL(data_area) command to determine the level of z/OS; CICS returns 010400 for z/OS Release 1.4.

To ensure compatibility with previous releases, the CICS base element maintains its own level (identification) number. Each time new function is added to CICS and shipped with the CICS Transaction Server product, the CICS level number is incremented. The CICS version and release number are no longer implicit in the CICS level number.

The CICS level number in CICS TS 3.1 is 0640. This number is returned in the RELEASE parameter of the INQUIRE SYSTEM command. The 0640 number also appears in other forms such as 6.4.0 in output from offline utilities such as statistics and dump formatters to identify the level of utility being used, and as the suffix in module names such as DFHPD640.

Chapter 8. CICS-supplied utility programs

This chapter summarizes changes affecting CICS-supplied utility programs. It covers the following topics:
- "Changed utility programs"
- "New EJB utility sample programs" on page 41

Changed utility programs

There are changes to the following utility programs:
- "Changes to the CSD utility program, DFHCSDUP"
- "Changes to the statistics formatting utility program, DFHSTUP"
- "Changes to the trace formatting utility program, DFHTU640"
- "Changes to the IPCS dump exit routine, DFHPD640" on page 40
- "Changes to the Resource Manager for Enterprise Beans" on page 40
- "Changes to WebSphere utilities" on page 40

Changes to the CSD utility program, DFHCSDUP

The CSD utility program is enhanced to support new and changed resource types and attributes. See Chapter 4, "Resource definition (online) changes," on page 19 for details of all the changes to CSD resource definitions that are supported by DFHCSDUP.

Updating obsolete resource definitions

If you are sharing the CSD with earlier releases of CICS, and want to alter definitions that are used only on earlier releases, you must use the latest DFHCSDUP, even if some attributes are obsolete in the latest releases of CICS. To use the latest DFHCSDUP to update obsolete options on resource definitions, specify the COMPAT option in the PARM string to indicate that you want DFHCSDUP to operate in compatibility mode.

Changes to the statistics formatting utility program, DFHSTUP

The program is enhanced to format additional statistics reports for the new resource types.

These resource types can be coded on the SELECT TYPE and IGNORE TYPE parameters using the keywords
- CORBASERVER
- JVMPOOL
- REQUESTMODEL
- TCP/IP

See the *CICS Performance Guide* for details of statistics data.

Changes to the trace formatting utility program, DFHTU640

The trace formatting utility program is renamed to DFHTU640, where 640 is the level number of CICS. Always ensure you use the trace program with the correct level number for the release of CICS TS that created the trace data set you are formatting.

The program is enhanced to format trace entries written by the new domains and functions. The new identifiers that you can specify to DFHTU640 on the TYPETR parameter for these functional areas are the same as the CETR trace component codes.

See "Changes to CETR" on page 15 for a list of these.

Changes to the IPCS dump exit routine, DFHPD640

The dump formatting utility program is renamed to DFHPD640, where 640 is the level number of CICS. Always ensure you use the dump formatting program with the correct level number for the release of CICS TS that created the dump data set you are formatting.

The dump exit routine for formatting CICS system dumps is enhanced to format the control blocks for the new domains. To select or ignore dump data for any domains, specify the dump component keywords for those domains. The dump component keywords for use with the CICS IPCS dump exit routine are the same as the CETR trace component codes. See "Changes to CETR" on page 15 for a list of these.

Changes to the Resource Manager for Enterprise Beans

You can now use the Resource Manager for Enterprise Beans to determine which version of GIOP is advertised in the published IOR for a specific enterprise bean.

For details of the Resource Manager for Enterprise Beans, see the *CICS Operations and Utilities Guide*.

Changes to WebSphere utilities

CICS TS for z/OS, Version 2.3 ships with WebSphere Application Server Version 5. This section describes how the WebSphere® utilities used with CICS differ from their counterparts in WebSphere Application Server Version 4, which was shipped with CICS TS for z/OS, Version 2.2.

The Assembly Toolkit (ATK) and the Application Assembly Tool (AAT)

The Assembly Toolkit (ATK) and the Application Assembly Tool (AAT) can both be used to deploy enterprise beans for use with CICS TS for z/OS, Version 2.3. It is recommended that you use the most recent version available to you.

- The Assembly Toolkit (ATK) is a component of the Application Server Toolkit (ASTK), which is supplied with WebSphere Application Server Version 5.1. (ASTK is also available through other distribution channels.)
- The Application Assembly Tool (AAT) is the predecessor to the ATK, and is a component of WebSphere Application Server Advanced Single Server Edition Version 4.0, and WebSphere Application Server Version 5.0. It is not included in WebSphere Application Server Version 5.1.

If you are using ATK to deploy your enterprise beans, you can choose the format that is used for saving each ejb-jar file, such as EJB 1.1 or EJB 2.0. It is recommended that where possible, you create deployment descriptors for EJB 1.1.

The AAT supplied with WebSphere Application Server Version 5.0 (but not that supplied with WebSphere Application Server Version 4.0) supports EJB 2.0 JAR files. If you use the WebSphere Version 5.0 AAT to edit an EJB 1.1 ejb-jar file, by default AAT saves the edited file in EJB 1.1 format. However, if you create a *new* ejb-jar file, AAT saves it in EJB 2.0 format.

Although CICS currently supports only Version 1.1 of the Enterprise JavaBeans specification, it tolerates EJB 2.0 JAR files. It ignores any 2.0-specific features in the deployment descriptor, except for local interfaces. (If you try to deploy an EJB 2.0 JAR file that contains local interfaces, CICS issues an error message and the JAR file becomes unresolved.)

The WebSphere COS Naming Directory Server

The COS Naming Directory Server supplied with WebSphere Application Server Version 5 differs from that supplied with WebSphere Application Server Version 4. From Version 5 onwards:

- The default TCP/IP port used by the COS Naming Directory Server is 2809 (rather than 900 in WebSphere Version 4).
- Java objects must be published to a specially-architected location called "`domain/legacyRoot`". (CICS publishes Java objects to a context defined by the JNDIPREFIX option of the CORBASERVER definition, where the JNDI prefix is a relative path.) If you do not specify the `/domain/legacyRoot` path from the root node of the name space, CICS tries to publish Java objects to a JNDI prefix location relative to the root node itself. This works for the COS Naming Directory Server supplied with WebSphere Application Server Version 4, but fails with that supplied with later versions of WebSphere Application Server.

  The recommended way to specify the location of your name server is on the `com.ibm.cics.ejs.nameserver` property in the JVM system properties file. If you use the COS Naming Directory Server supplied with WebSphere Application Server Version 5, you should specify the location like this:

```
com.ibm.cics.ejs.nameserver=iiop://mycsserv.hursley.ibm.com:2809/domain/legacyRoot
```

For more information, see the *CICS System Definition Guide*.

New EJB utility sample programs

CICS supplies three new utility programs for use with enterprise beans. The programs illustrate how to:

1. Publish a connection factory to a JNDI namespace (the `CICSConnectionFactoryPublish` sample). You can use the sample to create a **ConnectionFactory** object suitable for use with the CCI Connector for CICS TS, and to publish it to the JNDI namespace used by the local CICS region. An enterprise bean or Java program, running on CICS, can then perform a JNDI lookup to obtain a reference to the connection factory.
2. Retract a previously-published connection factory from the JNDI namespace (the `CICSConnectionFactoryRetract` sample).
3. Look up a connection factory in the JNDI namespace (the `CCI Connector` sample). This sample also shows you how to use the CCI Connector for CICS TS to call a CICS server program.

The `CICSConnectionFactoryPublish`, `CICSConnectionFactoryRetract`, and `CCI Connector` samples are described in *Java Applications in CICS*.

Chapter 9. The global user exit programming interface

This chapter summarizes changes to the global user exit programming interface. It covers the following topics:

- "Obsolete global user-exit points"
- "Changes to the standard parameter list" on page 44
- "Changes to global user exit points" on page 45
- "New global user exit points" on page 46

See the *CICS Customization Guide* for information on the changed global user exit points.

Reassembling global user exit programs

The CICS global user exit programming interface is product-sensitive, and is dependent on the implementation of CICS facilities. All global user exit programs must be reassembled against the CICS TS Version 3 Release 1 libraries. You will have to modify some of them for changes to parameters, before they are reassembled.

Note the changes summarized in this chapter and described in detail in the other CICS manuals, and modify your global user exit programs accordingly.

When you have completed your program changes, reassemble *all* global user exit programs.

Obsolete global user-exit points

Table 18 shows the global user-exit points that are obsolete.

Table 18. Obsolete global user-exit points

Exit name	Module or domain	Reason
XTCTIN	Terminal control program	This exit was invoked on TCAM input events. It is no longer called because CICS Transaction Server for z/OS, Version 3 Release 1 does not support the TCAM/ACB interface, and it only supports the TCAM/DCB interface indirectly.
XTCTOUT	Terminal control program	This exit was invoked on TCAM output events. It is no longer called because CICS Transaction Server for z/OS, Version 3 Release 1 does not support the TCAM/ACB interface, and it only supports the TCAM/DCB interface indirectly.

Changes to the standard parameter list

There are changes to the DFHUEPAR standard parameter list, as follows:
- "Changes to the TCB two-character task indicators."
- "The complete list of TCB two-character task indicators."

Changes to the TCB two-character task indicators

The DFHUEPAR standard parameter list of TCB two-character codes and symbolic values addressed by the global user exit task indicator field, UEPGIND, is extended. TCB modes are represented in DFHUEPAR as both a two-character code and a symbolic value.

Table 19. TCB indicators removed from DFHUEPAR. Description

Symbolic value	2-byte code	Change	Description
UEPTH8	H8	Deletion	A Java hotpooling mode TCB
UEPTL9	L9	Addition	An L9 open TCB, used for OPENAPI programs that are in user key
UEPTX8	X8	Addition	An X8 open TCB, used for C and C++ programs, compiled with the XPLINK option, that are in CICS key
UEPTX9	X9	Addition	An X9 open TCB, used for C and C++ programs, compiled with the XPLINK option, that are in user key

The complete list of TCB two-character task indicators

Table 20. TCB indicators in DFHUEPAR

Symbolic value	2-byte code	Description
UEPTQR	QR	The quasi-reentrant mode TCB
UEPTRO	RO	The resource-owning mode TCB
UEPTCO	CO	The concurrent mode TCB
UEPTSZ	SZ	The FEPI mode TCB
UEPTRP	RP	The ONC/RPC mode TCB
UEPTFO	FO	The file-owning mode TCB
UEPTSL	SL	The sockets listener mode TCB
UEPTSO	SO	The sockets mode TCB
UEPTS8	S8	The secure sockets layer mode TCB
UEPTD2	D2	The CICS–DB2 housekeeping mode TCB
UEPTL8	L8	An L8 open TCB, used for OPENAPI TRUEs, or OPENAPI programs that are in CICS key
UEPTL9	L9	An L9 open TCB, used for OPENAPI programs that are in user key
UEPTJ8	J8	A J8 open TCB, used for JVMs that are in CICS key
UEPTJ9	J9	A J9 open TCB, used for JVMs that are in user key
UEPTJM	JM	A JM TCB, used for the master JVM that initializes the shared class cache

Table 20. TCB indicators in DFHUEPAR (continued)

Symbolic value	2-byte code	Description
UEPTX8	X8	An X8 open TCB, used for C and C++ programs, compiled with the XPLINK option, that are in CICS key
UEPTX9	X9	An X9 open TCB, used for C and C++ programs, compiled with the XPLINK option, that are in user key

Changes to global user exit points

Table 21 shows those global user exit points that are changed in some way.

Table 21. Changed global user exit points

Exit name	Description of changes
XPCFTCH	When the exit XPCFTCH is invoked from a C or C++ programs that was compiled with the XPLINK option, a flag is set indicating that any modified entry point address, if specified by the exit, will be ignored.
XPCTA	When the exit XPCTA is invoked from a C or C++ programs that was compiled with the XPLINK option, a flag is set indicating that a resume address, if specified by the exit, will be ignored.

Changes because of channels

Global user exit programs cannot access containers created by application programs. They can, however, create their own channels and pass them to programs which they call.

Minor changes to the following exits are described in the *CICS Customization Guide*:
- XFCAREQ
- XFCAREQC
- XFCREQ
- XFCREQC
- XICEREQ
- XICEREQC
- XNQEREQ
- XNQEREQC
- XPCREQ
- XPCREQC
- XTDEREQ
- XTDEREQC
- XTSEREQ
- XTSEREQC

New global user exit points

Table 22 shows the new global user-exit points.

Table 22. New global user-exit points

Module or domain	Exit name	When invoked
EXEC interface program	XPCERES	XPCERES is invoked by the EXEC interface program, on the target region, before CICS processes either of the following kinds of dynamically-routed link request: • A distributed program link (DPL) call • A Link3270 bridge request It is described in the *CICS Customization Guide*.
Interval control program	XICERES	XICERES is invoked by the interval control program, before CICS processes a non-terminal-related EXEC CICS START request that has been dynamically routed to this region. It is described in the *CICS Customization Guide*.

Chapter 10. User-replaceable programs

This chapter summarizes the changes that affect CICS user-replaceable programs.

It covers the following topics:
- "Obsolete user-replaceable programs"
- "Changes to user-replaceable programs"
- "New user-replaceable programs" on page 48

Reassembling user-replaceable programs

There are some changes in this release to the user-replaceable program interface. You should check whether these changes affect your own customized programs, and make any necessary changes. For example, there might be changes to the parameters passed to the programs, or there might be new actions that the programs need to take. To help you to identify any code changes that are required, compare your customized programs with the sample code in the user-replaceable sample programs provided with this CICS release.

You must reassemble all user-replaceable programs, whether or not you make any changes to them. This includes programs such as your terminal autoinstall control program.

See the *CICS Customization Guide* for programming information about user-replaceable programs.

Obsolete user-replaceable programs

Table 23 shows the obsolete user-replaceable programs.

Table 23. Obsolete user-replaceable programs

Module	Explanation
DFHSJJ8O	DFHSJJ8O is replaced by DFHJVMRO, see "DFHJVMRO (formerly DFHSJJ8O)" on page 48

Changes to user-replaceable programs

There are changes affecting the following user-replaceable programs:
- "The distributed routing program"
- "The IIOP security program, DFHXOPUS" on page 48
- "DFHCNV" on page 48

The distributed routing program

A new abend code, AIID, may be returned in the DYRABCDE field of the communications area passed to the distributed routing program, DFHDSRP, if the routing program is invoked for transaction abend. (Note that this invocation occurs on the *target* region, and then only if the routing program has specified, on a previous call on the routing region, that it should be re-invoked, on the target region, for transaction initiation, termination, and abend.)

AIID indicates to the distributed routing program that a routed method request for an enterprise bean or CORBA stateless object has failed, on the target region, because the target CorbaServer on the target region is disabled.

After a routing error, the routing program has better information on which to base a decision whether or not to remove an AOR from its routing set. For example:

- If a routing error occurs because the target AOR, or the connection to it, is unavailable, the routing program may decide to remove the AOR, temporarily, from its routing set, until sufficient time has elapsed for the AOR or connection to be repaired. In the meantime, it directs all requests to other AORs.
- If a routing error occurs because the target CorbaServer on the target AOR is disabled (and the AOR supports multiple CorbaServers), the routing program may decide *not* to remove the AOR from its routing set. It redirects the failed request to a different AOR, but recognizes that the first AOR remains a valid target for requests for objects supported by its other CorbaServers.

  For detailed information about how to code a distributed routing program to deal with a disabled CorbaServer, see the *CICS Customization Guide*.

The IIOP security program, DFHXOPUS

The sample IIOP security program, DFHXOPUS, has been changed to handle GIOP 1.2 request headers. The actions taken by the sample program have not changed. However, versions of GIOP from 1.2 onwards do not support the IIOP Principal field in request headers. In some circumstances (where SSL(YES) or SSL(NO) is specified on the TCPIPSERVICE definition and there is no RACF user ID associated with the SSL client certificate), DFHXOPUS uses the first eight characters of the IIOP Principal, if there is one, to derive the user ID that it returns. For GIOP 1.2 requests, the IIOP Principal is not present and so DFHXOPUS will never return it as the user ID.

DFHCNV

There is a new DFHCNV macro parameter operand.

The new operand SYSDEF has been added to the TYPE=INITIAL and TYPE=ENTRY macro parameters CLINTCP and SRVERCP. These macros define the user-replaceable data conversion table DFHCNV. The DFHCNV TYPE=INITIAL macro defines the beginning of the conversion table. It gives a list of valid code pages. The DFHCNV TYPE=ENTRY macro specifies a name and type to uniquely identify a data resource. There must be one for each resource for which conversion is required.

For information about the format of the changed parameters, see the *CICS Family: Communicating from CICS on System/390®* Guide.

New user-replaceable programs

The following user-replaceable programs are added:

- "DFHJVMRO (formerly DFHSJJ8O)"
- "DFHAPXPO" on page 49

DFHJVMRO (formerly DFHSJJ8O)

DFHJVMRO defines Language Environment run-time options for the Language Environment enclave in which the JVM runs. It replaces the user-replaceable program DFHSJJ8O, which was used in CICS Transaction Server for z/OS, Version

2 Release 1 and CICS Transaction Server for z/OS, Version 2 Release 2 to construct the Language Environment enclave for the JVM. DFHJVMRO performs the same functions as DFHSJJ8O.

DFHJVMRO defines storage allocation parameters for heap and stack and a number of other options. For CICS, the storage settings that are supplied in DFHJVMRO are more efficient than the default Language Environment storage settings, You can also modify these settings to match more closely with the storage usage of your JVMs.The source for DFHJVMRO is supplied in the CICSTS31.CICS.SDFHSAMP library.

For information on how you can tailor this user-replaceable program to your own requirements, see the *CICS Customization Guide.*

DFHAPXPO

DFHAPXPO is loaded during the PIPI preinitialization phase of each Language Environment enclave where C or C++ programs compiled with the XPLINK option are to be run. It allows you to alter the default Language Environment run-time options. See the *z/OS Version 1.4 Language Environment Programming Guide*, SC22-7561, for details of the Language Environment options that can be reset. The program must be written in Assembler language.

The source for DFHAPXPO is supplied in the CICSTS31.CICS.SDFHSAMP library.

For information on how you can tailor this user-replaceable program to your own requirements, see the *CICS Customization Guide.*

Chapter 11. Monitoring and statistics

This chapter deals with aspects of migration relating to the changes to monitoring and statistics. It covers the following topics:
- "Changes to monitoring and statistics data in SMF 110 records"
- "Changes to statistics records"
- "Calculating CICS and DB2 processor times for DB2 Version 6 or later" on page 53

Changes to monitoring and statistics data in SMF 110 records

There are changes to CICS monitoring and statistics data that could affect user- and vendor-written utilities that analyze and print CICS SMF monitoring and statistics records.

Check your utility programs that process CICS SMF records to ensure that they can process the SMF 110 records correctly. If you have utility programs provided by independent software vendors, you should ensure that these also are able to handle the SMF 110 records correctly.

You can identify SMF 110 records from different releases by using the record-version field in the SMF product section.

Increase in performance class data record length

Some performance data fields are added to performance class data records. The result of all these additions is that record length of performance class data records has increased significantly, with the maximum record length now up to 1836 bytes per record.

To avoid flooding your SMF data sets with unwanted data, and consequently filling them too quickly, you can reduce the amount of data written to SMF by using a monitoring control table (MCT) to selectively include or exclude specified fields. See the *CICS Resource Definition Guide* for information about coding an MCT to control data recording using the DFHMCT TYPE=RECORD macro.

For details of all new and changed monitoring fields, see the *CICS Performance Guide*.

Changes to statistics records

There are changes to CICS statistics records. These are usually because of new domains, or they are a result of enhancements to CICS. As a result, a number of statistics DSECTs have new or changed fields. The changed DSECTs are:

Copybook   **For functional area**

DFHDSGDS   Dispatcher global statistics.

DFHEJBDS   Enterprise beans

DFHMNTDS   Transaction performance monitoring resource statistics.

DFHPGRDS (new DSECT)
          JVM programs

DFHPIPDS (new DSECT)
          Pipeline resource statistics

DFHPIWDS (new DSECT)
Webservice resource statistics

DFHSJGDS   The JVM pool

DFHSJRDS (new DSECT)
JVM profiles

DFHSMSDS   Storage above 16MB

DFHSORDS   TCP/IP service resource statistics.

DFHWBRDS   Urimap resource statistics

DFHWBSDS   Urimap global statistics

Existing application programs are unaffected by the changes if they use the old versions of the following changed DSECTS:
DFHMNGDS
DFHSMSDS
DFHSORDS

This is because the new fields are added to the end and do not affect the offsets of the unchanged fields. (Not all of these DSECTS existed at all earlier releases of CICS, but if you were using one or more of them, your application will simply not see the new fields.)

The changes to DFHDSGDS DFHMNTDS [1]and DFHSJGDS are such that the old DSECTs are not compatible with the new DSECT, and application programs using these DSECTs must be recompiled. (Again, a reminder that not all of these DSECTS existed at all earlier releases of CICS, but if you were using one or more of them, this statement applies to you.)

Collecting statistics for Java programs that run in a JVM

Statistics for Java programs that run in a JVM are collected separately from statistics for other programs, because the JVM programs are not loaded by CICS. CICS does not collect statistics for JVM programs when a COLLECT or PERFORM STATISTICS PROGRAM command is issued; to see them, you need to use the COLLECT or PERFORM STATISTICS JVMPROGRAM command instead.

However, when you browse program names using the INQUIRE PROGRAM command, JVM programs *are* found. If you have an application that collects statistics for programs by browsing with the INQUIRE PROGRAM command, and then issuing the COLLECT STATISTICS PROGRAM command for the program names that it finds, the application will now receive a "not found" response when it attempts to collect statistics for any JVM programs. (Previously, zeros would have been returned as statistics values for these programs.)

To avoid receiving this response, make the application check the RUNTIME value for each program name that it finds. If the RUNTIME value is JVM, the application should not issue the COLLECT STATISTICS PROGRAM command for that program name. If you want to see the statistics for programs with a RUNTIME value of JVM, you can make the application issue the COLLECT STATISTICS JVMPROGRAM command for those programs. Note that the statistics information that is collected for JVM programs is not the same as that collected for other programs.

---

1. Data is not written to SMF by DFHMNTDS. It is only relevant when used through the COLLECT STATISTICS interface.

New and revised values in DFHSTIDS (statistics record identifiers)

The revised list of the statistics record identifiers, as described in the common statistics record copybook, DFHSTIDS, is shown in the *CICS Customization Guide*.

The values in that list, which are new or revised for CICS TS Version 3 Release 1, are shown in Figure 2.

```
STID        STID
Symbolic    Value  Copybook      Type of record
name

STIWBG       101   DFHWBEDS      URIMAPs (Global) ID
STIWBR       104   DFHWBRDS      URIMAPs (Resource) ID
STIPIR       105   DFHPIRDS      PIPELINE (Resource) ID
STIPIW       106   DFHPIWDS      WEBSERVICE (Resource) ID
```

Figure 2. New Statistics data record copybooks related to STID name and value

For details of all the statistics, and all the supporting copybooks, see the *CICS Performance Guide*.

Calculating CICS and DB2 processor times for DB2 Version 6 or later

When CICS is connected to DB2® Version 6 or later, and is exploiting the open transaction environment, the CICS DB2 attachment facility uses CICS-managed open TCBs rather than CICS DB2 subtask TCBs. This means the CICS monitoring facility can measure activity that was previously only reported in the DB2 accounting record (the SMF type 101 record). For example, CICS can now measure the processor time consumed on the DB2 thread and the processor time consumed in DB2 (the CLASS 1 and CLASS 2 CPU time). When CICS is using L8 open TCBs, the CPU time reported for these TCBs by the CICS monitoring facility includes the DB2 CLASS 1 processor time.

When CICS is connected to DB2 Version 6 or later, **do not** add together the processor time from the CICS records (SMF type 110 records) and the DB2 accounting records (SMF type 101 records) when calculating the total processor time for a single transaction, because the DB2 processor time would then be included twice. The total processor time for a single transaction is recorded in the USRCPUT field in the CICS records (performance class data field 008 from group DFHTASK). This field includes all processor time used by the transaction when it was executing on any TCB managed by the CICS dispatcher. "Important changes to accounting for processor time in the open transaction environment" on page 108 has more detailed information about this.

Part 2. Migration planning considerations

This part of the book deals with migration planning for some specific functional areas where there may be a need for special considerations.

These are:

Chapter 12. Redefining and initializing the local and global catalogs

When you migrate to a new CICS release, you need to delete, redefine and initialize the CICS local catalog (LCD) and global catalog (GCD).

1. Delete your existing local catalog and global catalog.

2. Define and initialize new local and global catalogs, following the instructions in the *CICS System Definition Guide* for setting up catalog data sets. When you initialize the catalogs, make sure that you use the CICS TS for z/OS, Version 3.1 versions of the DFHRMUTL and DFHCCUTL utility programs and the sample jobs.

3. When you start the CICS region for the first time after migration, make sure that this is an initial start (with the START=INITIAL parameter).

Chapter 13. Migration planning for application development

This chapter discusses changes to CICS translator support for high-level language language compilers. It covers the following topics:

- Translator support for high-level languages
- Runtime support
- Support for OO COBOL

Translator support for high-level languages

CICS translator support for pre-Language Environment compilers is withdrawn. These are:

- OS/VS COBOL (5740-CB1, 5740-LM1, and 5734-CB4)
- VS COBOL II (5668-958 and 5688-023)
- OS PL/I Version 1 (5734-PL1)
- OS PL/I Version 2 (5668-910 and 5668-909)
- SAA AD/Cycle® C/370™ (5688-216)

CICS now supports only the Language Environment-conforming compilers for application program development. For details of the supported compilers, see the *CICS Transaction Server for z/OS Release Guide*.

The following JCL procedures that are supplied in earlier releases for translating, compiling, and link-editing using the unsupported compilers are also withdrawn:

COBOL The DFHEITVL, DFHEXTVL, DFHEBTVL, DFHEITCL, and DFHEXTCL procedures.

PL/I The DFHEITPL, DFHEXTPL, and DFHEBTPL procedures.

C The DFHEITDL and DFHEXTDL procedures.

CICS now supplies the following procedures only, for use with Language Environment-conforming compilers:

Language	CICS-online	EXCI	Integrated translator
C	DFHYITDL	DFHYXTDL	DFHZITDL (without XPLINK) DFHZITFL (with XPLINK)
C++	DFHYITEL	DFHYXTEL	DFHZITEL (without XPLINK) DFHZITGL (with XPLINK)
COBOL	DFHYITVL	DFHYXTVL	DFHZITCL
PL/I	DFHYITPL	DFHYXTPL	DFHZITPL

The following CICS translator options, which all relate to the unsupported compilers, are obsolete:

- ANSI85
- LANGLVL
- FE

The CICS translators ignore these and issue a return code 4 warning message.

Runtime support

Although application program development support for old, obsolete compilers is withdrawn, CICS usually continues to provide runtime support for your existing application programs that were developed using these old compilers. However, to apply maintenance to these application programs, you should use one of the supported Language Environment-conforming compilers.

You are recommended to use the Language Environment run-time libraries for all CICS application programs, including those that were generated using the unsupported compilers. See the *CICS System Definition Guide* for examples of the DD statements you need in your JCL for the Language Environment libraries.

Support for OS/VS COBOL

In CICS TS 3.1, runtime support for OS/VS COBOL programs is withdrawn. If you attempt to use an OS/VS COBOL program, the abend code ALIK is issued, and CICS abnormally terminates the task and disables the program.

OS/VS COBOL programs must be upgraded to Language Environment-conforming COBOL, and recompiled against a level of COBOL compiler supported by CICS. Enterprise COBOL for z/OS Version 3 is the recommended compiler.

See the *CICS Application Programming Guide* for notes on converting OS/VS COBOL programs to Enterprise COBOL. The *Enterprise COBOL for z/OS: Compiler and Run-Time Migration Guide* has more detailed information about language differences, and describes facilities to help with conversion.

Support for OO COBOL

In CICS TS 3.1, COBOL class definitions and methods (object-oriented COBOL) cannot be used. This restriction includes both Java classes and COBOL classes.

Modules compiled in earlier CICS releases with the OOCOBOL translator option cannot execute in CICS TS 3.1. The OOCOBOL translator option was used for the older SOM-based (System Object Manager-based) OO COBOL, and runtime support for this form of OO COBOL was withdrawn in z/OS V1.2. The newer Java-based OO COBOL, which is used in Enterprise COBOL, is not supported by the CICS translator.

If you have existing SOM-based OO COBOL programs, you should rewrite your OO COBOL into procedural (non-OO) COBOL in order to use the Enterprise COBOL compiler. Note that the newer Java-based OO COBOL is not compatible with SOM-based OO COBOL programs, and is not intended as a migration path for SOM-based OO COBOL programs.

Chapter 14. Migration planning for BTAM and TCAM networks

This chapter covers migration for BTAM and TCAM networks. It covers the following topics:
- "BTAM networks"
- "TCAM networks"

BTAM networks

If you have a network of BTAM terminals connected to a back-level CICS terminal-owning region (TOR), you will not be able (as you were in previous CICS releases) to route transactions from them to a CICS TS for z/OS, Version 3.1 application-owning region (AOR). You must either upgrade your terminals or route to a previous version of CICS.

TCAM networks

the ACB interface of TCAM

If you have a network of terminals connected by the ACB interface of TCAM to a back-level CICS TOR, you will not be able (as you were in previous CICS releases) to route transactions from them to a CICS TS for z/OS, Version 3.1 AOR. You must migrate your connections to use TCAM/DCB or (preferably) ACF/VTAM, or route to a previous version of CICS. (All terminals that support TCAM/ACB also support ACF/VTAM.)

the DCB interface of TCAM

If you have a network of terminals connected by the DCB interface of TCAM to, for example, a CICS TS 2.3 TOR, you will not be able to migrate the TOR to CICS TS for z/OS, Version 3.1. To do so, you must migrate your connections to use ACF/VTAM.

If you have a network of terminals connected by the DCB interface of TCAM to a back-level CICS TOR, you will (as in previous CICS releases) be able to route transactions from them to a CICS TS for z/OS, Version 3.1 AOR. However, you are recommended to migrate your connections to use ACF/VTAM.

Chapter 15. Migration planning for connector applications

Support for the CICS Connector for CICS TS, introduced in CICS TS for z/OS, Version 2.1, is withdrawn.

A CICS connector is a software component that allows a Java client application to invoke a CICS application. CICS TS for z/OS, Version 2.3 introduced a new CICS connector, the CCI Connector for CICS TS, that performs a similar role to the CICS Connector for CICS TS—that is, it enables a Java program or enterprise bean running on CICS Transaction Server for z/OS to link to a CICS server program. However, whereas the old CICS Connector for CICS TS implemented the IBM-proprietary CCF interface, the new CCI Connector for CICS TS implements the industry-standard Common Client Interface (CCI) defined by the J2EE Connector Architecture Specification, Version 1.0.

Since CICS TS for z/OS, Version 2.3 it has been recommended that:
* When writing new connector applications, you use the CCI Connector for CICS TS rather than the CICS Connector for CICS TS
* You migrate any existing applications that use the CICS Connector for CICS TS to use the CCI Connector for CICS TS instead

Because runtime support for the CICS Connector for CICS TS is withdrawn in CICS TS for z/OS, Version 3.1, these recommendations have now become mandatory. For advice on using the CCI Connector for CICS TS in new applications, and on migrating existing applications that use the CICS Connector for CICS TS to use the CCI Connector for CICS TS instead, see the *Java Applications in CICS*.

Chapter 16. Migration planning for users of Debug Tool for z/OS

Debug Tool Version 5 Release 1, with PTF UQ88297 for APAR PQ94401, supports CICS TS for z/OS, Version 3.1. Earlier versions of the tool do not support this CICS release.

Debug Tool communicates with a remote debugger by using TCP/IP Sockets. With CICS TS for z/OS, Version 2.3 and later CICS releases, Debug Tool uses the CICS Sockets domain. With earlier CICS releases, Debug Tool used the TCP/IP Socket Interface for CICS feature of TCP/IP for MVS. If you use Debug Tool in remote debug mode:

- Ensure that TCP/IP support is enabled for the CICS region (see the *CICS Transaction Server for z/OS Installation Guide*).
- Now that the TCP/IP Socket Interface feature is not required by Debug Tool, determine whether you use it for any other purpose. If you do not, then you can remove the TCP/IP Socket Interface feature from your system.

Chapter 17. Migration planning for enhanced inter-program data transfer: channels as modern-day COMMAREAs

This topic describes:
- "Migrating from COMMAREAs to channels"
- "Coexistence with other CICS products" on page 69

Migrating from COMMAREAs to channels

This topic describes:
- "Migration of existing functions"
- "Migration to the new function"

Migration of existing functions

- CICS application programs that use traditional communications areas (COMMAREAS) to exchange data continue to work as before.
- If you employ a user-written dynamic or distributed routing program for workload management, rather than CICSPlex SM, you must modify your program to handle the new values that it may be passed in the DYRLEVEL, DYRTYPE, and DYRVER fields of the DFHDYPDS communications area.

Migration to the new function

This section describes how you can migrate several types of existing application to use channels and containers rather than communication areas (COMMAREAs).

It's possible to replace a COMMAREA by a channel with a single container. While this may seem the simplest way to move from COMMAREAs to channels and containers, it's not good practice to do this. Because you're taking the time to change your application programs to exploit this new function, you should implement the "best practices" for channels and containers. Channels have several advantages over COMMAREAs and it pays to design your channels to make the most of these improvements.

Migrating LINK commands that pass COMMAREAs

To migrate two programs which use a COMMAREA on a LINK command to exchange a structure, change the instructions shown in Table 24.

Table 24. Migrating LINK commands that pass COMMAREAs

Program	Before	After
| PROG1 | ```
EXEC CICS LINK
        PROGRAM(PROG2)
        COMMAREA(structure)
``` | ```
EXEC CICS PUT CONTAINER(structure-name)
        CHANNEL(channel-name)
        FROM(structure)
EXEC CICS LINK PROGRAM(PROG2)
        CHANNEL(channel-name)

...
EXEC CICS GET CONTAINER(structure-name)
        CHANNEL(channel-name)
        INTO(structure)
``` |

Table 24. Migrating LINK commands that pass COMMAREAs (continued)

Program	Before	After

PROG2	`EXEC CICS ADDRESS`
	`   COMMAREA(structure-ptr)`
	`...`
	`RETURN`
		`EXEC CICS GET CONTAINER(structure-name)`
		`        INTO(structure)`
		`...`
		`EXEC CICS PUT CONTAINER(structure-name)`
		`        FROM(structure)`
		`RETURN`

Migrating XCTL commands that pass COMMAREAs

To migrate two programs which use a COMMAREA on an XCTL command to pass a structure, change the instructions shown in Table 25.

Table 25. Migrating XCTL commands that pass COMMAREAs

Program	Before	After

PROG1	`EXEC CICS XCTL`
	`      PROGRAM(PROG2)`
	`      COMMAREA(structure)`
		`EXEC CICS PUT CONTAINER(structure-name)`
		`        CHANNEL(channel-name)`
		`        FROM(structure)`
		`EXEC CICS XCTL PROGRAM(PROG2)`
		`        CHANNEL(channel-name)`
		`...`

PROG2	`EXEC CICS ADDRESS`
	`   COMMAREA(structure-ptr)`
	`...`
		`EXEC CICS GET CONTAINER(structure-name)`
		`        INTO(structure)`
		`...`

Migrating pseudoconversational COMMAREAs on RETURN commands

To migrate two programs which use COMMAREAs to exchange a structure as part of a pseudoconversation, change the instructions shown in Table 26.

Table 26. Migrating pseudoconversational COMMAREAs on RETURN commands

Program	Before	After

PROG1	`EXEC CICS RETURN`
	`      TRANSID(PROG2)`
	`      COMMAREA(structure)`
		`EXEC CICS PUT CONTAINER(structure-name)`
		`        CHANNEL(channel-name)`
		`        FROM(structure)`
		`EXEC CICS RETURN TRANSID(TRAN2)`
		`        CHANNEL(channel-name)`

PROG2	`EXEC CICS ADDRESS` `   COMMAREA(structure-ptr)`	`EXEC CICS GET CONTAINER(structure-name)` `        INTO(structure)`

Migrating START data

To migrate two programs which use START data to exchange a structure, change the instructions shown in Table 27.

Table 27. Migrating START data

Program	Before	After

PROG1	`EXEC CICS START`
	`      TRANSID(TRAN2)`
	`      FROM(structure)`
		`EXEC CICS PUT CONTAINER(structure-name)`
		`        CHANNEL(channel-name)`
		`        FROM(structure)`
		`EXEC CICS START TRANSID(TRAN2)`
		`        CHANNEL(channel-name)`

Table 27. Migrating START data (continued)

Program	Before	After
PROG2	`EXEC CICS RETRIEVE` `        INTO(structure)`	`EXEC CICS GET CONTAINER(structure-name)` `              INTO(structure)`

Note that the new version of PROG2 is the same as that in the pseudoconversational example.

Migrating dynamically-routed applications

EXEC CICS LINK and EXEC CICS START commands, which can pass either COMMAREAs or channels, can be dynamically routed.

When a LINK or START command passes a COMMAREA rather than a channel, the routing program can, depending on the type of request, inspect or change the COMMAREA's contents. For LINK requests and transactions started by terminal-related START requests (which are handled by the *dynamic* routing program) but not for non-terminal-related START requests (which are handled by the *distributed* routing program) the routing program is given, in the DYRACMAA field of its communication area, the *address* of the application's COMMAREA, and can inspect and change its contents.

Note: The routing program's communication area is mapped by the DFHDYPDS DSECT.

If you migrate a dynamically-routed EXEC CICS LINK or START command to use a channel rather than a COMMAREA, the routing program is passed, in the DYRCHANL field of DFHDYPDS, the name of the channel. Note that the routing program is given the *name* of the channel, not its address, and so is unable to use the DYRCHANL field to inspect or change the contents of the channel's containers.

To give the routing program the same kind of functionality with channels, an application that uses a channel can create, within the channel, a special container named DFHROUTE. If the application issues a LINK or terminal-related START request (but not a non-terminal-related START request) that is to be dynamically routed, the dynamic routing program is given, in the DYRACMAA field of DFHDYPDS, the address of the DFHROUTE container, and can inspect and change its contents.

If you are migrating a program to pass a channel rather than a COMMAREA, you could use its existing COMMAREA structure to map DFHROUTE.

For introductory information about dynamic and distributed routing, see the *CICS Intercommunication Guide*. For information about writing a dynamic or distributed routing program, see the *CICS Customization Guide*.

Coexistence with other CICS products

A CICS TS 3.1 program can invoke a program on a remote CICS region and pass it a channel. For this to work successfully, the remote region must also be at the CICS TS 3.1 level.

Although pre-CICS TS 3.1 regions do not support channels, you can get them to tolerate channels by applying an APAR. By "tolerate" we mean that, if the back-level CICS region is passed a channel, it will return a meaningful abend code.

If a CICS TS 3.1 application tries to send a channel to a back-level region to which the appropriate APAR has been applied, the 3.1 transaction abends with a meaningful abend code. If a CICS TS 3.1 application tries to send a channel to a back-level region to which the appropriate APAR has *not* been applied, the results are unpredictable.

The following list shows the back-level CICS products that tolerate channels, with the APAR that must be applied in each case:

CICS Transaction Server for z/OS, Version 2 Release 3
> APAR PQ92437

CICS Transaction Server for z/OS, Version 2 Release 2
> APAR PQ92437

CICS Transaction Server for OS/390, Version 1 Release 3
> APAR PQ93048

CICS Transaction Sever for VSE/ESA Release 1.1
> APAR PQ83049

Chapter 18. Migration planning for enterprise beans

This chapter describes the changes you may have to make to your existing applications or configuration because of CICS enhanced support for enterprise beans. It covers the following topics:
- "Migrating without exploiting the new function"
- "Migrating to the new function" on page 72
- "Migrating an EJB server to CICS TS for z/OS, Version 3.1" on page 72
- "CICS connectors" on page 78

Migrating without exploiting the new function

You must:
- Upgrade the CSD to reflect the changes to the CORBASERVER resource definition (see "New resource definition types and new attributes" on page 20).
- Rewrite and recompile any programs that issue EXEC CICS INQUIRE CORBASERVER STATE commands. The STATE option of this command must be replaced with ENABLESTATUS. The program must handle the new CVDA values returned by ENABLESTATUS, rather than the obsolete values that were returned by STATE. See Table 15 on page 35.
- If you use a user-written distributed routing program, rewrite and recompile it to take account of the new abend code, AIID, that may be returned in the DYRABCDE field when a routed request fails on the target region. See "The distributed routing program" on page 47.
- If you use a user-written IIOP security program, and GIOP 1.2, you must rewrite and recompile it. See "The IIOP security program, DFHXOPUS" on page 48.
- If, in CICS TS 2.2, you used the COS Naming Directory Server supplied with WebSphere Application Server Version 4, and now intend to use that supplied with WebSphere Application Server Version 5, you may need to change the `com.ibm.cics.ejs.nameserver` property in your JVM system properties files. For WebSphere Version 5, the location of the COS Naming Directory Server should be specified like this:

`com.ibm.cics.ejs.nameserver=iiop://mycsserv.hursley.ibm.com:2809/domain/legacyRoot`

Note that, from WebSphere Application Server Version 5 onwards:
  – The default TCP/IP port used by the COS Naming Directory Server is 2809 (rather than 900 in WebSphere Version 4).
  – Java objects must be published to a specially-architected location called «`domain/legacyRoot`». (CICS publishes Java objects to a context defined by the JNDIPREFIX option of the CORBASERVER definition, where the JNDI prefix is a relative path.) If you do not specify the `/domain/legacyRoot` path from the root node of the name space, CICS tries to publish Java objects to a JNDI prefix location relative to the root node itself. This works for the COS Naming Directory Server supplied with WebSphere Application Server Version 4, but fails with that supplied with later versions of WebSphere Application Server.

If the location of the name server is hard-coded into your Java applications, to migrate from Version 4 to Version 5 of the WebSphere COS Naming Directory Server you may need to change your applications' source code.

For more information about specifying the location of the JNDI name server, see the *CICS System Definition Guide*.

Migrating to the new function

There are several ways in which you can take advantage of the new function introduced by CICS enhancements to EJB support:

- If you use a user-written distributed routing program, you can rewrite it to take advantage of the additional information that CICS supplies about route selection errors. For example, you could code your program so that:

  - If a routing error occurs because the target AOR, or the connection to it, is unavailable, the routing program removes the AOR, temporarily, from its routing set, until sufficient time has elapsed for the AOR or connection to be repaired. In the meantime, it directs all requests to other AORs.

  - If a routing error occurs because the target CorbaServer on the target AOR is disabled (and the AOR supports multiple CorbaServers), the routing program does *not* remove the AOR from its routing set. It redirects the failed request to a different AOR, but recognizes that the first AOR remains a valid target for requests for objects supported by its other CorbaServers.

    For detailed information about how to code a distributed routing program to deal with a disabled CorbaServer, see the *CICS Customization Guide*.

- You can change your procedures for updating enterprise beans in a production region. The solutions described in *Java Applications in CICS* take advantage of the new ability to disable a CorbaServer without bringing down CICS.

- You can use the SELECT TYPE parameter of the statistics utility program, DFHSTUP, to cause DFHSTUP to format and print enterprise bean statistics.

- You can use the sample statistics program, DFH0STAT, to produce an enterprise bean statistics report.

Migrating an EJB server to CICS TS for z/OS, Version 3.1

Upgrading a single-region CICS EJB/CORBA server

To migrate a single-region CICS EJB/CORBA server from CICS TS for z/OS, Version 2.2 to CICS TS for z/OS, Version 3.1:

1. Quiesce the workload.

2. Shut down the region.

3. Upgrade the region to CICS TS for z/OS, Version 3.1, following the standard migration procedures described in this book.

4. Review "Migration tips" on page 77, which describes some of the changes in EJB/CORBA support between CICS TS for z/OS, Version 2.2 and CICS TS for z/OS, Version 3.1. You should also refer to the "Setting up a single-region EJB server" section in *Java Applications in CICS*, which describes in detail how to set up a single-region EJB server in CICS TS for z/OS, Version 3.1.

5. Restart the region.

6. Republish the Interoperable Object References (IORs) for all the enterprise beans and stateless CORBA objects processed by the server. To do this, issue a PERFORM CORBASERVER(*CorbaServer_name*) PUBLISH command. This command can be issued using EXEC CICS, CEMT, the Resource Manager for enterprise beans, or from a CICSPlex SM EUI or WUI view. Remember to issue a separate command for each CorbaServer in the region.

Upgrading a multi-region CICS EJB/CORBA server

To migrate a multi-region CICS EJB/CORBA server from CICS TS for z/OS, Version 2.2 to CICS TS for z/OS, Version 3.1, you can use any of the following methods:

1. **Shut down the server, upgrade all the regions, and restart the server.**

   This approach is very similar to that described in "Upgrading a single-region CICS EJB/CORBA server" on page 72, except that:

   a. You must upgrade all the regions to CICS TS for z/OS, Version 3.1 before restarting the server. Again, follow the standard migration procedures described in this book.

   b. You should refer to the "Setting up a multi-region EJB server" section in *Java Applications in CICS*, which describes in detail how to set up a multi-region EJB server in CICS TS for z/OS, Version 3.1.

   c. To republish the IORs of enterprise beans and stateless CORBA objects, issue a PERFORM CORBASERVER(*CorbaServer_name*) PUBLISH command on at least one of the AORs. Remember to issue a separate command for each CorbaServer in the AOR.

   The advantage of this approach is its relative simplicity, compared to solutions 2 and 3. Its main disadvantage is that the server's applications are unavailable during the upgrade process.

2. **Create a separate, CICS TS for z/OS, Version 3.1, logical server and gradually migrate applications from the old, CICS TS 2.2, server to the new one.**

   The advantages of this approach are:

   a. Applications are kept available throughout the upgrade process.

   b. You can start with a minimal CICS TS for z/OS, Version 3.1 server, perhaps consisting of just two regions—one listener and one AOR. As more applications are migrated, you can expand the CICS TS for z/OS, Version 3.1 and simultaneously reduce the number of regions in the CICS TS 2.2 server, thereby conserving resources.

   c. It is probably easier to implement than solution 3.

   To set up a new CICS TS for z/OS, Version 3.1 multi-region EJB server, follow all the steps in the "Setting up an EJB server" chapter in *Java Applications in CICS*.

3. **Perform a "rolling upgrade".**

   In a "rolling upgrade", one region at a time is upgraded from the previous to the current level of CICS, while keeping the server operational.

   The advantages of this approach are:

   a. Applications are kept available throughout the upgrade process.

   b. Unlike solution 2, at no stage is it necessary to set up additional CICS regions.

   This method is described in detail in "Performing a "rolling upgrade"" on page 74.

Performing a "rolling upgrade"

> **Important**
>
> The mixed level of operation described in this section, in which different CICS regions in the same logical server are at different levels of CICS, is intended to be used only for rolling upgrades. It should not be used permanently, because it increases the risk of failure in some interoperability scenarios. The normal, recommended, mode of operation is that all the regions in a logical sever should be at the same level of CICS and Java.

This section describes how to perform a "rolling upgrade" of a multi-region CICS EJB/CORBA server from CICS TS for z/OS, Version 2.2 to CICS TS for z/OS, Version 3.1. The process consists of the following steps:
1. Checking that your logical server meets the criteria for a "rolling upgrade". See "Requirement."
2. "Preliminary steps"
3. "Migrating the listener regions" on page 75
4. "Migrating the AORs" on page 75
5. "Tidying up" on page 76

Requirement: Your server must consist of separate listener and application-owning regions. This is because the migration process requires all of the listener regions to be updated before any of the application-owning regions (AORs). If you run composite listener/AORs, which act both as request receivers and request processors, this cannot be done. And if you don't upgrade all the listeners before any of the AORs, your IIOP client applications may receive transient failures during the migration window, depending on the CICS version of the listener region that receives the request.

Preliminary steps:

1. Review "Migration tips" on page 77.

2. Ensure that APAR PQ 79565 is installed in all your CICS TS 2.2 regions. This APAR improves CICS TS 2.2 diagnostics, should CICS TS 3.1 workload arrive at a CICS TS 2.2 region. It also allows a CICS TS 2.2 request processor (AOR) to receive work from a CICS TS 3.1 request receiver (listener).

3. Set the AUTOPUBLISH option on all your CORBASERVER definitions to N0. Setting a CorbaServer to autopublish IORs into the JNDI name spaces could disrupt the migration process.

4. If you use a distributed routing program to balance method requests for enterprise beans and CORBA stateless objects across the AORs of your logical server, customize your routing program to use the DYRLEVEL parameter. DYRLEVEL is a migration aid. It contains the level of CICS required in the target AOR to successfully process the routed request. (Note that this is the **specific**—*not* the minimum—level of CICS required to process the request successfully.) In a mixed-level logical server, when your routing program is invoked for route selection (or route selection error), it can use the value of DYRLEVEL to determine whether to route the request to a CICS TS 2.2 or CICS TS 3.1 AOR.

   For details of how to use DYRLEVEL, and definitive information about writing a distributed routing program, see the *CICS Customization Guide*.

   Install your customized program on *all* the regions (both listeners and AORs) of the EJB server.

If you use CICSPlex SM to workload-balance method requests you can skip
this step. The CICSPlex SM routing program supplied with CICS TS for z/OS,
Version 3.1 checks the DYRLEVEL field and routes requests accordingly.

Migrating the listener regions:

1. Quiesce a listener region and bring it down.
2. Upgrade this single listener region to CICS TS for z/OS, Version 3.1, following
   the standard migration procedures described in this book.

> **Important**
>
> a. When you upgrade the CSD from CICS TS 2.2 to CICS TS 3.1 level, if
>   it is shared by any CICS TS 2.2 regions other than that being
>   upgraded, include the DFHCOMPA resource group (supplied with CICS
>   TS 3.1) in the startup group list of these regions. DFHCOMPA is a
>   compatibility group that provides a definition of DFJIIRP, the default
>   request processor program, that can be used by a CICS TS 2.2 region
>   when sharing a CICS TS 3.1 CSD.
>
>   This step is necessary because, in CICS TS 3.1, the JVM profile used
>   by DFJIIRP is DFHJVMCD. In CICS TS 2.2, it is DFHJVMPR.
>
> b. At this stage, don't enable any new, CICS TS 3.1-specific, options on
>   resource definitions, because they won't be understood by the CICS
>   TS 2.2 AORs. Use of these new features must wait until the whole
>   logical server—both listener regions and AORs—has been upgraded.

For definitive information about setting up a listener region in CICS TS 3.1, refer
to *Java Applications in CICS*.

3. Bring the listener back up. This region is now at the newer version of CICS but
   may continue to participate as part of the back-level logical server.
4. Repeat steps 1 through 3 for all of the listener regions in the logical server.

Migrating the AORs:

1. Quiesce an AOR and bring it down.
2. Update this single AOR to CICS TS for z/OS, Version 3.1, following the
   standard migration procedures described in this book. Part of this will involve
   updating the JVM profile used by the CorbaServers. Note the changes to JVM
   profiles and property files described in "Migration tips" on page 77.

> **Important**
>
> a. When you upgrade the CSD from CICS TS 2.2 to CICS TS 3.1 level, if
>   it is shared by any CICS TS 2.2 regions other than that being
>   upgraded, include the DFHCOMPA resource group (supplied with CICS
>   TS 3.1) in the startup group list of these regions.
>
> b. At this stage, don't enable any new, CICS TS 3.1-specific, options on
>   resource definitions.

---

3. Bring the AOR back up again.
4. Ensure that all TCPIPSERVICEs are open both in this AOR and in the listener
   regions.
5. Use the CEMT PERFORM DJAR PUBLISH command to re-publish the IORs of
   one or more enterprise beans in CICS TS 3.1 format. For each CorbaServer,

select one or more deployed JAR files to re-publish. When choosing deployed JAR files to re-publish, bear the following in mind:

- Try to pick DJARs whose entire work load can be processed by a single region.
- Wherever possible, all the beans used by an application should be migrated at the same time. For example, if bean A is known to call bean B the two beans should be migrated together. If this is not possible, bean A should be migrated first.

  This is particularly important if the beans are installed in the same CorbaServer but in different AORs that are at different levels of CICS. This is because a CICS TS 2.2 region cannot do a JNDI look up of an object in a CICS TS 3.1 region if both objects are in the same CorbaServer. For example, bean A in CorbaServer EJB1 in a CICS TS 2.2 AOR cannot look up bean B in CorbaServer EJB1 in a CICS TS 3.1 AOR.

  Note: If A and B are installed in different CorbaServers, or in AORs that are at the same level of CICS, they can be migrated separately.

Re-publish the selected DJARs to the JNDI name space, in the same location as that used by the CICS TS 2.2 AORs.

At this point :

- This AOR is ready to accept workload.
- The logical server contains a pool of CICS TS 2.2 AORs and a pool (currently containing only one region) of CICS TS 3.1 AORs.
- Any clients that look up the IOR of a re-published bean in the name space get the new IOR in CICS TS 3.1 format. Your customized routing program or CICSPlex SM directs such requests to the CICS TS 3.1 AOR.
- Any clients that have a stale, cached, IOR for a bean that's been re-published are still able to use the bean. Your customized routing program or CICSPlex SM directs such CICS TS 2.2-format requests to one of the CICS TS 2.2 AORs.

  Note: Many application servers cache the results of JNDI lookups locally to increase performance, so you may find that these caches have to be purged before the new IORs are used. Over a period of time, requests for re-published enterprise beans should move gradually from the pool of CICS TS 2.2 AORs to the pool of CICS TS 3.1 AORs.

6. Repeat steps 1 through 5 for all of the AORs in the logical server. As each AOR is upgraded:
   - Re-publish a different set of enterprise beans, so that gradually more and more beans are supported by the pool of CICS TS 3.1 regions.
   - It becomes less important, when selecting deployed JAR files to re-publish, to choose those whose entire work load can be processed by a single region—because there are more AORs in the CICS TS 3.1 pool.

   Eventually, all the AORs will be running CICS TS 3.1 and processing 100% of the workload.

Tidying up:

1. If required, reset the AUTOPUBLISH option on your CORBASERVER definitions to `YES`.
2. Enable any CICS TS 3.1-specific resource definition options that you want to use.

Migration tips

This section briefly lists some of the ways in which EJB and Java support has changed between CICS TS for z/OS, Version 2.2 and CICS TS for z/OS, Version 3.1. All these changes are described in detail in *Java Applications in CICS*. They are listed here, together with some general tips, as a reminder of things to be aware of when migrating an EJB server from CICS TS 2.2 to CICS TS 3.1.

1. In CICS TS 2.2, JVM profiles were stored in a PDS member. In CICS TS 3.1, they are stored in the HFS directory pointed to by the JVMPROFILEDIR system initialization parameter.

2. The default JVM profile used by CorbaServers in CICS TS 2.2 was DFHJVMPR. In CICS TS 3.1 it is DFHJVMCD.

3. The default JVM properties file used by CorbaServers in CICS TS 2.2 was `dfjjvmpr.props`. In CICS TS 3.1 it is `dfjjvmcd.props`.

4. Don't enable any new, CICS TS 3.1-specific, attributes on resource definitions during a "rolling upgrade". For example, on a CORBASERVER definition don't specify the ASSERTED option. (For a complete list of new, CICS TS 3.1-specific, attributes on CORBASERVER, DJAR, REQUESTMODEL, and TCPIPSERVICE resource definitions, see "New resource definition types and new attributes" on page 20.) Use of these new features must wait until the whole logical server—both listener regions and AORs—has been upgraded.

5. From a CICS TS 3.1 AOR, you can re-publish a deployed JAR file that has previously been published from a CICS TS 2.2 AOR without first retracting it. The IORs of the beans are updated to CICS TS 3.1 format. **However, you cannot do the reverse**. From a CICS TS 2.2 AOR, before re-publishing a deployed JAR file that has previously been published from a CICS TS 3.1 AOR you must first retract it; furthermore, because CICS TS 2.2 does not understand the format of CICS TS 3.1 IORs, *you must retract it from a CICS TS 3.1 AOR*.

   Bear this in mind if, for any reason, you need to back out the upgrade of one or more AORs. If you ever need to revert the IORs of enterprise beans that have been published from a CICS TS 3.1 AOR to CICS TS 2.2 level (so that they can be routed to a CICS TS 2.2 AOR once more) you must:

   a. Retract the deployed JAR file from a CICS TS 3.1 AOR

   b. Publish the deployed JAR file from a CICS TS 2.2 AOR

   Trying to re-publish the beans without retracting them first, or trying to retract them from the wrong level of CICS, results in an `InvalidUserKeyException: Bad version number` exception.

Potential problems

1. After the EJB server has been migrated to CICS TS 3.1, some clients may have stale, cached, IORs that point to the old server. This is because some application servers cache the results of JNDI lookups locally to increase performance. You may find that these caches have to be purged before the new IORs are used.

2. CICS TS 3.1 supports GIOP 1.2, whereas CICS TS 2.2 supported only GIOP 1.1. If a GIOP 1.2 message is received in a CICS TS 2.2 region it will be rejected. Under normal conditions this should never happen, because the maximum version of GIOP supported by CICS is stored in the IORs that CICS publishes. If a client knows that a given server only supports GIOP 1.1, it will never attempt to use anything more recent when communicating with that server. This means that CICS TS 3.1 can send GIOP messages to CICS TS 2.2.

   The problem will only occur if the client thinks it is talking to CICS TS 3.1 but its message is routed to a CICS TS 2.2 region. This will only happen if CICS TS

2.2 and CICS TS 3.1 regions are set up as sibling request processors (AORs) in the same logical server. (This is one reason why mixed-level logical servers are not recommended in CICS.) During a "rolling upgrade", the logical server does, of course, contain mixed-level request processors. However, if you follow the steps in "Performing a "rolling upgrade"" on page 74, the problem (of a GIOP 1.2 message being received in a CICS TS 2.2 region) will not occur.

3. CICS TS 3.1 uses a different format of IOR from CICS TS 2.2. If a GIOP 1.1 message intended for CICS TS 3.1 is routed to a CICS TS 2.2 region, the CICS TS 2.2 region will reject the request due to a unknown IOR format being in use. If all the regions in an EJB/CORBA server are at the same level of CICS and Java, this error cannot occur.

During a "rolling upgrade", the logical server does, of course, contain mixed-level regions. However, if you follow the steps in "Performing a "rolling upgrade"" on page 74, this problem will not occur.

CICS connectors

Existing applications that use the CICS Connector for CICS TS supplied with CICS TS for z/OS, Version 2.2—whether they use the connector's CCF Client Interface or the CICS Transaction Gateway API—will continue to work unchanged in CICS TS for z/OS, Version 3.1. However, if you need to update these applications, you should consider re-engineering them to use the CCI Connector for CICS TS.

If you create *new* CICS Java components that connect to CICS server programs, it is recommended that you use the CCI Connector for CICS TS rather than the CICS Connector for CICS TS. The use of the CICS Connector for CICS TS in new applications is deprecated.

If you need to update an *existing* application that uses the CICS Connector for CICS TS, whether you decide to re-engineer the application to use the CCI Connector for CICS TS may depend on factors such as the following:

- Which of the CICS Connector for CICS TS's interfaces the application uses—the CCF Client Interface or the CICS Transaction Gateway API
- If the CCF Client Interface is used, whether it is programmed directly or by means of higher-level constructs (for example, Command beans, Navigators, record beans) created by a tool such as VisualAge for Java Enterprise Access Builder

Table 28 summarizes the upgrade choices for CICS Java components that use either the CICS Connector for CICS TS or the CCI Connector for CICS TS, and states a preferred solution for each case.

Table 28. Suggested upgrade path for CICS Java components that use the CICS CCF or CCI connectors

Connector used by current program	Connector interface used by current program	Status in CICS TS 3.1	Suggested upgrade strategy
CICS Connector for CICS TS	CICS Transaction Gateway API (**ECIRequest**)	Supported	The CICS Transaction Gateway API is not deprecated and can continue to be used. Its drawback is that it's an IBM-proprietary interface.

Table 28. Suggested upgrade path for CICS Java components that use the CICS CCF or CCI connectors (continued)

Connector used by current program	Connector interface used by current program	Status in CICS TS 3.1	Suggested upgrade strategy
CICS Connector for CICS TS	CCF, programmed directly	Supported	CCF is deprecated, replaced by CCI. Re-engineer to use the CCI Connector for CICS TS, which performs better than the CICS Connector for CICS TS and uses an industry-standard interface. Program the connector either directly or by means of a rapid application development (RAD) tool that supports it.
CICS Connector for CICS TS	CCF, programmed with VAJ Enterprise Access Builder or similar	Supported	CCF is deprecated, replaced by CCI. Re-engineer to use the CCI Connector for CICS TS, which performs better than the CICS Connector for CICS TS and uses an industry-standard interface. Program the connector either directly or by means of a RAD tool that supports it. **Note:** It is possible to program the CCI Connector for CICS TS using VAJ Enterprise Access Builder, but this is not recommended because VAJ/EAB is no longer supported.
CCI Connector for CICS TS	CCI, programmed directly	Supported	CCI can be used indefinitely. Programming the CCI directly gives the best performance.
CCI Connector for CICS TS	CCI, programmed with VAJ Enterprise Access Builder or similar	Supported	To continue using VAJ/EAB, changes must be made to the application.

The CCI Connector for CICS TS is described in *Java Applications in CICS*.

Chapter 19. Migration planning for the integrated translator

To install application programs into CICS libraries, you can use one of the
CICS-supplied procedures for use with the integrated translator. Note that the
procedures, DFHZITCL (for COBOL), DFHZITPL (for PL/I), and DFHZITDL,
DFHZITEL, DFHZITFL, and DFHZITGL (for XL C/C++), include the CICS load
library SDFHLOAD in STEPLIB, which is necessary for the compilers to load the
translation tables. Also, if you are using the PL/I procedure or XL C/C++
procedures, the SYSLMOD DD statement in the binder step must refer to a PDSE
(not a PDS as for the older PL/I compilers). See "Translator support for high-level
languages" on page 59 for a list of all the procedures supplied for assembling and
compiling application programs.

To use the CICS-supplied procedures to invoke the integrated translator, add the
required language options to indicate that you want the compiler to invoke the
translator:

- To invoke the PL/I compiler and the integrated translator, specify the PL/I
  compiler preprocessing option (PP); for example,

  ```
  PP(CICS('opt1 opt2 optn ...'))
  ```

- To invoke the COBOL compiler and the integrated translator, specify CICS as a
  COBOL compiler option; for example, in the PARM string, as follows:

  ```
  PARM='NODYNAM,LIB,OBJECT,RENT,MAP,XREF,CICS(''COBOL3,SP'')'
  ```

- To invoke the XL C/C++ compiler and the integrated translator, specify CICS as a
  compiler option; for example, in the PARM string, as follows:

  ```
  PARM=('RENT SOURCE CICS(SP,DEBUG)')"
  ```

Notes:

1. If you specify CICS translator options for the integrated translator in the PARM
   string, you need double apostrophes as shown in this example. If, however, you
   specify the options in your source program, you need single apostrophes (for
   example, you might have `CBL  CICS('COBOL3,SP') APOST` as the CBL statement
   in your source program.

2. The COBOL compiler recognizes only the keyword CICS for defining translator
   options, not the alternative options XOPT or XOPTS as in the case of the
   stand-alone translator supplied with CICS TS.

3. If the CICS TS 1.3 translator is set up to run with a CICS TS 3.1 LPALIB, Error
   Return Code 16 will be produced.

   CICS TS 3.1 uses AMODE(31) settings, while CICS TS 1.3 uses AMODE(24). If
   you intend to run a CICS TS 1.3 translator, you must add to your STEPLIB, an
   SDFHLOAD that contains the modules DFHEITAB and DFHEITBS from CICS
   TS 1.3. This addition must be placed prior to the CICS TS 3.1 SDFHLOAD. This
   will allow the CICS TS 1.3 translator to operate in a CICS TS 3.1 region.

Nested COBOL program considerations

If you are compiling a COBOL application program that contains nested programs,
the rules regarding the use of DFHEIBLK and DFHCOMMAREA, described in the
CICS Application Programming Guide, apply only when you are using the CICS
stand-alone translator.

When you use the integrated translator to compile nested programs, observe the
following rules:

- You no longer need to code explicitly DFHEIBLK and DFHCOMMAREA on the USING phrase when calling a nested program, or on the PROCEDURE DIVISION USING phrase in the nested program, and they must be omitted.
- Because DFHCOMMAREA is not generated in a nested program, it cannot be REDEFINED as with the stand-alone translator. DFHCOMMAREA can be redefined with the global attribute in the outer-most (containing) program, and accessed from nested (contained) programs.

If you are unable to apply these rules for existing programs that you are modifying (including using COPY members to redefine DFHCOMMAREA in nested programs), continue using the stand-alone translator.

Chapter 20. Migration planning for improved Internet security

Migration of existing functions
- The default setting for the ENCRYPTION system initialization parameter has changed to STRONG. If you have no high encryption ciphers installed (security level 3) on z/OS, then you need to downgrade the default setting for the ENCRYPTION system initialization parameter.
- The NORMAL setting that has been used as the default in previous releases, has changed to MEDIUM for this release of CICS. For migration purposes, NORMAL is accepted as an alternative to MEDIUM.
- The SSLTCBS system initialization parameter is now obsolete and has been replaced by MAXSSLTCBS. MAXSSLTCBS controls the maximum number of S8 TCBs that are allowed to run concurrently in the open transaction environment (OTE) TCB pool for SSL.

Migration to the new function
- You can exploit the CIPHERS attribute to better control the encryption negotiation process between CICS and clients.
- You can use the CRLPROFILE and SSLCACHE system initialization parameters to verify certificates in the SSL handshake and improve the performance of the handshake through sharing the SSL cache across CICS regions.

Chapter 21. Migration planning for Java applications

This chapter covers migration for Java applications. It covers the following topics:
- "Migration for Java applications that run in a JVM"
- "Running Java programs in a JVM without exploiting the new JVM functions" on page 87
- "Running Java programs in a JVM using the new JVM functions" on page 89
- "Migration for Java programs that do not run in a JVM (hpj-compiled Java program objects)" on page 91

Migration for Java applications that run in a JVM

CICS Transaction Server for z/OS Version 3 Release 1 supports the JVM provided by the IBM Software Developer Kit for z/OS, Java 2 Technology Edition, Version 1.4.2, which features the persistent reusable JVM technology.

Note: There are two versions of the IBM Software Developer Kit for z/OS, Java 2 Technology Edition, Version 1.4.2, a 31-bit and a 64-bit version. CICS Transaction Server for z/OS Version 3 Release 1 supports only the 31-bit version, which must be at the 1.4.2 level.

CICS Transaction Server for z/OS, Version 2 Release 2 supported the JVM provided by the IBM Developer Kit for OS/390 Java 2 Technology Edition Version 1.3.1s, which also featured the persistent reusable JVM technology. Java programs that ran under CICS Transaction Server for z/OS, Version 2 Release 2 and CICS Transaction Server for z/OS, Version 2 Release 3 can also run under CICS Transaction Server for z/OS Version 3 Release 1.

However, the older type of JVM that was introduced in CICS Transaction Server for OS/390, Version 1 Release 3, which was not reusable, is no longer supported. Any Java programs that ran under CICS Transaction Server for OS/390, Version 1 Release 3 must be migrated to Java 2 to run under the reusable JVM. Application migration issues are discussed at:

```
http://java.sun.com/j2se/1.4/compatibility.html
http://java.sun.com/products/jdk/1.3/compatibility.html#incompatibilities1.3
and
http://java.sun.com/products/jdk/1.2/compatibility.html
```

Support for the JVM provided by the IBM Software Developer Kit for z/OS, Java 2 Technology Edition, Version 1.4.2 completely replaces the JVM support provided in CICS TS 1.3. However, you can modify a JVM to run as a single-use JVM and not attempt serial reuse. A single-use JVM is initialized, is used to run a single Java program, and then is automatically destroyed without attempting a JVM reset. You can modify a JVM to be a single-use JVM by specifying either REUSE=NO, or the older option `Xresettable=NO`, in the JVM profile. The single-use JVM is like the earlier JVM that was supported by CICS in CICS TS 1.3. New Java applications should not be developed in such a way that they can only run in a single-use JVM.

To avoid problems with deprecated APIs, you should develop all new Java programs for CICS Transaction Server for z/OS Version 3 Release 1 using an application development environment that supports Java 2 at the same version of Java as used by CICS. You may run code compiled with an older version of Java in a new runtime, provided that it does not use APIs that have been removed in the newer version of Java. Note also that enterprise beans that support the EJB 1.0 specification need to be migrated to the EJB 1.1 specification level using the

Assembly Toolkit (ATK) or the Application Assembly Tool (AAT), which are supplied with IBM WebSphere Application Server.. Enterprise beans developed using any version of the EJB specification after EJB 1.1 must restrict themselves to the EJB 1.1 APIs.

Invoking multiple Java programs in the same CICS task

The one-JVM-per-stack restriction has been removed. Previously, a stack of programs formed by a succession of EXEC CICS LINK commands, or JCICS program invocations, within the same CICS task, could not contain more than one JVM. (Distributed program link (DPL) requests were not restricted in this way.) As a CICS task could only use one JVM, applications designed in this way could only contain one Java component. Now, multiple JVMs can be allocated to a CICS task, so you can create an application that links together multiple Java programs in the same CICS region. You can also convert your existing complex applications piece by piece, by replacing programs written in other languages with programs written in Java.

Enterprise beans can now link to another Java program within the same CICS task. However, a single CICS task still cannot contain more than one enterprise bean, because CICS treats an execution of an enterprise bean as the start of a new task. You can create an application that includes more than one enterprise bean, but the application will not operate as a single task.

Execution key for Java programs that run in a JVM

The EXECKEY parameter on the PROGRAM resource definition is no longer ignored for Java programs. In CICS Transaction Server for OS/390, Version 1 Release 3 and CICS Transaction Server for z/OS, Version 2 Release 2, CICS made all Java programs execute in CICS key, but they now execute as specified by the EXECKEY parameter.

The default for this parameter is EXECKEY(USER), which means that the program runs in a JVM that executes in user key. (A new type of open TCB, the J9 TCB, is used for these JVMs.) As running applications in user key extends CICS storage protection, it could be beneficial to let most of your Java programs run in a JVM in user key.

You might find that in most cases, the PROGRAM resource definitions for your Java programs are still set to the default of EXECKEY(USER). If you changed the EXECKEY parameter on the PROGRAM resource definitions for any of your Java programs to specify EXECKEY(CICS), you will need to change them back to EXECKEY(USER) if you want them to run in user key. No changes are needed to your JVM profiles, because you can use the same JVM profile to create JVMs in both storage keys.

However, you might need to execute a JVM in CICS key if the Java program that uses the JVM is part of a transaction that specifies TASKDATAKEY(CICS). If this is the case, you need to ensure that the PROGRAM resource definition for the Java program specifies EXECKEY(CICS). Before CICS Transaction Server for z/OS, Version 2 Release 3, this setting would not have mattered, because CICS would have forced the program to execute in CICS key.

For enterprise beans, CIRP (the default transaction for REQUESTMODEL definitions) specifies TASKDATAKEY(USER), and the PROGRAM resource definition for DFJIIRP (the default request processor program) specifies EXECKEY(USER), so by default enterprise beans run in user key.

A JVM can be reused by programs that specify the same execution key on their PROGRAM resource definition. You can use the INQUIRE JVM command to find out the execution key in which a JVM has been invoked. A single CICS task can include Java programs running in CICS key, and Java programs running in user key.

Running Java programs in a JVM without exploiting the new JVM functions

If you are already running Java programs in a JVM, you need to take the following actions to continue running these applications:

- TThe Language Environment run-time library SCEERUN2 is now required to support the IBM JVM, in addition to the Language Environment run-time library SCEERUN. In your CICS startup job, the library SCEERUN2 must be defined in both the STEPLIB and DFHRPL concatenations. Both the libraries, SCEERUN and SCEERUN2, must be APF-authorized.For more information, see in "Installing CICS support for Language Environment" in the *CICS Transaction Server for z/OS Installation Guide*

- The library SDFJAUTH is now required for Java support. SDFJAUTH is the partitioned data set extended (PDSE) version of SDFHAUTH, and it contains some of the components of the SJ domain. A separate library is needed because these components are now built using XPLink (Extra Performance Linkage). As for the SDFHAUTH library, the SDFJAUTH library must be APF-authorized by adding it to the list of APF-authorized libraries in an appropriate member in SYS1.PARMLIB, and a STEPLIB DD statement must be provided for it in your startup job stream. "Authorizing the hlq.SDFHAUTH library" in the *CICS Transaction Server for z/OS Installation Guide* describes this procedure for the SDFHAUTH library, and you can follow the same procedure for the SDFJAUTH library.

- JVM profiles, which contain the JVM initialization options, are now kept as HFS files, rather than as members of a partitioned data set (PDS). The DFHJVM DD card in the CICS startup JCL, which referred to the PDS for the JVM profiles, is no longer required and should be removed. You can use several different JVM profiles in the same CICS region, and each is stored as a separate HFS file. The name of each JVM profile (that is, the name of the HFS file) must still be eight characters or less, so that it can be used in program definition. Use the JVMPROFILE attribute of a PROGRAM resource definition to name the JVM profile which is used to construct the JVM that runs the program.

- Ensure that the JVM profiles you want to use are in the HFS directory that is specified by the new JVMPROFILEDIR system initialization parameter, or use UNIX soft links to link to the JVM profiles from that directory. The default setting for the JVMPROFILEDIR system initialization parameter is `/usr/lpp/cicsts/cicsts31/JVMProfiles`. That is, the supplied setting for JVMPROFILEDIR points to the default directory for the sample JVM profiles. "Enabling CICS to locate the JVM profiles and JVM properties files" in *Java Applications in CICS* tells you how to ensure that CICS can access the JVM profiles in the JVMPROFILEDIR directory, and how to change it if required (for example, if you chose a different name during CICS installation for the directory containing the sample JVM profiles). Note that the JVM profiles DFHJVMPR and DFHJVMCD, and their associated JVM properties files, must always be available to CICS. Their uses are described later in this list. Both these JVM profiles must either be present in the directory that is specified by JVMPROFILEDIR, or linked to by means of UNIX soft links from that directory.

- If you have modified the supplied sample JVM profiles DFHJVMPR and DFHJVMPS, and you want to re-use them, migrate these from PDS members to HFS files, which you can do using the OCOPY TSO command. Note that for CICS Transaction Server for z/OS Version 3 Release 1 , there are changes to the options that are available in JVM profiles and JVM properties files. "Running Java programs in a JVM using the new JVM functions" on page 89 mentions some of these, and the *CICS Transaction Server for z/OS Release Guide* has more information about them. The *CICS System Definition Guide* has the full lists of options that you can specify using JVM profiles and JVM properties files. Before you re-use JVM profiles that you used for CICS TS for z/OS, Version 2.2 or CICS Transaction Server for OS/390, Version 1 Release 3, you should check whether you need to add any of the new options, or to copy any of the changes to existing options that have been made in the supplied sample JVM profiles for CICS Transaction Server for z/OS Version 3 Release 1.
- If you decide not to re-use your existing JVM profiles, you need to set up new JVM profiles that contain settings which are suitable for your installation, and the paths to the application classes and resources that are needed for your applications. Note that the JVM profile DFHJVMPR is used if a Java program is defined as using a JVM but no JVM profile is specified, and it is used for sample programs, so this JVM profile always needs to be configured so that it can be used in your CICS region. "Setting up JVM profiles and JVM properties files" in *Java Applications in CICS* tells you how to do this.
- CICS-defined programs now have their own JVM profile, DFHJVMCD, to make them independent of any changes you make to the default JVM profile DFHJVMPR. DFHJVMCD is used by the default request processor program DFJIIRP, which is used by the CICS-supplied CIRP request processor transaction, and by DFJIIRQ, the CICS-key equivalent of DFJIIRP. DFHJVMCD has an associated JVM properties file, `dfjjvmcd.props`. You need to make changes to DFHJVMCD and `dfjjvmcd.props` to ensure that the settings in them are suitable for your installation (including the configuration for your JNDI nameserver). "Customizing or creating JVM profiles and JVM properties files" in *Java Applications in CICS* tells you how to do this. You also need to add classes to the shareable application class path for the applications that will use the request processor program. For enterprise beans, you need to add any classes, such as classes for utilities, that are required by your enterprise beans but are *not* included in the deployed JAR files for the enterprise beans. For CORBA stateless objects, you need to add the JAR files for the applications and any additional classes that are required. "Enabling applications to use a JVM" in *Java Applications in CICS* tells you how to do this.
- Because JVM profiles are now HFS files, case is important. When you specify the name of a JVM profile (for example, in a PROGRAM resource definition), you must enter it using the same combination of upper and lower case characters that is present in the HFS file name. The CEDA panels accept mixed case input for the JVMPROFILE field irrespective of your terminal's UCTRAN setting. However, this does not apply when values for this field are supplied on the CEDA command line, or when you are using another CICS transaction such as CEMT or CECI. If you need to enter the name of a JVM profile in mixed case when you use CEDA from the command line or when you use another CICS transaction, ensure that the terminal you use is correctly configured, with upper case translation suppressed.
- The user-replaceable program DFHSJJ8O, which was used in CICS Transaction Server for z/OS, Version 2 Release 1 and CICS Transaction Server for z/OS, Version 2 Release 2 to construct the Language Environment enclave for the JVM, has been replaced by the user-replaceable program DFHJVMRO, which performs the same functions. If you had modified the settings specified in

DFHSJJ8O to match more closely with the storage usage of your JVMs, you should repeat this process with DFHJVMRO, bearing in mind that the storage usage of your JVMs might have changed. "Tuning Language Environment enclave storage for JVMs" in the CICS Performance Guide tells you how to carry out this tuning process, and "Using DFHJVMRO to modify the Language Environment enclave for a JVM" in the *CICS Customization Guide* tells you how to modify the settings in DFHJVMRO.

- CICS now uses the z/OS shared library region, which enables address spaces to share dynamic link library (DLL) files. This feature enables your CICS regions to share the DLLs that are needed to create JVMs, rather than each region having to load them individually. The storage that is reserved for the shared library region is allocated in each CICS region when the first JVM is started in the region. The amount of storage that is allocated is controlled by the SHRLIBRGNSIZE parameter in z/OS. The minimum is 16M, and the z/OS default is 64M. You should check the setting for this parameter and, if necessary, tune it. "Tuning the z/OS shared library region" in the *CICS Performance Guide* tells you how to carry out this tuning process.

Running Java programs in a JVM using the new JVM functions

You can take advantage of several new functions introduced by CICS enhancements to support for the IBM JVM:

- You can set up a shared class cache in each of your CICS regions, to enable the JVMs in each CICS region to share commonly-used class files and compiled classes. This means that the JVMs can start up faster, and you can reduce the cost of class loading. Most JVMs can use the shared class cache, but if you do not want certain JVMs to use it, you can set them to run independently (to be standalone JVMs). You can alter the shared class cache in your CICS region while CICS is running, and monitor its status. New system initialization parameters and new CEMT and EXEC CICS commands are provided to enable you to carry out these tasks. You also need to set up JVM profiles and JVM properties files for the worker JVMs and for the master JVM that initializes the shared class cache. The sample JVM profiles DFHJVMPC (for a worker JVM) and DFHJVMCC (for a master JVM), and their associated JVM properties files, are provided to help you do this. *Java Applications in CICS* has more information about the shared class cache.

- You can specify a new level of reusability for JVMs. Before CICS TS for z/OS, Version 2.3, the level of reusability for JVMs was specified by the Xresettable option in the JVM profile. JVMs could either be resettable, that is, reused and reset between reuses (Xresettable=YES), or they could be single-use JVMs, that are thrown away after a single Java program has run in them (Xresettable=NO). CICS TS for z/OS, Version 2.3 introduces a new level of reusability for JVMs, the continuous JVM. JVMs with this level of reusability are kept in the JVM pool to be reused by further Java programs, but they are not reset after each use. The absence of a reset means that this type of JVM has faster transaction throughput and lower CPU usage, but it also means that the application code that runs in the next Java program or transaction might be affected by the actions of the previous program invocation. The three levels of reusability—the resettable JVM, the new continuous JVM, and the single-use JVM—are now controlled by the REUSE option in the JVM profile. REUSE=RESET creates a resettable JVM, which has the same reusability as the JVM that was created by Xresettable=YES; it is reused and reset. REUSE=YES creates the new continuous JVM, which is reused but not reset. REUSE=NO creates a single-use JVM, which has the same reusability as the JVM that was created by Xresettable=NO. "How JVMs are reused and reset" in *Java Applications in CICS*

explains more about the three levels of reusability for JVMs, and has information about some important considerations for application design and development for Java programs that will run in a continuous JVM.

The `Xresettable` option can still be used in JVM profiles to create a resettable JVM or a single-use JVM, but it cannot be used to create a continuous JVM. The CICS-supplied sample JVM profiles for CICS TS for z/OS, Version 2.3 use the REUSE option rather than the `Xresettable` option. If you are re-using existing JVM profiles and you want to include the REUSE option, it is advisable to remove the `Xresettable` option, although if the options conflict the REUSE option overrides the `Xresettable` option.

- You can browse through the JVMs in a particular CICS region, or inquire on the status of a particular JVM, using the CEMT or EXEC CICS INQUIRE JVM command.

- You can find the full path name of the HFS file for a JVM profile using the EXEC CICS INQUIRE JVMPROFILE command.

- You can change the JVM profile that is used for a program while CICS is running, using the JVMPROFILE option on the CEMT or EXEC CICS SET PROGRAM command. This enables you to phase in changes to your JVM profiles. Any instances of the program that are currently running in a JVM with the old JVM profile are unaffected, and are allowed to finish running their current Java program. New instances of the program will use a JVM with the new JVM profile that you have specified.

- You can use the JVMPROFILE and JVMPROGRAM options on the CEMT or EXEC CICS statistics commands to obtain statistics about JVM profiles or Java programs that run in JVMs. (To collect statistics about the amount of Language Environment heap storage that is used by the JVM, which forms part of the JVM profile statistics, the LEHEAPSTATS=YES option must be specified in the relevant JVM profiles.) You can also see additional statistics about waiting times for TCBs of the correct mode (in the TCB Pool statistics) and JVMs that use the shared class cache (in the JVM Pool statistics).

- You can redirect output from your JVMs by using the USEROUTPUTCLASS option in JVM profiles to specify a Java class. You can redirect the output to various destinations, and you can add time stamps and headers to the records. Two CICS-supplied sample classes are provided, and you can use these sample classes as they are, or modify them, or write your own classes based on the samples. The output redirection facility enables developers using the same CICS region to separate out their own JVM output, and direct it to an identifiable destination of their choice. You should not use this facility in a production region, because it has a negative effect on the performance of JVMs. "Redirecting JVM output" in *Java Applications in CICS* tells you how to use the output redirection facility.

- You can use the CICS-supplied transaction CETR to control tracing for JVMs. In the new JVM trace options screens, you can set the JVM trace options for each trace level, and in the Transaction and Terminal Trace screen, you can activate JVM tracing for transactions that use the JVM.

  As alternatives to using CETR, you can set the default JVM trace options using the new CICS system initialization parameters JVMLEVEL0TRACE, JVMLEVEL1TRACE, JVMLEVEL2TRACE, and JVMUSERTRACE, or the EXEC CICS SET JVMPOOL command. You can also activate JVM trace using the CICS system initialization parameters SPCTRSJ or STNTRSJ, or the EXEC CICS SET TRACETYPE command. You can find out what the current JVM trace options are using the EXEC CICS INQUIRE JVMPOOL command.

If you need to trace a JVM during its whole lifetime, including start-up and reset as well as the periods when it is being used by a transaction, you can still set and activate trace options using the **ibm.dg.trc.external** system property in the JVM properties file.

JVM trace can produce a large amount of output, so you should normally activate JVM trace for special transactions, rather than turning it on globally for all transactions.

For more information about JVM tracing, see "Controlling tracing for JVMs" in *Java Applications in CICS*.

- You can specify any dynamic link library (DLL) using the `Xrun` option in a JVM profile, and all the `Xrun` values that you include in a JVM profile are honoured. This means that you can specify values for the `Xrun` option in a JVM profile, and still use the JVM trace function for JVMs with that profile. (CICS automatically specifies `Xrundfhapjvmt` to drive the JVMRAS interface to perform tracing, and before CICS Transaction Server for z/OS, Version 2 Release 3, this was overridden by any value that you had specifed for the Xrun option.) The DLLs that you specify using the `Xrundllname` option must be present in directories on the library path for the JVM, which is specified by the LIBPATH option in the JVM profile.

- You can use assertions in your Java programs and set up assertion checking at runtime. Using the ENABLEASSERTIONS and DISABLEASSERTIONS options in a JVM profile, you can specify that assertion checking should be enabled or disabled in all classes (except system classes), or in a package and in any subpackages that it has, or in an individual class. You can also specify that assertion checking should be enabled or disabled for all system classes in the JVM, using the SYSTEMASSERTIONS option. You can find more information about programming with assertions, and about enabling and disabling assertions, at http://java.sun.com/j2se/1.4.1/docs/guide/lang/assert.html.

Migration for Java programs that do not run in a JVM (hpj-compiled Java program objects)

Run-time support for Java program objects and for hot-pooling (HPJ) is withdrawn in CICS Transaction Server for z/OS Version 3 Release 1. Any Java programs that you had processed using the VisualAge for Java, Enterprise Edition for OS/390 bytecode binder (hpj) to run as Java program objects in CICS, must be migrated to run in a Java Virtual Machine (JVM). To do this:

- Set up the JVM environment, as described in "Setting up and using JVMs" in *Java Applications in CICS*.

- Place the class files (with the extension .class) for the Java programs in directories in HFS where they can be loaded by the JVM. Ensure that CICS has read and execute access to these directories, as described in "Giving CICS regions access to z/OS UNIX System Services and HFS directories and files" in *Java Applications in CICS*. In order to create Java program objects, you had to use the `javac` compiler (or an equivalent Java compiler, such as VisualAge for Java or WebSphere Studio Application Developer) to compile the Java source files into class files, and then use the VisualAge for Java, Enterprise Edition for OS/390 bytecode binder to compile the class files into Java program objects. If you saved the class files during this process, you can use these to run in the JVM. If you did not keep the class files, re-run the Java compiler against your Java source files to produce new class files. If you want to, you can build the class files into packages or JAR files (with the extension .jar) before placing them in the HFS directory.

- Modify the PROGRAM resource definitions to add the JVM, JVMCLASS, and JVMPROFILE options, and add the classes that the applications use to the class paths for their JVMs, as described in "Enabling applications to use a JVM" in *Java Applications in CICS*. Note that placing application classes on the shareable application class path, rather than on the standard class path, produces the best performance in a resettable JVM, and it should be your normal choice for loading application classes in a production environment.
- If the Java programs access DB2, follow the instructions in "Requirements to support Java programs in the CICS DB2 environment" in *CICS DB2 Guide* to add the necessary DB2 directories and files to the class paths in the JVM profiles, and ensure that you have applied any DB2 APARs that are needed for your version of DB2.
- Test that the Java programs work correctly in the JVMs that you have defined for them. Pay particular attention to the level of reusability that you choose for the JVMs, which is described in "How JVMs are reused and reset" in *Java Applications in CICS*.
  - Java program objects that are migrated to run in a **resettable JVM** are likely to suffer a significant performance degradation if they use Java methods that make the JVM unresettable. If a JVM is found to be unresettable, it is destroyed after use, so CICS incurs the CPU cost of initializing a new JVM. For details of the Java methods that make the JVM unresettable see the IBM Software Developer Kit for z/OS, Java 2 Technology Edition, Version 1.4.2 document, *Persistent Reusable Java Virtual Machine User's Guide*, SC34-6201. "Resettable JVMs (REUSE=RESET)" in *Java Applications in CICS* explains how to log unresettable actions when you are testing your applications in a JVM.
  - If your applications must perform unresettable actions, you can use a **single-use JVM**, but the performance of this type of JVM is inferior. With a single-use JVM, the CPU cost of initializing a new JVM is always incurred, and the only advantage over a resettable JVM is that no time is spent checking whether or not the JVM has been made unresettable. The best strategy is to re-design the programs as soon as possible to eliminate unresettable actions, so that they can run in a resettable JVM or in a continuous JVM.
  - A **continuous JVM** should not be selected until you are sure that your programs do not perform any unwanted unresettable actions, which would change the state of the JVM in undesirable ways. The continuous JVM is not destroyed if an unresettable action is performed in it, but careful coding is required to ensure that subsequent program invocations in the JVM are not adversely affected. "Continuous JVMs (REUSE=YES)" in *Java Applications in CICS* explains the design guidance for programs that are to be run in a continuous JVM. If you can re-design your programs following this guidance, the continuous JVM provides the best performance of the three types of JVM.

To avoid problems with deprecated APIs, you should develop all new Java programs forCICS Transaction Server for z/OS, Version 3 Release 1 using an application development environment that supports Java 2 at the same version of Java as used by CICS. You may run code compiled with an older version of Java in a new runtime, provided that it does not use APIs that have been removed in the newer version of Java. Note also that enterprise beans that support the EJB 1.0 specification need to be migrated to the EJB 1.1 specification level using the Assembly Toolkit (ATK) or the Application Assembly Tool (AAT), which are supplied with IBM WebSphere Application Server.. Enterprise beans developed using any version of the EJB specification after EJB 1.1 must restrict themselves to the EJB 1.1 APIs.

Chapter 22. Migration planning for Language Environment

In CICS Transaction Server for z/OS Version 3 Release 1, interfaces to the VS COBOL II, OS PL/I and C/370 runtimes are removed. Applications compiled and linked with these non Language Environment conforming products usually execute successfully under Language Environment in *compatibility mode*.

CICS support for OS/VS COBOL runtime is removed- Applications compiled at this level of COBOL must be re-compiled with a supported level of COBOL

Refer to the *Migration guides for Language Environment* and to the *Migration guides for the language in use* for further information.

Chapter 23. Migration planning for the Link3270 bridge with the ACCUM option

This chapter covers migration for users of the Link3270 bridge with the ACCUM option.

There are two versions of the Link3270 bridge in CICS TS for z/OS, Version 3.1.

Link3270 bridge with basic support, provides the same support as that provided by the CICS Transaction Server for z/OS, Version 2 Release 2 Link3270 bridge.

Link3270 bridge with extended support provides support for the ACCUM option on the SEND TEXT, SEND MAP, and SEND CONTROL commands. To provide this support, two new vectors are introduced: SEND PAGE and PURGE MESSAGE. New copybooks are also provided.

If you only need basic support, no action is necessary. Use the basic copybooks as before. There is no need to recompile any existing Link3270 bridge programs.

If you wish to take advantage of the extended support provided by CICS Transaction Server for z/OS, Version 2 Release 3, recompile your Link3270 bridge programs (or compile new programs) using the extended copybooks and the extended support will be provided.

More information about the use of the Link3270 bridge can be found in the *CICS External Interfaces Guide*

Chapter 24. Migration planning for multiregion operation (MRO)

This chapter covers migration for MRO users. It covers the following topics:
- "DFHIRP coexistence"
- "Migrating to the latest DFHIRP"
- "End-of-memory clean-up routine" on page 99.
- "CICS enables the READ TIMEOUT value" on page 99

DFHIRP coexistence

To use CICS multiregion operation (MRO) support, install DFHIRP in the link pack area (LPA). DFHIRP can only be used from the LPA. This means that in an MVS image there can only be one version of the module named DFHIRP, which must be at the highest release level of the CICS regions that run in that MVS image.

Within a Parallel Sysplex®, where MRO communication between MVS images is through XCF/MRO, the DFHIRP programs installed in the different MVS images can be at different release levels. However, the DFHIRP in an MVS image must be installed from the highest release of CICS running in that MVS image. For example, a CICS/ESA Version 4 DFHIRP can communicate with a CICS Transaction Server DFHIRP across XCF/MRO, but the CICS regions running in the MVS with the Version 4 DFHIRP cannot be later than CICS/ESA Version 4. See Figure 3 on page 98 for an illustration of valid configurations for MRO with different levels of DFHIRP installed in the sysplex.

Migrating to the latest DFHIRP

The CICS TS for z/OS, Version 3.1 DFHIRP module is compatible with earlier releases, and is designed to work with all releases of CICS.

The following steps are a guide to migrating to MRO, with the latest DFHIRP and DFHCSVC modules installed in the MVS link pack area (LPA). For information about how to perform some of these steps, such as installing the SVC or IRP modules in the LPA, see the *CICS Transaction Server for z/OS Installation Guide*. Note that these steps assume that RACF® is your external security manager (ESM).

1. **Install the CICS SVC**
   Install the CICS TS SVC routine, DFHCSVC, in the LPA, and specify a new CICS SVC number for this routine in the MVS SVCPARM table. (If the new DFHCSVC has to coexist with an older version, rename one of them so that both versions can be installed in the LPA. However, this is not recommended or necessary: DFHCSVC is compatible with earlier releases and the latest CICS TS version supports all the earlier releases of CICS.)

2. **Test the CICS SVC**
   Test the new SVC on CICS TS stand-alone regions, without using any MRO. You can do this running the CICS IVP, DFHIVPOL.

3. **Install the IRP**
   Install the CICS TS interregion communication program, DFHIRP, in a suitable LPA library, and IPL MVS (with the CLPA option).

   Do not use the dynamic LPA function to replace DFHIRP for migration between releases, as this can cause incompatibility between control blocks, resulting in abend situations.

4. **Test current production release**
   Test your production MRO CICS regions, under your existing release of CICS, but using the new SVC number and the new DFHIRP. For this test, run without any logon or bind-time security checking—that is, do not define any RACF FACILITY class profiles.

5. **Define RACF FACILITY resource class profiles**
   Define the required DFHAPPL.*applid* profiles in the RACF FACILITY general resource class. When the profiles are ready for all the MRO regions, test the production regions again with the new SVC and DFHIRP, this time using the FACILITY class profiles for logon and bind-time security checking.

6. **Cutover to production with the new IRP**
   If the production MRO regions successfully log on to the new IRP with the new SVC, and bind-time security checking works successfully, use the new DFHIRP and SVC for the production regions.

7. **Test MRO with CICS TS regions**
   With the production regions running successfully under the CICS TS SVC and IRP, you can initialize and test some CICS Transaction Server regions using MRO. These test regions can coexist in the same MVS image as the production regions, all using the same SVC and IRP.

```
┌──────────────────────────── SYSPLEX1 ──────────────────────────────┐
│  ┌─────────────────────┐  ┌─────────────────────┐  ┌─────────────────────┐  │
│  │        MVSA         │  │        MVSB         │  │        MVSC         │  │
│  │ L ┌───────────────┐ │  │ L ┌───────────────┐ │  │ L ┌───────────────┐ │  │
│  │ P │  CICS TS 2.2  │ │  │ P │  CICS TS 1.3  │ │  │ P │   CICS 4.1    │ │  │
│  │ A │    DFHIRP     │ │  │ A │    DFHIRP     │ │  │ A │    DFHIRP     │ │  │
│  │   └───────────────┘ │  │   └───────────────┘ │  │   └───────────────┘ │  │
│  │   ┌───────────────┐ │  │   ┌───────────────┐ │  │   ┌───────────────┐ │  │
│  │   │  CICS TS 2.2  │ │  │   │  CICS TS 1.3  │ │  │   │   CICS 4.1    │ │  │
│  │   │    region     │ │  │   │    region     │ │  │   │    region     │ │  │
│  │   └───────────────┘ │  │   └───────────────┘ │  │   └───────────────┘ │  │
│  │   ┌───────────────┐ │  │   ┌───────────────┐ │  │   ┌───────────────┐ │  │
│  │   │  CICS TS 1.3  │ │  │   │   CICS 4.1    │ │  │   │   CICS 3.3    │ │  │
│  │   │    region     │ │  │   │    region     │ │  │   │    region     │ │  │
│  │   └───────────────┘ │  │   └───────────────┘ │  │   └───────────────┘ │  │
│  │   ┌───────────────┐ │  │   ┌───────────────┐ │  │                     │  │
│  │   │   CICS 4.1    │ │  │   │   CICS 3.3    │ │  │                     │  │
│  │   │    region     │ │  │   │    region     │ │  │                     │  │
│  │   └───────────────┘ │  │   └───────────────┘ │  │                     │  │
│  └─────────────────────┘  └─────────────────────┘  └─────────────────────┘  │
└────────────────────────────────────────────────────────────────────┘
```

Figure 3. Illustration of valid configurations using DFHIRP in a sysplex

Note: All the CICS regions in SYSPLEX1 shown in Figure 3 can communicate across MRO links, because the DFHIRP in each MVS supports XCF/MRO.

- In MVSA, DFHIRP must be at the CICS TS 2.2 level, because CICS TS 2.2 is the latest release in this MVS image.
- In MVSB, DFHIRP must be at the CICS TS 1.3 level, because CICS TS 1.3 is the latest release in MVSB.
- MVSC is running with the CICS/ESA 4.1 DFHIRP installed, because CICS/ESA 4.1 is the latest release of CICS MVSC.

End-of-memory clean-up routine

An MRO end-of-memory clean-up routine, which is also used by console message-handling support, is no longer needed in DFHIRP. Because of this change in MRO, DFHSSEN, which is an alias of DFHIRP in earlier releases, is supplied as a separate module. Installing DFHSSEN in the LPA continues to be required for console message-handling support.

See the *CICS Transaction Server for z/OS Installation Guide* for more information about requirements for CICS console message-handling support.

CICS enables the READ TIMEOUT value

In earlier releases of CICS, read time-out is ignored for MRO sessions, tasks waiting on an MRO connection can hang indefinitely. If, for example, transactions in an AOR stall or deadlock for any reason the corresponding relay transactions in an MRO-connected TOR are left hanging. Eventually a problem in an AOR can cause the TOR to stall completely. Although you can prevent new transactions from being routed to, or queued for, the offending AOR, there has not been, until now, an easy way to purge every task that is in flight.

Now, CICS enables the READ TIMEOUT value on the transaction profile definition for MRO sessions. The field in the profile already exists. CICS honours it for ISC but used to ignore it for MRO. CICS now checks this RTIMOUT value for MRO sessions. A new abend code is issued if the RTIMOUT value is exceeded for MRO sessions. By enabling the READ TIMEOUT value on the transaction profile definition for MRO sessions, a cause of intersystem queuing (sometimes referred to as "sympathy sickness") is eliminated.

The situations which may benefit from this change include some which, because they result from stalls or deadlocks elsewhere, are unlikely to resolve themselves unaided.

Chapter 25. Migration planning for Named Counter and Shared Temporary Storage servers

Named Counter servers and Shared Temporary Storage servers can be at different levels in different MVS images, even when connected to the same coupling facility structure. They still function correctly in these circumstances. Each MVS image only uses a single server for each coupling facility structure, so all accesses to the coupling facility structure from each MVS image execute server code that is at the same level.

All of the CF-related servers including the named counter server can be closed down immediately using the server CANCEL command.

There is no difference between the temporary storage server and the named counter server relating to recovery, in that both servers store all of their data in the coupling facility structure, so it will be lost if the structure fails or the coupling facility is re-IPLed. There may be a distinction from the application point of view, in that the scratch pad information stored in the temporary storage structure may be less important to continuous operation than sequence numbers stored in named counters.

The Application Programming Guide contains a section on Named Counter Recovery considerations. In a production environment, it is important to have some separate means of recreating the named counters if the coupling facility structure is lost, for example by determining the latest used counters from data bases.

Chapter 26. Migration planning for threadsafe programming and the open transaction environment (OTE)

This chapter describes the changes you might have to make to your existing applications or configuration to exploit the open transaction environment (OTE) and to make your applications threadsafe. It covers the following topics:
- "What is the open transaction environment and how can I benefit from it?"
- "How can I make my applications exploit the open transaction environment?" on page 106
- "Important changes to accounting for processor time in the open transaction environment" on page 108

What is the open transaction environment and how can I benefit from it?

Before CICS Transaction Server for OS/390, Version 1 Release 3, user applications and exits operated in a restricted, or closed, environment. Although the applications could use the functionally-rich CICS application program interface (API), direct invocation of other services was not supported. This is because CICS ran all user transactions under a single z/OS TCB, known as the CICS quasi-reentrant (QR) TCB. Direct invocation of other services outside the scope of the CICS permitted interfaces could interfere with the use by CICS of the QR TCB. In particular, requests resulting in the suspension ("blocking") of the QR TCB, which happens when an MVS wait is issued, would cause all CICS tasks to wait.

The open transaction environment (OTE) function was added to CICS Transaction Server for OS/390, Version 1 Release 3 and later versions. The open transaction environment is an environment where CICS application code can use non-CICS services (facilities outside the scope of the CICS API) within the CICS address space, without interference with other transactions. Applications that exploit the open transaction environment run on their own open TCB, rather than on the QR TCB. Unlike the QR TCB, CICS does not perform sub-dispatching on an open TCB. If the application running on an open TCB invokes a non-CICS service which blocks the TCB, the TCB blocking does not affect other CICS tasks. For example, some services provided by DB2, MVS, UNIX System Services, or TCP/IP, might result in TCB blocking.

The open TCBs that are used in the open transaction environment are managed in separate pools, with each pool containing a different type, or operational mode, of open TCB. Each mode has a specific purpose, and is handled by CICS in a different way. The types, or modes, of open TCB are:

J8 mode TCBs and J9 mode TCBs
> Open TCBs that are used to run Java programs under a Java Virtual Machine (JVM). The JVM is created on the TCB.

> J8 TCBs are used for JVMs when the Java programs are defined as executing in CICS key, and J9 mode TCBs are used for JVMs when the Java programs are defined as executing in user key and storage protection is active. The maximum number of these TCBs that CICS will create in the JVM pool is controlled by the MAXJVMTCBS system initialization parameter. "How CICS manages JVMs in the JVM pool" in *Java Applications in CICS* has more information about how CICS manages JVMs and their TCBs.

L8 mode TCBs and L9 mode TCBs
are both used to run OPENAPI programs, that is those defined as
OPENAPI by their PROGRAM resource definition.
- L8 mode TCBs are used for CICSKEY OPENAPI application programs.
- L9 mode TCBs are used for USERKEY OPENAPI application programs.

The MAXOPENTCBS system initialization parameter controls the number of
L8 and L9 TCBs in the OPEN TCB pool.

L8 mode TCBs
are also used when programs need access to a resource manager through
a task-related user exit (TRUE) enabled using the OPENAPI option on the
ENABLE PROGRAM command.

The CICS DB2 task-related user exit operates in OPENAPI mode (it is an
open API TRUE). In this situation, the CICS DB2 attachment facility uses L8
TCBs for DB2 request processing. "Overview: How threads work" in the
CICS DB2 Guide has more information about how CICS uses open TCBs
as thread TCBs for the CICS DB2 attachment facility. "Enabling CICS DB2
applications to exploit the open transaction environment (OTE) through
threadsafe programming" in the *CICS DB2 Guide* explains what your CICS
DB2 application programs must do in order to gain performance benefits by
continuing to run on the L8 mode TCB after the DB2 request has been
completed.

L8 mode TCBs
are also used by CICS itself, because CICS uses OPENAPI CICSKEY
programs which run on L8 TCBs:

- when accessing doctemplates and HTTP static responses that are stored
  on Hierarchical File System (HFS).
- when processing WebService requests and parsing XML.

X8 mode TCBs and X9 mode TCBs
are both used to run C and C++ programs compiled with the XPLINK
option. X8 TCBs are used for programs in CICS key, and X9 mode TCBs
are used for programs in user key. Each instance of an XPLink program
uses one X8 or X9 TCB. in the *CICS Application Programming Guide* has
more information about using XPLink.

"System initialization parameters for open TCBs" in the *CICS System Definition
Guide* has more information about how CICS manages open TCBs, and about
specifying the MAXJVMTCBS, MAXOPENTCBS and MAXXPTCBS system
initialization parameters.

Note: All references to DB2 assume that you are using one of the currently
supported versions of DB2. At the time of publishing these are Versions 7, 8
and 9. There are differences for versions prior to Version 6, and if you are
attempting to work with any such unsupported combination, you should read
the relevant information from earlier CICS publications.

The first benefit of the open transaction environment was that Java programs could
run under CICS, either as hot-pooled Java program objects or in a JVM. From
CICS Transaction Server for z/OS, Version 2 Release 2, applications that involve a
task-related user exit enabled using the OPENAPI option on the ENABLE
PROGRAM command can also exploit the open transaction environment to provide
performance benefits. Task-related user exits like this are known as open API
TRUEs. An open API TRUE will be given control under an L8 mode open TCB, and
can use non-CICS APIs without having to create, manage and switch between

subtask TCBs. The CICS DB2 task-related user exit now operates as an open API TRUE . Existing or new CICS DB2 applications written in any language that access DB2 have the opportunity to gain the performance benefits provided by the open transaction environment.

These performance benefits can be gained because open TCBs, unlike the QR TCB or subtask thread TCBs, may be used for both non-CICS API requests (including requests to DB2) and application code. Because application code can be executed on the open TCB, the application should not need to switch between different TCBs several times during the execution of a CICS DB2 application. Before the open transaction environment was available, CICS used the QR TCB for the CICS DB2 task-related user exit and for the application program's code. Subtask thread TCBs were used for requests to DB2, and switching between the subtask TCB and the QR TCB took place for every DB2 request. With the open transaction environment, the same L8 TCB can be used by the CICS DB2 task-related user exit, the SQL requests that CICS makes to DB2, *and* any subsequent application code. The most important condition to be met to achieve this is that the user application program, the EXEC CICS commands used in the application, and any user exit programs involved with the application program, must be **threadsafe**. Only code that has been identified as threadsafe is permitted to execute on open TCBs. The next topic explains in detail what it means for an application to be threadsafe.

Where the correct conditions are met, the use of open TCBs for CICS DB2 applications decreases usage of the QR TCB, and avoids TCB switching. An ideal CICS DB2 application program for the open transaction environment is a threadsafe program, containing only threadsafe EXEC CICS commands, and using only threadsafe user exit programs. An application like this will move to an L8 TCB when it makes its first SQL request, and then continue to run on the L8 TCB through any amount of DB2 requests and application code, requiring no TCB switching. This situation produces a significant performance improvement where an application program issues multiple SQL calls. The gains are also significant when using an enterprise bean, because when enterprise beans make DB2 requests, they require additional TCB switches to and from the enterprise bean's own TCB. If the application program does not issue many SQL calls, the performance benefits might not be as significant.

In order to gain these performance benefits for CICS DB2 applications, you *must* meet the following conditions:

1. CICS must be connected to DB2 Version 6 or later. "Migrating to a different release of DB2" in the *CICS DB2 Guide* has full details of the CICS DB2 configuration needed to support the open transaction environment, including APARs that must be applied for DB2 and for CICS.

2. The system initialization parameter FORCEQR must not be set to YES. FORCEQR forces programs defined as threadsafe to run on the QR TCB, and it might be set to YES as a temporary measure while problems connected with threadsafe-defined programs are investigated and resolved.

3. The CICS DB2 application must have threadsafe application logic (that is, the native language code in between the EXEC CICS commands must be threadsafe), use only threadsafe EXEC CICS commands, and be defined to CICS as threadsafe. It must use only threadsafe dynamic plan exits, task-related user exits and global user exits. Only code that has been identified as threadsafe is permitted to execute on open TCBs. If your CICS DB2 application is not defined as threadsafe, or if it uses EXEC CICS commands or exits which are not threadsafe, TCB switching will take place and some or all of

the performance benefits of OTE exploitation will be lost. "How can I make my applications exploit the open transaction environment?" discusses all these requirements in more detail.

Additional information about how CICS DB2 applications use open TCBs can be found in "SQL, threadsafe and other programming considerations for CICS DB2 applications" in the *CICS DB2 Guide*

How can I make my applications exploit the open transaction environment?

When an application is defined to CICS as quasi-reentrant, it executes on the QR TCB. When running under this TCB, a program can be sure that no other quasi-reentrant program can run until it relinquishes control during a CICS request. Quasi-reentrancy therefore allows programs to access globally shared resources—for example, the CICS common work area (CWA)—without the need to protect those resources from concurrent access by other programs. Such resources are effectively locked exclusively to the running program, until it issues its next CICS request.

In the CICS open transaction environment (OTE), when application programs, task-related user exits (TRUEs), global user exit programs, and user-replaceable modules are defined to CICS as threadsafe, they can run concurrently on open TCBs. Because of this, they cannot rely on quasi-reentrancy to protect shared resources from concurrent access by another program. Furthermore, quasi-reentrant programs might also be placed at risk if they access shared resources that can also be accessed by a user task running concurrently under an open TCB. The techniques used by user programs to access shared resources must therefore take into account the possibility of simultaneous access by other programs. To gain the performance benefits of the open transaction environment while maintaining the integrity of shared resources, serialization techniques must be used to prohibit concurrent access to shared resources. Programs that use appropriate serialization techniques when accessing shared resources are described as threadsafe.

As we saw in "What is the open transaction environment and how can I benefit from it?" on page 103, only applications that involve a task-related user exit (TRUE) enabled using the OPENAPI option, such as applications which access DB2 resources, can gain performance benefits from being threadsafe. These are the only applications that might or might not be able to run on an open TCB.

The goal of making programs threadsafe is to enable them to remain on an open TCB, rather than switching back and forth between the open TCB and the QR TCB. TCB switching occurs in the following circumstances:

- When a program that is not defined as threadsafe makes a DB2 request, CICS switches from the QR TCB (where the program is executing) to an open TCB, and back to the QR TCB again when the DB2 request is complete.
- When a user exit program that is not defined as threadsafe is used in the course of a DB2 request, CICS switches from the open TCB (where the DB2 request is executing) to the QR TCB. The user exit program is executed on the QR TCB, and then the task is switched back to the open TCB to complete the DB2 request. For example, the XRMIIN and XRMIOUT global user exits might be invoked in the course of the DB2 request. If the exit programs are not defined as threadsafe, this TCB switching occurs. If the exit programs are defined as threadsafe, processing will continue throughout on the open TCB.

- When a program that is defined as threadsafe and is executing on an open TCB invokes any EXEC CICS commands which are not threadsafe, CICS switches back from the open TCB to the QR TCB to execute the non-threadsafe code. The program then continues to execute on the QR TCB. If the program does not make any further DB2 requests, then the switch back to the QR TCB is only a disadvantage because it increases the usage of your QR TCB for the time taken to run any remaining application code. However, if the program makes any further DB2 requests, CICS must switch back again to the open TCB.
- When a program that is defined as threadsafe and is executing on an open TCB invokes a task-related user exit program which is not defined as threadsafe, CICS switches back to the QR TCB and gives control to the task-related user exit program. When the task-related user exit program completes processing, the application program continues to execute on the QR TCB, in the same way as it would after issuing a non-threadsafe EXEC CICS command.
- When a program that is defined as threadsafe and is executing on an open TCB invokes a threadsafe CICS command, it is possible for a global user exit to be invoked as part of executing the command. If a global user exit program is used which is not defined as threadsafe, CICS switches back to the QR TCB and gives control to the global user exit program. When the user exit program completes processing, CICS switches back to the open TCB to continue processing the threadsafe CICS command.
- When a program that is defined as threadsafe and is executing on an open TCB completes, CICS switches back to the QR TCB for task termination. This switch is always necessary.

The maximum TCB switching for a CICS DB2 application would occur if your program used a non-threadsafe user exit program and a non-threadsafe EXEC CICS command after every DB2 request.

If you want to make an application program remain on an open TCB:

1. **Ensure that the program's logic is threadsafe.** That is, the native language code between the EXEC CICS commands must be threadsafe. If you define a program to CICS as threadsafe but include application logic that is not threadsafe, the results are unpredictable, and CICS is not able to protect you from the possible consequences. "Threadsafe programs" in the *CICS Application Programming Guide* tells you how to produce threadsafe application logic.

2. **Ensure that the program uses only threadsafe EXEC CICS commands.** The commands that are threadsafe are indicated in the command syntax diagrams in the *CICS Application Programming Reference* and the *CICS System Programming Reference* with the statement "This command is threadsafe", and are listed in Appendix L of the *CICS Application Programming Reference* and Appendix D of the *CICS System Programming Reference*. If you include a non-threadsafe EXEC CICS command in a program which is running on an open TCB, CICS switches back from the open TCB to the QR TCB to ensure that the command is processed safely. The TCB switching could be detrimental to the application's performance.

   As well as checking EXEC CICS commands that you code explicitly, be aware of high-level language constructs or Language Environment callable services used by your program that result in using CICS services. CICS services used in this way might involve non-threadsafe CICS commands, and cause a switch back to the QR TCB. In particular, the COBOL statement DISPLAY UPON SYSOUT, some types of PL/I and C++ output, and the Language Environment callable services CEEMOUT and CEE3DMP, write data to the Language Environment transient data destinations CESE and CESO. This involves an EXEC CICS WRITE TD command, which is not threadsafe.

3. **Ensure that the program is defined to CICS as threadsafe.** Use the CONCURRENCY attribute of the program resource definition to do this. By defining a program to CICS as threadsafe, you are only specifying that the application logic is threadsafe, not that all the EXEC CICS commands included in the program are threadsafe. CICS can ensure that EXEC CICS commands are processed safely by using TCB switching. In order to permit your program to run on an open TCB, CICS needs you to guarantee that your application logic is threadsafe.

4. **Ensure that any user exit programs in the execution path used by the program are coded to threadsafe standards** *and* **defined to CICS as threadsafe.** This might include dynamic plan exits, global user exits, or task-related user exits. (Note for task-related user exits, enabling the exit program using the OPENAPI option on the ENABLE PROGRAM command means that CICS overrides the CONCURRENCY setting on the exit's program definition with OPENAPI.) When CICS is connected to DB2 Version 6 or later, the CICS DB2 task-related user exit DFHD2EX1 is threadsafe. "SQL, threadsafe and other programming considerations for CICS DB2 applications" in the *CICS DB2 Guide* has more information on other exits that are particularly important for CICS DB2 requests. These exits include the default dynamic plan exit DSNCUEXT (which is not defined as threadsafe), the alternative dynamic plan exit DFHD2PXT (which is defined as threadsafe), and the global user exits XRMIIN and XRMIOUT. Also be aware of the global user exits XEIIN and XEIOUT, which are invoked before and after EXEC CICS commands, and XPCFTCH, which is invoked before a PPT-defined program receives control. Be sure that user exit programs supplied by any vendor software are coded to threadsafe standards and defined to CICS as threadsafe.

5. **If you are coding a user exit program** (a global user exit or a task-related user exit), you can define it as threadsafe so that it can be used on the same L8 TCB as a threadsafe application which calls it. Additionally, a task-related user exit can be enabled using the OPENAPI option on the ENABLE PROGRAM command so that it will be given control under an L8 TCB, use non-CICS APIs without having to create and manage subtask TCBs, and exploit the open transaction environment for itself. (Enabling the exit program using the OPENAPI option on the ENABLE PROGRAM command means that CICS overrides the CONCURRENCY setting on the exit's program definition with OPENAPI.) Global user exit programs can be treated in the same way as an ordinary application program—by using threadsafe application logic and threadsafe EXEC CICS commands, and defining the program as threadsafe. "Writing global user exit programs" in the *CICS Customization Guide* has general information about writing this type of program. For task-related user exit programs, see "Writing a task-related user exit program" in the *CICS Customization Guide* for more detailed information about how this type of program can exploit the open transaction environment safely. Note when you enable an exit program using the OPENAPI option, this indicates to CICS that the program's logic is threadsafe.

Important changes to accounting for processor time in the open transaction environment

The CICS DB2 attachment facility uses CICS-managed open TCBs rather than CICS DB2 subtask TCBs. This means the CICS monitoring facility can measure activity that was previously (with DB2 Version 5 or earlier,) only reported in the DB2 accounting record (the SMF type 101 record). For example, CICS can now measure the processor time consumed on the DB2 thread and the processor time consumed in DB2 (the CLASS 1 and CLASS 2 CPU time). When CICS is using L8 open

TCBs, the CPU time reported for these TCBs by the CICS monitoring facility includes the DB2 CLASS 1 processor time.

When CICS is connected to DB2 Version 6 or later, **do not** add together the processor time from the CICS records (SMF type 110 records) and the DB2 accounting records (SMF type 101 records) when calculating the total processor time for a single transaction, because the DB2 processor time would then be included twice. The total processor time for a single transaction is recorded in the USRCPUT field in the CICS records (performance class data field 008 from group DFHTASK). This field includes all processor time used by the transaction when it was executing on any TCB managed by the CICS dispatcher. CICS-managed TCBs include the QR, RO, CO, J8, J9, L8, L9, X8 and X9 mode TCBs.

In the open transaction environment, the CICS L8 task processor time can also include the cost of creating a DB2 thread. If a transaction causes a DB2 thread to be created, you can expect the total task processor time accounted for to be higher than that accounted for by a CICS system running with earlier DB2 releases. Correspondingly, if at the end of a transaction, the thread is terminated (because it is unprotected and no other task is waiting to use it), then the cost of thread termination is included in the CICS L8 task processor time. Again, this cost is not accounted for by a CICS system connected to DB2 Version 5 or earlier.

From a DB2 perspective, when the CLASS 1 recording becomes active for a thread, it is now recording time spent on the L8 open TCB. Because the L8 TCB is used for both CICS activity and DB2 activity, this includes processor time spent in the CICS-DB2 attachment facility, including trace calls, and also includes processor time spent running application code (if the application is threadsafe) and threadsafe CICS commands on the open TCB. If a thread is reused, the thread housekeeping processor time is also included in the CLASS 1 processor time. As in previous releases, there is a proportion of thread creation and thread termination processing that is not captured by CLASS 1 time. The CLASS 1 processor time does not include any time spent running application code on the QR TCB. (When an application issues a non-threadsafe CICS command, it is forced back to the QR TCB.) When CICS is connected to DB2 Version 5 or earlier, CLASS 1 processor time does not include any processor time spent in application code, because all application code is executed under the QR TCB. The CLASS 2 processor time recorded by DB2, which is a subset of the CLASS 1 processor time showing the time spent in DB2 itself, is not affected by the open transaction environment.

For more information about calculating processor times for CICS and DB2 and a full explanation of how processor times are recorded, see "Accounting for processor usage in a CICS DB2 environment" in the *CICS DB2 Guide*.

Chapter 27. Migration planning for CICS Web support applications

This topic tells you about:

- "Migration of existing CICS Web support applications"
- "Migration to the new CICS Web support function" on page 113

Migration of existing CICS Web support applications

CICS Transaction Server for z/OS, Version 3 Release 1 is designed to support your existing CICS Web support architecture for both Web-aware and non-Web-aware application programs. The `EXEC CICS WEB` API command changes are designed to allow existing Web-aware application programs that send and receive HTTP messages to work unchanged, until you choose to migrate them to take advantage of the enhancements that are now available. If you continue to use existing CICS Web support applications, note these migration points:

- **If you are using CICS Web support to process non-HTTP requests, specify the new USER protocol on the TCPIPSERVICE definition that defines the port for these requests.** This also applies to HTTP requests with nonstandard request methods, which are now rejected if they are received on the HTTP protocol (previously, they were accepted and processed as non-HTTP). Processing for all non-HTTP requests must now be carried out under the USER protocol, so that they are protected from the basic acceptance checks which CICS carries out for requests using the HTTP protocol. The requests are flagged as non-HTTP and passed unchanged to the analyzer program for the TCPIPSERVICE. CICS Web support facilities are used for handling the request, but no acceptance checks are carried out for messages sent and received using this protocol.

  Note: Because only one active TCPIPSERVICE definition can exist for each port, non-HTTP requests can no longer use the same port as HTTP requests. The well-known port numbers 80 (for HTTP) and 443 (for HTTPS) must have the HTTP protocol and therefore cannot accept non-HTTP requests. Web clients must specify any changed port in the URL for their requests.

- **Check the settings for your TCPIPSERVICE resource definitions with the HTTP protocol.**
  1. The SOCKETCLOSE attribute must no longer have a zero setting (SOCKETCLOSE(0)).
     - A zero setting for SOCKETCLOSE means that CICS closes the connection immediately after receiving data from the Web client, unless further data is waiting. This means that persistent connections cannot be maintained.
     - A non-zero setting for SOCKETCLOSE enables persistent connections with both HTTP/1.1 clients, and HTTP/1.0 clients (where the client supports this).
  2. The new MAXDATALEN option should be specified to limit the maximum length of data that may be received by CICS as an HTTP server.
     - This setting helps to guard against denial of service attacks involving the transmission of large amounts of data.

3. If you are using SSL, there are some changes to the security options available on the TCPIPSERVICE resource definition. The topic *Improvements to Internet security* in the *CICS Transaction Server Release Guide* explains these changes.

- **The code page conversion table (DFHCNV) is no longer required for CICS Web support.** However, if you want to continue to use an analyzer program that you coded in an earlier CICS release to reference DFHCNV, you must either continue to supply the entries in the code page conversion table, or change the analyzer program. Changing the analyzer program involves coding two new output parameters to specify the client and server code pages, in place of the output parameter that specified the name of a DFHCNV entry. If you do this, you do not need to migrate your DFHCNV entries.

- **If you use a code page other than 037 (the EBCDIC Latin character set) in your CICS Web support applications, use the LOCALCCSID system initialization parameter to specify this code page.** The LOCALCCSID system initialization parameter supplies the code page into which CICS converts inbound HTTP headers and query strings, including form data transmitted in a query string. Before CICS Transaction Server for z/OS, Version 3 Release 1, the code page for this conversion was specified by the DFHWBHH template in the DFHCNV code page conversion table. You can set the LOCALCCSID system initialization parameter to any EBCDIC code page into which the ASCII Latin-1 character set ISO-8859-1 (code page 819) can be converted. If LOCALCCSID is set to an unsuitable code page, CICS uses the default 037 for inbound HTTP headers and query strings.

- **If you have modified the user-replaceable Web error program DFHWBEP to customize the HTTP responses provided in error situations, be aware that CICS now uses additional status codes, and uses some existing status codes in a wider range of situations.**
  1. Check that your program is using an appropriate range of input parameters to identify the situation to which the customized response applies, rather than relying on the status code alone. The error code, abend code, message number, response and reason codes, or program name can be used to identify the situation that has given rise to the HTTP response. If these checks are not made, you might find that where CICS is using the status code for a new purpose, an inappropriately customized response is returned.
  2. Check that your program includes logic to pass through unchanged any HTTP responses with status codes that are not known to the program.

- **The DFHWBCLI interface, as supplied in SupportPac CA1M for CICS TS 2.3, is still supported in CICS Transaction Server for z/OS, Version 3 Release 1.** To gain enhanced functionality, you can migrate HTTP client applications that used the DFHWBCLI interface, to use the CICS WEB API commands for client requests (with the SESSTOKEN option). One important difference to note is that in the CICS WEB API, the use of a proxy server is specified by a user exit on the WEB OPEN command (XWBOPEN), and the URL of the proxy server is supplied by that user exit. The topic *Support for HTTP client requests from CICS applications* in the *CICS Transaction Server Release Guide* describes how HTTP client requests can now be made.

  If you continue to use the DFHWBCLI interface, note that the WBCLI_MEDIATYPE parameter is now required when using the DFHWBCLI interface for SEND requests that use the HTTP POST method (requests where WBCLI_FUNCTION_SEND and WBCLI_METHOD_POST are both set).

Migration to the new CICS Web support function

CICS Web support in CICS Transaction Server for z/OS, Version 3 Release 1 has many enhancements to provide automatic and administrator control of functions that were previously handled by user-replaceable programs. In particular, you are recommended to investigate migration possibilities for the following elements of your CICS Web support architecture:

- You should usually be able to replace the request processing functions of your analyzer program with URIMAP resource definitions, which can be changed and controlled using CICS system programming commands. URIMAP definitions can be used to match the URLs of requests and map them to application programs, and specify a converter program, alias transaction and user ID. If your analyzer program provides additional functions, you can continue to use it instead of a URIMAP definition, or you can combine it with a URIMAP definition. While migrating to the use of URIMAPs:
  1. You can introduce URIMAP resource definitions progressively for a small number of requests at a time. Depending on the type of processing carried out by your analyzer program, and the type of application that handles the request, you can choose whether or not to continue using the analyzer program in the processing path for each request.
  2. You might prefer to select and publish new URLs for requests handled by URIMAP resource definitions, rather than retaining your existing URLs. When you are ready to discontinue the use of the old processing path for a request, you can set up a URIMAP definition to permanently redirect requests from the old URL to the new URL.
  3. Ensure that your analyzer program still contains basic handling procedures for unrecognized requests, even if it is no longer involved in the processing path for any requests. The analyzer program is still required on the TCPIPSERVICE definition, and receives requests in situations such as the end user mis-typing a URL.

- For application programs that do not use the EXEC CICS WEB API commands but produce an HTTP response in a COMMAREA, CICS Web support is not able to assist with assembling the message structure correctly, or to carry out its full range of checks on the response. To take advantage of all the available CICS Web support facilities, it is recommended that you plan to convert these applications to Web-aware application programs that use the WEB API commands.

- URIMAP resource definitions can be used to deliver the contents of a CICS document or HFS file as a static response, or to deliver a redirection response, without involving a user-written application program. You could consider using this mechanism, instead of an application program, for simple responses that do not involve dynamic processing.

- Check that code page conversion is operating in the most efficient way. With minor changes to your application, you can take advantage of new CICS Web support facilities to:
  – Avoid setting up and using a code page conversion table (DFHCNV) for CICS Web support.
  – Allow CICS to identify and use the Web client's character set for code page conversion, rather than specifying this yourself.
  – Use the local system default (LOCALCCSID system initialization parameter) to identify the application program's code page, rather than specifying this yourself.
  – Convert to and from the UTF-8 and UTF-16 character sets.

In some cases, making these changes could enable you to discontinue the use of an analyzer program.

Chapter 28. Migration planning for the XPLINK option with C and C++ programs

This chapter describes the changes you might have to make to your existing applications or configuration to exploit the XPLINK option with C and C++ programs.

To continue running your C and C++ programs without exploiting the XPLINK option of the compiler, no action is needed.

The following steps show you how to take advantage of the support that is now available for XPLINK:

* Ensure that your C or C++ program is reentrant, and threadsafe, or modify it so that it conforms to these standards, see Chapter 26, "Migration planning for threadsafe programming and the open transaction environment (OTE)," on page 103
* If your program uses the XPCFTCH or XPCTA exits, take note of the advice in "Global User exits and XPLink" in the *CICS Application Programming Guide* that:
    – CICS disregards any attempt by XPCFTCH to modify the entry point.
    – CICS disregards any attempt by XPCTA to define a resume address.

    This is because the batch Language Environment runtime used for XPLink programs does not give control to CICS when a program abends, but goes through its own abend handling. When control reaches CICS, the Language Environment enclave has terminated, so CICS is unable to honor an entry point address or a resume address. You must find other ways to manage such requirements, or conclude that this program is not a suitable candidate for XPLINK optimization. One possible solution is to write a Language Environment abnormal termination exit, as described in the chapter "Customizing user exits" in the *z/OS Language Environment Customization* manual.
* Recompile the program using the XPLINK compiler option.
* Update the concurrency attribute of the PROGRAM resource definition for this program, setting the value to threadsafe.

Chapter 29. Migration planning for Business Transaction Services (BTS)

This chapter describes what you should do when migrating your BTS environment to CICS TS 3.1.

Migrating the DFHLRQ data set

The local request queue data set is used to store pending BTS requests, such as timer requests, or requests to run activities. It is recoverable, and is used to ensure that, if CICS fails, no pending requests are lost.

Requests that CICS can execute immediately, such as requests to run activities, are stored on the data set only briefly. Requests that CICS cannot execute immediately, such as timer or unserviceable requests, might be stored for longer periods. When CICS has processed a request, the request is deleted from the data set.

If you have outstanding BTS activities for BTS processes within CICS, you should migrate the contents of your DFHLRQ data set as part of the migration. A utility such as IDCAMS COPY could be used to update the CICS TS 3.1 DFHLRQ data set with the contents of the DFHLRQ data set from your previous CICS release.

Be aware that even if you are not explicitly making use of BTS services in your applications, it is possible that they are being exploited by vendor code or IBM-supplied products executing within your CICS environment.

PTFs for earlier CICS releases modified the dynamic routing DSECT used for dynamic DPL and dynamic start requests. This caused the structure of DFHLRQ records to change. The PTF numbers are:

CICS TS 1.3
PTF UQ82768 (APAR PQ75814)

CICS TS 2.2
PTF UQ82632 (APAR PQ75834)

CICS TS 2.3
PTF UQ85555 (APAR PQ81378)

If you have one of these PTFs applied to your existing CICS system, the DFHLRQ records match the format of DFHLRQ records used in CICS TS 3.1, and so can be migrated using a utility such as IDCAMS COPY. However, if you do not have one of these PTFs applied to your existing CICS system, the DFHLRQ record format is **not** compatible with that used in CICS TS 3.1. In this case, you should complete your BTS workload prior to migration to CICS TS 3.1, and start with an empty DFHLRQ at that release.

Repository data sets

When a process is not executing under the control of the CICS business transaction services domain, its state and the states of its constituent activities are preserved by being written to a VSAM data set known as a repository.

To use BTS, you must define at least one BTS repository data set to MVS. You may decide to define more than one, assigning a different set of process-types to each.

One reason for doing this might be storage efficiency, for example, if some of your
process-types tend to produce longer records than others.

If you operate BTS in a sysplex, several CICS regions may share access to one or
more repository data sets. This enables requests for the processes and activities
stored on the data sets to be routed across the participating regions. As you
migrate your CICS releases, you may therefore still share older versions of
repository data sets. The expectation is that you would define and use different
repository data sets as and when you wanted to assign different sets of
process-types, rather than just because a CICS migration has occurred.

Part 3. Changes to CICSPlex SM

This part of the book contains information about migrating to CICS TS CICSPlexSM from an earlier release:

Chapter 30. Operations views changes

This chapter summarizes the changes to CICSPlex SM operations views.It contains the following topic:

- "New WUI operations views"
- "Changed operations views"

There are no new ISPF end user interface views in CICS Transaction Server for z/OS, Version 3 Release 1. See Chapter 36, "New Web User Interface starter set views," on page 137 for a complete list of new Web User Interface starter set views.

New WUI operations views

A number of new Web User Interface operations views have been introduced. To locate these views from the WUI main menu, click **CICS operations views** and follow the menu hyperlink in Table 29. There are no new ISPF end user interface views in CICS Transaction Server for z/OS, Version 3 Release 1.

Table 29. New Web User Interface operations view sets

Menu hyperlink	Associated resource
TCP/IP service operations views —> URI host	HOST
TCP/IP service operations views —> Pipeline	PIPELINE
TCP/IP service operations views —> URI map	URIMAP
TCP/IP service operations views —> URIMAP global statistics	URIMPGBL
TCP/IP service operations views —> Web service	WEBSERV

Table 30. New Web User Interface operations detailed view

View	Menu hyperlink	Associated resource
EYUSTARTTASK.DETAIL10	Active tasks —> Channel usage	TASK

Changed operations views

A number of operations views have changed. These changes have been made to both the ISPF end user interface view and the corresponding Web User Interface starter set view

Some views have added fields, others have changed the way they present information. (For example the fields might be presented in a different order, or some fields might have moved onto new screens.)

The views that have changed are shown in Table 31 on page 122.

Table 31. Changed operations views

End user interface view	Web User Interface starter set view	What has changed
CICSRGND	EYUSTARTCICSRGN	A new attribute is added: DEBUGTOOL
CONNECT	Connections operations views —> ISC and MRO connections	New statistics attributes added.
DOCTEMP	Document template operations views —> Document template	New attribute and CVDA HFSFILE
EJCOBEAD	EYUSTARTEJCOBEAN	New attributes added: BEANACTIVATE, BEANPASSIVAT, BEANCREATES, BEANREMOVES, BEANMETHCALL
EJCOSE	Enterprise Java component operations views —> CorbaServers	New attributes CIPHERS and NUMCIPHERS
EJCOSED	EYUSTARTEJCOSE	New attributes added: ENABLESTATUS, OBJACTIVATES, OBJSTORES, FAILACTIVATE
EJDJBEAD	EYUSTARTEJDJBEAN	New attributes added: BEANACTIVATE, BEANPASSIVAT, BEANCREATES, BEANREMOVES, BEANMETHCALL
JVMPOOLD	EYUSTARTJVMPOOL	New attributes added: JVMLVL0TRACE, JVMLVL1TRACE, JVMLVL2TRACE, JVMUSERTRACE, SJGCURRCACHE, SJGPEAKCACHE, SJGREQSCACHE.
PROGRAM	Program operations views —> Programs	HOTPOOLING attribute removed
PROGRAMD	EYUSTARTPROGRAM	New attributes JVMPROFILE, and PGRJUSECOUNT are added
TASK	Task operations views —> Active tasks	New attributes DSCHMDLY and MAXSTDLY
TCPIPS	EYUSTARTTCPIPS	New attributes are added: • PRIVACY
TCPIPS	TCP/IP service operations views —> TCP/IP service	New attributes CIPHERS and NUMCIPHERS
TCPIPGBL	TCP/IP service operations views —> TCP/IP global satus	New attributes CRLSERVER and SSLCACHE
WORKREQ	Task operations views —> Work request	Changed attribute WORKTYPE

Chapter 31. Monitor view changes

This chapter summarizes the changes to CICSPlex SM monitor views. It contains the folowing topics:

- "Monitor view removed"

Monitor view removed

The following Web User Interface monitor view has been removed:

- EYUSTARTMJOURNAL

You should review your applications and remove references to this obsolete resource.

Chapter 32. Business Application Services changes

This chapter summarizes the changes to Business Application Services (BAS) definition views. It contains the following topics:

- "New CICS resource definition (BAS) views"
- "Changed CICS resource definition (BAS) views"
- "Changed BAS definition objects" on page 126
- "Changes with RASGNDEF processing" on page 126

There are no new ISPF end user interface views in CICS Transaction Server for z/OS, Version 3 Release 1. See Chapter 36, "New Web User Interface starter set views," on page 137 for a complete list of new Web User Interface starter set views.

New CICS resource definition (BAS) views

A number of new Web User Interface CICS resource definition for business application services (BAS) view sets have been introduced. To access these views from the WUI main menu, click **Administration views —> Basic CICS resource administration views** and follow the menu hyperlink in Table 32.

There are no new ISPF end user interface views in CICS Transaction Server for z/OS, Version 3 Release 1.

Table 32. New CICS resource definition (BAS) view sets

Menu hyperlink	Associated resource
CICS resource definitions —> Pipeline definitions	PIPEDEF
CICS resource definitions in a resource group	RESINGRP
CICS resource definitions —> URI mapping definitions	URIMPDEF
CICS resource definitions —> Web service definitions	WEBSVDEF

Changed CICS resource definition (BAS) views

There are changes to existing CICS resource definition for business application services (BAS) views. These changes have been made to both the Web User Interface view and the corresponding ISPF end user interface view. To access the WUI views from the WUI main menu, click **Administration views —> Basic CICS resource administration views —> CICS resource definitions** and follow the menu hyperlink in Table 33.

Table 33. Changes to BAS views

WUI menu hyperlink	End user interface view	What has changed
Document template definitions	DOCDEF	New attribute and CVDA HFSFILE
Enterprise java corbaserver definitions	EJCODEF	New attribute CIPHERS

Table 33. Changes to BAS views (continued)

WUI menu hyperlink	End user interface view	What has changed
Program definitions	PROGDEF	HOTPOOLING attribute not valid from CICS TS 3.1.
TCP/IP service definitions	TCPDEF	New attribute CIPHERS
Terminal definitions	TERMDEF	CONSOLE attribute not valid from CICS TS 3.1.

Changed BAS definition objects

Table 34 shows the changes to existing BAS definition objects. These changes have been made to both the ISPF end user interface view and the corresponding Web User Interface starter set view.

Table 34. Changed BAS definition object

BAS object	Web User Interface starter set view	What has changed
EJCODEF	EYUSTARTEJCODEF	New attribute added: STATUS
TCPDEF	EYUSTARTTCPDEF	• Attribute changed: AUTHENTICATE • New attribute added: PRIVACY

Changes with RASGNDEF processing

The change described in this topic was implemented through APARs for the following versions of CICSPlex SM:

• CICSPlex SM Version 1.4: APAR PK15477
• CICSPlex SM Version 2.2: APAR PK17773
• CICSPlex SM Version 2.3: APAR PK17773
• CICSPlex SM Version 3.1: APAR PK17787

If the version of CICSPlex SM **to** which you are migrating has the PTF for that APAR applied, but the version of CICSPlex SM **from** which you are migrating does **not** have the PTF for that APAR applied, then changes could occur in the way PROGDEFs and TRANDEFs are installed through RASGNDEFs (resource assignment definitions).

If a PROGDEF or TRANDEF is automatically installed through a RASGNDEF that specifies a USAGE of REMOTE and a MODE of STAT, then with CICSPlex SM Version 3.1 and the PTF for APAR PK17787, the REMOTESYSTEM used when the PROGDEF or TRANDEF is installed in the target system will be the CICS system ID (SYSIDNT) of the related system. In previous versions of CICSPlex SM which did not have the equivalent PTF applied, if the PROGDEF or TRANDEF specified a REMOTESYSTEM, or the RASGNDEF override specified a REMOTESYSTEM, this would be used.

You need to ensure that all BAS definitions are updated to tolerate this change before migrating to CICSPlex SM Version 3.1 with the PTF for APAR PK17787.

Table 35 illustrates the differences in processing from previous versions of CICSPlex SM where the PTF for the APAR is not applied. CICA is the actual SYSIDNT of the target system. CICB is the actual SYSIDNT of the related system.

Table 35.

PROGDEF or TRANDEF REMOTE-SYSTEM	RASGNDEF USAGE	RASGNDEF MODE	RASGNDEF REMOTE-SYSTEM OVERRIDE	Target REMOTE-SYSTEM *without* **PTF** applied	Target REMOTE-SYSTEM *with* **PTF applied**
none	REMOTE	STAT	none	CICB	CICB
none	REMOTE	STAT	CICX	CICX	CICB (1)
CICZ	REMOTE	STAT	none	CICZ	CICB (1)
CICZ	REMOTE	STAT	CICX	CICX	CICB (1)
none	REMOTE	DYNAM	none	CICA	CICA (2)
none	REMOTE	DYNAM	CICX	CICX	CICX
CICZ	REMOTE	DYNAM	none	CICZ	CICZ
CICZ	REMOTE	DYNAM	CICX	CICX	CICX

Notes:

1. Note the difference from previous versions of CICSPlex SM where the PTF for the APAR is not applied.

2. CICSPlex SM BAS does not provide a value for this during install. CICS defaults to the target system's SYSIDNT.

Chapter 33. CICSPlex SM Problem determination changes

This chapter summarizes the changes to CICSPlex SM Problem determination. It contains the following topic:

- "MAS Sysdump / Trandump changes"

MAS Sysdump / Trandump changes

When the CICSPlex SM MAS agent starts, it will automatically add one CICS TRANDUMPCODE (TRANDUMP) entry for transaction dump code 'EYUN' and two SYSDUMPCODE (SYSDUMP) entries for system dump codes 'EYU0XZPT' and EYU0XZSD'.

These codes are primarily used for CICSPlex SM Web User Interface users who may want to use the ADD action from the EYUSTARTTRANDUMP and EYUSTARTSYSDUMP viewsets to add their own TRANDUMP or SYSDUMP entries.

Chapter 34. The CICSPlex SM API

This chapter summarizes changes to the CICSPlex SM API. It contains the following topics:

- "Programs that connect to a previous release of CICSPlex SM"
- "Member DFHEILID has moved"
- "Obsolete resource tables"
- "New resource tables" on page 132
- "Changed resource tables" on page 133

Programs that connect to a previous release of CICSPlex SM

CICSPlex SM API programs that use the CONNECT verb specifying a VERSION keyword for a previous release of CICSPlex SM can experience significant increases in both CPU consumption by the CMAS address space, and data space storage utilization by the Environment Services System Services (ESSS) address space.

API programs that specify a CRITERIA string to limit the size of a result set on a GET or PERFORM OBJECT request, or use the SPECIFY FILTER verb, can experience the increase in CMAS CPU and ESSS storage. Batch job runtimes might also increase.

You are not required to recompile your CICSPlex SM API programs when you migrate releases. However, if you do not recompile affected programs, the CMAS has to convert the records from the current release format, to the level specified on the VERSION keyword on the CONNECT verb. This transformation process is highly intensive for CPU and storage when the result set is very large, for example, 300K to 500K records. The increases are observed in most cases when a criteria string is used to filter the result set; for example, specifying a criteria for the PROGRAM object using the NAME key for a specific or generic program. In this scenario, CICSPlex SM has to retrieve all program objects and return them to the CMAS where the API is connected, transform the records to the version of the API, and then apply the filtering.

If you recompile your programs to specify the VERSION keyword to match the current release of CICSPlex SM, this conversion does not take place, and there is no significant increase in storage or CPU consumption.

Member DFHEILID has moved

Member DFHEILID has moved from the SDFHC370 library to the SDFHSAMP library. The change affects the COPYLINK step of the sample jobs supplied by CICS and CICSPlex SM to translate, compile, link-edit and install application programs. The sample job supplied by CICSPlex SM is supplied in member EYUEITDL of the SEYUPROC library. If you are using an older version of this supplied sample job, or if you have created your own job based on the supplied sample, ensure that you apply the change for DFHEILID to the COPYLINK step of the job.

Obsolete resource tables

The following resource tables have been removed because they are obsolete.

DSKJRNL
JOURNAL
JRNLDEF
JRNINGRP
SMFJRNL
TAPEJRNL
VOLUME
XTASK

New resource tables

Table 36 shows new resource tables.

Table 36. New resource tables

Resource table	What is it?
BATCHREP	Batched repository update job. (See note 1 below.)
BRFACIL	LINK3270 bridge facility
CLCACHE	Shared class cache
HOST	URI virtual host information
JVM	JVM summary
JVMPROF	JVM profiles
METANAME	Information about all CVDASs, CVDATs, and EYUDAs
METAPARM	Information about a parameter for an action
PARMAVA	Allowable values for a parameter
PIPEDEF	PIPELINE resource definition
PIPELINE	Processing nodes acting on a service request and response
RESINGRP	Resource definitions in a resource group
URIMAP	URI map information
URIMPDEF	URI map definition
URIMPGBL	URI map global statistics
WEBSERV	Run time environment of a program deployed in a web services setting
WEBSVDEF	Web service resource definition
WORKREQ	EJB work requests and associated transactions

Notes:

1. The new BATCHREP resource table allows you to submit batched repository updates to run in a specified CMAS using the CICSPlex SM API. The CMAS to which the utility connects must be at CICS TS 3.1 or later.

Changed resource tables

The following resource tables have been changed. Review these resource tables for possible impact on any real-time analysis (RTA) evaluation definitions (EVALDEF) or application programming interface (API) programs you may be using.

• CICSRGN	• EJCODEF	• TCPDEF
• CONNECT	• EJCOSE	• TCPIPGBL
• DOCDEF	• PROGDEF	• TCPIPS
• DOCTEMP	• PROGRAM	• TERMDEF
• DSPGBL	• TASK	• TSMODEL (see note 4)
		• WORKREQ

• ATTR	• EJDJBEAN	• TASK
• CICSRGN	• FILEDEF (see note 1)	• TCPDEF
• CONNECT	• JVMPOOL	• TCPIPGBL
• DOCDEF	• OBJACT (see note 2)	• TCPIPS
• DOCTEMP	• OBJECT	• TERMDEF
• DSPGBL	• PROGDEF	• TERMNL
• EJCOBEAN	• PROGRAM	• TSMODEL (see note 4)
• EJCODEF	• REQID (see note 3)	• WORKREQ
• EJCOSE		

Notes:

1. For the FILEDEF resource table the attributes RKP and SRCHM are no longer valid. You should review any CICSPlex SM API applications that use these attributes and make any corrections needed.

2. OBJACT base tables for GET, CREATE, SET, UPDATE and REMOVE are now returned when EXEC CPSM GETDEF is issued. Previously only actions applicable to EXEC CPSM PERFORM were returned as OBJACT base tables.

3. The datatype of the REQID base table attribute NAME has changed its data type from HEX to CHAR.

4. The TSMODEL base table attribute called DESCRIPTION in earlier releases, has been renamed to RSVRD1 in CICS TS 3.1.

Chapter 35. Changes to generic alert structures used by CICSPlex SM

This chapter describes a change to SNA generic alerts and resolutions as they are used by CICSPlex SM.

"Product Set ID" (X'10') MS common subvector is a "Product ID" (X'11') common subvector that identifies the product as IBM Software (X'04'). It contains a "Product Number" (X'08') Product ID subfield that identifies the product number. This has changed from 5695081 to 5655M15.

Chapter 36. New Web User Interface starter set views

This chapter summarizes the additions to CICSPlex SM Web User Interface starter set views.

The new views are listed in Table 37 on page 138.

Note:
All Web User Interface starter set views begin with the prefix EYUSTART. This prefix has been omitted from the following list in the interests of brevity.

Table 37. New Web User Interface views

ACTION	DB2TDEF	PRNINGRP
APACTV	DBCTLSS	PROCDEF
APSPEC	DOCDEF	PROFDEF
BRFACIL	DOCINGRP	PROGDEF
CLCACHE	DTRINGRP	PROINGRP
CMDMPAPS	EJCINGRP	PRTNDEF
CMDMSAPS	EJCODEF	RASGNDEF
CMTCMDEF	EJDINGRP	RASINDSC
CMTPMDEF	EJDJDEF	RASPROC
CONINGRP	ENQINGRP	RDSCPROC
CONNDEF	ENQMDEF	RESDESC
CPLEXDEF	EVALDEF	RESGROUP
CPLXCMAS	FENODDEF	RESINDSC
CRESAIMD	FEPOODEF	RQMDEF
CRESCONN	FEPRODEF	RQMINGRP
CRESDB2C	FETRGDEF	RTAACTV
CRESDB2E	FILEDEF	RTADEF
CRESDB2T	FILINGRP	RTAGROUP
CRESDOCT	FNOINGRP	RTAINAPS
CRESDSNM	FPOINGRP	RTAINGRP
CRESENQM	FPRINGRP	RTAINSPC
CRESFECO	FSEGDEF	RTASPEC
CRESFEND	FSGINGRP	SESINGRP
CRESFEPO	FTRINGRP	SESSDEF
CRESFETR	JRMINGRP	STAINGRP
CRESGLUE	JRNINGRP	STATDEF
CRESJRNL	JRNLDEF	SYSLINK
CRESJRNM	JRNMDEF	TCLINGRP
CRESMODE	JVM	TCPDEF
CRESPART	JVMPROF	TCPINGRP
CRESPRGM	LNKSMSCG	TDQDEF
CRESPROF	LNKSMSCS	TDQINGRP
CRESPRTY	LNKSRSCG	TERMDEF
CRESRQMD	LNKSRSCS	TRANDEF
CRESSDMP	LNKSWSCG	TRANGRP
CRESTCPS	LNKSWSCS	TRMINGRP
CRESTDMP	LSRDEF	TRNCLDEF
CRESTDQ	LSRINGRP	TRNINGRP
CRESTERM	MAPDEF	TSMDEF
CRESTRUE	MAPINGRP	TSMINGRP
CRESTSMD	MONDEF	TYPINGRP
CSGLCGCG	MONGROUP	TYPTMDEF
CSGLCGCS	MONINGRP	WLMATGRP
CSYSDEF	MONINSPC	WLMAWDEF
CSYSGRP	MONSPEC	WLMDEF
D2CINGRP	PARINGRP	WLMGROUP
D2EINGRP	PARTDEF	WLMINGRP
D2TINGRP	PERIODEF	WLMINSPC
DB2CDEF	PGMINGRP	WLMSPEC
DB2EDEF	PRCINGRP	WORKREQ

Chapter 37. Web User Interface security changes

In CICS Transaction Server for z/OS, Version 3 Release 1 you can use your external security manager to control user access to views, menus, help information and the View Editor. In order to do this you need to create an approptiate profile in the FACILITY class. The following ESM FACILITY profiles are available:

EYUWUI.`wui_server_applid`.VIEW.`viewsetname`
— used to protect view sets.

EYUWUI.`wui_server_applid`.MENU.`menuname`
— used to protect menus

EYUWUI.`wui_server_applid`.HELP.`helpmembername`
— used to protect help pages.

EYUWUI.`wui_server_applid`.EDITOR.
— used to protect the View Editor..

where `wui_server_applid` is the CICS APPLID of the server.

Users can be given read or update access to views and menus:
- Read — to use the views or menus in the main interface; that is to allow you to prepare and protect views for specific user groups.
- Update — to create, update or remove items in the view editor or import using COVC; that is, allow you to open the view editor to more users, while restricting the view-sets and menus that individuals can modify.

If the ESM that you are using, neither grants nor refuses access to a profile (for example, if no RACF profile is defined), all users who are successfully signed on to the Web User Interface have access to the resources. You can make `not authorized` the default by setting up a generic profile.

Note: This security is designed to protect the views and menus themselves and not the objects they manage, which is covered by normal CICSPlex SM security.

Chapter 38. System initialization parameter changes for CICSPlex SM

The following CICS System Initialization Table parameter value has changed:

WRKAREA=2048
>   The size of the Common Work Area has increased to 2048 bytes.

For a complete list of CICS system initialization parameters for a CMAS, see the *CICS Transaction Server for z/OS Installation Guide*.

Chapter 39. Changes to CMAS journalling

Three new fields have been added to the EYUBCPJR DSECT:

- Two new fields have been added to CPJR_DEF_DATA:
  CPJR_DEF_MAJORVR
  Major version
  CPJR_DEF_MINORVR
  Minor version

- A new field has been added to CPJR_OPS_DATA:

  CPJR_ACTION
  Name of action

  CPJR_ACTION replaces CPJR_OPSTYPE which has been removed.

The EYUBCPJR DSECT is described in the *CICS Transaction Server for z/OS Installation Guide*.

Chapter 40. Migrating to CICS TS 3.1 CICSPlex SM

This chapter presents information about the compatibility of previous releases of CICSPlex SM and CICS TS Version 3.1CICSPlex SM.

> **Notes on terminology**
>
> Throughout this chapter, CICSPlex SM releases are referred to as follows:
>
> **Version 3.1**
> > Version 3 Release 1 (the CICSPlex SM element of CICS Transaction Server for z/OS, Version 3 Release 1)
>
> **Version 2.3**
> > Version 2 Release 3 (the CICSPlex SM element of CICS Transaction Server for z/OS, Version 2 Release 3)
>
> **Version 2.2**
> > Version 2 Release 2 (the CICSPlex SM element of CICS Transaction Server for z/OS, Version 2 Release 2)
>
> **Release 4**
> > Version 1 Release 4 (the CICSPlex SM element of CICS Transaction Server for OS/390 Release 3)
>
> The CICSPlex SM elements of CICS Transaction Server for z/OS Version 3 Release 1, CICS Transaction Server for z/OS, Version 2 Release 3 and CICS Transaction Server for z/OS, Version 2 Release 2 are **not** available as separate products.

Topics in this chapter describe:
- "Running CICSPlex SM Version 3.1 and an earlier release concurrently"
- "Conditions for running CICSPlex SM Version 3.1 and earlier releases concurrently" on page 146
- "Performing migration procedures" on page 147
- "A phased migration scenario" on page 154

MASs running the following CICS releases are directly-connectable to CICSPlex SM Version 3.1:

- CICS Transaction Server for z/OS Version 3 Release 1

- CICS Transaction Server for z/OS, Version 2 Release 3

- CICS Transaction Server for z/OS, Version 2 Release 2

- CICS Transaction Server for OS/390 Version 1 Release 3

- CICS for Windows component of IBM TXSeries 4.3.0 (with PTF 4)

- CICS for Windows component of IBM TXSeries 5.0

Running CICSPlex SM Version 3.1 and an earlier release concurrently

You can run CICSPlex SM Version 3.1, Version 2.3, Version 2.2, and Release 4 at the same time, with interconnected CMASs at different levels. The ability to do this allows gradual migration of the environment to Version 3.1. However, in CICS TS for z/OS, Version 3.1, a CICSPlex SM CMAS will run only in a CICS system at the same release level.

There are some conditions for running Version 3.1 with Release 4 that you should understand. Read the following section, "Conditions for running CICSPlex SM

Version 3.1 and earlier releases concurrently," then read "Performing migration procedures" on page 147 to understand how to migrate your supported releases before attempting the extra migration steps.

Removal of support for Windows remote MAS

Previous releases of CICSPlex SM have supported the CICS for Windows component of TXSeries, Version 4.3.0.4 and TXSeries, Version 5.0 (also known as NT 4.3 and NT 5.0) in the management of a remote managed application system (RMAS). This support is no longer necessary and the CICSPlex SM TXSeries agent has been removed for CICS Transaction Server for z/OS, Version 3.1 and later releases. Therefore, it is no longer possible to set up a CICSPlex SM remote MAS agent for Windows.

You can continue to use the CICS Transaction Servers 2.3 or 2.2 for TXSeries support in CICSPlex SM.

Conditions for running CICSPlex SM Version 3.1 and earlier releases concurrently

The following conditions apply to environments in which CICSPlex SM Version 3.1 and earlier releases of CICSPlex SM are running concurrently:

- The APARs in the following list must be applied to Release 4:

  Apply to Release 4 only
  - PQ46169
  - PQ54384
  - PQ54004

  > **Latest information:**
  > The lists above were correct at the time of publication, but you should expect changes to be made as APARs are answered. The Preventive Service Planning section (3.2) of the *CICS Transaction Server for z/OS Program Directory* advises you to review the current PSP information for the most up-to-date details, and tells you how to obtain this information.

- If you intend to use CICS TS 1.3 or CICS TS 2.2 with CICS TS 3.1, you must install the following maintenance to those releases:

  For CICS TS 1.3
  install APAR PQ65166, PTF UQ71532

  For CICS TS 2.2
  install APAR PQ65168, PTF UQ71534

- In order for a CAS, a CMAS, and a MAS (including those MASs that act as Web User Interface servers), to communicate, they must all be running the same release of CICSPlex SM. That is:
  - A CMAS must be connected to a CAS running at the same release as the CMAS. You can access a CMAS directly only through a CAS running at the same release level. This is true both when the context is a CMAS and when the context is a CICSplex that is connected to the CMAS.
  - A MAS (including those MASs that act as Web User Interface servers) must be connected to a CMAS running at the same release of CICSPlex SM as the MAS.

- A CAS running at Version 3.1 cannot be connected to a CAS running at Release 4.
- A CMAS running at Version 3.1 can be connected to a CMAS running at Version 2.3, Version 2.2 or Release 4. However:
  - In a CICSplex that consists of CMASs at the Version 3.1 level and the Version 2.3, Version 2.2 or Release 4 level, the maintenance point CMAS must be at the Version 3.1 level. That is, when a CICSplex contains CMASs at more than one level, the first CMAS converted to Version 3.1 must be the maintenance point.
  - If you are using the API, EUI, or Web User Interface to manage MASs connected to a CMAS at an earlier release, you must ensure that the MASs are managed indirectly from the Version 3.1 CMAS. You must ensure that:
    - All API programs run so that they are connected to the Version 3.1 CMAS.

      Note: This is only required if the API program needs to access new fields or later level CICS systems. If the API program connects to a lower level CMAS, any resource tables which contain new or updated fields for the new release would not be returned to the API program connected to the lower release level CMAS.

    - All TSO EUI sessions connect to the Version 3.1CAS.

    - All Web User Interface servers connect to the Version 3.1 CMAS.
  - When multiple CMASs at different CICSPlex SM release levels are running on the same MVS image, you must run a CAS for each release of the CMASs running on that MVS image. CASs running at different CICSPlex SM release levels cannot communicate directly.
  - You cannot view resources of a CICS Version 3.1 region using a CMAS running at an earlier release.

Performing migration procedures

The migration from a previous release of CICSPlex SM to CICS TS Version 3.1 CICSPlex SM for a CMAS and all MASs (including those MASs that act as Web User Interface servers) that are connected to it, as well as for the CAS to which the CMAS is connected, should be completed before CICSPlex SM is restarted. When other CMASs at the previous release level are not migrated to this release, a separate CAS running at the previous release level must be provided to which the other CMASs can now connect. This is so that you can access the EUI at the other CMASs. The CAS running at the previous release level should only be used for administration of the CMAS-to-CMAS communications, for example using CMTCMDEF and CMTPMDEF, and not for normal operations or definition work.

Several skeleton post-installation members are distributed with CICSPlex SM. You should generate these post-installation members for use during the migration. (For information about generating the post-installation members, see the *CICS Transaction Server for z/OS Installation Guide*.)

To enable you to revert to the previous release of CICSPlex SM if you encounter problems during the migration to CICS TS Version 3.1 CICSPlex SM, you should take back-up copies of the previous release components such as JCL, CLISTs, CICS tables, CMAS data repositories, and WUI repositories before you start the migration process.

Note: You can use the procedures in this section to migrate from Version 2.3, Version 2.2 or Release 4 of CICSPlex SM to CICS TS Version 3.1 CICSPlex SM.

Converting a CAS to Version 3.1

In order to provide for concurrent previous release and Version 3.1 CASs you must create a separate Version 3.1 CAS environment.

To convert a CAS from Version 2.3, Version 2.2 or Release 4 to Version 3.1, complete these steps.

Note: If you have used the CICSPlex SM EUI for your existing level of CICSPlex SM, log off TSO completely and then log on again before you make any changes to definitions or access the EUI for your new level of CICSPlex SM. Do the same if you need to access the EUI for your existing level of CICSPlex SM after accessing the EUI for the new level. Logging off TSO ensures that your allocation is cleared properly.

- Review the IEASYSxx member in the SYS1.PARMLIB library. The NSYSLX value may need to be increased. (For information about the NSYSLX value for CICSPlex SM, see the *CICS Transaction Server for z/OS Installation Guide*.)

- Authorize the new Version 3.1 libraries. (For information about how to do this, see the *CICS Transaction Server for z/OS Installation Guide*.)

- (Only when running both a previous release and Version 3.1.) Define the VTAM requirements for the Version 3.1 CAS. You must perform the following steps:
  - Create a VTAM application definition
  - Update the configuration list
  - Activate the major nodes

  When the Version 3.1 CAS is going to communicate with another Version 3.1 CAS on a system that also is running multiple releases of CICSPlex SM, you must also define the cross-domain resources. (For information about performing these steps, see the *CICS Transaction Server for z/OS Installation Guide*.)

- Review the JCL in the EYUDEFDS member generated by the EYUISTAR job to ensure that the following steps were generated when the post-installation jobs were created:
      IPRMDEL
      IPRMALOC

  Then run the job to create a new BBIPARM parameter repository data set for the Version 3.1CAS.

  If you are running Version 2.3, Version 2.2 or Release 4, as well as Version 3.1, and your earlier releases currently share a single BBIPARM data set, your Version 3.1 CASs can share the same BBIPARM data set. (For information about using EYUDEFDS, see the *CICS Transaction Server for z/OS Installation Guide*.)

- Update your TSO sign-on procedure to use the Version 3.1 data sets. Use generated member EYUTSODS to allocate the libraries temporarily. (For information about updating your TSO sign-on procedure, see the *CICS Transaction Server for z/OS Installation Guide*.)

- Review the changes made to the CICSPlex SM global security parameters for Version 3.1. Make sure the BBACTDEF DD statement in the CAS startup procedure references a data set containing the BBMTSS member distributed with Version 3.1. (For information about this member, see the *CICS Transaction Server for z/OS Installation Guide*

- Use the JCL procedure in the EYUCAS member to start the CAS, verifying the SSID and the DD statements for the Version 3.1 data sets. If you are running both a previous release of CICSPlex SM and Version 3.1, the SSIDs for the two CASs must be different. (For information about the JCL in EYUCAS, see the *CICS Transaction Server for z/OS Installation Guide*.)

The Version 3.1 CAS is now ready for use.

Converting a CMAS to Version 3.1

You must migrate your CICSPlex SM CMAS to CICS TS Version 3.1 at the same time as you migrate the CICS system on which it runs. This is because in CICS Transaction Server for z/OS, Version 2 Release 3 a CICSPlex SM CMAS will run only in a CICS system at the same release level. During startup the CMAS checks the CICS release level and terminates with message EYUXL0142 if the release does not match.

To convert a CMAS to Version 3.1:

• Review the IEASYSxx member in the SYS1.PARMLIB library.

> **Note:** Some of the parameters in the IEASYSxx member may need to be modified when you are running both a previous release and Version 3.1 of CICSPlex SM, because an Environment Services System Services (ESSS)space will be started for each release. (For information about NSYSLX and the ESSS, see the *CICS Transaction Server for z/OS Installation Guide*.)

• Authorize the Version 3.1 libraries. (For information about how to do this, see the *CICS Transaction Server for z/OS Installation Guide*.)

• Update the MVS link list with the Version 3.1 library. (For information about this step, see the *CICS Transaction Server for z/OS Installation Guide*.)

• Update the CSD file with the Version 3.1 group of resource definitions and CICS startup group list. (For information about how to do this, see the *CICS Transaction Server for z/OS Installation Guide*.)

• Update the CICS SIT GRPLIST parameter to reference the CICSPlex SM Version 3.1 group list EYU310L0. (For information about the CICS SIT parameters, see the *CICS Transaction Server for z/OS Installation Guide*.)

• Reassemble the CICS resource definition table load modules. Reference the library containing the Version 3.1 resource entry copy books to update the resource tables with the newest entries. (For information about assembling the resource definition table load modules, see the *CICS Transaction Server for z/OS Installation Guide*.)

• Convert the data repository to Version 3.1. (For information about how to convert the data repository, see the *CICS Transaction Server for z/OS Installation Guide*.)

> **Note:** The conversion utility migrates the contents of the existing data repository to a newly allocated data repository. The existing data repository is not modified.

• Edit the JCL you use to start the CMAS, changing the CICSPlex SM library names and the DD statements to the Version 3.1 names. If you are running an earlier version of EYUCMAS, you should add the following DD statement:

```
//BBIPARM   DD  DISP=SHR,DSN=CICSTS31.CPSM.EYUIPRM
```

(For information about the CMAS startup JCL, see the *CICS Transaction Server for z/OS Installation Guide*.)

• Verify the CICSPlex SM system parameters referenced by the EYUPARM DD statement. Ensure that the SSID value is the same as the value used to start the CAS to which this CMAS connects. (For information about these parameters, see the *CICS Transaction Server for z/OS Installation Guide*.)

The CMAS is ready to be cold started.

When you have successfully migrated all your systems to CICSPlex SM Version 3.1 you can delete the previous release groups and group lists from each CMAS's CSD. (For information about how to do this, see "Deleting the previous release definitions from CSD files" on page 153.)

Converting a MAS to Version 3.1

To convert a MAS to Version 3.1, you need to do the following:

- Authorize the Version 3.1 libraries. (For information about doing this, see the *CICS Transaction Server for z/OS Installation Guide*.)
- Reassemble the CICS resource definition table load modules. Reference the library containing the Version 3.1 resource entry copy books to update the resource tables with the newest entries. (For information about updating the resource definition table load modules for a MAS, see the *CICS Transaction Server for z/OS Installation Guide*.)
- Update the CSD file with the Version 3.1 group of resource definitions. At Version 3.1, this group is EYU310G1. (For information about updating the CSD file for a MAS, see the *CICS Transaction Server for z/OS Installation Guide*.)
- If all MASs that use the same CSD and group list are being migrated at the same time, update the CSD to add the Version 3.1 group to, and remove the previous release group from the group list. However, if the same CSD and group list are being used by MASs that will run multiple releases of CICSPlex SM concurrently, create a new group list in the CSD by performing the following steps:
  - APPEND the old group list to a new group list name.
  - Update the CSD to add the Version 3.1 MAS group.
  - Add the Version 3.1 MAS group to the new group list.
  - Remove the previous release MAS group from the new group list.

For more information about adding a (new release) group to the group list in the CSD, see the *CICS Transaction Server for z/OS Installation Guide*.

To create a new group list in the CSD, use a statement of the following form as input to DFHCSDUP:

```
APPEND LIST(old_list) TO(new_list)
```

To remove a previous release group from a group list, use a statement of the following form as input to DFHCSDUP:

```
REMOVE LIST(new_list) GROUP(old_group)
```

where `new_list` is the group list used by the MAS and `old_group` is the previous release group to be removed. The `old_group` name depends on the type of MAS and whether CICSPlex SM code is used from the LPA. Table 38 lists the release group names for each environment.

Table 38. MAS CSD groups for previous releases of CICSPlex SM

Environment	Version 2.3 Group	Version 2.2 Group
Local MAS – USELPACOPY(NO)	EYU230G1	EYU220G1
Remote MAS – USELPACOPY(NO)	EYU230G2	EYU220G2
Local MAS – USELPACOPY(YES)	EYU230GB	EYU220GB
Remote MAS – USELPACOPY(YES)	EYU230GC	EYU220GC

If a new group list is created, the GRPLIST CICS system initialization parameter for the MAS, in the SIT or in startup overrides, should be changed to specify the name of the new group list.

- When previous release modules are in the link pack area (LPA), you must ensure the Version 3.1 modules are used in place of the previous release modules. (For information about how to do this, see the *CICS Transaction Server for z/OS Installation Guide*.)
- Edit the JCL used to start the MAS changing the previous release of CICSPlex SM library names to the Version 3.1 names. (For information about the MAS startup JCL, see the *CICS Transaction Server for z/OS Installation Guide*.)
- In the sequential data set or partitioned data set member identified by the CICS SYSIN statement, verify that the CICS system initialization table (SIT) parameter **EDSALIM** is included, and set it to a value of at least 50MB. 50MB is the minimum EDSALIM required to start the MAS agent for CICSPlex SM Version 2.3 and later versions.

The MAS is ready to be cold started.

When you have successfully migrated all your systems to CICSPlex SM Version 3.1 you can delete the previous release groups from each MAS's CSD. (For information about how to do this, see Deleting the previous release definitions from CSD files.)

Workload management

If you use the workload management functions of CICSPlex SM and you use your own version of the CICSPlex SM user-replaceable Workload Routing Action Module, EYU9WRAM, you must recompile and link-edit your version of EYU9WRAM using the Version 3.1 libraries. For information on how to do this, see the description of customizing the dynamic transaction routing program in the *CICSPlex System Manager Managing Workloads* manual.

If your application programs have been modified to make a call to EYU9XLOP using the EYUAWTRA commarea, they must also be recompiled and link-edited with the Version 3.1 libraries.

Application programming interface

CICSPlex SM API programs written to run in a previous release MAS can be run in a Version 3.1 MAS. You can either continue to access the data provided by the previous release or access the new data available from Version 3.1. For a discussion of the compatibility between releases of the API, see the *CICSPlex System Manager Application Programming Guide*.

Converting a Web User Interface Server to Version 3.1

Both the Web User Interface server and the CMAS that it connects to must be at the highest level of CICSPlex SM and CICS within the CICSplex. This means that both must be at the same level as the maintenance point CMAS.

Before you migrate a Web User Interface server, you must migrate the CMAS that it connects to. You must migrate the Web User Interface server before you migrate any other MASs. If the CMAS that the Web User Interface server connects to is not the maintenance point CMAS, you must migrate the maintenance point CMAS at the same time.

As the CICS system that acts as your Web User Interface server is a local MAS, all the considerations that apply to a local MAS also apply to a Web User Interface server.

To convert a Web User Interface server to Version 3.1 you should:

- Migrate the MAS that acts as your Web User Interface server.
- # Update the CSD file with the Version 3.1 Web User Interface group of definitions. At Version 3.1, this group is EYU310GW.
- Migrate the contents of the Web User Interface server repository (EYUWREP).

If you have Web User Interface servers connected to CMASs other than the maintenance point CMAS, which have many other MASs connected to them, you might not want to migrate the other MASs at the same time as the CMAS. In that case you might consider using the following phased migration path:

1. Define a new Version 3.1 CMAS on the same MVS image as the Web User Interface server.
2. Connect the Version 3.1 CMAS to the CICSplex that the Web User Interface server CMAS is connected to. (It will not become available for use until the maintenance point CMAS has been migrated. If you see message EYUCP0022E at this time, there is no need to take any action.)
3. Migrate the maintenance point CMAS to Version 3.1 and take down the Web User Interface server at the same time. Migrate the Web User Interface server to Version 3.1 and, when you restart it, connect it to the Version 3.1 CMAS. The Version 3.1 CMAS should now connect successfully to the Version 3.1 maintenance point CMAS.
4. Migrate the remaining MASs when required, and connect them to the Version 3.1 CMAS as you restart them.
5. When you have moved all the MASs to the Version 3.1 CMAS, you can remove the original CMAS.

Assuming you are running the latest CICSPlex SM 2.3 and 3.1 maintenance levels, you can convert one LPAR at a time from 2.3 to 3.1.
- A CICSPlex SM 3.1 WUI can only connect to a 3.1 CMAS.
- A CICSPlex SM 2.3 WUI can only connect to a 2.3 CMAS.
- A CICSPlex SM 2.3 WUI connected to a 2.3 CMAS can retrieve data from a MAS connected to a 3.1 CMAS (assuming it is not a resource type which is unique to CICS TS 3.1) if the CMAS participates in the management of the CICSplex.
- A CICSPlex SM 2.2 WUI can only connect to a 2.2 CMAS.
- A CICSPlex SM 2.2 WUI connected to a 2.2 CMAS can retrieve data from a MAS connected to a 3.1 CMAS (assuming it is not a resource type which is unique to CICS TS 3.1) if the CMAS participates in the management of the CICSplex.
- A CICSPlex SM 3.1 WUI connected to a 3.1 CMAS can retrieve data from any MAS connected to any CMAS if the CMAS participates in the management of the CICSplex.

Migrating the MAS and updating the Web User Interface CSD group

To migrate the MAS and update the Web User Interface CSD group you should follow the instructions for converting a MAS as described in "Converting a MAS to Version 3.1" on page 150.

You must also replace the CSD group EYU230GW with EYU310GW in the group
list used by the Web User Interface server or create a new group list containing
EYU310GW. (For CICS Transaction Server for z/OS Version 3 Release 1
EYU310GW is included in the CSD when the CSD file is updated with the Version
3.1 resource definitions, EYU964G1).

Migrating the contents of the Web User Interface server repository (EYUWREP)

InCICS Transaction Server for z/OS Version 3 Release 1 some internal Web User Interface repository record versions have been incremented to facilitate the new features in view definitions For this reason, if your existing Web User Interface repository contains customized view sets or menus, it is essential that you migrate your view set and menu definitions.

To migrate the Web User Interface server repository to the current version:

- Export your view set and menu definitions with your Web User Interface server still running at your current release. It is not necessary for the Web User Interface server to be connected to a CMAS to do this. For information about exporting definitions see the *CICSPlex System Manager Web User Interface Guide* .
- Create a new Web User Interface server repository using the JCL described in the *CICS Transaction Server for z/OS Installation Guide*.
- Start the Web User Interface server at the new release using the new Web User Interface server repository.
- Import the new starter set definitions (the supplied set of view set and menu definitions with names beginning EYUSTART).
- Import your previous release view set and menu definitions, specifying the SKIP option on the Duplicate Names field of the COVC panel. This prevents the new starter set definitions being overwritten by starter set definitions exported from a previous release.

    Note: If you do accidently overwrite the new starter set definitions with starter set definitions exported from a previous release, you can re-import the new starter set definitions specifying the OVERWRITE option on the Duplicate Names field of the COVC panel.

For information about importing view set and menu definitions see the *CICSPlex System Manager Web User Interface Guide* . For information about the starter set see the *CICSPlex System Manager Web User Interface Guide*.

You do not need to make any changes to existing customized views and menus you may have created but you can consider modifying or creating view sets to take into account the new attributes and resources.

Notes:

1. You can import into a CICS TS for z/OS, Version 3.1 Web User Interface server repository, Version 2.3 or Version 2.2 view set and menu definitions.
2. You can import view set and menu definitions exported by a CICS TS for z/OS, Version 3.1 Web User Interface server into a CICS Transaction Server for z/OS, Version 2 Release 3 Web User Interface server repository. However, any new attributes or resources introduced in CICS TS for z/OS, Version 3.1 are not accessible in the CICS Transaction Server for z/OS, Version 2 Release 3 Web User Interface server. You may wish to remove these attributes and view sets using the View Editor. For information about the View Editor see the *CICSPlex System Manager Web User Interface Guide*.

Deleting the previous release definitions from CSD files

When you have successfully migrated all your systems to CICSPlex SM Version 3.1, you can delete the Version 2.2 definitions from each CMAS's and MAS's CSD. This can be done by upgrading each CSD using module EYU9R220, which is supplied in CICSTS31.CPSM.SEYULOAD.

```
//CSDUP    EXEC PGM=DFHCSDUP
//STEPLIB  DD  DSN=cics.index.SDFHLOAD,DISP=SHR
//         DD  DSN=cpsm.index.SEYULOAD,DISP=SHR
//DFHCSD   DD  DSN=cics.dfhcsd,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
 UPGRADE USING(EYU9R220)
 /*
```

Figure 4. JCL to delete previous release groups and group lists from the CSD

When this JCL is run, EYU9R220 attempts to delete all Version 2.2 groups and group lists from the CSD. However, because not all of the items the job attempts to delete are actually defined in the CSD, DFHCSDUP gives a return code of 04. The DFHCSDUP SYSPRINT output lists those items that were deleted and those that were not found. For further information about updating the CSD, see the *CICS Transaction Server for z/OS Installation Guide*.

A phased migration scenario

Figure 5 on page 155, Figure 6 on page 156, Figure 7 on page 159, and Figure 8 on page 162 and the discussions that accompany them show a CICSPlex SM environment at an earlier release and the steps you would take to convert that environment to Version 3.1. Note that this scenario presents one way you might perform the migration; you might find another set of procedures to be more appropriate to your own environment.
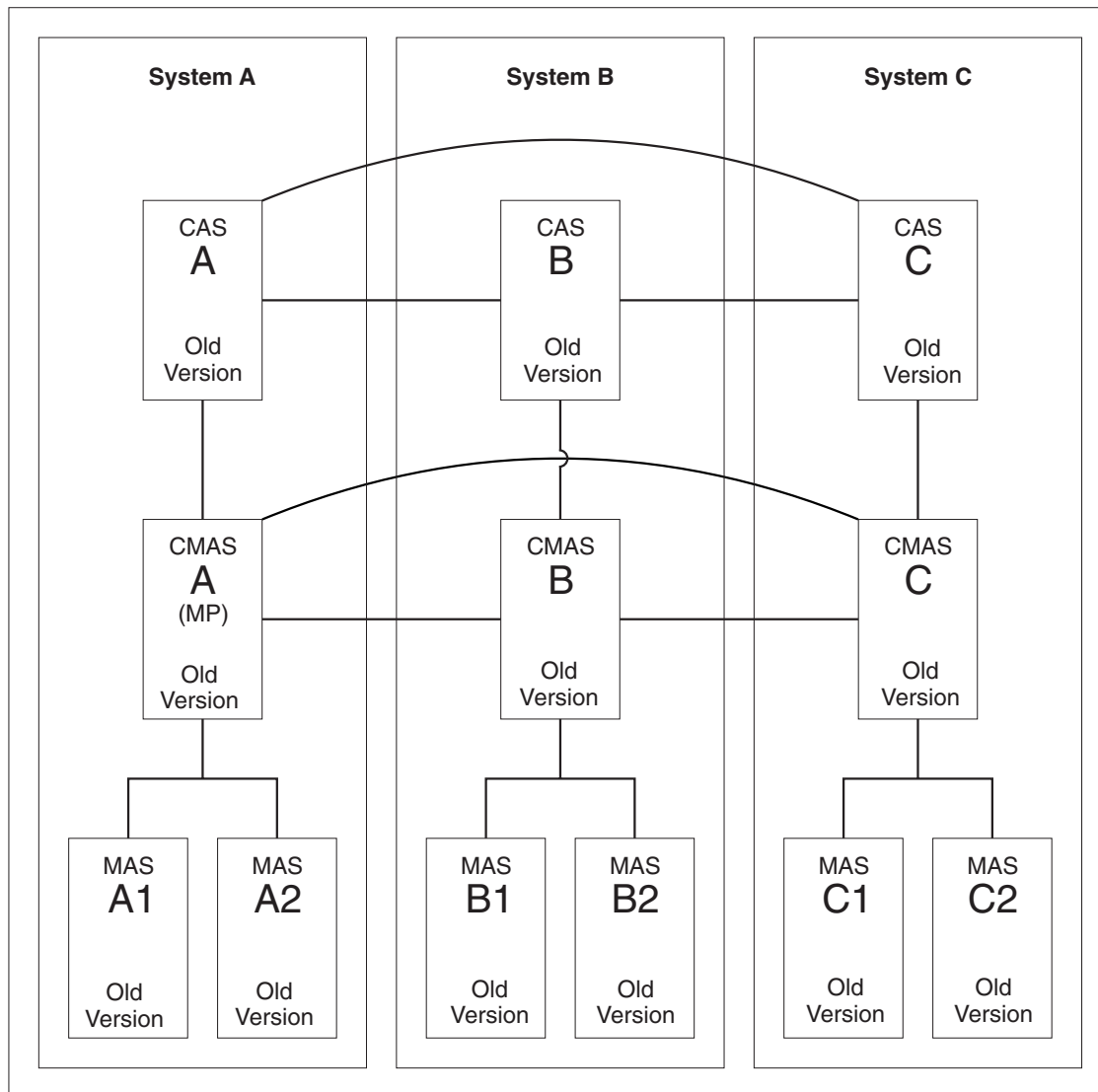
The environment

Figure 5. An environment at an earlier release

Figure 5 shows a CICSPlex SM environment that is made up of the following components:
- 3 MVS systems (System A, System B, System C)
- 3 CASs
    All interconnected
- 3 CMASs
    All interconnected
    CMAS A connects to CAS A (both are in System A)
        (This is the maintenance point CMAS.)
    CMAS B connects to CAS B (both are in System B)
    CMAS C connects to CAS C (both are in System C)
- 1 CICSplex
    CMAS A is the maintenance point
- 6 CICS regions
    6 local MASs

MAS A1 and MAS A2 connect to CMAS A (all are in System A)
MAS B1 and MAS B2 connect to CMAS B (all are in System B)
MAS C1 and MAS C2 connect to CMAS C (all are in System C)
Systems A, B and C are at the old CICS TS release.

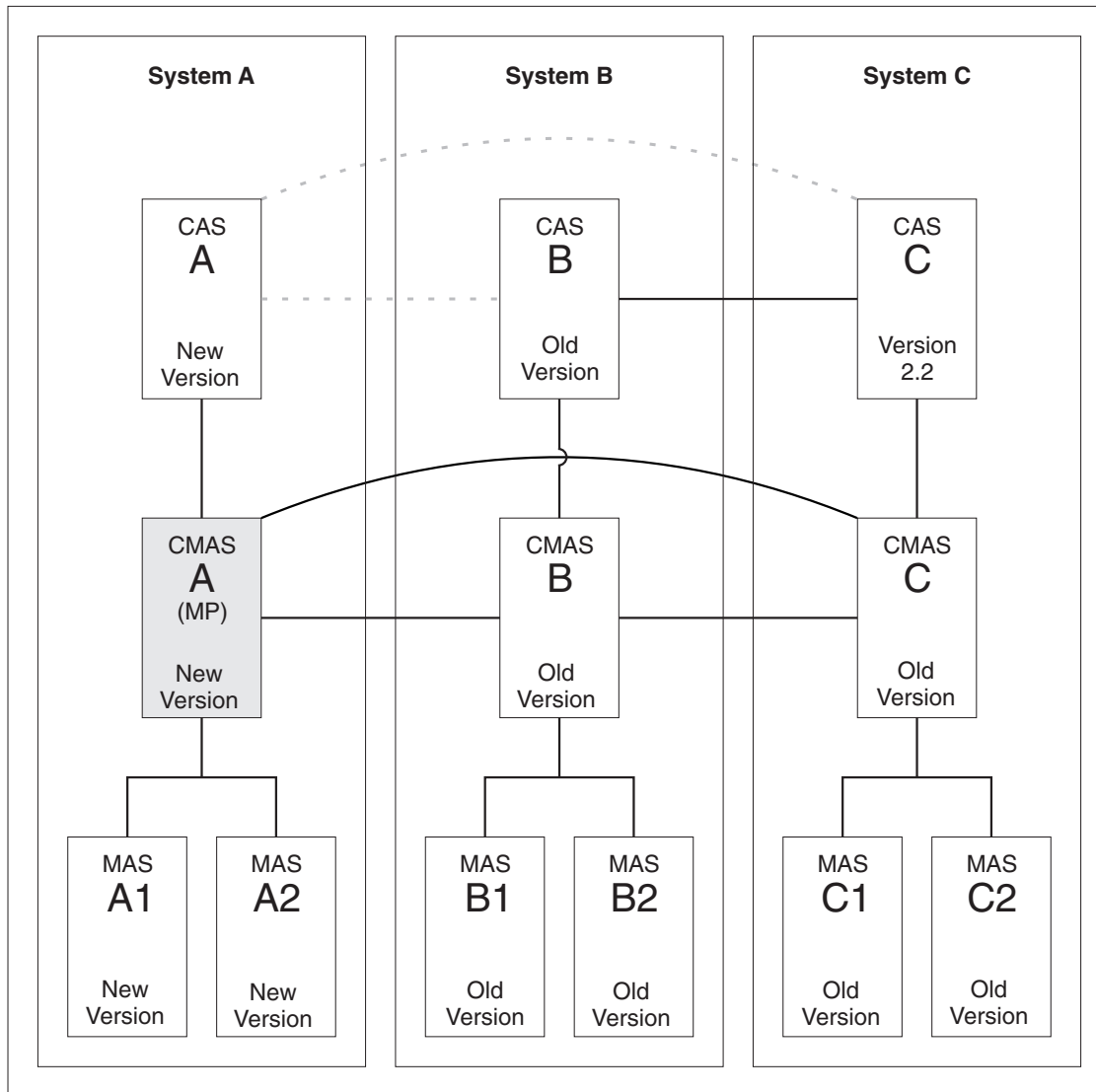## Objective 1: Convert MP CMAS to the new version

Figure 6. Converting the maintenance point CMAS to the new version

As shown in Figure 6, when you complete Objective 1 the connections from CAS A to CAS B and from CAS A to CAS C will be removed.

The conversion of the maintenance point CMAS A to the new version requires conversion for the following:
- MVS System A CICSPlex SM TSO users
- CAS A
- CMAS A
- MAS A1
- MAS A2

Step 1: Terminate executing regions that are to be converted

- If the following systems are in execution, terminate them:
  - CAS A
  - CMAS A
  - MAS A1
  - MAS A2

Step 2: Convert MVS System A CICSPlex SM TSO users to the new version

- Create the appropriate data set allocations to point to the new version data sets. This must not affect the allocations for TSO users on MVS Systems B and C, which are still at the previous level.

Step 3: Convert CAS A to the new version

- Ensure that CAS B and CAS C are started.
- Remove the CAS A links to CAS B and to CAS C. Depending upon whether the CASs share the BBIPARM data set, you must do one of the following:
  - If CAS B and CAS C share the BBIPARM data set:
    - From the CAS B CASDEF view:
      - Issue the EDIT action command to provide editing access to the BBIPARM data set.
      - Issue the DELete action command to delete the entry for CAS A.
      - Issue the SAVE command to save the changes in the BBIPARM data set.
  - If CAS B and CAS C do not share the BBIPARM data set:
    - From the CAS B CASDEF view, DELete the entry for CAS A, as described above.
    - From the CAS C CASDEF view, DELete the entry for CAS A, as described for the CAS B view.
- Change the appropriate IEAAPFxx member of the SYS1.PARMLIB library to authorize the CICSTS31.CPSM.SEYUAUTH library.
- Update the JCL used to start CAS A to point to the new data sets.

  Note: The BBIPARM data set *must not* contain a BBMTYB00 member. Member BBMTYB00 will be created dynamically when CAS A is first started. The new member will be reused when CAS A is subsequently restarted.

- Start CAS A.
- From the CAS A CASDEF view:
  - If communication out of the CICSPlex is required, issue the CHANGE action command to modify the VTAM APPL name for the current CAS (as indicated by a value of YES in the Cur Sys field). Because the value is **null** (or blank) before you do this, you may find it easier to think of the action as *creating* the VTAM APPL name.
  - If communication out of the CICSPlex is not required, leave the VTAM APPL name blank in order to use XCF communication. The XCF group name should be EYUGR310 (this value should be picked up automatically).
  - Issue the INStall action command to install the change.
  - Issue the SAVE action command to save the changes in the BBIPARM data set.

Step 4: Convert CMAS A to the new version

- Ensure that modules EYU9A310 and EYU9X310 in the CICSTS31.CPSM.SEYULINK data set are in the MVS link-list concatenation.

- Update the CMAS A CSD file, using the resource definitions supplied in the CICSTS31.CPSM.SEYULOAD data set.
- Update the CICS group list for CMAS A.
- Run EYU9XDUT to convert the EYUDREP data set for CMAS A to the new Version.

> **Note:** After converting the EYUDREP data set for CMAS A, the next time CMAS A is started it must point to the converted EYUDREP data set. If it does not, data repository updates may be lost. This can lead to invalid results, which can include other CMAS's isolating themselves when they connect to this CMAS.

- Update the JCL used to start CMAS A to point to the new data sets and to connect to the correct CASNAME.
- Ensure that the startup JCL for the CMAS includes the following statement:

```
//BBIPARM DD DISP=SHR,DSN=bbiparm.data.set
```

  The file should point to the same data set allocated to the BBIPARM file in the CAS to which this CMAS connects.
- Start CMAS A.

Step 5: Convert MAS A1 and MAS A2 to the new version
- Update the MAS A1 and MAS A2 CSD files, using the resource definitions supplied in the CICSTS31.CPSM.SEYULOAD data set.
- Update the CICS group lists for MAS A1 and MAS A2.
- Update the JCL used to start MAS A1 and for MAS A2 to point to the new data sets.
- Start MAS A1 and MAS A2.

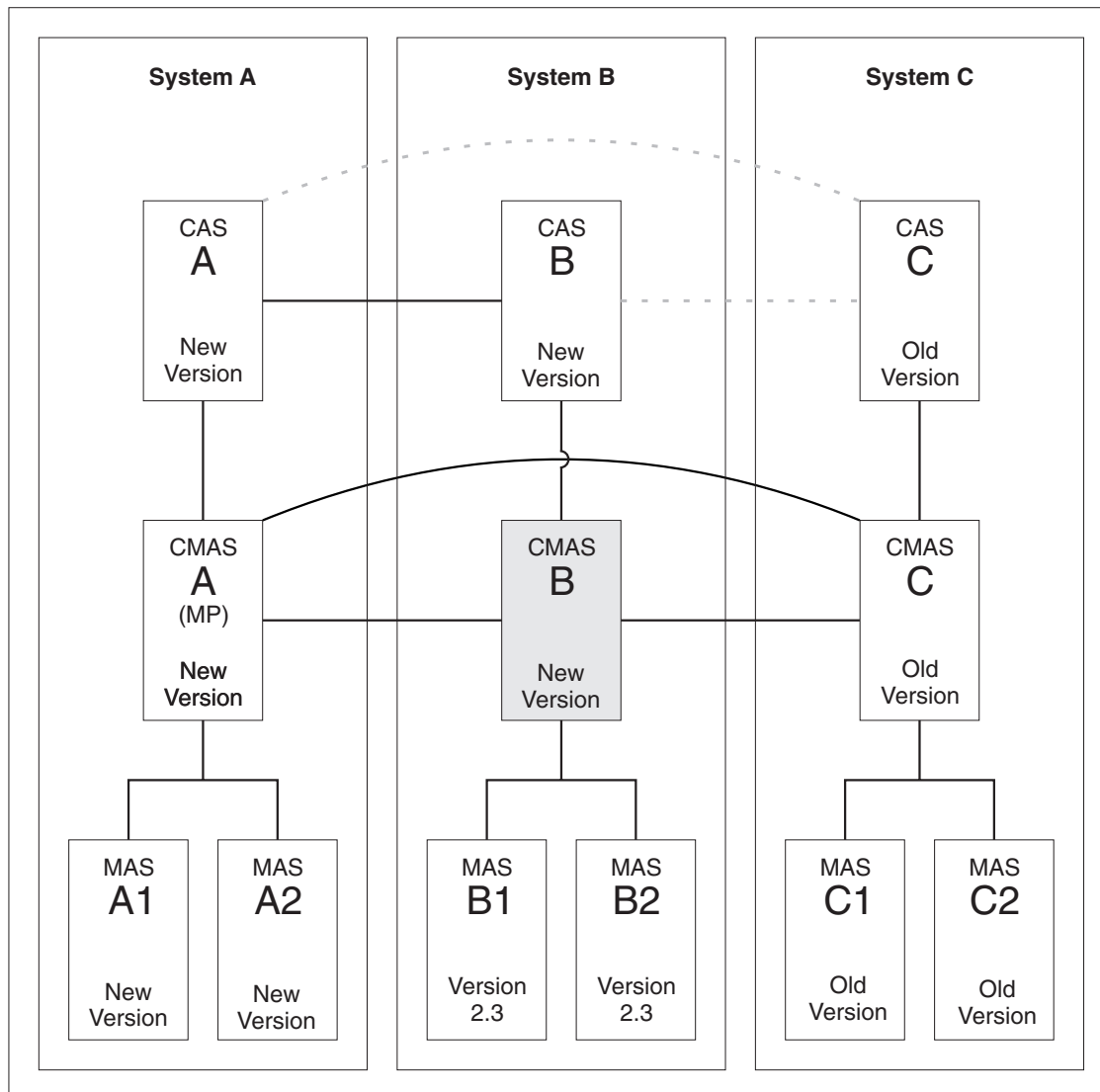# Objective 2: Convert CMAS B to the new version

Figure 7. Converting CMAS B to the new version

As shown in Figure 7, when you complete Objective 2 the connection from CAS A to CAS B will be reestablished and the connection from CAS B to CAS C will be removed. The conversion of CMAS B to the new version requires conversion for the following:

- MVS System B CICSPlex SM TSO users
- CAS B
- CMAS B
- MAS B1
- MAS B2

Step 1: Terminate executing regions that are to be converted

- IF THE FOLLOWING SYSTEMS ARE IN EXECUTION, TERMINATE THEM:
  - CAS B
  - CMAS B
  - MAS B1

– MAS B2

Step 2: Convert MVS System B CICSPlex SM TSO users to the new version

- Create the appropriate data set allocations to point to the new version data sets. This must not affect the allocations for TSO users on MVS System C, which is still at the previous level.

Step 3: Convert CAS B to the new version

- Ensure that CAS A and CAS C are started.
- Remove the link from CAS B to CAS C.
    - Bring up CAS C.
    - From the CAS C SYSTEMS view, DELete the entry for CAS B.
- Change the appropriate IEAAPFxx member of the SYS1.PARMLIB library to authorize the CICSTS31.CPSM.SEYUAUTH library.
- Update the JCL used to start CAS B to point to the new data sets.

    Note: If CAS A and CAS B are not going to share the BBIPARM data set, then the BBIPARM data set for CAS B *must not* contain a BBMTYB00 member. Member BBMTYB00 will be created dynamically when CAS B is first started. The new member will be reused when CAS B is subsequently restarted.

- Start CAS B.
- From the CAS B CASDEF view:
    - Issue the CHANGE action command to modify the VTAM ApplName for the current CAS (as indicated by a value of YES in the Cur Sys field). Because the value is **null** (or blank) before you do this, you may find it easier to think of the action as *creating* the VTAM APPL name.
    - Issue the INStall action command to install the change.
    - Issue the SAVE action command to save the changes in the BBIPARM data set.
- Link CAS B to CAS A. The procedure for doing this depends upon whether the CASs share the BBIPARM data set.

    If CAS A and CAS B share the BBIPARM data set:
    - From the CAS A CASDEF view, issue the INStall action to install the definition for B.

    If CAS A and CAS B do not share the BBIPARM data set:
    - From the CAS A CASDEF view:
        - Issue the ADD action command to add a definition for CAS B.
        - Issue the INStall action command to install the new definition.
        - Issue the SAVE action command to save the changes in the BBIPARM data set.
    - From the CAS B CASDEF view:
        - Issue the ADD action command to add a definition for CAS A.
        - Issue the INStall action command to install the new definition.
        - Issue the SAVE action command to save the changes in the BBIPARM data set.

Step 4: Convert CMAS B to the new version

- Ensure that modules EYU9A310 and EYU9X310 in the CICSTS31.CPSM.SEYULINK data set is in the MVS link-list concatenation.
- Update the CMAS B CSD file, using the resource definitions supplied in the CICSTS31.CPSM.SEYULOAD data set.
- Update the CICS group list for CMAS B.
- Run EYU9XDUT to convert the EYUDREP data set for CMAS B to the new version.

    Note: After converting EYUDREP data set for CMAS B, the next time CMAS B is started, it must point to the converted EYUDREP data set. If it does not, data repository updates may be lost. This can lead to invalid results, which include other CMAS's isolating themselves when they connect to this CMAS.

- Update the JCL used to start CMAS B to point to the new data sets.
- Ensure that the startup JCL for the CMAS includes the following statement:

    ```
    //BBIPARM DD DISP=SHR,DSN=bbiparm.data.set
    ```

    The file should point to the same data set allocated to the BBIPARM file in the CAS to which this CMAS connects.

- Start CMAS B.

Step 5: Convert MAS B1 and MAS B2 to the new version

- Update the MAS B1 and MAS B2 CSD files, using the resource definitions supplied in the CICSTS31.CPSM.SEYULOAD data set.
- Update the CICS group lists for MAS B1 and MAS B2.
- Update the JCL used to start MAS B1 and for MAS B2 to point to the new data sets and to connect to the correct CASNAME.
- Start MAS B1 and MAS B2.

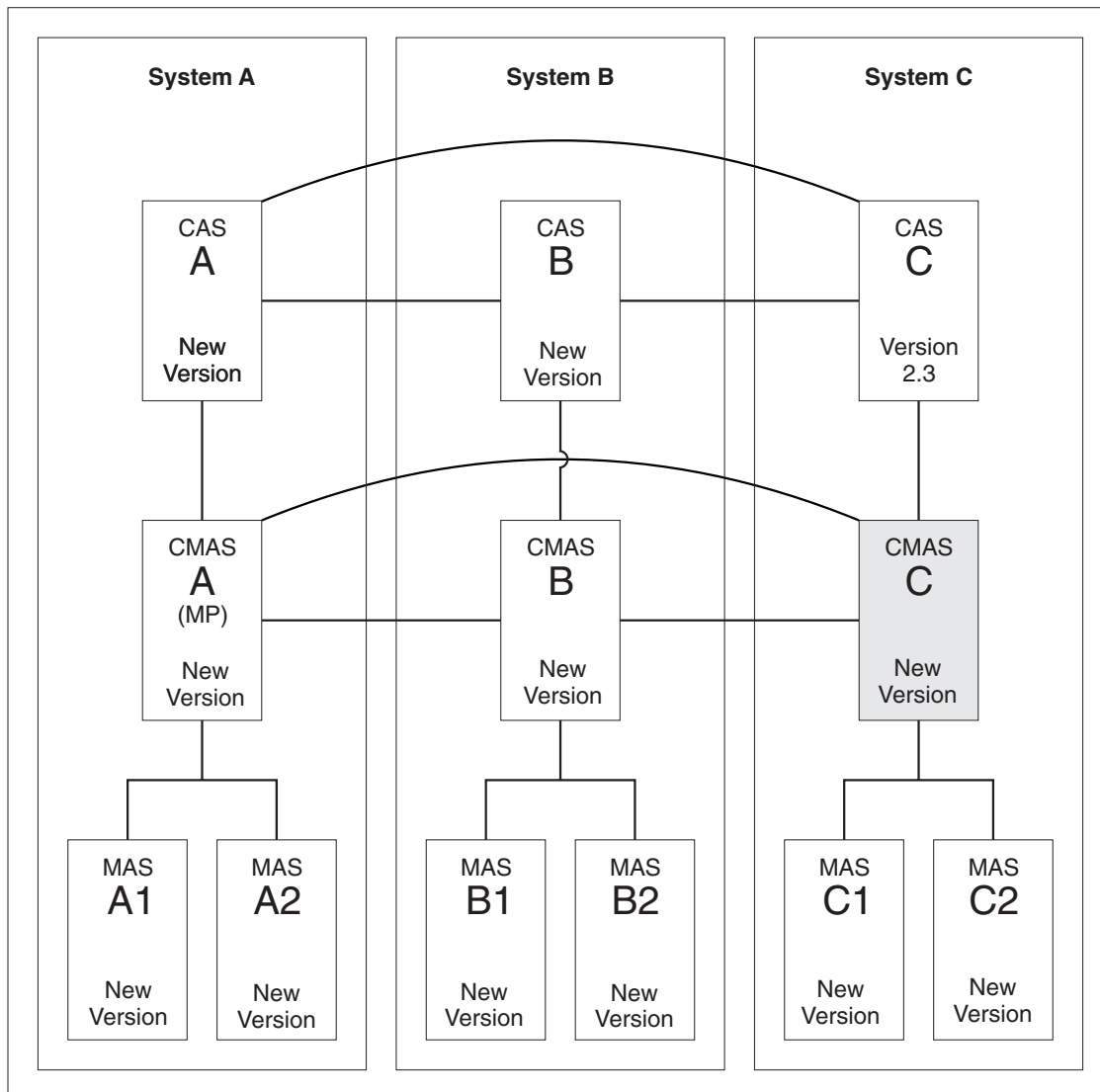# Objective 3: Convert CMAS C to the new version

Figure 8. Converting CMAS C to the new version

As shown in Figure 8, when you complete Objective 3 the connections from CAS C to CAS B and from CAS C to CAS A will be reestablished.

The conversion of CMAS C to the new version requires conversion for the following:
- MVS System C CICSPlex SM TSO user
- CAS C
- CMAS C
- MAS C1
- MAS C2

Step 1: Terminate executing regions that are to be converted
- If the following systems are in execution, terminate them:
  - CAS C
  - CMAS C
  - MAS C1

– MAS C2

Step 2: Convert MVS System C CICSPlex SM TSO users to the new version

- Create the appropriate data set allocations to point to the new data sets.

Step 3: Convert CAS C to the new version

- Ensure that CAS A and CAS B are started.
- Change the appropriate IEAAPFxx member of the SYS1.PARMLIB library to authorize the CICSTS31.CPSM.SEYUAUTH library.
- Update the JCL used to start CAS C to point to the new data sets.

  Note: If CAS C is not going to share the BBIPARM data set with CAS A and CAS B, then the BBIPARM data set for CAS C *must not* contain a BBMTYB00 member. Member BBMTYB00 will be created dynamically when CAS C is first started. The new member will be reused when CAS C is subsequently restarted.

- Start CAS C.
- From the CAS C CASDEF view:
  – Issue the CHANGE action command to modify the VTAM ApplName for the current CAS (as indicated by a value of YES in the Cur Sys field). Because the value is **null** (or blank) before you do this, you may find it easier to think of the action as *creating* the VTAM APPL name.
  – Issue the INStall action command to install the change.
  – Issue the SAVE action command to save the changes in the BBIPARM data set.
- Link CAS C to CAS A and to CAS B. The procedure for doing this depends upon whether the CASs share the BBIPARM data set.

  If the CASs share the BBIPARM data set:
  – From the CAS A CASDEF view, issue the INStall action to install the definition for CAS C.
  – From the CAS B CASDEF view, issue the INStall action to install the definition for CAS C.

  If the CASs do not share the BBIPARM data set:
  – From the CAS A CASDEF view:
    - Issue the ADD action command to add a definition for CAS C.
    - Issue the INStall action command to install the new definition.
    - Issue the SAVE action command to save the changes in the BBIPARM data set.
  – From the CAS B CASDEF view:
    - Issue the ADD action command to add a definition for CAS C.
    - Issue the INStall action command to install the new definition.
    - Issue the SAVE action command to save the changes in the BBIPARM data set.

- From the CAS C CASDEF view:
  - Issue the ADD action command to add a definition for CASs A and B.
  - Issue the INStall action command to install the new definitions.
  - Issue the SAVE action command to save the changes in the BBIPARM data set.

Step 4: Convert CMAS C to the new version

- Ensure that modules EYU9A310 and EYU9X310 in the CICSTS31.CPSM.SEYULINK data set are in the MVS link-list concatenation.
- Update the CMAS C CSD file, using the resource definitions supplied in the CICSTS31.CPSM.SEYULOAD data set.
- Update the CICS group list for CMAS C.
- Run EYU9XDUT to convert the EYUDREP data set for CMAS C to the new version.

  Note: After converting EYUDREP data set for CMAS C, the next time CMAS C is started, it must point to the converted EYUDREP data set. If it does not, data repository updates may be lost. This can lead to invalid results, can which include other CMAS's isolating themselves when they connect to this CMAS.

- Update the JCL used to start CMAS C to point to the new data sets and to connect to the correct CASNAME.
- Ensure that the startup JCL for the CMAS includes the following statement:

  ```
  //BBIPARM DD DISP=SHR,DSN=bbiparm.data.set
  ```

  The file should point to the same data set allocated to the BBIPARM file in the CAS to which this CMAS connects.
- Start CMAS C.

Step 5: Convert MAS C1 and MAS C2 to the new version

- Update the MAS C1 and MAS C2 CSD files, using the resource definitions supplied in the CICSTS31.CPSM.SEYULOAD data set.
- Update the CICS group lists for MAS C1 and MAS C2.
- Update the JCL used to start MAS C1 for MAS C2 to point to the new data sets.
- Start MASs C1 and C2.

Part 4. CICS messages and codes

This part of the book contains information about changes to CICS messages and abend codes:

- Chapter 41, "Messages and codes," on page 167

Chapter 41. Messages and codes

This chapter contains information about changes to CICS messages and abend codes.

- "Changes to messages and codes"

Changes to messages and codes

This section lists CICS messages and abend codes that are added, changed, or deleted.

See the *CICS Messages and Codes* manual for a full description of CICS messages.

- "New CICS messages"
- "Changed CICS messages" on page 179
- "Deleted messages" on page 179
- "New abend codes" on page 180
- "Deleted abend codes in CICS" on page 181

New CICS messages

In the following new messages, *condmsg* indicates that, where possible, a conditional message from the linked system is appended to this message.

Note: Many of these new messages are issued by CICS components with the following component codes:

DP Debug Tool Interface domain

PI Pipeline Manager domain

DFH5288 E GET-COMMAND TERMINATED AT USER'S REQUEST. RC=*xx*.

DFH5550 E *keyword1* **IMPLIES** *keyword2*. **THE DEFAULT VALUE HAS BEEN ASSUMED**

DFHAC2204 *time applid* **A commit failure has occurred during syncpoint processing for transaction** *tranid*. *condmsg*

DFHAC2205 *time applid* **A backout failure has occurred during syncpoint processing for transaction** *tranid*. *condmsg*

DFHAM4912 E *applid* **Install of** *resource resourcename* **failed because** *attribute* **is invalid for this release.**

DFHAM4928 E *applid* **Install of** *{TCPIPSERVICE |CORBASERVER} resourcename* **failed because the specified certificate is** *{expired | not yet current | not owned by this CICS | not trusted}*.

DFHAM4929 E *applid {URIMAP}(resourcename)* **was not installed because of conflicting attributes.**

DFHAM4930 E *applid* **URIMAP(***urimap1***) not installed because it maps the same URI as** *urimap2*.

DFHAM4931 E *applid* **The installation of WEBSERVICE** *resourcename* **failed because the associated** *{WSBIND file | PIPELINE}* **does not exist.**

DFHAM4932 E *applid* **The installation of** *{PIPELINE |WEBSERVICE} resourcename* **failed because the** *{hfsfile | PIPELINE}* **setup was not correct.**

DFHAM4933 E *applid* **The installation of PIPELINE** *resourcename* **failed because the WSDIR file specified is not accessible.**

DFHAP0801I *applid* **z/OS Conversion Services are not available.**

DFHAP0802 *applid* **Data conversion using CCSID** *ccsid1* **and CCSID** *ccsid2* **is not supported by this system.**

DFHAP0900 *applid* **MQ support for CICS Web Services is not available.**

DFHAP1206 *applid* **CEECCICS module not found.**

DFHAP1207 *applid* **CEEPIPI module not found.**

DFHAP1208 *applid* **Language Environment cannot support the Cobol language.**

DFHAP1209 *applid* **Language Environment cannot support the C/C++ languages.**

DFHAP1210 *applid* **Language Environment cannot support the PL/I language.**

DFHAP1211I *applid* **Language Environment initialization completed.**

DFHAP1218 *date time applid* **CEEPIPI function** *pipifn* **failed with return code** *r15rc*.

DFHAP1226 *date time applid* **Program** *program* **is defined as EXECKEY(USER) but transaction** *transaction* **as TASKDATAKEY(CICS): these attributes are incompatible.**

DFHBA0105 *date time applid terminal userid tranid* **processtype definition entry** *processtype* **has been installed.**

DFHBA0235 **The container record being processed is not complete.**

DFHCA5288 E **Get-command terminated at user's request. RC=***retcode*

DFHCA5550 W *date time applid netname tranid keyword1* **implies** *keyword2* **. The default value has been assumed.**

DFHCA5551 E *date time applid* **Command not executed.** *keyword1* **cannot be specified as generic unless** *keyword2* **is also generic.**

DFHCE3540 **Ensure that passwords are entered in the correct case.**

DFHCQ0001 *applid* **An abend (code** *aaa/bbbb***) has occurred at offset** *X'offset'* **in module** *modname***.**

DFHCQ0002 *applid* **A severe error (code** *X'code'***) has occurred in module** *modname***.**

DFHCQ0100I *applid* **Console queue initialization has started.**

DFHCQ0101I *applid* **Console queue initialization has ended.**

DFHCQ0102I *applid* **Console queue initialization has failed.**

DFHCQ0103I *applid* **MVS console queue is open.**

DFHCQ0104I *applid* **MVS console queue is closed.**

DFHCQ0105I *applid* **CICS is busy. MVS modify command has been rejected.**

DFHCQ0200I *applid* **CEKL transaction enabled.**

DFHCQ0201I *applid* **CEKL transaction enabled only for INQUIRE.**

DFHCQ0210I *applid* **CEKL command ignored; INQUIRE or SET keyword must be specified.**

DFHCQ0211I *applid* **CEKL command ignored; input expected.**

DFHCQ0212I *applid* **CEKL command ignored; input beginning** *input* **is too long.**

DFHCQ0213I *applid* **CEKL command ignored. Keyword,** *keyword* **is repeated.**

DFHCQ0214I *applid* **CEKL command ignored. Keyword,** *invkwd* **is not supported.**

DFHCQ0215I *applid* **CEKL command ignored; Keyword,** *invkwd* **is ambiguous.**

DFHCQ0216I *applid* **CEKL command ignored;** *invval* **is non-numeric.**

DFHCQ0217I *applid* **CEKL command ignored;** *invkwd* **is out of range.**

DFHCQ0218I *applid* **CEKL command ignored;** *delim* **expected after** *kywd***.**

DFHCQ0220I *applid* **CEKL INQUIRE command ignored; TASK keyword must be specified.**

DFHCQ0221I *applid* **CEKL INQUIRE command ignored; keyword expected.**

DFHCQ0224I *applid* **CEKL INQUIRE command ignored; keywords conflict.**

DFHCQ0230I *applid* **CEKL SET command ignored; TASK keyword must be specified.**

DFHCQ0231I *applid* **CEKL SET command ignored; keyword expected.**

DFHCQ0232I *applid* **CEKL SET command ignored; task number must be specified.**

DFHCQ0233I *applid* **CEKL SET command ignored; PURGE or FORCEPURCE or KILL keyword must be specified.**

DFHCQ0234I *applid* **CEKL SET command ignored; keywords conflict.**

DFHCQ0240I *applid* **CEKL INQUIRE: task number** *taskno* **not found.**

DFHCQ0241I *applid* **CEKL INQUIRE: task number** *taskno***, found.**

DFHCQ0242I *applid* **CEKL INQUIRE: no tasks matched selection criteria.**

DFHCQ0243I *applid* **CEKL INQUIRE:** *notasks* **task(s) matched selection criteria.**

DFHCQ0250I *applid* **CEKL SET: task number** *taskno* **not found.**

DFHCQ0251I *applid* **CEKL SET: deferred PURGE request issued for task number** *taskno***.**

DFHCQ0252I *applid* **CEKL SET: PURGE request issued for task number** *taskno***.**

DFHCQ0253I *applid* **CEKL SET: PURGE request ignored; task number** *taskno* **is being purged.**

DFHCQ0254I *applid* **CEKL SET: PURGE request ignored; task number** *taskno* **is being forcepurged.**

DFHCQ0255I *applid* **CEKL SET: PURGE request ignored; task number** *taskno* **is being killed.**

DFHCQ0256I *applid* **CEKL SET: FORCEPURGE request issued for task number** *taskno***.**

DFHCQ0257I *applid* **CEKL SET: FORCEPURGE request ignored; task number** *taskno* **is being forcepurged.**

DFHCQ0258I *applid* **CEKL SET: FORCEPURGE request ignored; task number** *taskno* **is being killed.**

DFHCQ0259I *applid* **CEKL SET: KILL request issued for task number** *taskno***.**

DFHCQ0260I *applid* **CEKL SET: KILL request ignored; task number** *taskno* **is being killed.**

DFHCQ0261I *applid* **CEKL SET: PURGE request rejected for system task number** *taskno***.**

DFHCQ0262I *applid* **CEKL SET: FORCEPURGE request rejected for system task number** *taskno***.**

DFHCQ0263I *applid* **CEKL SET: KILL request rejected for system task number** *taskno***.**

DFHCQ0264I *applid* **CEKL SET: PURGE request rejected; task number** *taskno* **is not purgeable.**

DFHCQ0265I *applid* **CEKL SET: FORCEPURGE request ignored; task number** *taskno* **is being purged.**

DFHCQ0266I *applid* **CEKL SET: KILL request ignored; task number** *taskno* **is being purged.**

DFHCZ0380 *date time applid userid termid tranid program name* **An attempt to allocate storage in** *module* **failed.**

DFHCZ0381 *date time applid userid termid tranid program name* **JNI call 'GetByteArrayElements()' in** *module* **failed.**

DFHCZ0382 *date time applid userid termid tranid program name* **JNI call 'GetStringUTFChars()' in** *module* **failed.**

DFHCZ0383 *date time applid userid termid tranid program name* **JNI call 'NewObject()' in** *module* **failed.**

DFHCZ0384 *date time applid userid termid tranid program name* **JNI call 'GetMethodID()' in** *module* **failed.**

DFHCZ0385 *date time applid userid termid tranid program name* **JNI call 'FindClass()' in** *module* **failed.**

DFHCZ0386 *date time applid userid termid tranid program name* **JNI call 'GetFieldID()' in** *module* **failed.**

DFHDB2024I *date time applid* **The CICS-DB2 attachment is in standby. For resync purposes only, connection has been made to DB2 restart-light subsystem** *db2-id{ | group }db2-group*

DFHDP0001 *applid* **An abend (code** *aaa/bbbb***) has occurred at offset** *X'offset'* **in module** *modname***.**

DFHDP0002 *applid* **A severe error (code** *X'code'***) has occurred in module** *modname***.**

DFHDP0100 *applid* **An unexpected exception response with reason code** *X'code'* **has been returned by CICS file control to module** *modname***. The debugging profiles file,** *filename***, is not usable.**

DFHDP0200 *applid* **Debug Tool is back level. LE Debugging profiles may be defined but not used on this system.**

DFHDP0300 *applid num* **debugging profile(s) have been inactivated.**

DFHDP0301 *applid* **The file DFHDPFMB is not available. No debugging profiles have been inactivated.**

DFHDS0010 *applid* **Kill request accepted for transaction id** *transid***, transaction number** *tranum***, userid** *userid***.**

DFHDS0011 *applid* **Kill request reaccepted for transaction id** *transid***, transaction number** *tranum***, userid** *userid***.**

DFHDU0217 *applid* **SDUMPX request completed with a return code of X'04'. A complete or partial dump has been taken.**

DFHEJ0761 I *date time applid userid* **CorbaServer** *CorbaServer_name* **has been set to be enabled.**

DFHEJ0762 I *date time applid userid* **CorbaServer** *CorbaServer_name* **has been set to be disabled.**

DFHEJ5043 E *date time applid userid* **An exception occurred processing DJar** *file_name* **. The following exception message may help to diagnose the problem:** *exception_message*

DFHEJ5044 E   *date time applid userid* **Error found in the deployment descriptor for DJar** *djar_name*.

DFHEJ5045 E   *date time applid userid* **Error found in the deployment descriptor for DJar** *djar_name*. **Duplicate element of type** *element_type* **found with name** *element_name* .

DFHEJ5046 E   *date time applid userid* **DJar** *djar_name* **is invalid.**

DFHEJ5047 E   *date time applid userid* **Error found in the deployment descriptor for DJar** *djar_name*. **An element of type** *element_type* **and value** *element_value* **references another element that cannot be found.**

DFHEJ5048 E   *date time applid userid* **Invalid Resource found in DJar** *djar_name*. **Class** *class_name* **for bean** *bean_name* **does not implement** *interface_name* .

DFHEJ5049 E   *date time applid userid* **Error found in the deployment descriptor for DJar** *djar_name*. **Bean** *bean_name* **implements the SessionSynchronization interface but has a n incompatible deployment descriptor.**

DFHEJ5050 E   *date time applid userid* **Error found in the deployment descriptor for DJar** *djar_name*. **An element of type** *element_type* **and value** *element_value* **for bean** *bean_name* **has an invalid value.**

DFHEJ5051 E   *date time applid userid* **DJar** *djar_name* **is not fully deployed. Class** *class_name* **cannot be found.**

DFHEJ5052 E   *date time applid userid* **Error found in the deployment descriptor for DJar** *djar_name*. **An element of type** *parent_element* **is missing a subelement of type** *missing_element*.

DFHEJ5053 E   *date time applid userid* **Missing Resource in DJar** *djar_name*. **Resource** *resource_name* **cannot be found.**

DFHEJ5054 E   *date time applid userid* **Error found in the deployment descriptor for DJar** *djar_name*. **An element of type** *element_type* **and value '***element_value***' is not a valid XML NMTOKEN.**

DFHEJ5055 E   *date time applid userid* **Error found in the deployment descriptor for DJar** *djar_name*. **No Session beans defined.**

DFHEJ5056 E   *date time applid userid* **Error found in the deployment descriptor for DJar** *djar_name*. **An element of type** *element_type* **and name** *element_name* **is missing a JNDI binding.**

DFHEJ5057 E   *date time applid userid* **Error found in the deployment descriptor for DJar** *djar_name*. **An unexpected element of type** *element_type* **was found.**

DFHEJ5058 E   *date time applid userid* **XML Parse failure in the deployment descriptor for DJar** *djar_name*. **Problem found at line** *line_number* **and column** *column_number*. **The XML parser returned the following exception message:** *XML_message* .

DFHEJ5059 W   *date time applid userid*. **One or more non-Session beans were found and ignored for DJar** *djar_name*.

DFHEJ5060 E   *date time applid userid* **A problem was found in the manifest file for DJar** *djar_name*. **The problem was found in the following manifest entry:** *attribute*.

DFHEJ5061 E   *date time applid userid* **DJar** *djar_name* **contains a bean whose name includes characters CICS canno t accept. The bean name is:** *bean_name* .

DFHEJ5062 E   *date time applid userid* **Error found in the deployment descriptor for DJar** *djar_name*. **Bean** *bean_name* **has EJB 2.0 local interfaces. These are not supported in CICS.**

DFHFC0175 *date time applid tranid trannum termid userid*. **Transactional VSAM unit of recovery** *X'urid'* **running in job** *jobname* **on Transactional VSAM instance** *TVSInstance* **in MVS** *mvsid* **holds** *{add to end lock | internal lock | exclusive lock on key | shared lock on key }X'keyid'* **in data set** *dsname* **causing** *{true | false}* **contention.**

DFHFC0177 *date time applid tranid termid userid*. **Transactional VSAM unit of recovery id** *X'urid'* **running in** *jobname/TVSInstance* **in MVS** *mvsid* **holds** *{add to end lock | internal lock | exclusive lock on key | shared lock on key }X'key1'* **on data set** *dsname1* **and is waiting for** *{add to end lock |internal lock | exclusive lock on key |shared lock on key }X'key2'* **on data set** *dsname2*.

DFHFC0179 *date time applid termid userid*. **Transaction** *tranid* **with transaction number** *trannum* **encountered an RLS retained lock held on data set** *dsname* **by unit of recovery** *X'urid'* **within Transactional VSAM instance** *TVSInstance*.

DFHII0110 E *date time applid client_ip_addr tcpipservice* **The request receiver received a request with no object key. Reason(**X'pointId'**).**

DFHII0216 E *date time applid client_ip_addr tcpipservice* **The request receiver asynchronous socket receive failed with an IO error.**

DFHII0227 E *date time applid client_ip_addr tcpipservice* **The request receiver socket has been closed. Replies outstanding:** *replies*. **Fragments in progress:** *fragments*

DFHII0228 E *date time applid client_ip_addr tcpipservice* **The request receiver socket has been closed.**

DFHII0229 E *date time applid client_ip_addr tcpipservice* **The request receiver received a GIOP fragment with no preceeding request for requestId:** *requestId*

DFHII0250 E *date time applid* **The request processor received an invalid reply fragment.**

DFHII0251 E *date time applid client_ip_addr tcpipservice* **The request receiver received a GIOP fragment whose length is not divisible by 8.**

DFHII0252 E *date time applid* **The request processor received a fragmented reply whose length is not divisible by 8.**

DFHII1050 W *date time applid* **Maximum version of GIOP has not been specified. Defaulting to GIOP 1.1 .**

DFHKE0414 BAD RETURN FROM MVS SERVICE CSRL16J.

DFHKE0500 *applid* **MAXPROCUSER exceeded while executing '**service-routine**'.**

DFHKE0501 *applid* **The Kernel received a return value of** *X'rvalue'*, **a return code of** *X'rcode'* **and a return reason of** *X'rreason'* **from the** *uss* **service-routine.**

DFHME0139 *applid* **(Module:**modname**) Message** *msgno* **has been suppressed by KILL processing.**

DFHOT0004 *applid* **A possible loop has been detected at offset** *X'offset'* **in module** *modname*.

DFHPI0001 *applid* **An abend (code** *aaa/bbbb*) **has occurred at offset** *X'offset'* **in module** *modname*.

DFHPI0002 *applid* **A severe error (code** *X'code'*) **has occurred in module** *modname*.

DFHPI0004 *applid* **A possible loop has been detected at offset** *X'offset'* **in module** *modname*.

DFHPI0110 *date time applid* **An attempt to start transaction CPIH by something other than an attach request from web domain has been made. This is not allowed.**

DFHPI0111   *date time applid tranid* **Call to Websphere MQ function** *function* **returned with reason code** *reason_code*. **Transaction abended.**

DFHPI0112   *date time applid tranid* **Unable to locate URIMAP to match HOST** *hostname* **and PATH** *pathname*. **Unable to process inbound MQ message.**

DFHPI0113   *date time applid tranid* **URIMAP** *urimapname* **has been located for HOST** *hostname* **and PATH** *pathname*, **but does not have USAGE(PIPELINE). Unable to process inbound MQ message.**

DFHPI0114   *date time applid tranid* **The pipeline MQ transport mechanism failed because a call to Websphere MQ function** *function* **returned with reason code** *reason_code*.

DFHPI0300   *date time applid* **CICS could not invoke WEBSERVICE** *WebService* **because it was unable to find container .**

DFHPI0301   *date time applid* **CICS was unable to link to program** *program_name* **while attempting to invoke webservice** *WebService*. *{The program abended. | The program was not defined. | The program was not enabled. | The program was not loadable. | No further details are available.}*

DFHPI0400   *date time applid tranid* **The CICS pipeline HTTP transport mechanism failed to send a request because** *{it was using an invalid host codepage | of a socket error}*.

DFHPI0401   *date time applid tranid* **The CICS pipeline HTTP transport mechanism failed to send a response or receive a request because** *{the codepage was not found | of a socket error | the connection was closed}*.

DFHPI0402   *date time applid tranid* **The CICS pipeline HTTP transport mechanism failed to send a request because the URI specified an unknown host:** *hostname*.

DFHPI0500   *date time applid tranid* **The CICS Pipeline Manager DFHPIPM encountered an error while trying to link to program** *program_name*. *{The program abended. | The program was not defined. | The program was not enabled. | The program was not loadable. | No further details are available.}* **PIPELINE:** *pipeline*.

DFHPI0501   *date time applid tranid* **The CICS Pipeline Manager cannot proceed as the pipeline is unusable.** *{The pipeline was not found. | The pipeline is disabled. | The pipeline was of the wrong type.}* **PIPELINE:** *pipeline*.

DFHPI0502   *date time applid tranid* **The CICS Pipeline Manager has failed to receive a request from the underlying transport. TRANSPORT:** *transport*, **PIPELINE:** *pipeline*.

DFHPI0503   *date time applid tranid* **The CICS Pipeline Manager has failed to send a response on the underlying transport. TRANSPORT:** *transport*, **PIPELINE:** *pipeline*.

DFHPI0504   *date time applid tranid* **The CICS Pipeline Manager has failed to communicate with a remote server due to an error in the underlying transport. TRANSPORT:** *transport*, **PIPELINE:** *pipeline*.

DFHPI0505   *date time applid tranid* **The CICS Pipeline Manager has failed to communicate with a remote server as no URI was provided. PIPELINE:** *pipeline*.

DFHPI0506   *date time applid tranid* **The CICS Pipeline Manager has failed to communicate with a remote server due to an invalid URI scheme being specified. URI:** *uri*, **PIPELINE:** *pipeline*.

DFHPI0507   *date time applid tranid* **The CICS Pipeline Manager has failed to receive a response from an application handling task.** *{The request timed out. | The application task abended. | The connection to the application task was closed.}* **PIPELINE:** *pipeline*.

DFHPI0508 E *date time applid* **The pipeline manager is unable to create or join a request stream because it is unable to reach the target for transaction** *tranid* **with userid** *userid***.**

DFHPI0509 E *date time applid* **The pipeline manager is unable to create or join a request stream because transaction** *tranid* **is not installed.**

DFHPI0510 E *date time applid* **The pipeline manager is unable to create or join a request stream because it has encountered a severe error for transaction** *tranid* **with userid** *userid***.**

DFHPI0600 *date time applid* **The CICS SOAP handler has been passed a container that is not DATATYPE(CHAR).**

DFHPI0601 *date time applid* **The CICS SOAP handler has been passed data that does not begin with a '<' character.**

DFHPI0700 S *date time applid userid* **PIPELINE** *pipeline* **PL/I support is required in order to use pipelines.**

DFHPI0701 I *date time applid userid* **PIPELINE** *pipeline* **has been created.**

DFHPI0702 E *date time applid userid* **PIPELINE** *pipeline* **encountered an error in the configuration file** *filename* **for pipeline at offset** *X'offset'***. The element name is** *elementname***.**

DFHPI0703 I *date time applid userid* **PIPELINE** *pipeline* **is about to scan the WSDIR directory.**

DFHPI0704 I *date time applid userid* **PIPELINE** *pipeline* **Implicit scan has completed. Number of wsbind files found in the WSDIR directory:** *num_files***. Number of successful webservice creates:** *num_ok***. Number of failed webservice creates:** *num_failed***.**

DFHPI0705 E *date time applid userid* **PIPELINE** *pipeline* **encountered an error writing the configuration to the der ived shelf** *derived-shelf***. The response code from the HFS write was** *X'uss-response'* **and the reason code was '***X'uss-reason'***'.**

DFHPI0706 E *date time applid userid* **PIPELINE** *pipeline* **resolution failed because it cannot be determined if this is a requester or provider pipeline.**

DFHPI0707 E *date time applid userid* **PIPELINE** *pipeline* **resolution failed because namespace prefixes are not supp orted in the XML configuration file.**

DFHPI0708 E *date time applid userid* **PIPELINE** *pipeline* **resolution failed because the XML configuration file cann ot be found.**

DFHPI0709 E *date time applid userid* **PIPELINE** *pipeline* **resolution failed because the XML configuration file cann ot be copied to the derived shelf.**

DFHPI0710 I *date time applid userid* **PIPELINE** *pipeline* **was successfully discarded.**

DFHPI0711 E *date time applid userid* **PIPELINE** *pipeline* **resolution failed because the SAX parser returned error c ode** *X'errcode'***. The error was at offset** *X'offset'* **in the CFGFILE. The first eight bytes of data at this offset are: '***hexdata'***.**

DFHPI0712 E *date time applid userid* **PIPELINE** *pipeline* **failed to install due to insufficient access rights to a HFS file.**

DFHPI0713 E *date time applid userid* **PIPELINE** *pipelinename* **The pipeline resolution transaction CPIR did not attach.**

DFHPI0714 E *date time applid userid* **PIPELINE** *pipeline* **failed to install. The directory specified in the WSDIR p arameter is invalid.**

DFHPI0715 I *date time applid userid* **PIPELINE** *pipeline* **Explicit scan has completed. Number of wsbind files found in the WSDIR directory:** *num_files***. Webservices successful create/update:** *num_ok***. Webservices not requiring update:** *num_nun***. Webservices failed create/update:** *num_failed***.**

DFHPI0716 E *date time applid userid* **Unable to dynamically create a webservice for PIPELINE** *pipeline*. **The complete WSBIND file name is too long.**

DFHPI0720 E *date time applid userid* **PIPELINE** *pipeline* **encountered an error in the configuration file** *filename* **at offset** *X'offset'*. **Found :** *element_found* **yet expected :** *element_expected*.

DFHPI0730 *date time applid* **An attempt to register a remote webservice as a participant in unit of work -** *X'uowid'* **has failed.**

DFHPI0800 E *date time applid userid* **Atomic Transaction processing failed because the SAX parser returned error code** *X'errcode'*. **The error was at offset** *X'offset'* **in the SOAP message.**

DFHPI0901 I *date time applid userid* **New WEBSERVICE** *WebService* **is being created during a scan against PIPELINE** *Pipeline*.

DFHPI0902 I *date time applid userid* **WEBSERVICE** *WebService* **is being updated during a scan against PIPELINE** *Pipeline*.

DFHPI0903 I *date time applid userid* **New URIMAP** *UriMap* **is being created during a scan against PIPELINE** *Pipeline* **for WEBSERVICE** *WebService*.

DFHPI0904 I *date time applid userid* **URIMAP** *UriMap* **could not be created for WEBSERVICE** *WebService* **in PIPELINE** *Pipeline*. **The URI that could not be allocated is: '** *Uri*'.

DFHPI0910 I *date time applid userid* **WEBSERVICE** *WebService* **within PIPELINE** *Pipeline* **has been created.**

DFHPI0911 E *date time applid userid* **WEBSERVICE** *WebService* **within PIPELINE** *Pipeline* **was not created because:** *{there is insuffient storage | there is a directory domain error | the specified PIPELINE is not installed | a lock cannot be obtained | there is a duplicate resource error}*.

DFHPI0912 I *date time applid userid* **WEBSERVICE** *WebService* **was successfully discarded.**

DFHPI0913 I *date time applid userid* **WEBSERVICE** *WebService* **is being discarded.**

DFHPI0914 E *date time applid userid* **WEBSERVICE** *WebService* **is UNUSABLE because:** *{the WSBind file was not found | CICS is not authorized to read the WSBind file | there is insufficient storage to load the WSBind file | the HFS read for the WSBind file failed | writing the WSBind file to the shelf failed | the PIPELINE is incompatible with this WEBSERVICE | the CPIR resolution transaction could not be attached | the direction of the PIPELINE can't be determined | the WSBind file is corrupt | the WSBind file has an invalid version number | the WSBind file has an out of date version number | the WSBind file product number was not recognised}*.

DFHPI0915 I *date time applid userid* **WEBSERVICE** *WebService* **is now INSERVICE and is ready for use.**

DFHPI0916 E *date time applid userid* **WEBSERVICE** *WebService* **within PIPELINE** *Pipeline* **was not created because it clashes with another WEBSERVICE of the same name in PIPELINE** *Pipeline*.

DFHPI0998 *date time applid* **The Outbound Router program, DFHPIRT, has received a non-NORMAL response while attempting to get the pipeline name from the DFHWS-PIPELINE container. The resulting error code is** *X'code'*.

DFHPI0999 *date time applid* **The Outbound Router program, DFHPIRT, has received an error from the Pipeline Manager.**

DFHPI1001 *date time applid* **Validation of a** *{request | response}* **message for webservice** *webservicename* **and operation** *operationname* **failed. The failure response contains the following message: '***message*'.

DFHPI1002	*date time applid* **Validation of a** *{request /* *response}* **message for webservice** *webservicename* **and operation** *operationname* **was successful.**
DFHPI1003	*date time applid* **No current channel located. Validation cannot occur.**
DFHPI1004	*date time applid* **The attempt to link to DFHPIVAL to perform validation failed.**
DFHPI1005	*date time applid* **Attempt to link to program** *progname* **failed. SOAP conversions cannot be performed. Module:** *modname*
DFHPI1006	*date time applid* **The WSBIND file used for WEBSERVICE** *WebService* **is not a type which CICS can use. Module:** *modname*
DFHRD0123 I	*date time applid terminal userid tranid* **INSTALL URIMAP(***urimap-name***)**
DFHRD0124 I	*date time applid terminal userid tranid* **INSTALL PIPELINE(***pipeline-name***)**
DFHRD0125 I	*date time applid terminal userid tranid* **INSTALL WEBSERVICE(***webservice-name***)**
DFHRM0315I	**AUTOCOLD OR AUTOINIT SHOULD NOT BE USED.**
DFHSJ0518	*date time applid* **An attempt to obtain the JNI extension interface pointer for a JVM, has failed. A CICS request to get a pointer to the Java Native Interface (JNI) Extension Interface for a Java Virtual Machine (JVM) has failed.**
DFHSJ0519	*date time applid* **The setting for environment variable** *env_var1* **conflicts with that for** *env_var2* **in JVM Profile** *JVMprof.* **The value of environment variable** *env_var3* **is assumed.**
DFHSJ0520	*date time applid* **The setting for environment variable** *env_var1* **in JVM Profile** *JVMprof* **is not valid for a Master JVM.**

DFHSJ0701	*date time applid* **Transaction** *transid* **can only be initiated by CICS.** *transid* **will terminate. Transaction** *transid* **is a CICS system transaction**
DFHSJ0702	*date time applid* **Transaction** *transid* **unsuccessfully attempted to create a JM TCB.** *transid* **will terminate. Transaction** *transid*
DFHSJ0703	*date time applid* **Transaction** *transid* **unsuccessfully attempted to change TCB mode.** *transid* **will terminate. Transaction** *transid*
DFHSJ0704	*date time applid* **Transaction** *transid* **unsuccessfully attempted to restore its TCB mode.** *transid* **will terminate. Transaction** *transid*
DFHSJ0705	*date time applid* **Transaction** *transid* **unsuccessfully attempted to delete a JM TCB.** *transid* **will terminate. Transaction** *transid*
DFHSJ0706	*date time applid* **During processing of transaction** *tranid,* **a call to CEEPIPI with function code INIT_SUB_DP has failed. (Return code was -** *X'rc'* **).**
DFHSJ0707	*date time applid* **During processing of transaction** *tranid,* **a call to CEEPIPI with function code CALL_SUB has failed. (Return code was -** *X'rc'* **, sub-routine return code was -** *X'subrc'* **).**
DFHSJ0708	*date time applid* **During processing of transaction** *tranid,* **a call to CEEPIPI with function code TERM has failed. (Return code was -** *X'rc'* **).**
DFHSJ0801	*date time applid* **An attempt to create a Java Virtual Machine using the JNI has failed.**
DFHSJ0802	*date time applid* **Attempt to load DLL** *dllname* **has failed. Runtime error message is** *errmsg*
DFHSJ0803	*date time applid* **Attempt to change the HFS working directory to** *pathname* **has failed. Runtime error message is** *errmsg*

DFHSJ0900 *date time applid* **Illegal Java version. CICS requires at least Java version** *min_ver* **but has found Java version** *current_ver* .

DFHSJ0901 I *date time applid* **Current version of Java is:** *current_ver* .

DFHSJ0902 *date time applid userid termid tranid program name* **Uncaught exception from application.**

DFHSJ0903 *date time applid userid termid tranid program name* **Exception** *exception* **occurred invoking main method in class** *className*.

DFHSJ0904 *date time applid userid termid tranid program name* **Exception** *exception* **occurred creating object reference for class** *className*.

DFHSJ0905 *date time applid userid termid tranid program name* **Class name** *className* **is invalid.**

DFHSJ0906 *date time applid* **The CICS Java Wrapper class failed to find the requested plugin** *plugin*.

DFHSJ0907 *date time applid tranidprogram name trannum userid termid* **The CICS Java Wrapper plugin** *plugin* **has thrown exception** *exception*.

DFHSM0137 *applid* **The amount of MVS storage available to CICS is low.**

DFHSM0138 *applid* **The amount of MVS storage available to CICS is no longer low.**

DFHSM0139 *applid* **The amount of MVS storage available to CICS is critically low.**

DFHSM0140 *applid* **The amount of MVS storage available to CICS is no longer critically low.**

DFHST0232 S **An abend (code** *aaa/bbbb*) **has occurred in extract exit program** *modname*. **Extract exit processing has been terminated.**

DFHST0233 I **DFHSTUP report suppressed. Report options requested only an EXTRACT userprogram to be invoked.**

DFHST0234 S **EXTRACT terminated at user's request RC=**retcode.

DFHST0235 S **An abend (code** *aaa/bbbb*) **has occurred in program** *modname*.

DFHTR2006 **UNKNOWN ENTRY PASSED FOR FORMATTING.**

DFHUS0070 *applid* **Security check for CICS region userid (**userid**) has failed. SAF codes are (**X'safresp',X'safreas'**). ESM codes are (**X'esmresp',X'esmreas'**). USAD reason code is (**reason**).**

DFHWB0152 E *date time applid client_ip_addr tcpipservice* **The request receiver SOCB notify gate is unable to attach transaction** *transaction*.

DFHWB0153 E *date time applid client_ip_addr tcpipservice* **The web asynchronous socket receive failed with an IO error.**

DFHWB0734 *date time applid tranid* **CICS Web attach processing failed because the SSL handshake with the client has failed. Host IP address:** *hostaddr*. **Client IP address:** *clientaddr*.{ | TCPIPSERVICE: }*tcpipservice* **A Secure Sockets Layer connection from a client with address**

DFHWB0736 *date time applid tranid* **The method in the received HTTP request is not implemented by the server. Host IP address:** *hostaddr*. **Client IP address:** *clientaddr*. **TCPIPSERVICE:** *tcpipservice*.

DFHWB0737 *date time applid tranid* **CICS Web support has detected that the version of the incoming HTTP request is higher than the version that CICS supports. Host IP address:** *hostaddr*. **Client IP address:** *clientaddr*. **TCPIPSERVICE:** *tcpipservice*.

DFHWB0738 *date time applid tranid* **CICS Web Support has detected that the incoming HTTP request has a version that is at least HTTP/1.1 but has no host header. Host IP address:** *hostaddr*. **Client IP address:** *clientaddr*. **TCPIPSERVICE:** *tcpipservice*.

DFHWB0739 *date time applid tranid* **An invalid EXPECT header has been received. Client IP address:** *clientaddr*. **TCPIPSERVICE:** *tcpipservice*

DFHWB0740 *date time applid tranid* **An HTTP/1.0 client has sent an EXPECT header which is not supported. Client IP address:** *clientaddr*. **TCPIPSERVICE:** *tcpipservice*

DFHWB0741 *date time applid tranid* **An HTTP socket receive request has timed out. Client IP address:** *clientaddr*. **TCPIPSERVICE:** *tcpipservice*

DFHWB0742 *date time applid tranid* **Conversion of HTTP header failed. Host IP address:** *hostaddr*. **Client IP address:** *clientaddr*.{ | *TCPIPSERVICE: }tcpipservice* **The conversion of the inbound HTTP header has failed and this**

DFHWB0743 *date time applid tranid* **The CICS Web characterset codepage is invalid. Host IP address:** *hostaddr*. **Client IP address:** *clientaddr*.{ | *TCPIPSERVICE: }tcpipservice* **The CICS Web characterset codepage is invalid and this has**

DFHWB0744 *date time applid tranid* **The CICS Web host codepage is invalid. Host IP address:** *hostaddr*. **Client IP address:** *clientaddr*.{ | *TCPIPSERVICE: }tcpipservice* **The CICS Web host codepage is invalid and this has**

DFHWB0745 *date time applid tranid* **Conversion of user data failed. Host IP address:** *hostaddr*. **Client IP address:** *clientaddr*.{ | *TCPIPSERVICE: }tcpipservice* **The conversion of the inbound user data has failed and this**

DFHWB0746 *date time applid tranid* **The maximum length of data that can be received has been exceeded. Client IP address:** *clientaddr*. **TCPIPSERVICE:** *tcpipservice*

DFHWB0747 *date time applid tranid* **A Content-Length and Transfer-Encoding conflict has been detected. Client IP address:** *clientaddr*. **TCPIPSERVICE:** *tcpipservice*

DFHWB0748 *date time applid tranid* **An invalid Chunk Size header has been received. Client IP address:** *clientaddr*. **TCPIPSERVICE:** *tcpipservice*

DFHWB0749 *date time applid tranid* **An invalid Trailer has been received. Client IP address:** *clientaddr*. **TCPIPSERVICE:** *tcpipservice*

DFHWB0750 *date time applid tranid* **HTTP warning request header received. Warning:** *warnvalue* **Host IP address:** *hostaddr* **Client IP address:** *clientaddr* **A warning header has been received on an incoming**

DFHWB0751 *date time applid tranid* **A precondition specified by an If-Unmodified-Since header has failed. Client IP address:** *clientaddr*. **TCPIPSERVICE:** *tcpipservice*.

DFHWB0752 *date time applid tranid* **HTTP Warning response header received. Warning:** *warnvalue* **Client IP address:** *clientaddr* **Server IP address:** *serveraddr*. **A warning header has been received in response to an**

DFHWB0753 *date time applid* **Transaction** *tranid* **chunked request incomplete. Session token:** *X'sesstoken'*. **The transaction has terminated with an incomplete chunked**

DFHWB0754 *date time applid tranid* **An invalid Chunk has been received. Client IP address:** *clientaddr*. **TCPIPSERVICE:** *tcpipservice*

DFHZC2119 E *date time applid termid tranid* **LUSTAT received on pipeline session incorrectly requests a definite response. ((***instance***) Module name: {DFHZRAC})**

DFHZC6908 I *date time applid* **Autoinstall in progress for netname** *netname*. **TN3270 IP address is** *tnaddr*.

DFHZN2200 *date time applid* **FREE IMPLICIT failed during syncpoint processing for session** *session id* **, remote system** *netname*. **Transid** *transid*. **The original abend code was** *abend code*.

DFH5551 E **COMMAND NOT EXECUTED.** *keyword1* **CANNOT BE SPECIFIED AS GENERIC UNLESS** *keyword2* **IS ALSO GENERIC.**

DFH5552 S **COMMAND NOT EXECUTED. CIPHER VALUE '***value***' IS NOT IN THE VALID SET (***list***).**

Changed CICS messages

The following is a list of the changed messages. Either the actual message text, or any of the supporting explanatory text, may have changed:

DFHAM4800	DFHAM4857	DFHDU1609
DFHAM4801	DFHAM4860	DFHEJ0755
DFHAM4805	DFHAM4861	DFHEJ1513
DFHAM4808	DFHAM4862	DFHEJ1521
DFHAM4809	DFHAM4863	DFHEJ5114
DFHAM4810	DFHAM4866	DFHFC0164
DFHAM4815	DFHAM4867	DFHFC0166
DFHAM4819	DFHAM4868	DFHFC0174
DFHAM4820	DFHAM4869	DFHII0108
DFHAM4830	DFHAM4875	DFHII0202
DFHAM4831	DFHAM4876	DFHII0212
DFHAM4840	DFHAM4877	DFHII0225
DFHAM4844	DFHAM4879	DFHII0601
DFHAM4845	DFHAM4881	DFHLG0760
DFHAM4846	DFHAM4882	DFHLG0772
DFHAM4847	DFHAM4883	DFHSJ0506
DFHAM4848	DFHAM4884	DFHSJ0508
DFHAM4849	DFHAM4888	DFHSO0123
DFHAM4852	DFHAM4891	DFHST0229
DFHAM4854	DFHAM4910	DFHST0230
DFHAM4855	DFHAP1213	DFHST0231
DFHAM4856	DFHBR0427	DFH5287E
DFHAM4834E	DFHDH0105	DFHTO6003E
DFHAM4889E		DFHTO6004E
DFHAM4920E		

Deleted messages

The following is a list of the messages deleted:

DFHAP1201	DFHAP1205	DFHII0605
DFHAP1202	DFHII0203	DFHII0606
DFHAP1204	DFHII0211	DFHSO0127

DFHAP1219	DFHAU2222	DFHAU5016
DFHAP1220	DFHAU2224	DFHAU5017
DFHAP1221	DFHAU2225	DFHAU5018
DFHAP1222	DFHAU2226	DFHAU5019
DFHAP1223	DFHAU2227	DFHAU5020
DFHAP1224	DFHAU2228	DFHAU5021
DFHAP1225	DFHAU2229	DFHAU5022
DFHAU2101	DFHAU2230	DFHAU5023
DFHAU2102	DFHAU2231	DFHAU5024
DFHAU2103	DFHAU2233	DFHAU5025
DFHAU2104	DFHAU2234	DFHAU5026
DFHAU2105	DFHAU2235	DFHAU5027
DFHAU2106	DFHAU3301	DFHAU5028
DFHAU2107	DFHAU3302	DFHAU5029
DFHAU2110	DFHAU3303	DFHAU5030
DFHAU2111	DFHAU3304	DFHAU5031
DFHAU2114	DFHAU3305	DFHAU5032
DFHAU2115	DFHAU3306	DFHAU5033
DFHAU2116	DFHAU3307	DFHAU5034
DFHAU2117	DFHAU3308	DFHAU5035
DFHAU2118	DFHAU3310	DFHAU5036
DFHAU2119	DFHAU3311	DFHAU5037
DFHAU2120	DFHAU3312	DFHAU5038
DFHAU2121	DFHAU3313	DFHAU5039
DFHAU2122	DFHAU3314	DFHAU5040
DFHAU2125	DFHAU3315	DFHAU5042
DFHAU2127	DFHAU4100	DFHAU5043
DFHAU2201	DFHAU4200	DFHCZ0150
DFHAU2202	DFHAU5000	DFHCZ0151
DFHAU2203	DFHAU5001	DFHCZ0152
DFHAU2204	DFHAU5002	DFHCZ0153
DFHAU2205	DFHAU5004	DFHCZ0154
DFHAU2206	DFHAU5005	DFHCZ0155
DFHAU2210	DFHAU5006	DFHCZ0156
DFHAU2211	DFHAU5007	DFHCZ0157
DFHAU2212	DFHAU5008	DFHCZ0158
DFHAU2214	DFHAU5008	DFHCZ0159
DFHAU2216	DFHAU5009	DFHSI1513
DFHAU2217	DFHAU5010	DFHSI1520
DFHAU2218	DFHAU5011	DFHZC6310
DFHAU2220	DFHAU5014	DFHZC6311
DFHAU2221	DFHAU5015	

New abend codes

The following is a list of the new abend codes added to CICS:

ABRX	ADPI	ALIH	ATDS
ACSK	AEZR	ALII	ATD3
ADPA	AEZS	ANQF	ATD9
ADPB	AEZT	ASJJ	AWBN
ADPC	AEZU	ASOD	AZIG
ADPD	AIID		

AALW	AKKE	APID	AWSH
AALX	AKKG	APIE	AWSL
AD3H	AKKH	APIF	AWSN
AEYF	ALIJ	APIG	AWSP
AEZV	ALIK	APIH	AWSQ
AEZW	ALX1	APII	AWSR
AEZX	ALX2	APIJ	AWSS
AIIP	ALX3	APIK	AWST
AITI	ALX4	APIL	AWSU
AKKA	ALX5	APIM	AXGA
AKKB	APIA	AWBP	AXTS
AKKC	APIB	AWSC	AXTU
AKKD	APIC		

Deleted abend codes in CICS

The following is a list of the abend codes deleted from CICS:

AEC3	AEC5	ALIB	
AEC4	AEC6	ALIE	
AJHA	AJH9	AUYF	AUZK
AJHB	ALIA	AUYG	AUZL
AJHC	APC4	AUYH	AUZN
AJHD	APC5	AUYI	AUZO
AJHE	APC6	AUYJ	AUZQ
AJHF	APC7	AUZA	AUZR
AJH0	APC8	AUZB	AUZS
AJH1	APC9	AUZC	AUZU
AJH2	AUXA	AUZD	AUZV
AJH3	AUXB	AUZF	AUZY
AJH4	AUYA	AUZH	AUZZ
AJH5	AUYC	AUZI	AUZ1
AJH8	AUYE	AUZJ	

The following is a list of abend codes which, if they exist in your system, are deleted from CICS for this release. (We say, "IF they exist", because these codes are added to earlier releases of CICS by a PTF which may or may not have been applied at the time you migrate to CICS TS 3.1).

AXF9
AXTT

Part 5. Requirements

The hardware and software requirements for CICS Transaction Server for z/OS, Version 3 Release 1 are described in the *CICS Transaction Server for z/OS Release Guide* . This includes information about pre-requisite program products that appeared here in the documentation for earlier releases of CICS.

Part 6. Appendixes

185

Bibliography

The CICS Transaction Server for z/OS library

The published information for CICS Transaction Server for z/OS is delivered in the following forms:

The CICS Transaction Server for z/OS Information Center
The CICS Transaction Server for z/OS Information Center is the primary source of user information for CICS Transaction Server. The Information Center contains:

- Information for CICS Transaction Server in HTML format.
- Licensed and unlicensed CICS Transaction Server books provided as Adobe Portable Document Format (PDF) files. You can use these files to print hardcopy of the books. For more information, see "PDF-only books."
- Information for related products in HTML format and PDF files.

One copy of the CICS Information Center, on a CD-ROM, is provided automatically with the product. Further copies can be ordered, at no additional charge, by specifying the Information Center feature number, 7014.

Licensed documentation is available only to licensees of the product. A version of the Information Center that contains only unlicensed information is available through the publications ordering system, order number SK3T-6945.

Entitlement hardcopy books
The following essential publications, in hardcopy form, are provided automatically with the product. For more information, see "The entitlement set."

The entitlement set

The entitlement set comprises the following hardcopy books, which are provided automatically when you order CICS Transaction Server for z/OS, Version 3 Release 1:

Memo to Licensees, GI10-2559
CICS Transaction Server for z/OS Program Directory, GI10-2586
CICS Transaction Server for z/OS Release Guide, GC34-6421
CICS Transaction Server for z/OS Installation Guide, GC34-6426
CICS Transaction Server for z/OS Licensed Program Specification, GC34-6608

You can order further copies of the following books in the entitlement set, using the order number quoted above:

CICS Transaction Server for z/OS Release Guide
CICS Transaction Server for z/OS Installation Guide
CICS Transaction Server for z/OS Licensed Program Specification

PDF-only books

The following books are available in the CICS Information Center as Adobe Portable Document Format (PDF) files:

CICS books for CICS Transaction Server for z/OS
General

CICS Transaction Server for z/OS Program Directory, GI10-2586
CICS Transaction Server for z/OS Release Guide, GC34-6421
CICS Transaction Server for z/OS Migration from CICS TS Version 2.3, GC34-6425

> *CICS Transaction Server for z/OS Migration from CICS TS Version 1.3*,
> GC34-6423
>
> *CICS Transaction Server for z/OS Migration from CICS TS Version 2.2*,
> GC34-6424
>
> *CICS Transaction Server for z/OS Installation Guide*, GC34-6426

Administration

> *CICS System Definition Guide*, SC34-6428
>
> *CICS Customization Guide*, SC34-6429
>
> *CICS Resource Definition Guide*, SC34-6430
>
> *CICS Operations and Utilities Guide*, SC34-6431
>
> *CICS Supplied Transactions*, SC34-6432

Programming

> *CICS Application Programming Guide*, SC34-6433
>
> *CICS Application Programming Reference*, SC34-6434
>
> *CICS System Programming Reference*, SC34-6435
>
> *CICS Front End Programming Interface User's Guide*, SC34-6436
>
> *CICS C++ OO Class Libraries*, SC34-6437
>
> *CICS Distributed Transaction Programming Guide*, SC34-6438
>
> *CICS Business Transaction Services*, SC34-6439
>
> *Java Applications in CICS*, SC34-6440
>
> *JCICS Class Reference*, SC34-6001

Diagnosis

> *CICS Problem Determination Guide*, SC34-6441
>
> *CICS Messages and Codes*, GC34-6442
>
> *CICS Diagnosis Reference*, GC34-6899
>
> *CICS Data Areas*, GC34-6902
>
> *CICS Trace Entries*, SC34-6443
>
> *CICS Supplementary Data Areas*, GC34-6905

Communication

> *CICS Intercommunication Guide*, SC34-6448
>
> *CICS External Interfaces Guide*, SC34-6449
>
> *CICS Internet Guide*, SC34-6450

Special topics

> *CICS Recovery and Restart Guide*, SC34-6451
>
> *CICS Performance Guide*, SC34-6452
>
> *CICS IMS Database Control Guide*, SC34-6453
>
> *CICS RACF Security Guide*, SC34-6454
>
> *CICS Shared Data Tables Guide*, SC34-6455
>
> *CICS DB2 Guide*, SC34-6457
>
> *CICS Debugging Tools Interfaces Reference*, GC34-6908

CICSPlex SM books for CICS Transaction Server for z/OS

General

> *CICSPlex SM Concepts and Planning*, SC34-6459
>
> *CICSPlex SM User Interface Guide*, SC34-6460
>
> *CICSPlex SM Web User Interface Guide*, SC34-6461

Administration and Management

> *CICSPlex SM Administration*, SC34-6462
>
> *CICSPlex SM Operations Views Reference*, SC34-6463
>
> *CICSPlex SM Monitor Views Reference*, SC34-6464
>
> *CICSPlex SM Managing Workloads*, SC34-6465
>
> *CICSPlex SM Managing Resource Usage*, SC34-6466
>
> *CICSPlex SM Managing Business Applications*, SC34-6467

Programming

> *CICSPlex SM Application Programming Guide*, SC34-6468
>
> *CICSPlex SM Application Programming Reference*, SC34-6469

Diagnosis

CICSPlex SM Resource Tables Reference, SC34-6470
CICSPlex SM Messages and Codes, GC34-6471
CICSPlex SM Problem Determination, GC34-6472

CICS family books

Communication

CICS Family: Interproduct Communication, SC34-6473
CICS Family: Communicating from CICS on System/390, SC34-6474

Licensed publications

The following licensed publications are not included in the unlicensed version of the Information Center:

CICS Diagnosis Reference, GC34-6899
CICS Data Areas, GC34-6902
CICS Supplementary Data Areas, GC34-6905
CICS Debugging Tools Interfaces Reference, GC34-6908

Other CICS books

The following publications contain further information about CICS, but are not provided as part of CICS Transaction Server for z/OS, Version 3 Release 1.

Designing and Programming CICS Applications	SR23-9692
CICS Application Migration Aid Guide	SC33-0768
CICS Family: API Structure	SC33-1007
CICS Family: Client/Server Programming	SC33-1435
CICS Transaction Gateway for z/OS Administration	SC34-5528
CICS Family: General Information	GC33-0155
CICS 4.1 Sample Applications Guide	SC33-1173
CICS/ESA 3.3 XRF Guide	SC33-0661

Determining if a publication is current

IBM regularly updates its publications with new and changed information. When first published, both hardcopy and BookManager® softcopy versions of a publication are usually in step. However, due to the time required to print and distribute hardcopy books, the BookManager version is more likely to have had last-minute changes made to it before publication.

Subsequent updates will probably be available in softcopy before they are available in hardcopy. This means that at any time from the availability of a release, softcopy versions should be regarded as the most up-to-date.

For CICS Transaction Server books, these softcopy updates appear regularly on the *Transaction Processing and Data Collection Kit* CD-ROM, SK2T-0730-xx. Each reissue of the collection kit is indicated by an updated order number suffix (the -xx part). For example, collection kit SK2T-0730-06 is more up-to-date than SK2T-0730-05. The collection kit is also clearly dated on the cover.

Updates to the softcopy are clearly marked by revision codes (usually a # character) to the left of the changes.

Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully.

You can perform most tasks required to set up, run, and maintain your CICS system in one of these ways:
- using a 3270 emulator logged on to CICS
- using a 3270 emulator logged on to TSO
- using a 3270 emulator as an MVS system console

IBM Personal Communications provides 3270 emulation with accessibility features for people with disabilities. You can use this product to provide the accessibility features you need in your CICS system.

Index

A

AAT (Application Assembly Tool)
    changes in support of enterprise beans   40
Application Assembly Tool (AAT)
    changes in support of enterprise beans   40
application develoment
    compiler support   59
application programming interface
    changes   29, 32
Assembly Toolkit (ATK)
    changes in support of enterprise beans   40
ATK (Assembly Toolkit)
    changes in support of enterprise beans   40

B

BTAM and TCAM networks   61, 63
BTAM networks   61
Business Transaction Services (BTS)
    DFHLRQ data set   117
    migration considerations   117
    repository data set   117

C

CCI Connector for CICS TS
    migration planning   78
CCRL transaction   11
CEBR, changes   12
CEMT commands, changed
    INQUIRE CLASSCACHE
        CLASSCACHE, new   12
    INQUIRE JVM
        JVM, new   12
    INQUIRE WORKREQUEST
        WORKREQUEST, new   13
    PERFORM CLASSCACHE
        CLASSCACHE, new   13
    SET CLASSCACHE
        CLASSCACHE, new   13
    SET WORKREQUEST
        WORKREQUEST, new   13
certificate revocation list transaction (CCRL)   11
CETR, changes   15
changed CEMT commands   13
changed CICS messages   179
changed CICSPlex SM operations views
    TCPIPS   122
changed global user exit programs
    XRSINDI   45
changed system initialization parameters   5
changed utility programs
    Resource Manager for Enterprise Beans   40
changes
    affecting global user exits   43
    affecting the API   29
    affecting the SPI   33

changes *(continued)*
    API RESP2 values   32
    resource definitions (macro)   27
    to CEBR   12
    to CETR   15
    to CICSPlex SM API   131, 132
    to CICSPlex SM operations views   121
    to CICSPlex SM Web User Interface views   137
    to control table parameters   27
    to IBM-supplied resource definitions   21
    to installation   3
    to monitoring and statistics   51
    to RDO attributes   19
    to system initialization parameters   5
    to user-replaceable programs   47
    to utility programs   39
CICS integrated translator   81
    nested programs   81
CICS-supplied transactions
    changed CEMT commands   13
    changes to CEBR   12
    changes to CETR   15
    CJMJ   17
    DFH$CAT1 CLIST   17
    migration considerations   11
    new CEMT commands   12
    new RACF category 1 transactions   17
CICSPlex SM
    changes to API   131, 132
    changes to operations views   121
    changes to problem determination   129
    connecting to previous releases   131
    new operations views   121
    Web User Interface views   137
CICSRGND operations view, new fields   122
compatibility groups, new   21
compiler support   59
control tables
    DFHCNV   28
    DFHTCTDY   28
    reassembling   28
CSD
    DFHCSDUP changed   39
    sharing between releases   24
CSD, upgrading   23
    SCAN function   22

D

DEBUGTOOL, new system initialization parameter   7
deleted abend codes   181
deleted messages   179
DFH$AFFY, CSD group   22
DFH$JAVA, CSD group   22
DFHAPXPO   49
DFHAUGRP, CSD group   22
DFHCNV   48
    reassembling tables   28

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply in the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM United Kingdom Laboratories, MP151, Hursley Park, Winchester, Hampshire, England, SO21 2JN. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. A current list of IBM trademarks is available on the Web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other product and service names might be trademarks of IBM or other companies.

Sending your comments to IBM

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM.

Feel free to comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this book.

Please limit your comments to the information in this book and the way in which the information is presented.

To ask questions, make comments about the functions of IBM products or systems, or to request additional publications, contact your IBM representative or your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

You can send your comments to IBM in any of the following ways:

- By mail, to this address:

  IBM United Kingdom Limited
  User Technologies Department (MP095)
  Hursley Park
  Winchester
  Hampshire
  SO21 2JN
  United Kingdom

- By fax:
  - From outside the U.K., after your international access code use 44–1962–816151
  - From within the U.K., use 01962–816151

- Electronically, use the appropriate network ID:
  - IBMLink: HURSLEY(IDRCF)
  - Internet: idrcf@hursley.ibm.com

Whichever you use, ensure that you include:

- The publication title and order number
- The topic to which your comment applies
- Your name and address/telephone number/fax number/network ID.

199

IBM®

Product Number: 5655-M15

Spine information:

IBM

CICS TS for z/OS

Migration Guide

Version 3
Release 1