CICS Transaction Server for z/OS

Migration from CICS TS Version 2.3

Version 3 Release 2

GC34-6857-00

CICS Transaction Server for z/OS

Migration from CICS TS Version 2.3

Version 3 Release 2

GC34-6857-00

Note!
Before using this information and the product it supports, be sure to read the general information under “Notices” on page 211.

This edition applies to Version 3 Release 2 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.

© Copyright International Business Machines Corporation 2001, 2007. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Preface ix

Part 1. Changes to CICS externals 1

Chapter 1. Changes to installation 3
CICSPlex SM installation integrated with CICS installation 3
Requirement for unique CICS region applid 4
Requirement for MEMLIMIT value 4
Release levels on INQUIRE SYSTEM command 4
Installing IPIC support 4

Chapter 2. Changes to system initialization parameters 7
Obsolete system initialization parameters. 7
Changed system initialization parameters 8
New system initialization parameters 9

Chapter 3. Changes to CICS-supplied transactions 13
Obsolete options on CEMT commands 13
Changed CEMT commands 13
New CEMT commands 20
Changes to CETR. 21
Changes to CWXN 22
New transaction CCRL 23
New transaction CEMN 23
New transaction CJGC 24
New transaction CJPI 24
New transaction CWXU. 24
Additions to CICS RACF category 1 transactions 24

Chapter 4. Changes to resource definition 27
Obsolete CICS-supplied resource definitions 27
Changed resource definition attributes 27
Changes to CICS-supplied resource definitions 28
Resource definitions moved to group DFHCOMPA 28
New resource definition types and new attributes 28
New CICS-supplied resource definitions. 29
DFHDP. 30
DFHDPWB 30
DFHSO 30
DFH$WEB 30
Changes to control tables (macro resource definition). 31

Chapter 5. Changes to the application programming interface (API) 33
Obsolete API commands and options 33
Changed API commands 33
New API commands 36
Changes to EXEC CICS VERIFY PASSWORD 37
Changes to sample programs 38

Chapter 6. Changes to the system programming interface (SPI) 41
Obsolete options or values on SPI commands 41
Obsolete CVDA values 42
New options or values on SPI commands 42

© Copyright IBM Corp. 2001, 2007

New SPI commands 50
Changes to DOCTEMPLATE SPI commands 52

Chapter 7. Changes to CICS-supplied utility programs 53
New CICS JVM Application Isolation Utility. 54

Chapter 8. Changes to global user exits, task-related user exits, and the exit programming interface 55
Obsolete global user exit points. 55
Changes to the DFHUEPAR standard parameter list 55
Changes to global user exits 56
Changes to global user exit points because of channels. 59
New global user exit points 59
Changes to the exit programming interface 60
Changes to task-related user exits. 60

Chapter 9. Changes to the external CICS interface (EXCI) 63

Chapter 10. Changes to user-replaceable programs 65
Changes to DFHCNV 65
New user-replaceable programs 65
DFHAPXPO 65
DFHISAIP. 66

Chapter 11. Changes to CICS monitoring 67
New and changed monitoring data fields 67
Effects of monitoring clock changes on performance data 73
Changes to the format of CICS SMF 110 monitoring records 74
Changes to the monitoring sample program DFH$MOLS 74
DFH$MOLS support for data for earlier CICS releases 76

Chapter 12. Changes to CICS statistics 79

Chapter 13. Changes to problem determination for CICS-MQ, CICS-DBCTL and CICS-DB2 81

Part 2. Migration tasks for CICS Transaction Server 83

Chapter 14. Migration procedures for CICS Transaction Server. 85
Redefining and initializing the local and global catalogs 85
Enabling z/OS conversion services 85
Upgrading the CSD for CICS-supplied and other IBM-supplied resource definitions 86
Migrating user-modified CICS-supplied resource definitions 88
Migrating copies of CICS-supplied resource definitions 89
CSD compatibility between different CICS releases 89

Chapter 15. Migration for application programs 91

Chapter 16. Migration for Business Transaction Services (BTS) 95

Chapter 17. Migration for BTAM and TCAM networks 97
Using dummy consoles instead of dummy TCAM terminals 97

Chapter 18. Migrating from COMMAREAs to channels 99
Migrating LINK commands that pass COMMAREAs 99

Migrating XCTL commands that pass COMMAREAs 99
Migrating pseudoconversational COMMAREAs on RETURN commands . . . 100
Migrating START data 100
Migrating dynamically-routed applications 100
Coexistence of programs that use channels with back-level CICS releases 101

Chapter 19. Migrating to extended addressing for ESDS 103

Chapter 20. Migration for intercommunication 105
Giving each CICS region a unique applid 105
Migrating connections to IP interconnectivity 105
The migration utility for converting APPC and MRO connections to IPIC connections 109
Using TCP/IP management and control 113
Migration for multiregion operation (MRO) 113
Using multiple CICS XCF groups 114

Chapter 21. Migrating to the Enterprise Workload Manager 117

Chapter 22. Migration for Java applications 119
Key changes to CICS support for Java applications 120
Migration from resettable to continuous JVMs 121
Possible Java application behavior changes in continuous JVMs 122
Auditing Java applications for the use of static variables 124
Migrating storage settings in JVM profiles from resettable JVMs 126
Migrating class paths in JVM profiles 127
Migrating class paths in JVM profiles: library path 128
Migrating class paths in JVM profiles: middleware classes 128
Migrating class paths in JVM profiles: standard class path 129
Migrating Java garbage collection settings 129
Migration for HPJ-compiled Java programs (Java program objects) 130
Migration for Java applications that use a CICS connector 130

Chapter 23. Migration for threadsafe programming and the open transaction environment (OTE) 133
How can I make my applications threadsafe? 135
Important changes to accounting for processor time in the open transaction environment 138

Chapter 24. Migration for CICS Web support applications 141
Migrating to resource security for CICS document templates and z/OS UNIX files 145

Chapter 25. Migration for CICS Web services 147
Migration for the Web services assistants 147
Migration for MTOM/XOP support 147

Chapter 26. Migrating to the XPLINK option for C and C++ programs 149

Part 3. Changes to CICSPlex SM externals 151

Chapter 27. Changes to CICSPlex SM installation and definition 153
CICSPlex SM installation integrated with CICS installation 153
Removal of the CAS 153
Dynamic creation of CICS resource definitions for CICSPlex SM 154
New method for WUI and CICSplex definition 155

New CICSPlex SM WUI server initialization parameters 155
New EYU9XDBT utility for CMAS and CICSplex definition 156
Change to Common Work Area size for a CMAS 156
Changes with RASGNDEF processing in CICSPlex SM 156
Change to generic alert structures used by CICSPlex SM. 157

Chapter 28. Changes to CICSPlex SM views and resource tables 159
Removal of the CICSPlex SM TSO end user interface (EUI). 159
New data type SCLOCK12 159
Changes to CICSPlex SM Web User Interface security. 161
Obsolete CICSPlex SM views, resource tables and attributes 162
CICSPlex SM resource tables no longer supported in CICS Transaction Server for z/OS, Version 3 Release 2 162
Changed CICSPlex SM views and resource tables 163
New CICSPlex SM views and resource tables 165
New Business Application Services definition objects 166
Changed Business Application Services definition objects. 167

Chapter 29. National language support for CICSPlex SM messages . . . 169

Chapter 30. Programs that connect to a previous release of CICSPlex SM 171

Part 4. Migration tasks for CICSPlex SM 173

Chapter 31. Running CICSPlex SM Version 3.2 and earlier releases concurrently 175

Chapter 32. Removal of support for Windows remote MAS 177

Chapter 33. Performing migration procedures for CICSPlex SM 179
Converting a CMAS to Version 3.2 179
Converting a MAS to Version 3.2. 180
Migration for CICSPlex SM workload management 181
Application programming interface 182
Converting a Web User Interface Server to Version 3.2 182
Migrating the MAS and updating the Web User Interface CSD group . . . 183
Migrating the contents of the Web User Interface server repository (EYUWREP) 183
Deleting the previous release definitions from CSD files 184

Chapter 34. A phased migration scenario. 187
The environment. 187
Objective 1: Add a WUI server at the earlier release. 189
Objective 2: Convert MP CMAS to the new version 190
Objective 3: Convert CMAS B to the new version. 192
Objective 4: Convert CMAS C to the new version. 194

Part 5. Appendixes 197

Bibliography 199
The CICS Transaction Server for z/OS library 199
The entitlement set 199
PDF-only books 199
Other CICS books 201
Determining if a publication is current 201

Accessibility 203

Index 205

Notices 211
Trademarks. 212
Trademarks and service marks 212

Preface

This book is about migration to CICS® Transaction Server for z/OS®, Version 3 Release 2.

This book provides the relevant information for users who plan to migrate from CICS Transaction Server for z/OS, Version 2 Release 3. 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.

If you are migrating from an older release, select the Migration Guide for the release from which you are migrating. The Migration Guides for older releases include additional information about changes that took place in the intervening releases.

Note: The oldest release for which migration information is provided with CICS Transaction Server for z/OS, Version 3 Release 2 is CICS Transaction Server for OS/390®, Version 1 Release 3. If you are migrating from a release of CICS earlier than this, you are recommended to read the *Release Guide* and the *Migration Guide* (where applicable) for any additional intervening releases.

This book 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.

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

You should also have read about the new function in this release of CICS Transaction Server as described in the *CICS Transaction Server for z/OS Release Guide*.

Notes on terminology

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

CICS TS, unless stated otherwise, refers to the release of CICS Transaction Server for z/OS to which you are migrating.

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

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

© Copyright IBM Corp. 2001, 2007

Part 1. Changes to CICS externals

© Copyright IBM Corp. 2001, 2007

Chapter 1. Changes to installation

When migrating to CICS Transaction Server for z/OS, Version 3 Release 2, note these changes to the installation process and important new requirements.

This release of CICS Transaction Server can be installed using the SMP/E RECEIVE, APPLY, and ACCEPT commands. The SMP/E dialogs can be used to accomplish the SMP/E installation steps. The process is in line with IBM® Corporate Standards, and might be familiar to those who have installed other z/OS products.

The DFHISTAR process for installing CICS Transaction Server is still available if you prefer it.

The *CICS Transaction Server for z/OS Installation Guide* has information about all the processes for installing CICS Transaction Server.

SCEERUN2 and SCEERUN libraries

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.

CICSPlex SM installation integrated with CICS installation

You can now edit the DFHISTAR job to modify both the CICS and CICSPlex SM installation parameters for your environment. EYUISTAR is no longer available as a job to modify CICSPlex SM installation parameters.

DFHISTAR produces customized JCL for CICS and CICSPlex SM. It now includes a number of parameters that are significant only for CICSPlex SM, a number of parameters that are significant only for CICS, and a number of parameters that are common to CICS and CICSPlex SM.

For CICSPlex SM, DFHISTAR generates sample JCL procedures to:

- Create CMAS data sets
- Start a CMAS
- Create Web User Interface (WUI) data sets
- Start a WUI
- Create MAS data sets
- Run a MAS
- Move MAS modules to the link pack area (LPA)

These procedures enable you to create a simple CICSPlex SM configuration that consists of a CMAS, a WUI and a managed CICS system (MAS). The CICSPlex SM Starter Set, which contained samples of JCL for this purpose, is no longer provided.

© Copyright IBM Corp. 2001, 2007

Requirement for unique CICS region applid

If your CICS regions are part of a z/OS sysplex, you must ensure that their applids (as specified on the APPLID system initialization parameter) are unique within the sysplex. If CICS is an XRF partner, its specific applid must be unique within the sysplex. In CICS TS for z/OS, Version 3.2, if an applid is not unique (that is, it duplicates the (specific or only) applid of any other CICS region currently active in the sysplex), on startup CICS issues a message and fails to initialize.

This requirement is to support IPIC connections, but it is required whether or not you plan to use that type of connection. It is also to allow multiple XCF groups to contain CICS regions.

Requirement for MEMLIMIT value

To use storage above the 2GB boundary (above the bar) when migrating to CICS TS for z/OS, Version 3.2, you need to set your MEMLIMIT value equal to or greater than 2GB.

The MEMLIMIT value can be set using any of these methods:

1. JCL. MEMLIMIT can either be set to a specific value in JCL or as NOLIMIT, if REGION=0M is specified.
2. SMFPRM PARMLIB member. A MEMLIMIT value can be set in SYS1.PARMLIB(SMFPRMxx).
3. IEFUSI z/OS global user exit.

System usage and workload remain the same as before you altered the MEMLIMIT value. The MEMLIMIT value cannot be altered on a running system.

If MEMLIMIT is set lower than 2GB, but higher than EDSALIM, a warning message is displayed. If MEMLIMIT is set lower than the EDSALIM value, an error message is displayed and CICS does not start up.

Release levels on INQUIRE SYSTEM command

You can use the EXEC CICS INQUIRE SYSTEM CICSTSLEVEL command to determine the version and release number of CICS. Use the EXEC CICS INQUIRE SYSTEM OSLEVEL command to determine the level of z/OS.

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 level number in CICS TS 3.2 is 030200. This number is returned in the RELEASE parameter of the INQUIRE SYSTEM command.

The level number also appears in the alternative decimal form 6.5 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 DFHPD650.

Installing IPIC support

This topic describes how to install support for IPIC.

System initialization parameters

To activate them at CICS startup, specify TCPIP=YES as a system initialization parameter. (The default value of the **TCPIP** parameter is NO.) For reference information about the **TCPIP** system initialization parameter, see the *CICS System Definition Guide*.

Defining IPIC connections

Before you can use IPIC, you must:

- define and install IPCONNs with attributes appropriate to your CICS environment. For guidance information about defining IPCONNs, see the *CICS Intercommunication Guide*. For reference information about IPCONN definitions, see the *CICS Resource Definition Guide*. IPIC connections can be autoinstalled, see the *CICS Customization Guide*.
- define and install a TCPIPSERVICE definition with the PROTOCOL attribute set to IPIC. See the *CICS Resource Definition Guide* for reference information about TCPIPSERVICE definitions.

Chapter 2. Changes to system initialization parameters

This section summarizes the changes to CICS system initialization parameters.

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

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).

Obsolete system initialization parameters

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.

System initialization parameters made obsolete in CICS Transaction Server for z/OS, Version 3 Release 2

MNSUBSYS

This parameter, used in earlier releases to specify the subsystem identification in monitoring SYSEVENT class records, is obsolete. If it is specified, it is rejected with a message.

System initialization parameters made obsolete in earlier CICS releases

Table 1. System initialization parameters made obsolete in earlier CICS releases

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. In CICS Transaction Server for z/OS, Version 3 Release 1, support for the TCAM/ACB interface was removed. The TCAM/DCB interface is supported only indirectly.

© Copyright IBM Corp. 2001, 2007

Changed system initialization parameters

For more information on the changed system initialization parameters, see The system initialization parameter descriptions in the *CICS System Definition Guide*.

System initialization parameters changed in CICS Transaction Server for z/OS, Version 3 Release 2

APPLID={DBDCCICS|applid}

CICS application identifiers (APPLIDs) are now required to be unique within the local sysplex: see “Requirement for unique CICS region applid” on page 4.

Also, APPLID can now be used as the application identifier of this CICS region on IPIC connections.

The following text has been added to the description of the APPLID parameter:

If CICS is running in a sysplex, its applid must be unique within the sysplex. Note that, if the CICS extended recovery facility (XRF) is used by any of the regions in the sysplex, the specified applid must not duplicate the *specific* applid of any XRF CICS region. If, on CICS startup, the specified applid is found to duplicate the (specific or only) applid of any other CICS region currently active in the sysplex, CICS issues message DFHPA1946 and fails to initialize.

This parameter can be used also as the application identifier of this CICS region on IPIC connections.

When you define this CICS region to another CICS region, in an MRO or ISC over SNA CONNECTION definition you specify the applid as the NETNAME; in an IPIC IPCONN definition you specify the applid as the APPLID.

CONFDATA={SHOW|HIDETC}

CONFDATA now applies to initial input data received on IPIC connections (IS data), as well as to initial input data received on VTAM RECEIVE ANY operations, MRO connections, and FEPI screens and RPLAREAs:

- **IS**: CICS does not trace the initial input received on an IPIC link.

  Trace points SO 0201 and SO 0202 suppress buffer data with the message ″Trace data suppressed because it may contain sensitive data″. Subsequent trace point SO 029D (buffer continuation) and buffer data from trace points WB 0700 and WB 0701 is suppressed.

  If the transaction definition specifies CONFDATA(NO), IS trace entries are created with the user data, as normal.

  If the transaction definition specifies CONFDATA(YES), user data from IS trace points IS 0702 and IS 0906 is replaced with ″SUPPRESSED DUE TO CONFDATA=HIDETC IN SIT″. Data from IS trace points IS 0603 and IS 0703 is not shown.

EDASALIM

{**34M**|number}

The default is now 34MB. If you have created your SIT using previously supplied defaults, you should update the table to use the new CICS-supplied defaults.

ICVTSD={500|number}
ICVTSD, the terminal scan delay value that determines how quickly CICS deals with some terminal I/O requests made by applications, now applies also to IP interconnectivity input.

MSGCASE={MIXED|UPPER}
In previous releases, this parameter applied only to messages displayed by the CICS message domain. It now also applies to messages displayed by the CPSM message domain.

MIXED
This is the default in the SIT; all messages displayed by the CICS message domain or the CPSM message domain remain in mixed case.

UPPER
The message domain displays all mixed case messages in uppercase only.

Note: Mixed case output is not displayed correctly on Katakana display terminals and printers. Uppercase English characters appear correctly as uppercase English characters, but lowercase appears as Katakana symbols. If you have any Katakana terminals connected to your CICS region, specify MSGCASE=UPPER.

Note: If you wish to use uppercase English for your CICS-WebSphere MQ components, you must set MSGCASE=UPPER, and ensure that ASSIGN NATLANGINUSE returns E (US English).

UOWNETQL=`user_defined_value`
On VTAM=NO regions, UOWNETQL, or its default value, is now used as the default NETWORKID of this CICS region on the IPCONN definitions that define IPIC connections.

System initialization parameters changed in earlier CICS releases

Table 2. Changed system initialization parameters in earlier CICS releases

Keywords	Operands	Explanation
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.

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.

For information on the new system initialization parameters, see The system initialization parameter descriptions in the *CICS System Definition Guide*.

For information about the new function relating to these new system initialization parameters, see What's new.

New system initialization parameters added in CICS Transaction Server for z/OS, Version 3 Release 2

FCQRONLY={NO|YES}

Specifies whether you want CICS to force all CICSAPI user application programs that are specified as threadsafe to run file control requests under the CICS QR TCB, as if they were specified as quasi-reentrant programs.

NO CICS honors the CONCURRENCY(THREADSAFE) attribute on program resource definitions, and allows user application programs to run applicable file control request on an open TCB to avoid unnecessary TCB switching.

YES CICS forces all file control requests to run under the CICS QR TCB, as if they were specified as CONCURRENCY(QUASIRENT) programs. With all file requests on the QR TCB, CICS is able to minimise the amount of locking required at the expense of additional TCB switches if requests are run on open TCBs.

The **FCQRONLY=YES** can improve the performance of file-owning regions.

XHFS={YES|NO}

specifies whether CICS is to check the transaction user's ability to access files in the z/OS UNIX® System Services file system. At present, this checking applies only to the user ID of the Web client when CICS Web support is returning z/OS UNIX file data as the static content identified by a URIMAP definition.

Note: The checking is performed only if you have specified YES for the SEC system initialization parameter. However, the RESSEC option on the transaction resource definition does **not** affect this security checking.

YES CICS is to check whether the user identified as the Web client is authorized to access the file identified by the URIMAP that matches the incoming URL. This check is in addition to the check performed by z/OS UNIX System Services against the CICS region user ID. If access to the file is denied for either of these user IDs, the HTTP request is rejected with a 403 (Forbidden) response.

NO CICS is not to check the client user's access to z/OS UNIX files. Note that the CICS region user ID's access to these files is still checked by z/OS UNIX System Services.

Restrictions You can specify the **XHFS** parameter in the SIT, PARM, or SYSIN only.

XRES={YES|name|NO}

specifies whether you want CICS to perform resource security checking for DOCTEMPLATE (CICS document template) resources, and optionally specifies the general resource class name in which you have defined the resource security profiles. If you specify YES, or a general resource class name, CICS calls the external security manager to verify that the userid associated with a transaction is authorized to use the resource. This checking is performed every time a transaction tries to access a CICS document template.

The actual profile name passed to the external security manager is the name of the DOCTEMPLATE resource definition for the CICS document template to be checked, prefixed by its resource type, DOCTEMPLATE. For example, for a document template whose resource definition is named "WELCOME", the

profile name passed to the external security manager is DOCTEMPLATE.WELCOME. Even if a command references the document template using its 48-character template name, the shorter name (up to 8 characters) of the DOCTEMPLATE resource definition is always used for security checking.

Note: The checking is performed only if you have specified YES for the **SEC** system initialization parameter and specified the RESSEC(YES) option on the transaction resource definition.

YES CICS calls the external security manager, using the default CICS resource class name of RCICSRES, to check whether the userid associated with a transaction is authorized to use the resource it is trying to access. The resource class name is RCICSRES and the grouping class name is WCICSRES.

name CICS calls the external security manager, using the specified resource class name prefixed by the letter R, to check whether the userid associated with a transaction is authorized to use the resource it is trying to access. The resource class name is R*name* and the grouping class name is W*name*. The resource class name specified must be 1 through 7 characters.

NO CICS does not perform any security checks for DOCTEMPLATE resources, allowing any user to access any CICS document template.

Restrictions You can specify the **XRES** parameter in the SIT, PARM, or SYSIN only.

New system initialization parameters added in earlier CICS releases

Table 3. New system initialization parameters added in earlier releases

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.
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.
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.

Table 3. New system initialization parameters added in earlier releases (continued)

Keywords	Operands	Explanation
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.
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.
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.
SSLCACHE	{**CICS**| SYSPLEX}	Specifies whether SSL is to use the local or sysplex caching of session ids.

Chapter 3. Changes to CICS-supplied transactions

This section summarizes the changes to CICS-supplied transactions.

Obsolete options on CEMT commands

Obsolete options on CEMT commands in CICS Transaction Server for z/OS, Version 3 Release 2

INQUIRE CLASSCACHE: The value Reset is no longer displayed for the REUSEST option. If the shared class cache is not started, the value Unknown is displayed. In this situation, CICS cannot identify the reuse status, but when the shared class cache is started, the status always becomes Reuse.

INQUIRE JVM: The value Reset is no longer displayed for the REUSEST option.

INQUIRE MONITOR: The SUBSYSTEMID option is obsolete and has been removed.

SET JVMPOOL: The TERMINATE function on the SET JVMPOOL command is now deprecated. You should use the improved function on the PERFORM JVMPOOL command instead.

Obsolete options on CEMT commands in earlier CICS releases

Table 4. Obsolete options on CEMT commands in earlier CICS releases

CEMT command	Option	Comment
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.

Changed CEMT commands

Changed CEMT commands in CICS Transaction Server for z/OS, Version 3 Release 2

DISCARD: new IPCONN resource type

© Copyright IBM Corp. 2001, 2007

Ipconn(*value*)
specifies the name of the IPCONN to be removed. The name can be up to 8 characters long. You cannot discard an IPCONN unless it is in OUTSERVICE status.

DISCARD: new LIBRARY resource type

LIBRARY(*name*)
specifies the name of a LIBRARY resource that you want to remove. The name can be up to 8 characters long. The LIBRARY must be disabled. The DISCARD LIBRARY command removes an installed resource definition and its corresponding catalog entry from an active CICS system.

Note: Specifying a LIBRARY name of DFHRPL is invalid, and will result in the message 'NOT VALID FOR RPL' being displayed.

INQUIRE DOCTEMPLATE: new SIZE option

Size
returns the amount of storage, in bytes, used by the cached copy of the document template. A value of zero is returned if there is no cached copy of the template at the time of the enquiry.

INQUIRE DSAS: removal of SOSSTATUS option, new MEMLIMIT, SOSABOVEBAR, SOSABOVELINE and SOSBELOWLINE options

Memlimit(*value*)
displays the amount of storage available above the 2GB boundary (above the bar), for use by the CICS region. A value of NOLIMIT indicates that no limit has been imposed on the amount of storage that the region can attempt to use.

Sosabovebar(*value*)
displays whether CICS is short-on-storage in the dynamic storage areas above the 2GB boundary (above the bar).
Notsos
CICS is not short-on-storage in any of the dynamic storage areas above the 2GB boundary.
Sos CICS is short-on-storage in at least one of the dynamic storage areas above the 2GB boundary.

Sosaboveline(*value*)
displays whether CICS is short-on-storage in the dynamic storage areas above the 16MB line, but below the 2GB boundary.
Notsos
CICS is not short-on-storage in any of the dynamic storage areas above the 16MB line (but below the 2GB boundary).
Sos CICS is short-on-storage in at least one of the dynamic storage areas above the 16MB line (but below the 2GB boundary).

Sosbelowline(*value*)
displays whether CICS is short-on-storage in the dynamic storage areas below the 16MB line.
Notsos
CICS is not short-on-storage in any of the dynamic storage areas below the 16MB line.
Sos CICS is short-on-storage in at least one of the dynamic storage areas below the 16MB line.

INQUIRE FILE: new RBATYPE option

Rbatype
displays whether, for VSAM files, the data set uses extended addressing. The values are:

Extended
This VSAM data set uses extended relative byte addressing and therefore can hold more than 4 gigabytes of data.

Notapplic
One of the following is true:
- The data set is BDAM.
- The file is remote.
- The file is not open.

Notextended
This VSAM data set does not use extended relative byte addressing and therefore cannot hold more than 4 gigabytes of data.

You cannot modify the contents of this field.

INQUIRE IRC: new XCFGROUP option

Xcfgroup
displays the name of the cross-system coupling facility (XCF) group of which this region is a member.

If this region is not a member of an XCF group (because it has not signed on to IRC), XCFGROUP displays the XCF group the region would be in if XCF were opened.

For introductory information about XCF/MRO, see Cross-system multiregion operation (XCF/MRO) in the *CICS Intercommunication Guide*.

INQUIRE MONITOR: new COMPRESSST option

COMpressst
displays whether data compression is performed for monitoring records. The values are:
Compress
Data compression is performed.
Nocompress
Data compression is not performed.

Note: You can reset this value by overtyping it with a different value.

INQUIRE PROGRAM: The USECOUNT option now displays a use count for Java programs. In earlier CICS releases, this count was not available.

INQUIRE PIPELINE: new options

Ciddomain*(value)*
Displays the name of the domain that is used to generate MIME content-ID values that identify binary attachments.

Mode*(value)*
Displays the operating mode of the pipeline.

PROVIDER
CICS is using the pipeline as a service provider of Web services.

REQUESTER
CICS is using the pipeline as a service requester of Web services.

UNKNOWN
The operating mode of the pipeline cannot be determined.

Mtomnoxopst*(value)*
Displays the status of the pipeline for sending outbound messages in MIME format when binary attachments are not present.

MTOMNOXOP
Outbound messages are sent in MIME format, even when there are no binary attachments present.

NOMTOMNOXOP
Outbound messages are only sent in MIME format when there are binary attachments present.

Mtomst*(value)*
Displays the status of the MTOM handler in the pipeline.

MTOM
The MTOM handler is enabled in the pipeline.

NOMTOM
The MTOM handler is not enabled in the pipeline.

Respwait *(number)*
Displays the number of seconds that an application program waits for an optional message from a remote Web service. If no value is displayed, the default timeout value of the transport protocol is being used.

- The default timeout value for HTTP is 10 seconds.
- The default timeout value for WebSphere® MQ is 60 seconds.

Sendmtomst*(value)*
Displays the status of the pipeline for sending outbound messages in MIME format.

NOSENDMTOM
Outbound messages are never sent in MIME format.

SAMESENDMTOM
Outbound messages are only sent in MIME format when the inbound message is in MIME format.

SENDMTOM
Outbound messages are always sent in MIME format.

SOAPlevel*(value)*
Displays the level of SOAP that is supported in the pipeline. The SOAP level can be 1.1 or 1.2. If the pipeline is not being used for SOAP messages, a value of NOTSOAP is displayed.

Xopdirectst*(value)*
Displays the status of the pipeline for handling XOP documents and binary attachments in direct or compatibility mode.

XOPDIRECT
The pipeline is processing XOP documents and binary attachments in direct mode.

NOXOPDIRECT
The pipeline is processing XOP documents and binary attachments in compatibility mode.

Xopsupportst*(value)*
Displays the status of the application handler for processing XOP documents and binary attachments directly.

XOPSUPPORT
The application handler supports the direct handling of XOP documents and binary attachments.

NOXOPSUPPORT
The application handler does not support the direct handling of XOP documents and binary attachments.

INQUIRE SYSTEM: removal of SOSSTATUS option, new SOSABOVEBAR, SOSABOVELINE and SOSBELOWLINE options

Sosabovebar(*value***)**
displays whether CICS is short on storage in the dynamic storage areas above the 2GB boundary (above the bar).
Notsos
CICS is not short on storage in any of the dynamic storage areas above the 2GB boundary.
Sos CICS is short on storage in at least one of the dynamic storage areas above the 2GB boundary.

Sosaboveline(*value***)**
displays whether CICS is short on storage in the dynamic storage areas above the 16MB line, but below the 2GB boundary.
Notsos
CICS is not short on storage in any of the dynamic storage areas above the 16MB line (but below the 2GB boundary).
Sos CICS is short on storage in at least one of the dynamic storage areas above the 16MB line (but below the 2GB boundary).

Sosbelowline(*value***)**
displays whether CICS is short on storage in the dynamic storage areas below the 16MB line.
Notsos
CICS is not short on storage in any of the dynamic storage areas below the 16MB line.
Sos CICS is short on storage in at least one of the dynamic storage areas below the 16MB line.

INQUIRE TCPIPSERVICE: new options: A new value of Identify can be returned on the ATTACHSEC option:

Attachsec
indicates the level of attach-time security used by the connection. Values are:

Identify
Incoming attach requests must specify a user identifier. (IDENTIFY is used when the connecting system has a security manager; for example, if it is another CICS region.)

Local
CICS does not require a userid or password from clients.

Notapplic
This option has no meaning for Web interface or IIOP TCP/IP services.

Verify
Incoming attach requests must specify a user identifier and a user password.

Realm (*value***)**
returns the 56-character realm that is used during the process of HTTP basic authentication.

There is also a new value for the PROTOCOL option:

IPic
IP interconnectivity.

INQUIRE WEBSERVICE: new options

CCSID(*value*)
Displays the CCSID that is used to encode data between the application program and the Web service binding file at run time. This value is set using the optional **CCSID** parameter in the Web services assistant when the Web service binding file was generated. If the *value* is 0, the default CCSID for the CICS region that is specified by the **LOCALCCSID** system initialization parameter is used.

Mappinglevel(*value*)
Displays the mapping level that is used to convert data between language structures and Web service description (WSDL) documents. The value of the mapping level is 1.0, 1.1, 1.2 or 2.0. The default is to use a mapping level of 1.0.

Minrunlevel(*value*)
Displays the minimum runtime level that is required to run the Web service in CICS. The value of the runtime level is 1.0, 1.1, 1.2 or 2.0.

Xopdirectst*(value)*
Indicates whether the Web service is capable of handling XOP documents and binary attachments in direct mode.

NOXOPDIRECT
The Web service cannot handle XOP documents and binary attachments in direct mode. This is either because validation is switched on for the Web service, or because the Web service implementation does not support the handling of XOP documents and binary attachments. Compatibility mode is used instead.

XOPDIRECT
The Web service can handle XOP documents and binary attachments in direct mode.

Xopsupportst*(value)*
Indicates whether the Web service implementation is capable of handling XOP documents and binary attachments.

NOXOPSUPPORT
The Web service implementation is not capable of handling XOP documents and binary attachments.

XOPSUPPORT
The Web service implementation is capable of handling XOP documents and binary attachments. This is true for any CICS-generated web service created by a level of CICS that supports MTOM/XOP.

PERFORM STATISTICS: new LIBRARY option

LIBrary
LIBRARY statistics are to be written immediately to the SMF data set.

SET MONITOR: new options

COMpress
Data compression is to be performed for monitoring records.

NOCOMpress
Data compression is not to be performed for monitoring records.

SET PIPELINE: new RESPWAIT option

Respwait(*value*)
Specifies the time in seconds that an application program should wait for a response message from a remote Web service. The value can range from 0 to 9999 seconds.

If you do not specify a value, the default timeout value of the transport protocol is used.

- The default timeout value for HTTP is 10 seconds.
- The default timeout value for MQ is 60 seconds.

PERFORM STATISTICS: Statistics can be written for the new resource types DOCTEMPLATE, IPCONN, and MQCONN.

DISCARD DOCTEMPLATE, INQUIRE DOCTEMPLATE, SET DOCTEMPLATE: If resource security for document templates is active in the CICS region, with the XRES system initialization parameter set on (which is the default), and assuming RESSEC(YES) is specified for CEMT, these commands are subject to resource security checking.

Changed CEMT commands in earlier CICS releases

Table 5. Changed CEMT commands in earlier CICS releases

CEMT command	Option	Explanation
INQUIRE DISPATCHER	ACTSSLTCBS ACTXPTCBS MAXSSLTCBS MAXXPTCBS	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.
INQUIRE DOCTEMPLATE	HFSFILE	Returns the fully-qualified name of the z/OS UNIX System Services file where the template resides.

Table 5. Changed CEMT commands in earlier CICS releases (continued)

CEMT command	Option	Explanation
INQUIRE PROGRAM	APIST 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	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	CRLSERVER SSLCACHE	CRLSERVER displays the name and port number of the LDAP server that is used to store certificate revocation lists for SSL connections. SSLCACHE displays whether CICS is using local (CICS) or sysplex caching of session ids.
INQUIRE TCPIPSERVICE	MAXDATALEN	Displays the maximum length of data that may be received by CICS as an HTTP server.
INQUIRE WORKREQUEST	SOAP	Specifies that any action you request is limited to Web service work requests.
PERFORM STATISTICS	PIPELINE WEBSERVICE	New resource types for which statistics can be recorded.
SET DISPATCHER	MAXSSLTCBS MAXXPTCBS	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 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	Specifies the maximum length of data that may be received by CICS as an HTTP server.
SET WORKREQUEST	SOAP	Specifies that any action you request is limited to Web service work requests.

New CEMT commands

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

New CEMT commands in CICS Transaction Server for z/OS, Version 3 Release 2

INQUIRE IPCONN

Retrieve information about IPCONNs.

INQUIRE LIBRARY

Retrieve information about LIBRARY resources.

PERFORM JVMPOOL

Start and terminate JVMs in the JVM pool.

SET IPCONN

Change the attributes of an IPCONN or cancel outstanding AIDs.

SET LIBRARY

Change the attributes of LIBRARY resource.

SET DOCTEMPLATE

Refresh the cached copy of a document template installed in your CICS region, or phase in a new copy of a CICS program or exit program that is defined as a document template.

New CEMT commands in earlier CICS releases

Table 6. New CEMT commands in earlier CICS releases

CEMT command	Function
DISCARD PIPELINE	Use the command to remove a PIPELINE from the CICS system and the CICS catalog.
DISCARD URIMAP	Remove a URIMAP from the CICS system and the CICS catalog.
DISCARD WEBSERVICE	Remove a WEBSERVICE from the CICS system and the CICS catalog.
INQUIRE HOST	Retrieve information about virtual hosts in the local CICS region.
INQUIRE PIPELINE	Retrieve information about an installed PIPELINE.
INQUIRE URIMAP	Retrieve information about URIMAP resource definitions.
INQUIRE WEBSERVICE	Retrieve information about an installed WEBSERVICE.
PERFORM PIPELINE	Initiate a scan of the Web service binding directory that is specified in the WSBIND attribute of the PIPELINE definition.
SET HOST	Enable or disable a virtual host.
SET PIPELINE	Enable or disable a PIPELINE.
SET URIMAP	Enable or disable a URIMAP definition, and apply or remove redirection for a URIMAP definition.
SET WEBSERVICE	Set the validation status of a WEBSERVICE.

Changes to CETR

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

IS Intersystems communication and MRO
PI Pipeline Manager domain
RA Resource manager adapters

Changes to CWXN

If you are migrating from a CICS release earlier than CICS Transaction Server for z/OS, Version 3 Release 1, there are several changes to the processing carried out by the CICS-supplied transaction CWXN, the Web attach transaction.

The most significant of the changes 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.

New transaction CCRL

CCRL, the certificate revocation lists transaction, was introduced in CICS Transaction Server for z/OS, Version 3 Release 1. Use CCRL 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 CRLSERVER system initialization parameter is defined with the name of the LDAP server that stores the certificate revocation lists.

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 CEMN

CEMN, the CICS monitoring facility transaction, was introduced in CICS Transaction Server for z/OS, Version 3 Release 2.

CEMN gives you an alternative to the INQUIRE MONITOR and SET MONITOR system programming commands and the equivalent CEMT commands. You can use the transaction to inquire on the settings for the CICS monitoring facility, and to change some of the settings without needing to restart CICS.

The topic in the *CICS Supplied Transactions* explains how to use CEMN and lists the settings that can be changed using it.

CEMN is a Category 2 transaction.

New transaction CJGC

CJGC, the CICS JVM garbage collection transaction, was introduced in CICS Transaction Server for z/OS, Version 3 Release 2.

When CICS initiates garbage collection in a JVM, this transaction is used for the process, so that the time spent in garbage collection is assigned to CJGC rather than to one of the user transactions that used the JVM.

If garbage collection is caused by an allocation failure in the JVM, rather than being scheduled by CICS, this takes place while the user application is running, and the CJGC transaction is not used.

The GC_HEAP_THRESHOLD option in the JVM profile specifies the heap utilization limit at which CICS initiates garbage collection. The default is 85%. If GC_HEAP_THRESHOLD is set to 100, CICS never initiates garbage collection, so the CJGC transaction is not used.

CJGC is a Category 1 transaction.

New transaction CJPI

CJPI was introduced in CICS Transaction Server for z/OS, Version 3 Release 2. It starts up new JVMs as a result of a PERFORM JVMPOOL command.

CJPI is a Category 1 transaction.

New transaction CWXU

CWXU, the CICS Web user-defined protocol attach transaction, was introduced in CICS Transaction Server for z/OS, Version 3 Release 1.

From 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.

CWXU 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.

Additions to CICS RACF category 1 transactions

There are some new CICS internal system transactions added to the list of category 1 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).

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

The new category 1 transactions are:

- CJGC
- CJPI
- CPIR
- CRTP
- CWXU

Chapter 4. Changes to resource definition

This section summarizes the changes to CICS resource definitions.

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

Obsolete CICS-supplied resource definitions

These obsolete resource 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.

DFH$JAVA

CICS-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

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

DFH$AFFY

CICS-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.

Changed resource definition attributes

Resource definition attributes changed in CICS Transaction Server for z/OS, Version 3 Release 2

TCPIPSERVICE resource definition: new values

- On the ATTACHSEC option, a new value of IDENTIFY is added:

  IDENTIFY
  specifies that incoming attach requests must specify a user identifier.

- On the PROTOCOL option, a new value of ISC is added:

  IPIC IPIC protocol is used. Specify IPIC for TCPIPSERVICEs that are to be used for distributed program link (DPL) over IP interconnectivity connections (which are also known as *IPCONNs*).

- On the URM option, a new value of NO is added, and a new user-replaceable program can be specified:

  NO Autoinstall is not permitted on this TCPIPSERVICE. This is only applicable for PROTOCOL(IPIC).

  – For the IPIC protocol, specify the name of the autoinstall user program for IPCONNs, if required. For PROTOCOL(IPIC), if you do not specify this attribute CICS uses the CICS-supplied, default, IPCONNs autoinstall user program, DFHISAIP.

© Copyright IBM Corp. 2001, 2007

Changes to CICS-supplied resource definitions

Some CICS-supplied resource definitions are changed or obsolete, and are moved to new compatibility groups. Following the upgrade of your CSD, the resource definitions listed here 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 the appropriate DFHCOMP compatibility groups in their place. See Chapter 5, “Changes to the application programming interface (API),” on page 33.

Resource definitions moved to group DFHCOMPA

Group DFHJIIRP became obsolete in CICS Transaction Server for z/OS, Version 3 Release 1. The definition that was in this group is now in the compatibility group DFHCOMPA.

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

New resource definition types and new attributes

New resource definition types and attributes added in CICS Transaction Server for z/OS, Version 3 Release 2

New IPCONN resource definition type

Support for IPIC introduces a new CICS resource, the IPIC connection (IPCONN). An IPIC connection is a Transport Control Protocol/Internet Protocol (TCP/IP) communication link to a remote system. An IPCONN definition specifies the *outbound* attributes of the TCP/IP connection. The *inbound* attributes of the connection are specified by the TCPIPSERVICE definition named on the TCPIPSERVICE option of the IPCONN definition. For more information about this new resource definition type, see IPCONN resource definitions, in the *CICS Resource Definition Guide*.

New LIBRARY resource definition type

Support for dynamic program library management introduces a new CICS resource, the LIBRARY. A LIBRARY represents a partitioned data set or sequence of concatenated partitioned data sets containing program entities that make up an application or group of applications (defined by the System Programmer). A LIBRARY definition specifies the name of the LIBRARY, the datasets belonging to that LIBRARY, its CRITICALity, and its RANKING in the overall LIBRARY search order. For more information about this new resource definition type, see LIBRARY resource definitions, in the *CICS Resource Definition Guide*.

PIPELINE resource definition: new attribute

RESPWAIT(*value*)
: Specifies the number of seconds that an application program should wait for a response message from a remote Web service. The value can range from 0 to 9999 seconds.

  If you want to use the default timeout value of the transport protocol, specify `DEFT`.

- The default timeout value for HTTP is 10 seconds.
- The default timeout value for WebSphere MQ is 60 seconds.

TCPIPSERVICE resource definition: new attributes

REALM(*string*)
specifies the realm that is used for HTTP basic authentication. You can only specify this attribute for the HTTP protocol.

The realm is provided by CICS in the WWW-Authenticate header, and is seen by the end user during the process of basic authentication. It identifies the set of resources to which the authentication information requested (that is, the user ID and password) will apply.

If you do not specify a realm, the default used by CICS is `CICS application aaaaaaaa`, where *aaaaaaaa* is the applid of the CICS region.

The realm can be up to 56 characters, and can include embedded blanks. It is specified in mixed case, and the case is preserved. Do not specify opening and closing double quotes, as CICS provides these when assembling the WWW-Authenticate header.

Acceptable characters:

```
A-Z a-z 0-9 $ @ # . - _ % & ? ! : | ' = ¬ + * , ; < > ( )
```

Space characters are also permitted. If parentheses (″(″ and ″)″) are used, you must use them as pairs of opening and closing parentheses.

New resource definition types and attributes added in earlier CICS releases

Table 7. New resource definition types and attributes added in earlier releases

Resource type	New keywords	Explanation
PIPELINE		For information about this new resource definition type, see PIPELINE resource definitions the *CICS Resource Definition Guide*.
URIMAP		For information about this new resource definition type, see URIMAP resource definitions in the *CICS Resource Definition Guide*.
WEBSERVICE		For information about this new resource definition type, see WEBSERVICE resource definitions in the *CICS Resource Definition Guide*.

New CICS-supplied resource definitions

New groups of resource definitions are added to your CSD when you run the UPGRADE command.

DFHDP

The CICS-supplied group DFHDP, introduced in CICS Transaction Server for z/OS, Version 3 Release 1, contains the resource definitions for the 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

The CICS-supplied group DFHDPWB, introduced in CICS Transaction Server for z/OS, Version 3 Release 1, contains the resource definitions for the application debugging profile manager web interface.

The group contains:
PROGRAM definitions
DFHDPWB, DFHDPWM0, DFHDPWM1, DFHDPWM2, DFHDPWM3, DFHDPWM4, DFHDPWM5, DFHDPWM6, DFHDPWT0, and DFHDPWF0

DFHSO

The CICS-supplied group DFHSO, introduced in CICS Transaction Server for z/OS, Version 3 Release 1, contains the resource definitions for external sockets support.

The group contains:
PROGRAM definitions
DFHSOCI and DFHSOLI

DFH$WEB

The CICS-supplied group DFH$WEB, introduced in CICS Transaction Server for z/OS, Version 3 Release 2, contains most of the samples for CICS Web support. The exception is the Assembler sample program DFH$WB1A, which is provided in the existing DFHWEB resource definition group.

DFH$WEB contains:

- PROGRAM resource definitions for:
  - DFH$WB1C, sample C program for verifying the operation of CICS Web support
  - DFH$WBCA, sample Assembler program for sending client requests in chunks and receiving a chunked response
  - DFH$WBCC, sample C program for sending client requests in chunks and receiving a chunked response
  - DFH0WBCO, sample COBOL program for sending client requests in chunks and receiving a chunked response
  - DFH$WBHA, sample Assembler program for a server to receive chunked requests and send a chunked response
  - DFH$WBHC, sample C program for a server to receive chunked requests and send a chunked response

  - DFH0WBHO, sample COBOL program for a server to receive chunked requests and send a chunked response
  - DFH$WBPA, sample Assembler program for pipelining client requests
  - DFH$WBPC, sample C program for pipelining client requests
  - DFH0WBPO, sample COBOL program for pipelining client requests
- Sample URIMAP definitions:
  - DFH$URI1, for accessing DFH$WB1C
  - DFH$URI2, used by the sample programs for pipelining client requests
  - DFH$URI3, used by the sample programs for chunking
  - DFH$URI4, used by the sample programs for chunking

Changes to control tables (macro resource definition)

There are a number of changes to control tables. When migrating to CICS Transaction Server for z/OS, Version 3 Release 2, reassemble all CICS control tables using the CICS TS 3.2 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. You also need to reassemble any DFHCNV data conversion tables that you use, because CICS initialization fails if you try to load DFHCNV tables assembled using macros from an earlier release.

DFHMCT monitoring control table

The new performance class data fields added for CICS Transaction Server for z/OS, Version 3 Release 2 have corresponding new values which can be defined 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. Control data recording-DFHMCT TYPE=RECORD in the *CICS Resource Definition Guide* lists all the fields that can now be included or excluded using the macro.

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.

A new option COMPRESS is available on the DFHMCT TYPE=INITIAL macro. The option can be used to activate data compression for monitoring records. The default is NO, meaning that data compression is not performed. YES specifies that you do want monitoring record data compression to be performed.

Changes to DFHTCT control table

CICS no longer supports local TCAM terminals. The TYPE=SDSCI, TYPE=LINE and TYPE=TERMINAL resource definition macros can no longer be used to define local TCAM terminals.

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.

Chapter 5. Changes to the application programming interface (API)

CICS provides API compatibility from release to release, but functional changes to some CICS components can affect some of the CICS API commands.

Program compatibility

Except for the specific cases described in this section, CICS Transaction Server 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 High level language support in the *CICS Transaction Server for z/OS Release Guide*.

Obsolete API commands and options

Table 8. Obsolete API commands and options

Commands	Changes
• 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.

Changed API commands

Changes to API commands in CICS Transaction Server for z/OS, Version 3 Release 2

DOCUMENT CREATE and DOCUMENT SET: There is a new error condition NOTAUTH, which might be received if resource security for document templates is active in the CICS region.

GET CONTAINER CHANNEL: new INTOCODEPAGE option and other changes

© Copyright IBM Corp. 2001, 2007

INTOCODEPAGE(data-value)
specifies an IANA-registered alphanumeric charset name or a Coded Character Set Identifier (CCSID) for the code page into which the character data in the container is to be converted, using up to 40 alphanumeric characters, including appropriate punctuation. Use this option instead of the CCSID option if you prefer to use an IANA-registered charset name, as specified in the Content-Type header for an HTTP request. CICS converts the IANA name into a CCSID, and the subsequent data conversion process is identical. Also use this option if you prefer to specify the CCSID in alphanumeric characters, rather than as a fullword binary number.

Where an IANA name exists for a code page and CICS supports its use, the name is listed with the CCSID in the *CICS Family: Communicating from CICS on zSeries*.

The description of the INTOCCSID option has changed, and there is a new error condition CODEPAGEERR with a number of new RESP2 values.

PUT CONTAINER CHANNEL: new FROMCODEPAGE option and other changes

FROMCODEPAGE(data-value)
specifies an IANA-registered alphanumeric charset name or a Coded Character Set Identifier (CCSID) for the current code page of the character data to be put into the container, using up to 40 alphanumeric characters, including appropriate punctuation. Use this option instead of the CCSID option if you prefer to use an IANA-registered charset name, as specified in the Content-Type header for an HTTP request. CICS converts the IANA name into a CCSID, and the subsequent data conversion process is identical. Also use this option if you prefer to specify the CCSID in alphanumeric characters, rather than as a fullword binary number.

FROMCCSID and FROMCODEPAGE are effective only on the PUT CONTAINER command that creates the container. This is because, for CHAR containers, the data is stored in the CCSID specified on the original PUT CONTAINER command that created the container. If you want to use a different CCSID, the application program should delete the existing container before issuing the new PUT CONTAINER command, thus recreating the container.

Where an IANA name exists for a code page and CICS supports its use, the name is listed with the CCSID in the *CICS Family: Communicating from CICS on zSeries*.

If FROMCODEPAGE is specified, DATATYPE(DFHVALUE(CHAR)) is implied.

If FROMCCSID and FROMCODEPAGE are not specified, but a DATATYPE of CHAR is specified, the value for conversion defaults to the CCSID of the region (or, for CICS-created channels, the CCSID of the channel). The default CCSID of the region is specified on the LOCALCCSID system initialization parameter.

The description of the FROMCCSID option has changed, and there is a new error condition CODEPAGEERR with a number of new RESP2 values.

QUERY SECURITY: The QUERY SECURITY command can now be used to determine whether the user has access to the resource definitions for CICS document templates (DOCTEMPLATE).

READ, READNEXT, READPREV, RESETBR, STARTBR, and WRITE: new XRBA option

XRBA
specifies that the record identification field specified in the RIDFLD option contains an extended relative byte address. This option should be used when reading, browsing, or writing records in an extended ESDS.

If you specify XRBA on a STARTBR command, all other commands within the same browse must also specify XRBA.

On the READ and STARTBR commands, there is a new RESP2 value 59 for the INVREQ response. On the READ, READNEXT, READPREV, RESETBR, and STARTBR commands, there is a new RESP2 value 81 for the NOTFND response.

WEB EXTRACT: new options and other changes

REALM*(data-area)*
specifies the realm or security environment that contains the data you are requesting. If you are issuing a WEB EXTRACT command in response to a HTTP 401 message, REALM is the realm value in the most recently received WWW-Authenticate header.

REALMLEN*(data-area)*
specifies the buffer length supplied for the REALM option, as a fullword binary variable. If you are issuing a WEB EXTRACT command in response to a HTTP 401 message, REALMLEN is the length of the realm name in the most recently received WWW-Authenticate header.

The description of the HTTPVERSION option has changed.

WEB SEND (Client): new options for specifying authentication credentials

AUTHENTICATE*(cvda)*
This option allows you to specify user authentication details (credentials), to control access to restricted data. The CVDA values that apply for CICS as an HTTP client are:

NONE specifies that there are no restrictions on accessing this data, therefore no credentials are required. This is the default value for AUTHENTICATE.

BASICAUTH
specifies that HTTP Basic Authentication credentials are required for this session. These details can be supplied within the command or by using the XWBAUTH global user exit.

PASSWORDLEN*(data-value)*
specifies the buffer length supplied for the PASSWORD option as a fullword binary variable.

PASSWORD*(data-value)*
specifies the password associated with the USERNAME that is allowed access to this data. The PASSWORD option is only required if the USERNAME option is used.

USERNAME*(data-value)*
specifies the user ID or logon name that is allowed access to this data. If the USERNAME is specified, you also need to use the PASSWORD option.

USERNAMELEN*(data-value)*
specifies the buffer length supplied for the USERNAME option as a fullword binary variable.

WEB SEND (Server and Client) and WEB CONVERSE: new DOCSTATUS option

DOCSTATUS*(cvda)*
indicates whether the document should be deleted or not deleted during processing of the WEB SEND command. The CVDA values are:

DOCDELETE
CICS deletes the document after the document contents are saved for sending. Storage allocated for the document is released immediately. If you make subsequent requests for the document, these generate a TOKENERR response.

NODOCDELETE
CICS does not delete the document during processing of the WEB SEND command. This is the default value for DOCSTATUS.

WEB OPEN: The description of the HTTPRNUM and HTTPVNUM options has changed.

WEB RETRIEVE: If a WEB SEND command specifies the option DOCSTATUS(DOCDELETE), the WEB RETRIEVE command cannot retrieve the document, and a NOTFND response with a RESP2 value of 1 is returned.

Changes to API commands in earlier CICS releases

Table 9. Changed API commands in earlier CICS releases

Commands	Changes
• WEB WRITE HTTPHEADER • WEB READ HTTPHEADER • WEB STARTBROWSE HTTPHEADER • WEB READNEXT HTTPHEADER • WEB ENDBROWSE HTTPHEADER • WEB EXTRACT • FORMATTIME • EXTRACT TCPIP	A number of new options are available on these commands.
• VERIFY PASSWORD	When the command EXEC CICS VERIFY PASSWORD is issued, CICS now enforces the revoked status of a user ID or a user's group connection. "Changes to EXEC CICS VERIFY PASSWORD" on page 37 explains the consequences for your application programs.

New API commands

New API commands added in CICS Transaction Server for z/OS, Version 3 Release 2

EXEC CICS DOCUMENT DELETE

DOCUMENT DELETE

Conditions: NOTFND

This command is threadsafe.

New API commands added in earlier CICS releases

Table 10. New API commands added in earlier CICS releases

Commands	Function
• 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	Commands for channels and containers.
• EXEC CICS INVOKE WEBSERVICE • EXEC CICS SOAPFAULT ADD • EXEC CICS SOAPFAULT CREATE • EXEC CICS SOAPFAULT DELETE	Commands for Web services.
• EXEC CICS CONVERTTIME • EXEC CICS WEB OPEN • EXEC CICS WEB CONVERSE • EXEC CICS WEB CLOSE • EXEC CICS WEB PARSE URL • EXEC CICS WEB SEND (CLIENT) • EXEC CICS WEB RECEIVE (CLIENT)	Commands for CICS as an HTTP client. Some also apply to CICS as an HTTP server.

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 Transaction Server for z/OS, Version 3 Release 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.

Changes to sample programs

There are a number of changes to the samples provided with CICS Transaction Server for z/OS, Version 3 Release 2 to demonstrate the use of the EXEC CICS API commands. Unless otherwise stated, sample programs are supplied in the SDFHSAMP library.

Verification samples, DFH$WB1A and DFH$WB1C

The sample programs for verifying the operation of CICS Web support, DFH$WB1A (Assembler) and DFH$WB1C (C), are updated to use the EXEC CICS WEB commands. In addition, a sample URIMAP definition DFH$URI1 is provided, which can be used to access DFH$WB1C. The CICS-supplied sample analyzer DFHWBADX can be used to access both DFH$WB1A and DFH$WB1C.

The PROGRAM resource definition for DFH$WB1C, and the URIMAP definition DFH$URI1, are provided in the new DFH$WEB resource definition group. DFH$WB1A is provided in the DFHWEB resource definition group, which is installed as part of DFHLIST.

Pipelining samples, DFH$WBPA (Assembler), DFH$WBPC (C), and DFH0WBPO (COBOL)

New sample programs are provided to demonstrate how CICS can pipeline client requests to an HTTP server. The sample programs use the sample client URIMAP definition, DFH$URI2, to pipeline requests to a CICS region which has been set up as an HTTP server, to be handled there by the verification sample program DFH$WB1C.

The PROGRAM resource definitions for the pipelining sample programs, and the URIMAP definition DFH$URI2, are provided in the new DFH$WEB resource definition group.

Chunking samples: DFH$WBHA and DFH$WBCA (Assembler), DFH$WBHC and DFH$WBCC (C), DFH0WBHO and DFH0WBCO (COBOL)

New sample programs DFH$WBCA (Assembler), DFH$WBCC (C), and DFH0WBCO (COBOL) demonstrate how CICS, as an HTTP client, can send a request in sections or chunks to an HTTP server, and receive a chunked message in response. New sample programs DFH$WBHA (Assembler), DFH$WBHC (C), and DFH0WBHO (COBOL) demonstrate how CICS, as an HTTP server, can receive a request in chunks from an HTTP client and send a chunked response.

The sample programs send and receive requests between CICS regions, in which CICS Web support is running. The client chunking samples (DFH$WBCA,

DFH$WBCC and DFH0WBCO) are handled by DFH$WBHA, the Assembler server chunking sample (you can update the server URIMAP to point at a different server program if required). The PROGRAM resource definitions for the chunking sample programs, and the URIMAP definitions DFH$URI3 and DFH$URI4, are provided in the DFH$WEB resource definition group.

Chapter 6. Changes to the system programming interface (SPI)

This section summarizes the changes affecting the CICS system programming interface (SPI).

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 section 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 section, CICS continues to provide compatibility with future releases, at source and object level, for application programs that use the unaffected SPI commands.

Obsolete options or values on SPI commands

The system programming interface command options or values listed in this topic are obsolete. You should remove these options from your applications, because they represent functions that are no longer available, so the behavior of applications that use these options will change.

Obsolete options or values on SPI commands in CICS Transaction Server for z/OS, Version 3 Release 2

INQUIRE CLASSCACHE: The value RESET is no longer returned for the REUSEST option. If the shared class cache is not started, the value UNKNOWN is displayed. In this situation, CICS cannot identify the reuse status, but when the shared class cache is started, the status always becomes REUSE.

INQUIRE JVM: The value RESET is no longer returned for the REUSEST option.

INQUIRE JVMPROFILE: The value RESET is no longer returned for the REUSEST option.

INQUIRE MONITOR: The SUBSYSTEMID option is obsolete.

SET JVMPOOL: The TERMINATE function on the SET JVMPOOL command is now deprecated. You should use the improved function on the PERFORM JVMPOOL command instead.

© Copyright IBM Corp. 2001, 2007

Obsolete options on system programming commands

Table 11. Obsolete options on system programming commands

Command	Obsolete options	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.
CREATE TYPETERM	XRFSIGNOFF	This XRF-specific attribute is replaced by RSTSIGNOFF, which acts for either signon retention for VTAM® persistent sessions or XRF.
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	JVMDEBUG	The JVMDEBUG option is obsolete and CICS always returns NODEBUG as the CVDA value if JVMDEBUG is specified.
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.

Obsolete CVDA values

The CVDA values listed in this topic are obsolete. You should remove these CVDA values from your applications, because they represent functions that are no longer available, so the behavior of applications that use these CVDA values will change.

Table 12. Obsolete CVDA values

CVDA	value
HOTPOOL	1065
NOTHOTPOOL	1066

New options or values on SPI commands

The system programming interface commands listed in this topic have new options or new values for options.

New options or values on SPI commands in CICS Transaction Server for z/OS, Version 3 Release 2

CREATE PIPELINE: new RESPWAIT option

RESPWAIT(*value*)
Specifies the number of seconds that an application program should wait for a response message from a remote Web service. The value can range from 0 to 9999 seconds.

If you want to use the default timeout value of the transport protocol, specify `DEFT`.

- The default timeout value for HTTP is 10 seconds.
- The default timeout value for WebSphere MQ is 60 seconds.

CREATE TCPIPSERVICE: new options and values

REALM(*data-area***)**
returns the 56-character realm that is used during the process of HTTP basic authentication. This value is returned only when PROTOCOL has a value of HTTP. If no realm is specified for this service, the default realm used by CICS is returned, which is `CICS application aaaaaaaa`, where *aaaaaaaa* is the applid of the CICS region.

- New value on the ATTACHSEC option:

  IDENTIFY
  specifies that incoming attach requests must specify a user identifier.

- New value on PROTOCOL option:

  IPIC IPIC protocol is used. Specify IPIC for TCPIPSERVICEs that are to be used for distributed program link (DPL) over IP interconnectivity connections (which are also known as *IPCONNs*).

ENABLE PROGRAM: The QUASIRENT and THREADSAFE options now apply to global user exit programs, as well as to task-related user exit programs. The meaning of value `X'802000'`, which may be returned on the INVEXITREQ condition of the EXEC CICS ENABLE command, has changed.

EXTRACT STATISTICS: new values on the RESTYPE option

DOCTEMPLATE
Requests statistics for a document template; RESID identifies the particular DOCTEMPLATE resource definition.

IPCONN
Requests statistics for an IPCONN.

LIBRARY
Requests statistics for a LIBRARY resource

MQCONN
Requests statistics for a WebSphere MQ Connection.

INQUIRE DOCTEMPLATE: new CACHESIZE option

CACHESIZE(*data-area***)**
returns a fullword binary field giving the amount of storage, in bytes, used by the cached copy of the document template. A value of zero is returned if there is no cached copy of the template at the time of the inquiry.

INQUIRE FILE: new RBATYPE option

RBATYPE(*cvda*)
returns a CVDA value identifying whether, for VSAM files, the data set uses extended addressing. CVDA values are:

EXTENDED
This VSAM data set uses extended relative byte addressing and therefore can hold more than 4 gigabytes of data.

NOTAPPLIC
One of the following is true:
- The data set is BDAM.
- The file is remote.
- The file is not open.

NOTEXTENDED
This VSAM data set does not use extended relative byte addressing and therefore cannot hold more than 4 gigabytes of data.

INQUIRE IRC: new XCFGROUP option

XCFGROUP(*data-area*)
returns the 8-character name of the cross-system coupling facility (XCF) group of which this region is a member.

If this region is not a member of an XCF group (because it has not signed on to IRC), XCFGROUP contains the XCF group the region would be in if XCF were opened.

For introductory information about XCF/MRO, see Cross-system multiregion operation (XCF/MRO) in the *CICS Intercommunication Guide*.

INQUIRE MONITOR: new COMPRESSST option

COMPRESSST(*cvda*)
returns a CVDA value indicating whether data compression is active for the CICS SMF 110 monitoring records output by the CICS monitoring facility. CVDA values are:

COMPRESS
Data compression is being performed for the monitoring records.

NOCOMPRESS
Data compression is not being performed for the monitoring records.

INQUIRE MVSTCB: new syntax

The INQUIRE MVSTCB command has a new syntax, which uses the new option SET and the existing option NUMELEMENTS to provide a list of descriptors for individual storage elements owned by the TCB which you are browsing. The descriptors contain the new information about the storage key and storage in use for each storage element, as well as the information which was formerly provided about addresses, lengths and MVS subpools for each element.

The options ELEMENTLIST, LENGTHLIST and SUBPOOLLIST are now obsolete, but are supported for compatibility with applications developed in releases before CICS Transaction Server for z/OS, Version 3 Release 2. These options do not provide the new information about the storage key and storage in use for each element. You cannot use these options in combination with the new SET option. All new applications should use the new syntax with the SET option.

INQUIRE PROGRAM: new and changed options

- The new LIBRARY option displays the 8-character name of the LIBRARY resource from which the program was loaded.
- The new LIBRARYDSN option displays the 44-character name of the data set from which the program was loaded.
- The USECOUNT option now displays a use count for Java programs. In earlier CICS releases, this count was not available, and a value of -1 was returned.
- The CONCURRENTST option now applies to global user exit programs, as well as to task-related user exit programs.

INQUIRE PIPELINE: new options

CIDDOMAIN(*data-area*)
Returns the domain name that is used to generate MIME content-ID values to identify binary attachments in containers. The name can be up to 255 characters long.

MODE(*cvda*)
Returns the operating mode of the pipeline. CVDA values are:

PROVIDER
CICS is using the pipeline as a service provider.

REQUESTER
CICS is using the pipeline as a service requester.

UNKNOWN
The operating mode of the pipeline cannot be determined.

MTOMNOXOPST(*cvda*)
Returns a value that indicates whether MTOM should be used for outbound SOAP messages when there are no binary attachments present. The values are:

MTOMNOXOP
Use MTOM, even when there are no binary attachments present.

NOMTOMNOXOP
Do not use MTOM unless there are binary attachments present.

MTOMST(*cvda*)
Returns a value that indicates whether support for MTOM has been enabled in the pipeline. The values are:

MTOM
MTOM support has been enabled in the pipeline.

NOMTOM
MTOM support has not been enabled in the pipeline.

RESPWAIT(*data-area*)
Returns the number of seconds that an application program waits for an optional response message from a remote Web service. If the returned value is -1, no value has been set for the pipeline and the default timeout value of the transport protocol is being used.

- The default timeout value for HTTP is 10 seconds.
- The default timeout value for WebSphere MQ is 60 seconds.

SENDMTOMST(*cvda*)
Returns a value that indicates when MTOM should be used for outbound SOAP messages. The values are:

NOSENDMTOM
Do not use MTOM for outbound SOAP messages.

SAMESENDMTOM
Use MTOM for outbound SOAP message responses when the inbound message is received in MTOM format.

SENDMTOM
Always use MTOM for outbound SOAP messages.

SOAPLEVEL(*data-area*)
Returns an eight byte character string of the SOAP level that is used in the PIPELINE. The value of the SOAP level is 1.1 or 1.2. If the pipeline is not being used for SOAP messages, a value of NOTSOAP is returned.

SOAPRNUM(*data-area*)
Returns a fullword binary value of the release number for the SOAP level that is used in the PIPELINE. The value of the release number is 1 or 2.

SOAPVNUM(*data-area*)
Returns a fullword binary value of the version number for the SOAP level that is used in the PIPELINE. The value of the version number is 1.

XOPDIRECTST(*cvda*)
Returns a value that indicates whether the pipeline can currently handle XOP documents in direct mode. The values are:

XOPDIRECT
The pipeline supports the direct processing of XOP documents and binary attachments.

NOXOPDIRECT
The pipeline does not support the direct processing of XOP documents and binary attachments. Compatibility mode is in operation.

XOPSUPPORTST(*cvda*)
Returns a value that indicates whether the application handler for the pipeline supports the processing of XOP documents and binary attachments. The values are:

XOPSUPPORT
The application handler supports XOP documents.

NOXOPSUPPORT
The application handler does not support XOP documents.

INQUIRE SYSTEM: new options

MEMLIMIT(*data-area***)**
returns a doubleword binary field giving the maximum amount, in bytes, of storage available above the 2GB boundary (above the bar), for use by the CICS region. A value of -1 indicates that no limit has been imposed on the amount of storage that the region can attempt to use (also known as NOLIMIT). The MEMLIMIT value can be set as a PARMLIB member, by JCL or through the IEFUSI global user exit.

SOSABOVEBAR(*cvda***)**
returns a CVDA value indicating whether CICS is short on storage in the dynamic storage areas above the 2GB boundary (above the bar).

NOTSOS
CICS is not short on storage in any of the dynamic storage areas above the 2GB boundary.

SOS CICS is short on storage in at least one of the dynamic storage areas above the 2GB boundary.

SOSABOVELINE(*cvda***)**
returns a CVDA value indicating whether CICS is short on storage in the dynamic storage areas above the 16MB line, but below the 2GB boundary.

NOTSOS
CICS is not short on storage in any of the dynamic storage areas above the 16MB line (but below the 2GB boundary).

SOS CICS is short on storage in at least one of the dynamic storage areas above the 16MB line (but below the 2GB boundary).

SOSBELOWLINE(*cvda***)**
returns a CVDA value indicating whether CICS is short on storage in the dynamic storage areas below the 16MB line.

NOTSOS
CICS is not short on storage in any of the dynamic storage areas below the 16MB line.

SOS CICS is short on storage in at least one of the dynamic storage areas below the 16MB line.

INQUIRE SUBPOOL: The new DSA name GCDSA (above the bar CICS dynamic storage area) is added to the values returned by the DSANAME option.

INQUIRE TASK: new options

IPFACILITIES(*ptr-ref***)**
returns the address of a list of 4-byte binary tokens, each of which identifies an IPCONN session that the task is using to communicate with another system. If there are no such IP facilities for this task, the IPFACILITIES pointer contains a null value.

CICS obtains the storage for the list and frees it when the inquiring task issues another INQUIRE TASK command or ends; the task cannot free the storage itself.

IPFLISTSIZE(*data-area***)**
returns a fullword binary field giving the number of IP facilities associated with this task. (That is, it returns the number of items in the list addressed by the IPFACILITIES option.)

If this task has no IP facilities, IPFLISTSIZE contains zero.

INQUIRE TCPIPSERVICE: new options and values

REALM(*data-area***)**
returns the 56-character realm that is used during the process of HTTP basic authentication. This value is returned only when PROTOCOL has a value of HTTP. If no realm is specified for this service, the default realm used by CICS is returned, which is `CICS application aaaaaaaa`, where *aaaaaaaa* is the applid of the CICS region.

- New value on the ATTACHSEC option:

  IDENTIFY
  Incoming attach requests must specify a user identifier. (IDENTIFY is used when the connecting system has a security manager; for example, if it is another CICS region.)

- New value on PROTOCOL option:

  IPIC IP interconnectivity (IPIC).

INQUIRE WEBSERVICE: new options

CCSID(*data-area*)
Returns the CCSID that is used to encode the character data in the application data structure at run time. This value is set using the optional **CCSID** parameter in the Web services assistant when the Web serving binding file was generated. If the *data-area* is 0, the default CCSID for the CICS region that is specified by the **LOCALCCSID** system initialization parameter is used.

MAPPINGLEVEL(*data-area*)
Returns an eight byte character string of the mapping level that is used to convert data between language structures and Web service description (WSDL) documents. The value of the mapping level is 1.0, 1.1, 1.2 or 2.0.

MAPPINGRNUM(*data-area*)
Returns a fullword binary value of the release number for the mapping level that is used to convert data between language structures and Web service description (WSDL) documents. The value of the release number is 0, 1, or 2.

MAPPINGVNUM(*data-area*)
Returns a fullword binary value of the version number for the mapping level that is used to convert data between language structures and Web service description (WSDL) documents. The value of the version number is 1 or 2.

MINRUNLEVEL(*data-area*)
Returns an eight byte character string of the minimum runtime level that is required to run the Web service in CICS. The value of the runtime level is 1.0, 1.1, 1.2 or 2.0.

MINRUNRNUM(*data-area*)
Returns a fullword binary value of the release number for the minimum runtime level that is required to run the Web service in CICS. The value of the release number is 0, 1, or 2.

MINRUNVNUM(*data-area*)
Returns a fullword binary value of the version number for the minimum runtime level that is required to run the Web service in CICS. The value of the version number is 1 or 2.

XOPDIRECTST(*cvda*)
Returns a value that indicates whether the web service is currently able to handle XOP documents in direct mode. The values are:

NOXOPDIRECT
The web service cannot currently handle XOP documents and binary attachments directly. This is true when the web service implementation does not support the direct handling of XOP documents and binary attachments, or Web service validation is switched on.

XOPDIRECT
The web service can currently handle XOP documents and binary attachments directly. This is true when the web service implementation supports the direct handling of XOP documents and Web service validation is not switched on.

XOPSUPPORTST(*cvda*)
Returns a value that indicates whether the web service implementation is capable of handling XOP documents and binary attachments in direct mode. The values are:

NOXOPSUPPORT
The web service implementation does not support the direct handling of XOP documents and binary attachments.

XOPSUPPORT
The web service implementation supports the direct handling of XOP documents and binary attachments. This is true for any web services that are generated and deployed using the Web services assistant.

PERFORM STATISTICS RECORD: new statistics types

DOCTEMPLATE
records specific statistics for each document template installed in the CICS region.

IPCONN
records specific statistics for all IPIC connections installed in the CICS region.

LIBRARY
records specific statistics for all LIBRARY resources.

MQCONN
records global statistics for the WebSphere MQ connection.

SET MONITOR: new COMPRESSST option

COMPRESSST(*cvda***)**
specifies whether you want data compression to be performed for the CICS SMF 110 monitoring records output by the CICS monitoring facility. If you change the setting for the data compression option, the new setting applies to all monitoring records written from that point on, even if they are for a task being processed at the time the change is made. The new setting also applies to any records which are in the buffer waiting to be written to SMF at the time the change is made. The change only applies until a CICS restart.

COMPRESS
CICS is to perform data compression for the monitoring records. (In some situations, some of the records might not be compressed.)

NOCOMPRESS
CICS is not to perform data compression for the monitoring records.

SET PIPELINE: new RESPWAIT option

RESPWAIT(*data-area*)
Specifies the number of seconds that an application program should wait for an optional response message from a remote Web service. The value can range from 0 to 9999 seconds. If you do not specify a value, the default timeout value of the transport protocol is used.

- The default timeout value for HTTP is 10 seconds.
- The default timeout value for WebSphere MQ is 60 seconds.

New options or values on SPI commands in earlier CICS releases

See System commands in the *CICS System Programming Reference* for information on these new options.

Table 13. New options or values on SPI commands in earlier CICS releases

Command	Option	Explanation
INQUIRE WORKREQUEST	WORKTYPE	A new value, SOAP, is added.
PERFORM STATISTICS RECORD	PIPELINE	Write statistics for these new resource types.
SET WORKREQUEST	WORKTYPE	A new value, SOAP, is added

New SPI commands

For information on the new commands, see System commands the *CICS System Programming Reference*.

For information about the new function relating to these new commands, see What's new in the *CICS Transaction Server for z/OS Release Guide*.

New system programming commands added in CICS Transaction Server for z/OS, Version 3 Release 2

CREATE IPCONN

Define and install an IPCONN in the local CICS region.

CREATE LIBRARY

Create a LIBRARY resource in the local CICS region.

DISCARD IPCONN

Remove an IPCONN definition.

DISCARD LIBRARY

Remove a specified LIBRARY from the running CICS system.

INQUIRE ASSOCIATION

Retrieve association information for a specified task from its associated data control block (ADCB).

INQUIRE ASSOCIATIONLIST

Retrieve a list of tasks, based on user correlation data contained in the tasks' association information.

INQUIRE IPCONN

Retrieve information about an IPCONN.

INQUIRE LIBRARY

Retrieve information about a LIBRARY.

SET IPCONN

Change the attributes of an IPCONN or cancel outstanding AIDs.

SET LIBRARY

Change the attributes of a LIBRARY resource.

PERFORM JVMPOOL

Start and terminate JVMs in the JVM pool.

SET DOCTEMPLATE

Refresh the cached copy of a document template installed in your CICS region, or phase in a new copy of a CICS program or exit program that is defined as a document template.

New system programming commands added in earlier CICS releases

Table 14. New system programming commands added in earlier CICS releases

Command	Description
CREATE PIPELINE	Define a PIPELINE in the local CICS region.
CREATE URIMAP	Define a URIMAP in the local CICS region.
CREATE WEBSERVICE	Define a WEBSERVICE in the local CICS region.
DISCARD PIPELINE	Remove a PIPELINE from the CICS system and the CICS catalog.
DISCARD URIMAP	Remove a URIMAP from the CICS system and the CICS catalog.
DISCARD WEBSERVICE	Remove a WEBSERVICE from the CICS system and the CICS catalog.
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.
INQUIRE PIPELINE	Retrieve information about an installed PIPELINE.
INQUIRE URIMAP	Retrieve information about URIMAP resource definitions.
INQUIRE WEBSERVICE	Retrieve information about an installed WEBSERVICE.
PERFORM PIPELINE	Initiate a scan of the Web service binding directory that is specified in the WSBIND attribute of the PIPELINE definition.
SET HOST	Enable or disable a virtual host.
SET PIPELINE	Enable or disable a PIPELINE.
SET URIMAP	Enable or disable a URIMAP definition, and apply or remove redirection for a URIMAP definition.

Table 14. New system programming commands added in earlier CICS releases (continued)

Command	Description
SET WEBSERVICE	Set the validation status of a WEBSERVICE.

Changes to DOCTEMPLATE SPI commands

If resource security for document templates is active in the CICS region, with the XRES system initialization parameter set on (which is the default), the SPI commands for CICS document templates can be affected.

The following commands can be affected:

- EXEC CICS CREATE DOCTEMPLATE
- EXEC CICS DISCARD DOCTEMPLATE
- EXEC CICS INQUIRE DOCTEMPLATE
- EXEC CICS SET DOCTEMPLATE

Document templates specified by these commands are subject to resource security checking if RESSEC(YES) is specified in the transaction resource definition for the transaction that issues the command.

If resource security checking is in place, the user ID for the transaction must have an appropriate level of access to the DOCTEMPLATE resource definition involved:

Table 15. Access required for system programming commands involving document templates

Action	Access required
CREATE	ALTER
DISCARD	ALTER
INQUIRE	READ
SET	UPDATE

For all these commands, there is a new RESP2 value 101 for the condition NOTAUTH:

NOTAUTH

101 The user associated with the issuing task is not authorized to access this DOCTEMPLATE resource definition in the way required by this command.

The XHFS system initialization parameter, which specifies access control for z/OS UNIX files, does not affect any system programming commands. z/OS UNIX files are only referenced by EXEC CICS commands when they are defined as CICS document templates. In this situation, resource security for CICS document templates (specified by the XRES system initialization parameter) controls access to them for users.

Chapter 7. Changes to CICS-supplied utility programs

This section summarizes the changes to the utilities supplied by CICS.

CSD utility program, DFHCSDUP

The CSD utility program is enhanced to support all the new and changed resource types and attributes. See Chapter 4, “Changes to resource definition,” on page 27 for details of all the changes to CSD resource definitions that are supported by DFHCSDUP.

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.

Statistics formatting utility program, DFHSTUP

The statistics formatting utility program is enhanced to format additional statistics reports for the new resource types. See Chapter 12, “Changes to CICS statistics,” on page 79 for information about the new keywords available on the SELECT TYPE and IGNORE TYPE parameters.

Sample statistics utility program, DFH0STAT

DFH0STAT, the sample statistics utility program, is enhanced to produce additional statistics reports.

DFH0STAT now displays a 4 digit count for the hours in time fields instead of a 2 digit count, and also displays the time to 6 decimal places (down to one microsecond) instead of 5 decimal places. The new format for the time fields is `hhhh:mm:ss.000000`. The new format is used in the Dispatcher TCB Modes Report.

DFH0STAT now provides a new report for LIBRARYs, which will provide details about the data sets in the concatenation as well as the LIBRARY statistics. The DFHRPL Analysis currently provided by DFH0STAT has been revised to provide a DFHRPL and LIBRARY analysis showing the programs from the DFHRPL concatenation and those from LIBRARYs.

Trace formatting utility program, DFHTU650

The trace formatting utility program is renamed to DFHTU650. 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 DFHTU650 on the **TYPETR** parameter for these functional areas are the same as the CETR trace component codes.

© Copyright IBM Corp. 2001, 2007

IPCS dump exit routine, DFHPD650

The dump formatting utility program is renamed to DFHPD650. 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.

New CICS JVM Application Isolation Utility

The CICS JVM Application Isolation Utility is provided to help system administrators and application programmers discover static variables in Java applications which they use, or plan to use, in their CICS regions. The application developers should then review the findings of the utility and determine whether the application might exhibit unintended behavior when it runs in a continuous JVM. The utility can be used when migrating Java workloads from resettable to continuous JVMs.

The CICS JVM Application Isolation Utility is shipped with CICS Transaction Server for z/OS, Version 3 Release 2 as a JAR file named `dfhjaiu.jar`. It runs under z/OS UNIX System Services as a standalone utility. You do not need to have a CICS Transaction Server for z/OS, Version 3 Release 2 region or any other CICS region running when you use the utility.

The CICS JVM Application Isolation Utility is a code analyzer tool which inspects Java bytecodes in Java Archive (JAR) files and class files. It does not alter any Java bytecodes. It is provided as a means to help identify potential issues before they arise in a continuous JVM under CICS. The Java application does not need to be running in a CICS region when it is inspected.

Chapter 8. Changes to global user exits, task-related user exits, and the exit programming interface

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 Transaction Server for z/OS, Version 3 Release 2 libraries. You will have to modify some of them for changes to parameters, before they are reassembled.

Note the changes summarized in this section 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.

Impact of changes to the exit programming interface (XPI)

Any changes to the exit programming interface (XPI) listed here mean that you might need to make changes to global user exit programs that contain XPI calls. Check whether your global user exit programs are affected by the changes to the XPI, and modify your programs accordingly.

Obsolete global user exit points

Table 16. 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 2 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 2 does not support the TCAM/ACB interface, and it only supports the TCAM/DCB interface indirectly.

Changes to the DFHUEPAR standard parameter list

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.

© Copyright IBM Corp. 2001, 2007

Table 17. TCB indicators changed in 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

Changes to global user exits

Threadsafe PLT-enabled global user exit programs

Global user exit programs that are enabled from first-phase PLT programs can now be defined as threadsafe. In previous CICS releases, this technique was available to task-related user exit programs but not to global user exit programs. To define a first-phase PLT global user exit program as threadsafe, specify the THREADSAFE keyword on the EXEC CICS ENABLE command. This overrides the CONCURRENCY(QUASIRENT) setting on the system-autoinstalled program definition.

Changes to resource management install and discard exit XRSINDI

The range of values in the 1-byte field addressed by the UEPIDTYP parameter is extended to cover the install and discard of the following new resource types:

UEIDLBRY
A LIBRARY resource

UEIDIPCO
An IPIC connection (“IPCONN”)

UEIDPIPE
A pipeline (PIPELINE)

UEIDURIM
A URI map (URIMAP)

UEIDWEBS
A Web service (WEBSERVICE)

Changes to global user exits in CICS Transaction Server for z/OS, Version 3 Release 2

XEIIN, XEIOUT, XEISPIN, and XEISPOUT, EXEC interface program exits

A new parameter, UEP_EI_PBTOK, is added to the exit-specific parameter lists of these exits:

UEP_EI_PBTOK

Address of a 4-byte field containing the z/OS Workload Manager (WLM) Performance Block Token. An exit program can use this token to access information (such as the service class token, SERVCLS, or the current EWLM correlator, EWLM_CHCORR) in the WLM Performance Block. To do so, it must use the WLM EXTRACT macro, IWMMEXTR, passing the Performance Block Token as the MONTKN input parameter. The following example shows how to do this.

```
*        Mapping of the PB and eWLM constants
         IWMYCON

*        Area to return the Current Correlator
         ecurcorr DS    XL(L'PB_EWLM_CURRENTCORRELATOR)   Current correaltor

*        Code to return the Current Correlator
         XC    ecurcorr,ecurcorr            Clear return area
         space 1
         L     R2,UEP_EI_PBTOK              Address the A(PB)
         space 1
         IWMMEXTR MONTKN=(R2),EWLM_CHCORR=ecurcorr
         space 1
         LTR   R15,R15                      Extract worked ?
         BE    DOSPIMOK                      Yes
         B     DOSPIMER                      No
         space 1
```

Figure 1. Using the IWMMEXTR macro to extract the current EWLM correlator

An exit program must not attempt to modify the Performance Block: if it does so, the results are unpredictable.

XFCFRIN and XFCFROUT, file control domain exits

- A new value of UEP_FC_XRBA may be returned in the UEP_FC_RECORD_ID_TYPE exit-specific parameter.

  UEP_FC_XRBA
  VSAM extended ESDS access

- The following new return codes may be returned in UEP_FC_REASON:

  UEP_FC_REASON_KSDS_AND_XRBA
  Extended relative byte addressing (XRBA) was specified with a KSDS, CMT, or UMT data set.

  UEP_FC_REASON_NOT_EXTENDED
  Extended relative byte addressing was specified, with an XRBA number greater than 4 gigabytes, but the data set uses standard relative byte addressing (RBA).

XFCREQ and XFCREQC, file control EXEC interface API exits

A new value of X'08' (XRBA) may be returned in the FC_EIDOPT8 field of the EXEC interface descriptor (EID), which is pointed to by the first address in the command-level parameter structure:

FC_EIDOPT8

Indicates whether certain keywords that do not take values were specified on the request.

X'80' DEBKEY specified.

X'40' DEBREC specified.

X'20' TOKEN specified.

X'08' XRBA specified. If the XRBA bit is on, FC_RIDFLD (described in DSECT DFHFCEDS) points to an 8-byte extended relative byte address (XRBA).

XFCLDEL, XFCBFAIL, XFCBOVER, and XFCBOUT, file control exits

If you have exit programs that run at these exit points, you might need to recode them to cope with the format of the new log records that are issued for extended addressing ESDS data sets.

XMEOUT, message domain exit

New parameters are added for CICSPlex SM messages:

UEPCPID
Address of a 3-byte product ID. The possible values are:

DFH CICS messages.

EYU CICSPlex SM messages.

UEPCPDOM
Address of a 2-byte field containing the domain identifier of the message.

UEPCPNUM
Address of a 4-byte field containing the message number.

UEPCPSEV
Address of the message severity code.

XPCREQ, XPCREQC, and XPCERES program control program exits

A new parameter, UEP_PC_PBTOK, is added to the exit-specific parameter lists of these exits:

UEP_PC_PBTOK
Address of a 4-byte field containing the z/OS Workload Manager (WLM) Performance Block Token. An exit program can use this token to access information (such as the service class token, SERVCLS, or the current EWLM correlator, EWLM_CHCORR) in the WLM Performance Block. To do so, it must use the WLM EXTRACT macro, IWMMEXTR, passing the Performance Block Token as the MONTKN input parameter.

An exit program must not attempt to modify the Performance Block: if it does so, the results are unpredictable.

XRMIIN and XRMIOUT, resource manager interface program exits

A new parameter, UEP_RM_PBTOK, is added to the exit-specific parameter lists of these exits:

UEP_RM_PBTOK
Address of a 4-byte field containing the z/OS Workload Manager (WLM) Performance Block Token. An exit program can use this token to access information (such as the service class token, SERVCLS, or the current EWLM correlator, EWLM_CHCORR) in the WLM Performance Block. To do so, it must use the WLM EXTRACT macro, IWMMEXTR, passing the Performance Block Token as the MONTKN input parameter.

An exit program must not attempt to modify the Performance Block: if it does so, the results are unpredictable.

Changes to global user exits in earlier CICS releases

Table 18. Changed global user exits in earlier CICS releases

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 to global user exit points 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 in the *CICS Customization Guide*:

- XFCAREQ
- XFCAREQC
- XFCREQ
- XFCREQC
- XICEREQ
- XICEREQC
- XNQEREQ
- XNQEREQC
- XPCREQ
- XPCREQC
- XTDEREQ
- XTDEREQC
- XTSEREQ
- XTSEREQC

New global user exit points

New global user exit points added in CICS Transaction Server for z/OS, Version 3 Release 2

Application Associated Data exit XAPADMGR

The XAPADMGR exit is for use with distributed transactions. It allows you to add user information to a task's Associated Data Origin Descriptor, at the point of origin of the distributed transaction. This information could later be used as, for example, search keys for processing carried out through CICSPlex SM.

CICS provides a sample global user exit program, DFH$APAD, for use at the XAPADMGR exit point. The exit program is invoked, if enabled, when non-system tasks for which no input Origin Descriptor Record is provided are attached.

HTTP client send exit XWBAUTH

XWBAUTH enables you to specify basic authentication credentials (username and password) for a target server. XWBAUTH passes these to CICS on request, to create an Authorization header. XWBAUTH is called during processing of an EXEC CICS WEB SEND (Client) or EXEC CICS WEB CONVERSE command. The host name and path information are passed to the user exit, with an optional qualifying realm.

The XISQUE exit, for managing intersystem queues on IPIC connections

You can use the XISQUE exit to control the number of queued distributed program link (DPL) requests for sessions on IP interconnectivity (IPIC) connections.

The XISQUE exit enables you to detect queuing problems (bottlenecks) early. It is invoked only for DPL requests across IPCONNs.

XISQUE enables allocate requests to be queued or rejected, depending on the length of the queue. It also allows an IPCONN on which there is a bottleneck to be terminated and then re-established.

New global user exit points added in earlier CICS releases

Table 19. New global user exit points added in earlier CICS releases

Module or domain	Exit name	When invoked
HTTP client open exit	XWBOPEN	XWBOPEN is invoked during processing of an EXEC CICS WEB OPEN command. It enables systems administrators to specify proxy servers that should be used for HTTP requests by CICS as an HTTP client, and to apply a security policy to the host name specified for those requests.
HTTP client send exit	XWBSNDO	XWBSNDO is invoked during processing of an EXEC CICS WEB SEND or EXEC CICS WEB CONVERSE command for an HTTP request by CICS as an HTTP client. It enables systems administrators to specify a security policy for HTTP requests by CICS as an HTTP client.

Changes to the exit programming interface

The INQUIRE_SHORT_ON_STORAGE call

A new output parameter, SOS_ABOVE_THE_BAR, has been added to the INQUIRE_SHORT_ON_STORAGE storage control call.

SOS_ABOVE_THE_BAR(NO|YES),
returns YES if CICS is currently short-on-storage above the 2GB boundary, and NO if not.

Changes to task-related user exits

Invocation by CICS context management

Task-related user exit programs can now be invoked at an additional invocation point. Currently, a task-related user exit program can be invoked by:

- An application program

- CICS SPI manager
- CICS syncpoint manager
- CICS task manager
- CICS termination manager
- The Execution Diagnostic Facility (EDF)

To this list is added CICS context management.

A CICS application that interacts with another (non-CICS) product that supports the Application Response Measurement (ARM) workload balancing and reporting standard can use a task-related user exit program, invoked by CICS context management, to support cross-product workload monitoring. Sometimes, such a task-related user exit program is supplied by the non-CICS, ARM-enabled, product.

A task-related user exit program signals that it wants to be invoked by CICS context management by setting a bit in the schedule flag word: see The schedule flag word, in the *CICS Customization Guide*. It can set this bit when it is invoked by an application program or by the CICS task manager at start-of-task.

Note that the only way to cause the exit program to be invoked by CICS context management is for the exit program itself, on a preliminary invocation, to set the bit in the schedule flag word. Calls by the CICS termination manger, for instance, can be scheduled by specifying the SHUTDOWN option on the EXEC CICS ENABLE command that enables the exit program. There is no equivalent option on the EXEC CICS ENABLE command to cause the exit program to be invoked by CICS context management. (How to use options of the EXEC CICS ENABLE command to cause a task-related user exit program to be invoked for specific types of call is described in Enabling for specific invocation-types, in the *CICS Customization Guide*.)

If the context management bit in the schedule word is set for the current transaction, CICS context management invokes the exit program whenever the transaction issues a non-terminal-related EXEC CICS START command. (The exit program is not invoked for terminal-related EXEC CICS START commands.)

If the transaction to be started is remote, the correlator is passed to the remote transaction only if the remote region is connected by an MRO link.

On invocation, the exit program is passed a context-related parameter list: see CICS context management parameters, in the *CICS Customization Guide*.

Typically, the job of the exit program is to extract the ARM correlator (which relates to the transaction to be started), if one is included in the work request, and to make it available to CICS.

How the correlator is passed in the work request depends on the type and format of the request. It might be passed in the request header, for instance. The location of the correlator, in the work request, must be understood by the exit program.

Typically, on invocation the exit program first checks whether it has been called for a transaction in which it is “interested”, or for an unrelated transaction. If it’s for the former, what happens next may depend on whether there is a workload manager correlator included in the request.

If there is a correlator included in the request *and* the 512-byte data area addressed by the UECON_CORRELATOR_PTR field of its parameter list does not already contain a correlator, the exit program should:

1. Use the IWMMEXTR macro to extract the correlator from the work request.
2. If the correlator is in character format, convert it to binary format. (Character format might have been used to pass the work request over external protocols, for example.) ARM correlators must be between 4 and 512 bytes long and in binary format. The first two bytes must contain the length of the correlator (including the length field itself). If the correlator is less than 512 bytes long, it must be padded on the right with binary zeros.
3. Make the correlator available to CICS, by placing it in the data area addressed by the UECON_CORRELATOR_PTR field of its parameter list.

Important: If, on entry to the exit program, the data area addressed by the UECON_CORRELATOR_PTR field already contains a correlator, the exit program should not change it. (If the content of the data area starts with `X'0000'` you can assume that the data area does not contain a correlator.)

If there is no correlator included in the work request, the exit program may or may not provide one. If it does not pass a correlator it should leave the data area addressed by the UECON_CORRELATOR_PTR field set to binary zeros (its value on invocation of the exit program).

When the user exit program returns, if there is a correlator in the UECON_CORRELATOR_PTR field CICS checks that it is of the correct length. If the correlator fails this check, CICS ignores it.

At the attach of the transaction started by the EXEC CICS START command, if there is a valid correlator the monitoring domain passes it to the z/OS Workload Manager (WLM). The WLM does one of the following:

- Accepts the correlator as valid. In this case, the WLM returns a new correlator that is known as a **child correlator**.
- Rejects the correlator as invalid or unrecognized. In this case, the WLM treats this as an edge transaction, and generates a new **edge correlator**.

The new correlator is in EWLM format. CICS uses it to identify the piece of work in any further WLM calls.

Chapter 9. Changes to the external CICS interface (EXCI)

The external CICS interface (EXCI) is an application programming interface that enables a non-CICS program (a client program) running in MVS to call a program (a server program) running in a CICS region and to pass and receive data by means of a communications area.

Changes to the EXCI options table

The EXCI options table, generated by the DFHXCOPT macro, enables you to specify a number of parameters that are required by the external CICS interface. A new option, XCFGROUP, is added to the EXCI options table.

XCFGROUP={DFHIR000|name}
Specifies the name of the cross-system coupling facility (XCF) group to be joined by this client program.

Note: XCF groups allow CICS regions in different MVS images within the same sysplex to communicate with each other across multi-region operation (MRO) connections. For introductory information about XCF/MRO, and instructions on how to set up XCF groups, see Cross-system multiregion operation (XCF/MRO) in the *CICS Intercommunication Guide*.

Each client program can join a maximum of one XCF group.

DFHIR000
The default XCF group name.

name
The group name must be eight characters long, padded on the right with blanks if necessary. The valid characters are A-Z 0-9 and the national characters $ # and @. To avoid using the names IBM uses for its XCF groups, do not begin group names with the letters A through C, E through I, or the character string “SYS”. Also, do not use the name “UNDESIG”, which is reserved for use by the system programmer in your installation.

It is recommended that you use a group name beginning with the letters “DFHIR”.

© Copyright IBM Corp. 2001, 2007

Chapter 10. Changes to user-replaceable programs

This section summarizes the changes that affect CICS user-replaceable programs. You must reassemble all user-replaceable programs, whether or not you make any changes to them.

You should check whether the changes listed for this release to the user-replaceable program interface 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.

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

Changes to DFHCNV

When migrating to CICS Transaction Server for z/OS, Version 3 Release 2, you need to reassemble any DFHCNV data conversion tables that you use, because CICS initialization fails if you try to load DFHCNV tables assembled using macros from an earlier release. There is a new DFHCNV macro parameter operand SYSDEF.

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 Defining the conversion table in *CICS Family: Communicating from CICS on zSeries*.

New user-replaceable programs

DFHAPXPO

DFHAPXPO was introduced in CICS Transaction Server for z/OS, Version 3 Release 1. It 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 Language Environment Programming Guide*, SA22-7561, 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 CICSTS32.CICS.SDFHSAMP library.

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

© Copyright IBM Corp. 2001, 2007

DFHISAIP

DFHISAIP was introduced in CICS Transaction Server for z/OS, Version 3 Release 2. Its purpose is to manage the autoinstall of IP interconnectivity (IPIC) connections.

The IPCONN autoinstall user program is similar to the APPC autoinstall user program. Like the APPC autoinstall user program, the IPIC autoinstall user program can choose an installed connection to use as a template for the new connection. The main differences are that the template is an IPCONN rather than a CONNECTION definition, and that the use of the template is optional.

If IPCONN autoinstall is active, CICS installs the new IPCONN connection using:

- The information in the connect flow
- The IPCONN template, optionally selected by the IPCONN autoinstall user program
- Values returned by the user program in its communications area
- CICS-supplied values

DFHISAIP, the Assembler language version, is the default user program for autoinstall of IPIC connections. Sample COBOL, PL/I, and C versions are also supplied. The source for all the versions of the sample program is supplied in the CICSTS32.CICS.SDFHSAMP library.

For more information about this program, see Writing a program to control autoinstall of IPIC connections, in the *CICS Customization Guide*.

Chapter 11. Changes to CICS monitoring

There are changes to CICS monitoring data that could affect user-written and vendor-written utilities that analyze and print CICS SMF 110 monitoring records.

The length of a standard performance class monitoring record, as output to SMF, has increased to 2352 bytes. This does not take into account any user data that you add, or any system-defined data fields that you exclude by using a monitoring control table (MCT). CICS Transaction Server for z/OS, Version 3 Release 2 introduces a data compression facility for SMF 110 monitoring records, which can provide a significant reduction in the volume of data written to SMF.

The offsets have changed for a number of the default CICS dictionary entries in the dictionary data sections of CICS monitoring SMF 110 records.

The length of a monitoring clock for performance class data has increased from 8 bytes to 12 bytes. This affects all performance class data fields defined as ″TYPE-S″, and also affects any user-defined event-monitoring points (EMPs) which involve clocks. User clocks are defined in the monitoring control table (MCT) using DFHMCT TYPE=EMP macros. Note that the monitoring clocks for transaction resource class data are **not** changed, and they remain at 8 bytes.

Check your utility programs that process CICS SMF records to ensure that they can still process 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. This is particularly important if you want to activate data compression for monitoring records. You need to make sure that the product is able to identify compressed CICS SMF 110 monitoring records, and expand the data section using the z/OS Data Compression and Expansion Services, so that the monitoring records can be processed correctly. If the reporting tool is not able to do this, you could use the CICS-supplied monitoring sample program DFH$MOLS with the EXPAND control statement to produce an output data set containing the SMF 110 monitoring records in their expanded format, for the tool to work with.

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

New and changed monitoring data fields

A number of new performance data fields are added to performance class data records. Some existing performance class and exception class data fields are changed.

New performance data fields added in CICS Transaction Server for z/OS, Version 3 Release 2

Group DFHCICS

360 (TYPE-C, 'OAPPLID', 8 BYTES)
The applid of the CICS region in which this work request (transaction) originated; (for example, the region in which the CWXN task ran).

361 (TYPE-T, 'OSTART', 8 BYTES)
The time at which the originating task (for example, the CWXN task) was started.

© Copyright IBM Corp. 2001, 2007

362 (TYPE-P, 'OTRANNUM', 4 BYTES)
The number of the originating task (for example, the CWXN task).

363 (TYPE-C, 'OTRAN', 4 BYTES)
The transaction ID (TRANSID) of the originating task (for example, the CWXN task).

364 (TYPE-C, 'OUSERID', 8 BYTES)
The originating Userid-2 or Userid-1 (for example, from CWBA), depending on the originating task.

365 (TYPE-C, 'OUSERCOR', 64 BYTES)
The originating user correlator.

366 (TYPE-C, 'OTCPSVCE', 8 BYTES)
The name of the originating TCPIPSERVICE.

367 (TYPE-A, 'OPORTNUM', 4 BYTES)
The port number used by the originating TCPIPSERVICE.

368 (TYPE-C, 'OCLIPADR', 16 BYTES)
The IP address of the originating client (or Telnet client).

369 (TYPE-A, 'OCLIPORT', 4 BYTES)
The TCP/IP port number of the originating client (or Telnet client).

370 (TYPE-A, 'OTRANFLG', 8 BYTES)
Originating transaction flags, a string of 64 bits used for signaling transaction definition and status information:

Byte 0
The facility-type of the originating transaction:

Bit 0 None (`X'80'`)

Bit 1 Terminal (`X'40'`)

Bit 2 Surrogate (`X'20'`)

Bit 3 Destination (`X'10'`)

Bit 4 3270 bridge (`X'08'`)

Bit 5 Reserved

Bit 6 Reserved

Bit 7 Reserved

Byte 1
Transaction identification information:

Bit 0 System transaction (x'80')

Bit 1 Mirror transaction (x'40')

Bit 2 DPL mirror transaction (x'20')

Bit 3 ONC/RPC Alias transaction (x'10')

Bit 4 WEB Alias transaction (x'08')

Bit 5 3270 Bridge transaction (x'04')

Bit 6 Reserved (x'02')

Bit 7 CICS BTS Run transaction

Byte 2
Reserved.

Byte 3
Transaction definition information:

Bit 0 Taskdataloc = below (x'80')

Bit 1 Taskdatakey = cics (x'40')

Bit 2 Isolate = no (x'20')

Bit 3 Dynamic = yes (x'10')

Bits 4–7
Reserved

Byte 4
The type of the originating transaction:
X'01' None
X'02' Terminal
X'03' Transient data
X'04' START
X'05' Terminal-related START
X'06' CICS business transaction services (BTS) scheduler
X'07' Transaction manager domain (XM)-run transaction
X'08' 3270 bridge
X'09' Socket domain
X'0A' CICS Web support (CWS)
X'0B' Internet Inter-ORB Protocol (IIOP)
X'0C' Resource Recovery Services (RRS)
X'0D' LU 6.1 session
X'0E' LU 6.2 (APPC) session
X'0F' MRO session
X'10' External Call Interface (ECI) session
X'11' IIOP domain request receiver
X'12' Request stream (RZ) instore transport
X'13' IP interconnectivity session

Byte 5
Reserved.

Byte 6
Reserved.

Byte 7
Recovery manager information:

Bit 0 Indoubt wait = no

Bit 1 Indoubt action = commit

Bit 2 Recovery manager - UOW resolved with indoubt action

Bit 3 Recovery manager - Shunt

Bit 4 Recovery manager - Unshunt

Bit 5 Recovery manager - Indoubt failure

Bit 6 Recovery manager - Resource owner failure

Bit 7 Reserved

371 (TYPE-C, 'OFCTYNME', 4 BYTES)
The facility name of the originating transaction. If the originating transaction is not associated with a facility, this field is null. The transaction facility type, if any, can be identified using byte 0 of the transaction flags, OTRANFLG (370), field.

Group DFHDOCH

223 (TYPE-A, 'DHDELCT', 4 BYTES)
The number of document handler DELETE requests issued by the user task.

Group DFHSOCK

288 (TYPE-A, 'ISALLOCT, 4 BYTES)
The number of allocate session requests issued by the user task for sessions using IPIC

300 (TYPE--S, 'ISIOWTT', 8 BYTES)
The elapsed time for which a user task waited for control at this end of an (IPIC) connection.

305 (TYPE--C, 'ISIPICNM', 8 BYTES)
The name of the IPIC connection whose TCP/IP service attached the user task.

330 (TYPE--A, 'CLIPPORT', 4 BYTES)
The port number of the client or Telnet client.

Performance data fields changed in CICS Transaction Server for z/OS, Version 3 Release 2

Group DFHSOCK

244 (TYPE-C, 'CLIPADDR', 16 BYTES)
The client IP address (in the form *nnn.nnn.nnn.nnn*) or Telnet client IP address.

Group DFHTASK

164 (TYPE-A, 'TRANFLAG', 8 BYTES)
Transaction flags, a string of 64 bits used for signaling transaction definition and status information:

Byte 0
Transaction facility identification

Bit 0 Transaction facility name = none (x'80')

Bit 1 Transaction facility name = terminal (x'40')

If this Bit is set, FCTYNAME and TERM contain the same terminal id.

Bit 2 Transaction facility name = surrogate (x'20')

Bit 3 Transaction facility name = destination (x'10')

Bit 4 Transaction facility name = 3270 bridge (x'08')

Bits 5–7
Reserved

Byte 1
Transaction identification information

Bit 0 System transaction (x'80')

Bit 1 Mirror transaction (x'40')

Bit 2 DPL mirror transaction (x'20')

Bit 3 ONC/RPC Alias transaction (x'10')

Bit 4 WEB Alias transaction (x'08')

Bit 5 3270 Bridge transaction (x'04')

Bit 6 Reserved (x'02')

Bit 7 CICS BTS Run transaction

Byte 2

z/OS workload manager request (transaction) completion information

Bit 0 Report the total response time (begin-to-end phase) for completed work request (transaction)

Bit 1 Notify that the entire execution phase of the work request is complete

Bit 2 Notify that a subset of the execution phase of the work request is complete

Bit 3 This transaction has been reported to the z/OS workload manager as completing abnormally because it has tried to access DB2® and a "connection unavailable" response has been returned. This occurs when all the following are true:

1. Bit 0 is set.
2. CICS is not connected to DB2.
3. The CICS-DB2 adapter is in standby mode (STANDBYMODE(RECONNECT) or STANDBYMODE(CONNECT)).
4. CONNECTERROR(SQLCODE) is specified, causing the application to receive a -923 SQL code.

Bits 4-7
Reserved

Byte 3

Transaction definition information

Bit 0 Taskdataloc = below (x'80')

Bit 1 Taskdatakey = cics (x'40')

Bit 2 Isolate = no (x'20')

Bit 3 Dynamic = yes (x'10')

Bits 4–7
Reserved

Byte 4

Transaction origin type:

X'01' None

X'02' Terminal

X'03' Transient data

X'04' START

X'05' Terminal-related START

X'06' CICS business transaction services (BTS) scheduler

X'07' Transaction manager domain (XM)-run transaction

X'08' 3270 bridge

X'09' Sockets domain

X'0A' CICS Web support (CWS)

X'0B' Internet Inter-ORB Protocol (IIOP)

X'0C' Resource Recovery Services (RRS)

X'0D' LU 6.1 session

X'0E' LU 6.2 (APPC) session

X'0F' MRO session

X'10' External Call Interface (ECI) session

X'11' IIOP domain request receiver

X'12' Request stream (RZ) instore transport

X'13' IPIC session

Byte 5

Transaction status information

Bits 0–5
Reserved

Bit 6 Task purged on an open TCB

Bit 7 Task abnormally terminated

Note: If bit 6 is set, the task has been purged while running on an open TCB, and its transaction timing clocks have been left in an unreliable state. Because of this, the clocks will be set to zero when the record is written by the CICS Monitoring Facility (CMF).

Byte 6

Reserved

Byte 7

Recovery manager information

Bit 0 Indoubt wait = no

Bit 1 Indoubt action = commit

Bit 2 Recovery manager - UOW resolved with indoubt action

Bit 3 Recovery manager - Shunt

Bit 4 Recovery manager - Unshunt

Bit 5 Recovery manager - Indoubt failure

Bit 6 Recovery manager - Resource owner failure

Bit 7 Reserved

Note: Bits 2 through 6 will be reset on a SYNCPOINT request when the MNSYNC=YES option is specified.

275 (TYPE-S, 'JVMRTIME', 12 BYTES)

Before CICS Transaction Server for z/OS, Version 3 Release 2, the JVMRTIME field (group name: DFHTASK, field id: 275) recorded the time spent resetting the JVM environment to its initial state between uses of the JVM. This time was only measurable for resettable JVMs, and usually registered as zero for continuous JVMs. The resettable mode is now withdrawn, but the precision of the CICS monitoring clocks has been increased, so the JVMRTIME field is now able to measure the time spent in JVM cleanup between uses of a continuous JVM. This time includes deleting local references for each task and handling any exception raised. It also includes the time taken to destroy the JVM when CICS ceases to require it.

Before CICS Transaction Server for z/OS, Version 3 Release 2, the JVMRTIME field also recorded the time spent on garbage collections scheduled by CICS. This type of garbage collection was included in the activity measurements for the transaction immediately before the garbage collection took place. Garbage collections scheduled by CICS now take place under a separate transaction, CJGC, and are not recorded in the JVMRTIME field for user transactions.

Exception records changed in CICS Transaction Server for z/OS, Version 3 Release 2

EXCMNTRF (TYPE-C, 8 BYTES)
Transaction flags—a string of 64 bits used for signaling transaction definition and status information. For details, see field 164 (TRANFLAG) in performance data group DFHTASK.

Effects of monitoring clock changes on performance data

The monitoring clocks for performance class data now record dispatch time and CPU time much more precisely, and can do this over a longer period. When you migrate toCICS Transaction Server for z/OS, Version 3 Release 2, you might see differences in the times reported in your performance class data.

The changes to the monitoring clock format should **not** themselves have any measurable impact on the performance of your transactions. However, because of the increased precision and capacity of the clocks, you might see some times for individual transactions being reported differently in your CICS performance class data.

Because the monitoring clocks are more precise, you might see a higher dispatch time or CPU time being reported for any transactions that suffered from under-reporting in previous CICS releases. This is because when the monitoring clocks used units of 16 microseconds, the time recorded was rounded **down** to a multiple of 16 microseconds; that is, only completed 16-microsecond units were recorded. If a transaction was dispatched on a CICS TCB for 24 microseconds, 16 microseconds would be added to the time on the clock, but the other 8 microseconds would go unreported. However, in CICS Transaction Server for z/OS, Version 3 Release 2, with the monitoring clocks recording every microsecond, the 24-microsecond dispatch for the same transaction is reported in full. You are most likely to notice an increase in the amount of dispatch time or CPU time reported where you have a transaction with a high level of TCB switching, such as a nonthreadsafe transaction which makes a number of DB2 requests.

Because the monitoring clocks have a greater capacity, you should see more useful reporting of times for long-running transactions. In previous CICS releases,

transactions that ran for longer than the clock capacity of around 19 hours could not be reported correctly in the performance class data, because the timer component and period count would wrap around after that time. In CICS Transaction Server for z/OS, Version 3 Release 2, the clock components are still not protected against wraparound, but because of the increased clock capacity it is unlikely ever to occur. This means that the time used by long-running transactions can be presented accurately.

Changes to the format of CICS SMF 110 monitoring records

CICS SMF 110 monitoring records are divided into three parts: an SMF header, an SMF product section, and a CICS data section. If data compression is active, the CICS data section is compressed before the record is written to SMF, and must be expanded before use. There is a new field in the SMF product section to identify a compressed monitoring record and give its length after compression.

Effect of data compression

When data compression is active, CICS uses the standard z/OS Data Compression and Expansion Services (CSRCESRV) to compress the CICS data section of each monitoring record before writing it to SMF. The SMF header and SMF product section of records are not compressed.

When CICS SMF 110 monitoring records have been compressed, they need to be identified, and the data section needs to be expanded using the z/OS Data Compression and Expansion Services, before the records can be processed by SMF 110 reporting tools.

Data compression only applies to SMF 110 records written by CICS monitoring, with subtype X'0001' in the record subtype field in the SMF header. It does not apply to the other types of SMF 110 records created by CICS, that is, records written by CICS journaling, CICS statistics, the TS data sharing server, the coupling facility data table (CFDT) server, and the named counter sequence number server.

New product header field SMFMNCRL

The new field SMFMNCRL in the SMF product section of monitoring records identifies where data compression has been used for a monitoring record, and gives the compressed length of the CICS data section.

```
SMFMNCRL DS    XL2           COMPRESSED RECORD LENGTH
```

A zero value in this field indicates that the CICS data section in the record does not contain compressed data. A nonzero value in this field indicates that the CICS data section in the record does contain compressed data, and that the z/OS Data Compression and Expansion Services must be used to expand the data section before processing.

The value of the field shows the length of the CICS data section after compression. The maximum expanded length of the data section will be 32598 bytes.

Changes to the monitoring sample program DFH$MOLS

DFH$MOLS is a sample program which you can modify or adapt to your own purposes. It is intended to show you how you can code your own monitoring utility program to produce reports from the data collected by the CICS monitoring domain (MN), and written to SMF data sets. From CICS Transaction Server for z/OS,

Version 3 Release 2, DFH$MOLS is able to identify any SMF 110 monitoring records that have been compressed, and expand them using the z/OS Data Compression and Expansion Services (CSRCESRV), before printing reports.

New monitoring clock field format

DFH$MOLS now reports clock fields in the format `ddd hh:mm:ss.000000`, showing a count for days, hours, minutes and seconds, followed by 6 decimal places (down to one microsecond).

New EXPAND control statement

Use this option if some or all of the input monitoring records were compressed, and you want to create an output data set with these records in their expanded format, along with the records that were never compressed.

EXPAND
specifies that the monitoring data is to be written to an output data set, including any compressed SMF 110 monitoring records in their expanded format, along with the records that were never compressed. The output data set of SMF 110 monitoring records can be used by reporting tools which are not able to use the z/OS Data Compression and Expansion Services (CSRCESRV) to expand compressed records.

A monitoring record with a compressed data section is identified by the compressed record length in the SMFMNCRL field in the SMF product section, which is only present for a compressed record.

If you just want to print reports, or to unload the records into a fixed length format, you do not need to specify the EXPAND option. DFH$MOLS identifies and expands any compressed monitoring records automatically before working with them. You only need to specify the EXPAND option if you want to create an output data set of SMF 110 monitoring records.

DDNAME=name
specifies the ddname for the output data set to hold the SMF 110 monitoring records. If you do not code this keyword, the default ddname SYSUT2 is used, and your job stream must include a SYSUT2 DD statement. If you code this keyword to specify a different ddname, your job stream must include the corresponding DD statement.

NEWDCB
To ignore the DCB information from the original data set, specify NEWDCB. Supply the new DCB information on the JCL for the output data set.

Note:

1. When the EXPAND control statement is specified, the only parameter for IGNORE and SELECT statements that operates during creation of the output data set is the APPLID option. The PRCSTYPE, TASKNO, TERMID, TRANID, and USERID parameters are ignored while the output data set is being produced. You can also select records for the output data set by date, using the DATE parameter, or by time, using the TIME parameter.
2. Monitoring data is not automatically printed when the EXPAND control statement is specified. If this statement is specified, and you also want to print monitoring data, you need to specify the PRINT control statement explicitly. When you specify the PRINT statement

to print monitoring records, all of the selection parameters on your IGNORE and SELECT statements now operate for the selection of the monitoring records for printing.

New messages

The following new messages are produced by DFH$MOLS if problems are encountered in expanding compressed monitoring data records.

118: UNABLE TO EXPAND A COMPRESSED RECORD, RC='nn'; REPORT IS TERMINATED

The DFH$MOLS program was unable to expand the compressed data section for an SMF 110 monitoring record. This abend is issued when the z/OS Data Compression and Expansion service CSRCESRV FUNCTION=EXPAND was unable to expand the data section in the SMF record. For more information on the return codes issued by the CSRCESRV service, see the *z/OS MVS Assembler Services Reference ABE-HSP*.

This message is followed by an MVS abend U118 with a dump.

119: UNABLE TO OPEN DDNAME '*xxxxxxxx*'; REPORT IS TERMINATED

The DFH$MOLS program was unable to open the data set specified on the DD statement used for the EXPAND control statement. '*xxxxxxxx*' is either SYSUT2, which is the default, or the ddname specified by the DDNAME= parameter on the EXPAND control statement. Ensure that the JCL for the job was correct.

This message is followed by an MVS abend U119 without a dump.

120: UNEXPECTED CSRCESRV QUERY ERROR, RC='nn'; REPORT IS TERMINATED

The DFH$MOLS program received an unexpected (non-zero) return code from the z/OS Data Compression and Expansion service CSRCESRV FUNCTION=QUERY. For more information on the return codes issued by the CSRCESRV service, see the *z/OS MVS Assembler Services Reference ABE-HSP*.

This message is followed by an MVS abend U118 with a dump.

DFH$MOLS support for data for earlier CICS releases

The CICS Transaction Server for z/OS, Version 3 Release 2 release of DFH$MOLS no longer processes monitoring data for CICS/ESA® Version 4 or CICS/ESA Version 3, or for the oldest CICS Transaction Server releases. The UNLOAD control statement has additional restrictions.

In CICS Transaction Server for z/OS, Version 3 Release 2, DFH$MOLS can process SMF 110 monitoring data records for the following releases:

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

However, the UNLOAD control statement (which unloads performance class monitoring data into a fixed length record format) can only be used with monitoring data for CICS Transaction Server for z/OS, Version 3 Release 2, and not with

monitoring data for any earlier CICS releases. Any version or release of DFH$MOLS cannot process monitoring data for a version or release *later* than itself, so you should always use the DFH$MOLS from the highest version or release available to you.

Chapter 12. Changes to CICS statistics

There are changes to CICS statistics records. These are because of new domains, or as a result of enhancements to CICS. There are some new statistics DSECTs, and a number of statistics DSECTs have new or changed fields. Application programs using the changed DSECTs might need to be recompiled.

New statistics DSECTs

Copybook
For functional area

DFHDHDDS
Document template statistics

DFHISRDS
IPCONN resource statistics

DFHLDBDS
LIBRARY resource statistics

DFHMQGDS
WebSphere MQ Connection statistics

DFHPIPDS
Pipeline resource statistics

DFHPIWDS
Webservice resource statistics

DFHWBRDS
URIMAP resource statistics

DFHWBSDS
URIMAP global statistics

Changed statistics DSECTs

Copybook
For functional area

DFHA17DS
File resource statistics

DFHDSRDS
MVS TCB resource statistics

DFHDSTDS
MVS TCB global statistics

DFHLDGDS
Loader statistics

DFHLDRDS
Loader statistics for programs

DFHMNGDS
Monitoring domain statistics

DFHMNTDS
Transaction performance monitoring resource statistics. (Data is not written to SMF by DFHMNTDS. It is only relevant when used through the COLLECT STATISTICS interface.)

© Copyright IBM Corp. 2001, 2007

DFHSJGDS
The JVM pool

DFHSJRDS
JVM profiles

DFHSMSDS
Storage above 16MB

DFHSORDS
TCP/IP service resource statistics

Existing application programs are unaffected by the changes if they use the old versions of the following changed DSECTS:
DFHDSTDS
DFHDSRDS
DFHMNGDS
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 the other changed DSECTs listed are such that the old DSECTs are not compatible with the new DSECT, and application programs using these DSECTs must be recompiled.

New values in DFHSTIDS (statistics record identifiers)

The new DSECTs added for CICS Transaction Server for z/OS, Version 3 Release 2 have corresponding values in the common statistics record copybook, DFHSTIDS. The revised list of the statistics record identifiers is shown in CICS statistics data section in the *CICS Customization Guide*.

The values in that list which are new for CICS Transaction Server for z/OS, Version 3 Release 2 are:

```
STILDB   31    DFHLDBDS   LIBRARY (resource) id
STIMQG   74    DFHMQGDS   MQ connection stats (global) id
STIISR   109   DFHISRDS   IPCONN (resource) id
STIDHD   112   DFHDHDDS   DOCTEMPLATE (resource) id
```

The statistics formatting utility program, DFHSTUP

The statistics formatting utility program is enhanced to format additional statistics reports for the new statistics. New resource types can be coded on the SELECT TYPE and IGNORE TYPE parameters using these keywords:
- DOCTEMPLATE
- IPCONN
- LIBRARY
- MQCONN
- PIPELINE
- URIMAP
- WEBSERVICE

Chapter 13. Changes to problem determination for CICS-MQ, CICS-DBCTL and CICS-DB2

CICS-MQ components are now shipped with CICS. As a result, tracing has changed for CICS-MQ, CICS-DBCTL and CICS-DB2 as follows:

- All trace entries produced by the CICS-MQ components will use the CICS trace domain. If you have user tracing enabled for the sole purpose of WebSphere MQ tracing, user tracing can be turned off, saving the overhead of application trace.
- The CICS-DBCTL Attach and CICS-DB2 Attach will change to use RA (Resource Manager Adapter) Level 1 and Level 2 tracing, instead of FC (File Control) Level 1 and Level 2 tracing.

CICS-MQ messages are changed from CSCQxxx to DFHMQ0xxx. Ensure your message retrieval applications are aware of this change.

© Copyright IBM Corp. 2001, 2007

Part 2. Migration tasks for CICS Transaction Server

© Copyright IBM Corp. 2001, 2007

Chapter 14. Migration procedures for CICS Transaction Server

Complete these tasks when you migrate any CICS Transaction Server region to CICS Transaction Server for z/OS, Version 3 Release 2.

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 Setting up the catalog data sets in the *CICS System Definition Guide*. When you initialize the catalogs, make sure that you use the CICS Transaction Server for z/OS, Version 3 Release 2 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).

Enabling z/OS conversion services

To get the benefits of z/OS conversion services for data conversion, perhaps because your system requires support for the conversion of UTF-8 or UTF-16 data to EBCDIC, you need to enable the z/OS conversion services and install a conversion image which specifies the conversions that you want CICS to perform.

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

© Copyright IBM Corp. 2001, 2007

Upgrading the CSD for CICS-supplied and other IBM-supplied resource definitions

Upgrade the CICS-supplied resource definitions using the UPGRADE function of the CSD utility program (DFHCSDUP). (If you are migrating from CICS Transaction Server for OS/390, Version 1 Release 3, you also need to change the CSD average and maximum record size.) If you have resource definitions in your CSD that support other IBM products (such as z/OS), you might need to upgrade these also.

If you need to share your upgraded CSD with different CICS releases, see “CSD compatibility between different CICS releases” on page 89.

1. If you are migrating from CICS Transaction Server for OS/390, Version 1 Release 3, redefine the CSD to VSAM with an increased average and maximum record size, RECORDSIZE(200 2000). Use any of these methods to do this:
   a. Take a backup, then delete the data set, define a new one with the correct record size, and REPRO the backup into the new data set.
   b. Rename the old data set as a backup, then create a new data set and REPRO the renamed data set into the new one. “Sample jobs for additional CSD modifications” implements this method.
   c. Define a new data set with the correct record size and other attributes, and then REPRO the old data set into the new one.
2. Run the DFHCSDUP utility program, specifying the UPGRADE command, to upgrade the CICS-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 DFHCSDUP UPGRADE command in the *CICS Operations and Utilities Guide*.
3. If you have resource definitions in your CSD that support other IBM products, upgrade these as required. For example, if your Language Environment resource definitions are not at the correct z/OS level, you should delete and replace the CSD group containing these. The Language Environment resource definitions are in the SCEESAMP library in member CEECCSD. “Sample jobs for additional CSD modifications” has an example job to delete and replace the CSD group containing these.

Sample jobs for additional CSD modifications

If you need to upgrade the Language Environment resource definitions in your CSD, you can use a job like this.

```
//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=CICSTS32.CICS.SDFHLOAD,DISP=SHR
//DFHCSD   DD DSN=CICSTS32.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=CICSTS32.CICS.SDFHLOAD,DISP=SHR
//DFHCSD   DD DSN=CICSTS32.CICSHURS.DFHCSD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD DSN=SYS1.ZOS170.SCEESAMP(CEECCSD),DISP=SHR
/*
//
```

Figure 2. Upgrading Language Environment resource definitions

If you are migrating from CICS Transaction Server for OS/390, Version 1 Release 3, you can use a job like this to increase the record size for the CSD.

```
//BAKUPCSD JOB (1,BELL),CLASS=A
//ALTERDEF EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=A
//AMSDUMP  DD SYSOUT=A
//SYSIN    DD *
  ALTER CICSTS32.CICSH.DFHCSD.* -
      NEWNAME(CICSTS32.CICSH.DFHCSD.*.BACKUP)
  ALTER CICSTS32.CICSH.DFHCSD -
      NEWNAME(CICSTS32.CICSH.DFHCSD.BACKUP)
  IF LASTCC = 0 THEN -
      DEFINE CLUSTER (                        -
                NAME( CICSTS32.CICSH.DFHCSD ) -
                REC(10000)                    -
                VOLUME(SYSDA)                 -
                KEYS( 22 0 )                  -
                INDEXED                       -
                RECORDSIZE( 200 2000 )        -
                FREESPACE( 5 5 )              -
                SHAREOPTIONS( 2 )             -
                )                             -
      INDEX     (                             -
                NAME( CICSTS32.CICSH.DFHCSD.INDEX ) -
                )                           -
      DATA      (                           -
                NAME( CICSTS32.CICSH.DFHCSD.DATA )    -
                )
/*
//REPROCSD EXEC PGM=IDCAMS,REGION=0M,COND=(5,LT,ALTERDEF)
//SYSPRINT DD SYSOUT=A
//AMSDUMP  DD SYSOUT=A
//SYSIN    DD *
      REPRO INDATASET(CICSTS32.CICSH.DFHCSD.BACKUP) -
            OUTDATASET(CICSTS32.CICSH.DFHCSD)
/*
//
```

Figure 3. Sample job to rename and redefine the CSD

Migrating user-modified CICS-supplied resource definitions

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

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

The SCAN function searches for the CICS-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 CICS-supplied definition and a user-modified version. If you have copied and changed the name of an CICS-supplied definition, the SCAN command enables you to specify the changed name as an alias.

See The DFHCSDUP SCAN command in 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 might 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 results in message DFH5151. You can ignore this message.

Example

JVMPROFILE(DFHJVMCD) is added to the definition of program DFHADJR. Therefore, DFH$CSDU contains the following command:

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

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

CSD compatibility between different CICS releases

Most releases of CICS make changes to the CICS-supplied groups of resource definitions that are included in the DFHLIST group list. 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.

When you have upgraded a CSD, if 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 20 on page 90 shows you which DFHCOMP*x* 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 the table. For example, to run a CICS TS 1.3 region, with the CSD upgraded to CICS TS 3.2, append the compatibility group DFHCOMP9 followed by DFHCOMP8 at the end of your group list.

Table 20. Required compatibility groups for earlier releases of CICS

	CICS TS 3.2 CSD	**CICS TS 3.1 CSD**	**CICS TS 2.3 CSD**	**CICS TS 2.2 CSD**	**CICS TS 1.3 CSD**
Shared with CICS TS 3.1	DFHCOMPC	None	Do not share	Do not share	Do not share
Shared with CICS TS 2.3	DFHCOMPB	DFHCOMPB	None	Do not share	Do not share
Shared with CICS TS 2.2	DFHCOMPA	DFHCOMPA	DFHCOMPA	None	Do not share
Shared with CICS TS 1.3	DFHCOMP9 DFHCOMP8	DFHCOMP9 DFHCOMP8	DFHCOMP9 DFHCOMP8	DFHCOMP9 DFHCOMP8	None

Chapter 15. Migration for application programs

CICS translator support for pre-Language Environment compilers is withdrawn. Runtime support is usually provided for your existing application programs that were developed using these old compilers, with the exception of OS/VS COBOL and OO COBOL programs, which do not have runtime support.

Withdrawal of support for pre-Language Environment compilers

The compilers for which translator support is withdrawn 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)

For details of the compilers that are supported by CICS, see High-level language support in the *CICS Transaction Server for z/OS Release Guide*.

The following JCL procedures that were 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	—
C++	DFHYITEL	DFHYXTEL	—
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 for programs developed using pre-Language Environment compilers

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

© Copyright IBM Corp. 2001, 2007

apply maintenance to these application programs, you should use one of the supported Language Environment-conforming compilers.

Applications compiled and linked with pre-Language Environment compilers usually execute successfully using the runtime support provided by Language Environment. They do not usually have to be recompiled or re-link-edited. In some circumstances, you might need to adjust Language Environment runtime options to enable these applications to execute correctly. Refer to the *z/OS Language Environment Run-Time Application Migration Guide*, and the *Compiler and Run-Time Migration Guide* for the language in use, for further information. Because pre-Language Environment compilers are not Language Environment-conforming, programs compiled by these compilers cannot take advantage of all Language Environment facilities in a CICS region.

The runtime libraries provided by Language Environment replace the native runtime libraries that were provided with older compilers such as VS COBOL II, OS PL/I and C/370. The native runtime libraries provided with pre-Language Environment compilers are not supported. Language libraries, other than the Language Environment libraries, should not be present in your CICS startup JCL. If, perhaps for commonality with other CICS systems, the JCL for your CICS startup job includes other language libraries, the Language Environment libraries must be above all the other language libraries in the JCL concatenations of the CICS startup job for both STEPLIB and DFHRPL. This ensures that the programs are processed by Language Environment.

Withdrawal of runtime support for OS/VS COBOL

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.

Withdrawal of runtime support for OO COBOL

In this CICS release, 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 this CICS release. 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 16. Migration for Business Transaction Services (BTS)

When migrating your BTS environment to CICS Transaction Server for z/OS, Version 3 Release 2, you might need to migrate the DFHLRQ data set. 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.

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 for z/OS, Version 3.2 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, or if your existing CICS system is a later release than those listed here, the DFHLRQ records match the format of DFHLRQ records used in CICS TS for z/OS, Version 3.2, and so can be migrated using a utility such as IDCAMS COPY. However, if your existing CICS system is listed here and you do not have one of these PTFs applied, the DFHLRQ record format is **not** compatible with that used in CICS TS for z/OS, Version 3.2. In this case, you should complete your BTS workload prior to migration to CICS TS for z/OS, Version 3.2, and start with an empty DFHLRQ at that release.

© Copyright IBM Corp. 2001, 2007

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.

Chapter 17. Migration for BTAM and TCAM networks

Transactions cannot be routed to a CICS Transaction Server for z/OS, Version 3 Release 2 application-owning region (AOR) from a network of BTAM terminals, or from a network of terminals connected by the ACB interface of TCAM. You are recommended to migrate your connections to use ACF/VTAM.

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 older CICS releases) to route transactions from them to a CICS Transaction Server for z/OS, Version 3 Release 2application-owning region (AOR). You must either upgrade your terminals or route to an AOR atCICS Transaction Server for z/OS, Version 2 Release 3 or an earlier release.

TCAM networks

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 older CICS releases) to route transactions from them to a CICS Transaction Server for z/OS, Version 3 Release 2 AOR. You must migrate your connections to use TCAM/DCB or (preferably) ACF/VTAM, or route to an AOR at CICS Transaction Server for z/OS, Version 2 Release 3 or an earlier release. (All terminals that support TCAM/ACB also support ACF/VTAM.)

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 Transaction Server for z/OS, Version 3 Release 2. 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 Transaction Server for z/OS, Version 3 Release 2. However, you are recommended to migrate your connections to use ACF/VTAM.

Using dummy consoles instead of dummy TCAM terminals

The EXEC CICS SIGNON command does not work in transactions that have no principal facility: that is, non-terminal transactions. One unofficial technique to solve this problem was suggested in early Marketing Question and Answer forums. This technique involved creating dummy TCAM terminals in the terminal control table (TCT) defined by the DFHTCT macros, and specifying these as the principal facility for tasks that needed to use the SIGNON command. Now that local TCAM support is removed from CICS, it is no longer possible to use TCAM terminals for this purpose.

If you have been using this technique in the past, it may be possible for you to use dummy consoles instead of dummy TCAM terminals. You can create a dummy console using the EXEC CICS CREATE command:

```
EXEC CICS CREATE TERMINAL(tttt) ATTRLEN(47) ATTRIBUTES(attribs)
```

where *attribs* is a character data area that contains the string

```
'CONSNAME(nnnnnnnn) TYPETERM(DFHCONS) NATLANG(E)'
```

© Copyright IBM Corp. 2001, 2007

tttt and *nnnnnnnn* are a unique terminal ID and console name that you have chosen for this purpose. The terminal ID must be unique in the CICS region. The console name does not need to correspond to a real console, assuming that the transaction does not attempt any terminal I/O. DFHCONS is the CICS-supplied TYPETERM definition for console devices, provided in the CICS-supplied CSD group DFHTYPE.

This dummy console with a TERMINAL resource definition could now be used where you previously used a dummy TCAM terminal. You need to set the terminal ID as available for automatic transaction initiation (ATI) before START commands can be issued against it. You can set a terminal to ATI using the command EXEC CICS SET TERMINAL(*tttt*) ATI. The CICS-supplied TYPETERM definition DFHCONS does not specify the ATI attribute, so the default setting ATI(NO) applies until you issue the SET TERMINAL command. When the terminal ID is available for ATI, you can schedule transactions against it. These transactions have the dummy console as their principal facility, and so can use the EXEC CICS SIGNOFF and EXEC CICS SIGNON commands. Each transaction requires one dummy console.

It must be emphasized that the use of dummy TCAM terminals or dummy consoles to enable a non-terminal transaction to issue the EXEC CICS SIGNON command is **not recommended**, and has never previously been suggested in the CICS product manuals. These techniques are described here only for migration purposes, and you should redesign affected applications as soon as possible to eliminate the use of dummy consoles.

When programs exploited the technique of using dummy TCAM terminals, their intention was usually to use the EXEC CICS SIGNON command to change the user ID of the running transaction. However, note that the behavior of the EXEC CICS SIGNON command was changed in CICS Transaction Server for z/OS, Version 2 Release 1, so that it does not normally affect the user ID currently in effect for the transaction issuing the command. The EXEC CICS SIGNON command can only change the user ID of the running transaction if you have enabled the XSNEX global user exit, and if that exit sets a return code of 4. Note that the XSNEX global user exit was provided only for migration purposes when the behavior of the EXEC CICS SIGNON command was changed, and the use of XSNEX to change the user ID of a running transaction is not recommended.

The correct way to associate a user ID with a non-terminal transaction is to start the transaction with the specified user ID using the following command:

```
EXEC CICS START TRANSID(tttt) USERID(uuuuuuuu)
```

If you need to authenticate the user ID before doing this, use the following command to validate the user's password:

```
EXEC CICS VERIFY PASSWORD(pppppppp) USERID(uuuuuuuu)
```

Neither of these commands require the transaction to have an associated principal facility. If you have any applications which use the migration aids described in this topic (dummy TCAM terminals, dummy consoles, or the XSNEX user exit), you should redesign them as soon as possible to use this correct method.

Chapter 18. Migrating from COMMAREAs to channels

CICS application programs that use traditional communications areas (COMMAREAS) to exchange data continue to work as before. If you want to take advantage of the new function, you can migrate existing applications to use channels and containers rather than 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. This section explains how to do this for several types of existing application.

User-written dynamic or distributed routing programs require migration work whether or not you plan to implement channels and containers in your own applications. 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.

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 this table.

Table 21. 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)`

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 this table.

© Copyright IBM Corp. 2001, 2007

Table 22. 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 this table.

Table 23. 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 this table.

Table 24. 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)
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. You can migrate these commands to use a channel in place of a COMMAREA.

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 Introduction to CICS dynamic routing in the *CICS Intercommunication Guide*. For information about writing a dynamic or distributed routing program, see Writing a dynamic routing program in the *CICS Customization Guide*.

Coexistence of programs that use channels with back-level CICS releases

A CICS Transaction Server for z/OS, Version 3 Release 2 program can invoke a program on a remote CICS region and pass it a channel. For this to work successfully, the remote region must be at CICS Transaction Server for z/OS, Version 3 Release 1 or a later level. Although CICS regions before CICS Transaction Server for z/OS, Version 3 Release 1 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 returns a meaningful abend code.

If a CICS TS Version 3 application tries to send a channel to a back-level region to which the appropriate APAR has been applied, the transaction abends with a meaningful abend code. If a CICS TS Version 3 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 CICS releases prior to CICS Transaction Server for z/OS, Version 3 Release 1 which can tolerate channels are shown in this list, 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 19. Migrating to extended addressing for ESDS

To use an extended ESDS data set, you need to migrate the data set, and convert existing CICS application programs that use 32-bit relative byte addressing (RBA) to 64-bit extended relative byte addressing (XRBA).

Converting a standard ESDS to an extended addressing ESDS

Note: Before migrating a standard ESDS data set to use extended addressing, if your dataset is defined to use forward recovery it is essential to upgrade your forward recovery product to one that can read the new log records written for extend addressing ESDS data sets. If you use CICS VR, the release required is CICS VSAM Recovery for z/OS V4.2.

To convert an existing standard ESDS to an extended addressing ESDS, you must recreate the data set as follows:

1. If you want to continue to use the contents of the existing data set, take a copy of its contents. You can use the `AMS REPRO` function to do this.
2. Delete the existing data set.
3. Create a new data set. You can base the AMS definition of the new data set on that of the old data set. The only mandatory change is that the DATACLAS parameter of the definition of the new data set must name an SMS data class that specifies both extended format and extended addressing.

   How to define SMS data classes is described in the *DFSMS Storage Administration Reference* manual.
4. If necessary, restore the contents of the data set from the copy taken previously.

Converting a program from 32-bit RBA to 64-bit XRBA

To convert an existing program from using 32-bit RBA to 64-bit extended relative byte addressing (XRBA) you must:

1. Replace the RBA keyword with the XRBA keyword, on all the following commands:
   - EXEC CICS READ
   - EXEC CICS READNEXT
   - EXEC CICS READPREV
   - EXEC CICS RESETBR
   - EXEC CICS STARTBR
   - EXEC CICS WRITE
2. Replace all 4-byte areas used for keys with 8-byte areas. This step is very important.

   If you change ″RBA″ to ″XRBA″ but don’t change the length of the key areas:

   a. On STARTBR and READ commands, CICS will treat your 4-byte RBAs as being the top half of 8-byte XRBAs. In most cases this produces a huge XRBA number. You should be able to track down this error quite quickly because the program immediately receives a ″no record at RBA″ response.

   b. WRITE commands may produce more subtle, and therefore probably more serious, errors. The command will feed back an 8-byte XRBA, which will overwrite the 4 bytes immediately following the key area.

© Copyright IBM Corp. 2001, 2007

Using RBA-insensitive programs to access extended ESDS data sets

It is possible to reuse existing 32-bit RBA programs, that do not actually make use of the RBAs, to access 64-bit extended ESDS data sets.

For example, there is a common type of application in which records are first written sequentially and later browsed sequentially from the beginning. Although RBAs are passed between CICS and the program, the program actually makes no use of them. The program is interested only in reading or writing the next record. We refer to such programs as being “RBA-insensitive”. Other programs, such as those that directly read or update records at named RBAs, we refer to as “RBA-sensitive”.

Existing 32-bit RBA-insensitive programs can access 64-bit extended ESDS data sets without change. Both RLS and non-RLS modes are supported.

Thirty-two-bit RBA-sensitive programs cannot access 64-bit extended ESDS data sets, even if the data set contains less than 4 gigabytes of data.

Connecting a back-level AOR to a CICS TS for z/OS, Version 3.2 FOR

In this scenario, old-style 32-bit RBA programs try to access files on a CICS TS for z/OS, Version 3.2 file-owning region (FOR). This works in either of the following cases:

- The target file in the FOR has not been converted from conventional ESDS to extended addressing ESDS.
- The target file has been converted to extended addressing ESDS but the program is RBA-insensitive.

If the target file has been converted to extended addressing ESDS, a 32-bit RBA-sensitive program running in the AOR cannot access it. The program receives an ILLOGIC response.

Connecting a CICS TS for z/OS, Version 3.2 AOR to a back-level FOR

In this scenario, new-style 64-bit XRBA programs try to access files on a back-level file-owning region.

Because the target region supports only 32-bit RSAs, it does not understand a 64-bit XRBA. The program receives an ILLOGIC response.

Chapter 20. Migration for intercommunication

In CICS TS for z/OS, Version 3.2 there are the following intercommunication-related migration tasks:

- Ensuring that each of your CICS regions has a unique APPLID: see “Giving each CICS region a unique applid.” This is a mandatory requirement.
- Migrating APPC or MRO connections to IP interconnectivity (IPIC): see “Migrating connections to IP interconnectivity.”
- Separating your CICS regions into multiple XCF groups: see “Using multiple CICS XCF groups” on page 114.

Giving each CICS region a unique applid

If your CICS regions are part of a z/OS sysplex, you must ensure that their *applid*s (as specified on the APPLID system initialization parameter) are unique within the sysplex. If CICS is an XRF partner, its *specific* applid must be unique within the sysplex. In CICS TS for z/OS, Version 3.2, if, on CICS startup, a specified applid is found to duplicate the (specific or only) applid of any other CICS region currently active in the sysplex, CICS issues a message and fails to initialize.

Migrating connections to IP interconnectivity

Note: If you do not want to use IPIC connections, no migration work is required. Existing MRO, APPC, and LUTYPE6.1 connections will continue to operate as before.

To migrate APPC or MRO connections to IPIC you must:

1. Install support for IPIC. How to do this is described in “IP interconnectivity (IPIC)”, in the *CICS Transaction Server for z/OS Installation Guide*.
2. Migrate your existing connections to TCP/IP. CICS supplies a migration utility to help you do this: see “The migration utility for converting APPC and MRO connections to IPIC connections” on page 109.

For reference purposes, and in case you want to migrate your APPC and MRO connections manually, rather than by running the migration utility, the following four tables show the attributes of APPC CONNECTION, APPC SESSIONS, MRO CONNECTION, and MRO SESSIONS definitions, respectively, and the equivalent attributes on IPCONN definitions.

APPC

Table 25. Migrating APPC connections to IPIC. CONNECTION options and their IPCONN equivalents.

CONNECTION options	APPC possible values	IPCONN equivalent value
ACCESSMETHOD	VTAM	Not applicable.
ATTACHSEC	LOCAL | IDENTIFY | VERIFY | PERSISTENT | MIXIDPE	USERAUTH LOCAL | IDENTIFY | VERIFY | NO | CERTIFICATE
AUTOCONNECT	NO | YES | ALL	NO | YES
BINDSECURITY	NO | YES	SSL NO | YES
DATASTREAM	USER	Not applicable
INDSYS	Not applicable (indirect connections only).	Not applicable (indirect connections only).

© Copyright IBM Corp. 2001, 2007

Table 25. Migrating APPC connections to IPIC. CONNECTION options and their IPCONN equivalents. (continued)

CONNECTION options	APPC possible values	IPCONN equivalent value
INSERVICE	YES | NO	As is.
MAXQTIME	NO | 0 - 9999	As is.
NETNAME	The VTAM APPLID of the remote region. (For XRF, the generic applid. For connections to a VTAM generic resource, either the applid or generic resource name.)	combination of APPLID and NETWORKID
PROTOCOL	APPC.	Not applicable.
PSRECOVERY	SYSDEFAULT | NONE	Not applicable.
QUEUELIMIT	NO | 0 - 9999	As is.
RECORDFORMAT	U	Not applicable
REMOTENAME	Name (sysid) by which the remote system is known to itself.	Not applicable
REMOTESYSNET	Applid of the remote system that owns the remote resource, if the link to the remote system is indirect.	Not applicable
REMOTESYSTEM	Name (sysid) of the remote system, or sysid of the next system in the path, if the link to the remote system is indirect.	Not applicable
SECURITYNAME	RACF ID of the remote system.	As is.
SINGLESESS	NO | YES	Not applicable.
USEDFLTUSER	NO | YES	Not applicable
XLNACTION	KEEP | FORCE	As is.

Table 26. Migrating APPC connections to IPIC. SESSIONS options and their IPCONN equivalents.

SESSIONS options	APPC possible values	IPCONN equivalent value
AUTOCONNECT	NO | YES | ALL	Not applicable.
BUILDCHAIN	YES	Not applicable.
CONNECTION	Name of CONNECTION that this SESSIONS definition applies to.	Not applicable.
DISCREQ	Not applicable.	Not applicable.
IOAREALEN	Not applicable.	Not applicable
MAXIMUM	1 - 999, 0 - 999	SENDCOUNT & RECEIVECOUNT
MODENAME	Name of a VTAM LOGMODE.	Not applicable.
NEPCLASS	Transaction class for the node error program.	Not applicable.
NETNAMEQ	Not applicable.	Not applicable
PROTOCOL	APPC.	Not applicable.
RECEIVECOUNT	Not applicable.	Derived from MAXIMUM.
RECEIVEPFX	Not applicable.	Not applicable
RECEIVESIZE	RU size to receive: 1 - 30720	Not applicable

Table 26. Migrating APPC connections to IPIC. SESSIONS options and their IPCONN equivalents. (continued)

SESSIONS options	APPC possible values	IPCONN equivalent value
RECOVOPTION	SYSDEFAULT | CLEARCONV | RELEASESESS | UNCONDREL | NONE	Not applicable.
RELREQ	NO | YES	Not applicable.
SENDCOUNT	Not applicable.	Derived from MAXIMUM.
SENDPFX	Not applicable.	Not applicable
SENDSIZE	RU size to send: 1 - 30720	Not applicable
SESSNAME	Not applicable.	Not applicable
SESSPRIORITY	0 - 255	Not applicable
USERAREALEN	Length of TCTTE user area: 0 - 255	Not applicable
USERID	ID for sign on.	Not applicable

MRO

We don't envisage that you will want to migrate many of your existing MRO connections to use IPIC. By definition, these are all CICS-to-CICS connections between regions in the same sysplex. For this type of connection, MRO currently holds more advantages than IPIC. For example, there is no performance improvement to be gained by using IPIC; and MRO supports all the base CICS intercommunication functions, whereas IPIC supports only DPL.

Table 27. Migrating MRO connections to IPIC. CONNECTION options and their IPCONN equivalents.

CONNECTION options	MRO possible values	IPCONN equivalent value
ACCESSMETHOD	IRC | XM	Not applicable.
ATTACHSEC	LOCAL | IDENTIFY	USERAUTH LOCAL | IDENTIFY | VERIFY | NO | CERTIFICATE
AUTOCONNECT	Not applicable.	NO | YES
BINDSECURITY	Not applicable.	SSL NO | YES
DATASTREAM	USER	Not applicable
INDSYS	Not applicable (indirect connections only).	Not applicable (indirect connections only).
INSERVICE	YES | NO	As is.
MAXQTIME	NO | 0 - 9999	As is.
NETNAME	The APPLID specified in remote region's SIT.	`host.domain.country[:port]`
PROTOCOL	Blank.	Not applicable.
PSRECOVERY	Not applicable.	Not applicable.
QUEUELIMIT	NO | 0 - 9999	As is.
RECORDFORMAT	U	Not applicable
REMOTENAME	Not applicable.	Not applicable
REMOTESYSNET	Not applicable.	Not applicable
REMOTESYSTEM	Not applicable.	Not applicable
SECURITYNAME	Not applicable.	As is.
SINGLESESS	Not applicable.	Not applicable.
USEDFLTUSER	NO | YES	Not applicable

Table 27. Migrating MRO connections to IPIC. CONNECTION options and their IPCONN equivalents. (continued)

CONNECTION options	MRO possible values	IPCONN equivalent value
XLNACTION	KEEP | FORCE	As is.

Table 28. Migrating MRO connections to IPIC. SESSIONS options and their IPCONN equivalents.

SESSIONS options	MRO possible values	IPCONN equivalent value
AUTOCONNECT	Not applicable.	Not applicable.
BUILDCHAIN	Not applicable	Not applicable.
CONNECTION	Name of CONNECTION that this SESSIONS definition applies to.	Not applicable.
DISCREQ	Not applicable.	Not applicable.
IOAREALEN	Default TIOA size: 0 - 32767 , 0 - 32767	Not applicable
MAXIMUM	Not applicable.	Not applicable
MODENAME	Not applicable.	Not applicable.
NEPCLASS	Transaction class for the node error program.	Not applicable.
NETNAMEQ	Not applicable.	Not applicable
PROTOCOL	LU61.	Not applicable.
RECEIVECOUNT	Number of receive sessions: 1 - 999	As is.
RECEIVEPFX	Termid prefix.	Not applicable
RECEIVESIZE	Not applicable.	Not applicable
RECOVOPTION	Not applicable.	Not applicable.
RELREQ	Not applicable.	Not applicable.
SENDCOUNT	Number of send sessions: 1 - 999	As is.
SENDPFX	Termid prefix.	Not applicable
SENDSIZE	Not applicable.	Not applicable
SESSNAME	Not applicable.	Not applicable
SESSPRIORITY	0 - 255	Not applicable
USERAREALEN	Length of TCTTE user area: 0 - 255	Not applicable
USERID	ID for sign on.	Not applicable

Using the CICS Transaction Gateway ECI Resource Adapter over an IPIC connection

If you use one of the CICS Transaction Gateway ECI Resource Adapters to enable a Java client to call a CICS server program, you may want to rewrite and redeploy your client program in order to take advantage of some of the new features introduced in CICS TS for z/OS, Version 3.2. These new features include support for:

- Containers. For information about using containers instead of COMMAREAs with your CICS server programs, see the *CICS Application Programming Guide*.
- The Enterprise Workload Manager (EWLM). For information about using the EWLM, see ../../com.ibm.cics.ts.java.doc/topics/../../com.ibm.cics.ts.performance.doc/topics/dfht3_ewlm_intro.dita#dfht3_ewlm_intro the *CICS Performance Guide*.

- IPIC connections (also known as *IPCONNs*) to CICS. For information about the attributes of IPIC connections, see the *CICS Resource Definition Guide*.

  You cannot install static IPCONN connections to non-CICS Java clients: these connections are always autoinstalled. See the *CICS Customization Guide*.
- Secure Sockets Layer (SSL) authentication. SSL is supported on IPIC connections between the CICS TG and CICS, and as on client-to-CICS TG connections. For information about using SSL authentication, see the *CICS RACF Security Guide*.

For guidance about how to program the CICS resource adapters, see the *CICS Transaction Gateway Programming Guide*. For definitive reference information about the resource adapter classes, see the *CICS Transaction Gateway Programming Reference*.

The migration utility for converting APPC and MRO connections to IPIC connections

This topic describes how to use the DFH0IPCC utility program to convert existing APPC and MRO connections to IPIC connections (IPCONNs). DFH0IPCC is a sample program for use with the DFHCSDUP system definition utility program.

Note: This utility generates a set of statements, which form a skeleton that you should review and update as required before using it as input to DFHCSDUP.

For information about creating IPCONN and TCPIPSERVICE , see CREATE IPCONN and CREATE TCPIPSERVICEin the *CICS System Programming Reference*.

IPCONN attribute mapping

The following table summarizes how the DFH0IPCC utility program maps the CONNECTION attributes to the new IPCONN definition.

Table 29.

IPCONN attribute	Migrated From / Created By	Comments
APPLID	CONNECTION (NETNAME)	Direct migration.
AUTOCONNECT	CONNECTION (AUTOCONNECT)	Direct migration, but if ALL, set the new value to YES.
CERTIFICATE	N/A	Blank.
CIPHERS	N/A	Blank.
DESCRIPTION	N/A Optional.	Blank. You can modify this in the DFH0IPCC output.
GROUP	CONNECTION (GROUP) SESSIONS (GROUP)	Not changed.
HOST	APPLID Table	Must be specified in APPLID table.
INSERVICE	CONNECTION (INSERVICE)	Direct migration.
IPCONN	CONNECTION (CONNECTION)	Direct migration.
MAXQTIME	CONNECTION (MAXQTIME)	Direct migration.

Table 29. (continued)

IPCONN attribute	Migrated From / Created By	Comments
NETWORKID	APPLID Table	No equivalent, leave blank if not specified in APPLID table or in default.
PORT	APPLID Table	Must be specified in APPLID table.
QUEUELIMIT	CONNECTION (QUEUELIMIT)	Direct migration.
RECEIVECOUNT	Sum of SESSIONS (MAXIMUM)	Direct migration from MRO SESSIONS equivalent setting, or derived from APPC SESSIONS MAXIMUM setting.
SENDCOUNT	Sum of SESSIONS (MAXIMUM)	Direct migration from MRO SESSIONS equivalent setting, or derived from APPC SESSIONS MAXIMUM setting.
SSL	N/A	Left blank. You can modify this in the DFH0IPCC output.
TCPIPSERVICE	APPLID Table	Always “DFHIPIC” or as in APPLID Table.
XLNACTION	CONNECTION (XLNACTION)	Direct migration.

Input to the DFH0IPCC utility program

The DFH0IPCC utility program takes input supplied in an editable table, known as an *APPLID table*. This table is used to store the APPLIDs of all the regions in the relevant setup, along with the corresponding HOST name of the region and the listen PORT of the TCPIPSERVICE used to deal with inbound TCP/IP connections. This table is supplied as a blank template named DFHIPAPP.DATA, This table is FB80 format with 192 records per block. The following table shows the format of the table.

Table 30. Format of DFHIPAPP.DATA

DFHIPAPP.DATA column	Length	Description
APPLID	char 8	Unique identifier, or .DEFAULT Use **.DEFAULT** to specify default values for NETID or TCPIPSERVICE. The leading dot (.) prevents the word DEFAULT being used as a valid APPLID. Only one .DEFAULT row is allowed in the table.
Separator	char 1	Any alphanumeric character
NETID	char 8	Network identifier. When left blank, the default NETID specified by the .DEFAULT row is used.
Separator	char 1	Any alphanumeric character
PORT	char 5	Listening port number
Separator	char 1	Any alphanumeric character
HOST	char 55	TCP/IP host name. If blank, the default TCPIPSERVICE=DFHIPIC, or the name specified in .DEFAULT is used.

Table 30. Format of DFHIPAPP.DATA (continued)

DFHIPAPP.DATA column	Length	Description
Continuation column	char 1	Normally blank. Any non-blank character in this field indicates that the host name is longer than 55 characters and continues in the HOST column in the following row.

The following shows an example of an APPLID table.

```
APPLID. |NETID.  |PORT.|HOST.
--------+--------+-----+------------------------------------------------------+
.DEFAULT|NETW0   |     |TCPIPSERVICE=DFHIPIC
APPLID1 |        |9001 |
APPLID2 |NETW2   |9002 |myhost2.com
APPLID2 |NETW3   |9003 |myhost.service.with.hostname.longer.than.fifty.five.chax
        |        |     |raracters.com
```

You can use any method to fill the table, manually, for example, or by some utility, such as a spreadsheet or script, so long as the fixed-length format is preserved. You can remove, or omit, any of the provided comments or header lines.

TCPIPSERVICE names

Because an IPCONN cannot deduce the TCPIPSERVICE name of a partner region, the utility cannot produce TCPIPSERVICE definitions, rather they must be manually defined. The way the utility works means that all TCPIPSERVICE names in regions for which the utility produces IPCONN definitions must be the same.

All IPCONNs created by the DFH0IPCC utility program have the default TCPIPSERVICE (DFHIPIC), unless you supply a different name via the .DEFAULT row in the APPLID file. If you specify another name, you should use this for all TCPIPSERVICEs you define.

IPCONN names

The IPCONN names are generated to avoid duplicates. The DFH0IPCC utility program uses the name of the CONNECTION definition because there is a one-to-one relationship between a CONNECTION definition and the IPCONN definition created from it. If a CONNECTION and IPCONN with the same name exist in the same group, CICS chooses the IPCONN in preference to the CONNECTION.

Using the DFH0IPCC utility program

In the description that follows we use the phrase *interconnected regions* to mean the set of CICS regions that are to be connected by IPIC connections.

1. Create TCPIPSERVICE definitions for each of the interconected regions, specifying:
   - PROTOCOL(IPIC)
   - TCPIPSERVICE(DFHIPIC) or TCPIPSERVICE(*servicename*)

   Specify other options, such as PORTNUMBER, according to the requirements of the region where the TCPIPSERVICE is to be installed. The number of definitions you require depends on (for example) the number of unique port numbers you need to specify.
2. Put each TCPIPSERVICE in a resource definition group of its own. Add one or more resource groups to each CICS system definition file (CSD) used by the

interconnected regions, the number depending on the number of CICS regions the CSD serves and the number of unique TCPIPSERVICEs they require. Then install exactly one such TCPIPSERVICE, named DFHIPIC, or user-defined service name, in each of the interconnected regions.

3. Complete an APPLID table. (CICS provides an empty, template, table called DFHIPAPP.DATA.) The table must contain the application identifiers (APPLIDs), network IDs, where applicable, TCP/IP port numbers, and host names of all the interconnected CICS regions. If the previously defined TCPIPSERVICE definitions were named anything other than DFHIPIC, the table must contain a .DEFAULT record with TCPIPSERVICE=servicename in the HOST column.
4. Copy this table to every system that contains a CSD used by the interconnected regions.
5. Create a JCL that can be used to invoke DFH0IPCC through DFHCSDUP.

   The following example shows a typical invocation JCL.

```
//IPCJOB   JOB user,CLASS=A,USER=user,PASSWORD=pass
/*ROUTE PRINT user
//CSDUPJOB EXEC PGM=DFHCSDUP,REGION=0M
//STEPLIB  DD DSN=loadlibrary,DISP=SHR
//         DD DSN=loadlibrary,DISP=SHR
//DFHCSD   DD DSN=csdfilename,DISP=SHR
//SYSPRINT DD SYSOUT=A
//CSDCOPY  DD UNIT=VIO
//APPLTABL DD DSN=applidtablename,
//            DISP=SHR,UNIT=SYSDA,SPACE=(CYL,(2,1)),
//            DCB=(RECFM=FB,BLKSIZE=15360,LRECL=80)
//LOGFILE     DD DSN=logfilename,
//            DISP=(MOD,CATLG,CATLG),UNIT=SYSDA,SPACE=(CYL,(2,1)),
//            DCB=(RECFM=FB,BLKSIZE=15360,LRECL=80)
//OUTFILE  DD DSN=outputfilename,
//            DISP=(MOD,CATLG,DELETE),UNIT=SYSDA,SPACE=(CYL,(2,1)),
//            DCB=(RECFM=FB,BLKSIZE=15360,LRECL=80)
//SYSUDUMP DD SYSOUT=A
//SYSABEND DD SYSOUT=A
//SYSIN    DD *
EXTRACT GR(group1) USERPROGRAM(DFH0IPCC) OBJECTS
EXTRACT GR(group2) USERPROGRAM(DFH0IPCC) OBJECTS
EXTRACT GR(list1) USERPROGRAM(DFH0IPCC) OBJECTS
EXTRACT GR(list2) USERPROGRAM(DFH0IPCC) OBJECTS
/*
//
```

6. On one of the CSD-owning systems, use your customized JCL file to invoke the DFH0IPCC utility program.

   The JCL issues a DFHCSDUP EXTRACT command, passing the utility program as the *USERPROGRAM*. The DFH0IPCC program examines the specified lists and resource groups in the CSD, collecting information about the CONNECTION and SESSIONS definitions it finds. For each APPC or MRO pair of CONNECTION and SESSIONS definitions, it creates an IPCONN definition. Where appropriate, the attributes of the IPCONN definition are taken from the CONNECTION and SESSIONS definitions, with the values of the remaining attributes taken from the APPLID table or allowed to take their default values.When it has completed an IPCONN definition, the utility program writes a series of DEFINE statements, which form the SYSIN for your resulting DFHCSDUP invocation JCL.
7. Review the output produced by the utility program and check that the IPCONN definitions are correct for your installation, you might, as an example, need to change the SSL attributes. You will probably need to change user names and load libraries before you run to define the new IPCONN definitions in the CSD file.

8. Repeat steps 6 through 7 for each CSD file used by the interconnected CICS regions.

Using TCP/IP management and control

Migration of existing functions

You can continue to use CICS, including CICSPlex SM, without exploiting this new function. If you do not want to use TCP/IP management and control, no migration work is required.

Migration to the new function

You can make use of the new function with minimal changes to your monitoring and CICSPlex SM WUI configurations.

Migration for multiregion operation (MRO)

For CICS multiregion operation (MRO) support, the interregion communication program DFHIRP is installed in the link pack area (LPA). The CICS TS for z/OS, Version 3.2 DFHIRP module is compatible with earlier releases, and is designed to work with all releases of CICS. However, note that the CICS TS for z/OS, Version 3.2 version of DFHIRP (required for multiple XCF group support) can be used only on z/OS Version 1.7 or later.

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 still be installed from the **highest** release of CICS running in that MVS image. For example, a CICS TS 2.3 DFHIRP can communicate with a CICS TS for z/OS, Version 3.2 across XCF/MRO, but the CICS regions running in the MVS with the CICS TS 2.3 DFHIRP cannot be later than CICS TS 2.3.

The following steps are a guide to the migration process for MRO, to install the latest DFHIRP and DFHCSVC modules 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 Installing CICS modules in the MVS link pack area in 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 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 new SVC on stand-alone CICS regions, without using any MRO. You can do this running the CICS IVP, DFHIVPOL.
3. Install the CICS 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 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. Now 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. 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. With the production regions running successfully under the CICS 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.

Using multiple CICS XCF groups

XCF group limit relief allows multiple XCF groups to contain CICS regions. Although a CICS region can still join only one XCF group, that group need not be DFHIR000. Thus, although each group is still limited to 2047 members, there is no longer an absolute limit on the number of CICS regions that a sysplex can support. The effective limit of 2047 CICS regions that a single sysplex can support has been lifted.

Migration without exploiting the new function

If you are not constrained by the limit of 2047 members of an XCF group, you do not need to take any action. You can continue to use the default DFHIR000 XCF group; you don't need to specify DFHIR000 explicitly on the XCFGROUP parameter of the system initialization table and DFHXCOPT EXCI table.

Migrating to multiple CICS XCF groups

If you are constrained by the limit of 2047 XCF group members, you must determine how to split your CICS regions into related groups. Typically, you will not want to create a large number of XCF groups. A simple method of partitioning many regions is to put the production regions in a different group from the development and test regions.

Even if you are not constrained by the 2047 limit, you can still use the XCF group feature to isolate your development and test regions from your production regions.

If you decide to have multiple XCF groups, it is recommended that:

- You put your production regions in a different XCF group from your development and test regions
- You do not create more XCF groups than you need: two, separated as described, may be sufficient
- You try not to move regions between XCF groups
- You try not to add or remove regions from existing XCF groups

Releases of CICS prior to CICS TS for z/OS, Version 3.2 are able to join only the DFHIR000 group, so you should first migrate to CICS TS for z/OS, Version 3.2 those systems that need to use a different XCF group (for example, the production systems).

For details of how to set up and configure XCF/MRO, see Generating XCF/MRO support.

Chapter 21. Migrating to the Enterprise Workload Manager

This topic describes how to run the Enterprise Workload Manager (EWLM) with the z/OS workload manager (WLM).

Migration of existing functions

For an introduction to the EWLM, see Introduction to the Enterprise Workload Manager.

If you do not want to use the Enterprise Workload Manager no migration work is required.

Migration to the new function

This topic describes how to run the Enterprise Workload Manager (EWLM) with the z/OS Workload Manager (WLM).

1. If the EWLM managed server, domain manager, and Control Center are not already installed, install them now. For information on how to install and configure the EWLM, see the IBM Enterprise Workload Manager for z/OS documentation.
2. Optionally, write a task-related user exit program to be invoked by CICS context management. For guidance, see in the *CICS Customization Guide*.

   Writing a task-related user exit program is necessary only if you are developing a CICS application that interacts with another (non-CICS) EWLM-enabled product and you want to support cross-product workload monitoring. In these circumstances only, you need a task-related user exit program that is invoked by CICS context management. Even in these circumstances, however, such a task-related user exit program is typically supplied by the non-CICS, EWLM-enabled, product..
3. If necessary, install and enable the task-related user exit program. For information on how to do this, see in the *CICS Customization Guide*.

© Copyright IBM Corp. 2001, 2007

Chapter 22. Migration for Java applications

When migrating to a new CICS release, you are likely to require changes to your JVM profiles and to other aspects of your Java environment. You might also require changes to your Java applications and enterprise beans.

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

Note: There are two versions of the IBM Software Developer Kit for z/OS, Java 2 Technology Edition Version 1.4, a 31-bit and a 64-bit version. CICS TS 3.2 supports only the 31-bit version, which must be at the 1.4.2 level.

The JVM provided by the IBM Software Developer Kit for z/OS, Java 2 Technology Edition, Version 1.4.2 was also used by CICS Transaction Server for z/OS, Version 3 Release 1 and CICS Transaction Server for z/OS, Version 2 Release 3.

CICS Transaction Server for z/OS, Version 2 Release 2 used the JVM provided by the IBM Developer Kit for OS/390 Java 2 Technology Edition Version 1.3.1s. Java programs that ran under CICS Transaction Server for z/OS, Version 2 Release 2 can also run under CICS Transaction Server for z/OS, Version 3 Release 2.

Application migration issues for Java applications are discussed at:

```
http://java.sun.com/j2se/1.4.2/compatibility.html
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
```

To avoid potential problems with deprecated APIs, you should develop all new Java programs for CICS Transaction Server for z/OS, Version 3 Release 2 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.

If you already have JVM profiles and JVM properties files which you set up in a previous CICS release, you might want to migrate these for use with the new CICS release. The settings that are suitable for use in JVM profiles can change from one CICS release to another, so you should check the CICS documentation for any significant changes, and compare your existing JVM profiles to the latest CICS-supplied samples. Changes to the JVM profile options in this CICS release are documented in the *CICS Transaction Server for z/OS Release Guide*. A list of suitable options for the present release is in *Java Applications in CICS*. For JVM profiles created more than one or two releases ago, you are recommended to use the new samples supplied with CICS Transaction Server for z/OS, Version 3 Release 2 to help you create new files, rather than migrating your existing files.

Make a copy of your JVM profiles in a new location on z/OS UNIX to use with the new CICS release, and make the changes that are required to migrate them (for example, changing the path for the home directory for CICS files on z/OS UNIX). Do not try to use JVM profiles with more than one CICS release at the same time, because the settings will not be compatible.

© Copyright IBM Corp. 2001, 2007

Ensure that the JVM profiles you want to use are in the z/OS UNIX directory that is specified by the **JVMPROFILEDIR** system initialization parameter. *Java Applications in CICS* explains how to set the location for the JVM profiles.

The JVM profiles DFHJVMPR and DFHJVMCD, and their associated JVM properties files, must always be available to CICS and configured so that they can be used in your CICS region. *Java Applications in CICS* tells you how to do this.

Enterprise beans

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.

Key changes to CICS support for Java applications

If you are migrating from CICS Transaction Server for OS/390, Version 1 Release 3 or CICS Transaction Server for z/OS, Version 2, read this summary of the changes that have a migration impact for your Java environment.

- In CICS Transaction Server for z/OS, Version 3 Release 2, resettable JVMs, which were reset between each use, are no longer supported. Any Java programs that ran in resettable JVMs must be migrated to run in continuous JVMs. Resettable JVMs had the option REUSE=RESET in their JVM profiles (or the older option `Xresettable=YES`). All the CICS-supplied sample JVM profiles for reusable JVMs now specify the option REUSE=YES, rather than REUSE=RESET. This includes the default JVM profile DFHJVMPR, and the JVM profile DFHJVMCD for CICS-supplied system programs.
- The 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 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 CICS and CICSPlex© SM libraries 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.
- 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. *CICS Performance Guide* tells you how to carry out this tuning process.

Migration from resettable to continuous JVMs

Resettable JVMs are no longer supported in CICS Transaction Server for z/OS, Version 3 Release 2. Any Java programs that ran in resettable JVMs must be migrated to run in continuous JVMs. The migration process involves checking for certain actions in the program code, and then changing some options in your JVM profiles.

To migrate Java programs that ran in resettable JVMs, to run in continuous JVMs, follow these steps:

1. Check that your Java programs do not contain any code which might have an unwanted effect on serial isolation when the continuous JVM is reused by a subsequent program. The checks you should carry out are as follows:
   a. Check for any code that changes the state of the JVM (for example, changing the default time zone). Ensure that the program resets the JVM to the original state. If you need to police any application actions in the continuous JVM, the Java security manager can be used to do this.
   b. Check that any DB2 connections, or other task lifetime system resources, opened by the application are closed or released.
   c. Use the CICS JVM Application Isolation Utility to check for the use of any `static` variables in your Java programs. The use of `static` variables might cause Java programs that were designed to execute in a resettable JVM, to exhibit changed behavior when they execute in a continuous JVM. Possible Java application behavior changes in continuous JVMs explains potential issues. Review the findings of the utility and make any code changes that are necessary to preserve the original behavior. Auditing Java applications for the use of static variables tells you how to use the utility.
2. Examine the existing JVM profiles and JVM properties files for your applications. You can either make a new copy of your existing files and make changes to the options specified in them, or transfer the relevant settings from your existing files to new files based on the samples provided with CICS Transaction Server for z/OS, Version 3 Release 2. There are a number of changes to the options that you can specify in JVM profiles and JVM properties files, so you are recommended to use the new samples to help you create new files, rather than migrating your existing files.
3. Compare your existing JVM profiles and JVM properties files with the new CICS-supplied samples, and with the table of changed options shown in Changes to options in JVM profiles and JVM properties files. Identify the options and system properties which you customized in your existing files, and note any which are now obsolete or need to be specified differently.
4. Either transfer relevant settings from your existing files to new files based on the new CICS-supplied samples, or make appropriate changes to a new copy of your existing files. The most important changes to make are:
   a. Set the correct CICS and Java home directories to match your CICS Transaction Server for z/OS, Version 3 Release 2 installation. The correct directories are already specified in the CICS-supplied samples.
   b. Change REUSE=RESET to REUSE=YES, or replace `Xresettable` with REUSE=YES.
   c. Add the paths to classes that were specified on class paths in your existing files, to the appropriate class path in the new files. There are a number of

changes to the way class paths are specified in CICS Transaction Server for z/OS, Version 3 Release 2. “Migrating class paths in JVM profiles” on page 127 explains how to handle each of the changed class paths.

d. Migrate your storage settings from the existing files to the new files. The way in which storage is used in a continuous JVM differs in some respects from the way it is used in a resettable JVM. Migrating storage settings in JVM profiles from resettable JVMs explains how to specify suitable storage settings as a starting point for your continuous JVMs.

When you use the JVM profiles, if you have omitted any key changes, CICS issues warning messages to explain what changes are still required.

Possible Java application behavior changes in continuous JVMs

Because there is no reset operation in the continuous JVM, applications which were designed to execute in a resettable JVM might exhibit changed behavior when they execute in a continuous JVM. You might need to make changes to an application in order to preserve its original behavior while running in a continuous JVM.

In a resettable JVM, the state of the JVM was reset after each use, so that no application transaction (that is, code other than trusted middleware code) could affect the operation of subsequent transactions. The JVM reset cleaned up the JVM's storage heaps, reinitialized shareable application classes, and discarded and reloaded nonshareable application classes, meaning that no objects other than trusted static middleware objects could persist in the JVM from one use of the JVM to the next.

The continuous JVM does not reset the JVM's state between uses. This enables the persistence of static objects across tasks, which can be a powerful tool when used deliberately. For example, an application developer can use caching techniques to avoid reinitializing objects on each use. It can also, however, be a source of unexpected and erroneous behavior unless it is handled carefully.

Example 1: Altering static variables

The most common type of state change that an application can make is to alter the value of a `static` variable. `static` variables are shared by all instances of a class, unlike non-static variables which are allocated separately for each instance.

In a resettable JVM, when a class is first loaded, the JVM takes a copy of the initial value of each `static` variable and uses it to restore the variable to its original state at the end of each transaction. Consider the following trivial case:

```
public class HelloWorld
{
    public static int count = 0;

    public static void main(String args[])
    {
        count++;
        System.out.println("Hello World, count is " + count);
    }
}
```

In a resettable JVM, the `static` variable `count` is reset to zero by the JVM after each invocation of the HelloWorld `main()` method. The message therefore shows that `count` is 1 each time HelloWorld is invoked.

In a continuous JVM, however, `count` is not reset to its original value before the next invocation of the `main()` method, and the old, shared, value persists. The message therefore shows the count increasing by 1 on each invocation in subsequent transactions.

To preserve the original behavior while running in a continuous JVM, the HelloWorld class could be changed to make `count` an instance variable and initialise it on each invocation in a constructor:

```
public class HelloWorld
{
    public int count = 0;

    public static void main(String args[])
    {
        HelloWorld hw = new HelloWorld();
        hw.count++;
        System.out.println("Hello World, count is " + hw.count);
    }

    HelloWorld()
    {
        count = 0;
    }
}
```

Example 2: Altering the contents of static objects

A more subtle type of issue can arise when the `static` variable is an object reference whose internal state may change, as in this example:

```
import java.util.Hashtable;
import java.util.Enumeration;

class StaticHash
{
    private static final Hashtable myHashtable = new Hashtable();

    public static void main(String[] args)
    {
        int count = myHashtable.size();
        myHashtable.put("key" + count, "value" + count);

        Enumeration keys = myHashtable.keys();
        while (keys.hasMoreElements())
        {
            Object key = keys.nextElement();
            System.out.println("Found this key in the Hashtable: " + key);
        }
    }
}
```

In a resettable JVM, a new instance of `myHashtable` is created every time the JVM is reset, and it will only ever contain a single key, “key0”. In a continuous JVM, however, only one instance of `myHashtable` is created, and each time the class is run, a new key is added to it.

The issue could be solved in a similar manner to the first example, by making `myHashtable` an instance variable and creating the new `Hashtable` in a constructor. Alternatively, `myHashtable` could be left as a `static` reference and be reset each time by adding a constructor containing an invocation of `myHashtable.clear()`.

Auditing Java applications for the use of static variables

The CICS JVM Application Isolation Utility is provided to help system administrators and application programmers discover static variables in Java applications which they use, or plan to use, in their CICS regions. The application developers should then review the findings of the utility and determine whether the application might exhibit unintended behavior when it runs in a continuous JVM. The utility can be used when migrating Java workloads from resettable to continuous JVMs.

The CICS JVM Application Isolation Utility is shipped with CICS Transaction Server for z/OS, Version 3 Release 2 as a JAR file named `dfhjaiu.jar`. It runs under z/OS UNIX System Services as a standalone utility. You do not need to have a CICS Transaction Server for z/OS, Version 3 Release 2 region or any other CICS region running when you use the utility.

The CICS JVM Application Isolation Utility is a code analyzer tool which inspects Java bytecodes in Java Archive (JAR) files and class files. It does not alter any Java bytecodes. It is provided as a means to help identify potential issues before they arise in a continuous JVM under CICS. The Java application does not need to be running in a CICS region when it is inspected.

To inspect Java applications using the CICS JVM Application Isolation Utility, follow these steps:

1. Confirm that the CICS-supplied file `dfhjaiu.jar`, which is the CICS JVM Application Isolation Utility, is present in the `/utils/isolation` subdirectory of the home directory for CICS files on z/OS UNIX. The default name for the home directory is `/usr/lpp/cicsts/`*`cicsts32`*`/`, where *`cicsts32`* is defined by the USSDIR installation parameter when you installed CICS TS for z/OS, Version 3.2. You can add the `/utils/isolation` directory to the PATH environment variable in z/OS UNIX System Services, so that you do not need to give the full path to the file when you run the utility.
2. Confirm that the shell script `DFHIsoUtil`, which is used to run the CICS JVM Application Isolation Utility, is also present in the `/utils/isolation` subdirectory of the home directory for CICS files on z/OS UNIX. Check that the script file specifies the correct value for the CICS_HOME environment variable, and edit the file to change this if necessary.
3. Identify the class files or JAR files that you want to specify to the utility for inspection. Bear these points in mind:
   a. A Java application can involve classes and JAR files that are specified on several different class paths in the JVM profile or JVM properties file. Make sure you include all of them in your inspections.
   b. You can use wildcard characters in the file names, to inspect all the class files or JAR files in a given directory.
   c. When you specify a JAR file for inspection, the utility inspects all the classes contained in the JAR file.
   d. If you specify an individual class file for inspection, the utility inspects only the named class. If the class includes inner classes, the utility does not automatically inspect these. Specifying JAR files, or using wildcards to inspect a whole directory, ensures that any inner classes are included in the inspection.
4. Log in to a z/OS Unix System Services shell, and enter the command

   `DFHIsoUtil [-verbose] `*`filename`*` [`*`filename`*` ... `*`filename`*`]`

   In this command:

a. `DFHIsoUtil` is the name of the script file which runs the CICS JVM Application Isolation Utility. If you have not set an appropriate PATH environment variable and you are not working in the directory containing the script file, give the full path to the file, for example `/usr/lpp/cicsts/cicsts32/utils/isolation/DFHIsoUtil`.

b. The `-verbose` option makes the utility provide additional information. See "The -verbose option" on page 126.

c. *`filename`* specifies the names of one or more class files or JAR files that you have identified for the utility to inspect. Separate each file name with a space. Give the full path to the files if necessary. You can use wildcard (glob) characters in the file names.

For example, to inspect the class file HelloWorld and obtain the standard report (not the verbose report), enter the command

```
DFHIsoUtil HelloWorld.class
```

5. The report produced by the CICS JVM Application Isolation Utility is written to System.out. You can redirect this to another destination as required.
6. Review the findings of the utility, and then examine the source code for your Java applications. The reports produced by the utility identify some potential issues, but you will need to check whether or not these will actually affect the behavior of the application when it runs in a continuous JVM.

Example 1: Report showing alteration of static variables

When the CICS JVM Application Isolation Utility is used to inspect the HelloWorld class file used in Example 1 in "Possible Java application behavior changes in continuous JVMs" on page 122, the report looks like this:

```
CicsIsoUtil: CICS JVM Application Isolation Utility

Copyright (C) IBM Corp. 2006

Reading Class File: HelloWorld.class

    Method: public static void main(java.lang.String[])
      Static fields written in this method:
        public static int count

    Method: <clinit> (Class Initialization)
      Static fields written in this method:
        public static int count

  Number of methods inspected       : 3
  Total static writes for this class: 2

Number of Jar Files inspected    : 0
Number of Class Files inspected  : 1
```

The report shows that the `static` field `count` is written to during Class Initialization, and in the `main()` method. This indicates that `count` might behave differently when the class is used in a continuous JVM, than in a resettable JVM. The application programmer should examine the source code to decide whether `count` really will behave differently.

Example 2: Report showing alteration of the contents of static objects

When the CICS JVM Application Isolation Utility is used to inspect the StaticHash class file used in Example 2 in "Possible Java application behavior changes in continuous JVMs" on page 122, the report looks like this:

```
CicsIsoUtil: CICS JVM Application Isolation Utility

Copyright (C) IBM Corp. 2006

Reading Class File: StaticHash.class

    Method: <clinit> (Class Initialization)
      Static fields written in this method:
        private static final java.util.Hashtable myHashtable

  Number of methods inspected       : 3
  Total static writes for this class: 1

Number of Jar Files inspected    : 0
Number of Class Files inspected  : 1
```

Note that the `static` variable `myHashtable` is only written to during Class Initialization, yet the internal state of the `Hashtable` changes on each invocation.

This is a more difficult issue to assess. The output of the utility identifies that a static object exists. In this case, the object is a hash table; other items such as arrays might also be in this situation. The application developer should check the source code of the application to ensure that the state of the `static` object is not changed in a way that will unintentionally affect subsequent invocations of the class in a continuous JVM.

It is important to also check the entire graph of other objects that may be referenced from the original `static` object. Any `static` object may contain state of its own. This state may include further objects which are not defined as `static`, but are included within the static context of the parent object. A large graph of objects may be built up in this way, where only the root object is declared as `static`, but state held in any of the objects might be available for use by subsequent applications, because of the `static` root object. The application developer should check for application isolation issues at every level of the object graph, in addition to checking at the root level.

The -verbose option

Normally, the CICS JVM Application Isolation Utility does not print details of methods which do not write to `static` variables, or details of `static final String` variables. With the `–verbose` option specified, the utility does print these extra details, and also lists all static method invocations made.

This additional information can identify other potential issues with your applications. For example, this extract from a report shows code relating to the resettable JVM:

```
Static methods invoked by this method:
        boolean isResettableJVM()
           (defined in class: com.ibm.jvm.ExtendedSystem)
```

All methods in the `com.ibm.jvm.ExtendedSystem` class are related to the resettable JVM. They have all been deprecated, and should be removed from any application code.

Migrating storage settings in JVM profiles from resettable JVMs

You will probably need to adjust and tune the storage-related options in your JVM profiles when you migrate applications to run in continuous JVMs.

When you migrate an application from a resettable JVM to run in a continuous JVM, you should initially deal with each storage option that you have specified in the JVM profile as shown in Table 31.

Table 31. Migrating storage options in JVM profiles

Option (if specified)	Action
`-Xmx`	Use the setting from the resettable JVM profile
`-Xinitth`	Comment out (no longer used)
`-Xms`	Take the setting from the resettable JVM profile and increase it by the value of `-Xinitth` from the resettable JVM profile
`-Xinitacsh`	Comment out (no longer used)
`-Xinitsh`	Take the setting from the resettable JVM profile and increase it by the value of `-Xinitacsh` from the resettable JVM profile

These suggestions assume that the continuous JVM is running the same application or applications as the resettable JVM (that is, you are changing an existing resettable JVM profile to become a continuous JVM profile). If the mix of applications running in the continuous JVM is different, your choice of storage settings will not fit this model.

These suggestions also assume that the storage settings for the resettable JVM were correctly tuned for the needs of your applications. If that is not the case, migrating the storage settings according to this model will not improve that situation. In particular, note that the `-Xinitsh` option and the `-Xinitacsh` option only specify the **initial** storage allocations for the system heap and application-class system heap, and the JVM profile does not specify a maximum size for these heaps. The maximum size of these heaps was only restricted by the storage available in the Language Environment enclave for the JVM. If you tuned the storage for the resettable JVM, the `-Xinitsh` option and the `-Xinitacsh` option should already be set to the amount of storage that is actually used by the application.

Your new settings should be taken as a starting point for the continuous JVM. The way in which storage is used in a continuous JVM differs in some respects from the way it is used in a resettable JVM. In particular, bear in mind that the storage heaps in continuous JVMs are not automatically cleaned up after each program invocation. Because of this, depending on the application design and the extent to which each program cleans up after itself, compared to a resettable standalone JVM running the same workload, the continuous JVM might require either larger storage heap sizes or more frequent garbage collection.

You should re-tune your storage settings using the process described in

Note: Information to be provided in Iteration 3 or 4.

Migrating class paths in JVM profiles

There are a number of changes to the way class paths are specified in CICS Transaction Server for z/OS, Version 3 Release 2. You need to identify an appropriate class path for each of the items that you specified on class paths in your existing JVM profiles and JVM properties files.

Migrating class paths in JVM profiles: library path

In CICS Transaction Server for z/OS, Version 3 Release 2, the base library path is not visible in the JVM profile. You only need to specify any additional dynamic link library (DLL) files that you added to the library path. The option to use for this is LIBPATH_SUFFIX.

The base library path for the JVM is built automatically using the directories specified by the CICS_HOME and JAVA_HOME options in the JVM profile. It includes all the DLL files required to run the JVM, and the native libraries used by CICS. In previous CICS releases, the base library path had to be specified explicitly in the JVM profile, but now that is not required.

The LIBPATH option in the JVM profile is no longer used. For migration purposes, it is still accepted, but CICS issues a warning message when it is found (DFHSJ0538). If you leave any classes specified on this option, they are placed on the library path after the base library path.

You can extend the library path using the LIBPATH_SUFFIX option. When CICS builds the library path, these items are placed on the library path after the base library path directories. When you are creating, changing, or migrating JVM profiles, any items which you added to the library path in previous CICS releases (such as the DLL files required to use the DB2-supplied JDBC drivers) should now be specified using LIBPATH_SUFFIX. The CICS-supplied `/lib` and `/ctg` directories, and the IBM JVM-supplied `/bin` and `/bin/classic` directories, which were specified on the library path in the CICS-supplied sample JVM profiles in earlier CICS releases, do not now need to be specified explicitly in the JVM profile. These directories are now part of the base library path.

The option LIBPATH_PREFIX is available if items need to be placed before the base library path, but this option should only be used under the guidance of IBM support.

Migrating class paths in JVM profiles: middleware classes

In a continuous JVM in CICS Transaction Server for z/OS, Version 3 Release 2, the classes formerly treated as middleware classes must now be placed on the same class path as user application classes. These classes were specified on the trusted middleware class path options TMPREFIX and TMSUFFIX in the JVM profile.

For migration purposes, the trusted middleware class path options TMPREFIX and TMSUFFIX are still accepted, but CICS issues a warning message when they are used.

When you are creating, changing, or migrating JVM profiles, the classes formerly treated as middleware classes should now be placed on one of the following class paths:

- The shareable application class path, which is defined by the `-Dibm.jvm.shareable.application.class.path` system property in the JVM properties file. If you have a shared class cache in your CICS region, choose this option, and place the classes on the shareable application class path in the JVM properties file for the master JVM that initializes the shared class cache.
- The standard class path, which is defined by the CLASSPATH_SUFFIX option in the JVM profile. If you do not have a shared class cache, choose this option, and place the classes on the standard class path in the JVM profile for the JVM where the application will run.

When you have placed the classes on the correct class paths, remove the TMPREFIX and TMSUFFIX options from your JVM profiles.

Migrating class paths in JVM profiles: standard class path

In CICS Transaction Server for z/OS, Version 3 Release 2, the standard class path is constructed in a new way. You need to use the CLASSPATH_SUFFIX option to specify any nonshareable application classes.

CICS builds a base standard class path for the JVM using the `/lib` subdirectories of the directories specified by the CICS_HOME and JAVA_HOME options in the JVM profile. This standard class path contains the JAR files supplied by CICS and by the JVM. It is not visible in the JVM profile.

The CLASSPATH option in the JVM profile is no longer used. For migration purposes, it is still accepted, but CICS issues a warning message when it is found (DFHSJ0523).

Use the CLASSPATH_SUFFIX option to place classes on the standard class path. When you are creating, changing, or migrating JVM profiles, any items which you added to the standard class path in previous CICS releases should now be specified using CLASSPATH_SUFFIX.

If you are migrating JVM profiles from resettable (REUSE=RESET) to continuous (REUSE=YES), and there is no shared class cache in your CICS region, you should place application classes on the standard class path, rather than the shareable application class path. The shareable application class path was the recommended choice for a resettable JVM, because it enabled the classes to be cached in the JVM and reinitialized when the JVM was reset, whereas classes on the standard class path were discarded and reloaded. However, in a continuous JVM, classes on the standard class path are cached within the JVM and kept across reuses. The standard class path is now the recommended choice where there is no shared class cache, because it has greater compatibility with future releases of Java.

Migrating Java garbage collection settings

CICS now initiates garbage collection when heap utilization in the active part of the nonsystem heap reaches a specified limit. In earlier releases, CICS performed garbage collection in a JVM synchronously after a specified number of Java programs had been run. You might need to adjust the new garbage collection option GC_HEAP_THRESHOLD if you want to keep the same frequency of garbage collection as you had in previous releases.

- If you have specified the MAX_RESETS_TO_GC option in your JVM profiles, you should remove it. CICS now ignores this option, and issues a warning message at JVM startup if it is found.
- If you need to tune the frequency of garbage collections initiated by CICS in your JVMs, use the GC_HEAP_THRESHOLD option. This option specifies a percentage limit for heap utilization in the active part of the nonsystem heap, above which CICS initiates garbage collection. The default setting is 85%.
- If you had previously tuned your JVMs so that your applications used almost all of the storage in the active part of the nonsystem heap before garbage collection took place, you might find that the default setting for GC_HEAP_THRESHOLD causes more frequent garbage collections in your JVMs. If this is the case, you should adjust the GC_HEAP_THRESHOLD setting to a higher level to return to the same frequency of garbage collection.

Migration for HPJ-compiled Java programs (Java program objects)

Runtime support for Java program objects and for hot-pooling (HPJ) was 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).

1. Set up the JVM environment, as described in Setting up Java support in *Java Applications in CICS*.
2. Place the class files (with the extension .class) for the Java programs in directories in z/OS UNIX 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 permission to access z/OS UNIX directories and files in *Java Applications in CICS*. If you want to, you can build the class files into packages or JAR files (with the extension .jar) before placing them in the z/OS UNIX directory. 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.
   a. If you saved the class files during this process, you can use these to run in the JVM.
   b. If you did not keep the class files, re-run the Java compiler against your Java source files to produce new class files.
3. 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*.
4. If the Java programs access DB2, follow the instructions in Requirements to support Java programs in the CICS DB2 environment in the *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.
5. Test that the Java programs work correctly in the JVMs that you have defined for them.

Migration for Java applications that use a CICS connector

Runtime support for the CICS Connector for CICS TS, introduced in CICS TS for z/OS, Version 2.1, was withdrawn in CICS TS for z/OS, Version 3.1. You need to migrate any existing applications that use the CICS Connector for CICS TS to use the CCI Connector for CICS TS instead. All new connector applications must use the CCI Connector for CICS TS.

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, which performs a similar role to the CICS Connector for CICS TS. It enables a Java program or enterprise bean running on CICS Transaction Server for z/OS to link to a CICS server program. The old CICS Connector for CICS TS implemented the IBM-proprietary CCF interface, but 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.

For information on using the CCI Connector for CICS TS in new connector applications, and on migrating existing applications that use the CICS Connector for CICS TS to use the CCI Connector for CICS TS instead, see *Java Applications in CICS*.

Chapter 23. Migration for threadsafe programming and the open transaction environment (OTE)

The open transaction environment (OTE) function was introduced in CICS Transaction Server for OS/390, Version 1 Release 3 for Java applications, and extended in CICS Transaction Server for z/OS, Version 2 Release 2 and later releases to other types of application. 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.

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.

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
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. *Java Applications in CICS* has more information about how CICS manages JVMs and their TCBs.

The MAXJVMTCBS system initialization parameter controls the number of J8 and J9 TCBs in the JVM pool.

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 CICS key OPENAPI application programs.
- L9 mode TCBs are used for user key OPENAPI application programs.

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. An open API TRUE is given control under an L8 mode TCB, and can use non-CICS APIs without having to create subtask TCBs. The CICS DB2 task-related user

© Copyright IBM Corp. 2001, 2007

exit operates in OPENAPI mode (it is an open API TRUE), so the CICS DB2 attachment facility uses L8 TCBs for DB2 request processing.

L8 mode TCBs are also used by CICS itself, because CICS uses CICS key OPENAPI programs which run on L8 TCBs when processing Web services requests, parsing XML, and accessing z/OS UNIX files for CICS Web support.

The MAXOPENTCBS system initialization parameter controls the number of L8 and L9 TCBs in the pool of L8- and L9-mode open TCBs.

SP mode TCB and S8 mode TCBs
are used by CICS to manage SSL connections. The S8 TCBs run within a single enclave, which is owned by the SP TCB and also contains the SSL cache.

The MAXSSLTCBS system initialization parameter controls the number of S8 TCBs in the SSL pool.

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. The*CICS Application Programming Guide* has more information about using XPLink.

The MAXXPTCBS system initialization parameter controls the number of X8 and X9 TCBs in the pool of X8- and X9-mode open TCBs.

Existing or new CICS DB2 applications written in any language that access DB2 are automatically involved with the open transaction environment through the CICS DB2 task-related user exit. You have the opportunity to gain performance benefits for these applications. These performance benefits can be gained because open 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, as was the case before the open transaction environment was available, when subtask thread TCBs were used for requests to DB2. This situation also decreases usage of the QR TCB.

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**. 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. The next topic explains in detail what it means for an application to be threadsafe.

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.

For applications that do not make DB2 requests, you can opt to use the open transaction environment by defining user application programs, PLT programs, user replaceable modules or task related user exits as OPENAPI programs. (Global user exits cannot be defined as OPENAPI programs.) Defining a program as an OPENAPI program means that it always runs on an open TCB (L8 or L9), from the start of the program. This allows application workloads to be moved off the QR TCB onto multiple open TCBs, giving the possibility of achieving better throughput, particularly with CPU-intensive programs. It also allows you to use other (non CICS) APIs, but you should note that the use of other (non CICS) APIs within CICS is entirely at the discretion and risk of the user. No testing of other (non CICS) APIs within CICS has been undertaken and the use of such APIs is not supported by IBM Service.

OPENAPI programs must be defined to CICS as threadsafe and have threadsafe application logic. As for CICS DB2 applications, the use of nonthreadsafe EXEC CICS commands, nonthreadsafe task-related user exits, or nonthreadsafe global user exits causes a switch to the QR TCB, and should be avoided. OPENAPI programs have some additional obligations: for example, they must ensure that all non-CICS resources acquired specifically on behalf of the terminating task are freed, and they must not use certain MVS system services.

The use of OPENAPI programs can cause more TCB switching than ordinary threadsafe programs. After any switch to the QR TCB, there is an extra switch, because CICS switches back to the open TCB to continue running the application logic. Additional TCB switching may be involved because of the requirement for the key of the TCB to be correct for OPENAPI programs. OPENAPI TRUEs always run in CICS key on an L8 TCB, so, for example, if a user key OPENAPI program runs on an L9 TCB but makes a DB2 call, CICS switches to an L8 TCB to call DB2, then returns to the L9 TCB to continue running the program. Because of this, CICS DB2 applications should normally be defined as ordinary (CICSAPI) threadsafe programs, rather than as OPENAPI programs. CICS key CICS DB2 applications may be defined as OPENAPI programs if wanted.

How can I make my applications threadsafe?

Applications that involve a task-related user exit (TRUE) enabled using the OPENAPI option, such as applications which access DB2 resources, are already involved with the open transaction environment, and they can gain performance benefits from being threadsafe. For other applications, you can opt to use the open transaction environment by defining them as OPENAPI programs, in which case they must be threadsafe.

Applications can be defined to CICS as either quasi-reentrant or threadsafe. 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.

When applications 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.

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 from an open TCB to the QR TCB occurs in the following circumstances:

- 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. If the program is defined as OPENAPI, CICS then switches back to the open TCB to continue running the application logic. If the program is not defined as OPENAPI, it continues to execute on the QR TCB. For a CICS DB2 application, if the program is not defined as OPENAPI and 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 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 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 situation is the same as after a non-threadsafe EXEC CICS command: an OPENAPI program switches back to the open TCB, and a program not defined as OPENAPI continues to execute on the QR TCB.
- 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 completes, CICS switches back to the QR TCB for task termination. This switch is always necessary.

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

1. **Ensure that the system initialization parameter FORCEQR is not 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.

2. **Ensure that the program is defined to CICS as threadsafe.** Use the CONCURRENCY attribute of the program resource definition to do this. OPENAPI programs are required to be defined as threadsafe. 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.
3. **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.
4. **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 the threadsafe command list in the *CICS Application Programming Reference* and the threadsafe command list in 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 any high-level language constructs or Language Environment callable services used by your program that result in using CICS services. Some CICS services used in this way could involve non-threadsafe CICS commands, and cause a switch back to the QR TCB. For example, the COBOL statement DISPLAY UPON SYSOUT, and the Language Environment callable services CEEMOUT and CEE3DMP, write data to transient data queues. This involves an EXEC CICS WRITE TD command, which is not threadsafe. Check the programming documentation for your high-level language or Language Environment callable services to identify any CICS functions which they use, and ensure that the EXEC CICS commands relating to those functions are listed as threadsafe in the CICS documentation.
5. **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 that 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.)

   The CICS DB2 task-related user exit DFHD2EX1 is threadsafe. SQL, threadsafe and other programming considerations for CICS DB2 applications "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.

6. **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.

   Global user exit programs cannot be defined as OPENAPI, but they can be defined as threadsafe, and treated in the same way as an ordinary application program, by using threadsafe application logic and threadsafe EXEC CICS commands. Writing global user exit programs"Writing global user exit programs" in the *CICS Customization Guide* has general information about writing this type of program.

   A task-related user exit can be defined as threadsafe in the same way as an ordinary application program, or it 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.) 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 was 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 24. Migration for CICS Web support applications

CICS Transaction Server for z/OS, Version 3 Release 2 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 have existing CICS Web support applications from earlier releases, 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.

- **Choose what to do with the new XRES system initialization parameter, which sets resource security for document templates.** For XRES, YES is the default setting, and in this case CICS uses the default class names RCICSRES and WCICSRES. If security checking is active for the CICS region (SEC=YES system initialization parameter), your choices are:
  - Specify XRES=NO explicitly if you want to remove resource security for document templates, and allow any user ID to access them. If you specify XRES=NO you do not need to make any security changes.
  - Specify XRES=YES (the default).

© Copyright IBM Corp. 2001, 2007

- Specify XRES=*name* and define your own resource classes in either the RACF static class descriptor table or the RACF dynamic class descriptor table.
- If you decide to specify XRES=YES or XRES=*name*, follow the instructions in "Migrating to resource security for CICS document templates and z/OS UNIX files" on page 145.

When CICS is being initialized, it requests RACF to bring resource profiles into main storage to match all the resource classes that you specify on system initialization parameters. If CICS requests RACF to load a resource class that does not exist or is not correctly defined, CICS issues a message indicating that external security initialization has failed, and terminates CICS initialization.

- **Choose what to do with the new XHFS system initialization parameter, which specifies access control for z/OS UNIX files.** For XHFS, YES is the default setting, which means that access control for z/OS UNIX
- files is active. If security checking is active for the CICS region (SEC=YES system initialization parameter), your choices are:
  - Specify XHFS=NO explicitly if you want to remove access control for z/OS UNIX files, and allow any user ID to access them. If you specify XHFS=NO you do not need to make any security changes.
  - Specify XHFS=YES (the default). Access permissions for z/OS UNIX files are specified in z/OS UNIX System Services, so you do not need to define RACF profiles for individual files. However, if you are using access control lists (ACLs) to control access to z/OS UNIX files, you need to activate the FSSEC class in RACF. If you decide to specify XHFS=YES, follow the instructions in "Migrating to resource security for CICS document templates and z/OS UNIX files" on page 145.
- **The analyzer program now allows you to supply codepage conversion parameters to CICS Web Support instead of supplying the name of a DFHCNV table entry.** 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 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, 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.
- **CHARACTERSET and HOSTCODEPAGE options for the GET and POST methods are now the same, and in certain circumstances, you will now receive data in your local CCSID.** The CHARACTERSET (previously CLNTCODEPAGE) and HOSTCODEPAGE options will now take effect for forms submitted with the GET method as well as the POST method, and the defaults will be the same in both cases. This means that if the form uses the POST method and you do not specify the HOSTCODEPAGE option, and your

LOCALCCSID initialization parameter is not 037, you will receive your data in your local CCSID, instead of CCSID 037 (the default EBCDIC code page). To specify 037 as the host code page, you must either change your LOCALCCSID parameter to 037, or modify your application to explicitly use ″037″.

- **There are certain considerations for code page conversion to take place when using buffers (with either the INTO or SET option specified).** If you are receiving data into a buffer, and CHARACTERSET and CLICONVERT are not specified, the media type for the message must specify text as the data content type (according to the IANA definitions) for code page conversion to take place. For messages where no media type is given, but CLICONVERT is specified, code page conversion also takes place. If a non-text media type is present, CICS does not convert the message body.
- **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.
- **If you are using the SSL or TLS security protocols, check your SSL-related system initialization parameters and TCPIPSERVICE resource definitions.**
  1. 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 was used as the default in earlier releases, has changed to MEDIUM. For migration purposes, NORMAL is accepted as an alternative to MEDIUM.
  2. 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.
  3. You can use the new CRLSERVER 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.
  4. You can use the new CIPHERS option on TCPIPSERVICE resource definitions to specify a list of cipher suite codes for use with SSL or TLS. The PRIVACY attribute of the TCPIPSERVICE resource definition changes to reflect the CIPHERS attribute value.

Suggestions for migrating to the new enhancements

CICS Web support 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 block of storage, 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 use the WEB API commands. You can now use the WEB API commands in converter programs and in the user-replaceable Web error program DFHWBEP, which enables you to migrate with minimal disruption to your existing CICS Web support architecture. If you use DFHWBEP, you must specify ACTION(IMMEDIATE) in your command, as the default of ACTION(EVENTUAL) is not permitted.
- URIMAP resource definitions can be used to deliver the contents of a CICS document or z/OS UNIX 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.

Migrating to resource security for CICS document templates and z/OS UNIX files

If you decide to specify the system initialization parameters XRES=YES, XRES=*name*, or XHFS=YES, follow these instructions. XRES=YES and XHFS=YES are the defaults.

- If you decide to specify XRES=YES or XRES=*name* to activate resource security for CICS document templates, check the RESSEC attribute in the TRANSACTION resource definitions of any transactions in your CICS region which access document templates. This could include:
  - CEMT, and any other transactions that include EXEC CICS CREATE, DISCARD or INQUIRE DOCTEMPLATE commands.
  - CWXN, if you are using URIMAP definitions to provide static responses from CICS Web support.
  - Alias transactions for CICS Web support application programs. (CWBA is the default alias transaction.)
  - Transactions for other application programs that use EXEC CICS CREATE or INSERT DOCUMENT commands with the TEMPLATE option.

  If RESSEC=YES is specified for any of these transactions, you need to give the user IDs for the transaction permission to use the appropriate document templates. ALTER permission is required to create or discard document templates, and READ permission is required for all other uses (including the API commands DOCUMENT CREATE and DOCUMENT INSERT).

  Note: You cannot change the RESSEC attribute for CICS-supplied transactions in the CICS-supplied RDO groups. If you want to change this attribute, you need to copy the definitions to your own group, where you can change the attribute. CEMT is in group DFHOPER, and CWBA and CWXN are in group DFHWEB.
- In the resource classes that are specified by your XRES system initialization parameter, supply RACF profiles for all the CICS document templates used by transactions with RESSEC=YES in your CICS region.
  1. Make sure you use the correct profile name for the CICS document templates. You need to use the name of the DOCTEMPLATE resource definition (and not the 48-character TEMPLATENAME attribute which is used by EXEC CICS commands), prefixed by the resource type DOCTEMPLATE.
  2. Make sure you give permission to the correct user IDs. For CICS Web support, the user ID associated with the transaction can vary depending on your CICS Web support architecture. User IDs for access to document templates and z/OS UNIX files used by CICS Web support has more information.
- If you decide to specify XHFS=YES to activate access control for z/OS UNIX files, follow the instructions in Implementing security for z/OS UNIX files to allow Web clients to access these files.

  You do not need to check the RESSEC attribute in the TRANSACTION resource definition of the transactions that access the files. If XHFS=YES is specified as a system initialization parameter for the CICS region, all z/OS UNIX files used by CICS Web support as static responses are normally subject to security checking, regardless of the RESSEC attribute for the transaction that is accessing them.

  As an exception, where z/OS UNIX files are defined as CICS document templates and used in that way (for example, by applications), resource security for CICS document templates (specified by the XRES system initialization

parameter) controls access to them for users. In this situation, you do not need to set up resource security in z/OS UNIX System Services for the files. (However, the CICS region user ID always needs to have **read** permissions to z/OS UNIX files, even if they are defined as document templates.)

Chapter 25. Migration for CICS Web services

If you have used CICS Web services in earlier releases, note these migration points.

Migration for the Web services assistants

The Web services assistant batch jobs DFHWS2LS and DFHLS2WS both require a certain amount of memory to create Web service binding files.

The amount of memory required has increased to enable the Web services assistants to process large and complex Web service descriptions. As a result, the region size must now be at least 200 MB. You can either increase the size accordingly or set the region size to `0M`.

If you redeploy your existing Web services in a CICS TS 3.2 region, the regenerated Web service binding files are slightly larger in size.

Migration for MTOM/XOP support

MTOM/XOP support is provided as an optional set of elements in the pipeline configuration file.

If you want to enable your pipeline to take advantage of the MTOM/XOP support, you should be aware that:

- If you use your own application handler rather than the default that is provided by CICS Web services support, the pipeline processes MTOM messages in compatibility mode. You must specify DFHPITP as the application handler in your pipeline configuration file if you want the pipeline to process MTOM messages in direct mode.
- If you use the default CICS Web services application handler, the pipeline processes MTOM messages in direct mode. You should ensure that your message handlers can still run successfully when processing containers that hold XOP documents and binary attachments.
- You should only configure the attribute `send_mtom="yes"` in a provider pipeline configuration file when you are sure that your all of your Web service requesters are capable of receiving MTOM messages. The default value is `send_mtom="same"`, where MTOM messages are only sent when an MTOM message is received.

If you enable MTOM/XOP support in your pipeline, you can retrieve the options that you have specified by using the INQUIRE PIPELINE command.

© Copyright IBM Corp. 2001, 2007

Chapter 26. Migrating to the XPLINK option for C and C++ programs

To continue running your C and C++ programs without exploiting the XPLINK option of the compiler, no action is needed. To exploit the XPLINK option with C and C++ programs, you might have to make changes to your existing applications or configuration.

1. Ensure that your C or C++ program is reentrant, and threadsafe, or modify it so that it conforms to these standards. Chapter 23, “Migration for threadsafe programming and the open transaction environment (OTE),” on page 133 has more information about this.
2. 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*:
   - 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 “Customizing user exits” in the *z/OS Language Environment Customization* manual.
3. Recompile the program using the XPLINK compiler option.
4. Update the concurrency attribute of the PROGRAM resource definition for this program, setting the value to threadsafe.

© Copyright IBM Corp. 2001, 2007

Part 3. Changes to CICSPlex SM externals

© Copyright IBM Corp. 2001, 2007

Chapter 27. Changes to CICSPlex SM installation and definition

This section summarizes the changes to CICSPlex SM installation, initialization parameters, resource definition, and setup.

CICSPlex SM installation integrated with CICS installation

You can now edit the DFHISTAR job to modify both the CICS and CICSPlex SM installation parameters for your environment. EYUISTAR is no longer available as a job to modify CICSPlex SM installation parameters.

DFHISTAR produces customized JCL for CICS and CICSPlex SM. It now includes a number of parameters that are significant only for CICSPlex SM, a number of parameters that are significant only for CICS, and a number of parameters that are common to CICS and CICSPlex SM.

For CICSPlex SM, DFHISTAR generates sample JCL procedures to:

- Create CMAS data sets
- Start a CMAS
- Create Web User Interface (WUI) data sets
- Start a WUI
- Create MAS data sets
- Run a MAS
- Move MAS modules to the link pack area (LPA)

These procedures enable you to create a simple CICSPlex SM configuration that consists of a CMAS, a WUI and a managed CICS system (MAS). The CICSPlex SM Starter Set, which contained samples of JCL for this purpose, is no longer provided.

Removal of the CAS

Because of the removal of the CICSPlex SM TSO end user interface (EUI), you no longer need to set up and use a CAS (coordinating address space) to support a CICS Transaction Server for z/OS, Version 3 Release 2 CMAS (CICSPlex SM address space).

Any attempt to run EYUCAS JCL to start a CAS results in an abend. The removal of the CAS means that there are no CAS-related data sets to install and no CAS to CAS links to configure. This allows the installation of CICSPlex SM to be simplified and streamlined.

Any attempt to run CMAS startup JCL from previous releases will fail because of the references to obsolete components. All data sets beginning with the characters BB are now obsolete, and the CAS initialization program BBM9ZA00 is no longer included in the EYUAUTH library.

The CICSPlex SM system parameter CASNAME identified the CAS subsystem with which a CMAS was associated. This parameter was specified by means of the extrapartition transient data queue COPR assigned to the extrapartition transient data queue EYUPARM. With the removal of the CAS, this parameter is no longer

© Copyright IBM Corp. 2001, 2007

valid. Any attempt to specify CASNAME now results in the invalid parameter message EYUXL0206E. The CASNAME parameter is still valid for CICSPlex SM configurations prior to CICS Transaction Server for z/OS, Version 3 Release 2.

Datasets *.SEYUADEF, *.SEYUVDEF, and *.SEYUJCL, which were supplied in previous releases to support the EUI, are not shipped as part of CICS Transaction Server for z/OS, Version 3 Release 2.

All EUI and CAS-related messages and abend codes have been removed. This includes messages that begin with the prefix BB, unnumbered ISPF messages, and all Uxxxx abend codes. CAS IPCS dialogs and IPCS CICS VERBEXIT keyword are now obsolete.

The XLEC transaction, which was used to connect a CMAS to a CAS, is obsolete.

Message EYUXL0008I has been removed. The message `EYUXL0008I applid CICSPlex registration complete` was the final message issued for a successful CMAS startup.

The final message for a successful CMAS startup is now: `EYUXL0010I applid CMAS registration complete`.

Dynamic creation of CICS resource definitions for CICSPlex SM

The additional CICS resource definitions specifically required to run CICSPlex SM CMAS, WUI and MAS are now created dynamically during initialization and when a CICSPlex SM system is started by a transaction. You no longer need to manipulate the CICS CSD to obtain the default resource definitions. The CICSPlex SM Starter Set, which contained samples of CICSPlex SM definitions, is no longer provided.

This removes complexity from the CICSPlex SM installation process. There is no longer the need to run CSD UPGRADE jobs for your CMASes, WUIs, and MASes, and then use the lists and groups produced by the upgrade in the startup of these systems. The CSD UPGRADE process can be particularly complex where a CSD is shared across CICS releases.

Note: You must run CSD UPGRADE jobs for CICS. For details about upgrading the CICS resource definitions, see “Upgrading the CSD for CICS-supplied and other IBM-supplied resource definitions” on page 86. For information about sharing CSDs across CICS releases, see “CSD compatibility between different CICS releases” on page 89.

You continue to have the facility to alter certain CICSPlex SM definition properties:

- EYUPARMs COIRTASKPRI, COHTTASKPRI, MASALTLRTPRI, and TASKPRIORITY are available to set priorities for certain CICSPlex SM transactions.
- You can use the CICS system initialization parameters LPA and PRVMOD to control whether to search the LPA for CICSPlex SM modules.

If you want to change any other properties, you can include modified definitions on the CSD.

CICS autoinstalls the initial CICSPlex SM programs for a CMAS, MAS, and WUI.

New method for WUI and CICSplex definition

The EYU9XDUT CICSplex definition utility can provide the CICSPlex SM definitions to start a WUI and CICSplex as part of data repository initialization. You would previously have had to use the end user interface or a batch utility to create these definitions.

The utility optionally creates the following CICSPlex SM definitions:

- CPLEXDEF, CICSplex definition
- CPLXCMAS, CMAS in CICSplex
- PLEXCMAS, plex descriptor for the maintenance point CMAS
- CMASCPLX, CMAS in CICSplex
- CSYSDEF, CICS system definition for the WUI

The CMAS SYSID is the basis for the WUI plex name and the WUI name but you can override these using the WUIPLEX and WUINAME parameters in DFHISTAR. The WUI parameter in DFHISTAR specifies whether a WUI is to be created (the default is to create a WUI).

New CICSPlex SM WUI server initialization parameters

CICSPlex SM Web User Interface server initialization parameters can be specified in the start-up job or in a fixed block 80 data set.

New initialization parameters in CICS Transaction Server for z/OS, Version 3 Release 2

AUTOIMPORTDSN(*dsn_name***)**
Specifies the name of the data set containing IBM-supplied view and menu definitions. Currently the supplied set of WUI view and menu definitions are in the SEYUVIEW data set. If you specify a TDQ name, you must specify the name of a data set member using the AUTOIMPORTMEM parameter.

AUTOIMPORTMEM(*member_name***)**
Specifies the name of the data set member or members containing the specific IBM-supplied view and menu definitions you want to import. You can use an asterisk at the end of the input string to specify a group of data set members; for example EYUEA* would import all English language menus and view sets.

DEFAULTMAPBAS(name | EYUSTARTMAPBAS)
Specifies the name of the map object used to generate maps of business application services definitions.

DEFAULTMAPCOLL(value | 0)
Specifies the number of rows in a generated map below which a map opens in the expanded state. If the number of rows to be displayed is above this number, the map opens in a fully collapsed state. The default value of 0 means that in every generated map all of the rows are visible when opened.

DEFAULTMAPMON(name | EYUSTARTMAPMON)
Specifies the name of the map object used to generate maps of monitoring definitions.

DEFAULTMAPRTA(name | EYUSTARTMAPRTA)
Specifies the name of the map object used to generate maps of real-time-analysis definitions.

DEFAULTMAPWLM(name | EYUSTARTMAPWLM)
Specifies the name of the map object used to generate maps of workload management definitions.

New EYU9XDBT utility for CMAS and CICSplex definition

You can use the new EYU9XDBT utility to perform all CMAS and CICSplex definition activities once the basic CMAS environment has been established. You specify the required CICSplex names in some simple parameter, and the utility sets up the definitions for you.

You can use the utility for:

- Defining and removing CICSplexes to and from a CMAS.
- Defining and removing CICS regions to and from a CICSplex.
- Defining and removing CICS groups to and from a CICSplex.
- Adding and removing CICS regions to and from CICS groups.
- Importing, printing or exporting CICSPlex SM objects defined to CMAS or CICSplex contexts.

Change to Common Work Area size for a CMAS

The size of the Common Work Area has increased to 2048 bytes. The Common Work Area size is specified by the CICS system initialization parameter WRKAREA.

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

Changes with RASGNDEF processing in CICSPlex SM

The change described in this topic was implemented through APARs for previous releases of CICSPlex SM. If the version of CICSPlex SM from which you are migrating does **not** have the PTF for the APAR applied, then changes could occur in the way PROGDEFs and TRANDEFs are installed through RASGNDEFs (resource assignment definitions).

The relevant APARs for previous releases of CICSPlex SM are:

- CICSPlex SM Release 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 a PROGDEF or TRANDEF is automatically installed through a RASGNDEF that specifies a USAGE of REMOTE and a MODE of STAT, then 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 versions of CICSPlex SM that 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 3.2.

Table 32 illustrates the differences in processing from 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 32.

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

Note:

1. Note the difference from 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.

Change to generic alert structures used by CICSPlex SM

If you are migrating from CICS Transaction Server for OS/390, Version 1 Release 3 or CICS Transaction Server for z/OS, Version 2, there is 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 28. Changes to CICSPlex SM views and resource tables

This section summarizes the changes affecting CICSPlex SM views, resource tables, and Business Application Services definition objects.

Removal of the CICSPlex SM TSO end user interface (EUI)

With the new enhancements to the CICSPlex SM Web User Interface (WUI) functionality and provision of the EYU9XDBT batch facility, the CICSPlex SM WUI now provides the ability to perform the CICS management tasks supported by the CICSPlex SM TSO end user interface (EUI). As previously announced, the EUI has therefore been removed from CICS Transaction Server for z/OS, Version 3 Release 2.

All of the functionality of the MVS/TSO ISPF end user interface has been removed. This includes all associated views, panels, menus and action commands, together with the supporting CAS and all PlexManager functions. Equivalent functionality is available solely via the CICSPlex SM Web User Interface. Note there is no WUI equivalent function for the temporary maintenance point CMAS function of the EUI.

With the removal of all EUI-related components, the entire CICSPlex SM installation process has been redesigned to make it an integral part of the installation of CICS Transaction server. See “CICSPlex SM installation integrated with CICS installation” on page 153.

New data type SCLOCK12

The new data type SCLOCK12 is introduced in CICS Transaction Server for z/OS, Version 3 Release 2.

SCLOCK12

CICS monitoring facility (CMF) 12 byte interval store clock. Maintained internally as a binary value.

The first 8 bytes contain the time accumulated by the clock, and they are displayed externally as a formatted value, with the default format `HHHH:MM:SS.thmiju` (where *t* is tenths of seconds, *h* is hundredths of seconds, *m* is milliseconds, *i* is ten-thousandths of seconds, *j* is hundred-thousandths of seconds, and *u* is microseconds).

The last 4 bytes contain a count of the measurement periods during which the time was accumulated. The count can be displayed externally by selecting the count formatting option for the attribute.

EXEC CPSM API programs have access to the entire internal SCLOCK12 data value, but REXX applications only have access to the first 8 bytes containing the time.

When specified in an RTA EVALDEF, the last 4 bytes containing the count are not available. The other data must be entered in one of the following formats, with leading zeros, if necessary:

1. `HH:MM:SS`
2. `HH:MM:SS.thmi`
3. `HHHH:MM:SS.thmi`
4. `HHHH:MM:SS`

© Copyright IBM Corp. 2001, 2007

5. `HHHH:MM:SS.thmiju`

Only the first **three** of these formats are compatible with earlier releases of CICSPlex SM. If you need to use an EVALDEF involving SCLOCK12 data with an earlier release of CICSPlex SM, do not use format 4 or 5. This applies to EVALDEFs installed directly on a back-level CICSPlex SM system, and also to EVALDEFs installed as part of a batched repository update job (BATCHREP) or using the EYU9XDBT utility. For more information about clocks see the *CICS Performance Guide*.

The numeric value representing the internal data type for SCLOCK12 is 152.

Like the existing data type SCLOCK (the 8 byte interval store clock), the new data type SCLOCK12 can be used as a filter on the DATA/GET command, and it can be used when specifying summary expressions.

In views, SCLOCK12 is treated in the same way as SCLOCK. The time can be displayed in a number of different formats, and the count of measurement periods can also be displayed.

Resource table attributes converted to SCLOCK12 data type

Some resource table attributes that had the data type SCLOCK have been converted to the new data type SCLOCK12. The resource tables in which attributes have been converted are:

- TASK
- HTASK
- TASKRMI

Where a count of measurement periods was available for the SCLOCK data type before conversion, it is also available for the SCLOCK12 data type after conversion.

Application programs must be recompiled if they extract data from these resource tables using EXEC CPSM GET commands, Web User Interface server DATA/GET commands, or REXX TPARSE and TBUILD commands.

New time formatting options for clock data in CICSPlex SM views

Attributes with the data type SCLOCK12, which use the 12 byte CMF interval store clock, can be displayed in any of the time formats. For attributes with the data type SCLOCK (the 8 byte store clock), you can only use certain time formats.

In the time formats that include fractions of a second, `t` is tenths of seconds, `h` is hundredths of seconds, `m` is milliseconds, `i` is ten-thousandths of seconds, `j` is hundred-thousandths of seconds, and `u` is microseconds.

The time formats are:

- `HHHH:MM:SS.thmiju`, which shows a 4 digit count for hours, and displays the time to 6 decimal places (down to one microsecond). This is the default format for the data type SCLOCK12. It is not available for SCLOCK. This format is the same as the format used in the CICS statistics reports.
- `DDD.HH:MM:SS.thmiju`, which shows a count for days, and displays the time to 6 decimal places (down to one microsecond). This format is available for the data type SCLOCK12. It is not available for SCLOCK.

- `HH:MM:SS.thmi`, which shows a 2 digit count for hours, and displays the time to 4 decimal places (down to one ten-thousandth of a second). This is the default format for the data type SCLOCK, and it is also available for SCLOCK12.
- `HH:MM:SS`, which shows a 2 digit count for hours, and no decimal places. This format is available for both the data types SCLOCK and SCLOCK12.

The longer time formats `hhhh:mm:ss.thmiju` and `ddd.hh:mm:ss.thmiju` are new.

Choose one of the longer time formats for larger time values, such as those for long-running tasks, or for time values where you need maximum precision.

For attributes with the data types SCLOCK12 and SCLOCK, you can also display a count. The count is taken from the last 4 bytes of the clock data. It gives the number of measurement periods during which the time recorded by the timer component of the clock was accumulated.

You can use the CICSPlex SM Web User Interface view editor to customize your views to use the new time formats. You can edit view components from the **Tabular View Components** screen (select the **Table contents** option on that screen) or the **Detailed Form Components** screen (select **Form contents**). On the **Table contents** or **Form contents** screen, click **Append** or **Insert** to see the list of available view items. If the new time formats are available for an attribute in the view, the list displays the attribute with the new time formats, as well as with the old formats. Select the attribute with an appropriate time format to add it to your view. Then delete the attribute with the old time format from your view.

Changes to CICSPlex SM Web User Interface security

In CICS TS for z/OS, Version 3.2 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 appropriate 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..

`wui_server_applid` is the CICS APPLID of the server.

Users can be given read or update access to views and menus:

- Read access allows users to use the views or menus in the main interface. This enables you to prepare and protect views for specific user groups.
- Update access allows users to create, update or remove items in the view editor or import using COVC. This enables 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.

Obsolete CICSPlex SM views, resource tables and attributes

These CICSPlex SM views and resource tables have had certain functions removed, or have been removed completely, because of changes to CICS resource types and functions.

In the operations view **Enterprise Java component operations views > Java virtual machine (JVM) pool**, the field ″Number of JVM requests with JVM reset″ is displayed as ″Not applicable″ for CICS Transaction Server for z/OS, Version 3 Release 2 regions. The corresponding SJGREQSRESET attribute in the JVMPOOL resource table returns ″Not applicable″ for CICS Transaction Server for z/OS, Version 3 Release 2 regions.

In the operations view **Enterprise Java component operations views > Java virtual machine (JVM) profile**, the fields ″Number of CICS key JVMs not resettable″ and ″Number of USER key JVMs not resettable″ are displayed as ″Not applicable″ for CICS Transaction Server for z/OS, Version 3 Release 2 regions. The corresponding CJVMSUNRESET and UJVMSUNRESET attributes in the JVMPROFILE resource table return ″Not applicable″ for CICS Transaction Server for z/OS, Version 3 Release 2 regions.

In the Business Application Services (BAS) view **Administration views > Basic CICS resource administration views > CICS resource definitions > Program definitions**, and the operations view **Program operations views > Programs**, the HOTPOOLING attribute is not valid from CICS Transaction Server for z/OS, Version 3 Release 1

In the Business Application Services (BAS) view **Administration views > Basic CICS resource administration views > CICS resource definitions > Terminal definitions**, the CONSOLE attribute is not valid from CICS Transaction Server for z/OS, Version 3 Release 1.

CICSPlex SM resource tables no longer supported in CICS Transaction Server for z/OS, Version 3 Release 2

As previously announced, a number of CICSPlex SM resource tables are no longer supported in CICS Transaction Server for z/OS, Version 3 Release 2. The information in them has migrated to other tables.

The unsupported resource tables are XDSPGBL, XDSPPOOL, XJVMPOOL, XLSRPBUF, XMONITOR, XPROGRAM, XSTREAM, XTASK, and X2TASK.

The information they contained is available in other resource tables as follows:

XDSPGBL	DSPGBL
XDSPPOOL	DSPPOOL
XJVMPOOL	JVMPOOL
XLSRPBUF	LSRPBUF
XMONITOR	MONITOR

XPROGRAM	PROGRAM
XSTREAM	STREAMNM
XTASK	TASK
X2TASK	TASK

Edit and recompile your CICSPlex SM API programs to use the equivalent supported resource table. Before you migrate to CICS TS 3.2, recreate your WUI views and update your RTA definitions (EVALDEFs) to use the equivalent resource table on your earlier release.

Changed CICSPlex SM views and resource tables

These existing CICSPlex SM views and resource tables have been changed to support new or changed CICS resource types and functions.

Map function for resource definitions

All IBM-supplied tabular and detail views that display resource definitions now include a map button. The map function is invoked by clicking this button. It generates a visual representation of the associations between CICS resource definitions defined to CICSPlex SM for the selected resource. The map function is equivalent to the CICSPlex SM end user interface MAP command in releases of CICS TS where the EUI was provided.

Changed CICSPlex SM views

Table 33. Changed CICSPlex SM views

Changed CICS resource type or function	**Corresponding CICSPlex SM views which have changed**
Delay times for change mode and SSL TCBs	CICS operations views > Task operations views > Active tasks
Document template statistics and refresh (newcopy) function	CICS operations views > Document template operations views > Document template
Document deletion	CICS operations views > Task operations views > Active tasks History views > Completed tasks > task item > Request counts
Document templates in HFS files	CICS operations views > Document template operations views > Document template Administration views > Basic CICS resource administration views > Resource definitions > Document template definitions
ISC and MRO statistics	CICS operations views > Connections operations views > ISC and MRO connections
Java programs: use count and JVM profile	CICS operations views > Program operations views > Programs
JVMs: manual start up, and changes to termination	CICS operations views > Enterprise Java component operations views > Java virtual machine (JVM) pool

Table 33. Changed CICSPlex SM views (continued)

Changed CICS resource type or function	Corresponding CICSPlex SM views which have changed
JVMs: withdrawal of resettable mode	CICS operations views > Enterprise Java component operations views > Java virtual machine (JVM) poo CICS operations views > Enterprise Java component operations views > Java virtual machine (JVM) profile CICS operations views > Enterprise Java component operations views > Java virtual machine (JVM) status CICS operations views > Enterprise Java component operations views > JVM Class Cache status
LIBRARY resources	CICS operations views > Program operations views > Program
Local files	Local files > File ID
MVS workload manager statistics	CICS operations views > CICS region operations views > MVS workload management
Storage information for MVS TCBs	CICS operations views > CICS region operations views > MVS TCBs CICS operations views > CICS region operations views > Global MVS TCB information CICS operations views > CICS region operations views > MVS storage areas
TCP/IP service	CICS operations views > TCP/IP service operations views > TCP/IP service
TCPIPSERVICE resource definition attributes	Administration views > Basic CICS resource administration views > Resource definitions > TCP/IP service definitions
XCF group ID	CICS regions > region name
Internet security	CICS operations views > Enterprise Java component operations views > CorbaServers CICS operations views > TCP/IP service operations views > TCP/IP service CICS operations views > TCP/IP service operations views > TCP/IP global status Administration views > Basic CICS resource administration views > Resource definitions > Enterprise Java CORBASERVER definitions Administration views > Basic CICS resource administration views > Resource definitions > TCP/IP service definitions

Changed CICSPlex SM resource tables

Review these resource tables for possible impact on any RTA evaluation definitions (EVALDEF) or CICSPlex SM API programs that you are using.

- CLCACHE
- CMAS
- CICSRGN
- CONNECT
- DOCDEF
- DOCTEMP
- DSPGBL

- EJCODEF
- HTASK
- JVM
- JVMPOOL
- JVMPROF
- LIBDEF
- LIBDSN
- LIBINGRP
- LIBRARY
- LOCFILE
- MONITOR
- MVSESTG
- MVSTCB
- MVSTCBGL
- MVSWLM
- PROGDEF
- PROGRAM
- RESGROUP
- TASK
- TASKRMI
- TCPDEF
- TCPIPGBL
- TCPIPS
- TERMDEF
- TSMODEL (The TSMODEL base table attribute called DESCRIPTION in earlier releases, has been renamed to RSVRD1.)
- WORKREQ

New CICSPlex SM views and resource tables

These new views and resource tables have been added to CICSPlex SM to support CICS resource types and functions.

Table 34. New CICSPlex SM views and resource tables

Resource type or function	CICSPlex SM views	CICSPlex SM resource tables
Association data for TCP/IP-related tasks	CICS operations views > Association data operations views > Association data	TASKASSC
Batched repository update job	Administration views > General views > Batched repository update job	BATCHREP
CMASs and CICSplexes (this view previously supported only by the EUI)	Administration views > CMAS configuration administration views > CMAS in CICSplex definitions	CPLXCMAS (existing resource table)

Table 34. New CICSPlex SM views and resource tables (continued)

Resource type or function	CICSPlex SM views	CICSPlex SM resource tables
Transient data queues (this view previously supported only by the EUI)	CICS operations views > Transient data queue (TDQ) operations views > Topology data for transient data queue	CRESTDQ (existing resource table)
HOST resource definition (virtual hosts)	CICS operations views > TCP/IP service operations views > URI host	HOST
Historical data for tasks	EYUSTARTHTASK EYUSTARTMASHIST EYUSTARTTASKRMI	HTASK (existing resource table) MASHIST TASKRMI
IPIC connection	CICS operations views > Connection operations views > IPIC connections	IPCONN
WebSphere MQ connections	CICS operations views > DB2, DBCTL and WebSphere MQ operations views > WebSphere MQ connections	MQCONN
PIPELINE resource definition	CICS operations views > TCP/IP service operations views > Pipeline Administration views > Basic CICS resource definitions > Pipeline definitions	PIPELINE PIPEDEF CICS resource administration views >
Resource definitions in a resource group	Administration views > Basic CICS resource definitions in a resource group	BESSNGDR CICS resource administration views >
Channels	CICS operations views > Active tasks > Channel usage	TASK (existing resource table)
Task element storage	EYUSTARTTASKESTG	TASKESTG
Task file usage	EYUSTARTTASKFILE	TASKFILE
Task temporary storage queue usage	EYUSTARTTASKTSQ	TASKTSQ
URIMAP resource definition	TCP/IP service operations views > URIMAP global statistics Administration views > Basic CICS resource definitions > URI mapping definitions	URIMAP URIMPDEF CICS resource administration views >
URIMAP global statistics	CICS operations views > TCP/IP service operations views > URIMAP global statistics	URIMPGBL
Web services	CICS operations views > TCP/IP service operations views > Web service Administration views > Basic CICS resource definitions > Web service definitions	WEBSERV WEBSVDEF CICS resource administration views >

New Business Application Services definition objects

Table 35. New BAS definition objects

BAS object	What is it?
IPCINGRP	BAS definition that describes the membership of an IPIC connection definition (IPCONDEF) in a resource group.

Table 35. New BAS definition objects (continued)

BAS object	What is it?
IPCONDEF	CICS definition that describes an IPIC connection.
LIBINGRP	BAS definition that describes the membership of a LIBRARY definition (LIBDEF) in a resource group.
LIBDEF	CICS definition that describes an LIBRARY resource.

Changed Business Application Services definition objects

Table 36. Changed BAS definition objects

BAS object	Change
RASGNDEF	New REDEFTYPE values: IPCONDEF LIBDEF
RESDESC	New attributes added: IPCDEFRG IPCDEFTS IPCDEFRS LIBDEFRG LIBDEFTS LIBDEFRS
TCPDEF	• New attribute added: REALM • New value allowed for PROTOCOL attribute: IPIC

Chapter 29. National language support for CICSPlex SM messages

The capability of issuing CICSPlex SM messages, that have a destination of EYULOG, in national languages other than English, using the CICS message domain, has been added in CICS Transaction Server for z/OS, Version 3 Release 2. Also, the CICS XMEOUT global user exit has been enhanced to allow suppression and rerouting of CICSPlex SM messages that use the message domain. These messages may be suppressed or rerouted from the joblog or console but not from the EYULOG.

"Changes to global user exits" on page 56 lists the new fields added to XMEOUT to support this function.

Note: CICSPlex SM messages are not available through the CMAC transaction.

The following EYUPARMS have been removed:

- xxxCONMSG
- xxxTDQMSG

The following messages have been added to support the NLS-enablement of CICSPlex SM messages:

- EYUBM0329I
- EYUBM0330I
- EYUBM0331I
- EYUBM0332I
- EYUBM0333I
- EYUBM0334I
- EYUBM0335I
- EYUBM0336I
- EYUBM0337I
- EYUBM0338I
- EYUBM0339I
- EYUBM0340I
- EYUBM0341I
- EYUBM0342I
- EYUBM0343I
- EYUBM0344I
- EYUBM0345I
- EYUBM0346I
- EYUBM0347I
- EYUBM0348I
- EYUBN0013W
- EYUBN0014W
- EYUBN0015W
- EYUBN0016W
- EYUBN0017W

© Copyright IBM Corp. 2001, 2007

- EYUXL0030I
- EYUXL0031I
- EYUXL0032I

The following messages have been removed:

- EYUBM0322I
- EYUBM0323I
- EYUBM0324I
- EYUBM0325I
- EYUBM0326I
- EYUBM0327I
- EYUBN0012W
- EYUXL0020I

Chapter 30. 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.

© Copyright IBM Corp. 2001, 2007

Part 4. Migration tasks for CICSPlex SM

© Copyright IBM Corp. 2001, 2007

Chapter 31. Running CICSPlex SM Version 3.2 and earlier releases concurrently

You can run CICSPlex SM Version 3.2 and earlier releases concurrently, but note the conditions listed in this topic.

CICSPlex SM release names

Throughout this section, CICSPlex SM releases are referred to as follows:

Version 3.2
Version 3 Release 2 (the CICSPlex SM element of CICS Transaction Server for z/OS, Version 3 Release 2)

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)

The CICSPlex SM elements of CICS TS for z/OS, Version 3.2, CICS TS for z/OS, Version 3.1, CICS TS for z/OS, Version 2.3 and CICS TS for z/OS, Version 2.2 are **not** available as separate products.

You can run CICSPlex SM Version 3.2, Version 3.1, Version 2.3, and Version 2.2, at the same time, with interconnected CMASs at different levels. The ability to do this allows gradual migration of the environment to Version 3.2. However, in CICS TS for z/OS, Version 3.2, a CICSPlex SM CMAS will run only in a CICS system at the same release level.

MASs running the following CICS releases are directly connectable to CICSPlex SM Version 3.2:

- CICS TS for z/OS, Version 3.1
- CICS TS for z/OS, Version 2.3
- CICS TS for z/OS, Version 2.2
- CICS for Windows® component of IBM TXSeries® 4.3.0 (with PTF 4)
- CICS for Windows component of IBM TXSeries 5.0

Conditions for running CICSPlex SM Version 3.2 and earlier releases concurrently

The following conditions apply to environments in which CICSPlex SM Version 3.2 and earlier releases of CICSPlex SM are running concurrently:

- If you intend to use CICS TS 2.2 with CICS TS 3.2, you must install APAR PQ65168, PTF UQ71534 to that release:
- In order for 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:

© Copyright IBM Corp. 2001, 2007

  - 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 CMAS running at Version 3.2 can be connected to a CMAS running at Version 3.1, Version 2.3, or Version 2.2. However:
  - In a CICSplex that consists of CMASs at the Version 3.2 level and the Version 3.1, Version 2.3, or Version 2.2 level, the maintenance point CMAS must be at the Version 3.2 level. That is, when a CICSplex contains CMASs at more than one level, the first CMAS converted to Version 3.2 must be the maintenance point.
  - If you are using the API 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 3.2 CMAS. You must ensure that:
    - All API programs run so that they are connected to the Version 3.2 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 Web User Interface servers connect to the Version 3.2 CMAS.
  - You cannot view resources of a CICS Version 3.2 region using a CMAS running at an earlier release.
- The following definitions, if required, must be created using a WUI server or EUI running at the same CICSPlex SM release level as the maintenance point CMAS:
  - CPLEXDEF
  - CMTCMDEF
  - CSYSGRP
  - PERIODEF
  - MONSPEC
  - MONGROUP
  - MONDEF
  - RTAGROUP
  - RTADEF
  - WLMSPEC
  - WLMGROUP
  - WLMDEF
  - TRANGRP

  CICSPlex SM and the maintenance point CMAS release level must, again, be at the same release level if you use the API or BATCHREP to create these definitions.

Chapter 32. 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. 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.

© Copyright IBM Corp. 2001, 2007

Chapter 33. Performing migration procedures for CICSPlex SM

The migration from a previous release of CICSPlex SM to CICS TS Version 3.2 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.

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 .)

To enable you to revert to the previous release of CICSPlex SM if you encounter problems during the migration to CICS TS Version 3.2 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 3.1, Version 2.3, Version 2.2 or Release 4 of CICSPlex SM to CICS TS Version 3.2 CICSPlex SM.

Converting a CMAS to Version 3.2

You must migrate your CICSPlex SM CMAS to CICS TS Version 3.2 at the same time as you migrate the CICS system on which it runs. This is because 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.

Before you convert a CMAS to Version 3.2 ensure that you have updated the CSD file with the Version 3.2 group of resource definitions and CICS startup group list as part of your CICS migration procedures. For information about how to do this, see . You do not need to carry out an additional upgrade using a release-dependant set of definitions for CICSPlex SM

Now you need to do the following:

- 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 3.2 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 3.2 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 3.2 library. (For information about this step, see the *CICS Transaction Server for z/OS Installation Guide*.)
- Update the CSD file with the Version 3.2 group of resource definitions and CICS startup group list. (For information about how to do this, see “Upgrading the CSD for CICS-supplied and other IBM-supplied resource definitions” on page 86 and the the *CICS Transaction Server for z/OS Installation Guide*.)

© Copyright IBM Corp. 2001, 2007

- Make sure that the system initialization parameters XRES and XHFS are set to NO. The default for both is YES.
- Reassemble the CICS resource definition table load modules. Reference the library containing the Version 3.2 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 3.2. (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.
- 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 3.2 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 184.)

Converting a MAS to Version 3.2

Before you convert a MAS to Version 3.2 ensure that you have updated the CSD file with the Version 3.2 group of resource definitions and CICS startup group list as part of your CICS migration procedures. For information about how to do this, see . You do not need to carry out an additional upgrade using a release-dependant set of definitions for CICSPlex SM

Now you need to do the following:

- Authorize the Version 3.2 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 3.2 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*.)
- When previous release modules are in the link pack area (LPA), you must ensure the Version 3.2 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 System Manager library names to the Version 3.2 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.

- Ensure that you have deleted, redefined and initialized the CICS local catalog (LCD) and global catalog (GCD) using the DFHCCUTL and the DFHRMUTL utility programs.
- If you intend to use MAS history recording, it is recommended that you define new history data sets using the EYUJHIST sample job. If, however, you need to migrate your existing history data sets, you may still use the EYUJHIST sample job, following the migration instructions, supplied as comments, within the sample. The EYUJHIST sample is supplied uncustomized in the TDFHINST library and customized by DFHISTAR in the XDFHINST library. Remember to edit the MAS start up JCL to include the history data sets.

The MAS is ready to be cold started.

When you have successfully migrated all your systems to CICSPlex SM Version 3.2 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" on page 184.)

Migration for CICSPlex SM 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 3.2 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 3.2 libraries.

IPIC connections with workload management

CICSPlex SM workload management has been updated to support MASs connected with CICS IPIC connections (IPCONNs). There are no new external changes to workload management to exploit CICS IP connections but you may notice different behavior when CICS IPIC connections are introduced into a workload. For CICS TS Version 3.2, CICS IP connections can only be used by Distributed Program Link (DPL). If a routing request other than DPL is being executed, all normal workload management routing algorithms will apply. This means that MASs that are connected only by CICS IPIC connections can only support DPL requests.

Consider a simple workload installed on a routing region that needs to balance DPL requests to a pair of target regions. One target region is connected to the routing region via MRO, and the other target region is connected to the routing region via a CICS IPIC connection. In situations where the target regions are running at the same health and load levels, their connection link weights will be the deciding factor when determining a DPL target. In this situation, CICSPlex SM would always select the MRO-connected target region as the DPL target, because its link weight is less than that for IPIC connections. Now assume that the MRO-connected regions also have a CICS IPIC connection installed. If there are identical health and load levels for the target regions, routing requests may vary between the two target regions because CICS will always choose to use an IPIC connection for DPL requests over any other connection type. Therefore, when the MRO-connected target region had a CICS IPIC connection installed, this equalized the link weight of the DPL targets, as

they both now have a CICS IPIC connection. It is recommended that workloads are not actively spread across earlier CICS TS releases and a CICS TS Version 3.2 CICSplex. New workloads should be established for CTS Version 3.2 managed regions, and regions managed by earlier CICS TS releases should be isolated from them.

Application programming interface

CICSPlex SM API programs written to run in a previous release MAS can be run in a Version 3.2 MAS. You can either continue to access the data provided by the previous release or access the new data available from Version 3.2. 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 3.2

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.

It is advisable to increase the size of the DFHTEMP data set. This is used in the COVC import process. The standard CICS sample just has a primary allocation but you should include a secondary allocation for RECORDS as follows:

```
//DEFTS    JOB accounting info,name
//AUXTEMP  EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=A
//SYSIN    DD  *
           DEFINE CLUSTER(NAME(CICSTS32.CICS.CNTL.CICSqualifier.DFHTEMP)-
                  RECORDSIZE(4089,4089)            -
                  RECORDS(200 200)                 -
                  NONINDEXED                       -
                  CONTROLINTERVALSIZE(4096)        -
                  SHAREOPTIONS(2 3)                -
                  VOLUMES(volid))                  -
                DATA(NAME(CICSTS32.CICS.CNTL.CICSqualifier.DFHTEMP.DATA) -
                  UNIQUE)
/*
```

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 3.2 you should:

- Upgrade the CSD for CICS, see “Upgrading the CSD for CICS-supplied and other IBM-supplied resource definitions” on page 86.
- Migrate the MAS that acts as your Web User Interface server.
- 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 3.2 CMAS on the same MVS image as the Web User Interface server.
2. Connect the Version 3.2 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 3.2 and take down the Web User Interface server at the same time. Migrate the Web User Interface server to Version 3.2 and, when you restart it, connect it to the Version 3.2 CMAS. The Version 3.2 CMAS should now connect successfully to the Version 3.2 maintenance point CMAS.
4. Migrate the remaining MASs when required, and connect them to the Version 3.2 CMAS as you restart them.
5. When you have moved all the MASs to the Version 3.2 CMAS, you can remove the original CMAS.

Assuming you are running the latest CICSPlex SM 3.1 and 3.2 maintenance levels, you can convert one LPAR at a time from 3.1 to 3.2.

- A CICSPlex SM 3.2 WUI can only connect to a 3.2 CMAS.
- 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 3.2 CMAS (assuming it is not a resource type which is unique to CICS TS 3.2) 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 3.2 CMAS (assuming it is not a resource type which is unique to CICS TS 3.2) if the CMAS participates in the management of the CICSplex.
- A CICSPlex SM 3.2 WUI connected to a 3.2 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 3.2” on page 180. You must also update the CICS SIT GRPLIST parameter to reference the CICS-supplied default startup group list, DFHLIST.

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

InCICS TS for z/OS, Version 3.2 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.

You can import a view set and menu definitions from a previous release into a CICS TS for z/OS, Version 3.2 Web User Interface server repository.

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

1. 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* .
2. Create a new Web User Interface server repository using the JCL described in the *CICS Transaction Server for z/OS Installation Guide*.
3. Start the Web User Interface server at the new release using the new Web User Interface server repository.
4. Import the new starter set definitions (the supplied set of view set and menu definitions with names beginning EYUSTART).
5. Import your previous release view set definitions by specifying the data set that you had exporting them to on the AUTOIMPORTTDQ parameter making sure that the TDQ you specify references the data set that you exported to. This will need to be done for each type of resource (VIEW, MENU, USER, USERGRP, and so on) that you had previously customized. Now use COVC to import the new starter set definitions specifying the OVERWRITE option on the **Import option** field of the COVC panel to ensure that none of the new starter set views have been accidently overwritten by views from a previous release.

   For information about using COVC, see the *CICSPlex System Manager Web User Interface Guide*

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

You can also export view set and menu definitions from a CICS TS for z/OS, Version 3.2 Web User Interface server and import them into a server repository of a previous release. However, any new attributes or resources that are new in this release are not accessible in previous releases. You can 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 System Manager Version 3.2, you can delete the Version 3.1, Version 2.3 or Version 2.2 definitions from each CMAS's and MAS's CSD. This can be done by upgrading each CSD using module EYU9R*xxx*, where *xxx* is the release number for the previous release, for example EYU9R230 for Version 2.3. This module is supplied in CICSTS32.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(EYU9Rxxx)
/*
```

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

When this JCL is run, EYU9R*xxx* attempts to delete all the groups and group lists for that CICSPlex System Manager version 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*.

Chapter 34. A phased migration scenario

The figures “The environment,” “Objective 1: Add a WUI server at the earlier release” on page 189, “Objective 2: Convert MP CMAS to the new version” on page 190, “Objective 3: Convert CMAS B to the new version” on page 192, and “Objective 4: Convert CMAS C to the new version” on page 194 and the discussions that accompany them show a CICSPlex System Manager environment at an earlier release and the steps you would take to convert that environment to Version 3.2. 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

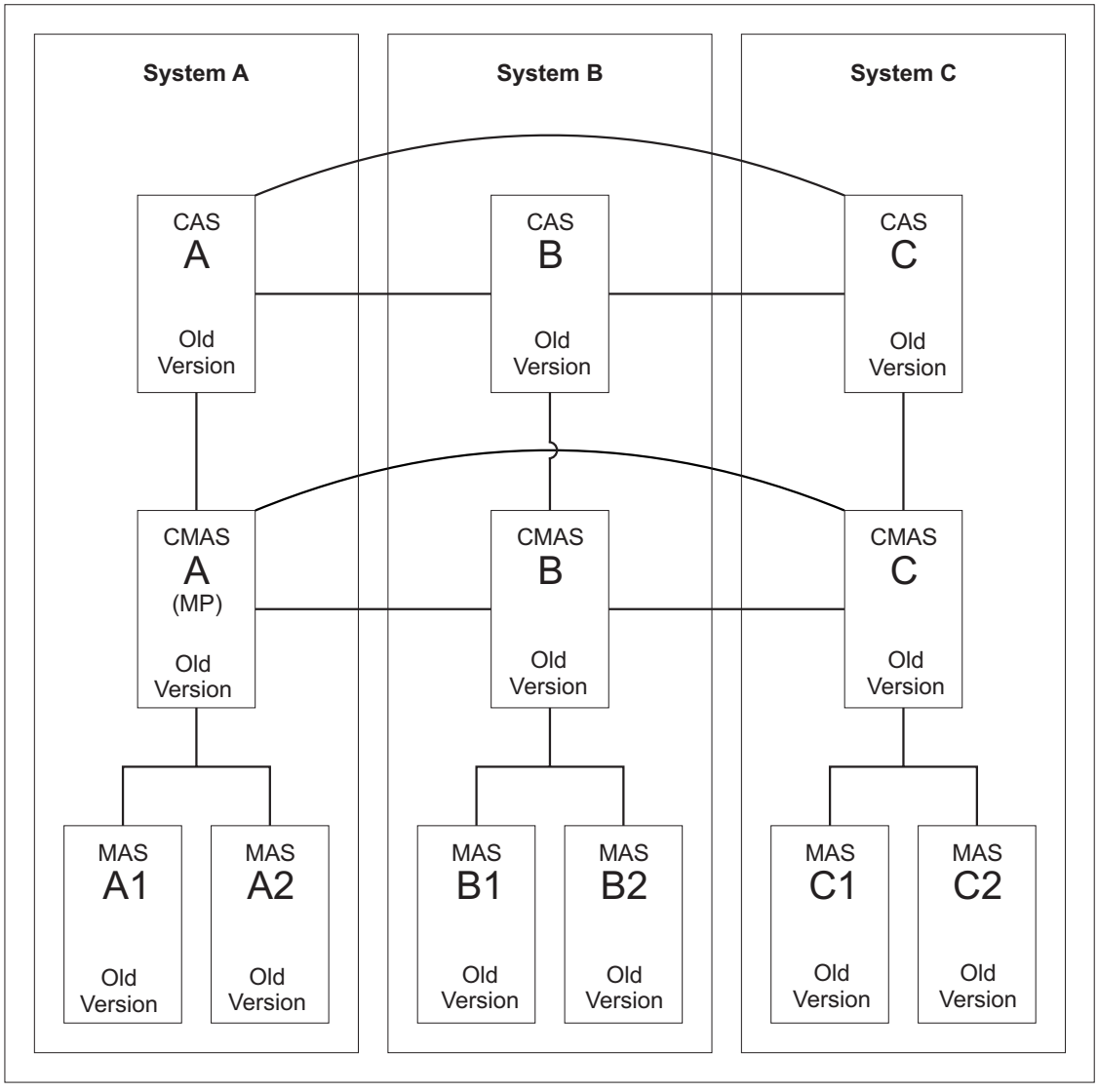

Figure 5. An environment at an earlier release

© Copyright IBM Corp. 2001, 2007

Figure 5 on page 187 shows a CICSPlex System Manager 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: Add a WUI server at the earlier release

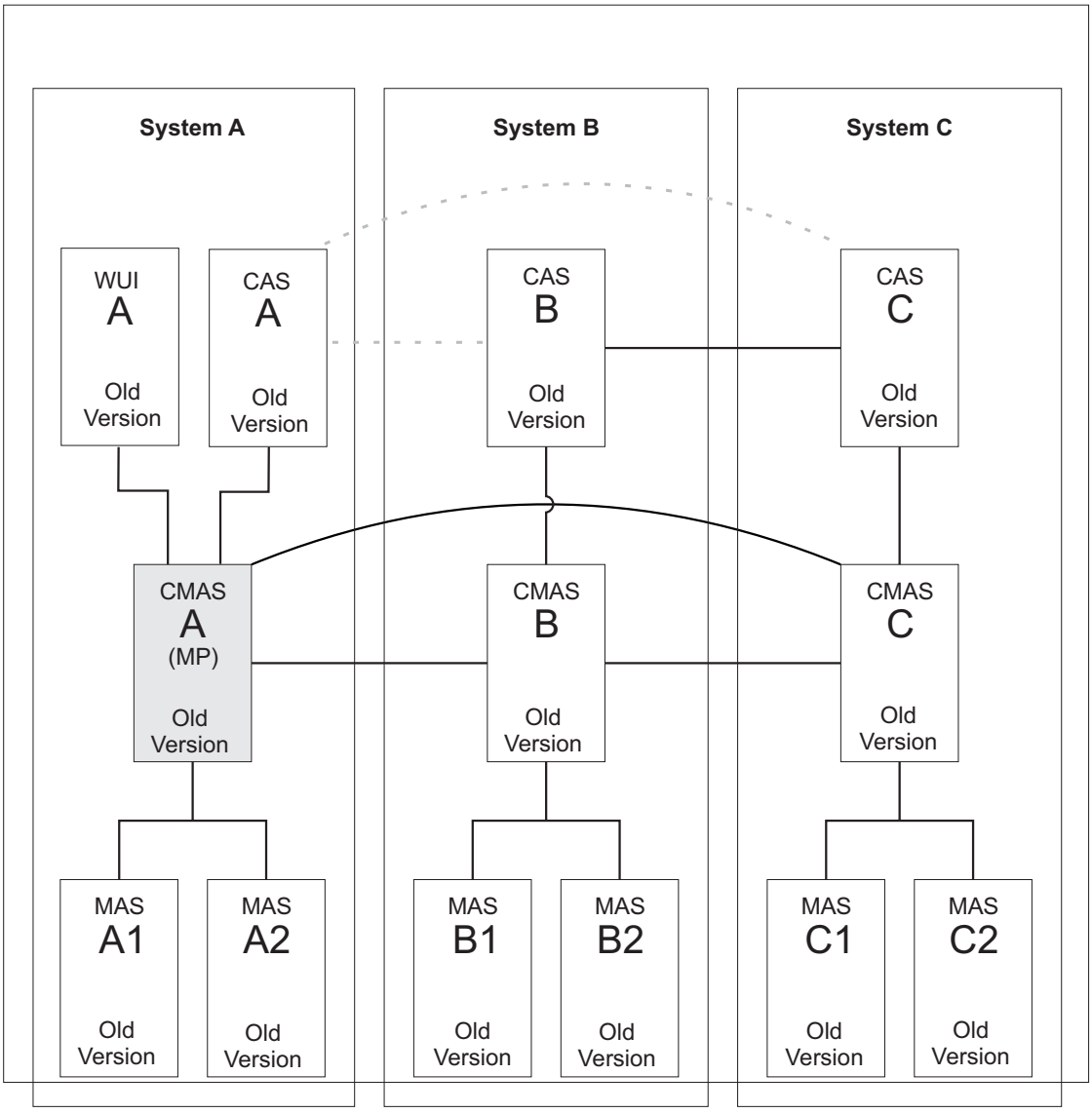

Figure 6. Adding a WUI server at the earlier release

When you complete Objective 1, a WUI at the earlier release level will have been connected to CMAS A.

The addition of a WUI to System A requires the following:

- Connect a WUI server to the maintenance point CMAS A at the old CICS Transaction Server release level.
- Create a separate CICSplex for the WUI server, defining CMAS A as the maintenance point.

Objective 2: Convert MP CMAS to the new version

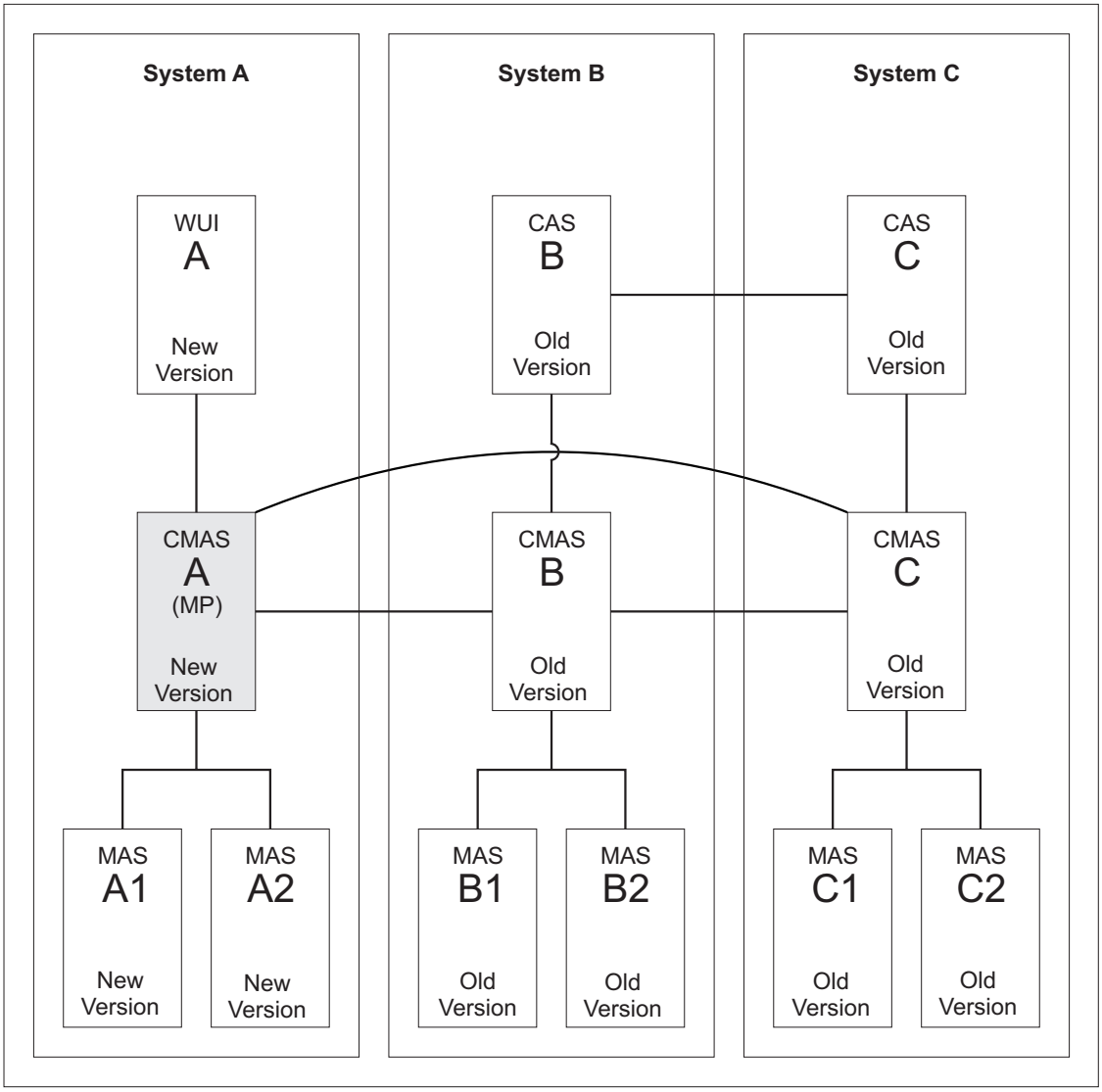

Figure 7. Converting the maintenance point CMAS to the new version

When you complete Objective 2, all CICS systems in System A are at the new version. The CMAS systems (A, B, and C) remain interconnected despite the different release levels they are at. CAS A and its connections are removed but CAS B and CAS C remain connected to one another.

The conversion of the maintenance point CMAS A to the new version requires conversion for the following:

- CMAS A
- WUI A
- MAS A1
- MAS A2

Step 1: Terminate executing regions that are to be converted

- If the following systems are running, terminate them:

- CMAS A
- WUI A
- MAS A1
- MAS A2

Step 2: Convert CMAS A to the new version

- Upgrade the CSD for CICS, see “Upgrading the CSD for CICS-supplied and other IBM-supplied resource definitions” on page 86.
- Change the appropriate IEAAPFxx member of the SYS1.PARMLIB library to authorize the CICSTS32.CPSM.SEYUAUTH library.
- Ensure that modules EYU9A320 and EYU9X320 in the CICSTS32.CPSM.SEYULINK data set are in the MVS link-list concatenation.
- Update the CICS group list for CMAS A to use DFHLIST.
- 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.
- Delete CASNAME from the EYUPARM parameters.
- Delete the BBACTDEF, BBVDEF, BBIPARM DD statements from the CMAS startup JCL.
- Start CMAS A.

Step 3: Convert WUI A to the new version

- Update the CICS group lists to use DFHLIST.
- Update the JCL used to start WUI A to point to the new data sets.
- Start WUI A.

Step 4: Convert MAS A1 and MAS A2 to the new version

- Update the CICS group lists for MAS A1 and MAS A2 to use DFHLIST and the lists of definitions for your own applications.
- 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 3: Convert CMAS B to the new version

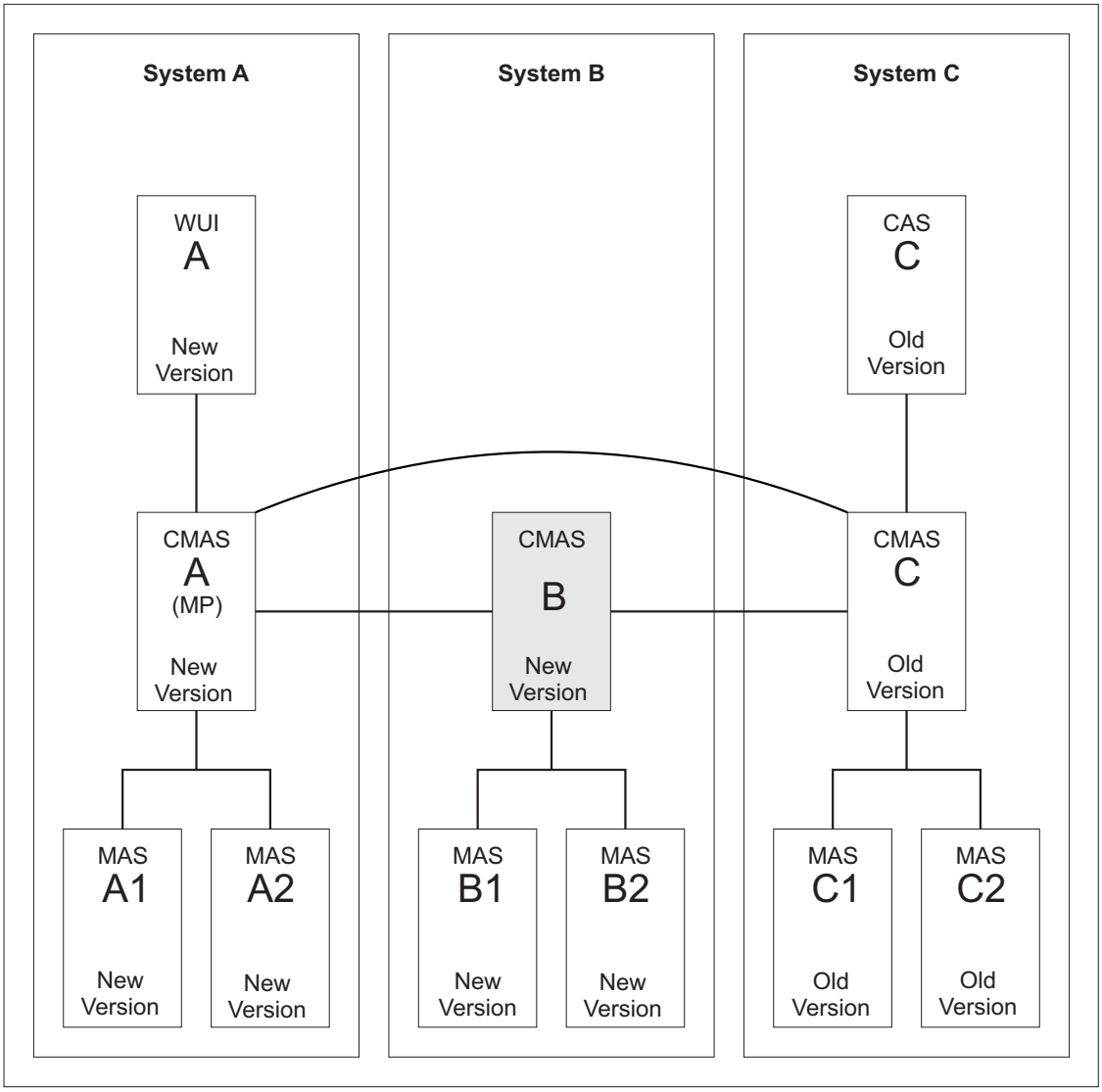

Figure 8. Converting CMAS B to the new version

When you complete Objective 3, all CICS systems in System B are at the new version. The CMAS systems (A, B, and C) remain interconnected despite being at different release levels. CAS B and its connection to CAS C are removed.

The conversion of CMAS B to the new version requires conversion for the following:

- CMAS B
- MAS B1
- MAS B2

Step 1: Terminate executing regions that are to be converted

- If the following systems are running, terminate them:
  - CMAS B
  - MAS B1

– MAS B2

Step 2: Convert CMAS B to the new version

- Change the appropriate IEAAPFxx member of the SYS1.PARMLIB library to authorize the CICSTS32.CPSM.SEYUAUTH library.
- Ensure that modules EYU9A320 and EYU9X320 in the CICSTS32.CPSM.SEYULINK data set is in the MVS link-list concatenation.
- Update the CICS group list for CMAS B to use DFHLIST.
- 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.
- Delete CASNAME from the EYUPARM parameters.
- Delete the BBACTDEF, BBVDEF and BBTPARM DD statements from the CMAS startup JCL.
- Start CMAS B.

Step 3: Convert MAS B1 and MAS B2 to the new version

- Update the CICS group lists for MAS B1 and MAS B2 to use DFHLIST and the lists of definitions for your own applications.
- Update the JCL used to start MAS B1 and for MAS B2 to point to the new data sets.
- Start MAS B1 and MAS B2.

Objective 4: Convert CMAS C to the new version

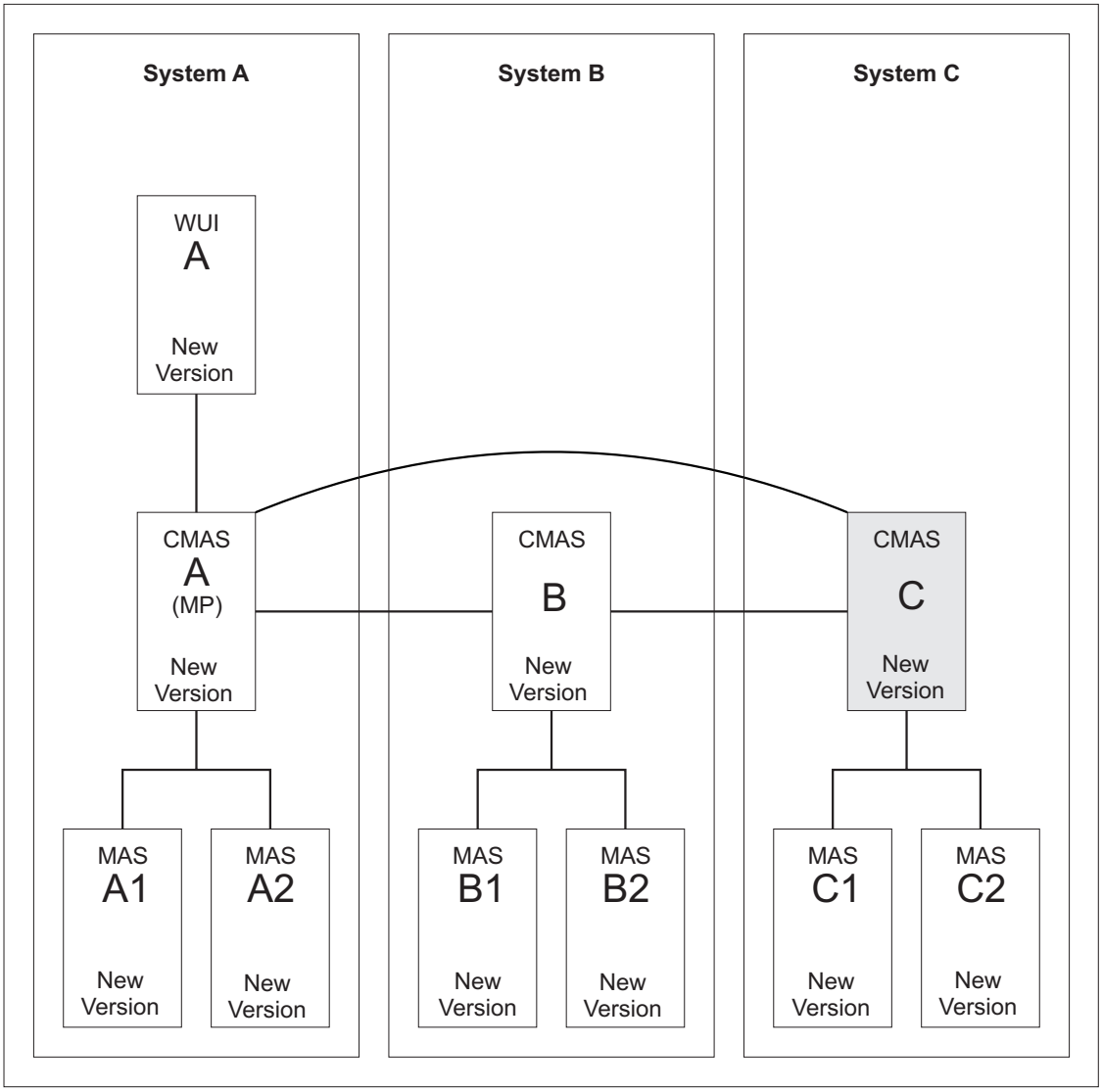

Figure 9. Converting CMAS C to the new version

When you complete Objective 4, all CICS systems are at the new version. All CASs will have been removed.

The conversion of CMAS C to the new version requires conversion for the following:

- 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:
  - CMAS C
  - MAS C1
  - MAS C2

Step 2: Convert CMAS C to the new version

- Change the appropriate IEAAPFxx member of the SYS1.PARMLIB library to authorize the CICSTS32.CPSM.SEYUAUTH library.
- Ensure that modules EYU9A320 and EYU9X320 in the CICSTS32.CPSM.SEYULINK data set are in the MVS link-list concatenation.
- Update the CICS group list for CMAS C to use DFHLIST.
- 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.
- Delete CASNAME from the EYUPARM parameters.
- Delete the BBACTDEF, BBVDEF, BBIPARM DD statements from the CMAS startup JCL.
- Start CMAS C.

Step 3: Convert MAS C1 and MAS C2 to the new version

- Update the CICS group lists for MAS C1 and MAS C2 to use DFHLIST and the lists of definitions for your own applications.
- Update the JCL used to start MAS C1 for MAS C2 to point to the new data sets.
- Start MASs C1 and C2.

Part 5. Appendixes

© Copyright IBM Corp. 2001, 2007

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 2:

Memo to Licensees, GI10-2559
CICS Transaction Server for z/OS Program Directory, GI13-0515
CICS Transaction Server for z/OS Release Guide, GC34-6811
CICS Transaction Server for z/OS Installation Guide, GC34-6812
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, GI13-0515
CICS Transaction Server for z/OS Release Guide, GC34-6811
CICS Transaction Server for z/OS Migration from CICS TS Version 3.1, GC34-6858

© Copyright IBM Corp. 2001, 2007

CICS Transaction Server for z/OS Migration from CICS TS Version 1.3, GC34-6855
CICS Transaction Server for z/OS Migration from CICS TS Version 2.2, GC34-6856
CICS Transaction Server for z/OS Installation Guide, GC34-6812

Administration

CICS System Definition Guide, SC34-6813
CICS Customization Guide, SC34-6814
CICS Resource Definition Guide, SC34-6815
CICS Operations and Utilities Guide, SC34-6816
CICS Supplied Transactions, SC34-6817

Programming

CICS Application Programming Guide, SC34-6818
CICS Application Programming Reference, SC34-6819
CICS System Programming Reference, SC34-6820
CICS Front End Programming Interface User's Guide, SC34-6821
CICS C++ OO Class Libraries, SC34-6822
CICS Distributed Transaction Programming Guide, SC34-6823
CICS Business Transaction Services, SC34-6824
Java Applications in CICS, SC34-6825
JCICS Class Reference, SC34-6001

Diagnosis

CICS Problem Determination Guide, SC34-6826
CICS Messages and Codes, GC34-6827
CICS Diagnosis Reference, GC34-6862
CICS Data Areas, GC34-6863-00
CICS Trace Entries, SC34-6828
CICS Supplementary Data Areas, GC34-6864-00

Communication

CICS Intercommunication Guide, SC34-6829
CICS External Interfaces Guide, SC34-6830
CICS Internet Guide, SC34-6831

Special topics

CICS Recovery and Restart Guide, SC34-6832
CICS Performance Guide, SC34-6833
CICS IMS Database Control Guide, SC34-6834
CICS RACF Security Guide, SC34-6835
CICS Shared Data Tables Guide, SC34-6836
CICS DB2 Guide, SC34-6837
CICS Debugging Tools Interfaces Reference, GC34-6865

CICSPlex SM books for CICS Transaction Server for z/OS

General

CICSPlex SM Concepts and Planning, SC34-6839
CICSPlex SM User Interface Guide, SC34-6840
CICSPlex SM Web User Interface Guide, SC34-6841

Administration and Management

CICSPlex SM Administration, SC34-6842
CICSPlex SM Operations Views Reference, SC34-6843
CICSPlex SM Monitor Views Reference, SC34-6844
CICSPlex SM Managing Workloads, SC34-6845
CICSPlex SM Managing Resource Usage, SC34-6846
CICSPlex SM Managing Business Applications, SC34-6847

Programming

CICSPlex SM Application Programming Guide, SC34-6848
CICSPlex SM Application Programming Reference, SC34-6849

Diagnosis

CICSPlex SM Resource Tables Reference, SC34-6850
CICSPlex SM Messages and Codes, GC34-6851
CICSPlex SM Problem Determination, GC34-6852

CICS family books

Communication

CICS Family: Interproduct Communication, SC34-6853
CICS Family: Communicating from CICS on zSeries, SC34-6854

Licensed publications

The following licensed publications are not included in the unlicensed version of the Information Center:

CICS Diagnosis Reference, GC34-6862
CICS Data Areas, GC34-6863-00
CICS Supplementary Data Areas, GC34-6864-00
CICS Debugging Tools Interfaces Reference, GC34-6865

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 2.

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.

© Copyright IBM Corp. 2001, 2007

Index

A

ACB interface 97
ACF/VTAM 97
AIBRIDGE system initialization parameter 9
Application Associated Data exit, new 59
application programming interface
  changed commands 33
  new commands 36
  obsolete commands and options 33
application programs
  compiler support 91
ATTACHSEC 17
AUTHENTICATE option
  WEB SEND command (Client) 35
AUTOIMPORTDSN 155
AUTOIMPORTMEM 155

B

BBM9ZA00 program 159
BRMAXKEEPTIME system initialization parameter 9
BTAM networks 97
Business Transaction Services (BTS)
  migration 95

C

CACHESIZE option
  INQUIRE DOCTEMPLATE command 43
CASNAME 159
CCI Connector for CICS TS 130
CCRL transaction 23
CCSID
  CEMT INQUIRE WEBSERVICE 18
CCSID option
  INQUIRE WEBSERVICE command 48
CEMN transaction 23
CEMT, changes 13
certificate revocation list transaction (CCRL) 23
CETR, changes 21
changed CEMT commands 13
changed global user exit programs
  XRSINDI 56
changed system initialization parameters 8
changes
  affecting the SPI 41
channels
  migration 99
    back-level CICS releases 101
    dynamically-routed applications 101
    LINK command 99
    RETURN command 100
    START command 100
    XCTL command 100
CICS Connector for CICS TS 130
CICS JVM Application Isolation Utility
  -verbose option 126
CICS JVM Application Isolation Utility *(continued)*
  sample report 124, 125
CICS Web support
  migration 141, 145
CICS-supplied transactions
  changed CEMT commands 13
  changes to CEMT 13
  changes to CETR 21
  changes to CWXN 22
  CICS-supplied transactions
    DFH$CAT1 CLIST 25
  CJGC 25
  CJPI 25
  CPIR 25
  CRTP 25
  CWXU 25
  migration 13
  new CEMT commands 20
  new RACF category 1 transactions 25
  obsolete CEMT command options 13
CICSPlex SM
  connecting to previous releases 171
  new BAS definition objects 166
CIDDOMAIN
  CEMT INQUIRE PIPELINE 15
CLINTCP system initialization parameter 9
COMMAREA
  migration to channels and containers 99
    back-level CICS releases 101
    dynamically-routed applications 101
    LINK command 99
    RETURN command 100
    START command 100
    XCTL command 100
compiler support 91
COMPRESS
  CEMT INQUIRE MONITOR 15
  CEMT SET MONITOR 19
COMPRESSST
  CEMT INQUIRE MONITOR 15
COMPRESSST option
  INQUIRE MONITOR command 44
  SET MONITOR command 49
connector applications 130
containers
  migration 99
    back-level CICS releases 101
    dynamically-routed applications 101
    LINK command 99
    RETURN command 100
    START command 100
    XCTL command 100
control tables
  migration 31
CRLSERVER system initialization parameter 9
CSD
  sharing between releases 89
CSD, upgrading 86

© Copyright IBM Corp. 2001, 2007

CSD, upgrading *(continued)*
SCAN function 88
CVDA values
BASICAUTH
WEB SEND command (Client) 35
DOCDELETE
WEB SEND command (Client) 36
NODOCDELETE
WEB SEND command (Client) 36
NONE
WEB SEND command (Client) 35
CWXN, changes 22

D

data conversion
migration 85
DCB interface 97
DEFAULTMAPBAS 155
DEFAULTMAPCOLL 155
DEFAULTMAPMON 155
DEFAULTMAPRTA 155
DEFAULTMAPWLM 156
defining IPIC connections 5
DFH$AFFY (obsolete CSD group) 27
DFH$JAVA (obsolete CSD group) 27
DFH$MOLS sample print program
control statements
EXPAND 75
DFH$WEB, CSD group 30
DFHAPXPO 65
DFHAUGRP (obsolete CSD group) 27
DFHCNV 141
migration 31, 65
DFHCSDUP
migration 53
DFHCSVCmigration 113
DFHDCT, obsolete 31
DFHDP, CSD group 30
DFHDPWB, CSD group 30
DFHDYPDS 99
DFHIRPmigration 113
DFHISAIP 66
DFHISTAR
CICSPlex SM installation parameters 3, 153
dfhjaiu.jar - CICS JVM Application Isolation Utility 124
DFHJIIRP 28
DFHJVM DD card 120
DFHJVMCD 120
DFHJVMRO 120
DFHLRQ data set 95
migration 95
DFHMCT monitoring control table
migration 31
DFHPDxxx
migration 53
DFHSIT, default system initialization table 7
DFHSJJ8O 120
DFHSO, CSD group 30
DFHSTUP
migration 53
DFHTUxxx
migration 53
DFHUEPAR
migration 56
DFHWBCLI 141
DFHWBEP
migration 141
dfjjvmcd.props 120
DOCSTATUS option
WEB SEND command (Client) 36
DOCTEMPLATE CVDA value
EXTRACT STATISTICS command 43
DOCTEMPLATE option
PERFORM STATISTICS command 49
DSKJRNL 162
dynamically-routed applications
migration to channels and containers 101

E

enterprise beans
migration 119
CICS Connector for CICS TS 130
Enterprise Workload Manager (EWLM)
migration 117
ESDS
migration to extended addressing 103
EWLM (Enterprise Workload Manager)
migration 117
EXEC CICS commands
API commands and options, obsolete 33
API commands, changed 33
API commands, new 36
CVDA values, obsolete 42
SPI command options, obsolete 41
SPI commands, changed 42
SPI commands, new 50
EXEC CICS WEB API
migration 141
exit programming interface (XPI)
migration 60
EXTENDED
CEMT INQUIRE FILE 15
extended relative byte addressing (XRBA)
migration 103
external CICS interface changes 63
EYU9XDUT utility 155
EYUISTAR 3, 153
EYUXL0206E message 159

F

FROMCODEPAGE option
GET CONTAINER (CHANNEL) command 34

G

global user exits
changed programs 56
migration 56
new programs 59

global user exits *(continued)*
obsolete global user exit points 55

H

hot-pooling 130
HPJ 130
HTTP client open exit, new 60
HTTP client send exit, new 59, 60

I

INQUIRE TCPIPSERVICE 17
installation
changes as a result of CICSPlex SM EUI removal 159
EUI removal
installation changes 159
Internet security
migration 141, 145
interregion communication program (DFHIRP)migration 113
intersystem communication over TCP/IP (IPIC)
defining connections 5
INTOCODEPAGE option
GET CONTAINER (CHANNEL) command 34
IPCINGRP, new BAS object 166
IPCONDEF, new BAS object 167
IPCONN
CEMT DISCARD 14
IPCONN CVDA value
EXTRACT STATISTICS command 43
IPCONN option
PERFORM STATISTICS command 49
IPFACILITIES option
INQUIRE TASK command 47
IPFLISTSIZE option
INQUIRE TASK command 47
IPIC value
INQUIRE TCPIPSERVICE command 48

J

Java
migration 120
Java applications
migration 119
Java program objects 130
JOURNAL 162
JRNINGRP 162
JRNLDEF 162
JVM
migration 120
resettable (no longer used) 120
JVM Application Isolation Utility 124
-verbose option 126
sample report 125
JVM profiles
migration 119

L

LIBDEF, new BAS object 167
LIBINGRP, new BAS object 167
LIBRARY
CEMT DISCARD 14
CEMT PERFORM STATISTICS 19
LIBRARY CVDA value
EXTRACT STATISTICS command 43
LIBRARY option
PERFORM STATISTICS command 49
LINK
migration to channels and containers 99
LOCALCCSID system initialization parameter 11

M

macro resource definition
migration 31
Mappinglevel
CEMT INQUIRE WEBSERVICE 18
MAPPINGLEVEL option
INQUIRE WEBSERVICE 48
MAPPINGRNUM option
INQUIRE WEBSERVICE 48
MAPPINGVNUM option
INQUIRE WEBSERVICE 48
MAXHPTCBS system initialization parameter 7
MAXSSLTCBS system initialization parameter 12
MAXXPTCBS system initialization parameter 12
MEMLIMIT
CEMT INQUIRE DSAS 14
MEMLIMIT option
INQUIRE SYSTEM command 46
migration
Open Transaction Environment (OTE) 133, 135
threadsafe programming 133, 135
XPLINK option 149
migration of CICSPlex SM
migrating a CMAS 179
migrating a MAS 180
Minrunlevel
CEMT INQUIRE WEBSERVICE 18
MINRUNLEVEL option
INQUIRE WEBSERVICE 48
MINRUNRNUM option
INQUIRE WEBSERVICE 48
MINRUNVNUM option
INQUIRE WEBSERVICE 48
MODE
CEMT INQUIRE PIPELINE 15
monitoring control table, DFHMCT
migration 31
monitoring facility transaction CEMN 23
MQCONN CVDA value
EXTRACT STATISTICS command 43
MQCONN option
PERFORM STATISTICS command 49
MRO (multiregion operation)migration 113
MTOMNOXOPST
CEMT INQUIRE PIPELINE 16

MTOMST
  CEMT INQUIRE PIPELINE 16
multiregion operation (MRO)migration 113

N

new BAS definition objects
  IPCINGRP 166
  IPCONDEF 167
  LIBDEF 167
  LIBINGRP 167
new CEMT commands 20
new CSD groups
  DFH$WEB 30
  DFHDP 30
  DFHDPWB 30
  DFHSO 30
new RDO attributes 28
new system initialization parameters 9
new user-exit programs 59
NOCOMPRESS
  CEMT INQUIRE MONITOR 15
  CEMT SET MONITOR 19
NOTAPPLIC
  CEMT INQUIRE FILE 15
NOTEXTENDED
  CEMT INQUIRE FILE 15
NOTSOS
  CEMT INQUIRE DSAS 14, 17
  CEMT INQUIRE SYSTEM 17

O

obsolete CSD groups
  DFH$AFFY 27
  DFH$JAVA 27
  DFHAUGRP 27
obsolete system initialization parameters 7
obsolete transactions
  XLEC 154
OO COBOL 91
open TCBs
  accounting 138
open transaction environment (OTE)
  processor times for transactions 138
Open Transaction Environment (OTE)
  migration 133, 135
OS/VS COBOL 91

P

PASSWORD option
  WEB SEND command (Client) 35
PASSWORDLEN option
  WEB SEND command (Client) 35
PIPELINE definition
  RESPWAIT attribute 28, 43
processor times for CICS and DB2 138
program compatibility, SPI 41

R

RBATYPE
  CEMT INQUIRE FILE 15
RBATYPE option
  INQUIRE FILE command 44
RDO
  new attributes 28
REALM attribute
  TCPIPSERVICE definition 29
REALM option
  INQUIRE TCPIPSERVICE command 43, 47
  WEB EXTRACT command 35
REALMLEN option
  WEB EXTRACT command 35
relative byte addressing (RBA)
  migration 103
remote MAS
  removal of support 177
repository data set 95
  migration 95
resettable JVM
  migration 122, 124, 125, 126
  withdrawal 120
resource definition
  changes 27
resource definition (online)
  new attributes 28
  upgrading the CSD
    SCAN function 88
RESPWAIT
  CEMT INQUIRE PIPELINE 16
  CEMT SET PIPELINE 19
RESPWAIT attribute
  PIPELINE definition 28, 43
RETURN
  migration to channels and containers 100

S

SCEERUN 120
SCEERUN2 120
SDFJAUTH 120
SENDMTOMST
  CEMT INQUIRE PIPELINE 16
SIT (system initialization table) parameters 7
SIZE
  CEMT INQUIRE DOCTEMPLATE 14
SMF 110 records
  migration 67
SMFJRNL 162
SOAPlevel
  CEMT INQUIRE PIPELINE 16
SOCKETCLOSE
  migration 141
SOS
  CEMT INQUIRE DSAS 14, 17
  CEMT INQUIRE SYSTEM 17
SOSABOVEBAR
  CEMT INQUIRE DSAS 14, 17

SOSABOVEBAR option
  INQUIRE SYSTEM command 46
SOSABOVELINE
  CEMT INQUIRE DSAS 14
  CEMT INQUIRE SYSTEM 17
SOSABOVELINE option
  INQUIRE SYSTEM command 47
SOSBELOWLINE
  CEMT INQUIRE DSAS 14
  CEMT INQUIRE SYSTEM 17
SOSBELOWLINE option
  INQUIRE SYSTEM command 47
SRVERCP system initialization parameter 12
SSL
  migration 141, 145
SSLCACHE system initialization parameter 12
SSLTCBS system initialization parameter 7
START
  migration to channels and containers 100
static variables in Java applications 122, 124
statistics records 79
system initialization parameters 7
  AIBRIDGE 9
  BRMAXKEEPTIME 9
  changed 8
  CLINTCP 9
  CRLSERVER 9
  LOCALCCSID 11
  MAXHPTCBS (obsolete) 7
  MAXSSLTCBS 12
  MAXXPTCBS 12
  new 9
  obsolete 7
  SRVERCP 12
  SSLCACHE 12
  SSLTCBS (obsolete) 7
  TCAM (obsolete) 7
system initialization table
  default 7
system programming interface
  changed SPI commands 42
  new commands 50
  new options
    PERFORM STATISTICS RECORD 50
  obsolete commands and options 42
  obsolete options
    CREATE PROGRAM 42
    CREATE TYPETERM 42
    INQUIRE CORBASERVER 42
    INQUIRE DISPATCHER 42
    INQUIRE PROGRAM 42
    SET DISPATCHER 42
    SET PROGRAM 42
  obsolete SPI command options 41
system programming interface (SPI) 41

T

TAPEJRNL 162
task-related user exits
  migration 60
TCAM networks 97
TCAM system initialization parameter 7
TCAM user exit, obsolete 55
TCAM/ACB 97
TCAM/DCB 97
TCPIPSERVICE definition
  REALM attribute 29
TCPIPSERVICE resource definition
  migration 141
threadsafe programming
  migration 133, 135
transactions
  CCRL 23
  CEMN 23

U

upgrading the CSD 86
  SCAN function 88
user-replaceable programs 65
  DFHAPXPO 65
  DFHCNV 65
  DFHISAIP 66
USERNAME option
  WEB SEND command (Client) 35
USERNAMELEN option
  WEB SEND command (Client) 36

V

VOLUME 162

W

Windows remote MAS
  removal of support 177
workload manager
  EWLM
    migration 117
WUI plex definition 155

X

X2TASK 162
XAPADMGR, new user exit 59
XCFGROUP
  CEMT INQUIRE IRC 15
XCFGROUP option
  INQUIRE IRC command 44
XCFGROUP, parameter of DFHXCOPT 63
XCTL
  migration to channels and containers 100
XDSPGBL 162
XDSPPOOL 162
XHFS system initialization parameter
  migration 141, 145
XJVMPOOL 162
XLEC transaction 154
XLSRPBUF 162
XMONITOR 162

XOPDIRECTST
  CEMT INQUIRE PIPELINE 16
  CEMT INQUIRE WEBSERVICE 18
XOPDIRECTST option
  INQUIRE WEBSERVICE command 48
XOPSUPPORTST
  CEMT INQUIRE PIPELINE 17
  CEMT INQUIRE WEBSERVICE 18
XOPSUPPORTST option
  INQUIRE WEBSERVICE command 49
XPCFTCH 149
XPCTA 149
XPLINK option
  migration 149
XPROGRAM 162
XRES system initialization parameter
  migration 141, 145
XSTREAM 162
XTASK 162
XTCTIN, obsolete user exit 55
XTCTOUT, obsolete user exit 55
XWBAUTH, new user exit 59
XWBOPEN, new user exit 60
XWBSNDO, new user exit 60

Z

z/OS conversion services 85

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.

© Copyright IBM Corp. 2001, 2007

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

The following terms are trademarks, or registered trademarks, of International Business Machines Corporation in the United States, or other countries, or both:

Trademarks and service marks

The following terms, used in this publication, are trademarks or service marks of IBM Corporation in the United States or other countries:

CICS	CICSPlex	CICS/ESA	CICS/MVS
C/370	DB2	IBM	
Language Environment		MVS	OS/390
Parallel Sysplex	RACF	SupportPac	TXSeries
VisualAge	VSE/ESA	VTAM	WebSphere
z/OS			

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft®, Windows, Windows NT®, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a trademark of The Open Group in the United States, or other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Readers' Comments — We'd Like to Hear from You

CICS Transaction Server for z/OS
Migration from CICS TS Version 2.3
Version 3 Release 2

Publication No. GC34-6857-00

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your 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. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Submit your comments using one of these channels:

- Send your comments to the address on the reverse side of this form.
- Send a fax to the following number: +44–1962–816151
- Send your comments via e-mail to: idrcf@hursley.ibm.com

If you would like a response from IBM, please fill in the following information:

Name	Address
Company or Organization	
Phone No.	E-mail address

Readers' Comments — We'd Like to Hear from You
GC34-6857-00

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

PLACE
POSTAGE
STAMP
HERE

IBM United Kingdom Limited
User Technologies Department (MP095)
Hursley Park
Winchester
Hampshire
SO21 2JN
United Kingdom

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line

Program Number: 5655-M15

GC34-6857-00

Spine information:

CICS Transaction Server for z/OS **Migration from CICS TS Version 2.3** Version 3 Release 2