CICS Transaction Server for z/OS

Release Guide

Version 3 Release 2

GC34-6811-01

CICS Transaction Server for z/OS

Release Guide

Version 3 Release 2

GC34-6811-01

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 363.

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.

Contents

Preface

What this book is about

This book provides information about new and changed function in CICS® Transaction Server for z/OS®, Version 3 Release 2. It gives an overview of the changes to reference information, and points you to the manuals where more detailed reference information is given.

The programming interface information given in this book is intended to show only what is new and changed from the previous release of CICS TS, and to highlight the benefits of the new function. For programming interface information, read the primary sources of programming interface and associated information in the following publications:

- *CICS Application Programming Reference*
- *CICS System Programming Reference*
- *CICS Customization Guide*
- *CICS External Interfaces Guide*
- *CICSPlex SM Application Programming Guide*
- *CICSPlex SM Application Programming Reference*

Who this book is for

This book is for those responsible for the following user tasks:

- Evaluation and planning
- System administration
- Programming
- Customization

What you need to know to understand this book

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

Notes on terminology

When the term "CICS" is used without any qualification in this book, it refers to the CICS element of IBM® CICS TS.

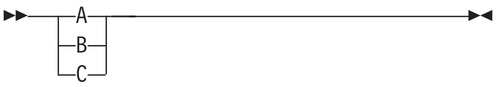
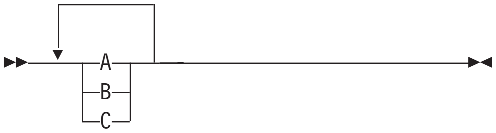
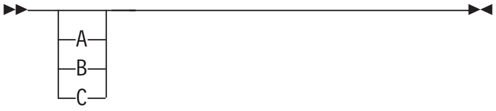
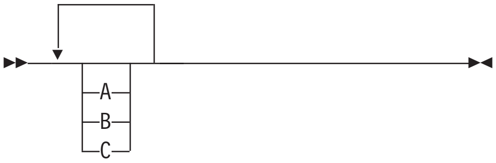
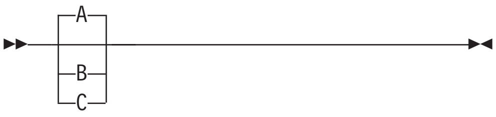
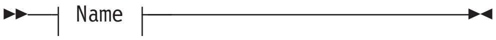
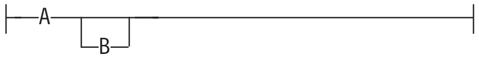
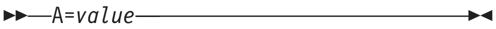
"CICSPlex SM" is used for the CICSPlex System Manager element of IBM CICS TS.

"MVS™" is used for the operating system, which is a base element of z/OS.

Syntax notation

Syntax notation specifies the permissible combinations of options or attributes that you can specify on CICS commands, resource definitions, and many other things.

The conventions used in the syntax notation are:

Notation	Explanation
	Denotes a set of required alternatives. You must specify one (and only one) of the values shown.
	Denotes a set of required alternatives. You must specify at least one of the values shown. You can specify more than one of them, in any sequence.
	Denotes a set of optional alternatives. You can specify none, or one, of the values shown.
	Denotes a set of optional alternatives. You can specify none, one, or more than one of the values shown, in any sequence.
	Denotes a set of optional alternatives. You can specify none, or one, of the values shown. **A** is the default value that is used if you do not specify anything.
 **Name:** 	A reference to a named section of syntax notation.
	**A**= denote characters that should be entered exactly as shown. *value* denotes a variable, for which you should specify an appropriate value.

Part 1. Summary of CICS Transaction Server for z/OS, Version 3 Release 2

This part contains a brief overview of the major new function in CICS Transaction Server for z/OS, Version 3 Release 2.

Chapter 1. CICS application connectivity and reuse

CICS Transaction Server for z/OS, Version 3 Release 2 delivers a set of capabilities which provide customer value by enabling business flexibility through IT simplification. These capabilities are represented in three themes:

- application connectivity
- application reuse
- service management

The capabilities represented by the *application connectivity* and *application reuse* themes enable you to: support integrated business processes, by extending existing applications beyond their original designs using standard application programming interfaces and protocols; and to create, from existing applications, business components that are flexible and configurable for use in new applications.

Support for WSDL 2.0

CICS now supports the creation and deployment of Web services using Web service descriptions that comply with the WSDL 2.0 specification.

You can now add support for WSDL 2.0 to existing Web service provider and requester applications, as well as creating new applications that can support both levels of WSDL. Alternatively, you can migrate your applications from using WSDL 1.1 to WSDL 2.0.

To help you achieve this, the following enhancements have been made:

- The Web services assistant tooling has been enhanced so that you can now create a Web service description that complies with WSDL 2.0 from a language structure, or create language structures from a WSDL 2.0 document. The batch jobs DFHWS2LS and DFHLS2WS also have new and changed parameters to provide you with more flexibility when creating your web service applications, including specifying absolute URIs for your Web service. Specifying absolute URIs means that you do not have to edit the generated Web service description.
- CICS can now support Web service request messages that have optional responses and one way Web service request messages that have an optional SOAP fault response. These interactions between requester and provider are defined as message exchange patterns (MEPs). CICS supports four of the message exchange patterns (MEPs) that are defined in the WSDL 2.0 specification.
- When CICS is acting as the Web service requester, you can now define how long CICS should wait for a response before returning to the application on an INVOKE WEBSERVICE command. The PIPELINE resource has a new attribute called RESPWAIT that determines how long CICS should wait in seconds. If you do not set a value for this attribute, either the default timeout for the transport protocol or the dispatcher timeout for the transaction is used instead.
- The Web services assistant API has been updated to include the new parameters.

CICS support for WSDL 2.0 is subject to some restrictions. See the *CICS Web Services Guide* for details.

Related concepts

Chapter 5, “Support for WSDL 2.0,” on page 31

Support for MTOM/XOP optimization of binary data

In standard SOAP messages, binary objects are base64 encoded and included in the message body. This increases their size by 33%, which for very large binary objects can significantly impact transmission time. Implementing MTOM/XOP provides a solution to this problem.

The *SOAP Message Transmission Optimization Mechanism (MTOM)* and *XML-binary Optimized Packaging (XOP)* specifications, often referred to as MTOM/XOP, define a method for optimizing the transmission of base64Binary data within SOAP messages.

- The MTOM specification defines a conceptual method for optimizing SOAP messages by sending base64Binary data in separate binary attachments using a MIME Multipart/Related message. Sending the data in binary format significantly reduces its size, thus optimizing the transmission of the SOAP message. This type of MIME message is referred to as an *MTOM message* in this information.
- The XOP specification defines an implementation for optimizing XML messages using binary attachments in a packaging format that includes but is not limited to MIME messages.

CICS implements support for these specifications in both requester and provider pipelines when the transport protocol is HTTP or HTTPS. As an alternative to including the base64Binary data directly in the SOAP message, CICS applications that are deployed as Web service providers or requesters can use this support to send and receive MTOM messages with binary attachments.

You can configure this support by using additional options in the pipeline configuration file.

Related concepts

Chapter 6, “Support for MTOM/XOP optimization of binary data,” on page 41

Support for WSDL 1.1 with SOAP 1.2

CICS support for Web services has been extended to comply with the *WSDL 1.1 Binding Extension for SOAP 1.2* specification.

This specification defines the binding extensions that are required to indicate that Web service messages are bound to the SOAP 1.2 protocol. The aim is to provide functionality that is comparable with the binding for SOAP 1.1.

CICS complies with this specification when generating Web service binding files from WSDL 1.1 documents with SOAP 1.2.

This specification is published as a formal submission request by the World Wide Web Consortium (W3C) at http://www.w3.org/Submission/wsdl11soap12/.

Related concepts

Chapter 7, “Support for WSDL 1.1 with SOAP 1.2,” on page 49

Support for Web Services Trust Language

CICS support for securing Web services has been enhanced to include an implementation of the *Web Services Trust Language* (or WS-Trust) specification.

CICS can now interoperate with a Security Token Service (STS), such as Tivoli® Federated Identity Manager, to validate and issue security tokens in Web services. This enables CICS to send and receive messages that contain a wide variety of security tokens, such as SAML assertions and Kerberos tokens, to interoperate securely with other Web services.

You can configure the CICS-supplied security handler to define how CICS should interact with an STS. The `<wsse_handler>` element in the pipeline configuration file now includes additional elements and attributes to configure this support. CICS can either validate or exchange the first security token or the first security token of a specific type in the message header.

If you want more sophisticated processing to take place, CICS provides a separate Trust client interface that you can use in a custom message handler. You can use the Trust client instead of the security handler or in addition to it.

CICS support for WS-Trust is subject to some restrictions. See How CICS complies with WS-Trust in the *CICS Web Services Guide* for details.

Related concepts

Chapter 8, “Support for Web Services Trust Language,” on page 51

IP interconnectivity

You can now make CICS TS-to-CICS TS distributed program link (DPL) calls over Transmission Control Protocol/Internet Protocol (TCP/IP) connections. Enhanced TCP/IP support for the Java™ Connector Architecture (JCA) is also provided.

In previous CICS releases, communication between CICS Transaction Server regions that were not in the same MVS image and not in the same sysplex required a communication link that implemented IBM’s Systems Network Architecture (SNA). For DPL calls, this requirement no longer applies; you can use a TCP/IP link instead. Support for DPL over TCP/IP is similar to that for DPL over SNA; for example, both two-phase commit and containers are supported.

The new functions extend CICS existing support for TCP/IP, which already includes ECI over TCP/IP and covers connections to, for example, CICS Clients, HTTP clients, and Java clients. Unlike, for example, ECI over TCP/IP, the new Java function supports two-phase commit and containers.

A new type of intercommunication link

In previous CICS releases, two main types of communication link could be used to connect CICS to other (CICS or non-CICS) systems:

- Multiregion operation (MRO) connections
- Intersystem communication (ISC) over SNA connections

CICS TS for z/OS, Version 3.2 introduces a third type of intercommunication link: *IP interconnectivity connections*. IPIC connections can be used between two CICS TS for z/OS, Version 3.2 regions. The regions may or may not be in the same z/OS sysplex. Currently, distributed program link (DPL) is the only type of base CICS intercommunication function supported.

Related concepts

Chapter 9, “IP interconnectivity,” on page 57

TCP/IP management and control

TCP/IP management and control allows you to monitor work that enters or leaves CICS over Transmission Control Protocol/Internet Protocol (TCP/IP) connections. It provides, for TCP/IP networks, a subset of the management functions already provided for Advanced Program-to-Program Communication (APPC, also known as "LUTYPE6.2") networks; plus some additional functions that are not available for APPC or multiregion operation (MRO) networks.

Note: By "TCP/IP network" we mean systems that are interconnected by:

- IPIC connections (IPCONN) . Currently, these can be used only between CICS TS 3.2 regions, and between a CICS TS 3.2 region and a Java client.
- TCP/IP connections from clients that carry, for example, Web Interface, IIOP, or SOAP over HTTP requests inbound to CICS.

TCP/IP management and control enables you, for example, to:

- Use CICSPlex SM, or an equivalent tool, to:
  - Get a CICSplex-wide view of the TCP/IP network.
  - Examine, in real time:
    - The TCP/IP network resources that a particular CICS region is using
    - The work passing in and out of a particular CICS region over the TCP/IP network
    - The CICS resources and tasks associated with a distributed transaction that flows across the CICSplex over the TCP/IP network
    - The CICS region in which a distributed transaction originated
- Save the data collected by CICS so that it can be examined off line, at some point after the tasks and resources that it relates to are no longer available.

Reasons why you might want to use TCP/IP management and control include:

- To diagnose connectivity problems
- To investigate other problems, such as transaction delays
- To track work across the CICSplex
- To capture system data over time, for use in capacity planning
- To monitor the CICSplex

Related concepts

Chapter 10, "TCP/IP management and control," on page 71

New application programming capabilities for CICS Web support

The capabilities of the application programming interface, and your architecture options for CICS Web support, are extended in several ways. For example, you can now use WEB API commands in converter programs and the DFHWBEP Web error program, and the WEB API commands support the use of containers and channels.

- New programming samples using the WEB API commands are provided to help you construct your own Web-aware application programs for both CICS as an HTTP server and CICS as an HTTP client. The new samples, provided in Assembler, C, and COBOL, demonstrate HTTP chunking and pipelining functions.
- The existing DFH$WB1A and DFH$WB1C samples, used to verify CICS Web support operation, are updated to use the WEB API commands, and a sample URIMAP definition is provided for use with DFH$WB1C.

- The WEB API commands can now be used in converter programs and in the user-replaceable Web error program DFHWBEP. Instead of constructing the responses from these programs manually in a block of storage, you can, if you wish, migrate to using WEB API commands to construct and send the response. You must however specify ACTION(IMMEDIATE) in your command, as the default of ACTION(EVENTUAL) is not permitted with DFHWBEP. Using the error program enables you to take advantage of all the available CICS Web support features, including assistance with structuring responses and with HTTP protocol compliance, and enhanced code page conversion capabilities, with minimal disruption to your existing CICS Web support architecture. The input and output parameter lists remain unchanged for these programs, and the information provided by these facilities can be used in combination with the WEB API commands.
- The Web error program DFHWBEP is now called if an error occurs during the delivery of static responses. You can use the program to customize the error responses that CICS sends to the Web client. CICS messages are also produced when the errors occur.
- Containers and channels can now be used on the WEB API commands, for both CICS as an HTTP server and CICS as an HTTP client.
  - The WEB SEND (Client and Server), WEB RECEIVE (Client and Server) and WEB CONVERSE commnads now provide the facility to send and receive the body of an HTTP message using a combination of containers and buffers.
  - Containers are also used to store HTTP headers, so you do not need to define a prefix for temporary storage queues for CICS Web support in the TCPIPSERVICE definition.
- On the PUT CONTAINER (CHANNEL) and GET CONTAINER (CHANNEL) commands, if you prefer, you can now specify a supported IANA-registered charset name for code pages for data conversion, instead of using the numeric Coded Character Set Identifiers (CCSIDs). You can also specify that data held in a container is retrieved without conversion, and return the CCSID of the unconverted data.
- CICS documents and document templates can now be converted to and from the UTF-8 and UTF-16 character encodings.
- The WEB API commands for examining form fields in a request (WEB READ FORMFIELD and WEB STARTBROWSE FORMFIELD) now provide support for data received in UTF-8 and UTF-16 formats.
- The WEB READ FORMFIELD, WEB STARTBROWSE FORMFIELD and DOCUMENT RETRIEVE commands now use the CHARACTERSET option to allow you to specify a code page for your client and application program.
- The WEB READ FORMFIELD and WEB STARTBROWSE FORMFIELD commands now respect CHARACTERSET and HOSTCODEPAGE for GET requests as well as POST requests.
- To improve performance, CICS now sends an HTTP request with the OPTIONS method to identify the version of an HTTP server only when this is required, rather than each time you open an HTTP client connection to a server.

Related concepts

Chapter 11, “New application programming capabilities for CICS Web support,” on page 77

Security changes for CICS Web support

For CICS as an HTTP server, basic authentication, client certificate authentication and resource level security are now available for CICS documents or z/OS UNIX® files delivered as a static response (using a URIMAP definition with the TEMPLATENAME or HFSFILE option). You can now apply access controls for these items, based on a client's user ID, without needing to use an application program to handle the requests. The resource level security capability can also be used to provide a more granular level of security for CICS document templates used by an application program for Web delivery as part of an application-generated response. In addition, the realm supplied for basic authentication can now be customized.

Related concepts

Chapter 12, "Security changes for CICS Web support," on page 91

Improved management of CICS documents and document templates

CICS now allows you to release storage by deleting documents when they are no longer required in a transaction. New caching facilities improve the performance of applications that use CICS document templates.

- CICS now allows you to delete documents that are no longer required during a transaction.
- CICS now caches a copy of most types of document template. When applications reference the template, they use the cached copy, improving performance.
- You can use the SET DOCTEMPLATE NEWCOPY command to refresh a document template. For cached document templates, the command refreshes the cached copy of the document template. For document templates produced from programs and exit programs, the command phases in a new copy of the program.
- CICS now collects statistics for document templates with the new CICS statistics type DOCTEMPLATE. The statistics show the number of times each document template is referenced, and the number of times a cached copy was made, refreshed, used and deleted.

Related concepts

Chapter 13, "Improved management of CICS documents and document templates," on page 107

Optimized support for data conversion

The CICS internal CCSID conversion interface, used in container support and the Web interface, now provides optimized support when all characters in the input data are within a set that can be converted directly using a single-byte to single-byte translate operation. When the full conversion process uses the z/OS unicode conversion service, this optimization can result in a significant reduction in CPU time for conversion processing. For example, tests have shown that when SOAP messages contain data only within the ASCII single-byte subset of UTF-8, the optimized conversion processing between UTF-8 and EBCDIC shows a reduction of over 15% in transaction CPU time for processing a 32 KB input message, and over 30% for generating a 32 KB output message.

The first time a conversion between a specific pair of Coded Character Set Identifiers (CCSIDs) is attempted, CICS builds a test table that indicates which single-byte input codes can be translated directly, and a translate table giving the single-byte output codes for the valid single-byte input codes. At the same time, CICS also builds the corresponding pair of tables for the reverse conversion. On each conversion call for that pair of CCSIDs, CICS uses the appropriate test table (via the translate and test instruction) to check whether the input data can be translated with a simple translate. If so, CICS copies the input data to the output buffer and translates it rather than performing the full conversion process.

The test table is also used to optimize the logic to determine the length required for the converted data, for example, when processing the GET CONTAINER command with the NODATA option, where the specified or assumed CCSID differs from the CCSID in which the container was stored. If the test shows that all input characters are within the set eligible for single-byte translation, the output length is assumed to be the same as the input length, avoiding the need to perform a test conversion to determine the actual length.

Related concepts

Chapter 14, “Optimized support for data conversion,” on page 113

Chapter 2. CICS service management: configuration enhancements and constraint relief

CICS Transaction Server for z/OS, Version 3 Release 2 delivers a set of capabilities which provide customer value by enabling business flexibility through IT simplification. These capabilities are represented in three themes:

- application connectivity
- application reuse
- service management

The capabilities represented by the *service management* theme enable you to effectively manage large runtime configurations using modern user interfaces, so that you can meet demanding service level and IT governance objectives. CICS Transaction Server for z/OS, Version 3 Release 2 provides functions that simplify product configuration and management, and that relieve several architectural constraints.

Dynamic program library management

CICS TS 3.2 introduces dynamic program LIBRARY resources providing the ability to enable the data sets from which program artifacts will be loaded to be defined dynamically without it being necessary to restart the CICS region. This is in addition to the existing means of defining the data sets statically in the DFHRPL concatenation.

CICS TS 3.2 introduces a new resource type of LIBRARY, representing a partitioned data set (PDS/PDSE) or sequence of concatenated partitioned data sets (PDS/PDSEs), containing program entities that make up an application or group of applications. DFHRPL is a special example of a LIBRARY that cannot be altered in a running CICS system.

Traditionally, data sets containing program artifacts have been defined to CICS in DFHRPL in the startup JCL. To change any of the data sets it is necessary to edit the JCL and restart CICS. There is often a need to change the data sets used to load programs to do the following:

- Apply emergency fixes to a program while CICS is running.

  If the fixed program is put in a data set earlier in the concatenation, it can be loaded in place of the broken program. This can be done today, but you typically have to include a special data set for fixes in the DFHRPL concatenation, and move the fixed program into this data set. It gets difficult to know what is in the special fix data set.
- Add new programs or new versions of programs while CICS is running.

  You are more likely to do this currently in development or test systems than in production, as you might have restrictions on changes allowed in production, however, enabling this in production will provide a benefit towards continuous operations.

Having to restart CICS is frustrating, and impacts the continuous availability of the CICS system. This new feature allows for better organization and management of applications within a CICS system while maintaining continuous availability.

For further information on using dynamic program LIBRARY resources, see the *CICS Application Programming Guide*.

Related concepts

Chapter 15, “Dynamic program library management,” on page 117

Support for the z/OS Enterprise Workload Manager

Currently, CICS supports the z/OS Workload Manager (WLM). In CICS TS for z/OS, Version 3.2, support is added for the z/OS Enterprise Workload Manager (EWLM).

EWLM’s key feature is that it makes possible end-to-end workload monitoring in distributed environments that contain multiple, interacting, server products.

Related concepts

Chapter 16, “Support for the z/OS Enterprise Workload Manager,” on page 123

Additional statistics for MVS Workload Manager

The CICS monitoring domain statistics, and the monitoring section in the System Status report produced by the sample statistics program DFH0STAT, now include MVS workload manager goal information for the CICS address space.

The statistics show:

- The performance goal type for the CICS address space.
- The goal value (for a velocity goal only).
- The importance level of the performance goal.
- Whether or not the CICS address space is designated as CPU critical or as storage critical.

Related concepts

Chapter 17, “Additional statistics for MVS Workload Manager,” on page 133

Threadsafety for PLT-enabled global user exit programs

It is now possible to define global user exit programs that are enabled by first-phase program list table (PLT) programs as threadsafe (in previous CICS releases, this technique was available to task-related user exit programs but not to global user exit programs).

Typically, global user exit programs are enabled during CICS initialization, by ENABLE commands issued by PLT programs. To ensure that the exit programs are available as early as possible during CICS startup, global user exit programs such as those that run at the recovery exits are typically enabled during the first phase of PLT processing.

Because first-phase PLT programs run so early in CICS initialization, no resource definitions are available. This means that you cannot use installed PROGRAM definitions (or the program autoinstall user program) to define first-phase PLT programs to CICS, nor to define the user exit programs that first-phase PLT programs enable. Instead, default definitions are installed automatically by CICS. Whether or not program autoinstall is specified as active on the PGAIPGM system initialization parameter, the autoinstall user program is not invoked to allow the definitions to be modified.

This type of autoinstall by CICS is known as *system autoinstall*.

It is recommended that you write your global user exit programs to be threadsafe. However, the system-autoinstalled program definition specifies CONCURRENCY(Quasirent); that is, the exit programs are defined as quasi-reentrant. 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.

Related concepts

Chapter 18, "Threadsafety for PLT-enabled global user exit programs," on page 135

Storage management above the 2GB boundary

CICS now provides 64-bit storage, allowing you to use storage above the 2GB boundary (above the bar). This capability removes all size restrictions from inter-program data transfer, and provides a means of exploiting z/OS 64-bit capabilities.

Related concepts

Chapter 19, "Storage management above the 2GB boundary," on page 137

Shared data tables larger than 2 GB

Shared data tables are no longer restricted to 2 gigabytes (GB) of data per CICS region.

The data component of a shared data table may now be spread across more than one data space. Furthermore, the table entry and index components of the table are now stored in separate data spaces, rather than in the CICS address space. This allows the total control information for all tables to have a combined size of up to 4 GB (2 GB of table entry descriptors and 2 GB of index nodes).

When shared data tables support is activated, three data spaces are initially created: DFHDT001 (for table entry descriptors), DFHDT002 (for index nodes), and DFHDT003 (for record data storage). Each one is initially allocated 16 MB of virtual storage. Additional data space storage is allocated automatically as required for each data space. If a data space for record data storage reaches the maximum size of 2 GB, a new data space is allocated dynamically. This continues until the table entry descriptor or index node data space is full, the maximum supported number of data spaces (currently 100) are in use, or any data space limit specified by the IEFUSI MVS installation exit has been reached.

The list of data spaces associated with a CICS region is included in the output from the MVS system command `D J,jobname`, so you can use this command to check whether any additional data spaces have been allocated beyond the initial three.

This enhancement should be almost entirely transparent to existing users of shared data tables. The total virtual and real storage requirement for existing data tables has not significantly changed, because most of the additional control information to support larger tables is integrated into existing data areas without needing to expand them.

Related concepts

Chapter 20, "Shared data tables larger than 2 GB," on page 143

Extended addressing for entry sequenced data sets (ESDS)

In previous releases, any entry sequenced data sets (ESDS) used by CICS were restricted in size to 4 gigabytes (GB). This was because CICS programs used 32-bit numbers to address individual records. In CICS TS for z/OS, Version 3.2, CICS programs can use 64-bit numbers to address records, which removes the 4GB limit on the size of the data set.

An entry sequenced data set is a Virtual Storage Access Method (VSAM) dataset that behaves rather like a sequential data set. As new records are added to an ESDS they are appended to the end of the data set. It is not possible to insert a new record between two existing records. Nor is it possible to delete a record after it has been created.

A common way of using an ESDS is to write a number of records to the data set using the WRITE command and to read all of the records back again by a browse. (This is the typical way in which you would use a sequential data set.) To read records back, you use a STARTBR command to position the cursor at the beginning (or end) of the data set, and follow this with a READNEXT (or READPREV) command to read all records within the ESDS.

Records in an ESDS can be either fixed length or variable length.

In the original ESDS design, as each record is added to an ESDS it is assigned a *relative byte address* (RBA), which is an unsigned 32-bit number. The RBA is the number of bytes from the beginning of the ESDS at which the record is located. The use of RBAs implies that an ESDS may not contain more than 4 gigabytes of data.

VSAM now supports *extended ESDS* that use 64-bit RBAs. CICS now supports 64-bit RBAs and extended ESDS.

Related concepts

Chapter 21, “Extended addressing for entry sequenced data sets (ESDS),” on page 145

Greater precision and capacity for monitoring clocks

The CICS monitoring clocks for performance class data now measure dispatch (elapsed) time and CPU time for CICS TCBs in single microseconds, rather than in units of 16 microseconds. The clock capacity, which was around 19 hours, is now only bounded by the capacity of the local store clock, which is several years.

Related concepts

Chapter 22, “Greater precision and capacity for monitoring clocks,” on page 149

Data compression for monitoring records

CICS can now perform data compression on the SMF 110 monitoring records output by the CICS monitoring facility. Data compression can provide a significant reduction in the volume of data written to SMF. The records are compressed and expanded using standard z/OS services.

Related concepts

Chapter 23, “Data compression for monitoring records,” on page 155

Monitoring facility transaction CEMN

The new CEMN monitoring facility transaction provides an operator interface which you can use to check on the options currently in effect for the CICS monitoring facility, and to change some of the settings without needing to restart CICS.

Related concepts

Chapter 24, "Monitoring facility transaction CEMN," on page 161

WebSphere MQ monitoring and statistics changes

The CICS-MQ adapter, CICS-MQ bridge and CICS-MQ trigger monitor (previously part of WebSphere® MQ), are now shipped with CICS TS, introducing changes to monitoring and statistics.

A new CICS statistics type MQCONN and corresponding new CICSPlex SM view set MQCONN - WebSphere MQ Connection provide global statistics information on the connection between a CICS region and WebSphere MQ.

Two new DFHDATA monitoring fields, WMQREQCT and WMQGETWT, provide information on WebSphere MQ requests and wait times.

Related concepts

Chapter 25, "WebSphere MQ monitoring and statistics changes," on page 163

Additional storage information for MVS TCBs

The INQUIRE MVSTCB command now provides additional information about the MVS storage key of each storage element allocated to each TCB, and the amount of storage actually in use within each storage element. The CICS global and resource statistics for MVS TCBs now report the storage actually in use, as well as the storage allocated to TCBs.

The amount of storage in use, as displayed by the new command option and statistics, is the amount of storage actually GETMAINed by the task. Previously, the statistics only displayed the storage allocated to the TCBs, which is always allocated in page multiples (4096 bytes). The new information gives a more accurate picture of the storage actually being used.

Related concepts

Chapter 26, "Additional storage information for MVS TCBs," on page 165

XCF group limit relief

The effective limit of 2047 CICS regions that a single sysplex can support has been lifted.

Multiregion operation (MRO) enables CICS regions that are running in the same MVS image to communicate without the need for an access method such as ACF/VTAM or TCP/IP. When used with the MVS cross-system coupling facility (XCF), MRO allows CICS regions in different MVS images, but within the same sysplex, to communicate without an SNA access method.

Currently, all the CICS regions in a sysplex that use XCF/MRO must join the same XCF group, DFHIR000; and an XCF group is limited to 2047 members. Effectively, this imposes a limit on the number of CICS regions that a sysplex can support.

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 new function also brings organizational benefits: it would be possible, for example, to place production regions into a different XCF group from development and test regions.

Related concepts

Chapter 27, “XCF group limit relief,” on page 167

Configuration and problem determination improvements for Java programs

There are a number of new features to help you configure JVMs more simply, identify problems with JVMs more quickly, and obtain detailed problem diagnosis information more easily.

- CICS validates the Java and CICS home directories specified in a JVM profile when you attempt to start the JVM, to check that the directories exist, that CICS has **read** permissions for them, and that the install check file is present.
- CICS issues warning or error messages at runtime if you include a deprecated option in a JVM profile.
- CICS builds a base library path and a base class path for you using the supplied directories for CICS and Java files, so these directories do not now need to be specified explicitly in JVM profiles. There are new options that you can use to specify additional items on the class paths, which indicate the correct location for the classes more clearly.
- A new symbol &JVM_NUM; is available in JVM profiles to insert a unique JVM number in the names of output and dump files produced by the JVM. The unique JVM number is also added to file names produced by the **-generate** option.
- The sample JVM profiles and documentation include more guidance about specifying Java dump options.
- The options specified for JVMs are traced when the JVMs are started.
- You can specify any JVM option in a JVM profile, prefixed with a hyphen, and it is passed through to the JVM. You are no longer restricted to the subset of options previously recognized by CICS in JVM profiles.
- The messages that CICS produces relating to JVMs now provide more diagnostic information and advice.
- CICS now formats the output from the JVM trace facility, adding the description of each trace point from the TraceFormat.dat file and placing the inserts, so that you do not have to interpret the data manually.
- The USECOUNT option on the INQUIRE PROGRAM command now displays a use count for Java programs.

Related concepts

Chapter 28, “Configuration and problem determination improvements for Java programs,” on page 175

Improved scheduling for garbage collection in JVMs

Instead of performing garbage collection in a JVM after a specified number of Java program executions, CICS now schedules garbage collection when a specified percentage of the storage in the active part of the nonsystem heap is used. The garbage collection is carried out as a separate transaction, so it does not affect the statistics for user transactions.

Related concepts

Chapter 29, “Improved scheduling for garbage collection in JVMs,” on page 189

JVM startup, termination and timeout control

You can now control startup and timeout for JVMs (Java Virtual Machines). You can start JVMs manually using the PERFORM JVMPOOL command, in addition to those started by CICS. You can also change the timeout threshold in the JVM profile for your JVMs, so that idle JVMs do not have to become eligible for termination after 30 minutes of inactivity as at present, but can continue to exist for a specified length of time up to 7 days, or never time out. These functions give you greater control over the availability of your JVMs to meet peak demand. The JVM termination facility has also been refined. When you make changes to a JVM profile or a shared Java class, you can now implement these by terminating only the appropriate subset of JVMs in the pool, rather than by terminating the whole JVM pool.

Related concepts

Chapter 30, “JVM startup, termination and timeout control,” on page 193

Chapter 3. CICS service management: CICSPlex SM improvements

CICS Transaction Server for z/OS, Version 3 Release 2 delivers a set of capabilities which provide customer value by enabling business flexibility through IT simplification. These capabilities are represented in three themes:

- application connectivity
- application reuse
- service management

The capabilities represented by the *service management* theme enable you to effectively manage large runtime configurations using modern user interfaces, so that you can meet demanding service level and IT governance objectives. CICS Transaction Server for z/OS, Version 3 Release 2 provides enhancements to CICSPlex SM functions that support these goals.

Integrated installation of CICSPlex SM

The installation of CICSPlex SM is integrated with the installation of CICS.

The following improvements reduce the complexity of installing and configuring CICSPlex SM:

- You can now edit the DFHISTAR job to modify the CICS and CICSPlex SM installation parameters for your environment. You no longer need to edit, separately, a EYUISTAR job. You modify and submit one set of input parameters in the DFHISTAR job.. DFHISTAR produces customized JCL for CICS and CICSPlex SM.
- You can create dynamically, during initialization and when a CICSPlex SM system is started by a transaction, CICS resource definitions for CICSPlex SM objects. You no longer need to manipulate the CSD files (using DFHCSDUP) to create the resource definitions necessary for the CMAS, MAS, and WUI.
- You can now run the EYU9XDUT utility to create the definitions required to start a WUI and its CICSplex. You would previously have had to use the end-user interface or a batch utility to create such definitions.

Related concepts

Chapter 31, “Integrated installation of CICSPlex SM,” on page 201

EYU9XDBT CICSPlex SM definition utility

EYU9XDBT is a new CICSPlex SM utility that provides an easy-to-use command interface for performing CMAS and CICSplex definition activities.

The EYU9XDBT CICSPlex SM definition utility uses the CICSPlex SM API to enable you to specify the required CICSplex names in some simple parameters, and the utility sets up the definitions for you. Unlike the BATCHREP utility, you do not need to manually edit an input file.

You can use this utility to perform all CMAS and CICSplex definition activities once the basic CMAS environment has been established. Such activities include:

- 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
- Creating CMAS to CMAS link definitions.
- Importing, printing or exporting CICSPlex SM objects defined to CMAS or CICSplex contexts.

It is limited to data repositories at the same release level as CICSPlex SM. EYU9XDBT is used during installation to set up your initial CICSPlex SM environment.

The following samples are provided:

EYUJXBT0
Contains annotated EYU9XDBT JCL syntax for use as a quick reference.

EYUJXBT1
Contains sample JCL for invoking EYU9XDBT and defining a CICSplex, a CICS system group and a CICS system definition.

EYUJXBT2
Contains sample JCL for invoking EYU9XBTP and creating a CMAS to CMAS link definition.

Related concepts

Chapter 32, “EYU9XDBT CICSPlex SM definition utility,” on page 213

Expanding on a summary view record count

The CICSPlex SM Web User Interface (WUI) has been improved to enable you to expand a summary view to display the details of summarized records. You can now click on a record count field to open a new tabular view displaying those records that relate to the selected summary row showing the state of the system at the time the initial summary occurred.

The expanded tabular view shows the ordinary filters that have been defined for the view, plus the filter that has been used to expand the summary view. For example, in the case of a **Local or dynamic transactions** (LOCTRAN) view summarized on the **Number of times transaction used** column, the use count filter and value would appear on the expanded view in addition to any previously applied filters.

The expanded view is a normal, filtered, tabular view. You can perform any further actions on it that you would normally be allowed to on a tabular view including additional summarizations. When you click the back button on the expanded view, you are returned to the summarized view from which the expanded view was launched.

This function is supported by the new API command EXPAND. This takes the summarized result set created using the GROUP command and creates a new result set containing one record for each of the records summarized by GROUP in an individual summary record. This allows you to perform further actions on the result set including using additional GROUP or FETCH commands.

Related concepts

Chapter 33, “Expanding on a summary view record count,” on page 215

Improved help for the CICSPlex SM Web User Interface

The CICSPlex SM Web User Interface (WUI) has been improved with the introduction of detailed help for all IBM-supplied and user-defined views and menus.

Related concepts

Chapter 34, "Improved help for the CICSPlex SM Web User Interface," on page 217

Support for the map function in the CICSPlex SM Web User Interface

The CICSPlex SM Web User Interface has been improved by the addition of a map function equivalent to the CICSPlex SM end user interface MAP command in previous releases of CICS TS.

The associations between CICS resource definitions defined to CICSPlex SM can be complex and difficult to visualize. For example, a CICS system can be associated with a specification and a specification might contain one or more groups. In turn there can be definitions within the groups. This type of structure is often portrayed as the branches of the tree and the WUI map function provides a method of generating a visual representation of this tree structure for a selected resource. This representation, called a map, can portray business application services (BAS), resource monitoring (MON), real-time analysis (RTA), or workload management (WLM) definitions. Maps allow you to verify that the relationships between your definitions are what you expect.

Related concepts

Chapter 35, "Support for the map function in the CICSPlex SM Web User Interface," on page 219

Extended CICSPlex SM support for TDQs and CMASs

The CICSPlex SM Web User Interface (WUI) has been improved by the addition of new WUI views providing more information about transient data queues, and help with the management of CMASs. Additionally, API support has been extended to the CPLXCMAS resource table.

There are two new WUI view sets:

Topology data for transient data queues

This is a single view providing tabular information about all intrapartition, extrapartition, and indirect transient data queue (TDQ) resources within the specified context and scope. It identifies the name and types of transient data queues and contains links to the appropriate type-specific TDQ view. It is associated with the CRESTDQ resource.

- To open this view from the WUI main menu, click **CICS operations views** → **Transaction data queue operations views** → **Topology data for transient data queues** .

CMAS in CICSplex definitions

This view set lists CICSplexes and the CMASs associated with them. By setting your context to a specific CMAS you can see all the CMASs that manage the CICSplexes for which the context CMAS is the maintenance point. It is associated with the CPLXCMAS resource. You can use the **Unassign** action to remove CMASs from the management of the CICSplex.

- To open this view from the WUI main menu, click **Administration views → CMAS configuration administration views → CMAS in CICSplex definitions**.
- Click on a record in the **CMAS** column to open the associated detailed view of the selected CMAS.

The addition of CICSPlex SM API support for the CPLXCMAS resource table enables you to write applications that use the API command UNASSIGN to remove a CMAS from a CICSplex management role.

Related concepts

Chapter 36, “Extended CICSPlex SM support for TDQs and CMASs,” on page 225

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 this release. 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.

The following EYUPARMS have been removed:

- xxxCONMSG
- xxxTDQMSG

The following messages have been added to support the NLS-enablement of CICSPlex SM messages:

- EYUBM0329I to EYUBM0348I
- EYUBN0013W to EYUBN0017W
- EYUXL0030I to EYUXL0032I

The following messages have been removed:

- EYUBM0322I to EYUBM0327I
- EYUBN0012W
- EYUXL0020I

Related concepts

Chapter 37, “National language support for CICSPlex SM messages,” on page 227

Improved CICSPlex SM history function

The CICSPlex SM MAS history function has been improved so that it is now possible to retrieve additional performance class monitoring data for the resource managers used by your transactions, by specifying RMI=YES in your MCT, and application naming data by specifying APPLNAME=YES in your MCT. Also, it is now possible to use the CICSPlex SM Web User Interface supplied EYUSTARTHTASK tabular and detailed views to retrieve historical task data from the historical data store.

Related concepts

Chapter 38, “Improved CICSPlex SM history function,” on page 229

Other changes to CICSPlex SM

A number of changes have been made to the CICSPlex SM Web User Interface (WUI) to make it more functional and enhance its usability and serviceability. Several new resource tables and view sets have also been added.

Sorting and summarizing on CICS system name

New icons have been added to appropriate CICSPlex SM Web User Interface tabular views to enable sorting and summarizing of the CICS system name column.

Terminology improvements

The terminology used in WUI views and menus has been simplified in order to improve consistency and reduce the length of some titles and phrases. The use of shorter titles has led to a reduction in the width of some columns enabling views to display more data. No new terms have been introduced.

New resource tables and WUI view sets

A number of new task-related resource tables and associated WUI view sets have been added to CICSPlex SM.

To access these views from the WUI main menu, click **CICS operations views** → **Task operations views**

Table 1. New resource tables and view sets

Resource table	WUI view set	Description
TASKESTG	Task element storage EYUSTARTTASKESTG	Information about CICS storage elements for tasks.
TASKFILE	File usage by task EYUSTARTTASKFILE	Information about tasks and the CICS files they have used.
TASKRMI	RMI usage by an individual task EYUSTARTTASKRMI	Information about the use tasks have made of the CICS Resource Manager Interface (RMI).
TASKTSQ	TS queue usage by task EYUSTARTTASKTSQ	Information about tasks and their associated CICS temporary storage queues.

Using the WUI to control CMAS and MAS tracing

You can use the Web User Interface (WUI) to control the tracing that occurs in an active CMAS and MAS. Two new views are provided: the **MASs known to CICSplex** trace view and the **CMAS detail** trace view. The trace flags are displayed as strings of bits in the range 1 through 32, separated by commas. You change the trace flag settings by editing the display.

Changes to the WUI data repository import function

The WUI server repository import function has been improved to make it easier to update WUI views and menus as a result of program temporary fixes (PTFs) from IBM service teams. It is now possible to import menus and view sets singly or in groups. In previous releases the whole set of supplied views and menus needed to be regenerated in order to implement any change. The import function now allows you to update the supplied views and menus without having to shut down the WUI server.

The new function, uses the import panel of the CICS COVC transaction. As well as allowing you to import a transient data queue containing a complete set of views and menus, you can now import data set members containing view sets or menus. To facilitate this there are two new fields on the panel; **Input Data set Name**, and the **Input Data set member name**. You can choose either to specify a TDQ name to import a complete set of supplied or customized view and menu definitions as in previous releases, or a data set and member name to import specific views or menus. You can use the asterisk as a trailing wildcard character on the data set member name field to specify a group of views or menus. You cannot import both a TDQ and a data set at the same time using COVC.

The supplied set of WUI view and menu definitions is currently located in the SEYUVIEW data set. The composition of this data set has changed to facilitate the new function. Formerly SEYUVIEW contained three members, one each for the English, Japanese and simplified Chinese versions of the definitions. Now the data set includes one member for each view set and menu in each of the three supplied languages. These data set members are named **EYU**`ltccc`, where:

- `l` specifies the language; **E** for English, **S** for simplified Chinese and **K** for Japanese.
- `t` identifies a set of views. The current supplied WUI views and menus are all identified by the letter **A**.
- `nnn` identifies the resource with which the views are associated.

As an alternative to the COVC transaction, you can configure a WUI server to import automatically menus and view sets from a specified data set or data set and member at startup. To facilitate this there are two new optional WUI server initialization parameters:

AUTOIMPORTDSN()
Identifies the name of a data set to import.

AUTOIMPORTMEM()
Identifies the name of a data set member to import. You can use an asterisk as a trailing wildcard character to specify a group of views or menus.

You can also use the AUTOIMPORTTDQ parameter to automatically import a specified TDQ when you start a WUI server.

The following WUI server messages are introduced in support of the new import function:
EYUVS0929E
EYUVS0930W
EYUVS0931E
EYUVS0113W
EYUVS0114E
EYUVS1050E
EYUVS1051E
EYUVS1052E
EYUVS1053E
EYUVS1054E

EYUVS1055E

Change to the WUI data repository export function

The COVC export function is used to export WUI definitions so that you can back up or distribute definitions to other WUI servers, or migrate definitions to other releases. WUI data repository definitions consist of view sets, menus, map objects, user objects and user group profiles. In previous releases you could specify only one resource type in an export operation. It is now possible to export all data repository definition types at the same time. This makes it easier to export your definitions to a single TDQ.

To facilitate this the COVC Export panel has been changed to allow a value of `All` in the **Type** field. You can use the All in conjunction with the **Name** field that identifies specific or generic name of the objects to be exported. This field utilizes an asterisk as a trailing wildcard character. Specifying `All` with an asterisk in the **Name** field results in all the definitions being exported from the repository. If you use `All` with, for example `TEST*` , COVC will export all of definitions that have a name starting with TEST, whatever their type.

Change to the codepage conversion table (DFHCNV)

The default codepage conversion table (DFHCNV) has been changed so that CICSPlex SM codepages are included automatically. That is, it is no longer necessary to include a copy statement for EYU$CNV1 in the DFHCNV source.

Related concepts

Chapter 39, “Other changes to CICSPlex SM,” on page 231

Chapter 4. Discontinued functions

Some functions which were supported in CICS Transaction Server for z/OS, Version 3 Release 1 have been discontinued, or reduced in scope, inCICS Transaction Server for z/OS, Version 3 Release 2.

Removal of the CICSPlex SM TSO end user interface

With the new enhancements to the CICSPlex SM Web User Interface (WUI) functionality and provision of the EYU9XDBT definition utility, 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.

It has not been possible to use the EUI to manage the more modern features of CICS since the EUI was stabilized at the CICS Transaction Server for z/OS, Version 2 Release 2 level of functionality. Its removal in this release:

- Improves and streamlines the installation of CICSPlex SM.
- Makes migration scenarios more straightforward.
- Reduces the complexity of system configuration by reducing the number of address spaces that have to be managed.

The EUI is replaced by the WUI, which is:

- Customizable to your business needs.
- Easier to learn and use.
- Accessible to authorized users from any location that can launch a web browser.
- Fully accessible to those with restricted vision or mobility.
- National language support (NLS) enabled.

Related concepts

Chapter 40, "Removal of the CICSPlex SM TSO end user interface," on page 237

Removal of resettable mode for JVMs (Java virtual machines)

Continuous JVMs, which are not reset between each use, generally perform better than resettable JVMs and are more consistent with other versions of Java. Resettable JVMs are no longer supported, and migration to continuous JVMs is required in this CICS release. The CICS JVM Application Isolation Utility, a code checking and reporting utility, is provided to help identify areas where you should check the behavior of Java programs that were designed to run in resettable JVMs, before migrating them to run in continuous JVMs. Configuration and tuning for continuous JVMs is simpler than it was for resettable JVMs, and some unnecessary JVM profile options are now deprecated.

Continuous JVMs have several advantages over resettable JVMs:

- They have a lower CPU cost per transaction, because they do not require a reset between each use.
- They are simpler to set up and tune, primarily because they have fewer different storage heaps than resettable JVMs.

- Certain constraints placed on programs in resettable JVMs do not apply for continuous JVMs, enabling developers to maximize the performance of their applications.
- They are compatible with future versions of Java, and are more like the standard JVMs used by other products.

Related concepts

Chapter 41, "Removal of resettable mode for JVMs (Java virtual machines)," on page 239

Removal of DFH$MOLS support for data for earlier CICS releases

The CICS Transaction Server for z/OS, Version 3 Release 2 release of DFH$MOLS does not process monitoring data for releases earlier than CICS Transaction Server for OS/390®, Version 1 Release 3. 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.

Related concepts

Chapter 42, "Removal of DFH$MOLS support for data for earlier CICS releases," on page 247

Removal of the DFHLSCU utility

The log stream sizing utility DFHLSCU has been removed from CICS.

Note: The utility is still available as SupportPac™ CD14 to assist in the migration of CICS MVS/ESA™ regions to CICS Transaction Server. The CICS SupportPacs are available from the following IBM Web site:

http://www-1.ibm.com/support/docview.wss?rs=1083&uid=swg27007241

Related concepts

Chapter 43, "Removal of the DFHLSCU utility," on page 249

Part 2. CICS application connectivity and reuse

CICS Transaction Server for z/OS, Version 3 Release 2 delivers a set of capabilities which provide customer value by enabling business flexibility through IT simplification. These capabilities are represented in three themes:

- application connectivity
- application reuse
- service management

The capabilities represented by the *application connectivity* and *application reuse* themes enable you to: support integrated business processes, by extending existing applications beyond their original designs using standard application programming interfaces and protocols; and to create, from existing applications, business components that are flexible and configurable for use in new applications.

Chapter 5. Support for WSDL 2.0

CICS now supports the creation and deployment of Web services using Web service descriptions that comply with the WSDL 2.0 specification.

You can now add support for WSDL 2.0 to existing Web service provider and requester applications, as well as creating new applications that can support both levels of WSDL. Alternatively, you can migrate your applications from using WSDL 1.1 to WSDL 2.0.

To help you achieve this, the following enhancements have been made:

- The Web services assistant tooling has been enhanced so that you can now create a Web service description that complies with WSDL 2.0 from a language structure, or create language structures from a WSDL 2.0 document. The batch jobs DFHWS2LS and DFHLS2WS also have new and changed parameters to provide you with more flexibility when creating your web service applications, including specifying absolute URIs for your Web service. Specifying absolute URIs means that you do not have to edit the generated Web service description.
- CICS can now support Web service request messages that have optional responses and one way Web service request messages that have an optional SOAP fault response. These interactions between requester and provider are defined as message exchange patterns (MEPs). CICS supports four of the message exchange patterns (MEPs) that are defined in the WSDL 2.0 specification.
- When CICS is acting as the Web service requester, you can now define how long CICS should wait for a response before returning to the application on an INVOKE WEBSERVICE command. The PIPELINE resource has a new attribute called RESPWAIT that determines how long CICS should wait in seconds. If you do not set a value for this attribute, either the default timeout for the transport protocol or the dispatcher timeout for the transaction is used instead.
- The Web services assistant API has been updated to include the new parameters.

CICS support for WSDL 2.0 is subject to some restrictions. See the *CICS Web Services Guide* for details.

Support for WSDL 2.0 in the Web services assistant

The Web services assistant batch jobs DFHWS2LS and DFHLS2WS have new and changed parameters to support the creation of WSDL 2.0 documents from language structures, and language structures from WSDL 2.0 documents. These parameters also reflect additional options and flexibility that have been introduced by the WSDL 2.0 specification.

Changes to DFHLS2WS

When you create a new Web service from a language structure, you can now decide which version of WSDL to use when generating the Web service description and the version of SOAP protocol that should be used in the binding. You also have the option of creating a Web service that complies with both 1.1 and 2.0 versions of the WSDL specification and can use either SOAP protocol as a binding.

For existing Web service applications that are deployed in CICS, you can use DFHLS2WS to extend support to include WSDL 2.0 by creating and deploying an additional WSDL 2.0 document and binding file.

The following new parameters have been added to DFHLS2WS:

SOAPVER=**1.1**|**1.2**|**ALL**

Specifies which SOAP level to use in the generated Web service description. This parameter is only available when the **MINIMUM-RUNTIME-LEVEL** is set to `2`.

1.1 The SOAP 1.1 protocol should be used as the binding for the Web service description.

1.2 The SOAP 1.2 protocol should be used as the binding for the Web service description.

ALL Both the SOAP 1.1 or 1.2 protocol can be used as the binding for the Web service description.

If you do not specify a value for this parameter, the default value depends on the version of WSDL that you want to create. If you only require WSDL 1.1, then the SOAP 1.1 binding is used. If you only require WSDL 2.0, then the SOAP 1.2 binding is used. If you require both WSDL 1.1 and WSDL 2.0, then both SOAP 1.1 and 1.2 bindings are used for each Web service description.

WSDL_1.1=*value*

The fully qualified z/OS UNIX name of the file into which the Web service description is written. The Web service description conforms to the WSDL 1.1 specification. DFHLS2WS creates the file (but not the directory structure) if it does not already exist. The file extension is .wsdl. This parameter produces the same result as the **WSDL** parameter, so you can only specify one or the other.

WSDL_2.0=*value*

The fully qualified z/OS UNIX name of the file into which the Web service description is written. The Web service description conforms to the WSDL 2.0 specification. DFHLS2WS creates the file (but not the directory structure) if it does not already exist. The file extension is .wsdl. This parameter can be used in conjunction with the **WSDL** or **WSDL_1.1** parameters. It is only available when the **MINIMUM-RUNTIME-LEVEL** is set to `2`.

You can also now specify an absolute URI instead of a relative URI for a Web service. The URI is used in both the creation of the URIMAP resource when a PIPELINE SCAN command is issued, and as the `soap:address` element in the generated WSDL document. This means that you do not have to edit the generated WSDL to add a server name and port. Note that the URIMAP resource continues to specify a host name of `*` and does not specify a port number, even when you use an absolute URI.

The following parameter has changed in DFHLS2WS:

URI=*value*

This parameter specifies the relative or absolute URI that a client will to use to access the Web service. CICS uses the value specified when it generates a URIMAP resource from the Web service binding file created by DFHLS2WS. The parameter specifies the path component of the URI to which the URIMAP definition applies.

Changes to DFHWS2LS

DFHWS2LS now automatically determines the WSDL version of Web service description that has been supplied as input. The batch job has been enhanced to provide you with more flexibility in how to handle the Web service description.

- `wsdl:Bindings` elements can be associated with multiple `wsdl:Service` elements in Web service descriptions. A new parameter has been added to enable you to select a specific Service element within the Web service description.
- When you are creating a service requester application, you can now specify a subset of `wsdl:Operation` elements that you want to invoke and create a Web service binding file based on that subset. This can be useful when you have a large WSDL file with many Operations. By only using a subset of Operation elements, you can save on storage by generating a smaller Web service binding file.

The following new parameters have been added to DFHWS2LS:

OPERATIONS=*value*
For Web service requester applications, specifies a subset of valid `wsdl:Operation` elements from the Web service description that should be used to generate the Web service binding file. Each Operation element should be separated by a space; the list can span more than one line if necessary. This parameter can be used for both WSDL 1.1 and WSDL 2.0 documents.

WSDL-SERVICE=*value*
Specifies the `wsdl:Service` element that should be used when the Web service description contains more than one Service element for a Binding element. If you specify a value for the **BINDING** parameter, then the Service element that you specify for this parameter must be consistent with the specified Binding element. You can use this parameter with either WSDL 1.1 or WSDL 2.0 documents.

WSDL and message exchange patterns

A WSDL 2.0 document contains a message exchange pattern (MEP) that defines the way that SOAP 1.2 messages should be exchanged between the Web service requester and Web service provider.

CICS supports four out of the eight message exchange patterns that are defined in the WSDL 2.0 Part 2: Adjuncts specification for both service provider and service requester applications. These are:

In-Only
A request message is sent to the Web service provider, but the provider is not allowed to send any type of response to the Web service requester.

In-Out
A request message is sent to the Web service provider, and a response message is returned to the Web service requester. The response message could be a normal SOAP message or a SOAP fault.

In-Optional-Out
A request message is sent to the Web service provider, and a response message is optionally returned to the Web service requester. If there is a response, it could be either a normal SOAP message or a SOAP fault.

Robust In-Only
A request message is sent to the Web service provider, and no response

message is returned to the Web service requester unless an error occurs. If there is an error, a SOAP fault message is sent to the requester.

New containers for WSDL 2.0 support

New context containers have been provided in the Web service provider and requester pipelines to support WSDL 2.0.

Web Service Description Language 2.0

Web Services Description Language (WSDL) is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information.

WSDL 2.0 provides a model as well as an XML format for describing Web services. It enables you to separate the description of the abstract functionality offered by a service from the concrete details of a service description, such as ″how″ and ″where″ that functionality is offered. It also describes extensions for Message Exchange Patterns, SOAP modules, and a language for describing such concrete details for SOAP 1.2 and HTTP. The WSDL 2.0 specification also resolves many technical issues and limitations that are present in WSDL 1.1.

The latest specification for WSDL 2.0 is published as a W3C candidate recommendation at http://www.w3.org/TR/wsdl20.

Changes to CICS externals

The following changes have been made to the CICS externals to support the Web services improvements.

Changes to resource definition

PIPELINE resource

The PIPELINE resource has a 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.

Changes to the system programming interface

INQUIRE WEBSERVICE command

The INQUIRE WEBSERVICE command has 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.

INQUIRE PIPELINE command

The INQUIRE PIPELINE command has new options:

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.

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.

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.

SET PIPELINE command

The SET PIPELINE command has new options:

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.

CREATE PIPELINE command

The CREATE PIPELINE command has new options:

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.

Changes to CEMT

INQUIRE PIPELINE command

The INQUIRE PIPELINE command has new options:

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.

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.

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.

INQUIRE WEBSERVICE command

The INQUIRE WEBSERVICE command has 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.

SET PIPELINE command

The SET PIPELINE command has a new 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.

Changes to the CICSPlex SM programming interface

Changes to resource tables

The PIPELINE resource table has new fields for the SOAP level that is supported by the pipeline, the operating mode of the pipeline, and the time out that applies to requester mode pipelines.

The PIPEDEF resource table has a new field for the time out that applies to requester mode pipelines.

The WEBSERV resource table has new fields for the Coded Character Set ID (CCSID) for the Web service, the mapping level that was used to generate the Web service binding file, and the minimum runtime level that is required to run the Web service in CICS.

Changes to CICSPlex SM views and menus

Pipeline definition view

The following attribute has been added to the PIPEDEF (EYUSTARTPIPEDEF.CREATE) view:

Field	Attribute name	Description
Response wait time for Requester Pipelines	RESPWAIT	Specifies a time control, in seconds, on the wait time for an application program to wait for an optional reponse message from a remote web service. The value can range from 0 to 9999 seconds, or will have the standard null value of -1 if RESPWAIT(DEFT) is specified on the PIPELINE definition. If RESPWAIT(DEFT) was specified for this attribute, 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. Note that the value of this attribute **may not** be reset to -1 (DEFT) - only 0 to 9999 may be applied. If you need to reset the RESPWAIT value to -1, you will have to delete the current PIPELINE object, and INSTALL another instance of it, where the RESPWAIT value specifies DEFT.

Pipeline detail view

The following attributes have been added to the PIPELINE (EYUSTARTPIPELINE.DETAILED) view:

Field	Attribute name	Description
Pipeline operation mode	PIPEMODE	The mode that that pipeline is operating in.

Field	Attribute name	Description
Response wait time for Requester Pipelines	RESPWAIT	Specifies a time control, in seconds, on the wait time for an application program to wait for an optional reponse message from a remote web service. The value can range from 0 to 9999 seconds, or will have the standard null value of -1 if RESPWAIT(DEFT) is specified on the PIPELINE definition. If RESPWAIT(DEFT) was specified for this attribute, 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. Note that the value of this attribute **may not** be reset to -1 (DEFT) - only 0 to 9999 may be applied. If you need to reset the RESPWAIT value to -1, you will have to delete the current PIPELINE object, and INSTALL another instance of it, where the RESPWAIT value specifies DEFT.
SOAP level supported by the pipeline	SOAPLEVEL	Specifies the version of SOAP that is supported in the pipeline. Values can be blank, 1.1 or 1.2.

Web service detail view

The following attributes have been added to the WEBSERV (EYUSTARTWEBSERV.DETAILED) view:

Field	Attribute name	Description
Coded character set ID	CCSID	The name of the CCSID that is used to encode data between the application and the Web service binding file at run time.
Mapping level used in WSBind file	MAPPINGLEVEL	The level of mapping that is used to convert data between language structures and the Web services description (WSDL) document. Values are 1.0, 1.1, 1.2 and 2.0.
Minimum runtime level required by WSBind file	MINRUNLEVEL	The minimum runtime level that is required to run the Web service in CICS. Values are: MINIMUM, 1.0, 1.1, 1.2, 2.0 or CURRENT.

Changes to statistics

Pipeline statistics now indicate the operating mode of a pipeline; that is, whether the pipeline is used for Web service provider or requester applications.

Changes to problem determination

Additional diagnostics are available to help you solve problems that relate to Web services.

Deployment diagnostics

The Web services assistant batch jobs provide additional information and error messages to help you resolve errors with parameter values or a conflict with the combination of parameters that you have selected. These messages are recorded in the log file for the batch jobs.

The installation process for the WEBSERVICE resource now produces additional messages that indicate when there is a conflict between the definitions for the Web service and the pipeline that is associated with the WEBSERVICE resource.

Runtime diagnostics

The diagnostics that have been added to help you diagnose problems when a Web service is deployed include:

- New pipeline domain messages (DFHPI*xxxx*).
- New pipeline domain trace points.
- Web service validation of SOAP messages against WSDL 2.0 documents.
- New RESP2 values for the INVOKE WEBSERVICE command.

Chapter 6. Support for MTOM/XOP optimization of binary data

In standard SOAP messages, binary objects are base64 encoded and included in the message body. This increases their size by 33%, which for very large binary objects can significantly impact transmission time. Implementing MTOM/XOP provides a solution to this problem.

The *SOAP Message Transmission Optimization Mechanism (MTOM)* and *XML-binary Optimized Packaging (XOP)* specifications, often referred to as MTOM/XOP, define a method for optimizing the transmission of base64Binary data within SOAP messages.

- The MTOM specification defines a conceptual method for optimizing SOAP messages by sending base64Binary data in separate binary attachments using a MIME Multipart/Related message. Sending the data in binary format significantly reduces its size, thus optimizing the transmission of the SOAP message. This type of MIME message is referred to as an *MTOM message* in this information.
- The XOP specification defines an implementation for optimizing XML messages using binary attachments in a packaging format that includes but is not limited to MIME messages.

CICS implements support for these specifications in both requester and provider pipelines when the transport protocol is HTTP or HTTPS. As an alternative to including the base64Binary data directly in the SOAP message, CICS applications that are deployed as Web service providers or requesters can use this support to send and receive MTOM messages with binary attachments.

You can configure this support by using additional options in the pipeline configuration file.

MTOM messages and binary attachments in CICS

CICS supports and controls the handling of MTOM messages in both Web service provider and requester pipelines using an MTOM handler program and XOP processing.

You enable and configure the MTOM handler and XOP processing using options that are defined in the pipeline configuration file. When enabled, the MTOM handler accepts and unpackages inbound MTOM messages containing XOP documents and binary attachments, and outbound MTOM messages are packaged and sent. If the MTOM handler is not enabled in the pipeline and CICS receives an MTOM message, it is rejected with a SOAP fault.

You can configure a provider pipeline to:

- Accept MTOM messages, but never send MTOM response messages.
- Accept MTOM messages and send the same type of response message.
- Accept MTOM messages, but only send MTOM messages when there are binary attachments present.
- Accept MTOM messages and always send MTOM response messages.
- Process XOP documents and binary attachments in direct or compatibility mode.

You can configure a requester pipeline to:

- Never send an MTOM message, but accept MTOM response messages.

- Only send MTOM messages when there are binary attachments, and accept MTOM response messages.
- Always send MTOM messages and accept MTOM response messages.
- Process XOP documents and binary attachments in direct or compatibility mode.

Modes of support

There are certain scenarios where CICS cannot support the XOP document format in messages directly. For example, the Web Services Security support and Web service validation cannot parse the `<xop:Include>` elements in the XOP document. Therefore, two modes of support are provided in the pipeline to handle XOP documents and any associated binary attachments.

direct mode

In direct mode, the binary attachments associated with an inbound or outbound MTOM message are passed in containers through the pipeline and handled directly by the application, without the need to perform any data conversion.

compatibility mode

Compatibility mode is used when the pipeline processing requires the message to be in standard XML format, with any binary data stored as base64Binary fields within the message. For inbound messages, the XOP document and binary attachments are reconstituted into a standard XML message, either at the beginning of the pipeline when Web Services Security is enabled, or at the end of the pipeline when Web service validation is enabled. For outbound messages, a standard XML message is created and passed along the pipeline. It is converted to XOP format by the MTOM handler just before CICS sends it.

Compatibility mode is much less efficient than direct mode because binary data gets converted to base64 format and back again. However, it does allow your Web services to interoperate with other MTOM/XOP Web service requesters and providers without needing to change your applications.

Changes to pipeline configuration elements

New elements

`<cics_mtom_handler>`

Enables the CICS-supplied MTOM handler program, that provides support for MTOM MIME multipart/related messages that contain XOP documents and binary attachments. MTOM support is enabled for all inbound messages that are received in the pipeline, but MTOM support for outbound messages is conditionally enabled subject to further options.

For details, see “The <cics_mtom_handler>element” on page 342.

`<dfhmtom_configuration>`

Specifies configuration information for the CICS-supplied MTOM handler program, which provides support for MIME messages that contain XOP documents and binary attachments. If you do not specify any configuration for MTOM, CICS assumes default values.

For details, see “The <dfhmtom_configuration>element” on page 342.

`<mtom_options>`

Specifies when to use MTOM for outbound SOAP messages.

For details, see “The <mtom_options>element” on page 344.

`<xop_options>`

Specifies whether XOP processing can take place in direct or compatibility mode.

For details, see “The <xop_options>element” on page 345.

`<mime_options>`

Specifies the domain name that should be used when generating MIME content-ID values, that are used to identify binary attachments.

For details, see “The <mime_options>element” on page 343.

New containers for MTOM/XOP support

New context containers have been provided in the Web service provider and requester pipelines to handle inbound and outbound messages in MTOM format.

Changes to CICS externals

Changes to the system programming interface

INQUIRE PIPELINE command

The INQUIRE PIPELINE command has 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.

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.

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.

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 WEBSERVICE command

The INQUIRE WEBSERVICE command has new options:

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.

Changes to CEMT

INQUIRE PIPELINE command

The INQUIRE PIPELINE command has new options:

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

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.

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.

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.

The INQUIRE WEBSERVICE command

The INQUIRE WEBSERVICE command has new options:

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.

Changes to the CICSPlex SM programming interface

Changes to resource tables

The PIPELINE resource table has new fields for:

- The MTOM status of the pipeline.
- If and when outbound MTOM messages should be sent.
- The MTOM utilization when binary attachments are not present.
- The pipeline application handler support for handling XOP documents.
- The current mode of the pipeline.

- The domain name that is used to generate MIME content-ID values to identify binary attachments.

The WEBSERV resource table has new fields for the capability of the Web service to support XOP documents, and whether the Web service is currently able to support XOP documents.

Changes to CICSPlex SM views and menus

Pipeline detail view

The following attributes have been added to the PIPELINE (EYUSTARTPIPELINE.DETAILED) view:

Field	Attribute name	Description
Outbound SOAP message MTOM status	SENDMTOMST	Returns a value that indicates when MTOM should be used for outbound SOAP messages. The values are: YES : Always use MTOM for outbound SOAP messages. NO : Do not use MTOM for outbound SOAP messages. SAME : Use MTOM for outbound SOAP message responses when the inbound message is received in MTOM format.
Pipeline application handler XOP capability	XOPSUPPORTST	Returns a value that indicates whether the application handler for the pipeline supports the processing of XOP documents and binary attachments.
Pipeline direct mode XOP status	XOPDIRECTST	Returns a value that indicates whether the pipeline can currently handle XOP documents in direct mode.
SOAP MTOM status	MTOMST	Returns a value that indicates whether support for MTOM has been enabled in the pipeline. The values are: SUPPORTED : MTOM support has been enabled in the pipeline. NOTSUPPORT : MTOM support has not been enabled in the pipeline.
Use MTOM even when no XOP attachments are present	MTOMNOXOPST	Returns a value that indicates whether MTOM should be used for outbound SOAP messages when there are no binary attachments present. The values are: YES : Use MTOM, even when there are no binary attachments present. NO : Do not use MTOM unless there are binary attachments present.

Web service detail view

The following attributes have been added to the WEBSERV (EYUSTARTWEBSERV.DETAILED) view:

Field	Attribute name	Description
Current Web service direct mode XOP status	XOPDIRECTST	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.
Web service XOP capability	XOPSUPPORTST	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.

Changes to problem determination

Additional diagnostics are available to help you solve problems that relate to supporting MTOM/XOP in the pipeline.

Runtime diagnostics

The diagnostics that have been added to help you diagnose problems when using MTOM/XOP include:

- New pipeline domain messages (DFHPI*xxxx*).
- New pipeline domain trace points, ranging from PI 1300 up to PI 1506.

Chapter 7. Support for WSDL 1.1 with SOAP 1.2

CICS support for Web services has been extended to comply with the *WSDL 1.1 Binding Extension for SOAP 1.2* specification.

This specification defines the binding extensions that are required to indicate that Web service messages are bound to the SOAP 1.2 protocol. The aim is to provide functionality that is comparable with the binding for SOAP 1.1.

CICS complies with this specification when generating Web service binding files from WSDL 1.1 documents with SOAP 1.2.

This specification is published as a formal submission request by the World Wide Web Consortium (W3C) at http://www.w3.org/Submission/wsdl11soap12/.

Chapter 8. Support for Web Services Trust Language

CICS support for securing Web services has been enhanced to include an implementation of the *Web Services Trust Language* (or WS-Trust) specification.

CICS can now interoperate with a Security Token Service (STS), such as Tivoli Federated Identity Manager, to validate and issue security tokens in Web services. This enables CICS to send and receive messages that contain a wide variety of security tokens, such as SAML assertions and Kerberos tokens, to interoperate securely with other Web services.

You can configure the CICS-supplied security handler to define how CICS should interact with an STS. The `<wsse_handler>` element in the pipeline configuration file now includes additional elements and attributes to configure this support. CICS can either validate or exchange the first security token or the first security token of a specific type in the message header.

If you want more sophisticated processing to take place, CICS provides a separate Trust client interface that you can use in a custom message handler. You can use the Trust client instead of the security handler or in addition to it.

CICS support for WS-Trust is subject to some restrictions. See How CICS complies with WS-Trust in the *CICS Web Services Guide* for details.

Enhancements to the security handler

The security handler supports WS-Trust by interoperating with a Security Token Service that is specified in the pipeline configuration file.

The CICS security handler uses the information in the pipeline configuration file to send a Web service request to the Security Token Service (STS). The type of request that is sent depends on the action that you want the STS to perform.

In a service provider pipeline

In a service provider pipeline, the security handler supports two types of actions. You can configure the security handler to:

- Send a request to the STS to validate the first instance of a security token, or the first security token of a specific type, in the WS-Security header of the inbound message.
- Send a request to the STS to exchange the first instance of a security token, or the first security token of a specific type, in the WS-Security header of the inbound message, for a security token that CICS can understand.

The security handler dynamically creates a pipeline to send the Web service request to the STS. This pipeline exists until a response is received from the STS, after which it is deleted. If the request is successful, the STS returns an identity token or the status of the token's validity. The security handler places the token in the DFHWS-USERID container.

If the STS encounters an error, it returns a SOAP fault to the security handler. The security handler then passes a fault back to the Web service requester.

In a service requester pipeline

In a service requester pipeline, the security handler can only request to exchange a token with the STS. The pipeline configuration file defines what type of token the STS should issue to the security handler.

If the request is successful, the token is placed in DFHWS-USERID and then included in the outbound message header. If the STS encounters an error, it returns a SOAP fault to the security handler. The security handler then passes the fault back through the pipeline to the Web service requester application.

The security handler is only capable of requesting one type of action from the STS for the pipeline. It can also only exchange one type of token for an outbound request message, and is limited to handling the first token in the WS-Security message header, either the first instance or of a specific type. These options cover the most common scenarios for using an STS, but might not offer you the processing that you require for handling inbound and outbound messages.

If you want to provide more specific processing to handle many tokens in the inbound message headers or exchange multiple types of tokens for outbound messages, it is recommended that you use the Trust client interface. Using this interface, you can create a custom message handler to send your own Web service request to the STS.

The Trust client interface

The Trust client interface enables you to interact with a Security Token Service (STS) directly, rather than using the security handler. This means that you have the flexibility to provide more advanced processing of tokens than the security handler.

The Trust client interface is an enhancement to the CICS-supplied program DFHPIRT. This program is normally used to start a pipeline when a Web service requester application has not been deployed using the CICS Web services assistant. Its functionality has now been extended so that it can act as the Trust client interface to the STS.

You can invoke the Trust client interface by linking to DFHPIRT from a message handler or header processing program, passing a channel called DFHWSTC-V1 and a set of security containers. Using these containers, you have the flexibility to request either a validate or issue action from the STS, select what token type to exchange and pass the appropriate token from the message header. DFHPIRT dynamically creates a pipeline, composes a Web service request from the security containers, and sends it to the STS.

DFHPIRT waits for the response from the STS and passes this back in the DFHWS-RESTOKEN container to the message handler. If the STS encounters an error, it returns a SOAP fault. DFHPIRT puts the fault in the DFHWS-STSFAULT container and returns to the linking program in the pipeline.

You can use the Trust client interface without enabling the security handler in your service provider and service requester pipelines, or you can use the Trust client interface in addition to the security handler.

Changes to pipeline configuration elements

New elements

`<auth_token_type>`

Specifies what type of identity token is required.

For details, see “The <auth_token_type>element” on page 341.

`<sts_authentication>`

Specifies that a Security Token Service (STS) should be used for authentication and determines what type of request is sent.

For details, see “The <sts_authentication>element” on page 346.

`<sts_endpoint>`

Specifies the location of the Security Token Service (STS).

For details, see “The <sts_endpoint>element” on page 348.

The <authentication>element

You can now specify `trust="blind"` in your service provider and service requester pipeline configuration files. If you specify this attribute value, CICS does not verify the identity token that is present in the SOAP message header of inbound or outbound messages.

The combination of attribute values is summarized in the tables below.

Table 2. The **mode** *and* **trust** *attributes in a service requester pipeline*

trust	mode	Meaning
none	none	No credentials are added to the message
	basic	*Invalid combination of attribute values*
	signature	Asserted identity is not used. CICS uses a single X.509 security token which is added to the message, and used to sign the message body. The certificate is identified with the `<certificate_label>` element, and the algorithm is specified in the `<algorithm>` element.
blind	none	*Invalid combination of attribute values*
	basic	Asserted identity is not used. CICS adds an identity token to the message, but does not provide a trust token. The identity token is a username with no password. The user ID placed in the identity token is the contents of the DFHWS-USERID container (which, by default, contains the running task’s user ID).
	signature	*Invalid combination of attribute values*
basic	(any)	*Invalid combination of attribute values*

Table 2. The **mode** *and* **trust** *attributes in a service requester pipeline (continued)*

trust	mode	Meaning
signature	none	*Invalid combination of attribute values*
	basic	Asserted identity is used. CICS adds the following tokens to the message: • The trust token is an X.509 security token. • The identity token is a username with no password. The certificate used to sign the identity token and message body is specified by the `<certificate_label>`. The user ID placed in the identity token is the contents of the DFHWS-USERID container (which, by default, contains the running task's user ID).
	signature	*Invalid combination of attribute values*

Table 3. The **mode** *and* **trust** *attributes in a service provider pipeline*

trust	mode	Meaning
none	none	Inbound messages need not contain any credentials, and CICS does not attempt to extract or verify any credentials that are found in a message. However, CICS will check that any signed elements have been correctly signed.
	basic	Inbound messages must contain a username security token with a password. CICS puts the username in the DFHWS-USERID container.
	signature	Inbound messages must contain an X.509 security token that has been used to sign the message body.
blind	none	*Invalid combination of attribute values*
	basic	Inbound messages must contain an identity token, where the identity token contains a user ID and optionally a password. CICS puts the user ID in the DFHWS-USERID container. If no password is included, CICS uses the user ID without verifying it. If a password is included, the security handler DFHWSSE1 verifies it.
	signature	Inbound messages must contain an identity token, where the identity token is the first X.509 certificate in the SOAP message header. The certificate does not need to have signed the message. The security handler extracts the matching user ID and places it in the DFHWS-USERID container.

Table 3. The **mode** *and* **trust** *attributes in a service provider pipeline (continued)*

trust	mode	Meaning
basic	none	*Invalid combination of attribute values*
	basic	Inbound messages must use asserted identity: • The trust token is a username token with a password • The identity token is a second username token without a password. CICS puts this username in container DFHWS-USERID.
	signature	Inbound messages must use asserted identity: • The trust token is a username token with a password • The identity token is an X.509 certificate. CICS puts the user ID associated with the certificate in container DFHWS-USERID.
signature	none	*Invalid combination of attribute values*
	basic	Inbound messages must use asserted identity: • The trust token is an X.509 certificate • The identity token is a username token without a password. CICS puts the username in container DFHWS-USERID. The identity token and the body must be signed with the X.509 certificate.
	signature	Inbound messages must use asserted identity: • The trust token is an X.509 certificate • The identity token is a second X.509 certificate. CICS puts the user ID associated with this certificate in container DFHWS-USERID. The identity token and the body must be signed with the first X.509 certificate (the trust token).

New containers for the Trust client interface

The Trust client interface uses new containers on the channel DFHWSTC-V1 to send and receive security tokens from a Security Token Service (STS).

Changes to CICS externals

The following changes have been made to the CICS externals to support WS-Trust.

Changes to problem determination

Additional diagnostics are available to help you solve problems that relate to WS-Trust.

Deployment diagnostics

The installation process for the PIPELINE resource now produces additional messages that indicate when there is a conflict in the definitions of the security handler DFHWSSE1.

Runtime diagnostics

Additional runtime diagnostics have been provided to help you diagnose problems when CICS is using a Security Token Service to exchange security tokens. These include:

- New pipeline domain messages (DFHPI*xxxx*).
- New pipeline domain trace points.

Chapter 9. IP interconnectivity

You can now make CICS TS-to-CICS TS distributed program link (DPL) calls over Transmission Control Protocol/Internet Protocol (TCP/IP) connections. Enhanced TCP/IP support for the Java Connector Architecture (JCA) is also provided.

In previous CICS releases, communication between CICS Transaction Server regions that were not in the same MVS image and not in the same sysplex required a communication link that implemented IBM's Systems Network Architecture (SNA). For DPL calls, this requirement no longer applies; you can use a TCP/IP link instead. Support for DPL over TCP/IP is similar to that for DPL over SNA; for example, both two-phase commit and containers are supported.

The new functions extend CICS existing support for TCP/IP, which already includes ECI over TCP/IP and covers connections to, for example, CICS Clients, HTTP clients, and Java clients. Unlike, for example, ECI over TCP/IP, the new Java function supports two-phase commit and containers.

A new type of intercommunication link

In previous CICS releases, two main types of communication link could be used to connect CICS to other (CICS or non-CICS) systems:

- Multiregion operation (MRO) connections
- Intersystem communication (ISC) over SNA connections

CICS TS for z/OS, Version 3.2 introduces a third type of intercommunication link: *IP interconnectivity connections*. IPIC connections can be used between two CICS TS for z/OS, Version 3.2 regions. The regions may or may not be in the same z/OS sysplex. Currently, distributed program link (DPL) is the only type of base CICS intercommunication function supported.

Terminology

New and changed terminology used by IP interconnectivity.

Enterprise Workload Manager (EWLM)
IBM's implementation of the **Application Response Measurement** (ARM) standard. EWLM extends the capabilities of the **z/OS Workload Manager**, which operates on z/OS, to all members of the IBM eServer™ family. Its key feature is that it makes possible end-to-end workload monitoring in distributed environments that contain multiple, interacting, server products.

intercommunication
Communication between the local CICS region and a remote system, which may or may not be another CICS region. In CICS, the term embraces **intersystem communication (ISC)** and **multiregion operation (MRO)**.

intersystem communication (ISC)
A CICS facility that provides inbound and outbound support for communication with other computer systems. ISC can be used to connect CICS to both CICS and non-CICS systems, both inside and outside the local sysplex. Contrast with **multiregion operation (MRO)**, and see also **intersystem communication over SNA** and **IP interconnectivity**.

IP interconnectivity

TCP/IP connections that can be used between CICS regions. The CICS TS-CICS TS connections support distributed program link (DPL). Two-phase commit and containers are supported.

multiregion operation (MRO)

Communication between CICS regions in the same z/OS image or z/OS sysplex without the use of SNA or TCP/IP network facilities. This allows several CICS regions to communicate with each other, and to share resources such as files, terminals, temporary storage, and so on. The systems must be in the same operating system (z/OS image); or, if the XCF access method is used, in the same z/OS sysplex. Contrast with **intersystem communication (ISC)** .

Changes to CICS externals

Changes to system initialization parameters

The following system initialization parameters have changed:

APPLID={DBDCCICS|applid}

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.

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.

TCPIP

IPIC requires “YES” to be specified on the TCPIP system initialization parameter.

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.

Changes to resource definition

New resources

IPCONN

An IPCONN (also known as an IPIC connection) is a CICS resource that represents an outbound Transport Control Protocol/Internet Protocol (TCP/IP) communication link to a remote system.

For details, see “IPCONN resource definitions” on page 281.

TCPIPSERVICE resource

The TCPIPSERVICE resource has new attribute values:

- The PROTOCOL attribute has a new value:

  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*).

- The URM attribute has a new value:

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

- For IPIC, the program name specified in the URM attribute specifies the name of the autoinstall user program for IPCONN. If you do not specify this attribute, CICS uses the CICS-supplied default IPCONN autoinstall user program, DFHISAIP.

Changes to the system programming interface

Note: Changes to the EXTRACT STATISTICS and PERFORM STATISTICS command are described in “Changes to statistics” on page 65

New SPI commands

CREATE IPCONN

Define and install an IPCONN in the local CICS region.

For details of the command, see “CREATE IPCONN” on page 291.

DISCARD IPCONN

Remove an IPCONN definition.

For details of the command, see “DISCARD IPCONN” on page 295.

INQUIRE IPCONN

Retrieve information about an IPCONN.

For details of the command, see “INQUIRE IPCONN” on page 304.

SET IPCONN

Change the attributes of an IPCONN or cancel outstanding AIDs.

For details of the command, see “SET IPCONN” on page 317.

CREATE TCPIPSERVICE command

The CREATE TCPIPSERVICE command has a new 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*).

INQUIRE TCPIPSERVICE command

The ATTACHSEC option returns a new value:

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.)

The PROTOCOL option returns a new value:

IPIC IP interconnectivity (IPIC).

Changes to CEMT

Note: Changes to the PERFORM STATISTICS command are described in “Changes to statistics” on page 65

New CEMT commands

INQUIRE IPCONN

Retrieve information about IPCONNs.

For details of the command, see “CEMT INQUIRE IPCONN” on page 325.

SET IPCONN

Change the attributes of an IPCONN or cancel outstanding AIDs.

For details of the command, see “CEMT SET IPCONN” on page 332.

DISCARD command

The DISCARD command has a new option:

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.

INQUIRE TCPIPSERVICE

The ATTACHSEC option returns a new value:

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.)

The PROTOCOL option returns a new value:

IPic
IP interconnectivity.

Changes to CICSPlex SM views and menus

CICSPlex SM views

The CICSPlex SM Web User Interface has been modified to manage and report on the new objects described in "New CICSPlex SM resource tables."

The following view sets have been added:

IPIC connections (EYUSTARTIPCONN)
The **IPIC connections** views display information about the state of a currently-installed IPCONNS (known as IPIC connections) in a TCP/IP network.

To access from the WUI main menu, click:

CICS operations views > Connection operations views > IPIC connections, or

CICS operations views > TCP/IP service operations views > IPIC connections

IPIC connection definitions (EYUSTARTIPCONDEF)
The **IPIC connection definitions** views display information about IPCONN resource definitions that describe remote systems that a CICS system communicates with using IP interconnectivity (IPIC) connections.

To access from the WUI main menu, click:

Administration views > Basic CICS resource administration views> Resource definitions > IPIC connection definitions, or

Administration views > Fully functional Business Application Services (BAS) administration views > Resource definitions > IPIC connection definitions, or

New CICSPlex SM resource tables

The following resource tables are new:

CRESIPCN
A CICSPlex SM manager resource that describes the topology data for an IPIC connection.

ERMCIPCN
A CICSPlex SM notification resource that describes an IPIC connection resource.

IPCINGRP
A CICSPlex SM Business Application Services resource that describes the membership of an IPIC connection definition (IPCONDEF) in a resource group (RESGROUP).

IPCONDEF
A CICSPlex SM Business Application Services resource that describes the attributes of an IPCONN definition. It allows IPIC connections to be defined and stored in the CICSPlex SM data repository.

IPCONN
A CICS resource that, in a TCP/IP network, describes the state of a currently-installed IPIC connection (IPCONN).

All CICSPlex SM resource tables are described in the *CICSPlex System Manager Resource Tables Reference*.

Changed CICSPlex SM resource tables

There are changes to the following base resource tables:

TCPDEF (TCPIPSERVICE definition)
A new value, IPIC, has been added to the possible values of the PROTOCOL field:

IPIC IP interconnectivity (IPIC) protocol.

TCPIPS (TCP/IP service)
A new value, IPIC, has been added to the possible values of the PROTOCOL field:

IPIC IP interconnectivity (IPIC) protocol.

RESGROUP (Resource group)
The new resource-type, IPCONDEF, can be specified on the RESTYPE option of the INSTALL command.

RASGNDEF (Resource assignment definitions)
The new resource-type, IPCONDEF can be specified as a resource type (RDEFTYPE)

RESDESC (Resource description definitions)
The following fields have been added for the resource group for IP connection definitions (IPCDEFRG):

IPCDEFTS
Target scope for IP connection definitions.

IPCDEFRS
Related scope for IP connection definitions.

Changes to global user exits

New global user exit XISQUE

Global user exit XISQUE enables you to manage intersystem queues on IP connections.

For more information about XISQUE, see the *CICS Customization Guide*.

Changes to the install and discard exit XRSINDI

Parameter **UEPIDTYP** has new values:

UEIDIPCO
An IPIC connection

Changes to user-replaceable programs

New autoinstall user-replaceable program

A new user-replaceable program is introduced, whose purpose is to manage the autoinstall of IPIC connections ("IPCONN.")

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

Autoinstall templates for IPCONNs

Unlike autoinstall for other resources, autoinstall for IPCONNs does not require model definitions, although they are recommended. However, unlike the model definitions used to autoinstall terminals, those used to autoinstall IPCONNs do not need to be defined explicitly as models. Instead, CICS can use any previously-installed IPCONN definition as a "template" for a new definition.

The purpose of a template is to provide CICS with a definition that can be used for all connections with the same properties. You customize the supplied autoinstall user program to select an appropriate template for each new connection, depending on the information it receives from CICS.

Any installed IPCONN definition can be used as a template but, for performance reasons, your template should be an installed definition that you do not actually use. The definition is locked while CICS is copying it, and if you have a large number of IPCONNs being autoinstalled at one time the delay may be noticeable.

When the user program is invoked

The user program is invoked when both the following conditions are met:

1. A TCPIPSERVICE that is defined with PROTOCOL(IPIC) receives either a connect flow containing a NETWORKID and APPLID combination for which there is no installed IPCONN definition, or a connect flow with a null APPLID.
2. The URM attribute of the receiving TCPIPSERVICE specifies the name of an autoinstall user program. If the URM attribute contains "NO", the autoinstall request is rejected.

Requirements for autoinstall

For IPCONN autoinstall to be active:

1. The receiving region must have installed at least one TCPIPSERVICE that specifies PROTOCOL(IPIC).
2. The name of the IPCONN autoinstall user program must be specified on the URM option of the installed TCPIPSERVICE definition.

Note: This differs from autoinstall of APPC connections, where the name of the autoinstall user program is specified on the AIEXIT system initialization parameter. There is no equivalent system initialization parameter for IPCONN autoinstall. Instead, the name of the autoinstall user program is specified on the TCPIPSERVICE definition.

As with APPC, putting the template IPCONNs out-of-service disables the autoinstall function.

The autoinstall user program for IPCONN is described in detail in the *CICS Customization Guide*.

Changes to monitoring

Performance data group DFHSOCK

Group DFHSOCK contains new fields:

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.

The following existing field is changed:

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

Any of these fields can be excluded from performance-class monitoring records by specifying its number on the EXCLUDE option of the DFHMCT TYPE=RECORD macro used to build the monitoring control table.

Performance data group DFHTASK

The TRANFLAG field has been modified as follows:

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

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

Exception class records

The following existing field in exception-class records has been changed:

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.

Changes to statistics

CICS now collects statistics on the usage of IP connections. The statistics perform a similar role to ISC/IRC system and mode entry statistics.

The statistics are recorded by specifying the IPCONN option on the CEMT PERFORM STATISTICS and EXEC CICS PERFORM STATISTICS RECORD commands, and retrieved online using the EXTRACT STATISTICS command specifying RESTYPE(IPCONN). They are mapped by the DFHISRDS DSECT.

Changes to problem determination

New domain

The Inter-system (IS) domain manages the resources associated with IPIC requests.

The domain issues trace points with point IDs of the form IS *nnnn* and messages with numbers of the form DFHIS*nnnn*

Changes to security

The IPIC security model is similar to the APPC security model: it uses the same concepts and provides similar facilities, though the facilities are implemented differently, using Secure Sockets Layer (SSL) encryption and authentication.

IP security supports:

- Bind security
- Link security
- User security

IPIC security

This topic introduces the security mechanisms provided for IP interconnectivity connections.

Note: An IPIC connection is also called an“IPCONN”. For information about IPIC connections, see the *CICS Intercommunication Guide*.

The security mechanisms for IPCONNs are similar to those provided for APPC (LU6.2) connections (though they are implemented differently):

- **Bind-time security** prevents an unauthorized remote system from connecting to CICS. On IPCONNs, bind security is enforced by the exchange of Secure Sockets Layer (SSL) client certificates.
- **Link security** defines the complete set of CICS transactions and resources that the remote system is permitted to access across the IPCONN.
- **User security** checks that a user is authorized both to attach a CICS transaction and to access all the resources and SPI commands that the transaction is programmed to use. User security is a subset of link security: that is, a user cannot access a resource, even if it is included in the set defined as accessible by his userid, if is not also included in the set of resources accessible by the link userid.

IPIC bind-time security

A security check can be applied when a request to establish a connection is received from, or sent to, a remote system. This is called *bind-time security*. Its purpose is to prevent an unauthorized system from connecting to CICS.

When CICS uses IPIC to communicate with another CICS region, each CICS system must have an IPCONN resource and a TCPIPSERVICE resource defined.

Each CICS system uses the IPCONN to transmit data to the partner system TCPIPSERVICE, which acts as a receiver. The CICS region that starts the communication is the client, the remote system is the server.

For IPIC, bind security is supported by the exchange of Secure Sockets Layer (SSL) client certificates. To allow two CICS regions to connect successfully, and to prevent an unauthorized system from connecting:

- The SEC system initialization parameter must be “YES” on both regions.
- The IPCONN definitions on both the local and remote regions must specify:
  - SSL(YES).
  - CIPHERS(*cipher_suite_code_list*). This is a string of up to 56 hexadecimal digits that is interpreted as a list of up to 28 2-digit cipher suite codes. When you use CEDA to define the resource, CICS automatically initializes the attribute with a default list of acceptable codes, depending on the level of encryption that is specified by the ENCRYPTION system initialization parameter.
  - Optionally, CERTIFICATE(*X.509_certificate_label*). The named certificate is used as the client certificate, during the SSL handshake when the IPCONN is acquired. If CERTIFICATE is not specified, the default certificate defined in the key ring for the CICS region user ID is used.

The IPCONN defines the *outbound* side of the connection: these settings tell CICS to initiate an SSL handshake. During the SSL handshake, CICS will ask the partner region for the certificate specified on the TCPIPSERVICE. If the remote region TCPIPSERVICE specifies SSL(CLIENTAUTH), the remote system requests the certificate of the originating system as part of the handshake.

- The TCPIPSERVICE definitions on both the local and remote regions specify:
  - PROTOCOL(IPIC).
  - SSL(CLIENTAUTH) or SSL(YES).
  - CIPHERS(*cipher_suite_code_list*).
  - Optionally, CERTIFICATE(*X.509_certificate_label*). The named certificate is used as the server certificate. If CERTIFICATE is not specified, the default certificate defined in the key ring for the CICS region user ID is used.

The TCPIPSERVICE definitions define the *inbound* side of the connection: these settings tell CICS that it must receive a valid SSL client certificate before allowing the client to acquire the IPCONN. These settings also specify that CICS will send the TCPIPSERVICE CERTIFICATE, or the default, when not specified, as a server certificate to the client.

If the TCPIPSERVICE is specified with SSL(YES), the server does not ask for, nor receive, a client certificate during the handshake.

If the TCPIPSERVICEs in both CICS regions are specified with SSL(YES), both CICS regions are authenticated.

If the TCPIPSERVICEs in both CICS regions are specified with SSL(CLIENTAUTH), both CICS regions are authenticated twice.

For most circumstances, an adequate level of security is achieved by specifying TCPIPSERVICE with SSL(YES) in both regions, or SSL(NO) in one region and SSL(CLIENTAUTH) in the other.

Note: If LINKAUTH is specified CERTUSER, the IPCONN must refer to a TCPIPSERVICE that is defined with SSL(CLIENTAUTH).

When the TCPIPSERVICE is specified SSL(NO) on both regions, bind-time security is not possible

If the remote client is trusted by the CICS server, bind time security is not required, however, any user ID and password passed for transaction attach must be valid in the server region's external security manager.

IPIC link security

Link security restricts the resources a user can access, depending on the remote system from which they are accessed. The practical effect of link security is to prevent a remote user from attaching a transaction or accessing a resource for which the link user ID has no authority.

When link security is in use, all requests are given an authority defined by the *link user ID*. For IPCONNs, all requests for a connection have the same link user ID.

Specifying IPIC link security

To specify the link user ID of an IPCONN, use the LINKAUTH option to specify how the user ID for link security is established in a CICS system with security initialized (SEC=YES). You can specify the following:

CERTUSER

TCP/IP communication with the partner system must be configured for SSL and a certificate must be received from the partner system during SSL handshake.

The IPCONN must refer to a TCPIPSERVICE that is defined with SSL(CLIENTAUTH).

The received certificate must be defined to the external security manager so that it is associated with a user ID, which is used to establish link security.

SECUSER

Specifies that the user ID specified in SECURITYNAME is used to establish link security.

If you do not specify a value for SECURITYNAME, CICS uses the default user ID.

In a CICS system with security initialized (SEC=YES), the link user ID is used to establish the authority of the remote system. The link user ID must be a valid RACF user ID on this region. Access to protected resources on this region is based on the RACF user profile and its group membership.

At least one security check is always performed when a transaction is attached by a remote user, but the security checks are minimized if the specified link user ID matches the local region user ID:

If the user IDs match, only one security check is made, as described in “IPIC user security.”

If the user IDs do not match, then for USERAUTH=LOCAL, resource checks are done only against the link user ID. For USERAUTH=IDENTIFY or VERIFY there are always two resource checks. One is against the link user ID, and the second is against the user ID received from the remote user in the attach request. See also “IPIC user security.”

If a failure occurs in establishing link security, the link is given the security of the local region’s default user. This can happen, for example, when the link user ID has been revoked.

IPIC user security

User security causes a second check to be made against a user signed onto a terminal, in addition to the link security described in “IPIC link security” on page 67.

For IPIC connections, the level of user security is specified for inbound requests only, on the USERAUTH attribute of the IPCONN definition. For IPIC connections, you can specify the following types of user authentication:

LOCAL

CICS does not accept a user ID or password from clients. All requests will run under the link user ID, or the default user ID if there is no link user ID.

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

VERIFY
Incoming attach requests must specify a user identifier and a user password. Specify VERIFY when connecting systems are unidentified and cannot be trusted.

DEFAULTUSER
CICS will not accept a user ID and password from the partner system. All requests run under the default user ID.

For outbound requests, the level of user security is specified by the USERAUTH attribute of the IPCONN definition installed in the partner system. CICS sends a user ID when USERAUTH(IDENTIFY) is specified, but not when USERID(LOCAL) is specified. Because CICS does not send passwords to remote systems, USERAUTH(VERIFY) is not supported for communication between CICS TS for z/OS systems.

Chapter 10. TCP/IP management and control

TCP/IP management and control allows you to monitor work that enters or leaves CICS over Transmission Control Protocol/Internet Protocol (TCP/IP) connections. It provides, for TCP/IP networks, a subset of the management functions already provided for Advanced Program-to-Program Communication (APPC, also known as "LUTYPE6.2") networks; plus some additional functions that are not available for APPC or multiregion operation (MRO) networks.

Note: By "TCP/IP network" we mean systems that are interconnected by:

- IPIC connections (IPCONN) . Currently, these can be used only between CICS TS 3.2 regions, and between a CICS TS 3.2 region and a Java client.
- TCP/IP connections from clients that carry, for example, Web Interface, IIOP, or SOAP over HTTP requests inbound to CICS.

TCP/IP management and control enables you, for example, to:

- Use CICSPlex SM, or an equivalent tool, to:
  - Get a CICSplex-wide view of the TCP/IP network.
  - Examine, in real time:
    - The TCP/IP network resources that a particular CICS region is using
    - The work passing in and out of a particular CICS region over the TCP/IP network
    - The CICS resources and tasks associated with a distributed transaction that flows across the CICSplex over the TCP/IP network
    - The CICS region in which a distributed transaction originated
- Save the data collected by CICS so that it can be examined off line, at some point after the tasks and resources that it relates to are no longer available.

Reasons why you might want to use TCP/IP management and control include:

- To diagnose connectivity problems
- To investigate other problems, such as transaction delays
- To track work across the CICSplex
- To capture system data over time, for use in capacity planning
- To monitor the CICSplex

Terminology

This topic contains the new and changed terminology used by TCP/IP management and control.

intersystem communication (ISC)
A CICS facility that provides inbound and outbound support for communication with other computer systems. ISC can be used to connect CICS to both CICS and non-CICS systems, both inside and outside the local sysplex. Contrast with **multiregion operation (MRO)**, and see also **intersystem communication over SNA** and **IP interconnectivity**.

IP interconnectivity
TCP/IP connections that can be used between CICS regions. The CICS TS-CICS TS connections support distributed program link (DPL). Two-phase commit and containers are supported.

J2EE Connector architecture
A standard architecture for connecting the J2EE platform to heterogeneous enterprise information systems (EIS).

JCA J2EE Connector Architecture.

multiregion operation (MRO)
Communication between CICS regions in the same z/OS image or z/OS sysplex without the use of SNA or TCP/IP network facilities. This allows several CICS regions to communicate with each other, and to share resources such as files, terminals, temporary storage, and so on. The systems must be in the same operating system (z/OS image); or, if the XCF access method is used, in the same z/OS sysplex. Contrast with **intersystem communication (ISC)** .

transaction group ID
A CICS token that, in a TCP/IP network, binds all the tasks that make up a distributed transaction.

Changes to CICS externals

Changes to the system programming interface

To support the new function, there are two new system programming interface (SPI) commands, and new parameters and values have been added to several existing commands.

Note: Changes to the EXTRACT STATISTICS and PERFORM STATISTICS command are described in "Changes to statistics" on page 75

New SPI commands

INQUIRE ASSOCIATION

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

For details of the command, see "INQUIRE ASSOCIATION" on page 295.

INQUIRE ASSOCIATION LIST

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

For details of the command, see "INQUIRE ASSOCIATION LIST" on page 302.

INQUIRE TASK command

The INQUIRE TASK command has 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.

Changes to CICSPlex SM views and tables

CICSPlex SM views

The following view sets are new:

IP facilities (EYUSTARTIPFACIL)
The **IP facilities** views display information about the associations between active CICS tasks and the IP connections in use by those tasks.

To access from the WUI main menu, click:

CICS operations views > Connection operations views > IP facilities, or

CICS operations views > TCP/IP service operations views > IP facilities

Task association data (EYUSTARTTASKASSC)
The **Task association data** views set display information about all tasks in the system summarized by their transaction group ID, and showing the tasks' origin descriptors.

Note: A transaction group ID is a CICS token that binds all tasks that make up a local transaction or, in a TCP/IP network, a distributed transaction.

To access these views from the WUI main menu, click:

CICS operations views > Task operations views > Task association data

CICSPlex SM resource tables

The following resource tables are new:

IPFACIL
A CICS resource that describes the association between an active CICS task and the IPIC connection it uses.

TASKASSC
A CICS resource that, in a TCP/IP network, describes the tasks that make up a distributed transaction.

All CICSPlex SM resource tables are described in the *CICSPlex System Manager Resource Tables Reference*.

Changes to global user exits

There is one new global user exit point, XAPADMGR, the Application Associated Data exit, in the AP domain

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.

The exit program is invoked, if enabled, at the attach of non-system tasks for which no input Origin Descriptor Record is provided.

On input, the exit program is passed the task's association data. It could find other relevant information, for inclusion in the Associated Data Origin Descriptor, from other sources, using CICS commands.

Note: Distributed transactions that use DPL over IPIC connections pass their transaction group ID and origin data, including the user correlator, to be inherited by the mirror task in the target region.

The exit program could perform other activities, such as logging of information found in the association data, for purposes such as auditing or accounting of workloads.

Exit XAPADMGR

When invoked

At the attach of a non-system task that has no inherited Associated Data Origin Descriptor data passed to it.

Exit-specific parameters

UEPADCB

Address of the selectable association data control block. This is mapped by the DFHMNADS DSECT.

UEPADCBL

Length, in bytes, of the associated data control block.

UEPUCD

Address of a 64-byte output area in which the exit program can place the user correlation data.

Return codes

UERCNORM

Continue processing.

XPI calls

All can be used.

API and SPI calls

All can be used, except for:

- EXEC CICS ABEND
- EXEC CICS PERFORM SHUTDOWN

Sample exit program, DFH$APAD

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.

DFH$APAD performs the following processing:

- Provides addressability to the associated data provided as input to the exit.
- Chooses a field from this data and places it in the output buffer.
- Adds a field to the user correlation data in the output buffer.

Changes to monitoring

Performance data group DFHCICS

DFHCICS contains new fields:

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.

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.

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.

Any of the above fields can be excluded from performance-class monitoring records by specifying its number on the EXCLUDE option of the DFHMCT TYPE=RECORD macro used to build the monitoring control table.

Changes to statistics

CICS now collects statistics on the usage of IP connections. The statistics perform a similar role to ISC/IRC system and mode entry statistics.

The statistics are recorded by specifying the IPCONN option on the CEMT PERFORM STATISTICS and EXEC CICS PERFORM STATISTICS RECORD commands, and retrieved online using the EXTRACT STATISTICS command specifying RESTYPE(IPCONN). They are mapped by the DFHISRDS DSECT.

Chapter 11. New application programming capabilities for CICS Web support

The capabilities of the application programming interface, and your architecture options for CICS Web support, are extended in several ways. For example, you can now use WEB API commands in converter programs and the DFHWBEP Web error program, and the WEB API commands support the use of containers and channels.

- New programming samples using the WEB API commands are provided to help you construct your own Web-aware application programs for both CICS as an HTTP server and CICS as an HTTP client. The new samples, provided in Assembler, C, and COBOL, demonstrate HTTP chunking and pipelining functions.
- The existing DFH$WB1A and DFH$WB1C samples, used to verify CICS Web support operation, are updated to use the WEB API commands, and a sample URIMAP definition is provided for use with DFH$WB1C.
- The WEB API commands can now be used in converter programs and in the user-replaceable Web error program DFHWBEP. Instead of constructing the responses from these programs manually in a block of storage, you can, if you wish, migrate to using WEB API commands to construct and send the response. You must however specify ACTION(IMMEDIATE) in your command, as the default of ACTION(EVENTUAL) is not permitted with DFHWBEP. Using the error program enables you to take advantage of all the available CICS Web support features, including assistance with structuring responses and with HTTP protocol compliance, and enhanced code page conversion capabilities, with minimal disruption to your existing CICS Web support architecture. The input and output parameter lists remain unchanged for these programs, and the information provided by these facilities can be used in combination with the WEB API commands.
- The Web error program DFHWBEP is now called if an error occurs during the delivery of static responses. You can use the program to customize the error responses that CICS sends to the Web client. CICS messages are also produced when the errors occur.
- Containers and channels can now be used on the WEB API commands, for both CICS as an HTTP server and CICS as an HTTP client.
  - The WEB SEND (Client and Server), WEB RECEIVE (Client and Server) and WEB CONVERSE commnads now provide the facility to send and receive the body of an HTTP message using a combination of containers and buffers.
  - Containers are also used to store HTTP headers, so you do not need to define a prefix for temporary storage queues for CICS Web support in the TCPIPSERVICE definition.
- On the PUT CONTAINER (CHANNEL) and GET CONTAINER (CHANNEL) commands, if you prefer, you can now specify a supported IANA-registered charset name for code pages for data conversion, instead of using the numeric Coded Character Set Identifiers (CCSIDs). You can also specify that data held in a container is retrieved without conversion, and return the CCSID of the unconverted data.
- CICS documents and document templates can now be converted to and from the UTF-8 and UTF-16 character encodings.
- The WEB API commands for examining form fields in a request (WEB READ FORMFIELD and WEB STARTBROWSE FORMFIELD) now provide support for data received in UTF-8 and UTF-16 formats.

- The WEB READ FORMFIELD, WEB STARTBROWSE FORMFIELD and DOCUMENT RETRIEVE commands now use the CHARACTERSET option to allow you to specify a code page for your client and application program.
- The WEB READ FORMFIELD and WEB STARTBROWSE FORMFIELD commands now respect CHARACTERSET and HOSTCODEPAGE for GET requests as well as POST requests.
- To improve performance, CICS now sends an HTTP request with the OPTIONS method to identify the version of an HTTP server only when this is required, rather than each time you open an HTTP client connection to a server.

Identifying HTTP version of server: improved performance

For some activities specific to the HTTP/1.1 protocol, such as chunked transfer-coding, it might be important for the client application to know the HTTP version of the server. When you open an HTTP client connection to a server using the WEB OPEN command, CICS does not now automatically make a request to the server with the OPTIONS method. Instead, wherever possible, CICS waits until a response is received from the server and uses this to determine the server version. The OPTIONS request is made only where necessary.

In CICS Transaction Server for z/OS, Version 3 Release 1, to determine the HTTP version of the server, CICS made an HTTP request with the OPTIONS method whenever the WEB OPEN command was issued. The result of the request could be returned on the WEB OPEN command. In CICS Transaction Server for z/OS, Version 3 Release 2, CICS makes an HTTP request with the OPTIONS method on the WEB OPEN command only if you specify the HTTPVNUM and HTTPRNUM options on the WEB OPEN command.

When CICS receives the first response from the server, the server's HTTP version can be identified from the response, and CICS applies this for the remainder of the session. However, if no HTTP request with the OPTIONS method is made at the time of the WEB OPEN command, the server's HTTP version remains unidentified. In some circumstances, CICS needs to find out the HTTP version before the first response is received from the server, in order to apply version-specific behaviors, or to respond to a WEB EXTRACT command. If you take the following actions as an HTTP client before or during the first request that you make to the server, CICS needs to make an HTTP request with the OPTIONS method:

- Specifying the HTTPVNUM and HTTPRNUM options on the WEB OPEN command.
- Specifying the HTTPVERSION and VERSIONLEN options on the WEB EXTRACT command for CICS as an HTTP client.
- Using the WEB WRITE HTTPHEADER command to write a Trailer header (which is used with chunked messages).
- Using the WEB SEND command with the CHUNKING(CHUNKYES) option to send a client request involving a chunked message.
- Using the WEB SEND command with the ACTION(EXPECT) option to send an Expect header and await a 100-Continue response before sending the message body to the server.

To benefit from this performance improvement in your CICS Web support client applications, try to eliminate any unnecessary HTTP requests with the OPTIONS method:

- Remove the HTTPVNUM and HTTPRNUM options from the WEB OPEN command if you are not making use of the responses. You should only need this information if it is essential for you to confirm whether a planned action by your application, before or during its first request, is acceptable. After the first response is received from the server, the version will be known without the need for an HTTP request with the OPTIONS method.

- Before making your first request to the server, only issue the WEB EXTRACT command with the HTTPVERSION and VERSIONLEN options if you need to make use of the response. Again, issuing this command after the first response is received does not trigger an HTTP request with the OPTIONS method.
- When making your first request to the server, reserve the use of the ACTION(EXPECT) option on the WEB SEND command for the sending of large, or important, message bodies.

If your first request to the server is a chunked message, you cannot avoid an HTTP request with the OPTIONS method.

Actions by your application which are dependent on the HTTP version of the server include:

- Writing HTTP headers that request an action which might not be carried out correctly by a server below HTTP/1.1 level.
- Using HTTP methods that might be unsuitable for servers below HTTP/1.1 level.
- Using chunked transfer-coding.
- Sending a pipelined sequence of requests.

Bear in mind that you do not always need to check the HTTP version of the server before carrying out actions dependent on the version. Consult the HTTP specification to which you are working to see whether it is acceptable to attempt the action with a server of the wrong version. For example, some unsuitable HTTP headers might simply be ignored by the recipient. You might be able to attempt the request without checking, and handle any error response from the server.

Verifying the operation of CICS Web support

Sample programs DFH$WB1A (Assembler) and DFH$WB1C (C) are provided to help you test that CICS Web support is working.

DFH$WB1A can be accessed using the CICS-supplied sample analyzer program DFHWBADX. DFH$WB1C can be accessed using the supplied sample URIMAP definition DFH$URI1, or the sample analyzer program. If you plan to use CICS as an HTTP client, note that the CICS-supplied sample programs for pipelining client requests work with a CICS region that has DFH$WB1C and DFH$URI1 set up, so you might want to choose this option now.

The sample programs use EXEC CICS WEB and DOCUMENT commands to receive your request and construct and send a simple response. They construct HTTP responses like this:

```
DFH$WB1A on system applid successfully invoked through CICS Web support
```

where *applid* is the applid of the CICS system in which CICS Web support is running.

To run the sample programs:

1. Modify as necessary, and then install, the sample TCPIPSERVICE definition HTTPNSSL, which is provided in group DFH$SOT. The CICS-supplied sample analyzer program DFHWBADX is specified in the TCPIPSERVICE definition. You might need to change the following options:

   a. **PORTNUMBER**: HTTPNSSL uses port 80, the well known port number for HTTP. If port 80 is not reserved for the use of CICS, specify another port belonging to z/OS Communications Server that you have reserved for the use of CICS.
   b. **IPADDRESS**: HTTPNSSL does not specify an IP address, so this defaults to the IP address corresponding to the default z/OS Communications Server TCP/IP stack. This situation is the most usual. If you have multiple TCP/IP stacks in your z/OS image, and you want to use a nondefault stack, you need to specify the dotted decimal IP address corresponding that stack.
2. If you want to use the sample program DFH$WB1C, install its PROGRAM resource definition, which is provided in the DFH$WEB resource definition group. The PROGRAM resource definition for DFH$WB1A is in the DFHWEB resource definition group, which is already installed as part of DFHLIST.
3. If you are using the sample program DFH$WB1C, and you want to try using a URIMAP definition, install the supplied sample URIMAP definition DFH$URI1, which is provided in the DFH$WEB resource definition group.
4. At a Web browser, enter a URL that connects to CICS Web support using the following URL components:

   Scheme
   HTTP

   Host The host name assigned to the z/OS image. If you do not know the host name, you can use the dotted decimal IP address from the HTTPNSSL TCPIPSERVICE definition. If you did not specify the IP address explicitly, it has been filled in by CICS, and you can view it in the installed TCPIPSERVICE definition.

   Port number
   The port number specified in the TCPIPSERVICE definition. If this is 80, you do not need to specify it explicitly.

   Path
   - To access DFH$WB1A, use the path `/CICS/CWBA/DFH$WB1A`
   - To access DFH$WB1C, use the path `/sample_web_app` if you have installed the sample URIMAP definition, or the path `/CICS/CWBA/DFH$WB1C` if you want to use the sample analyzer program instead.
5. Unless you want to carry out further testing now, uninstall the sample TCPIPSERVICE definition HTTPNSSL, and disable the URIMAP definition DFH$URI1. You can replace HTTPNSSL with your own properly architected TCPIPSERVICE definition later on.

Changes to CICS externals

Application programming interface changes

The DOCUMENT RETRIEVE command

The DOCUMENT RETRIEVE command now uses the CHARACTERSET option to allow you to specify a code page for your Web client and application program. The alternative option, CLNTCODEPAGE, is used for migration purposes only.

The DOCUMENT RETRIEVE command can now be used to determine a required document length before allocating storage for the document. If you issue a DOCUMENT RETRIEVE with a MAXLENGTH of 0, this is no longer rejected with an invalid length error (LENGERR condition RESP2 = 1). The request is processed normally so that the document is truncated to a length of zero (LENGERR condition RESP2 = 2) and the exact length required is returned in the LENGTH field.

The GET CONTAINER (CHANNEL) and PUT CONTAINER (CHANNEL) commands

On the GET CONTAINER (CHANNEL) command, there are three new options, CCSID, INTOCODEPAGE and CONVERTST; and changes to the description of the INTOCCSID option.

CCSID(data-area)
returns a fullword that contains the Coded Character Set Identifier (CCSID) of the data returned by the CONVERTST(NOCONVERT) option. This option allows you to retrieve containers with a DATATYPE of CHAR, without converting the data. If a DATATYPE of BIT is specified for the container, this value is zero.

INTOCCSID(data-value)
specifies the Coded Character Set Identifier (CCSID) into which the character data in the container is to be converted, as a fullword binary number. If you prefer to specify an IANA name for the code page, or if you prefer to specify the CCSID as alphanumeric characters, use the INTOCODEPAGE option instead.

For CICS Transaction Server for z/OS applications, the CCSID is typically an EBCDIC CCSID. (However, it is possible to specify an ASCII CCSID if, for example, you want to retrieve ASCII data without it being automatically converted to EBCDIC.)

If INTOCCSID and INTOCODEPAGE are not specified, the value for conversion defaults to the CCSID of the region. The default CCSID of the region is specified on the **LOCALCCSID** system initialization parameter.

Only character data can be converted, and only then if a DATATYPE of CHAR was specified on the PUT CONTAINER command used to place the data in the container. (A DATATYPE of CHAR is implied if FROMCCSID or FROMCODEPAGE is specified on the PUT CONTAINER command.)

For more information about data conversion with channels, see the *CICS Application Programming Guide*.

For an explanation of CCSIDs, and a list of the CCSIDs supported by CICS, see the *CICS Family: Communicating from CICS on zSeries*.

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*.

CONVERTST(cvda)
specifies that data held in a container is retrieved without being converted.

NOCONVERT

The container data is retrieved without being converted. If you have used the WEB RECEIVE to store the HTTP body in a container, and you need to retrieve the body unconverted from that container, you must use the NOCONVERT option.

On the PUT CONTAINER (CHANNEL) command, there is a new option FROMCODEPAGE, which operates in the same way to specify the code page from which the data is to be converted when it is put into the container. There are similar changes to the description of the FROMCCSID option as for the INTOCCSID option shown above.

Both the GET CONTAINER (CHANNEL) command and the PUT CONTAINER (CHANNEL) commands have a new error condition CODEPAGEERR with a number of new RESP2 values. The new error condition is used for conversion errors when the INTOCODEPAGE and FROMCODEPAGE options are specified. The existing CCSIDERR error condition is used when the INTOCCSID and FROMCCSID options are specified.

The WEB CONVERSE command

The WEB CONVERSE command now supports use of channels and containers to send and receive an HTTP body. New options are BODYCHARSET, CHANNEL, CONTAINER, TOCHANNEL and TOCONTAINER . The command syntax and error condition RESP2 values are the same as those specified in the WEB RECEIVE and WEB SEND commands.

The WEB CONVERSE command also allows you to provide basic authentication credentials (a username and password), by specifying the new option, AUTHENTICATE. The command syntax and error condition RESP2 values are the same as those specified in the WEB SEND (Client) command.

The WEB EXTRACT command

There is a change to the description of the HTTPVERSION option on the WEB EXTRACT command:

HTTPVERSION*(data-area)*

For CICS as an HTTP server, this option specifies a buffer to contain the HTTP version for the Web client, as stated on its request.

For CICS as an HTTP client (with the SESSTOKEN option), this option specifies a buffer to contain the HTTP version of the server in the connection identified by the SESSTOKEN option. If CICS does not already know the HTTP version of the server, CICS makes a request to the server with the OPTIONS method to find out this information.

″1.1″ indicates HTTP/1.1, and ″1.0″ indicates HTTP/1.0 or lower.

The WEB EXTRACT command also allows you to specify the realm or security environment that contains the data you are requesting. The new options are REALM and REALMLEN, with associated RESP2 error conditions.

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 WEB OPEN command

There is a change to the description of the HTTPRNUM and HTTPVNUM options on the WEB OPEN command:

HTTPRNUM_(data-area)_

returns the release number for the HTTP version of the server, as a halfword binary value. (HTTPVNUM returns the version number.) For example, if the server is at HTTP/1.0 level, HTTPRNUM returns 0.

HTTPVNUM_(data-area)_

returns the version number for the HTTP version of the server, as a halfword binary value. (HTTPRNUM returns the release number.) For example, if the server is at HTTP/1.0 level, HTTPVNUM returns 1.

If you specify the HTTPVNUM and HTTPRNUM options, CICS obtains the HTTP version information when it opens the connection to the server. If the server does not provide HTTP version information in response to this request, or the version is lower than 1.0, CICS assumes that it is at HTTP/1.0 level.

Specify these options if it is essential for you to check the HTTP version information to confirm whether a planned action by your application, before or during its first request, will succeed. Actions dependent on the HTTP version include:

- Writing HTTP headers that request an action which might not be carried out correctly by a server below HTTP/1.1 level.
- Using HTTP methods that might be unsuitable for servers below HTTP/1.1 level.
- Using chunked transfer-coding.
- Sending a pipelined sequence of requests.

The additional HTTP request made by CICS to obtain the HTTP version information has an impact on performance, so do not specify these options if it is not necessary at this stage. When the first response has been received from the server, you can obtain this information using the WEB EXTRACT command.

The WEB READ FORMFIELD and WEB STARTBROWSE FORMFIELD commands

The WEB READ FORMFIELD and WEB STARTBROWSE FORMFIELD commands, like the other WEB API commands, now use the CHARACTERSET option to allow you to specify a code page for your Web client and application program. This applies to both GET and POST methods. The alternative option, CLNTCODEPAGE, is now used for migration purposes only.

The WEB RECEIVE (Client and Server) commands

The WEB RECEIVE (Client) and WEB RECEIVE (Server) commands now support use of channels and containers to receive an HTTP body. New options are BODYCHARSET, MEDIATYPE, TOCHANNEL and TOCONTAINER. Associated RESP2 error conditions are also provided.

BODYCHARSET*(data-area)*
specifies the character set of the HTTP request body.

The name of the character set can consist of up to 40 alphanumeric characters, including appropriate punctuation.
If the HTTP body is received into an application buffer, the character set returned is as follows:

- If the INTO or SET option is specified, and the HTTP body is converted, CICS returns the character set of the HTTP body before conversion.
- If the INTO or SET option is specified, and the HTTP body is not converted, CICS returns the charset specified in the Content-Type header. If charset information is not available, blanks are returned.

If the HTTP body is received into a named container, the character set returned is as follows:

- If the container is a CHAR container, CICS returns the character set of the encoded data.
- If the container is a BIT container, CICS returns blanks.

If the value returned is more than 40 bytes, the data is truncated. If the value returned is less than 40 bytes, the data is padded to the right with blanks.

MEDIATYPE *(data-area)*
specifies the data content of any message body provided, for example `text/xml`. The media type is up to 56 alphanumeric characters, including appropriate punctuation. For more information on media types, see IANA media types and character sets in the *CICS Internet Guide*.

TOCHANNEL*(data-value)*
specifies the name of the channel that the container belongs to. The name of the channel can consist of up to 16 alphanumeric characters, including appropriate punctuation. The acceptable characters are A-Z a-z 0-9 $ @ # / % & ? ! : | ″ = , ; < > . - and _. Leading and embedded blanks are not permitted. If the name is less than 16 characters, it is padded with trailing blanks.

If you plan to ship your channels between CICS regions, bear in mind that you should restrict your characters to standard alphanumeric characters (A-Z 0-9 & : = , ; <>. - _) to ensure they are represented in the same way by all EBCDIC code pages.

If the TOCHANNEL option is not specified, then CICS assumes the current channel.

TOCONTAINER*(data-value)*
specifies the name of the container where the data is placed. The name of the container can consist of up to 16 alphanumeric characters, including appropriate punctuation. The acceptable characters are A-Z a-z 0-9 $ @ # / % & ? ! : | ″ = , ; < > . - and _. Leading and embedded blanks are not permitted. If the name is less than 16 characters, it is padded with trailing blanks.

If you plan to ship your containers between CICS regions, bear in mind that you should restrict your characters to standard alphanumeric characters (A-Z 0-9 & : = , ; <>. - _) to ensure they are represented in the same way by all EBCDIC code pages.

Do not use container names starting with ″DFH″, unless requested to do so by CICS.

The TOCONTAINER option can only be specified on the first WEB RECEIVE command.

The WEB SEND (Client and Server) commands

The WEB SEND (Client) and WEB SEND (Server) commands now support use of channels and containers to send an HTTP body. New options are CHANNEL and CONTAINER, with associated RESP2 error conditions.

CHANNEL*(data-value)*
specifies the name of the channel that the container belongs to. The name of the channel can consist of up to 16 alphanumeric characters, including appropriate punctuation. Leading and embedded blanks are not permitted. If the name is less than 16 characters, it is padded with trailing blanks.

If the CONTAINER option is specified, CHANNEL is optional.

If the CHANNEL option is not specified, then CICS assumes the current channel.

CONTAINER*(data-value)*
specifies the name of the container where the HTTP body is held, before it is sent to the server. The name of the container can consist of up to 16 alphanumeric characters, including appropriate punctuation. Leading and embedded blanks are not permitted. If the name is less than 16 characters, it is padded with trailing blanks.

The WEB SEND (Client) command also allows you to provide basic authentication credentials (a username and password), by specifying the new option, AUTHENTICATE, with associated credentials PASSWORD, PASSWORDLEN, USERNAME and USERNAMELEN. New error condition RESP2 values are also provided.

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.

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.

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

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.

Changes to the JCICS application programming interface

JCICS exception provided for Document and Webservice methods

The JCICS application programming interface exception, NotAuthorisedException, can now be thrown by any of the following methods:

```
com.ibm.cics.server.Document ()
com.ibm.cics.server.Document.create*()
com.ibm.cics.server.Document.append*()
com.ibm.cics.server.Document.insert*()
com.ibm.cics.server.Webservice.invoke()
```

Support for new datatype for containers

The datatype CHAR is now supported by the JCICS API for use in the Container class. This datatype can be used in addition to the existing BIT datatype. Use of the new CHAR datatype is available through the following constructor, constants and methods:

New `com.ibm.cics.server.Container ()` constructor

New `Container.DATATYPE_BIT` and `Container.DATATYPE_CHAR` constants

New `getDatatype()` getter method

New version of the `get()` method

New version of the `getLength()` method

New version of the `put (byte[])` method

New document deletion facility in the Document class

The following methods are available in the Document class to support deletion of documents within the WEB SEND (Server and Client) commands:

New `com.ibm.cics.server.Document.delete()` method

New version of the `com.ibm.cics.server.Document.sendDocument()` method

New HttpClientRequest methods to support Client Basic Authentication

New methods are available in the HttpClientRequest class as follows:

```
com.ibm.cics.server.HttpClientRequest.setAuthenticate()
com.ibm.cics.server.HttpClientRequest.setUserName()
com.ibm.cics.server.HttpClientRequest.setPassword()
```

Support for CHARACTERSET option on DOCUMENT RETRIEVE, WEB RECEIVE, WEB READ FORMFIELD and STARTBROWSE FORMFIELD commands

All instances of `ClientCodepage` are changed to `Characterset`. This is a documentation change only, and does not affect existing code, or the externals of the class.

Support for channels and containers in WEB API (WEB RECEIVE and WEB SEND commands)

New methods are available in the HttpRequest and WebReceive classes to set the container and channel names for WEB RECEIVE (Server) as follows:

New `setContainer()` method

New `setChannel()` method

New `getContentAsContainer()` methods

New `getBodyCharset()` method

New methods are available in the HttpResponse class to set the container and channel names for WEB RECEIVE (Client) as follows:

New `setContainer()` method

New `setChannel()` method

New `getContentAsContainer()` methods

New `getBodyCharset()` method

A new method version is available in the HttpResponse class for WEB SEND (Server) as follows:

New version of `sendDocument()` method

A new method and additional exceptions to support containers are available in the HttpClientRequest class for WEB SEND (Client) as follows:

New `setContainer()` method

New `sendDocument()` exceptions

Changes to resource definition

New CICS-supplied resource definition group DFH$WEB

The new resource definition group DFH$WEB contains most of the samples for CICS Web support. The exception is the Assembler sample program DFH$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 sample programs

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.

Changes to problem determination

Messages

New messages DFHWB0758–DFHWB0761 are issued for errors that occur during the delivery of static responses from a URIMAP definition. The default error responses issued to the Web client in each of these situations can be tailored using the user-replaceable Web error program DFHWBEP.

Trace

New trace points WB 0509–WB 050D relate to errors that occur during the delivery of static responses from a URIMAP definition.

Chapter 12. Security changes for CICS Web support

For CICS as an HTTP server, basic authentication, client certificate authentication and resource level security are now available for CICS documents or z/OS UNIX files delivered as a static response (using a URIMAP definition with the TEMPLATENAME or HFSFILE option). You can now apply access controls for these items, based on a client's user ID, without needing to use an application program to handle the requests. The resource level security capability can also be used to provide a more granular level of security for CICS document templates used by an application program for Web delivery as part of an application-generated response. In addition, the realm supplied for basic authentication can now be customized.

Resource security for document templates

You can apply access controls to individual CICS document templates. Security checking for this resource is applied using the XRES system initialization parameter, which is set to YES by default. You can use this capability to secure individual Web pages delivered as static responses (using URIMAP definitions). You can also secure document templates that are used by application programs, either for Web delivery as part of an application-generated response, or for any other purpose.

The XRES system initialization parameter activates security checking for CICS document templates. The default setting for this system initialization parameter is YES, meaning that each time a document template is requested, CICS calls the external security manager to check that the user ID associated with the transaction is permitted to access the template. When YES is specified, the default resource class name RCICSRES and grouping class name WCICSRES are used. Alternatively, you can specify a different resource class name. If you set XRES to NO, no security checks are performed for document templates.

Access to CICS document templates is controlled in the following cases:

- Document templates delivered as a static response to a Web client's request (specified on the TEMPLATENAME attribute of the URIMAP definition for the request).
- Document templates delivered as part of an application-generated response to a Web client's request (used by an application program that handles the request).
- All EXEC CICS CREATE, INQUIRE, DISCARD and SET DOCTEMPLATE commands.
- All EXEC CICS DOCUMENT INSERT and CREATE commands with the TEMPLATE option.

As with security for other resources, for security checking to be carried out, the system initialization parameter SEC must be set to YES, and RESSEC(YES) must be specified in the transaction resource definition for the transaction attempting to access the resource.

When calling the external security manager, CICS uses the name of the DOCTEMPLATE resource definition for the CICS document template, prefixed by its resource type, DOCTEMPLATE. For example, for a document template whose resource definition is named “WELCOME”, the profile name passed to the security manager is DOCTEMPLATE.WELCOME. If you have used the system initialization parameter SECPRFX to add an additional prefix specific to the CICS region, this prefix is also used. For example, if the document template is in a development region where SECPRFX is set to TEST, then the profile name TEST.DOCTEMPLATE.WELCOME is passed to the security manager. You need to set up the profile names using this format.

The EXEC CICS DOCUMENT commands reference document templates using the 48-character name of the template (as specified in the TEMPLATENAME attribute of the DOCTEMPLATE resource definition). However, security checking for these commands uses the name of the DOCTEMPLATE resource definition that corresponds to the TEMPLATENAME attribute. This means that you only need to set up one profile name for each document template, using the name of the DOCTEMPLATE resource definition, and not the TEMPLATENAME attribute.

Note: Document templates can be retrieved from a variety of sources, including partitioned data sets, CICS programs, CICS files, z/OS UNIX System Services files, temporary storage queues, transient data queues, and exit programs. When resource security checking is carried out for a document template, CICS does **not** perform any additional security checking on the resource that supplies the document template, even if resource security is specified for that type of resource in the CICS region.

Security for z/OS UNIX files

Files stored in the z/OS UNIX System Services file system can be used to supply Web pages through CICS Web support, as static responses provided by URIMAP definitions. When access control for these files is specified, you can control access to them on the basis of the user IDs for individual Web clients. Access control for z/OS UNIX files is enabled by default.

Access control for z/OS UNIX files is activated by the XHFS system initialization parameter. The default for this parameter is YES, meaning that resource security for z/OS UNIX files is active. If you do **not** want resource security for these files, set this system initialization parameter to NO.

Access control for z/OS UNIX files is based on a user ID that is obtained from the Web client using basic authentication, or a user ID associated with a client certificate sent by the Web client. The user ID is used only during the process of security checking.

Access control for z/OS UNIX files differs from standard resource security for the other resource types controlled by X*name* system initialization parameters, in some important ways:

- Access controls for z/OS UNIX files are not managed directly by RACF®. They are specified in z/OS UNIX System Services, which makes use of RACF to manage user IDs and groups of user IDs, but keeps control of the permissions set for the files and directories. Because of this, you do not need to define RACF profiles for individual files, and you cannot use the QUERY SECURITY command to check access to them. You check and specify permissions for z/OS UNIX files and directories in the z/OS UNIX System Services shell environment, using z/OS UNIX commands. RACF is used to manage user profiles, groups and access control lists (ACLs). If you are using ACLs, you need to activate the FSSEC class for these to be checked.
- Security checking for z/OS UNIX files is not affected by 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 (and their

directories) are subject to security checking, regardless of the RESSEC attribute for the transaction that is accessing them. (However, the SEC system initialization parameter does affect whether or not security checking is carried out, as for all resources.)

- z/OS UNIX files are not referenced directly by any CICS application programming commands or system programming commands. They can only be 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. CICS does **not** perform any additional permissions check on the z/OS UNIX files using the Web client's user ID. This is the case even if access control is specified for z/OS UNIX files in the CICS region, or if resource security is not active for document templates. Where z/OS UNIX files are defined as CICS document templates, you therefore need to set up Web clients' user ID access controls in RACF for the CICS document templates, rather than in z/OS UNIX System Services for the z/OS UNIX files. (However, the CICS region user ID always needs to have **read** permissions on z/OS UNIX files, even if they are defined as document templates.) Note in particular that this situation applies to all application-generated responses from CICS Web support, and to any URIMAP definitions for static responses where the TEMPLATENAME attribute is used, rather than the HFSFILE attribute.

User IDs for access to document templates and z/OS UNIX files used by CICS Web support

When resource security is active for a transaction, the external security manager checks whether the user ID associated with the transaction is authorized to access the required resources. For CICS Web support, the user ID associated with the transaction for a particular Web request can be obtained from different sources. Depending on the level of security that you require, you can arrange your CICS Web support architecture to determine the user IDs that are used for resource security checking against the secured document templates or z/OS UNIX files.

Application-generated responses

For CICS Web support, the transaction for application-generated responses is an alias transaction, which can be specified in the URIMAP definition for the request or set by an analyzer program, and defaults to CWBA. CWBA is defined as RESSEC(NO), so if you require resource security for the alias transaction, you must either copy the CWBA definition to your own group and change its RESSEC attribute, or use a different alias transaction.

When a Web client makes a request to CICS Web support, and the response is provided by an application, CICS selects a userid for the alias transaction in the following order of priority:

1. A user ID that you set using an analyzer program. This user ID can override a user ID obtained from the Web client or supplied by a URIMAP definition.
2. A user ID that you obtained from the Web client using basic authentication, or a user ID associated with a client certificate sent by the Web client.
3. A user ID that you specified in the URIMAP definition for the request.
4. The CICS default user ID, if no other can be determined.

For application-generated responses, the user ID selected for the Web client applies to the whole alias transaction, and must be authorized to attach the transaction and use the Web application program, as well as to use secured document templates.

Web clients' user IDs do not need specific permissions on z/OS UNIX files for application-generated responses, because applications can only manipulate z/OS UNIX files using EXEC CICS commands when the files are defined as CICS document templates. Security checking is only carried out for the CICS document template, and not again for the z/OS UNIX file.

Static responses

The transaction for all static responses specified in URIMAP definitions is the default Web attach task CWXN, or any alias transaction that you have specified in place of CWXN using the TRANSACTION attribute on your TCPIPSERVICE definitions.

When a Web client makes a request to CICS Web support, and the response is a static response specified in a URIMAP definition, the user ID used for the Web client is a user ID that you obtained from the Web client using basic authentication, or a user ID associated with a client certificate sent by the Web client.

Resource security checking for document templates is controlled by the XRES system initialization parameter and the RESSEC attribute for the transaction (CWXN or its alias). Access control for z/OS UNIX files is controlled by the XHFS system initialization parameter only.

The user ID for the Web client is used only during the process of resource security checking for the document template or z/OS UNIX file that is to be delivered as the static response. The user ID must be authorized to access the document template or z/OS UNIX file.

HTTP authentication for static responses

In CICS Transaction Server for z/OS, Version 3 Release 1, basic authentication and client certificate authentication were not available where a static response with a URIMAP definition was used. Now, authentication is possible for both application-generated responses and static responses.

Basic authentication and client certificate authentication for static responses are implemented in the same way as for application-generated responses. The requirements for authentication are specified by the AUTHENTICATE and SSL attributes of the TCPIPSERVICE resource definition for the port where the requests are made.

You can require basic authentication, where the Web client specifies a user ID and password that is already known to the security manager, or client certificate authentication, where the Web client sends a certificate that is registered to the security manager and associated with a user ID. You can also allow automatic registration, where clients who send a valid certificate but are not yet known to your system can register their own user ID and password with the security manager. If basic authentication is used but the client's password has expired, the CICS-supplied utility program DFHWBPW manages the password renewal process. You can customize or replace the pages which DFHWBPW presents to the user.

Basic authentication and client certificate authentication can be used to police access to CICS Web support in general. In addition, when you have obtained an authenticated user ID for a Web client, you can use this ID to implement resource level security for the resources in the CICS region which you are using to provide the response.

A key difference between the use of authentication with static responses, and the use of authentication with application-generated responses, is that for static responses, it is not possible to override a user ID that you have obtained from the Web client using basic authentication or client certificate authentication. If you are obtaining authenticated user IDs for Web clients, these must be the basis of any resource security checking that you carry out. For application-generated responses, you can supply an override using an analyzer program.

For static responses, you can implement basic authentication or client certificate authentication on the TCPIPSERVICE resource definition without needing to implement security at the level of individual resources. This is because for static responses, the Web client's user ID is used only during the process of resource security checking, and the CICS region user ID still applies to all the other activities of the transaction. Web clients' user IDs only need to be authorized for specific resources (CICS documents or UNIX files) if you decide to implement resource level security. However, note that resource level security is active according to CICS system defaults, and must be deactivated if it is not wanted.

Realm for HTTP basic authentication

You can now customize the realm supplied by CICS during the process of basic authentication.

During HTTP basic authentication, the server requests authentication information (a user ID and password) from a client. The server makes this request by sending an HTTP response with a 401 status code, and a WWW-Authenticate header.

The format of a WWW-Authenticate header for HTTP basic authentication is:

```
WWW-Authenticate: Basic realm="Our Site"
```

The WWW-Authenticate header contains a realm attribute, which identifies the set of resources to which the authentication information requested (that is, the user ID and password) will apply. Web clients display this string to the end user when they request a user ID and password. Each realm may require different authentication information. Web clients may store the authentication information for each realm so that end users do not need to retype the information for every request.

You can now use the REALM attribute on the TCPIPSERVICE definition to customize the realm that is provided when CICS requests basic authentication. There is a TCPIPSERVICE resource definition for each port that you use for CICS Web support.

- If you require different authentication information for resources delivered using different TCPIPSERVICE definitions, specify different realms to make this clear to the end user.
- If end users use the same authentication information across your resources, you can specify the same realm on multiple TCPIPSERVICE definitions.
- If you do not specify the REALM attribute, the default realm is used. The default realm is

  ```
  realm="CICS application aaaaaaaa"
  ```

where *aaaaaaaa* is the applid of the CICS region. This is the realm that was provided by CICS during basic authentication before CICS Transaction Server for z/OS, Version 3 Release 2.

RFC 2617, *HTTP Authentication: Basic and Digest Access Authentication*, available from http://www.ietf.org/rfc/rfc2617.txt, has more detailed information about basic authentication.

Basic authentication assistance for HTTP client applications

The WEB SEND and WEB CONVERSE commands have been expanded to allow you to provide basic authentication credentials (a username and password). CICS sends these details in an Authorization header to a server that is expecting it or in response to a HTTP 401 WWW-Authenticate message. CICS converts the supplied username and password to the format that the HTTP basic authentication protocol is expecting. This allows you to supply your credentials in your usual EBCDIC character set through the WEB SEND or WEB CONVERSE command, or through the XWBAUTH user exit.

Providing credentials for basic authentication

When an HTTP 401 WWW-Authenticate message is received, your application must provide the username and password (credentials) required by the server for basic authentication. Your application can also provide these credentials without waiting for the 401 message.

1. Open a web session with the server using the WEB OPEN command, using the SESSTOKEN option. The SESSTOKEN will be returned to you when the session is opened successfully, and the session token must be used on all CICS WEB commands that relate to this connection.
2. Issue a WEB SEND command, specifying the SESSTOKEN for this connection. This WEB SEND command retrieves the realm from the server.
3. Issue a WEB RECEIVE command. The server returns a status code. Use the STATUSCODE option on the WEB RECEIVE command to check for a 401 response.
4. If the status code is 401 (the server requires authentication details), repeat your first WEB SEND request, but this time add the AUTHENTICATE(BASICAUTH) option. The XWBAUTH global user exit is called by the client application. This second WEB SEND command uses the realm received from the first WEB SEND command and the XWBAUTH exit to determine the required username and password.
5. You might prefer to specify AUTHENTICATE(BASICAUTH) in your initial WEB SEND command, instead of waiting for the 401 response. You can either:
   a. Supply your username and password in the WEB SEND command using the AUTHENTICATE(BASICAUTH) option.
   b. Invoke the XWBAUTH global user exit by specifying the AUTHENTICATE(BASICAUTH) option, but omitting your credentials. The user exit will be invoked, but the realm passed to the exit will be empty, as the realm has not yet been received from the server. The user exit must derive the required credentials from other parameters, for example, HOST and PATH.
6. If your application needs to know the realm that was sent in the 401 response, use the WEB EXTRACT command.

CICS passes the username and password credentials to the server in an Authentication header.

Changes to CICS externals

Changes to system initialization parameters

New system initialization parameters

XRES

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.

For details of the parameter, see “XRES” on page 278.

XHFS 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 file data as the static content identified by a URIMAP definition.

For details of the parameter, see “XHFS” on page 277.

For both these system initialization parameters, note that the default is YES, meaning that resource security is enabled by default. For a CICSPlex SM CMAS, both these system initialization parameters must be set to NO.

Application programming interface changes

The DOCUMENT CREATE and DOCUMENT INSERT 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 following commands can be affected:

- DOCUMENT CREATE with the TEMPLATE option
- DOCUMENT INSERT with the TEMPLATE option

Document templates used 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.

For both these commands, there is a new condition NOTAUTH:

NOTAUTH

The command has failed a resource security check. (If the NOTAUTH condition is not handled, applications that receive it may abend with code AEY7.)

Note that the EXEC CICS DOCUMENT commands reference document templates using the 48-character name of the template (as specified in the TEMPLATENAME attribute of the DOCTEMPLATE resource definition). However, security checking for the commands uses the name of the DOCTEMPLATE resource definition that corresponds to the TEMPLATENAME attribute. If resource security checking is in place, the user ID for the transaction must have READ access to this DOCTEMPLATE resource definition.

RESP2 value:

101 The user ID for the transaction does not have READ access to the DOCTEMPLATE resource definition for the document template named by the TEMPLATE option.

The QUERY SECURITY command

The QUERY SECURITY command can now be used to determine whether the user has access to the resource definitions for CICS document templates. Specify the resource type DOCTEMPLATE on the RESTYPE option of the command.

The QUERY SECURITY command cannot be used to determine whether the user has access to z/OS UNIX files. This is because access controls for UNIX files follow the system of permissions used by z/OS UNIX System Services, so they operate in a different way.

The WEB SEND and WEB EXTRACT commands

The WEB SEND and WEB EXTRACT commands have a new option (AUTHENTICATE) which allows you to specify basic authentication credentials (username and password) for client applications.

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.

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.

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

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.

The WEB EXTRACT command

The WEB EXTRACT command now includes an option to allow your client application to retrieve the realm.

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.

Changes to resource definition

TCPIPSERVICE resource

The TCPIPSERVICE resource has a new attribute:

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.

Changes to the system programming interface

CREATE TCPIPSERVICE and INQUIRE TCPIPSERVICE commands

The CREATE TCPIPSERVICE and INQUIRE TCPIPSERVICE commands each have a new option:

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.

Changes to CEMT

INQUIRE TCPIPSERVICE command

The INQUIRE TCPIPSERVICE command has a new option:

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

Changes to the CICSPlex SM programming interface

Changes to resource tables

The TCPDEF and TCPIPS resource tables have a new field for the REALM attribute on the TCPIPSERVICE resource definition.

Changes to CICSPlex/SM views and menus

The new REALM attribute on the TCPIPSERVICE definition, specifying the realm that is provided when CICS requests basic authentication, is included in the following views:

- TCP/IP service definition view in the TCP/IP service definitions (TCPDEF) view set. To access this view from the main menu, select **Administration views > Basic CICS resource administration views > Resource definitions > TCP/IP service definitions** or **Administration views > Fully functional Business Application Services (BAS) administration views > Resource definitions > TCP/IP service definitions**.
- TCP/IP service view in the TCP/IP service (TCPIPS) view set. To access this view from the main menu, select **CICS operations views > TCP/IP service operations views** > **TCP/IP service**.

Changes to global user exits

Global user exit XWBAUTH

A new global user exit, XWBAUTH, is available for use with the WEB SEND (Client) and WEB CONVERSE commands.

Two samples of XWBAUTH are also provided: DFH$WBX1 and DFH$WBX2.

DFH$WBX1 uses the DFHDDAPX XPI interface and functions for communicating with an LDAP server to retrieve credentials (username and password). Sample data for populating the LDAP server with basic authentication credentials is also provided, as member DFH$WBLD.

DFH$WBX2 executes as a WS-Trust Service Requester. It assumes that the URL of a Secure Token Service (STS) server has been stored in the user exit global workarea.by DFH$WBPI. The Secure Token Service may typically be provided by the Tivoli Federated Identity Manager (TFIM), but any equivalent STS server can be used.

Changes to the exit programming interface (XPI)

New Directory Domain XPI functions

A set of XPI functions that allow a global user exit to communicate with an LDAP server are available. The functions are part of the Directory Domain (DD). The new directory domain gate is AP (DDAP).

Changes to sample programs

New and changed HTTP client sample global user exit programs

A new sample program, DFH$WBX1 is available. DFH$WBX1 is a sample of the global user exit, XWBAUTH, that uses the DFHDDAPX XPI interface and functions for communicating with an LDAP server to retrieve credentials (username and password). Sample data for populating the LDAP server with basic authentication credentials is also provided, as member DFH$WBLD.

A new sample program, DFH$WBX2, is available. DFH$WBX2 is a sample of the global user exit, XWBAUTH, and executes as a WS-Trust Service Requester. It assumes that the URL of a Secure Token Service (STS) server has been stored in the user exit global workarea.by DFH$WBPI. The Secure Token Service may typically be provided by the Tivoli Federated Identity Manager (TFIM), but any equivalent STS server can be used.

The sample global user exit program, DFH$WBPI, has been changed to allow a security profile name (an LDAP bind profile or STS servername) to be stored in a global work area. The profile is supplied by the INITPARM system initialization parameter and enables the DFH$WBX1 program at exit XWBAUTH.

The HTTP client sample exit programs

These sample programs are for use with the Web domain exits, XWBOPEN and XWBAUTH,.

The XWBOPEN exit is invoked during processing of EXEC CICS WEB OPEN commands. XWBAUTH is called during processing of an EXEC CICS WEB SEND and EXEC CICS WEB CONVERSE commands. Both exits are used in making HTTP client requests from CICS as an HTTP client, which is a facility provided by CICS Web support.

The following sample exit programs are shipped in the CICS sample library, SDFHSAMP:

- DFH$WBPI, described in “DFH$WBPI”
- DFH$WBEX, described in “DFH$WBEX” on page 102
- DFH$WBX1, described in “DFH$WBX1” on page 102
- DFH$WBX2, described in “DFH$WBX2” on page 103
- DFH$WBGA, a copybook to map the global work area used by the DFH$WBPI, DFH$WBX1, DFH$WBX2, and DFH$WBEX samples.

DFH$WBPI

This program, whose purpose is to initialize the supplied Web-related global user exits, is specified in the PLTPI and is invoked during the CICS post-initialization phase. It is specified with the **INITPARM** system initialization parameter as follows:

```
INITPARM=(DFH$WBPI='PROXY=proxyurl,LDAPBIND=profilename,STS=sts-server-url')
```

where

PROXY=proxyurl
: This optional keyword stores the URL (in the form http://proxyserver) of a proxy server into the Web global work area, then enables the supplied DFH$WBEX sample program as the XWBOPEN global user exit.

LDAPBIND=profilename
: This optional keyword stores the name of an LDAP bind profile into the Web global work area, then enables the supplied DFH$WBX1 sample program as the XWBAUTH global user exit.

  Note that you cannot specify both LDAPBIND and STS. To do so causes DFH$WBPI to abend with code WBPI. Message DFHSI1580D is issued, which may cause CICS to be terminated.

STS=sts-server-url
: This optional keyword stores the URL (usually in the form https://sts-server) of a Secure Token Service into the Web global work area, then enables the supplied DFH$WBX2 sample program as the XWBAUTH global user exit.

  Note that you cannot specify both STS and LDAPBIND. To do so causes DFH$WBPI to abend with code WBPI. Message DFHSI1580D is issued, which may cause CICS to be terminated.

Note that the total length of the INITPARM quoted text cannot exceed 60 characters.

DFH$WBEX

This sample global user exit program is designed to check the host name specified on the EXEC CICS WEB OPEN command, and make any host name starting with `www` use a proxy server if a proxy server name is specified in the global work area.

If all the requests from your CICS system should use a single proxy server, you can use the proxy server name from the **INITPARM** system initialization parameter, that DFH$WBPI used to initialize the global work area.

- The proxy name must be specified as:

  ```
  INITPARM=(DFH$WBPI='PROXY=proxyurl')
  ```

where proxyurl is the URL if a proxy server. If you use a number of proxy servers or want to apply a security policy to different host names, you can load or build a table that matches host names to appropriate proxy servers or marks them as barred, which can then be used as a look-up table during processing of the EXEC CICS WEB OPEN command.

DFH$WBX1

This sample global user exit program has the following functions:

- If a GLUE global workarea is provided and it contains a non-zero LDAP connection token, it uses that token in subsequent SEARCH requests.
- If the exit is called at the XSTERM (system termination) exit point, it terminates the LDAP connection by invoking the DFHDDAP UNBIND_LDAP function. Otherwise, it obtains a connection token by issuing DFHDDAP BIND_LDAP and stores it in the global workarea. The LDAPBIND profile specified in the INITPARM parameter for DFH$WBPI is used to obtain LDAP credentials.
- Composes a distinguished name in the following format: `racfcid=uuuuuuuu, ibm-httprealm=rrrrrrrr, labeledURI=xxxxxxxx, cn=BasicAuth` where:

racfcid=uuuuuuuu
is the current userid, obtained from UEPUSER

ibm-httprealm=rrrrrrrr
is the HTTP 401 realm, obtained from UEPREALM (if this exists)

labeledURI=xxxxxxxx
is the target URL, obtained by concatenating “http://” with the hostname from UEPHOST and the path from UEPPATH

cn=BasicAuth
is an arbitrary suffix that is configured into the LDAP server for the purpose of storing Basic Authentication credentials.

- Issues DFHDDAP SEARCH_LDAP with this distinguished name
- If the SEARCH_LDAP fails, DFH$WBX1 removes the REALM parameter from the distinguished name and repeats the search. If the search fails again, DFH$WBX1 removes the **UID** parameter from the distinguished name and repeats the search. If the search fails for the third time, DFH$WBX1 returns from the exit with return code UERCERR.
- If the search was successful, issue DFHDDAP START_BROWSE_RESULTS
- Obtains the target **username** credential by obtaining the value of the **UID** attribute with DFHDDAP GET_ATTRIBUTE_VALUE. This is set into the response area provided by UEPUSNM.
- Obtains the target **password** credential by obtaining the value of the **UserPassword** attribute with DFHDDAP GET_ATTRIBUTE_VALUE. This is set into the response area provided by UEPPSWD.
- Releases the browse storage by issuing DFHDDAP END_BROWSE_RESULTS
- If the bind token was not stored in the global workarea, terminate the LDAP session by issuing DFHDDAP UNBIND_LDAP
- If all is successful, DFHWBX1 returns from the exit with return code UERCNORM.

DFH$WBX2

This sample global user exit program has the following functions:

- Obtains the destination HTTP host from UEPHOST/UEPHOSTL and the destination HTTP path from UEPPATH/UEPPATHL, and uses them to construct the URL of the HTTP server for which the basic authentication credentials are required, as follows: `http://hostname/pathname`.
- If a realm exists (that is, if UEPREALML is non-zero), DFH$WBX2 appends the realm from UEPREALM to the URL created above, separated by a number sign (#) to make it look like a URL fragment identifier, as follows: `http://hostname/pathname#realm`. If necessary, the realm is URL-encoded.
- Stores the URL in the DFHWS-SERVICEURI container in the DFHWSTC-V1 channel.
- Stores the URL of the Security Token Service (STS), obtained from the global work area, in the DFHWS-STSURI container in the DFHWSTC-V1 channel.
- Stores architecturally appropriate URIs into the DFHWS-STSACTION and DFHWS-TOKENTYPE containers in the DFHWSTC-V1 channel.
- Constructs a username token from the caller’s userid passed in UEPUSER, and store it in the DFHWS-IDTOKEN container in the DFHWSTC-V1 channel.
-

Note: It is not necessary to specify a pipeline in the DFHWS-PIPELINE container. The pipeline is dynamically created by DFHPIRT when CHANNEL(DFHWSTC-V1) is specified on the command.

- Links to DFHPIRT, specifying CHANNEL(DFHWSTC-V1), after constructing all the required containers. This sends the request to the STS and receives the response into the DFHWS-RESTOKEN container.
- If the LINK was successful, DFHWBX2 recovers the response from the DFHWS-RESTOKEN container, which contains an username token in XML format.
- Extracts the username and password from this token, and returns them as responses from the user exit in UEPUSNM/UEPUSNML and UEPPSWD/UEPPSWDL. Returns from the user exit with return code UERCNORM.
- If the LINK was unsuccessful, or if a SOAP fault response is returned, DFH$WBX2 returns from the exit with return code UERCERR.

The above implementation assumes that the STS server is configured to respond with an appropriate username token when presented with the URI formatted by DFH$WBX2 in the `AppliesTo` element of the STS issue request.

Security

Security for CICS document templates

The XRES system initialization parameter activates security checking for CICS document templates. This system initialization parameter is set to YES by default. The default resource class name is RCICSRES, and the default grouping class name is WCICSRES.

The following commands can be affected:

- DOCUMENT CREATE with the TEMPLATE option
- DOCUMENT INSERT with the TEMPLATE option
- CREATE DOCTEMPLATE
- DISCARD DOCTEMPLATE
- INQUIRE DOCTEMPLATE
- SET DOCTEMPLATE

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. For the DOCUMENT CREATE, CREATE DOCTEMPLATE, and DISCARD DOCTEMPLATE commands, ALTER access is required. For all other commands, READ access is required.

To implement security for CICS document templates, you need to define RACF profiles for your DOCTEMPLATE resource definitions, specify appropriate system initialization parameters, and activate resource security checking for the transactions that access the CICS document templates. The *CICS RACF Security Guide* explains how to do this.

Security for z/OS UNIX files

The XHFS system initialization parameter activates access control for z/OS UNIX files. This system initialization parameter is set to YES by default. Access control for z/OS UNIX files is not affected by the RESSEC attribute in the TRANSACTION resource definition of the transactions that access the files.

Access permissions for these files are specified in z/OS UNIX System Services, so you do not need to define RACF profiles for individual z/OS UNIX files. RACF is used to manage user profiles, groups, and access control lists (ACLs). The use of ACLs is the recommended solution for giving permissions to Web clients' user IDs. If you are using ACLs to control access to z/OS UNIX files, you need to activate the FSSEC class for these to be checked.

No CICS commands are affected by security for z/OS UNIX files. The files can only be referenced by 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.

When access control is specified for z/OS UNIX files, the authenticated user IDs used by Web clients need to have **read** access to the files and to the directories containing them. Authenticated user IDs will already have a user profile defined in your security manager. They will need to have a suitable z/OS UNIX user identifier (UID), and connect to a RACF group with a z/OS UNIX group identifier (GID) that has permissions for the files and the directories containing them.

The CICS region user ID must always have a minimum of **read** access to all z/OS UNIX files that it uses for CICS Web support, and to the directories containing them.

The *CICS RACF Security Guide* explains how to implement access controls for z/OS UNIX files.

CICSPlex SM security

CICSPlex SM honors the new system initialization parameters XRES and XHFS, and includes or excludes the designated resources from security checking. As with other system initialization parameters relating to resource security, for each MAS, you can specify YES, NO, or a class name for XRES. For XHFS, which does not use class names, you can specify YES or NO. However, for the CMAS, you *must* specify NO for XRES and XHFS.

Chapter 13. Improved management of CICS documents and document templates

CICS now allows you to release storage by deleting documents when they are no longer required in a transaction. New caching facilities improve the performance of applications that use CICS document templates.

- CICS now allows you to delete documents that are no longer required during a transaction.
- CICS now caches a copy of most types of document template. When applications reference the template, they use the cached copy, improving performance.
- You can use the SET DOCTEMPLATE NEWCOPY command to refresh a document template. For cached document templates, the command refreshes the cached copy of the document template. For document templates produced from programs and exit programs, the command phases in a new copy of the program.
- CICS now collects statistics for document templates with the new CICS statistics type DOCTEMPLATE. The statistics show the number of times each document template is referenced, and the number of times a cached copy was made, refreshed, used and deleted.

Deleting a document

You can use the DOCUMENT DELETE command to delete documents that are no longer required during a transaction. On execution of the command, the storage allocated to the document is released immediately. The DOCSTATUS(DOCDELETE) option of the WEB CONVERSE, WEB SEND (Client) and WEB SEND (Server) commands also allows document deletion.

DOCUMENT DELETE, WEB CONVERSE, WEB SEND (Client) and WEB SEND (Server) all use the DOCTOKEN to specify the 16–byte binary token of the document. The document token is returned when you create a document using the EXEC CICS DOCUMENT API commands.

To delete a document using the DOCUMENT DELETE command:

1. Specify the DOCTOKEN of the document you wish to delete. For example:
   ```
   EXEC CICS DOCUMENT DELETE
             DOCTOKEN(MYDOC)
   ```
2. The document is deleted from the document handling domain and storage is released immediately. If ACTION(EVENTUAL) is specified in the command, the Web domain retains a copy of the document.

WEB CONVERSE, WEB SEND (Client) and WEB SEND (Server) allow you to delete a document by specifying the DOCSTATUS(DOCDELETE) option. This option allows the application to indicate that it no longer requires the document once it has issued the CONVERSE or SEND command. The document is deleted from the document handling and Web domains on completion of the WEB SEND command, and storage is released immediately.

If you issue a WEB SEND, specifying DOCSTATUS(NODOCDELETE) and ACTION(EVENTUAL) in the command, it is possible to retrieve the document using the WEB RETRIEVE command. Using the DOCSTATUS(DOCDELETE) option or using the ACTION(IMMEDIATE) option will remove the document permanently from

Web storage, and the document cannot be retrieved. The *CICS Application Programming Reference* provides more information on the limitations of document retrieval after deletion of documents.

Caching and refreshing of document templates

To improve performance, the CICS document handler caches a copy of most document templates. When applications reference the template, they use the cached copy, improving performance. You can refresh the cached copy at any time if the document template changes. You can also phase in a new copy of programs and exit programs that are defined as document templates.

CICS always caches a copy of the following types of document template:

- Templates in a partitioned data set
- Templates in a CICS file
- Templates in a z/OS UNIX System Services file
- Templates in a temporary storage queue
- Templates in a transient data queue

When one of these types of document template is installed individually while CICS is running, it is read into the CICS document handler's storage. Requests from applications to access the document template receive the cached copy of the template, so CICS does not need to access the location where the document template is stored each time. Document templates that are installed during CICS startup are not cached at that time; each of these document templates is cached when it is referenced for the first time by an application.

If you make changes to a document template that has been cached, you can refresh the cached copy of the document template using the CEMT or EXEC CICS SET DOCTEMPLATE NEWCOPY command. (Note that with the SET DOCTEMPLATE command, which is not part of the EXEC CICS DOCUMENT API, you need to specify the name of the DOCTEMPLATE resource definition which defines the document template, rather than the 48-character name of the template.)

For the types of document template listed above, the SET DOCTEMPLATE NEWCOPY command deletes the copy of the document template which is currently cached by the CICS document handler, and replaces it with a copy read from the location where the document template is stored. (For templates in a partitioned data set, CICS first performs a BLDL (build list) to obtain the most current directory information, and then rereads the member.) When a new cached copy has been created, subsequent requests to use the document template use the new copy. The new copy will be used by later requests within the same task, as well as requests in other tasks.

If the CICS system becomes short on storage, the document handler deletes some of the cached copies of document templates to attempt to relieve the storage constraint. The document templates to be deleted are selected in order of size, largest first, taking into account the time since the cached copy was created (so that newly created copies are not released immediately).

If the CICS system is restarted with a warm start, the document templates that were previously cached are not reloaded. The cache is repopulated as each document template is referenced for the first time by an application.

The CICS statistics collected for document templates show the number of times each document template is referenced, and the number of times a cached copy was made, refreshed, used and deleted.

Templates in CICS programs

Document templates retrieved from CICS programs are never cached by the document handler, because programs are already cached elsewhere in CICS.

For this type of document template, you can use the SET DOCTEMPLATE NEWCOPY command to phase in a new copy of the program. The command is equivalent to SET PROGRAM PHASEIN for the specified program. Subsequent requests to use the document template use the new copy, including later requests within the same task.

Templates in exit programs

For document templates generated by an exit program, the exit program specifies (in its exit parameter list) whether or not a copy of the document template should be cached by the document handler. The default is that the document template is not cached. Templates that change dynamically should not be cached, but if the template does not change, caching is suitable as it improves the performance of requests. If the exit program does specify caching, the cached copy is made when the document template is referenced for the first time by an application.

For this type of document template, you can use the SET DOCTEMPLATE NEWCOPY command to phase in a new copy of the exit program. The command is equivalent to SET PROGRAM PHASEIN for the specified exit program. When you issue the command, CICS deletes any cached copy of the document template, phases in the new copy of the program, and creates a new cached copy of the document template if the exit program specifies caching. The refreshed exit program can specify a different setting for whether or not caching should take place, and CICS honors the change.

Changes to CICS externals

Application programming interface changes

New command: DOCUMENT DELETE

The DOCUMENT DELETE command enables you to delete documents that are no longer required during a transaction. The command allows the application to request deletion of a document and all storage related to the document. On execution of this command, the storage allocated to the document is released immediately. If the DOCUMENT DELETE command is not invoked, the document exists until the application ends.

For details of the command, see “DOCUMENT DELETE” on page 279.

The WEB CONVERSE and WEB SEND (Server and Client) commands

A new DOCSTATUS option allows you to specify if you want to delete a document during a transaction. This option applies to all commands where a DOCTOKEN is specified, as this indicates that the command is processing a document. The WEB SEND (Client) command provides more information on the DOCSTATUS option.

Changes to the behavior of the WEB RETRIEVE command

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.

Specifying caching for document templates in exit programs

CICS only caches a copy of document templates in exit programs if the exit program itself specifies caching in its exit parameter list.

A new field is added to the communication area to enable you to specify whether or not the document template is suitable for caching:

dhtx_cache_response

Use the characters '1' (cache) or '0' (do not cache) to specify whether or not the output from the exit program should be cached by the CICS document handler. The value of `dhtx_cache_response` is initialized to '0', so the default action is not to cache the response of the exit program, unless the exit changes this value.

When a document template is cached, subsequent requests receive the cached copy. The exit program is not called again as long as the cached copy is available, until the EXEC CICS SET DOCTEMPLATE NEWCOPY command is issued to refresh the exit program and the cached copy. A refreshed exit program can specify a different value for `dhtx_cache_response`, and CICS honors the change.

Templates that change dynamically should not be cached, but if the template does not change, caching is suitable as it improves the performance of requests.

The communication area for an exit program which supplies a document template is mapped by the following copybooks:

- DFHDHTXD (Assembler)
- DFHDHTXH (C)
- DFHDHTXL (PL/I)
- DFHDHTXO (COBOL)

Changes to the system programming interface

Note: Changes to the EXTRACT STATISTICS and PERFORM STATISTICS commands are described in “Changes to statistics” on page 112

New SPI commands

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.

For details of the command, see “SET DOCTEMPLATE” on page 314The SET DOCTEMPLATE command.

INQUIRE DOCTEMPLATE command

The INQUIRE DOCTEMPLATE command has a new 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.

Changes to CEMT

Note: Changes to the PERFORM STATISTICS command are described in "Changes to statistics" on page 112

New CEMT commands

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.

For details of the command, see "CEMT SET DOCTEMPLATE" on page 325.

INQUIRE DOCTEMPLATE command

The INQUIRE DOCTEMPLATE command has a new 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.

Changes to the CICSPlex SM programming interface

Changes to resource tables

The DOCTEMP resource table has new fields for the new document template statistics, and a new Newcopy action to refresh the cached copy of the document template.

The TASK and HTASK resource tables have a new field for the number of document delete requests made by a task.

Changes to CICSPlex SM views and menus

The ″Document template″ view displays the new document template statistics, including the amount of storage required for a cached copy of the document template, and information about usage of the document template and of cached copies. The view also has a new Newcopy action button, which enables you to refresh the cached copy of the document template after you have made changes to it. To access this view from the main menu, select **CICS operations views > Document template operations views > Document template**.

The ″Active tasks″ detailed view for request count information now displays the number of document delete requests for the task. To access this view from the main menu, select **CICS operations views > Task operations views > Active tasks**.

The ″Completed tasks Request counts″ detailed view for request count information now displays the number of document delete requests for the task. To access this view from the main menu, select **History views > Completed tasks > task item > Request counts**.

Changes to statistics

CICS now collects statistics on the usage of document templates. The statistics include information about:

- The amount of storage required for a cached copy of the document template. (This information is not provided for templates in a CICS program, which are never cached, or for templates in an exit program if they are not specified for caching.)
- The number of times the document template was referenced.
- The number of times a cached copy of the document template was placed into the cache, refreshed using the SET DOCTEMPLATE NEWCOPY command, used by an application, and deleted.

The statistics are recorded by specifying the DOCTEMPLATE option on the CEMT PERFORM STATISTICS and EXEC CICS PERFORM STATISTICS RECORD commands, and retrieved online using the EXTRACT STATISTICS command specifying RESTYPE(DOCTEMPLATE). They are mapped by the DFHDHDDS DSECT.

Document template statistics can be included in reports produced by the statistics reporting utility DFHSTUP, using the DOCTEMPLATE resource type, and are included in reports generated by the sample statistics program DFH0STAT.

Note: In previous releases, the sample statistics program DFH0STAT produced a report listing the document templates installed in the CICS region, but the report did not provide any information about the usage of the document templates.

Changes to monitoring

Performance data group DFHDOCH

Group DFHDOCH contains a new field:

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

Chapter 14. Optimized support for data conversion

The CICS internal CCSID conversion interface, used in container support and the Web interface, now provides optimized support when all characters in the input data are within a set that can be converted directly using a single-byte to single-byte translate operation. When the full conversion process uses the z/OS unicode conversion service, this optimization can result in a significant reduction in CPU time for conversion processing. For example, tests have shown that when SOAP messages contain data only within the ASCII single-byte subset of UTF-8, the optimized conversion processing between UTF-8 and EBCDIC shows a reduction of over 15% in transaction CPU time for processing a 32 KB input message, and over 30% for generating a 32 KB output message.

The first time a conversion between a specific pair of Coded Character Set Identifiers (CCSIDs) is attempted, CICS builds a test table that indicates which single-byte input codes can be translated directly, and a translate table giving the single-byte output codes for the valid single-byte input codes. At the same time, CICS also builds the corresponding pair of tables for the reverse conversion. On each conversion call for that pair of CCSIDs, CICS uses the appropriate test table (via the translate and test instruction) to check whether the input data can be translated with a simple translate. If so, CICS copies the input data to the output buffer and translates it rather than performing the full conversion process.

The test table is also used to optimize the logic to determine the length required for the converted data, for example, when processing the GET CONTAINER command with the NODATA option, where the specified or assumed CCSID differs from the CCSID in which the container was stored. If the test shows that all input characters are within the set eligible for single-byte translation, the output length is assumed to be the same as the input length, avoiding the need to perform a test conversion to determine the actual length.

Part 3. CICS service management: configuration enhancements and constraint relief

CICS Transaction Server for z/OS, Version 3 Release 2 delivers a set of capabilities which provide customer value by enabling business flexibility through IT simplification. These capabilities are represented in three themes:

- application connectivity
- application reuse
- service management

The capabilities represented by the *service management* theme enable you to effectively manage large runtime configurations using modern user interfaces, so that you can meet demanding service level and IT governance objectives. CICS Transaction Server for z/OS, Version 3 Release 2 provides functions that simplify product configuration and management, and that relieve several architectural constraints.

Chapter 15. Dynamic program library management

CICS TS 3.2 introduces dynamic program LIBRARY resources providing the ability to enable the data sets from which program artifacts will be loaded to be defined dynamically without it being necessary to restart the CICS region. This is in addition to the existing means of defining the data sets statically in the DFHRPL concatenation.

CICS TS 3.2 introduces a new resource type of LIBRARY, representing a partitioned data set (PDS/PDSE) or sequence of concatenated partitioned data sets (PDS/PDSEs), containing program entities that make up an application or group of applications. DFHRPL is a special example of a LIBRARY that cannot be altered in a running CICS system.

Traditionally, data sets containing program artifacts have been defined to CICS in DFHRPL in the startup JCL. To change any of the data sets it is necessary to edit the JCL and restart CICS. There is often a need to change the data sets used to load programs to do the following:

- Apply emergency fixes to a program while CICS is running.

  If the fixed program is put in a data set earlier in the concatenation, it can be loaded in place of the broken program. This can be done today, but you typically have to include a special data set for fixes in the DFHRPL concatenation, and move the fixed program into this data set. It gets difficult to know what is in the special fix data set.
- Add new programs or new versions of programs while CICS is running.

  You are more likely to do this currently in development or test systems than in production, as you might have restrictions on changes allowed in production, however, enabling this in production will provide a benefit towards continuous operations.

Having to restart CICS is frustrating, and impacts the continuous availability of the CICS system. This new feature allows for better organization and management of applications within a CICS system while maintaining continuous availability.

For further information on using dynamic program LIBRARY resources, see the *CICS Application Programming Guide*.

Terminology

This topic contains the new and changed terminology used by dynamic program LIBRARY management.

library search order
The order in which the list of LIBRARY resources installed in CICS is searched for programs to load.

library ranking
A number assigned to a LIBRARY resource that indicates its position in the overall search order relative to other LIBRARY resources. A LIBRARY with a smaller ranking number comes before a LIBRARY with a larger ranking number.

Changes to CICS externals

Changes to resource definition

To support dynamic program libraries, there is a new resource definition.

New LIBRARY resource

Support for dynamic program libraries introduces the LIBRARY resource. A LIBRARY represents a PDS/PDSE or sequence of concatenated PDS/PDSEs containing program entities that together make up an application or group of applications. DFHRPL is a special example of a LIBRARY which cannot be altered in a running CICS system.

For an overview of the LIBRARY resource, see LIBRARY resource definitions in the *CICS Resource Definition Guide*.

Changes to the system programming interface

Note: Changes to the EXTRACT STATISTICS , PERFORM STATISTICS and PERFORM STATISTICS RECORD commands are described in "Changes to statistics" on page 122

New SPI commands

CREATE LIBRARY

Create a LIBRARY resource in the local CICS region.

For details of the command, see CREATE LIBRARY.

DISCARD LIBRARY

Remove a specified LIBRARY from the running CICS system.

For details of the command, see DISCARD LIBRARY.

INQUIRE LIBRARY

Retrieve information about a LIBRARY.

For details of the command, see INQUIRE LIBRARY.

SET LIBRARY

Change the attributes of a LIBRARY resource.

For details of the command, see INQUIRE LIBRARY.

Changes to CEMT

Note: Changes to the PERFORM STATISTICS command are described in "Changes to statistics" on page 122

New CEMT commands

INQUIRE LIBRARY

Retrieve information about LIBRARY resources.

For details of the command, see CEMT INQUIRE LIBRARY.

SET LIBRARY

Change the attributes of LIBRARY resource.

For details of the command, see CEMT SET LIBRARY.

DISCARD command

The DISCARD command has a new option:

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 PROGRAM command

The INQUIRE PROGRAM command has been enhanced to show the LIBRARY and data set from which the loaded copy of the program was obtained.

Changes to the CICSPlex SM programming interface

New resource tables

The LIBRARY resource introduces 4 new resource tables:

LIBRARY
A CICS Resource that describes a LIBRARY in an active system being managed by CICSPlex® SM.

LIBDSN
A CICS Resource that describes the data sets that make up a LIBRARY.

LIBDEF
A CICS definition table that describes a LIBRARY.

LIBINGRP
A CPSM definition that describes the membership of a LIBRARY definition in a resource group.

Changes to resource tables

There are changes to the following resource tables:

PROGRAM resource table
The following new fields have been added:

LIBRARY
The name of the LIBRARY resource from which the program was loaded.

LIBRARYDSN
The name of the data set from which the program was loaded.

CICSRGN resource table
The following new fields have been added:

LDGLBSOU
The number of LIBRARY search order updates.

LDGLWSOU
The number of waits for a program load due to LIBRARY search order updates.

LDGLSORT
The amount of time spent updating the LIBRARY search order.

CPSM LOADER resource table
The following fields have been added:

LDGLBSOU
The number of LIBRARY search order updates.

LDGLWSOU
The number of waits for a program load due to LIBRARY search order updates.

LDGLSORT
The amount of time spent updating the LIBRARY search order.

Changes to CICSPlex SM views

New CICSPlex SM operational views

There are 2 new view sets, “LIBRARY” and “LIBRARY data set names”.

LIBRARY
The “LIBRARY” data views display information about the dynamic program LIBRARY resources including the static DFHRPL program LIBRARY

Table 4. Views in the supplied **LIBRARY** *(LIBRARY) view set*

View	Notes
LIBRARY EYUSTARTLIBRARY.DISABLE	Disable the LIBRARY. When disabled, a LIBRARY is not included in the LIBRARY search order. The data sets in this LIBRARY concatenation will not be searched for program artifacts to load. **Note:** The LIBRARY named DFHRPL cannot be disabled or discarded.
LIBRARY EYUSTARTLIBRARY.DISCARD	Discard a LIBRARY from the CICS system where it is installed. A LIBRARY must be disabled before it can be discarded. **Note:** The LIBRARY named DFHRPL cannot be disabled or discarded.
LIBRARY EYUSTARTLIBRARY.TABULAR	Tabular information about currently installed LIBRARYs.
LIBRARY EYUSTARTLIBRARY.DETAILED	Detailed information about a selected LIBRARY.
LIBRARY EYUSTARTLIBRARY.SET	Set LIBRARY attributes according to new values specified in input fields.
LIBRARY EYUSTARTLIBRARY.ENABLE	Enable the LIBRARY. When enabled, a LIBRARY is included in the LIBRARY search order. The data sets in this LIBRARY concatenation will be searched for program artifacts to load. **Note:** If an ENABLE fails, the LIBRARY remains disabled.

LIBRARY data set names

The “LIBRARY data set names” data views display information about the data sets associated with LIBRARY resources.

Table 5. Views in the supplied **LIBRARY data set names** *(LIBDSN) view set*

View	Notes
LIBRARY data set names EYUSTARTLIBDSN.TABULAR	Tabular information about currently installed LIBRARY data set names.
LIBRARY data set names EYUSTARTLIBDSN.DETAILED	Detailed information about a selected LIBRARY data set name.

New CICSPlex SM BAS views

There is a new view set, “LIBRARY definitions”.

LIBRARY definitions (LIBDEF)

The LIBRARY definitions (LIBDEF) views display information about the dynamic program LIBRARY definitions

Table 6. Views in the supplied **LIBRARY definitions** *(LIBDEF) view set*

View	Notes
LIBRARY definitions EYUSTARTLIBDEF.INSTALL	Install a LIBRARY definition in an active system.
LIBRARY definitions EYUSTARTLIBDEF.REMOVE	Remove a LIBRARY definition from the data repository.
LIBRARY definitions EYUSTARTLIBDEF.TABULAR	Tabular information about all LIBRARY definitions for the current context.
LIBRARY definitions EYUSTARTLIBDEF.DETAILED	Detailed information about a selected LIBRARY definition.
LIBRARY definitions EYUSTARTLIBDEF.ADDTOGRP	Add one or more LIBRARY definitions to a resource group.
LIBRARY definitions EYUSTARTLIBDEF.CREATE	Create a LIBRARY definition and add it to the data repository.

Changes to the exit programming interface (XPI)

Changes to existing XPI

Changes to the INQUIRE_PROGRAM and INQUIRE_CURRENT_PROGRAM XPI function now provide the name of the LIBRARY and data set from which the program was loaded.

See The INQUIRE_PROGRAM call or The INQUIRE_CURRENT_PROGRAM call for further information.

Changes to statistics

CICS now collects statistics on the usage of LIBRARY resources.

The statistics are recorded by specifying the LIBRARY option on the CEMT PERFORM STATISTICS and EXEC CICS PERFORM STATISTICS RECORD commands, and retrieved online using the EXTRACT STATISTICS command specifying RESTYPE(LIBRARY).

Chapter 16. Support for the z/OS Enterprise Workload Manager

Currently, CICS supports the z/OS Workload Manager (WLM). In CICS TS for z/OS, Version 3.2, support is added for the z/OS Enterprise Workload Manager (EWLM).

EWLM's key feature is that it makes possible end-to-end workload monitoring in distributed environments that contain multiple, interacting, server products.

Introduction to the Enterprise Workload Manager

The Enterprise Workload Manager (EWLM) is IBM's implementation of the **Application Response Measurement** (ARM) standard. EWLM extends the capabilities of the **z/OS Workload Manager**, which operates on z/OS, to all members of the IBM eServer family.

Note: This topic is a high-level overview of EWLM. For detailed information about EWLM, see the document *IBM Systems Virtualization Engine™ Enterprise Workload Management, Version 2 Release 1*, which is available from the IBM Systems Software Information Center at `http://publib.boulder.ibm.com/infocenter/eserver/v1r2/index.jsp?topic=/ewlminfo/eicaagettingstarted.htm`.

Because it runs on many types of server, EWLM can be used for end-to-end workload monitoring in distributed environments that contain multiple, interacting, server products.

Terminology

This topic contains the terminology used in describing CICS support for the z/OS Enterprise Workload Manager. For definitive explanations of all EWLM terms, see *IBM Systems Virtualization Engine Enterprise Workload Manager, Version 2 Release 1*.

Application Response Measurement
A workload reporting standard owned and defined by the Open Group. For information about the Open Group, see `www.opengroup.org`.

ARM See **Application Response Measurement**.

ARM correlator
Contextual information associated with an **ARM transaction**, used to track the flow of work from application to application. A correlator is a byte array used to carry information from one ARM sub-transaction to another and to associate related work in different systems. See also **parent correlator**, **child correlator**, and **edge correlator**.

ARM transaction
An entity that represents the overall work flow. An ARM transaction may consist of multiple ARM sub-transactions, each of which represents that part of the overall work done by one system or middleware application. A CICS transaction instance may be part of an ARM transaction.

ARM workload manager
A workload manager that implements the **Application Response Measurement** standard. IBM's implementation of the ARM standard is the **EWLM**.

child correlator
The **correlator** of a child **ARM transaction**. (The flow of a work request through a series of applications can be viewed as a call tree of parent and child ARM transactions. Each transaction in the tree may, or may not, have a unique correlator.)

correlator
See **ARM correlator**.

edge correlator
The first **correlator** assigned to an ARM transaction instance by an ARM workload manager. An ARM transaction instance being processed by a middleware **hop** is considered to be a **transaction edge** when no parent **ARM correlator** is available. In these circumstances the ARM transaction is given its first correlator (the edge correlator), by the ARM workload manager.

An edge correlator may also be assigned by the **Enterprise Workload Manager** if a parent correlator is passed to it that is invalid or unrecognizable. (The parent correlator may, for example, have originated in a different ARM management domain.)

Enterprise Workload Manager (EWLM)
IBM's implementation of the **Application Response Measurement** (ARM) standard. EWLM extends the capabilities of the **z/OS Workload Manager**, which operates on z/OS, to all members of the IBM eServer family. Its key feature is that it makes possible end-to-end workload monitoring in distributed environments that contain multiple, interacting, server products.

EWLM
See **Enterprise Workload Manager**.

EWLM Control Center
EWLM module that provides various user interfaces, including one accessible from a web browser, to the **EWLM domain manager**.

EWLM domain manager
The system that manages the set of heterogeneous distributed systems that constitute the **EWLM management domain**.

EWLM management domain
The set of heterogeneous distributed systems managed by the **EWLM domain manager**.

hop The flow of a work request from one system or middleware application (such as CICS) to another. Hop 0 is the place where an ARM transaction is first assigned a **correlator** for the work request. Hop 0 occurs at the **transaction edge**.

parent correlator
The **correlator** of a parent **ARM transaction** that is passed to its child ARM transactions. (The flow of a work request through a series of applications can be viewed as a call tree of parent and child ARM transactions. Each transaction in the tree has its own correlator.)

transaction edge
An ARM transaction instance for which no **parent correlator** is available.

WLM See **z/OS Workload Manager**.

z/OS Workload Manager
A workload balancing, management, and reporting system for z/OS

applications such as CICS. WLM uses a goal-oriented approach to provide automatic, dynamic, balancing of system resources (central processors and storage) across a z/OS sysplex.

The structure of the EWLM management domain

As shown in the following figure, the **EWLM domain manager** manages a set of heterogeneous distributed systems known as the **EWLM management domain**. The domain manager maintains a global view of performance and topology information that can be accessed by an operator or by other system management products. The management domain is administered from the **EWLM Control Center**, which provides various user interfaces, including one accessible from a web browser, to the EWLM domain manager. The management domain is subject to a user-created domain policy that is distributed from the domain manager to the participating systems.

An EWLM managed server component runs on each operating system instance in the management domain. This is responsible for collecting processing data about ARM transactions. This task requires assistance from the middleware (such as CICS) running on the systems, and from the local operating system or workload manager.

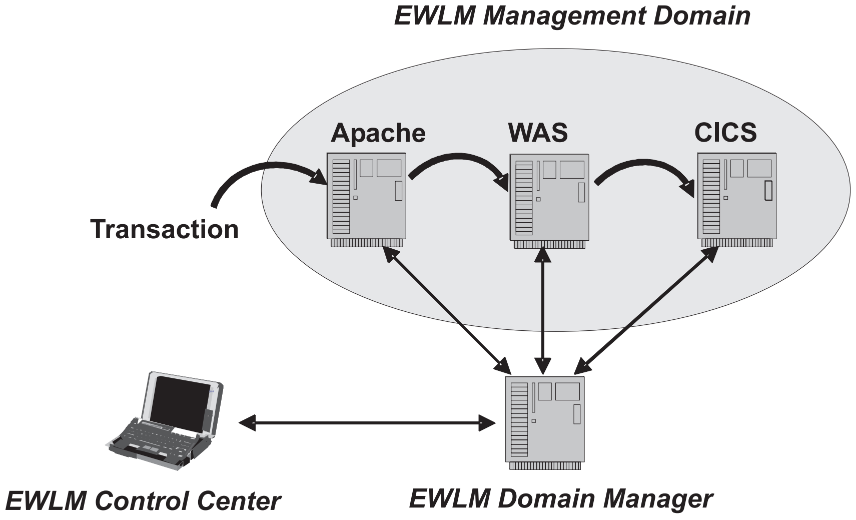


Figure 1. An Enterprise Workload Manager Domain

So that the EWLM managed server knows what type of work is running on the operating system and what goals are associated with each ARM transaction, middleware such as transaction servers must indicate when the processing of an ARM transaction starts and ends and what kind of transaction it is. The transaction servers and other middleware also have to flow an **ARM correlator** with the transaction, so that EWLM can detect where an ARM transaction is processed and how it flows through the management domain. This instrumentation is based on the Open Group ARM standard. For example, in the above figure the Apache web server flows the correlator associated with the ARM transaction to the WebSphere Application Server (WAS), and WAS flows the correlator to CICS.

Typically, more than one ARM correlator is associated with a work flow. If this is the case, the correlators have a hierarchical, parent - child, relationship. The first correlator is assigned by EWLM on the first eServer, at **hop 0** or the **transaction**

edge, and represents the whole ARM transaction. As the work flows between transaction servers or middleware, **child correlators** may be assigned to those parts of the whole ARM transaction that are processed on particular servers or by, for example, particular CICS transactions.

A CICS example

The following figure shows a work flow that spans two servers, WebSphere Application Server and CICS, being monitored by the Enterprise Workload Manager.

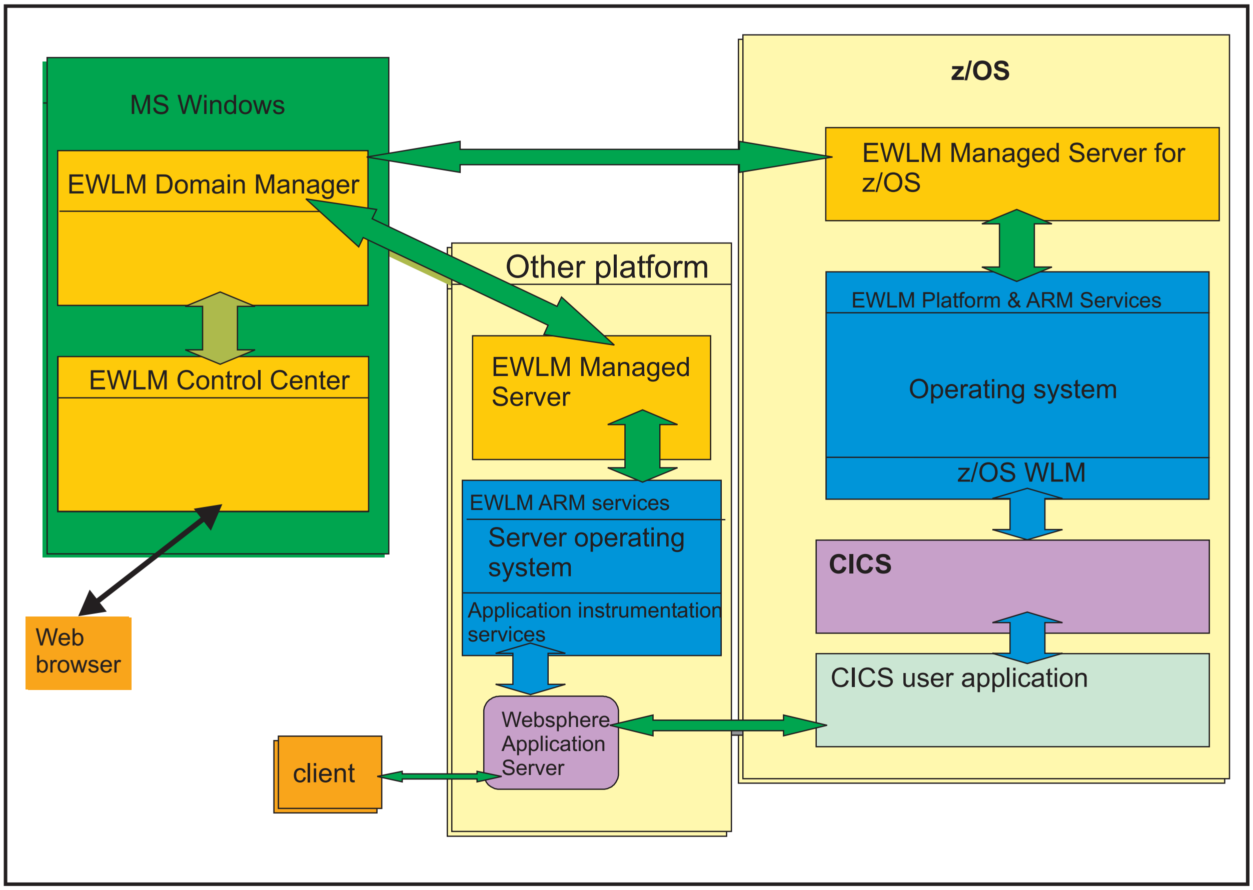


Figure 2. A work flow that spans two servers, one of them CICS, being monitored by the Enterprise Workload Manager

The picture shows the EWLM domain manager and Control Center running on a Microsoft® Windows® host. The Control Center is accessed through a web browser interface.

A single z/OS image can host only one EWLM managed server, which means that a single z/OS image can be part of only one EWLM management domain.

In the above picture, the CICS user application running in z/OS interacts with a program running in WebSphere Application Server on a non-z/OS platform. As it does so, the CICS monitoring domain calls the WLM. Note that *a CICS application does not make WLM, EWLM, or ARM API calls*. The EWLM and ARM support is provided by CICS, usually through the monitoring domain. CICS uses WLM macros to make the ARM and EWLM calls.

The calls between the user application program and CICS in the picture are simply the CICS API and resource manager interface (RMI) calls, some of which may cause work to be passed on to other systems (by, for example, function shipping, DPL, DB2®, or WebSphere MQ RMI calls). This in turn may cause CICS to make calls to the WLM to record the state of the work and to obtain child correlators.

Here is a typical EWLM scenario:

1. A piece of work that originated in another eServer enters the CICS system as an HTTP request. It carries an ARM correlator assigned by the first eServer. To the first eServer this is the **current correlator** for this piece of work. To CICS, the correlator assigned by the first eServer is a **parent correlator**.
2. A CICS listener program extracts the inbound correlator and makes it available on attach of the CICS transaction that processes the request.
3. As part of the creation of this CICS task, the extracted correlator is passed to the z/OS Workload Manager (WLM), which passes it to EWLM.
4. EWLM returns a new correlator (a **child correlator**) to be used to identify that part of the overall work flow that takes place in this CICS region.
5. The WLM returns the new correlator to CICS, which passes it on any further WLM calls. The new correlator is the **current correlator** for this CICS transaction instance.

Transport protocols

The following table shows the transport protocols supported by CICS for communication with other servers in the EWLM management domain, and the intercommunication functions supported in each case. These combinations of protocol and function support the use of EWLM correlators.

Table 7. The transport protocols supported by CICS for communication with other servers in the EWLM management domain

Transport protocol	Intercommunication functions that support EWLM correlators	Inbound to CICS	Outbound from CICS
EXCI	1. External CICS interface calls	1. No	1. No
HTTP 1.1	1. SOAP 2. Web services	1. Yes 2. Yes	1. Yes 2. Yes
IIOP	1. Exchanges with any server that supports IIOP (for example, WebSphere Application Server)	1. Yes	1. Yes
ISC over SNA (LU6.1 and LU6.2)	1. None	1. No	1. No
IP interconnectivity	1. Distributed program link (DPL)	1. Yes	1. Yes
MRO	1. Asynchronous processing (STARTs) 2. Business Transaction Services (BTS) RUN SYNCHRONOUS requests 3. Distributed program link (DPL) 4. Distributed transaction processing (DTP) 5. Function shipping 6. Non-terminal-related STARTs 7. Transaction routing	1. Yes 2. Yes 3. Yes 4. Yes 5. Yes 6. Yes 7. Yes	1. Yes 2. Yes 3. Yes 4. Yes 5. Yes 6. Yes 7. Yes

Table 7. The transport protocols supported by CICS for communication with other servers in the EWLM management domain (continued)

Transport protocol	Intercommunication functions that support EWLM correlators	Inbound to CICS	Outbound from CICS
Resource manager interface (RMI)	1. MQ 2. Calls to or from other resource managers (such as IMS™, DB2)	1. No 2. No	1. No 2. Yes

Prerequisites for EWLM support

This topic lists the hardware and software prerequisites for EWLM support.

Hardware requirements

There are no additional hardware requirements beyond those required to run CICS itself.

Software requirements

EWLM support requires either of the following:

- z/OS Version 1.7, plus APAR OA12935
- z/OS Version 1.8

Changes to CICS externals

Changes to system initialization parameters

The MNSUBSYS 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.

Changes to the system programming interface

INQUIRE MONITOR command

The SUBSYSTEMID option of the INQUIRE MONITOR command is obsolete and has been removed.

Changes to global user exits

Some existing global user exits now have access to the z/OS Workload Manager (WLM) Performance Block.

Exits XEIIN, XEIOUT, XEISPIN, and XEISPOUT

The exit-specific parameter lists of these exits have a new parameter:

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 3. 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.

Exits XPCREQ, XPCREQC, and XPCERES

The exit-specific parameter lists of these exits have a new parameter:

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.

Exits XRMIIN and XRMIOUT

The exit-specific parameter lists of these exits have a new parameter:

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 task-related user exits

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 *CICS Customization Guide*. It can set this bit when, for example, 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 ENABLE command that enables the exit program. There is no equivalent option on the ENABLE command to cause the exit program to be invoked by CICS context management.

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 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.

Changes to CEMT

INQUIRE MONITOR command

The SUBSYSTEMID option of the INQUIRE MONITOR command has been removed.

Changes to monitoring

Performance data group DFHTASK

The TRANFLAG field has been modified as follows:

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

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

Changes to problem determination

There are some new trace points in the monitoring domain.

Trace points

Table 8. New monitoring domain trace points

Point ID	Module	Lvl	Type	Data
MN 0A26	DFHMNXM	MN 2	WLM_IWMCLSFY_call	**1** EWLM correlator **2** EWLM outcorrelator **3** Service Class token **4** Service Class name **5** Report Class name **6** IWMCLSFY return code **7** IWMCLSFY reason code
MN 0A27	DFHMNXM	MN 2	WLM_DETACH_call	**1** Performance Block token **2** Performance Block **3** Abnormal flags **4** Owner data

Chapter 17. Additional statistics for MVS Workload Manager

The CICS monitoring domain statistics, and the monitoring section in the System Status report produced by the sample statistics program DFH0STAT, now include MVS workload manager goal information for the CICS address space.

The statistics show:

- The performance goal type for the CICS address space.
- The goal value (for a velocity goal only).
- The importance level of the performance goal.
- Whether or not the CICS address space is designated as CPU critical or as storage critical.

Changes to CICS externals

Changes to CICSPlex SM views and menus

Changes to the MVSWLM - MVS workload management view set

The following fields are added to the view set to show the new MVS workload manager information:

Table 9. New fields in MVSWLM - MVS workload management view set

Field	Attribute name
MVS workload manager goal type	MNGWLMGT
MVS workload goal value	MNGWLMGV
MVS CPU time is critical for the address space	MNGWLMCC
MVS storage is critical for the address space	MNGWLMSK
MVS workload goal importance	MNGWLMGI

Changes to statistics

The following new monitoring domain statistics are provided:

- MVS WLM Goal Type
- MVS WLM Goal Value
- MVS WLM Goal Importance
- MVS WLM CPU Critical
- MVS WLM Storage Critical

Chapter 18. Threadsafety for PLT-enabled global user exit programs

It is now possible to define global user exit programs that are enabled by first-phase program list table (PLT) programs as threadsafe (in previous CICS releases, this technique was available to task-related user exit programs but not to global user exit programs).

Typically, global user exit programs are enabled during CICS initialization, by ENABLE commands issued by PLT programs. To ensure that the exit programs are available as early as possible during CICS startup, global user exit programs such as those that run at the recovery exits are typically enabled during the first phase of PLT processing.

Because first-phase PLT programs run so early in CICS initialization, no resource definitions are available. This means that you cannot use installed PROGRAM definitions (or the program autoinstall user program) to define first-phase PLT programs to CICS, nor to define the user exit programs that first-phase PLT programs enable. Instead, default definitions are installed automatically by CICS. Whether or not program autoinstall is specified as active on the PGAIPGM system initialization parameter, the autoinstall user program is not invoked to allow the definitions to be modified.

This type of autoinstall by CICS is known as *system autoinstall*.

It is recommended that you write your global user exit programs to be threadsafe. However, the system-autoinstalled program definition specifies CONCURRENCY(Quasirent); that is, the exit programs are defined as quasi-reentrant. 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 CICS externals

Changes to the system programming interface

ENABLE command

Options on the ENABLE command have changed:

QUASIRENT

specifies that the global user exit program or task-related user exit program is quasi-reentrant, and relies on the serialization provided by CICS when accessing shared resources. The user exit program is restricted to the CICS permitted programming interfaces, and must comply with CICS quasi-reentrancy rules. CICS always invokes a quasi-reentrant user exit under the QR TCB.

A task-related user exit program is allowed to use MVS services. If it does so, it must switch to its own private TCB before issuing calls to these services, and switch back again before returning to its caller.

THREADSAFE

specifies that the global user exit program or task-related user exit program is written to threadsafe standards, and takes into account the possibility that, when accessing shared resources, other programs may be executing concurrently and attempting to modify the same resources. It uses appropriate serialization techniques when accessing any shared resources.

A threadsafe user exit program must be able to run under whichever TCB CICS invokes it. This could be either the QR TCB or an open TCB. (For task-related user exits only, if OPENAPI is also specified CICS will always invoke the task-related user exit under an L8 open TCB.)

INQUIRE EXITPROGRAM command

Options on the INQUIRE EXITPROGRAM command have changed:

CONCURRENTST
returns a CVDA indicating the concurrency status of the global or task-related user exit program. This is the value of the CONCURRENCY attribute of the PROGRAM definition, or of any override specified by the latest ENABLE command for this program.

CVDA values are:

QUASIRENT
The exit program is defined as being quasi-reentrant, and is able to run only under the CICS QR TCB when invoking CICS services through the CICS API. To use any MVS services, a task-related user exit program must switch to a privately-managed TCB.

THREADSAFE
The exit program is defined as threadsafe, and is capable of running under an open TCB.

For task-related user exit programs only, if the APIST option returns OPENAPI the program will always be invoked under an open TCB.

For both global and task-related user exit programs, an APIST option of CICSAPI means that the program is invoked under whichever TCB is in use by its user task when the program is given control. This could be either an L8 mode open TCB or the CICS QR TCB.

Changes to statistics

The *Global User Exits* report produced by the sample statistics program DFH0STAT now reports the concurrency status of exit programs.

Chapter 19. Storage management above the 2GB boundary

CICS now provides 64-bit storage, allowing you to use storage above the 2GB boundary (above the bar). This capability removes all size restrictions from inter-program data transfer, and provides a means of exploiting z/OS 64-bit capabilities.

Terminology

This topic contains the new and changed terminology used by the new storage management facility above the 2GB boundary.

above the bar
Pertaining to storage above the 2GB boundary.

grande CICS dynamic storage area (GCDSA)
The CICS-key dynamic storage area above the 2GB boundary.

grande dynamic storage area (GDSA)
The dynamic storage area above the 2GB boundary.

Dynamic storage above the 2GB boundary

The GDSA (above the bar dynamic storage area) is the dynamic storage allocated above the 2GB boundary (above the bar). It is also referred to as 64-bit storage.

GDSA refers as a whole to the dynamic storage above the 2GB boundary. This storage is divided into separate dynamic storage areas as follows:

The above the bar CICS DSA (GCDSA).
The CICS-key storage area for all storage above the 2GB boundary (above the bar).

The CICS private area provides more information on the dynamic storage areas used above and below the 2GB boundary.

How storage is allocated and limited

At system initialization, DSA and EDSA are allocated an amount of guaranteed storage, limited by the DSALIM and EDSALIM parameters. GDSA does not preallocate an amount of guaranteed storage and does not have a CICS upper limit of total storage. The limit for above-the-bar storage is controlled by the MEMLIMIT value assigned to the address space by the operating system. As other services in this address space begin to exploit above the bar storage, CICS uses only what it needs.

You can specify MEMLIMIT in the job card in CICS JCL or within the program execution line, as shown in the following example:

```
//CICS EXEC PGM=DFHSIP,PARM='SI',REGION=0M,MEMLIMIT=4G
```

For initialization, CICS recommends a minimum of 2GB of available storage above the bar. 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. For more information on specifying MEMLIMIT for your CICS job, see the *z/OS MVS JCL Reference*.

Short-on-storage (SOS) situations

CICS reserves amounts of storage (called storage cushions) in the DSA and EDSA for use when processing storage stress conditions. An SOS condition occurs when CICS needs to start using the storage cushion. CICS issues a message when SOS is entered and when it is relieved. While in an SOS situation, CICS takes steps to limit work, like preventing acquisition of new input messages, in order to have enough storage to process work already in progress.

For GDSA storage, CICS keeps track of the total amount of above-bar storage in use for the address space and considers an SOS condition when 90% of MEMLIMIT is in use. 5% more of MEMLIMIT is considered as logical storage cushion and CICS continues to allocate more GDSA storage until the logical CICS GDSA limit (95% of MEMLIMIT) is reached. The remaining 5% of MEMLIMIT is made available to other services that need above-the-bar storage to service work already in progress. As before CICS issues appropriate messages and takes steps to start limiting new work to try and keep enough storage to service work already in progress.

You should monitor the use of above bar storage and the MEMLIMIT applied, and make MEMLIMIT adjustments to meet the growing demands of their system. Because you cannot alter MEMLIMIT on a running system, you should factor the timing of adjustments and the adjusted value into the monitoring process. New MEMLIMITS can be introduced on the next start of the CICS region. As an added protection, CICS prevents any transaction from obtaining more than 10% of MEMLIMIT worth of above-the-bar storage.

Changes to CICS externals

Changes to the system programming interface

INQUIRE SUBPOOL command

The DSANAME option returns a new value of GCDSA, denoting the above the bar CICS dynamic storage area.

DSANAME(*data-area***)**
returns an 8-character field giving the name of the dynamic storage area (DSA) in which the specified subpool resides. The value can be one of the following, padded with trailing blanks (X’40’):

CDSA
ECDSA
ERDSA
ESDSA
GCDSA
RDSA
SDSA

INQUIRE SYSTEM command

The INQUIRE SYSTEM command has 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.

Changes to CEMT

INQUIRE DSAS command

The INQUIRE DSAS command has new 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.

The SOSSTATUS option of the INQUIRE DSAS command has been superseded.

INQUIRE SYSTEM command

The INQUIRE SYSTEM command has new 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.

The SOSSTATUS option of the INQUIRE SYSTEM command has been superseded.

Changes to the exit programming interface (XPI)

The INQUIRE_SHORT_ON_STORAGE call

The INQUIRE_SHORT_ON_STORAGE storage control call has a new output parameter:

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 monitoring

Changes to exception class data

A new exception record is produced when there is an above the bar CICS dynamic storage area (GCDSA) resource shortage:

EXCMNTYP Exception type	EXCMNRTY Resource type	EXCMNRID Resource ID	MEANING
EXCMNWT	’STORAGE’	’GCDSA’	Wait for GCDSA storage

Performance data group DFHCHNL

Group DFHCHNL contains new fields:

329 (TYPE-A, ’PGCSTHWM’, 4 BYTES)
Maximum amount (high-water mark), in bytes, of container storage allocated to the user task.

Changes to statistics

The following statistics types now contain information about storage usage above the 2GB boundary:

- Domain subpools statistics
- Summary domain subpools statistics
- Global statistics
- Summary global statistics

Important: The Storage Manager identifier, STISMDSA, has changed its value from 2 to 14 within the DFHSMSDS copy book of the CICS statistics data section. The DSECT structure has changed significantly and this might mean that your reports cannot address correctly.

Chapter 20. Shared data tables larger than 2 GB

Shared data tables are no longer restricted to 2 gigabytes (GB) of data per CICS region.

The data component of a shared data table may now be spread across more than one data space. Furthermore, the table entry and index components of the table are now stored in separate data spaces, rather than in the CICS address space. This allows the total control information for all tables to have a combined size of up to 4 GB (2 GB of table entry descriptors and 2 GB of index nodes).

When shared data tables support is activated, three data spaces are initially created: DFHDT001 (for table entry descriptors), DFHDT002 (for index nodes), and DFHDT003 (for record data storage). Each one is initially allocated 16 MB of virtual storage. Additional data space storage is allocated automatically as required for each data space. If a data space for record data storage reaches the maximum size of 2 GB, a new data space is allocated dynamically. This continues until the table entry descriptor or index node data space is full, the maximum supported number of data spaces (currently 100) are in use, or any data space limit specified by the IEFUSI MVS installation exit has been reached.

The list of data spaces associated with a CICS region is included in the output from the MVS system command `D J,jobname`, so you can use this command to check whether any additional data spaces have been allocated beyond the initial three.

This enhancement should be almost entirely transparent to existing users of shared data tables. The total virtual and real storage requirement for existing data tables has not significantly changed, because most of the additional control information to support larger tables is integrated into existing data areas without needing to expand them.

Chapter 21. Extended addressing for entry sequenced data sets (ESDS)

In previous releases, any entry sequenced data sets (ESDS) used by CICS were restricted in size to 4 gigabytes (GB). This was because CICS programs used 32-bit numbers to address individual records. In CICS TS for z/OS, Version 3.2, CICS programs can use 64-bit numbers to address records, which removes the 4GB limit on the size of the data set.

An entry sequenced data set is a Virtual Storage Access Method (VSAM) dataset that behaves rather like a sequential data set. As new records are added to an ESDS they are appended to the end of the data set. It is not possible to insert a new record between two existing records. Nor is it possible to delete a record after it has been created.

A common way of using an ESDS is to write a number of records to the data set using the WRITE command and to read all of the records back again by a browse. (This is the typical way in which you would use a sequential data set.) To read records back, you use a STARTBR command to position the cursor at the beginning (or end) of the data set, and follow this with a READNEXT (or READPREV) command to read all records within the ESDS.

Records in an ESDS can be either fixed length or variable length.

In the original ESDS design, as each record is added to an ESDS it is assigned a *relative byte address* (RBA), which is an unsigned 32-bit number. The RBA is the number of bytes from the beginning of the ESDS at which the record is located. The use of RBAs implies that an ESDS may not contain more than 4 gigabytes of data.

VSAM now supports *extended ESDS* that use 64-bit RBAs. CICS now supports 64-bit RBAs and extended ESDS.

Terminology

This topic contains the new and changed terminology used by ESDS extended addressing.

extended ESDS
An entry-sequenced data set (ESDS) set that supports extended relative byte addressing and thus may be larger than 4GB.

extended relative byte address (XRBA)
A 64-bit number giving the offset, in bytes, of a record from the beginning of an ESDS data set.

relative byte address (RBA)
A 32-bit number giving the offset, in bytes, of a record from the beginning of an ESDS data set. See also “extended relative byte address”.

Changes to CICS externals

Application programming interface changes

Changed API commands

A new option, XRBA, has been added to the following commands:
- READ
- READNEXT
- READPREV
- RESETBR
- STARTBR
- WRITE

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.

New error codes have been added, as follows:
- Applies to the READ and STARTBR commands:

  INVREQ

  59 XRBA was specified, but the data set is not an ESDS.
- Applies to the READ, READNEXT, READPREV, RESETBR, and STARTBR commands:

  NOTFND

  81 XRBA was specified, and the value of RIDFLD was greater than 4 GB, but the data set is not an extended ESDS.

Changes to the system programming interface

INQUIRE FILE command

The INQUIRE FILE command has a new 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.

Changes to CEMT

INQUIRE FILE command

The INQUIRE FILE command has a new 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.

Changes to CICSPlex SM views and menus

A new keyword, RBATYPE, is added to the LOCFILE base table:

Table 10. LOCFILE resource table attributes

Name	Datatype	Source	Len	Sum	Set	Description	Get Invalid	Set Invalid	Attr ID
RBATYPE	CVDAS	INQ	4	LIKE		Type of VSAM extended addressing	E530, E620, E630 and E640		54
RBATYPE Output Valid Values	EXTENDED NOTAPPLIC NOTEXTENDED								

Changes to global user exits

There are changes to the following global user exits in the file control domain:

XFCFRIN and XFCFROUT

- 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

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

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.

Changes to statistics

File control statistics now include statistics for extended entry sequenced data sets.

Chapter 22. Greater precision and capacity for monitoring clocks

The CICS monitoring clocks for performance class data now measure dispatch (elapsed) time and CPU time for CICS TCBs in single microseconds, rather than in units of 16 microseconds. The clock capacity, which was around 19 hours, is now only bounded by the capacity of the local store clock, which is several years.

Changes to the monitoring clock format

To achieve increases in precision and capacity, the format of monitoring clocks for performance class data has been changed.

A CICS monitoring clock consists of three components:

1. **Timer component.** This is a value giving the accumulated time recorded by the clock, expressed in units.
2. **8 reserved bits.** These are used for control of the clock.
3. **Period count.** The time recorded by the timer component is accumulated during one or more measurement periods. The period count is a value giving the number of measurement periods.

In CICS Transaction Server for z/OS, Version 3 Release 2, there are two changes to the clock format for the monitoring clocks for performance class data:

- The units of time recorded by the timer component are now local store clock (STCK) units of one microsecond, rather than units of 16 microseconds as they were in earlier releases.
- The timer component is now a 64-bit value, rather than a 32-bit value. With a 32-bit timer component, the clock capacity was around 19 hours, but with a 64-bit timer component, the clock capacity is only bounded by the capacity of the local store clock, which is several years.

The 8 reserved bits and the period count are unchanged.

Because of these changes, the overall size of a monitoring clock for performance class data has increased from 8 bytes to 12 bytes. The output from the clock can also be formatted to 6 decimal places.

These changes to the monitoring clock format apply **only** to clocks for performance class data. For clocks for transaction resource class data, the timer component is still a 32-bit value, measuring in units of 16 microseconds, and the clock still has a length of 8 bytes. (Transaction resource class data provides supplementary information about individual file and temporary storage queue resources accessed by a transaction.) For exception class data, there are no clocks.

Consequences of the monitoring clock changes

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. These enhancements, and the changes in clock format which were needed to achieve them, have some important consequences for your activities using performance class data.

Key benefits of monitoring clock changes

With the new monitoring clock format, you should be able to experience:

- Accurate reporting with even the fastest processors.
- More useful reporting for long-running transactions.
- More accurate diagnosis of performance timing issues.
- Greater capability for accounting and chargeback.

Changes you might see in your performance 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 CICS and CICSPlex SM system administration

CICS system administrators should note these consequences of the monitoring clock changes:

- The length of a monitoring clock for performance class data has increased from 8 bytes to 12 bytes. This affects many performance class data fields, and also affects any user-defined event-monitoring points (EMPs) which involve clocks.
- Because of the increase in the clock size, the length of a standard performance class monitoring record, as output to SMF, has increased.
- The offsets have changed for a number of the default CICS dictionary entries in the dictionary data sections of CICS monitoring SMF type 110 records.

For more details about all these changes, see “Changes to monitoring” on page 153.

CICSPlex SM handles clock data in both the old format (SCLOCK data type) and the new format (the new SCLOCK12 data type). CICSPlex SM system administrators should note these consequences of the monitoring clock changes:

- New time formatting options are available for Web User Interface views.
- Some base tables which had attributes with a datatype of SCLOCK, have had these converted to the new data type SCLOCK12, and application programs which extract data from the affected resource tables must be recompiled.
- Evaluation definitions (EVALDEF) which specify SCLOCK12 data to the maximum level of precision (6 decimal places, giving a precision of one microsecond) cannot be installed on earlier releases of CICSPlex SM. If you need to use an EVALDEF involving SCLOCK12 data with an earlier release of CICSPlex SM, either specify the data to 4 decimal places (giving a precision of one ten-thousandth of a second), or do not specify any decimal places (giving a precision of one second). 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 batch utility EYU9XDBT.

For more details about the changes to CICSPlex SM, see “Changes to the CICSPlex SM programming interface” and “Changes to CICSPlex SM views and menus” on page 152.

Changes to CICS externals

Changes to the CICSPlex SM programming interface

There is a new data type SCLOCK12 for resource table attributes.

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`
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.

Changes to CICSPlex SM views and menus

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.

Changes to CICS utilities

Changes to the monitoring sample program DFH$MOLS

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).

Changes to monitoring

Increase in monitoring clock size

Because of the increased size of the timer component, the overall length of a monitoring clock for performance class data has increased from 8 bytes to 12 bytes.

This change affects all performance class data fields defined as ″TYPE-S″. There are changed fields in the following performance data groups:

DFHCICS
DFHDATA
DFHDEST
DFHFEPI
DFHFILE
DFHJOUR
DFHPROG
DFHRMI
DFHSOCK
DFHSYNC
DFHTASK
DFHTEMP
DFHTERM

This change 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: The monitoring clocks for transaction resource class data are not changed, and they remain at 8 bytes.

Increase in length of performance class monitoring records

Because of the increase in the monitoring clock size and other changes, the length of a standard performance class monitoring record, as output to SMF, has increased from 1848 bytes in CICS Transaction Server for z/OS, Version 3 Release 1, to 2352 bytes in CICS Transaction Server for z/OS, Version 3 Release 2. This does not take into account any user data that you add, or any system-defined data fields that you exclude.

Note: CICS Transaction Server for z/OS, Version 3 Release 2 also introduces a data compression facility for SMF 110 monitoring records, which can provide a significant reduction in the volume of data written to SMF. See Chapter 23, “Data compression for monitoring records,” on page 155 for more information about this facility.

Changes to dictionary entries

Because of the increase in the monitoring clock size, the offsets have changed for a number of the default CICS dictionary entries in the dictionary data sections of CICS monitoring SMF type 110 records. The fields affected are the last 75 fields listed at the end of the sequence of default CICS dictionary entries, starting with field 007 in group DFHTASK.

Changes to statistics

New time format in statistics reports

The CICS-supplied statistics utility program DFHSTUP and the statistics sample program DFH0STAT now exploit the benefits of increased clock precision and capacity. They display a 4 digit count for the hours in time fields instead of a 2 digit count, and also display the time to 6 decimal places (down to one microsecond) instead of 5 decimal places. The new format is used in the Dispatcher domain statistics reports in DFHSTUP, and in the Dispatcher TCB Modes Report in DFH0STAT.

The new format for the time fields is:

```
hhhh:mm:ss.000000
```

This can also be expressed as

```
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.

Changes to DFHMNTDS DSECT

Because of the increase in the monitoring clock size, the DFHMNTDS DSECT, which maps the performance class monitoring data for a task, has changed significantly. Older versions of this DSECT are not compatible with the new DSECT, and application programs using this DSECT must be recompiled. The affected application programs are those which use the COLLECT STATISTICS MONITOR command, giving the number of a specific task for which statistics are required.

Changes to problem determination

New trace level for CICSPlex SM

A new trace level is available for special tracing for CMAS and MAS components and for Web User Interface trace services. The new trace level 19 is used to trace data formatting.

Note: Special trace points (levels 3-32) should be activated only at the request of customer support personnel.

Chapter 23. Data compression for monitoring records

CICS can now perform data compression on the SMF 110 monitoring records output by the CICS monitoring facility. Data compression can provide a significant reduction in the volume of data written to SMF. The records are compressed and expanded using standard z/OS services.

Activating data compression

To activate data compression for monitoring records, you need to specify the option COMPRESS=YES in your Monitoring Control Table (MCT), using the DFHMCT TYPE=INITIAL macro. The default for this option is NO, meaning that data compression is not used. If the system initialization parameter MCT=NO is specified, the default MCT built by CICS specifies COMPRESS=NO.

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.

Compressing monitoring record data

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. This process can provide a very considerable reduction in the volume of data written to SMF, and a corresponding reduction in I/O and CPU usage for the SMF address space. If you normally exclude monitoring data fields in order to reduce data volume, you might find that using data compression removes the need for you to do this, and enables you to collect complete monitoring data.

The collected monitoring data can include a mix of compressed records and records that have not been compressed. Records might be left not compressed because of the following situations:

- Depending on the data pattern of the record, compressing the data section could possibly result in a larger record. If this situation occurs, CICS chooses not to compress the record.
- Data compression might fail because of a problem involving the z/OS Data Compression and Expansion Services.
- Data compression might be switched off dynamically using the CEMN transaction or the CEMT or EXEC CICS SET MONITOR command.

Expanding monitoring record data

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

- The CICS-supplied monitoring sample program DFH$MOLS supports the expansion of compressed CICS SMF 110 monitoring records. DFH$MOLS automatically identifies any compressed monitoring records in the input, and uses the z/OS data expansion service to expand them before working with them. If you specify the EXPAND control statement, DFH$MOLS copies the compressed monitoring records to an output dataset in their expanded format, along with the records that were never compressed. See Sample monitoring data print program in the *CICS Operations and Utilities Guide* for further information on the DFH$MOLS program.

- If you use an SMF 110 reporting tool supplied by IBM or by another vendor, and you want to activate data compression, 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 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.

A reporting tool that is using the z/OS Data Compression and Expansion Services needs to know that:

- The field SMFMNCRL in the SMF product section of the record identifies where data compression has been used for a monitoring record, and gives the compressed length of the CICS data section. A zero value for this field means that data compression was not performed on the record.
- The maximum length of the CICS data section of an SMF 110 monitoring record, when expanded, is 32598 bytes.

For detailed information about the z/OS Data Compression and Expansion Services (CSRCESRV), see the *z/OS MVS Assembler Services Guide*, and the *z/OS MVS Assembler Services Reference ABE-HSP*.

Changes to CICS externals

Changes to resource definition

Monitoring control table macro (DFHMCT)

The DFHMCT macro has a new option:

COMPRESS={NO|YES}
This option specifies whether or not you want data compression to be performed for the CICS SMF 110 monitoring records output by the CICS monitoring facility.

NO This is the default, and specifies that you do not want monitoring record data compression to be performed for the CICS SMF 110 monitoring records output by the CICS monitoring facility.

YES Specifies that you do want monitoring record data compression to be performed for the CICS SMF 110 monitoring records output by the CICS monitoring facility. For information about data compression, see Data compression for monitoring records.

Changes to the system programming interface

INQUIRE MONITOR command

The INQUIRE MONITOR command has a new 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.

SET MONITOR command

The INQUIRE MONITOR command has a new 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.

Changes to CEMT

INQUIRE MONITOR command

The INQUIRE MONITOR command has a new 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.

SET MONITOR command

The SET MONITOR command has new options:

COMpress
Data compression is to be performed for monitoring records.

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

Changes to the CICSPlex SM programming interface

Changes to resource tables

The MONITOR resource table has a new field COMPRESSST for the status of data compression for monitoring records.

Changes to customization interfaces

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 statistics

CICS statistics for the monitoring domain now report:

- The status of data compression for the CICS region.

- The number of compressed monitoring records written to the SMF data set, and the number that were not compressed.
- The average record length for records that were compressed, and the average record length for records that were not compressed.

The statistics for numbers of records written to the SMF data set and for the average record length are only collected when data compression is active for the CICS region.

Changes to CICS utilities

Changes to the monitoring sample program DFH$MOLS

DFH$MOLS is able to identify any SMF 110 monitoring records that have been compressed, and expand them using the z/OS Data Compression and Expansion Services (CSRCESRV), before printing reports . If you specify the EXPAND control statement, DFH$MOLS copies the compressed monitoring records to an output dataset 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 other reporting tools.

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.

Changes to problem determination

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

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

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

Trace

There are new monitoring domain trace points, MN 0151–0153, MN 0512, and MN 0513, to trace data compression and calls to the z/OS Data Compression and Expansion Services (CSRCESRV).

Chapter 24. Monitoring facility transaction CEMN

The new CEMN monitoring facility transaction provides an operator interface which you can use to check on the options currently in effect for the CICS monitoring facility, and to change some of the settings without needing to restart CICS.

CEMN - CICS monitoring facility

Use the CEMN monitoring facility transaction to inquire on and set options for the CICS monitoring facility.

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 settings which can be changed using CEMN are:

Monitoring Status
Whether monitoring is required.

Exception Class
Whether exception class data is required.

Performance Class
Whether performance class data is required.

Resource Class
Whether resource class data is required.

Compression Status
Whether monitoring data is to be compressed.

Converse Status
Whether separate performance class records are produced for conversational tasks.

Syncpoint Status
Whether separate performance class records are produced for syncpoint requests.

Frequency
The interval at which CICS produces performance class records for long-running tasks.

Changing these settings affects the monitoring data which is recorded for tasks that are running at the time you make the change. The effects are the same as if you had changed the settings using the SET MONITOR system programming command or the CEMT SET MONITOR command. CEMT SET MONITOR explains how data for running tasks is accumulated, recorded, or lost when you change the settings for the CICS monitoring facility.

The settings for which information is displayed, but which cannot be changed using CEMN, are:

File Resource Limit
The maximum number of files for which resource class data is collected.

Tsqueue Resource Limit
The maximum number of temporary storage queues for which resource class data is collected.

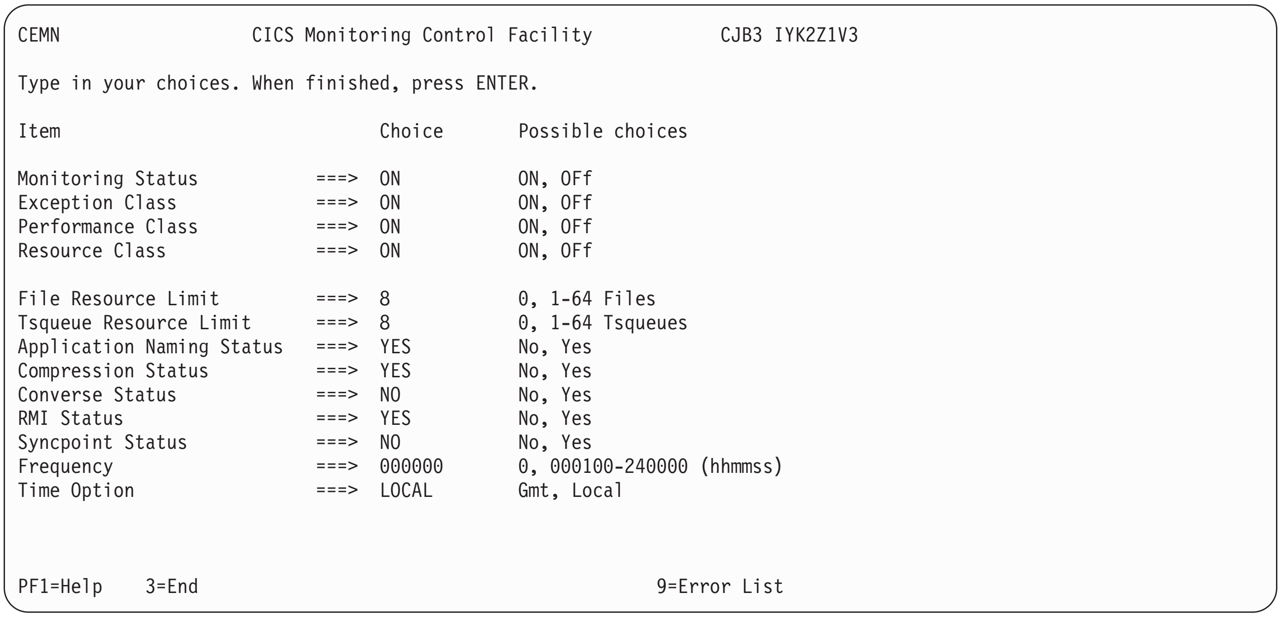
Application Naming Status
Whether CICS application naming support is enabled.

RMI Status
Whether additional performance monitoring is active for CICS resource managers.

Time Option
Whether the time stamp fields are returned in GMT or local time.

You can change the file resource limit, temporary storage queue resource limit, application naming status, and RMI status using the DFHMCT TYPE=INITIAL macro in the monitoring control table (MCT). You can change the time option using the MNTIME system initialization parameter. A CICS restart is required to implement any of those changes.

Start the CEMN transaction by typing CEMN on the command line of your display and pressing the ENTER key. You get the following display showing the current state of the CICS monitoring facility and the settings of the monitoring options in your own system:



```
CEMN                 CICS Monitoring Control Facility            CJB3 IYK2Z1V3

Type in your choices. When finished, press ENTER.

Item                              Choice       Possible choices

Monitoring Status           ===>  ON           ON, OFf
Exception Class             ===>  ON           ON, OFf
Performance Class           ===>  ON           ON, OFf
Resource Class              ===>  ON           ON, OFf

File Resource Limit         ===>  8            0, 1-64 Files
Tsqueue Resource Limit      ===>  8            0, 1-64 Tsqueues
Application Naming Status   ===>  YES          No, Yes
Compression Status          ===>  YES          No, Yes
Converse Status             ===>  NO           No, Yes
RMI Status                  ===>  YES          No, Yes
Syncpoint Status            ===>  NO           No, Yes
Frequency                   ===>  000000       0, 000100-240000 (hhmmss)
Time Option                 ===>  LOCAL        Gmt, Local



PF1=Help    3=End                                          9=Error List
```

Figure 4. CEMN transaction: initial screen

The input fields can be overtyped with the new values that you require. The File Resource Limit, Tsqueue Resource Limit, Application Naming Status, RMI Status, and Time Option fields are inquire-only fields. When you press ENTER, CEMN issues the necessary commands to set the new values. If there are any errors, PF9 can be pressed to display the error messages. If there is only one short error message, it appears near the bottom of this display.

CEMN is a Category 2 transaction.

Chapter 25. WebSphere MQ monitoring and statistics changes

The CICS-MQ adapter, CICS-MQ bridge and CICS-MQ trigger monitor (previously part of WebSphere MQ), are now shipped with CICS TS, introducing changes to monitoring and statistics.

A new CICS statistics type MQCONN and corresponding new CICSPlex SM view set MQCONN - WebSphere MQ Connection provide global statistics information on the connection between a CICS region and WebSphere MQ.

Two new DFHDATA monitoring fields, WMQREQCT and WMQGETWT, provide information on WebSphere MQ requests and wait times.

Changes to CICS externals

Changes to CICSPlex SM views and menus

New view set MQCONN - WebSphere MQ Connection

A new view set is provided to display status information and statistics for the WebSphere MQ connection for a CICS region. To access this view set from the main menu, click:

CICS operations views → DB2, DBCTL and WebSphere MQ operations views → WebSphere MQ connections

The views in the supplied view set are:

There are no action commands in WebSphere MQ connection views.

Changes to monitoring

Performance data group DFHDATA

Group DFHDATA contains new fields:

396 (TYPE-S, ’WMQGETWT’, 12 BYTES)
The elapsed time the user task waited for WebSphere MQ to service the user task’s GETWAIT request.

For more information, see Clocks and time stamps, and Transaction wait (suspend) times.

395 (TYPE-A, ’WMQREQCT’, 4 BYTES)
The total number of MQ requests issued by the user task.

Changes to statistics

CICS now collects statistics on the usage of WebSphere MQ connections. The statistics include information about:

- The status of the CICS-MQ adapter and of the connection between CICS and WebSphere MQ.
- The WebSphere MQ API calls made using the connection.
- The different WebSphere MQ API commands issued using the connection.

- The units of work committed, backed out, indoubt or unresolved on the connection.

The statistics are recorded by specifying the MQCONN option on the CEMT PERFORM STATISTICS and EXEC CICS PERFORM STATISTICS RECORD commands, and retrieved online using the EXTRACT STATISTICS command specifying RESTYPE(MQCONN). They are mapped by the DFHMQGDS DSECT.

WebSphere MQ connection statistics can be included in reports produced by the statistics reporting utility DFHSTUP, using the MQCONN resource type, and are included in reports generated by the sample statistics program DFH0STAT.

Chapter 26. Additional storage information for MVS TCBs

The INQUIRE MVSTCB command now provides additional information about the MVS storage key of each storage element allocated to each TCB, and the amount of storage actually in use within each storage element. The CICS global and resource statistics for MVS TCBs now report the storage actually in use, as well as the storage allocated to TCBs.

The amount of storage in use, as displayed by the new command option and statistics, is the amount of storage actually GETMAINed by the task. Previously, the statistics only displayed the storage allocated to the TCBs, which is always allocated in page multiples (4096 bytes). The new information gives a more accurate picture of the storage actually being used.

Changes to CICS externals

Changes to the system programming interface

INQUIRE MVSTCB command

The 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.

Changes to CICSPlex SM views and menus

Changes to the MVSTCB - MVS TCBs view set

The following fields are added to the view set to show information about storage in use by each MVS TCB.

Table 11. New fields in MVSTCB - MVS TCBs view set

Field	Attribute name
Private storage in use above 16MB	TCBSTGAINUSE
Private storage in use below 16MB	TCBSTGBINUSE

Changes to the MVSTCBGL - Global MVS TCB view set

The following fields are added to the view set to show information about storage in use by MVS TCBs.

Table 12. New fields in MVSTCBGL - Global MVS TCB view set

Field	Attribute name
Storage above 16MB being used by CICS TCBs	CICSTCBSTGAI
Storage above 16MB being used by non-CICS TCBs	NCICSTCBSGAI
Storage below 16MB being used by CICS TCBs	CICSTCBSTGBI
Storage below 16MB being used by non-CICS TCBs	NCICSTCBSGBI

Changes to the MVSESTG - MVS storage element view set

The following fields are added to the view set to show information about storage in use and the storage key for each storage element.

Table 13. New fields in MVSESTG - MVS storage element view set

Field	Attribute name
Number of storage bytes being used	INUSELENGTH
Storage key	STORAGEKEY

Changes to statistics

The following new statistics are provided for MVS TCBs:

- Global statistics for the total current private storage in use, for both CICS TCBs and non-CICS TCBs, below and above 16MB
- Resource statistics for private storage in use for individual TCBs, below and above 16MB

Chapter 27. XCF group limit relief

The effective limit of 2047 CICS regions that a single sysplex can support has been lifted.

Multiregion operation (MRO) enables CICS regions that are running in the same MVS image to communicate without the need for an access method such as ACF/VTAM or TCP/IP. When used with the MVS cross-system coupling facility (XCF), MRO allows CICS regions in different MVS images, but within the same sysplex, to communicate without an SNA access method.

Currently, all the CICS regions in a sysplex that use XCF/MRO must join the same XCF group, DFHIR000; and an XCF group is limited to 2047 members. Effectively, this imposes a limit on the number of CICS regions that a sysplex can support.

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 new function also brings organizational benefits: it would be possible, for example, to place production regions into a different XCF group from development and test regions.

Cross-system multiregion operation (XCF/MRO)

The cross-system coupling facility (XCF) is part of the MVS/ESA base control program, providing high performance communication links between MVS images that are linked in a sysplex (**sys**tems com**plex**) by channel-to-channel links, ESCON® channels, or coupling facility links.

The IRC provides an XCF access method that makes it unnecessary to use VTAM® to communicate between MVS images within the same MVS sysplex.

Each CICS region is assigned to an XCF group when it logs on to IRC, even if it is not currently connected to any regions in other MVS images. You specify the name of the XCF group on the XCFGROUP system initialization parameter. If XCFGROUP is not specified, the region becomes a member of the default CICS XCF group, DFHIR000.

When members of a CICS XCF group that are in different MVS images talk to each other, CICS selects the XCF access method dynamically, overriding the access method specified on the connection resource definition. The use of the MVS cross-system coupling facility enables MRO to function *between* MVS images in a sysplex environment, supporting all the usual MRO operations, [1] such as:

- Function shipping
- Asynchronous processing
- Transaction routing
- Distributed program link
- Distributed transaction processing.

[1] XCF/MRO does not support accessing shared data tables across MVS images. Shared access to a data table, across two or more CICS regions, requires the regions to be in the same MVS image. To access a data table in a different MVS image, you can use function shipping.

Each CICS region can be a member of only one XCF group, which it joins when it logs on to IRC. The maximum size of an XCF group is limited by the MVS MAXMEMBER parameter, and there is an absolute limit of 2047 members. If this limit is a problem for you (because, for example, it limits the number of CICS regions you can have in your sysplex), you can create multiple XCF groups, each containing a different set of regions. You could, for example, have one XCF group for production regions and another for development and test regions. If you do need 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

Note that CICS regions can use MRO or XCF/MRO to communicate *only with regions in the same XCF group*. Members of different XCF groups cannot communicate via MRO (or XCF/MRO), *even if they are in the same MVS image*.

CICS regions linked by XCF/MRO can be at different release levels; see Multiregion operation. Depending on the versions of CICS installed in the MVS images participating in XCF/MRO, the versions of DFHIRP installed in the link pack areas of the MVS images can be different. If a single MVS image contains different releases of CICS, all using XCF/MRO to communicate with regions in other images in the sysplex, the DFHIRP module in the MVS LPA should be that from the most current CICS release in the image, or higher. 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. For full details of software and hardware requirements for XCF/MRO, see Requirements for XCF/MRO.

Figure 5 on page 169 is a simple example of the use of XCF/MRO in a sysplex environment. In this example, there is only one CICS XCF group, DFHIR000. The members of DFHIR000 can communicate via XCF/MRO links across the two MVS images.

The MRO links between CICS1 and CICS2, and between CICS3 and CICS4, use either the IRC or XM access methods, as defined for the link. The MRO links between CICS regions on MVS1 and the CICS regions on MVS2 use the XCF method, which is selected by CICS dynamically.

In each MVS, the DFHIRP module in the LPA should be at the level of the highest CICS TS for z/OS release in the image.

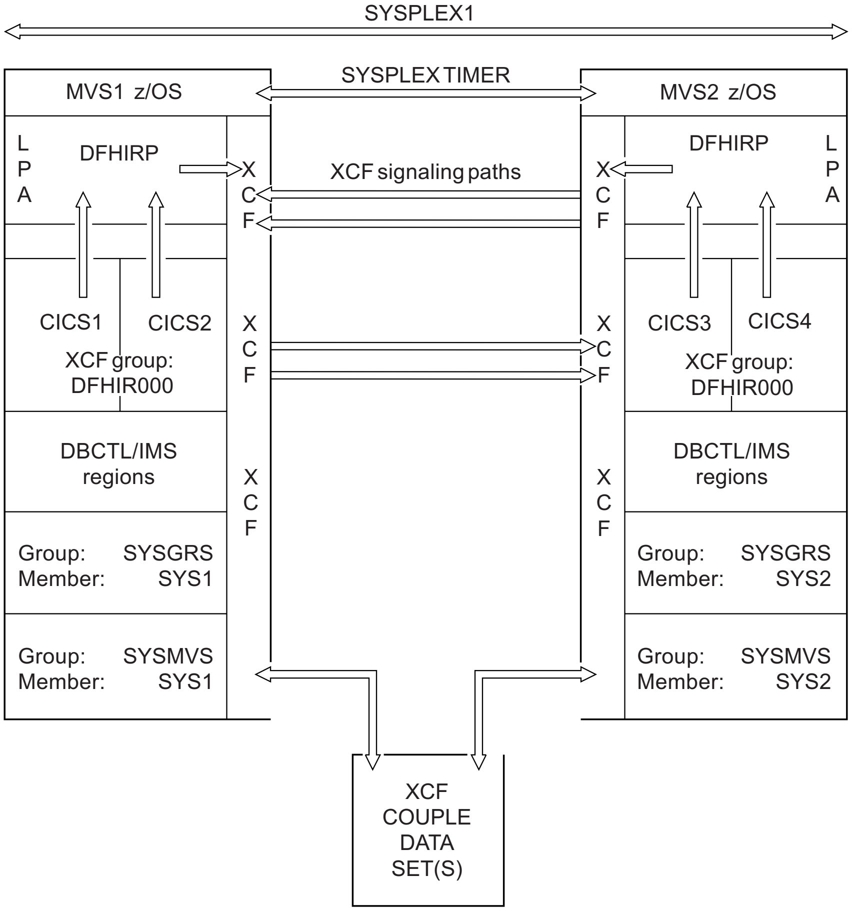


Figure 5. A sysplex (SYSPLEX1) containing a single CICS XCF group.

Figure 6 on page 170 is a slightly more complex example. In this case, there are two CICS XCF groups, DFHIR000 and DFHIR001. The members of each XCF group can communicate across the MVS images by means of XCF/MRO links.

To support multiple CICS XCF groups, both MVS images must be z/OS Version 1.7 or later and must use the CICS TS for z/OS, Version 3.2 or later version of DFHIRP. (Although z/OS has supported multiple XCF groups since Version 1.6, CICS TS for z/OS, Version 3.2 (required to join an XCF group other than DFHIR000) requires z/OS Version 1.7 or later.)

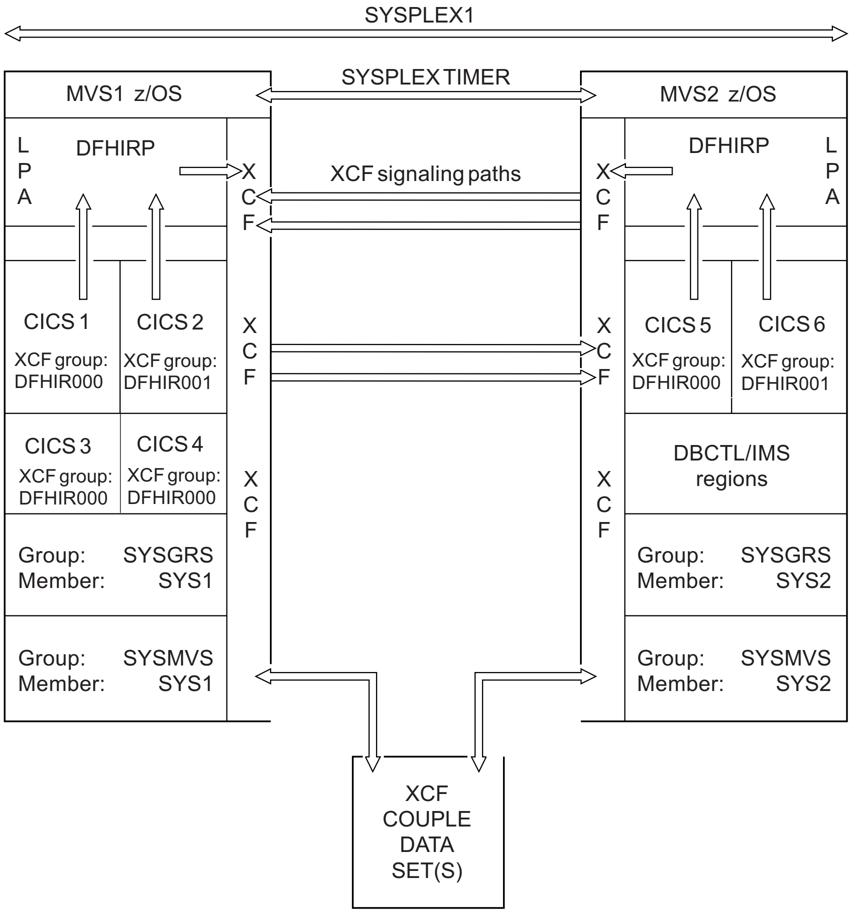


Figure 6. A sysplex (SYSPLEX1) containing two CICS XCF groups

Note that, in Figure 6:

- The members of the DFHIR000 XCF group in MVS1 (CICS 1, CICS 3, and CICS 4) use XCF/MRO, which is selected by CICS dynamically, to communicate with the member of the DFHIR000 XCF group in MVS2 (CICS 5). Similarly, CICS 2 in MVS1 uses XCF/MRO to communicate with CICS 6 in MVS 2; they are both members of the DFHIR001 group.
- CICS 1, CICS 3, and CICS 4 cannot use XCF/MRO to communicate with CICS 6, because CICS 6 is in a different XCF group. Similarly, CICS 2 cannot use XCF/MRO to communicate with CICS 5.
- Because they are in the same MVS image and the same XCF group, CICS 1, CICS 3, and CICS 4 can communicate with each other using either the MRO(IRC) or MRO(XM) access method, as defined for the links.
- CICS 5 cannot use any form of MRO to communicate with CICS 6, even though they are in the same MVS image, because they are in different XCF groups. Similarly, CICS 2 cannot use any form of MRO to communicate with CICS 1, CICS 3, or CICS 4.

Generating XCF/MRO support

1. Depending on the versions of CICS installed in the MVS images participating in XCF/MRO, the versions of DFHIRP installed in the images can be different. For all the MVS images containing CICS systems to be linked, ensure that the version of DFHIRP in the extended link pack area (ELPA) is at the required level. The DFHIRP module should be that from the most current CICS release in the image, or higher.

   Note: 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. (Although z/OS has supported multiple XCF groups since Version 1.6, CICS TS for z/OS, Version 3.2 (required to join an XCF group other than DFHIR000) requires z/OS Version 1.7 or later.)
2. Ensure that each CICS APPLID is unique within the sysplex.
3. Ensure that the value of the MAXMEMBER MVS parameter, used to define the XCF couple data sets, is high enough to cater for the largest CICS XCF group. The maximum size of any XCF group within a sysplex is limited by this value. The theoretical maximum size of any XCF group is 2047 members.

   External CICS interface (EXCI) users that use an XCF/MRO link also join an XCF group. You should therefore set the value of MAXMEMBER high enough to allow all CICS regions and EXCI XCF/MRO users in the largest CICS XCF group to join the group concurrently.

   To list the CICS regions and EXCI users in an XCF group, use the MVS DISPLAY command. For example, to list the CICS regions and EXCI users in the DFHIR001 XCF group, use the command:

   ```
   DISPLAY XCF,GROUP,DFHIR001,ALL
   ```

   Attention:

   Do not rely on the default value of MAXMEMBER, which may be too low to allow all the CICS regions and EXCI users in the largest XCF group to join the group. This is especially important if you have only a few CICS XCF groups.

   Likewise, do not set a value much larger than you need, because this will result in large couple data sets for XCF. The larger the data set, the longer it will take to locate entries.

   We suggest that you make the value of MAXMEMBER 10-15 greater than the combined number of CICS regions and EXCI users in the largest CICS XCF group.

Each CICS region joins an XCF group when it logs on to DFHIRP. Its member name is its APPLID (NETNAME) used for MRO partners. The XCF group name is specified on the XCFGROUP system initialization parameter. If XCFGROUP is not specified, the XCF group name defaults to DFHIR000.

At connect time, CICS invokes the IXCQUERY macro to determine whether the CICS region being connected to resides in the same MVS image. If it does, CICS uses IRC or XM as the MRO access method, as defined in the connection definition. If the partner resides in a different MVS image, CICS uses XCF as the access method, regardless of the access method defined in the connection definition.

Note: CICS regions can use MRO or XCF/MRO to communicate *only with regions in the same XCF group*. Members of different XCF groups cannot communicate via MRO (or XCF/MRO), *even if they are in the same MVS image*.

Changes to CICS externals

Changes to system initialization parameters

New system initialization parameter

XCFGROUP

Specifies the name of the cross-system coupling facility (XCF) group to be joined by this region.

For details of the parameter, see "XCFGROUP" on page 277.

Changed system initialization parameter

The APPLID system initialization parameter is changed as follows:

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.

If CICS is an XRF partner, its *specific* applid must be unique within the sysplex. If, on CICS startup, the specified specific 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.

Changes to the system programming interface

INQUIRE IRC command

The INQUIRE IRC command has a new 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)" on page 167 in the *CICS Intercommunication Guide*.

Changes to other programming interfaces

Changes to the EXCI options table

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.

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. XCF group limit relief adds a new option, XCFGROUP, 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”.

Changes to CEMT

INQUIRE IRC command

The INQUIRE IRC command has a new 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*.

Changes to statistics

The *System Status* report produced by the sample statistics program DFH0STAT now reports the XCF Group Name.

Changes to CICSPlex SM views and menus

A new CICS-queried attribute, XCFGROUP, is added to the CMAS and CICSRGN base tables.

Table 14. CMAS resource table attributes

Name	Datatype	Source	Len	Sum	Set	Description	Attr ID
XCFGROUP	CHAR	INQ	8	LIKE		XCF group ID	65

Table 15. CICSRGN resource table attributes

Name	Datatype	Source	Len	Sum	Set	Description	Get Invalid	Set Invalid	Attr ID
XCFGROUP	CHAR	INQ	8	LIKE		XCF group ID	E530, E620, E630 and E640		172

Chapter 28. Configuration and problem determination improvements for Java programs

There are a number of new features to help you configure JVMs more simply, identify problems with JVMs more quickly, and obtain detailed problem diagnosis information more easily.

- CICS validates the Java and CICS home directories specified in a JVM profile when you attempt to start the JVM, to check that the directories exist, that CICS has **read** permissions for them, and that the install check file is present.
- CICS issues warning or error messages at runtime if you include a deprecated option in a JVM profile.
- CICS builds a base library path and a base class path for you using the supplied directories for CICS and Java files, so these directories do not now need to be specified explicitly in JVM profiles. There are new options that you can use to specify additional items on the class paths, which indicate the correct location for the classes more clearly.
- A new symbol &JVM_NUM; is available in JVM profiles to insert a unique JVM number in the names of output and dump files produced by the JVM. The unique JVM number is also added to file names produced by the **-generate** option.
- The sample JVM profiles and documentation include more guidance about specifying Java dump options.
- The options specified for JVMs are traced when the JVMs are started.
- You can specify any JVM option in a JVM profile, prefixed with a hyphen, and it is passed through to the JVM. You are no longer restricted to the subset of options previously recognized by CICS in JVM profiles.
- The messages that CICS produces relating to JVMs now provide more diagnostic information and advice.
- CICS now formats the output from the JVM trace facility, adding the description of each trace point from the TraceFormat.dat file and placing the inserts, so that you do not have to interpret the data manually.
- The USECOUNT option on the INQUIRE PROGRAM command now displays a use count for Java programs.

Validation of JVM profile options

CICS carries out a number of checks on key options specified in your JVM profiles, whenever you start JVMs. This enables the early detection of problems in your JVM setup, and more informative messages to help you resolve the problems.

CICS carries out checks relating to the following JVM profile options:

CICS_HOME

CICS checks the following points for this directory:

- The directory exists on HFS.
- CICS has at least **read** permission to access the directory.
- The CICS_INSTALL_OK file is present in the directory, indicating a completed installation of the CICS files in this location on HFS.
- The CICS_INSTALL_OK file contains the correct CICS version number, indicating that you are not inadvertently using the installed files from a previous CICS release (which might happen if you migrated a JVM profile and did not update this option).

If any problems are found, CICS issues an error message and does not start the JVM. CICS also issues a warning message if you use the old CICS_DIRECTORY option instead of the CICS_HOME option.

JAVA_HOME

CICS checks similar points for this directory:

- The directory exists on HFS.
- CICS has at least **read** permission to access the directory.
- The JDK_INSTALL_OK file is present in the directory, indicating a completed installation of the IBM Software Developer Kit for z/OS, Java 2 Technology Edition, Version 1.4.2 files in this location on HFS.
- The Java release number in the JDK_INSTALL_OK file is the correct version supported by CICS.

If any problems are found, CICS issues an error message and does not start the JVM.

Deprecated class path options: LIBPATH, CLASSPATH, TMPREFIX, and TMSUFFIX

A warning message is issued at JVM startup if one of these options is found in a JVM profile, to prompt you to transfer its value to an appropriate class path. The message advises on the correct option to use instead.

Improved handling for class paths

In CICS Transaction Server for z/OS, Version 3 Release 2, a base library path and a base class path are built automatically by CICS, and do not need to be specified explicitly in JVM profiles. New options are provided to place items on the class paths before or after the base class paths, as necessary. CICS also tolerates any obsolete class path options still in your JVM profiles, and builds them into an appropriate class path.

The base library path for the JVM is built automatically using subdirectories of the directories specified by the CICS_HOME and JAVA_HOME options in the JVM profile. This path includes all the dynamic link library (DLL) files required to run the JVM, and the native libraries used by CICS. The base library path is not visible in the JVM profile.

For the library path, you now only need to specify any additional DLL files that you added to the library path in earlier releases. The option you should use for this is LIBPATH_SUFFIX. When CICS builds the library path, these items are placed on the library path after the base library path directories.

CICS builds a base class path for the JVM in a similar way. This class path contains the JAR files supplied by CICS and by the JVM. You do not need to specify these explicitly on your class paths.

If you need to check the exact contents of the base library path and the base class path for a particular JVM profile, you can temporarily specify the PRINT_JVM_OPTIONS=YES option in the JVM profile. When this option is specified, all the options passed to the JVM at startup, including the contents of the class paths, are printed to SYSPRINT. The output is produced every time a JVM is started with this option in its profile, so you should add the option to the appropriate JVM profile, wait for a JVM to be started with the profile (or issue the PERFORM JVMPOOL command to manually start a JVM with the profile), and then immediately remove the option from the profile.

If you are migrating JVM profiles which you set up in a previous CICS release, CICS accepts the old options LIBPATH, CLASSPATH, TMPREFIX, and TMSUFFIX and handles them appropriately. (The TMPREFIX and TMSUFFIX options were used to specify the trusted middleware class path, which is now obsolete.) Warning messages are issued at JVM startup if any of these options are found in the JVM profile, to prompt you to transfer them to an appropriate class path. Meanwhile, CICS combines the values of these options into the class paths it creates, together with the base class paths and the values of the new class path options.

Unique JVM number and other identifying information in output file names

When you use the &JVM_NUM; symbol in a value in a JVM profile or JVM properties file, such as an output file name, CICS substitutes the unique JVM number for the symbol at runtime. The &APPLID; symbol is used in the same way to include the CICS region applid in a value.As an alternative for `stdout` and `stderr` JVM output files, you can use the **-generate** option to include the unique JVM number, a time stamp, and the CICS region applid as part of the file name.

&JVM_NUM; symbol

At runtime, CICS replaces the &JVM_NUM; symbol with the JVM number, which is unique to the JVM. Using the unique JVM number means that you can distinguish output from each JVM from the output of other JVMs in the CICS region, and locate the output files for a JVM that is currently running.

The &JVM_NUM; symbol can be specified for any type of output from the JVM. The CICS-supplied sample JVM profiles demonstrate some possible uses for the &JVM_NUM; symbol:

- With the STDOUT and STDERR options, as part of the names of the z/OS UNIX files to be used for `stdout` (JVM output) and `stderr` (JVM error messages).
- With (for example) the JAVA_DUMP_TDUMP_PATTERN option, as part of the file name for TDUMPs from the JVM.

Using the &JVM_NUM; symbol guarantees that each JVM within the CICS region has its own unique output files during the lifetime of the CICS region.

The JVM number used for the &JVM_NUM; symbol is the same as the JVM number used on the EXEC CICS INQUIRE JVM and CEMT INQUIRE JVM commands to identify individual JVMs. You can use these commands to browse the JVMs in the JVM pool, identify their JVM numbers, and see which JVM is currently assigned to which task. If there is an issue with any task, you can use the relevant JVM number to locate the output files for the task's JVM.

&APPLID; symbol

At runtime, CICS replaces the &APPLID; symbol with the applid of the CICS region. The applid is always in upper case. The symbol can be specified for any type of output from the JVM.

Specifying the CICS region applid is helpful if you are using the same JVM profiles for multiple CICS regions. By using the &APPLID; symbol, you can share the same set of JVM profiles across CICS regions, and still have region-specific output destinations or working directories. You could use the symbol:

- With the WORK_DIR option, as part of the name of the working directory for the CICS region. This produces a different working directory for each CICS region. If you do this, ensure that you have created all the relevant directories on z/OS UNIX and given the CICS regions read, write and execute access to them.
- With the STDOUT and STDERR options, as part of the names of the z/OS UNIX files to be used for `stdout` (JVM output) and `stderr` (JVM error messages), in combination with the &JVM_NUM; option.
- With the JAVA_DUMP_TDUMP_PATTERN option, as part of the file name for TDUMPs from the JVM, again in combination with the &JVM_NUM; option.

Using the &APPLID; symbol in combination with the &JVM_NUM; symbol guarantees that output files are unique not only within the CICS region, but also across multiple CICS regions.

-generate option

The **-generate** option can be specified for the names of the z/OS UNIX files to be used for `stdout` (JVM output) and `stderr` (JVM error messages).

The **-generate** option appends the unique JVM number (as with the &JVM_NUM; symbol), the CICS region applid (as with the &APPLID; symbol), and also some additional qualifiers, to the file name that you have specified for the STDOUT or STDERR option in the JVM profile. In order, the qualifiers are:

- The applid of the CICS region.
- The unique JVM number.
- The time when the file was created (at JVM startup), in the form `yydddhhmmss`.
- The suffix `.txt`, a literal string suffix to indicate that the file contains readable data.

A typical output file name for a `stdout` file created with the **-generate** option might be:

```
dfhjvmout.IYK2ZIK1.0000000005.06004165342.txt
```

where:

- `dfhjvmout` is the fixed part of the file name.
- `IYK2ZIK1` is the applid of the CICS region.
- `0000000005` is the unique JVM number.
- `06004165342` is the time stamp showing when the JVM was created.
- `.txt` is the file suffix.

When you use the **-generate** option, the &APPLID; and &JVM_NUM; options are not required in the file name, because **-generate** supplies these pieces of information automatically.

Because the **-generate** option includes the JVM number, the resulting output file is unique to the JVM, and can be matched with the JVM number identified from the EXEC CICS INQUIRE JVM and CEMT INQUIRE JVM commands. Because it includes the CICS region applid, it is also unique across multiple CICS regions.

Standardization of JVM options

Before CICS Transaction Server for z/OS, Version 3 Release 2, you could only use a subset of the possible JVM options in a CICS JVM profile, and many of these options had to be coded in nonstandard ways so that CICS could recognize them. Now, you can specify any JVM option or system property in a JVM profile, and it is passed through to the JVM. You may specify the JVM options in the same format as for Java on other platforms, which helps to make Java configuration in CICS more consistent with Java configuration in other environments.

Some options in a JVM profile for CICS still take the form of a keyword and value separated by an = sign, and are not prefixed by a hyphen. These options are owned by CICS, and CICS makes use of their values during the setup process for the JVM. However, any option in a JVM profile that begins with a hyphen (-) character is passed through to the JVM without any parsing by CICS. These can be Java standard or nonstandard options, or Java system properties, and they are owned by the IBM JVM.

The options beginning with `X` that were used in JVM profiles before CICS Transaction Server for z/OS, Version 3 Release 2 should now be specified in the normal way for Java. For example, instead of specifying

```
Xmx=32M
```

you should now specify

```
-Xmx32M
```

The system properties that were used in JVM properties files before CICS Transaction Server for z/OS, Version 3 Release 2 should now be prefixed with `-D`, in the normal way for Java. For example, instead of specifying

```
java.security.policy=/usr/lpp/cicsts/cicsts32/lib/security/dfjejbpl.policy
```

you should now specify

```
-Djava.security.policy=/usr/lpp/cicsts/cicsts32/lib/security/dfjejbpl.policy
```

If you are using JVM profiles and properties files that you set up in a previous release, you do not need to migrate to the new style options immediately, because CICS still recognizes options coded in the old way.

Some of the options for JVM profiles and JVM properties files which were documented in the CICS documentation for previous CICS releases have now been removed from the CICS documentation. In most cases, the options are still valid, but because they can now be specified in the way that is normal for Java (rather than in a special way for CICS), the standard Java documentation can now be used. An example is the options relating to assertions.

There is no central repository of all options and system properties for the JVM. Some recommended sources of information are:

- The documentation for the IBM Software Developer Kit for z/OS, Java 2 Technology Edition, Version 1.4.2.
- *Persistent Reusable Java Virtual Machine User's Guide*, SC34-6201. This document lists command-line options, JVM options and system properties that are used in a JVM in a z/OS environment.
- The *IBM Developer Kit and Runtime Environment, Java 2 Technology Edition, Version 1.4.2 Diagnostics Guide*, SC34-6358, which is available to download

from www.ibm.com/developerworks/java/jdk/diagnosis/. This guide documents system properties that are used for JVM trace and problem determination.

The Java class libraries include other system properties, and applications might have their own system properties.

Because you can now specify any JVM option or system property in a JVM profile, not just the ones specifically recognized by CICS, it is technically not necessary to have a separate JVM profile and JVM properties file. You could specify all the options and system properties in the same file. However, to reduce your migration actions, you are recommended to continue with the existing structure of JVM profiles containing JVM options, which reference JVM properties files containing system properties. The CICS-supplied samples still work in this way.

UNIX System Services environment variables

In addition to the JVM options and system properties that are used in constructing the JVM, you can specify any UNIX System Services environment variables in a JVM profile. Any name and value pair in a JVM profile that is not recognized as a JVM option or system property, is treated as a UNIX System Services environment variable and is exported. UNIX System Services environment variables specified in a JVM profile apply only to JVMs created with that profile.

The JAVA_DUMP_OPTS and JAVA_DUMP_TDUMP_PATTERN options used in the CICS-supplied sample JVM profiles are UNIX System Services environment variables. Another example is the TZ environment variable, which can be specified to change the time zone for the JVM.

UNIX System Services environment variables can only be specified in a JVM profile, not in a JVM properties file.

Changes to CICS externals

Changes to installation

During CICS installation, a file CICS_INSTALL_OK is created in the CICS home directory of the z/OS UNIX file system. The CICS_INSTALL_OK file contains the CICS version and release number.

The name of the home directory is determined by the `pathprefix` and `ussdir` parameters of the DFHISTAR job. The default home directory name for CICS Transaction Server for z/OS, Version 3 Release 2 is `/usr/lpp/cicsts/cicsts32`. Subdirectories of this directory contain the CICS-supplied JVM profiles and properties files, and the CICS-supplied JAR files, among other items. In JVM profiles, this directory is identified by the CICS_HOME option.

When CICS receives a request to create a JVM and identifies the home directory from the CICS_HOME option in the JVM profile, it checks for the presence of the CICS_INSTALL_OK file in that directory, and checks that the version number is correct. If the CICS_INSTALL_OK file is not present, or the version number is incorrect, this indicates that the CICS_HOME option is incorrectly specified, and CICS does not start the JVM.

This is similar to the procedure for the IBM Software Developer Kit for z/OS, Java 2 Technology Edition, Version 1.4.2, which has a file JDK_INSTALL_OK in the JAVA_HOME directory of the z/OS UNIX file system.

Changes to options in JVM profiles and JVM properties files

There are a number of changed options in JVM profiles and JVM properties files as a result of the enhancements to Java 1.4.2 support.

All changes to options in JVM profiles and JVM properties files

All the changed options in JVM profiles and JVM properties files are detailed here. The table includes all the changes to the options that appeared in the CICS-supplied sample files in previous CICS releases.

Table 16. Changed options in JVM profiles and JVM properties files

Option	Status	CICS and Java launcher action	Replace with	Notes
REUSE=RESET	Obsolete	JVM does not start	REUSE=YES	CICS issues message DFHSJ0524 if found.
Xresettable=YES	Obsolete	JVM does not start	REUSE=YES	CICS issues message DFHSJ0525 if found.
ibm.jvm.crossheap.events	Obsolete	Java launcher ignores	n/a	Only used in resettable JVM.
ibm.jvm.events.output	Obsolete	Java launcher ignores	n/a	Only used in resettable JVM.
ibm.jvm.reset.events	Obsolete	Java launcher ignores	n/a	Only used in resettable JVM.
ibm.jvm.resettrace.events	Obsolete	Java launcher ignores	n/a	Only used in resettable JVM.
ibm.jvm.unresettable.events.level	Obsolete	Java launcher ignores	n/a	Only used in resettable JVM.
Xinitacsh	Obsolete	Java launcher ignores	Add value to -Xinitsh	Only used in resettable JVM.
Xinitth	Obsolete	Java launcher ignores	Add value to -Xms	Only used in resettable JVM.
TMPREFIX	Obsolete	CICS prefixes to shareable application class path	`-Dibm.jvm. shareable. application. class.path` system property if you have a shared class cache, CLASSPATH_PREFIX if you do not	CICS issues message DFHSJ0521 if found. Move classes with care.
TMSUFFIX	Obsolete	CICS places on shareable application class path	`-Dibm.jvm. shareable. application. class.path` system property if you have a shared class cache, CLASSPATH_SUFFIX if you do not	CICS issues message DFHSJ0522 if found.
MAX_RESETS_TO_GC	Obsolete	CICS ignores and uses default for GC_HEAP_THRESHOLD	GC_HEAP_THRESHOLD	CICS issues message DFHSJ0528 if found.
-generate (for STDOUT, STDERR)	Enhanced	Accepted	n/a	Now adds unique JVM number to generated output file names, in addition to CICS region applid, time stamp and suffix.

Table 16. Changed options in JVM profiles and JVM properties files (continued)

Option	Status	CICS and Java launcher action	Replace with	Notes
CICS_DIRECTORY	Renamed	CICS treats as CICS_HOME	CICS_HOME	CICS issues message DFHSJ0534 if found.
LIBPATH	Replaced by new equivalents	CICS treats as LIBPATH_ SUFFIX	LIBPATH_SUFFIX (LIBPATH_PREFIX also available)	CICS issues message DFHSJ0538 if found. You do not need to specify directories for base library path, only directories that you add.
CLASSPATH	Replaced by new equivalents	CICS treats as CLASSPATH_ SUFFIX	CLASSPATH_SUFFIX (CLASSPATH_PREFIX also available)	CICS issues message DFHSJ0523 if found.
VERBOSE	Withdrawn from sample profiles	Accepted	-verbose:gc	Works as before if specified in old format.
Xcheck (JVM default is NO)	Withdrawn from sample profiles	Accepted	-Xcheck	Only specify this if other than JVM default.
Xdebug (JVM default is NO)	Withdrawn from sample profiles	Accepted	-Xdebug (no value) to set debug on	Only specify this if other than JVM default.
Xnoclassgc (JVM default is NO)	Withdrawn from sample profiles	Accepted	-Xnoclassgc (no value) to specify no class garbage collection	Only specify this if other than JVM default.
Xverify (JVM default is remote)	Withdrawn from sample profiles	Accepted	n/a	Do not specify, use JVM default.
IDLE_TIMEOUT	New	Defaults to 30 minutes	n/a	Used to specify timeout threshold.
GC_HEAP_ THRESHOLD	New	Defaults to 85%	n/a	Used to specify heap utilization limit for CICS-scheduled garbage collection
CICS_HOME	New, replaces CICS_ DIRECTORY	Preferred	n/a	Used to specify home directory for CICS files in the z/OS UNIX file system.
CLASSPATH_PREFIX, CLASSPATH_SUFFIX	New, replace CLASSPATH	Preferred	n/a	Used for standard class path.
LIBPATH_PREFIX, LIBPATH_SUFFIX	New, replace LIBPATH	Preferred	n/a	Used for library path.
JAVA_DUMP_OPTS	New for CICS sample profiles	UNIX System Services environment variable set	n/a	Used to set dump options.
JAVA_DUMP_ TDUMP_PATTERN	New for CICS sample profiles	UNIX System Services environment variable set	n/a	Used to specify location for Java dumps.
DISPLAY_JAVA_ VERSION	New for CICS sample profiles	Preferred	n/a	Used to show JVM version in CICS MSGUSR log.

There is also a new symbol, &JVM_NUM;. When this symbol is used in a value in a JVM profile (for example, as part of the file name for a Java dump), CICS substitutes the unique JVM number for it at runtime. The new symbol can be specified for any type of output from the JVM, and it can be used in combination with the &APPLID; symbol (which provides the CICS region applid). The **-generate** option for `stdout` and `stderr` files also now adds the unique JVM number automatically.

Table 16 on page 181 lists only the options which were formerly used in the CICS-supplied sample files, together with the new options. Some options for JVM profiles and JVM properties files did not appear in the CICS-supplied sample files in previous CICS releases, but were documented in the CICS documentation. Some of these options have now been removed from the CICS documentation.

The `java.compiler` option has been undocumented because its primary use was to disable the Java just-in-time (JIT) compiler during the development process for applications in a resettable JVM. In a continuous JVM, this option is not required for that purpose.

The remaining undocumented options are still valid, but they can now be specified in the standard Java way (rather than in a special way for CICS), and so the documentation for the IBM Software Developer Kit for z/OS, Java 2 Technology Edition, Version 1.4.2 and other Java documentation can be used. If you have any of these options in an existing JVM profile for CICS, they are still accepted.

The main categories of valid options which have been undocumented are:

- The options relating to assertions. You can find more information about programming with assertions, and about enabling and disabling assertions, at `http://java.sun.com/j2se/1.4.2/docs/guide/lang/assert.html`.
- Various Java nonstandard options (beginning with `-X`), including `-Xmaxe`, `-Xmaxf`, `-Xmine`, `-Xminf`, `-Xrun`*`dllname`* and `-Xrs`. You can find more information about these options in *Persistent Reusable Java Virtual Machine User's Guide*, SC34-6201.
- Various JVM system properties, most of which should not be changed by users of the IBM JVM with CICS.

DFHJVMAT

DFHJVMAT is a user-replaceable program that you can use to override the options specified in a JVM profile. It can only be used for a single-use JVM, and not for a continuous JVM. The use of DFHJVMAT is not recommended for new development.

Only certain options in JVM profiles are available to DFHJVMAT. There are changes to the list of available options, as follows:

CICS_DIRECTORY
No longer available

CICS_HOME
New, replaces CICS_DIRECTORY

CLASSCACHE_MSGLOG
New

CLASSPATH
No longer available

CLASSPATH_PREFIX, CLASSPATH_SUFFIX
New, replace CLASSPATH

JAVA_DUMP_OPTS
New

LIBPATH
No longer available

LIBPATH_PREFIX, LIBPATH_SUFFIX
New, replace LIBPATH

TMPREFIX, TMSUFFIX
No longer available

Xresettable
No longer available

Several of the options available to DFHJVMAT are among the Java nonstandard options which have been undocumented. There is no further information about these options in the CICS documentation, and information about these can be found in the IBM Software Developer Kit for z/OS, Java 2 Technology Edition, Version 1.4.2 and other Java documentation.

Deprecated options due to configuration changes

These options in JVM profiles have been deprecated and replaced with new equivalents. If they are used, CICS issues a warning message and uses the value of the option in the correct way.

CICS_DIRECTORY
Specified the path for the home directory for CICS files in the z/OS UNIX file system. If this option is found, CICS issues the warning message DFHSJ0534, and treats the value of the option as the value of CICS_HOME. You should now use the CICS_HOME option to specify this path.

CLASSPATH
Specified the standard class path for the JVM. If this option is found, CICS issues the warning message DFHSJ0523, but still places the paths on the standard class path. You should now use the CLASSPATH_SUFFIX option to specify classes on this class path.

LIBPATH
Specified the library path for the JVM. If this option is found, CICS issues the warning message DFHSJ0538, and inserts the paths at the end of the library path, after the base library path. You should now use the LIBPATH_SUFFIX option to specify items that you added to the library path. Note that 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 part of the base library path built by CICS.

New options due to changes to configuration and problem determination

Some new options and a new symbol are available in the CICS-supplied JVM profiles as a result of changes to configuration and changes to assist with problem determination.

New options

The following options are now available in the CICS-supplied sample JVM profiles:

CICS_HOME=/usr/lpp/cicsts/cicsts32/
Specifies the path for the home directory for CICS files on z/OS UNIX. The value of the option is used to build the base library path and the base class path for the JVM. By default, this 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.

CLASSPATH_PREFIX
Adds directories to the beginning of the standard class path.

CLASSPATH_SUFFIX
Adds directory paths to the end of the standard class path. If you do not have a shared class cache, use the standard class path for all your application classes. If you have a shared class cache in your CICS region, using the standard class path increases storage usage, because the classes are stored in every JVM instead of just in the master JVM. Instead, you should normally use the shareable application class path.

LIBPATH_PREFIX
Adds directories to the beginning of the library path, before the base library path. Be aware that when you use this option, if DLL files in the specified directories have the same names as the CICS-supplied DLL files on the base library path, they are loaded in place of the CICS-supplied files.

LIBPATH_SUFFIX
Adds directories to the end of the library path, after the base library path. Use this option to specify directories containing any additional native libraries that are used by your applications, or by any services that are not included in the standard JVM setup for CICS. For example, the additional native libraries might include the DLL files needed to use the DB2 JDBC drivers. Any DLL files that you include on the library path for use by your applications should be compiled and linked with XPLink for optimum performance.

DISPLAY_JAVA_VERSION=
If this option is set to YES, whenever a JVM is started by an application, CICS writes message DFHSJ0901 to the MSGUSER log, showing the version and build of the IBM Software Developer Kit for z/OS, Java 2 Technology Edition which is in use.

JAVA_DUMP_OPTS=
A UNIX System Services environment variable. Specifies a set of Java dump options to obtain diagnostics for an abend in the JVM. Information about Java dump options can be found in the *IBM Developer Kit and Runtime Environment, Java 2 Technology Edition, Version 1.4.2 Diagnostics Guide*, SC34-6358, which is available to download from www.ibm.com/developerworks/java/jdk/diagnosis/.

JAVA_DUMP_TDUMP_PATTERN=
A UNIX System Services environment variable. Specifies the file name to be used for transaction dumps (TDUMPs) from the JVM. Java TDUMPs are written to a data set destination in the event of a JVM abend. You can use the symbols &APPLID; (CICS region applid) and &JVM_NUM; (unique JVM number) in this value, as shown in the CICS-supplied sample JVM profiles, to create unique dump file names for each JVM.

New symbol &JVM_NUM;

When you use the &JVM_NUM; symbol in a value in a JVM profile or JVM properties file, such as an output file name, CICS substitutes the unique JVM number for the symbol at runtime. As an alternative for `stdout` and `stderr` JVM output files, you can use the **-generate** option to include the unique JVM number, a time stamp, and the CICS region applid as part of the file name.

Using the unique JVM number means that you can distinguish output from each JVM from the output of other JVMs in the CICS region, and locate the output files for a JVM that is currently running.

The CICS-supplied sample JVM profiles demonstrate some possible uses for the &JVM_NUM; symbol.

Changes to the system programming interface

INQUIRE PROGRAM command

The USECOUNT option on the INQUIRE PROGRAM command now displays a use count for Java programs. In earlier CICS releases, this count was not available, and a value of -1 was returned.

Changes to CEMT

INQUIRE PROGRAM command

The Usecount option of the INQUIRE PROGRAM command now displays a use count for Java programs. In earlier CICS releases, this count was not available.

Changes to the CICSPlex SM programming interface

Changes to resource tables

In the PROGRAM resource table, the USECOUNT field now returns a use count for Java programs, as well as for other types of program. In earlier CICS releases, this count was not available, and a value of -1 was returned.

Changes to CICSPlex SM views and menus

In the ″Programs″ view, the field ″Total number of times program was executed″ now returns a use count for Java programs, as well as for other types of program. To access this view from the main menu, select **CICS operations views > Program operations views > Programs**.

Changes to problem determination

Messages

New messages are issued if CICS is attempting to start a JVM, and finds an unusable, ignored or deprecated option in a JVM profile or properties file, or encounters a problem with accessing one of the required directories. Messages relating to deprecated options are sent to CDEP, and other messages are sent to CSMT. The new messages are in the ranges DFHSJ0521–DFHSJ0526, and DFHSJ0531–DFHSJ0539.

The existing error message DFHSJ0513 no longer refers to the trusted middleware class path. It now refers to the standard class path, with text as follows:

DFHSJ0513

```
date time applid Unable to build class_path for JVM profile name:
{[CLASSPATH_PREFIX] [+] [CLASSPATH] [+] [CLASSPATH_SUFFIX] too long
| CICS_HOME not specified | JAVA_HOME not specified.} The JVM cannot
be started.
```

The existing message DFHSJ0901 is now issued at startup of a JVM if the DISPLAY_JAVA_VERSION option is specified in the JVM profile. It shows the current version of Java.

Abends

Abend ASJJ is issued if CICS is unable to read the JAVA_INSTALL_OK file, or the version number in the JAVA_INSTALL_OK file is not correct. The abend is preceded by one of the messages DFHSJ0531, DFHSJ0532, DFHSJ0533 or DFHSJ0900, which indicates the problem.

Trace

When CICS trace level 2 is set for the SJ domain, each time a JVM is started, the options in its JVM profile and JVM properties file are traced. The trace point is SJ 0536.

CICS now formats the output from the JVM trace facility, so you do not need to interpret the data manually. Each JVM trace point that is generated appears as an instance of a CICS trace point:

- SJ 4D02 is the trace point used for formatted JVM trace information.
- SJ 4D01 is used for any JVM trace points that cannot be formatted by CICS. If you see this trace point often, check that the TraceFormat.dat file supplied with the IBM Software Developer Kit for z/OS, Java 2 Technology Edition, Version 1.4.2 is present in the directory

  ```
  /usr/lpp/java142/J1.4/lib/
  ```

  where `/java142/J1.4/` is the path that is defined when you install IBM Software Developer Kit for z/OS, Java 2 Technology Edition, Version 1.4.2. CICS requires this file to format the JVM trace points.

Before CICS Transaction Server for z/OS, Version 3 Release 2, the trace point SJ 4D01 was used for all output from the JVM trace facility, but you should now expect to see SJ 4D02 instead.

Chapter 29. Improved scheduling for garbage collection in JVMs

Instead of performing garbage collection in a JVM after a specified number of Java program executions, CICS now schedules garbage collection when a specified percentage of the storage in the active part of the nonsystem heap is used. The garbage collection is carried out as a separate transaction, so it does not affect the statistics for user transactions.

CICS scheduling for garbage collection in JVMs

In earlier releases, CICS performed garbage collection in a JVM synchronously after a specified number of Java programs had been run, and the time spent in garbage collection was counted in the statistics for the last user transaction to run before garbage collection. Now, CICS initiates garbage collection in response to the use of storage in the nonsystem heap, and uses a separate transaction for the process, so that the user transaction statistics are not affected.

Garbage collection in JVMs can be triggered by an allocation failure, where a JVM runs out of space in the nonsystem heap and is unable to allocate any more objects. The JVM's Garbage Collector cleans up objects in the nonsystem heap that are no longer being referenced by applications, and frees some of the space. Garbage collection stops all other processes from running in the JVM for the duration of the garbage collection cycle.

As well as being triggered by allocation failures, garbage collection can be initiated by an application or by CICS. In CICS Transaction Server for z/OS, Version 3 Release 2, CICS initiates garbage collection when heap utilization in the active part of the nonsystem heap reaches a specified limit. The default is 85%, meaning that when 85% of the storage in the active part of the nonsystem heap is used, CICS schedules a garbage collection.

CICS checks heap utilization after every Java program execution. If the limit has been reached, the garbage collection transaction CJGC is scheduled to run in the JVM immediately after the current use of the JVM ends. Between these garbage collections, however, allocation failures could still occur if a Java program begins to run when heap utilization is below the limit, then uses all the remaining storage in the active part of the nonsystem heap, and still requires more storage.

The garbage collections scheduled by CICS are carried out as a separate system transaction, CJGC. Garbage collections caused by allocation failures, however, take place while an application is running in the JVM. If garbage collection takes place while an application is running, it delays the application, and it is counted in the CICS statistics for the user transaction.

In earlier CICS releases, the garbage collections scheduled by CICS took place after the application had finished running, but the process still took place under the user transaction and affected the statistics.

Depending on your performance goals, you might want to minimize task response times by setting the GC_HEAP_THRESHOLD option for the JVM so that garbage collection is normally initiated by CICS rather than being caused by allocation failures. You can change the heap utilization limit to any level that is appropriate for your applications.

If you do not want CICS to initiate garbage collection, you can set GC_HEAP_THRESHOLD to 100. If you do this, all garbage collections result from allocation failures while applications are running.

Changes to CICS externals

Changes to options in JVM profiles and JVM properties files

There are a number of changed options in JVM profiles and JVM properties files as a result of the enhancements to Java 1.4.2 support.

Deprecated option for garbage collection

The MAX_RESETS_TO_GC option is now deprecated.

MAX_RESETS_TO_GC

Specified the number of transactions after which CICS scheduled a garbage collection in the JVM. If this option is found, CICS issues the warning message DFHSJ0528, and ignores the option. You should now use the GC_HEAP_THRESHOLD option to specify a percentage heap utilization threshold after which CICS schedules a garbage collection.

New option for garbage collection

The GC_HEAP_THRESHOLD option is now available in the CICS-supplied sample JVM profiles.

GC_HEAP_THRESHOLD=

Specifies the heap utilization limit for the JVM’s nonsystem heap. When this percentage of the storage in the active part of the nonsystem heap is used, CICS schedules a garbage collection. CICS checks heap utilization after every Java program execution. If the limit has been reached, the garbage collection transaction CJGC is scheduled to run in the JVM immediately after the current use of the JVM ends.

The default heap utilization limit is 85 (85%). The minimum is 50. The maximum if you want CICS to schedule garbage collections is 99. If you specify a heap utilization limit of 100, CICS never schedules garbage collections, and all garbage collections result from allocation failures while applications are running.

This option is irrelevant for a single-use JVM, which is destroyed after a single Java program has run in it.

Changes to problem determination

Messages

Message DFHSJ0528 is issued if the MAX_RESETS_TO_GC option is used in the JVM profile. The option is ignored.

Message DFHSJ0529 is issued if the value specified for GC_HEAP_THRESHOLD is not between 50 and 100. The default value of 85 is applied and profile processing continues.

Trace

Trace point SJ 0226 indicates that CICS has requested garbage collection in the JVM.

Trace point SJ 0538 is issued if the call to attach the garbage collection transaction CJGC fails. In this case, garbage collection is performed as part of the current user transaction, rather than as a separate transaction.

Abends

Abend ASJK is issued if transaction CJGC is started incorrectly, for example by terminal input.

Abend ASJL is issued if program DFHSJGC is started under the wrong transaction code (that is, not CJGC).

Security

There is a new CICS-supplied transaction CJGC, the CICS JVM garbage collection transaction. This is a Category 1 transaction.

Chapter 30. JVM startup, termination and timeout control

You can now control startup and timeout for JVMs (Java Virtual Machines). You can start JVMs manually using the PERFORM JVMPOOL command, in addition to those started by CICS. You can also change the timeout threshold in the JVM profile for your JVMs, so that idle JVMs do not have to become eligible for termination after 30 minutes of inactivity as at present, but can continue to exist for a specified length of time up to 7 days, or never time out. These functions give you greater control over the availability of your JVMs to meet peak demand. The JVM termination facility has also been refined. When you make changes to a JVM profile or a shared Java class, you can now implement these by terminating only the appropriate subset of JVMs in the pool, rather than by terminating the whole JVM pool.

Starting JVMs manually

You can use the manual startup facility to start up JVMs immediately after terminating them, or to create JVMs in advance of application requests.

CICS normally manages the startup and termination of JVMs, in order to achieve a balanced level of capacity in the JVM pool to meet the demand from applications. CICS has sophisticated mechanisms to manage the number and type of JVMs in the pool, particularly when there is a need to optimize the performance of complex workloads at times of peak demand.

You might want to start up JVMs manually in certain situations:

- If you make changes to your JVM profiles or JVM properties files (including adding new classes or JAR files to class paths) or change shareable application classes while CICS is running, you need to terminate the JVMs that are affected. When new JVMs are started to replace them, the new JVMs implement your changes. You can start new JVMs manually, or let CICS do this automatically.
- If your Java workload is regular, predictable, and involves a limited number of different JVM profiles, you could consider starting up JVMs in advance of the demand from applications, so that they are ready for use as soon as they are required.

To start up JVMs manually, use the EXEC CICS or CEMT PERFORM JVMPOOL command. You need to specify the number of JVMs to be started, and the JVM profile and execution key that is to be used for them.

The number that you specify, added to the number of JVMs that already exist in the JVM pool, must not exceed the MAXJVMTCBS limit for the CICS region. You can check this by issuing the EXEC CICS or CEMT INQUIRE DISPATCHER command. MAXJVMTCBS shows the limit, and ACTJVMTCBS shows the number of JVMs that currently exist.

CICS does not start all the JVMs at once, but schedules the starts over a short period of time. Each JVM is available for use by an application as soon as it has been started. If a JVM is not used by an application, then like any other idle JVM, it becomes eligible for automatic termination at the timeout threshold that you have specified in the JVM profile.

If you have just terminated JVMs in order to implement changes to JVM profiles, and application activity in the CICS region is low, you can use the PERFORM

JVMPOOL command to start a JVM of the type where you applied the changes. This enables you to confirm, without waiting for an application request, that the JVM is able to start with the changed profile, and that the classes specified on your class paths can be loaded.

If the Java workload in your CICS region is regular and predictable, you might want to use the manual startup facility to create a JVM pool that anticipates the needs of your applications, rather than allowing CICS to do this in response to demand. This strategy might reduce the delay time for applications in periods when workload is increasing.

By configuring the timeout threshold (which defaults to 30 minutes), and starting up JVMs in advance of need, you could structure a JVM pool that always has enough capacity available for your requirements. For example, you could start up a sufficient number of JVMs to handle your peak workloads, with their timeout thresholds set so that they are only eligible for automatic termination after 24 hours of idleness. (You might want to set up a task that starts the appropriate number of JVMs when the CICS region is started.) With a JVM pool like this, CICS would not terminate the JVMs automatically at times of the day when the workload is reduced. They would only be terminated if the system was idle for an extended period, or if your workload reduced over the long term.

When you start up JVMs manually with a particular JVM profile, they are eligible for mismatching or stealing in the same way as JVMs started by CICS. Mismatching and stealing change the JVM profile or user key, so the JVM can no longer be used by the applications for which you originally started it up. Mismatching and stealing also involve restarting the JVM, which can negate any benefit you experience from starting the JVMs in advance. The possibility of mismatching and stealing increases with the number of different JVM profiles in the CICS region, so if you want to structure a JVM pool manually, the benefit is likely to be greatest if your applications use only one or a small number of JVM profiles.

Terminating JVMs

To terminate JVMs, use the PERFORM JVMPOOL TERMINATE system programming command, or the CEMT PERFORM JVMPOOL with the PHASEOUT, PURGE, or FORCEPURGE option. You can either choose to terminate all the JVMs in the JVM pool, as in earlier CICS releases, or you can now specify a JVM profile to terminate only the JVMs with that profile.

You need to terminate JVMs to implement changes to JVM profiles or to add new application classes. You also need to terminate JVMs to refresh shared Java classes (those on the shareable application class path). Changes to existing classes on the standard class path do not require termination of the JVMs. The standard class path, rather than the shareable application class path, is the recommended choice for standalone JVMs, but if you are in the process of migrating from resettable to continuous JVMs, you might still have classes on the shareable application class path in standalone JVMs.

The PERFORM JVMPOOL command does not terminate the shared class cache and the master JVM. If the CICS region has a shared class cache, and you want to update classes on the shareable application class path for worker JVMs, you need to use the EXEC CICS or CEMT PERFORM CLASSCACHE command to terminate or reload the shared class cache. The command also terminates the worker JVMs. If autostart is enabled, a new shared class cache is started as soon as it is required. Otherwise you need to start it manually.

To minimize disruption to your applications, try to terminate only those JVM profiles where you have made changes to the JVM profile, its associated JVM properties file, or the applications that use it. Terminating a subset of the JVM pool is more efficient than terminating the whole JVM pool. Make sure that you do terminate all the JVMs affected by your changes. For example, a shared Java class which you have changed might be listed on the shareable application class path in more than one JVM profile. In certain unusual circumstances, an application class might be used by JVMs with more than one profile, but this might not be obvious from the JVM profiles. This might be an issue, for example, if you use custom classloaders, or instantiate classes through reflection, or have enterprise beans which call other enterprise beans. If you are not sure whether an application class is used by JVMs with more than one profile, you might prefer to be safe and terminate the whole JVM pool.

CICS starts up new JVMs as soon as it receives requests from applications for each type of JVM. If you prefer, you can start JVMs manually using the PERFORM JVMPOOL command. If you have made any changes to the JVM profiles, the new JVMs use the changed options. If you have made any changes to your Java applications, the new JVMs load the new or changed classes.

Timeout threshold for JVMs

The new IDLE_TIMEOUT option in JVM profiles lets you specify how long an inactive JVM can remain in the JVM pool.

If there are too many JVMs in the JVM pool waiting to be reused, and the workload does not require them, CICS terminates them automatically. If a JVM is not used by any application during the period of time specified in the IDLE_TIMEOUT option in its JVM profile, it becomes eligible for automatic termination. The next time CICS checks on the idle JVMs, some of the JVMs that have reached their timeout thresholds and are still idle will be destroyed, together with their TCBs.

CICS does not immediately terminate all of the JVMs that have timed out; instead, they are terminated progressively over a period of time, so that a balanced level of capacity is maintained in the JVM pool. JVMs that have timed out and have not yet been terminated are still available to be reused by applications if there is an increase in demand, and if a JVM is reused it ceases to be eligible for automatic termination. CICS never automatically terminates the last JVM in the JVM pool.

You need to choose an appropriate IDLE_TIMEOUT value for JVMs with each JVM profile. You might prefer CICS to terminate inactive JVMs more quickly in order to free up system resources, and create new JVMs if there is an increase in demand from Java applications. In this case you should select a shorter timeout threshold. Alternatively, you might prefer CICS to keep unused JVMs available for a longer period, so that capacity is always available to meet your peak workloads, without incurring the CPU costs of JVM startup. In this case, you should select a longer timeout threshold.

The default timeout threshold is 30 minutes. You can specify a longer timeout threshold of up to 7 days. You can also specify a timeout threshold of zero, which means that JVMs with that profile are never terminated automatically because of inactivity. Under normal conditions, JVMs with a timeout threshold of zero are only terminated if they are selected for stealing or mismatching.

The process of automatic termination of inactive JVMs operates when conditions in the CICS region are normal. If MVS storage becomes constrained or severely constrained, CICS takes immediate action to manage that situation. During that process unused JVMs are destroyed regardless of their timeout thresholds, even if the timeout threshold is zero.

Changes to CICS externals

Changes to options in JVM profiles and JVM properties files

There are a number of changed options in JVM profiles and JVM properties files as a result of the enhancements to Java 1.4.2 support.

New option for timeout control

A new IDLE_TIMEOUT option is available in JVM profiles to specify the timeout threshold for JVMs.

IDLE_TIMEOUT={30|*number*}

Specifies the timeout threshold, in minutes, for JVMs with this JVM profile. If a JVM is inactive (not used by an application) for the specified amount of time, it becomes eligible for automatic termination. The next time CICS checks on the idle JVMs, if the JVM is still inactive, the JVM and its J8 or J9 TCB might be destroyed. (CICS does not immediately terminate all of the JVMs that have timed out; they are terminated progressively over a period of time.)

The default timeout threshold is 30 minutes, and the maximum is 10080 minutes (7 days). You can also specify a timeout threshold of zero, which means that JVMs with that profile are never terminated automatically because of inactivity. (JVMs with a timeout threshold of zero may be terminated if they are selected for stealing or mismatching, or if MVS storage becomes constrained or severely constrained.) If you specify an unacceptable value, CICS uses the default instead.

This option is ignored in the JVM profile for the master JVM, because the master JVM is never terminated automatically by CICS. It is also irrelevant for a single-use JVM, which is destroyed after a single Java program has run in it.

The IDLE_TIMEOUT option is included in the appropriate sample JVM profiles: DFHJVMPR (for a standalone JVM), DFHJVMPC (for a worker JVM), and DFHJVMCD (for CICS-supplied system programs).

Canges to the system programming interface

New SPI commands

PERFORM JVMPOOL

Start and terminate JVMs in the JVM pool.

For details of the command, see “PERFORM JVMPOOL” on page 312.

SET JVMPOOL command

The TERMINATE option on the SET JVMPOOL command is deprecated. You should use the PERFORM JVMPOOL command instead.

Changes to CEMT

New CEMT commands

PERFORM JVMPOOL

Start and terminate JVMs in the JVM pool.

For details of the command, see “CEMT PERFORM JVMPOOL” on page 330.

SET JVMPOOL command

The TERMINATE option SET JVMPOOL command is now deprecated. You should use the PERFORM JVMPOOL command instead.

Changes to the CICSPlex SM programming interface

Changes to resource tables

The JVMPOOL resource table has a new Initialize action to start JVMs manually. The existing Force purge, Phase out, and Purge actions can now be carried out either for a selected JVM profile, or for the whole JVM pool.

Changes to CICSPlex SM views and menus

In the Java virtual machine (JVM) pool view, there is a new Initialize action button, which enables you to start JVMs manually. The existing Force purge, Phase out, and Purge actions can now be carried out either for a selected JVM profile, or for the whole JVM pool. To access this view from the main menu, select **CICS operations views** → **Enterprise Java component operations views** → **Java virtual machine (JVM) pool**.

Changes to problem determination

Messages

Message DFHSJ0530 is issued if the value specified for IDLE_TIMEOUT is not in the range 0–10080. The default timeout threshold of 30 minutes is applied and JVM profile processing continues.

Trace

Trace point SJ 0537 indicates a problem in setting the timeout threshold

Trace point SJ 0538 is issued if the call to DFHDSIT to set the IDLE_TIMEOUT for the JVM fails. The timeout remains at the default value.

Security

There is a new CICS-supplied transaction CJPI, which starts up new JVMs as a result of a PERFORM JVMPOOL command. This is a Category 1 transaction.

Part 4. CICS service management: CICSPlex SM improvements

CICS Transaction Server for z/OS, Version 3 Release 2 delivers a set of capabilities which provide customer value by enabling business flexibility through IT simplification. These capabilities are represented in three themes:

- application connectivity
- application reuse
- service management

The capabilities represented by the *service management* theme enable you to effectively manage large runtime configurations using modern user interfaces, so that you can meet demanding service level and IT governance objectives. CICS Transaction Server for z/OS, Version 3 Release 2 provides enhancements to CICSPlex SM functions that support these goals.

Chapter 31. Integrated installation of CICSPlex SM

The installation of CICSPlex SM is integrated with the installation of CICS.

The following improvements reduce the complexity of installing and configuring CICSPlex SM:

- You can now edit the DFHISTAR job to modify the CICS and CICSPlex SM installation parameters for your environment. You no longer need to edit, separately, a EYUISTAR job. You modify and submit one set of input parameters in the DFHISTAR job.. DFHISTAR produces customized JCL for CICS and CICSPlex SM.
- You can create dynamically, during initialization and when a CICSPlex SM system is started by a transaction, CICS resource definitions for CICSPlex SM objects. You no longer need to manipulate the CSD files (using DFHCSDUP) to create the resource definitions necessary for the CMAS, MAS, and WUI.
- You can now run the EYU9XDUT utility to create the definitions required to start a WUI and its CICSplex. You would previously have had to use the end-user interface or a batch utility to create such definitions.

Using DFHISTAR to define CICSPlex SM installation parameters

The DFHISTAR job now has a single set of installation parameters for CICS and CICSPlex SM. You no longer need to edit, separately, a EYUISTAR job.

DFHISTAR produces customized JCL for CICS and CICSPlex SM. DFHISTAR customizes samples 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)

EYUISTAR is no longer available as a job to modify CICSPlex SM installation parameters.

CICSPlex SM parameters in DFHISTAR

A number of parameters in DFHISTAR apply only to CICSPlex SM. Other apply to both CICSPlex SM and CICS.

DFHISTAR parameters that apply to CICSPlex SM only

CMASNAME name

Specifies the 1- to 8-character name to be allocated to a CMAS. The default is CMAS01. The name can contain alphabetic, national, and numeric characters. However, the first character must be alphabetic or national.

The name of a CMAS must be unique within the CICSPlex SM environment. It should not be the same as the name of another CMAS, a CICSplex, a CICS system, or a CICS system group.

CSYSNAME name

This parameter is introduced to specify the 1- to 8-character name to be allocated to a MAS. The default is CSYS01. The name can contain alphabetic, national, and numeric characters. However, the first character must be alphabetic or national.

CSYSPLEX value
Specifies the 1- to 8-character name to be allocated to a CICSplex of managed systems. This identifier can contain alphabetic, national, and numeric characters. The default is CSYPLX01.

OLDDREP dsname
Identifies an existing data repository that is being used by a previous release of CICSPlex SM. The records in the existing data repository are migrated to a new data repository for CICS TS for z/OS, Version 3.2. The existing data repository is not modified. If you do not specify this parameter, a new data repository is created.

CMSSYSID value
Specifies the 4-character system identifier of the CMAS. The default is CM01.

TCPIPHST
Specifies the 1- to 8-character name allocated to the TCP/IP host name for the WUI server. The default is XXXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX.

TCPIPPRT
Specifies the 1- to 8-character identifier allocated to the TCP/IP port number for the WUI server. The identifier can contain numeric characters only, in the range 1 to 65535. The default is 12345.

TIMEZONE code
Specifies the time zone assigned to the data repository. The default is B.

WUI YES|NO
This parameter is introduced to specify whether to create a WUI plex. The default is YES.

WUINAME name
This parameter is introduced to specify the 1- to 8-character name to be allocated to a WUI. The default is created from the characters WUIN, followed by the CMSSYSID. For example, the default for CMSSYSID is WUINCM01. The name can contain alphabetic, national, and numeric characters. However, the first character must be alphabetic or national.

WUIPLEX name
This parameter is introduced to specify the 1- to 8-character name to be allocated to a WUI plex. The default is created from the characters WUIP, followed by the CMSSYSID. For example, the default for CMSSYSID is WUIPCM01. The name can contain alphabetic, national, and numeric characters. However, the first character must be alphabetic or national.

WUISYSID name
Specifies the 1- to 4-character name allocated to a WUI system identifier. The name can contain alphabetic, national, and numeric characters. However, the first character must be alphabetic or national. The default is WU01.

DFHISTAR parameters that apply to CICSPlex SM and CICS

A number of parameters in the DFHISTAR job are now applicable to CICSPlex SM as well as CICS. These are:

BLKU blocksize
Indicates the block size to be used when allocating data sets that have an UNDEFINED record length. The default is 32760.

DEFVOL volume disktype
Defines the default disk on which the contents of the disk volumes CMACVOL, DISTVOL, OPTVOL, SMPVOL, and TARGVOL will reside if the appropriate parameter is not coded in the DFHISTAR job. For example, if you do not code

the DISTVOL parameter, the CICSPlex SM distribution libraries will reside on the disk defined by DEFVOL. The default is CICS32 SYSALLDA.

volume
is one of the following:
- The volume serial identifier, in the range 1 through 6 characters, of the default volume.
- A period (.) if all volumes other than CMACVOL and SMPVOL that are not specifically defined by the appropriate parameter of the DFHISTAR job will be put onto any available volume. The CMACVOL and SMPVOL volumes will be put onto the same volume as the library specified by the TEMPLIB parameter.

disktype
is the UNIT parameter of the volume.

DINDEX library_prefix
Assigns a high-level index to the CICS Transaction Server SMP/E distribution libraries (except for the SDFHLINK and SDFHLPA target libraries) allocated by the installation process.

The *library_prefix* value must not be longer than 26 characters, and the leading character must be alphabetic. If you specify more than one level of index, the names must be separated by a period (for example, `DINDEX CICSTS32.TEST`). The default is CICSTS32.

DISTVOL volume disktype
Defines the disk on which the CICS Transaction Server distribution libraries will reside.

volume
is one of the following:
- The volume serial identifier, in the range 1 through 6 characters, of the volume on which the distribution libraries will reside.
- A period (.) if the CICS Transaction Server libraries are to be put onto any available volume.

disktype
is the UNIT parameter of the volume.

DSINFO dsindex volume disktype qualifier
Specifies details of the data sets to be created when you run the post-installation jobs:

dsindex
Assigns a high-level index to all the data sets defined by the post-installation jobs.

The leading character of *dsindex* must be alphabetic. *dsindex* can have one or two levels of index, but each level must be no longer than eight characters. If you specify more than one level of index, the names must be separated by a period (for example, `data.set.index`).

volume
is the volume identifier of the volume.

disktype
is the UNIT parameter for the volume.

qualifier
CICS element only. A partial qualifier added to the index for the data sets created by the jobs DFHCOMDS and DFHDEFDS. You can specify a partial qualifier of up to four alphanumeric characters; these characters are appended to the characters CICS to make the qualifier. If you specify a period (.) no qualifier is used.

The default is CICSTS32 CICS32 3390.

GINDEX library_prefix

Assigns a high-level index to the CICS Transaction Server SMP/E global libraries (except for the SDFHLINK and SDFHLPA target libraries) allocated by the installation process.

The *library_prefix* value must not be longer than 26 characters, and the leading character must be alphabetic. If you specify more than one level of index, the names must be separated by a period (for example, `GINDEX CICSTS32.TEST`). The default is CICSTS32.

GZONECSI cluster NEW|OLD volume disktype

Specifies details of the global zone CSI.

cluster

is the VSAM cluster name, minus the qualifier '.CSI'.

NEW|OLD

Specifies whether an existing global zone CSI is to be used. If you specify NEW, any existing global zone CSI with the specified *cluster* name is deleted, and a new global zone CSI is allocated. If you specify OLD, an existing global zone CSI is used.

volume

is either the volume serial (volser) identifier for the volume on which the global zone CSI is to be allocated or a period (.) if the CSI is to be put on a volume determined by the CICS Transaction Server installation process.

disktype

is the UNIT parameter for the volume.

The default is CICSTS32.GZONE.NEW CICS32 3390.

JOB accounting_information

Specifies the JOB statement and accounting information that you want substituted into the jobs generated by the DFHISTAR job. For example:

```
JOB //XXXXXXXX JOB 1,userid,MSGCLASS=A,MSGLEVEL=(1,1),
JOB //         CLASS=A,NOTIFY=userid
JOB /*JOBPARM SYSAFF=node1
JOB /*ROUTE PRINT node2.userid
```

Note:

1. Do not change `XXXXXXXX` given in the sample JOB statement in the DFHISTAR job. This is the 8-character job name that is substituted by the DFHISTAR job.
2. Normal JCL rules for coding JOB statements apply to the JOB parameter.
3. Delete (or comment out) extra lines of the JOB statement that you do not need.
4. Normal JCL rules apply when coding the JOB statement (for example, all lines except the last line must end in a comma).

LIB

Specifies the name of the installation output library to which the jobs generated by the DFHISTAR job are added. The default is CICSTS32.XDFHINST.

LINDEX library_prefix

Assigns a high-level index to the SDFHLPA, SDFHLINK, SDFJLPA, SEYULINK and SEYULPA libraries allocated by the installation process. The *library_prefix* value must be defined in the MVS Master Catalog.

The *library_prefix* value must not be longer than 26 characters, and the leading character must be alphabetic. If you specify more than one level of index, the names must be separated by a period (for example, `LINDEX SYS1.TEST`). The default is SYS1.CICSTS32.

SCEECICS dsname

Specifies the full data set name, up to 44 characters, of the SCEECICS library. This library is accessed, as read-only, during the installation of CICS Transaction Server. The default is SYS1.SCEECICS.

SCEERUN dsname

Specifies the full data set name, up to 44 characters, of the SCEERUN library. This library is accessed, as read-only, during the installation of CICS Transaction Server. The default is SYS1.SCEERUN.

SCEERUN2

Specifies the full data set name, up to 44 characters, of the SCEERUN2 library. This library is accessed, as read-only, during the installation of CICS Transaction Server. The default is SYS1.SCEERUN2.

SCEESAMP

Specifies the name of the Language Environment® library that contains the CEECCSD member. The default is SYS1.SCEESAMP.

SCOPE ALL|BASE|POST

Specifies whether you want to generate all the CICS Transaction Server installation and post-installation jobs, or only the post-installation jobs. When installing CICS Transaction Server from the distribution tape, you would normally specify `SCOPE ALL` (the default).

ALL

Specifies that you want to generate all the CICS Transaction Server installation jobs and all the post-installation jobs.

BASE

CICS element only. Specifies that you want to generate only the installation jobs (DFHINST1 through DFHINST6, DFHIHFS0, DFHIHFS1, and DFHISMKD) that you use to install CICS Transaction Server from the distribution tape.

POST

Specifies that you want to generate only the post-installation jobs, that you can use to create the CICS Transaction Server data sets, and run the IVPs.

SELECT jobname newname

Specifies the new name for a copy of a post-installation job to be generated when you run the DFHISTAR job. You can specify several SELECT parameters to select several post-installation jobs to be regenerated in one run of the DFHISTAR job. The SELECT parameter overrides the POST parameter; that is, if you use the SELECT parameter in the DFHISTAR job, only those jobs specified by SELECT are generated. There is no default value.

SMPVOL volume disktype

Specifies the disk that contains the permanent, non-VSAM SMP/E data sets for CICS Transaction Server that are associated with global or distribution zones, and are therefore unique.

volume

is one of the following:

- The volume serial identifier, in the range 1 through 6 characters, of the volume on which the permanent non-VSAM SMP/E data sets are to reside.
- A period (.) if the permanent non-VSAM SMP/E data sets are to be put onto the same volume as the library specified by the TEMPLIB parameter.

disktype

is the UNIT parameter for the volume.

Note: If you omit the SMPVOL parameter, the permanent non-VSAM SMP/E data sets for CICS Transaction Server will be put on the volume specified by the DEFVOL parameter. If the DEFVOL parameter is omitted, or if a period (.) is specified for its *volume* operand, the data sets will be put onto the same volume as the library specified by the TEMPLIB parameter.

SMPWORK disktype

Specifies the UNIT parameter for the disk that is to contain the temporary SMP/E work data sets (SMPWRK1, SMPWRK2, SMPWRK4, and SMPWRK6) needed to install CICS Transaction Server. The default is SYSALLDA.

TARGVOL volume disktype

Specifies details of the disk containing the CICS Transaction Server target libraries.

volume

is one of the following:

- The volume serial identifier, in the range 1 through 6 characters, of the volume on which the CICS Transaction Server target libraries are to reside.
- A period (.) if the CICS Transaction Server target libraries are to be put onto any available volume.

disktype

is the UNIT parameter for the volume.

Note: If you omit the TARGVOL parameter, the CICS Transaction Server target libraries will be put onto the volume specified by the DEFVOL parameter. If the DEFVOL parameter is omitted, or if a period (.) is specified for its *volume* operand, the CICS Transaction Server target libraries will be put onto any available volume.

TEMPLIB library_name

This specifies the name of the temporary installation library that contains the skeleton installation jobs. The default is CICSTS32.TDFHINST.

Specify this name on the SYSPROC DD statement of the DFHISTAR job.

TINDEX

Assigns a high-level index to the CICS Transaction Server SMP/E target libraries (except for the SDFHLINK, SDFHLPA, SDFJLPA, SEYULINK, and SEYULPA target libraries) allocated by the installation process.

Note:

1. The high-level index for the SDFHLINK and SDFHLPA libraries is defined by the LINDEX parameter.
2. The high-level index for the data sets created by the jobs DFHCOMDS and DFHDEFDS is defined by the *dsindex* operand of the DSINFO parameter.

The *library_prefix* value must not be longer than 26 characters, and the leading character must be alphabetic. If you specify more than one level of index, the names must be separated by a period (for example, `TINDEX CICSTS32.TEST`). The default is CICSTS32.

TZONE zonename

Specifies the name of the target zone to be used by SMP/E. This name must be unique to the target zone. It must not be longer than seven characters and the leading character must be alphabetic. The default is TZONE.

UTILITIES asmprog binder smpeprog copyutil

Specifies the names of utility programs to be used when installing CICS Transaction Server and programs that it uses.

asmprog

is the program name of the assembler. Specify ASMA90, for High Level Assembler/MVS & VM & VSE, which is required.

binder
is the program name of the linkage editor. Ensure that program IEWL references the z/OS program management binder.

smpeprog
is the program name of the SMP/E program. The IBM-supplied name is GIMSMP.

copyutil
is the program name of the data set copy utility program. The IBM-supplied name is IEBCOPY.

The default is ASMA90 IEWL GIMSMP IEBCOPY.

WORKUNIT
Specifies the UNIT parameter for the disk or disks on which work data sets are stored. The default is SYSDALLDA.

Obsolete DFHISTAR parameters

BLKISPF is now obsolete in the DFHISTAR job.

CICSPlex SM data set definition and startup jobs

A description of the JCL procedures generated by DFHISTAR for CICSPlex SM installation.

DFHISTAR generates a number of jobs to help you install the CICSPlex SM element of CICS Transaction Server:

EYUCMSDS
Creates and initializes the data sets required for a CMAS. It:

1. Creates the CICS data sets for the CMAS.
2. Creates the CMAS data repository.
3. Initializes the data repository using the EYU9XDUT utility.
4. Creates a CSD. This is shared by all the CICS regions and the region qualifier appears in the data set name. You can comment out this step if your CMAS is sharing an existing CSD.

EYUWUIDS
Creates and initializes the data sets required for a WUI. It:

1. Creates the CICS data sets for the WUI.
2. Creates the WUI data respository.
3. Initializes this data repository.

EYUCSYDS
Creates and initializes the data sets required for a managed CICS system (MAS). It:

1. Creates the CICS data sets for the MAS.
2. Includes a step to create a pair of history data sets. You can delete this step if you do not want to use the history function.

EYUCMASP
Starts a CMAS. This CMAS uses the CICS-supplied sample table, DFHSIT6$, but appropriate override values are supplied in the job.

EYUWUIP
Starts a WUI. This WUI uses the CICS-supplied sample table, DFHSIT6$, but appropriate override values are supplied in the job.

EYUCSYSP
Starts a MAS. This MAS uses the CICS-supplied sample table, DFHSIT6$, but appropriate override values are supplied in the job.

EYUCMASJ
JCL to start a CMAS. It executes EYUCMASP.

EYUWUIJ
JCL to start a WUI. It executes EYUWUIP.

EYUCSYSJ
JCL to start a MAS. It executes EYUCSYSP.

EYUCMSSP
CICS SIT override for a CMAS.

EYUWUISP
CICS SIT override for a WUI.

EYULMSSP
CICS SIT override for a MAS.

EYUCMS0P
EYUPARM parameters for a CMAS.

EYUWUI0P
EYUPARM parameters for a WUI.

EYULMS0P
EYUPARM parameters for a MAS.

EYUWUIIN
EYUWUI parameters for a WUI.

EYUJWREP
JCL to delete and define a WUI data repository (this function is also included in EYUWUIDS).

EYUJHIST
JCL to delete and define a pair of history data sets.

EYULPMOD
JCL for applying user modifications (USERMODs). It moves some MAS load modules to a link pack area library.

CICSPlex SM post-installation members

DFHISTAR allows you to customize post-installation members, so that you can create a simple CICSPlex SM configuration. The post-installation members are listed according to area: Members for a CMAS, a WUI and a managed CICS system (MAS).

A number of CICSPlex SM post-installation members are delivered, as skeletons, in the TDFHINST library. When you run DFHISTAR, the post-installation members are customized and saved in the XDFHINST library.These members allow you to create a simple CICSPlex SM configuration that consists of a CMAS, a WUI and a managed CICS system (MAS).

Post-installation members are split into three areas as shown in Table 17 on page 209, Table 18 on page 209 and Table 19 on page 209.

Table 17. Post-installation members for a CMAS

Member	Description
EYUCMASJ	JCL to start a CMAS. It executes EYUCMASP.
EYUCMASP	Starts a CMAS. This CMAS uses the CICS-supplied sample table, DFHSIT6$, but appropriate override values are supplied in the job.
EYUCMS0P	EYUPARM parameters for a CMAS.
EYUCMSDS	JCL to create and initialize the data sets for a CMAS.
EYUCMSSP	CICS SIT overrides for a CMAS.

For more information on CMAS data set customization, see CMAS data set creation and customization.

Table 18. Post-installation members for a WUI

Member	Description
EYUJWREP	JCL to delete and define a WUI data repository (this function is also included in EYUWUIDS).
EYUWUI0P	EYUPARM parameters for a WUI.
EYUWUIDS	JCL to create and initialize the data sets for a WUI.
EYUWUIIN	EYUWUI parameters for a WUI.
EYUWUIJ	JCL to start a WUI. It executes EYUWUIP.
EYUWUIP	Starts a WUI. This WUI uses the CICS-supplied sample system initialization table, DFHSIT6$, but appropriate override values are supplied in the job.
EYUWUISP	CICS SIT overrides for a WUI.

For more information on WUI customization, see WUI data set creation and customization.

Table 19. Post-installation members for a managed CICS system (MAS)

Member	Description
EYUCSYDS	JCL to create and initialize the data sets for a managed CICS system.
EYUCSYSJ	JCL to start a managed CICS system. It executes EYUCSYSP.
EYUCSYSP	Procedure to start a managed CICS system. The MAS uses the CICS-supplied sample system initialization table, DFHSIT6$, but appropriate override values are supplied in the job.
EYUJHIST	JCL to delete and define a pair of history data sets.
EYULMS0P	EYUPARM parameters for a managed CICS system.
EYULMSSP	CICS SIT overrides for a managed CICS system.
EYULPMOD	JCL to apply the USERMOD function, EYU$UM01, that moves some MAS load modules to a link pack area (LPA) library.

For more information on managed CICS system customization, see MAS data set creation and customization.

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.

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. For information about sharing CSDs across CICS releases, see CSD compatibility between different CICS releases.

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.

WUI plex definition

You can use the EYU9XDUT utility to create the definitions required for a WUI and its plex.

The EYU9XDUT CICSplex definition utility can provide the CICSPlex SM definitions to start a WUI and CICSplex as part of data repository initialization.

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).

Changes to CICSPlex SM externals

Changes to Samples

Obsolete samples

The CICSPlex SM Starter Set is no longer provided. Instead, you can use the DFHISTAR job to produce JCL to create the data sets required for a CMAS, start the CMAS, create the data sets required for a WUI, start the WUI, create the data sets required for a MAS, start the MAS.

Because the CICS definitions for CICSPlex SM objects are now created dynamically during initialization, a number of samples which formed input to CSD UPGRADE tasks are now obsolete. The obsolete samples are as follows:

- EYU964G1 MAS and WUI definitions (CPSM 3.2 agent running on CICS TS 3.1)
- EYU963G1 MAS and WUI definitions (CPSM 3.2 agent running on CICS TS 2.3)
- EYU962G1 MAS and WUI definitions (CPSM 3.2 agent running on CICS TS 2.2)
- EYU953G1 MAS and WUI definitions (CPSM 3.2 agent running on CICS TS 1.3)

New Samples

New samples are provided as follows:

EYU$CDEF
Contains the default CICS resource definitions for a CMAS

EYU$MDEF
Contains the default CICS resource definitions for a MAS

EYU$WDEF
Contains the default CICS resource definitions for a WUI

Messages

New and changed messages

There are a number of new and changed messages to support the integrated installation of CICSPlex SM:

EYUNX0011 to EYUNX0014
EYUNX0038E
EYUVS0960 to EYUVS0962
EYUVS0963E
EYUXD0601 to EYUXD0622
EYUXD0628E
EYUXD0700E
EYUXL0078W
EYUXL0154E to EYUXL0156E

Chapter 32. EYU9XDBT CICSPlex SM definition utility

EYU9XDBT is a new CICSPlex SM utility that provides an easy-to-use command interface for performing CMAS and CICSplex definition activities.

The EYU9XDBT CICSPlex SM definition utility uses the CICSPlex SM API to enable you to specify the required CICSplex names in some simple parameters, and the utility sets up the definitions for you. Unlike the BATCHREP utility, you do not need to manually edit an input file.

You can use this utility to perform all CMAS and CICSplex definition activities once the basic CMAS environment has been established. Such activities include:

- 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
- Creating CMAS to CMAS link definitions.
- Importing, printing or exporting CICSPlex SM objects defined to CMAS or CICSplex contexts.

It is limited to data repositories at the same release level as CICSPlex SM. EYU9XDBT is used during installation to set up your initial CICSPlex SM environment.

The following samples are provided:

EYUJXBT0
Contains annotated EYU9XDBT JCL syntax for use as a quick reference.

EYUJXBT1
Contains sample JCL for invoking EYU9XDBT and defining a CICSplex, a CICS system group and a CICS system definition.

EYUJXBT2
Contains sample JCL for invoking EYU9XBTP and creating a CMAS to CMAS link definition.

Changes to CICSPlex SM externals

Changes to problem determination

Messages

Messages in the range EYUXU1401 to EYUXU1456 are new.

Chapter 33. Expanding on a summary view record count

The CICSPlex SM Web User Interface (WUI) has been improved to enable you to expand a summary view to display the details of summarized records. You can now click on a record count field to open a new tabular view displaying those records that relate to the selected summary row showing the state of the system at the time the initial summary occurred.

The expanded tabular view shows the ordinary filters that have been defined for the view, plus the filter that has been used to expand the summary view. For example, in the case of a **Local or dynamic transactions** (LOCTRAN) view summarized on the **Number of times transaction used** column, the use count filter and value would appear on the expanded view in addition to any previously applied filters.

The expanded view is a normal, filtered, tabular view. You can perform any further actions on it that you would normally be allowed to on a tabular view including additional summarizations. When you click the back button on the expanded view, you are returned to the summarized view from which the expanded view was launched.

This function is supported by the new API command EXPAND. This takes the summarized result set created using the GROUP command and creates a new result set containing one record for each of the records summarized by GROUP in an individual summary record. This allows you to perform further actions on the result set including using additional GROUP or FETCH commands.

Changes to CICS externals

Changes to the CICSPlex SM programming interface

New command: EXPAND

This command supports the expansion of summary result sets. The command accepts a token from a summarized result set produced by the GROUP command, and a selected record identified by the position of the record pointer in the result set to be expanded. The position of the record pointer depends on the options that you specify on the command. It creates a new result set that contains all the records that are summarized in a summary record.

For details of the command, see “EXPAND” on page 337.

Changes to CICSPlex SM views

Changes to summary views

Summary views have been improved to make them easier to interpret and to facilitate the new expand feature. There are a number of changes:

- The contents of the **Record count** column has changed from displaying ordinary numbers to hyperlinks. Clicking on one opens a new expanded view displaying one row for each of the summarized records.
- The filters from the original unsummarized tabular view are displayed on the summary view as display-only filters. These filters are similar to collapsed filters on a normal view, however on the summary view there is no control to allow filter expansion. Furthermore any filters that are not in use are not displayed.

- Just below the filters is a new line of text that states **Summarized on** `columnname`, where `columnname` is the column used to summarize the original view.

The expanded view

The expanded view is a specialized form of the tabular view displaying one row for each of the records summarized on the selected row of the **Record count** column of the summary view. The view displays the state of the system as it was when the original summary took place. If you want to see a real time view, you can click **Refresh** or set the automatic refresh feature by clicking the adjacent radio button and setting the refresh interval (the default is sixty seconds).

Two messages are displayed on all expanded views:

- EYUVC1280I, which shows the original number of records and when they were collected, and
- EYUVC1380I, which shows the number of records in the expansion and when they were expanded. This is a new message.

The filters from the original unsummarized are displayed in the expanded view as display-only filters in the same way as they appear on the summary view.

Below the filters is a line of text stating:

Summarized on *columname*, Expanded on *value*.

where *columname* is the name of the column used to summarize the original view, and *value* is the content of the row in that column on which the expand operation took place.

Chapter 34. Improved help for the CICSPlex SM Web User Interface

The CICSPlex SM Web User Interface (WUI) has been improved with the introduction of detailed help for all IBM-supplied and user-defined views and menus.

Using the improved WUI help

Each WUI view and menu now has its own specific page of help information.

To display a help page, click on the **Help for this display** icon (?) located in the work frame, opposite the view or menu title. The help is displayed in a separate browser window.

For a view, the help page contains a general description of the view itself and specific information on the fields, filters and actions that the view contains in tabular form.

For a menu, the help page contains general information on the menu itself and on each menu filter, group and item.

By default help pages present each item of information in the same order that they are displayed on the view or menu. However if there are more than nine items in a table, it is possible to filter the information so that only those rows that contain a specified string in a specified column are displayed. This function is available only when using Microsoft Internet Explorer, or Mozilla-based browsers, such as Firefox, at version 5.0 or higher. JavaScript™ must be enabled.

Help is also available for views and menus you create yourself with the view editor. Each resource table is supplied with a general description for its derived views as well as help text for each attribute and action. You can replace the supplied general descriptions with text you have rewritten yourself, which is then displayed in preference to the supplied help. The supplied help text for individual fields, filters and other items on a help page is not editable. Supplied help is available for all resource table attributes, whether or not a particular attribute is present in a supplied view.

Changes to CICSPlex SM externals

There are major improvements to the help function on all WUI views and menus. There is also one new WUI client message; EYUVC1400E.

Changes to WUI views and menus

There are several changes to the way that help is implemented from WUI views and menus:

- The help function introduces a new help icon on all WUI views and menus linking to a page of specific help information containing descriptions of all the items on that view or menu.
- The way you open help for WUI messages has changed slightly. To open help for a message displayed on a WUI view or menu, click on the message number, which is now a hyperlink. This replaces the message help icon of previous releases.

- It is now possible to open the CICS Information Center directly from the WUI. If you specify the CICS system initialization parameter INFOCENTER in your WUI startup JCL, a new hyperlink labelled **Information Center** is displayed at the top of WUI views and menus, just to the left of the WUI general help icon. Clicking on this hyperlink opens a new browser window containing the CICS Information Center home page.

The method of opening the general WUI help has not changed. Click on the help icon at the top right of a WUI view or menu to open the WUI help contents page.

Chapter 35. Support for the map function in the CICSPlex SM Web User Interface

The CICSPlex SM Web User Interface has been improved by the addition of a map function equivalent to the CICSPlex SM end user interface MAP command in previous releases of CICS TS.

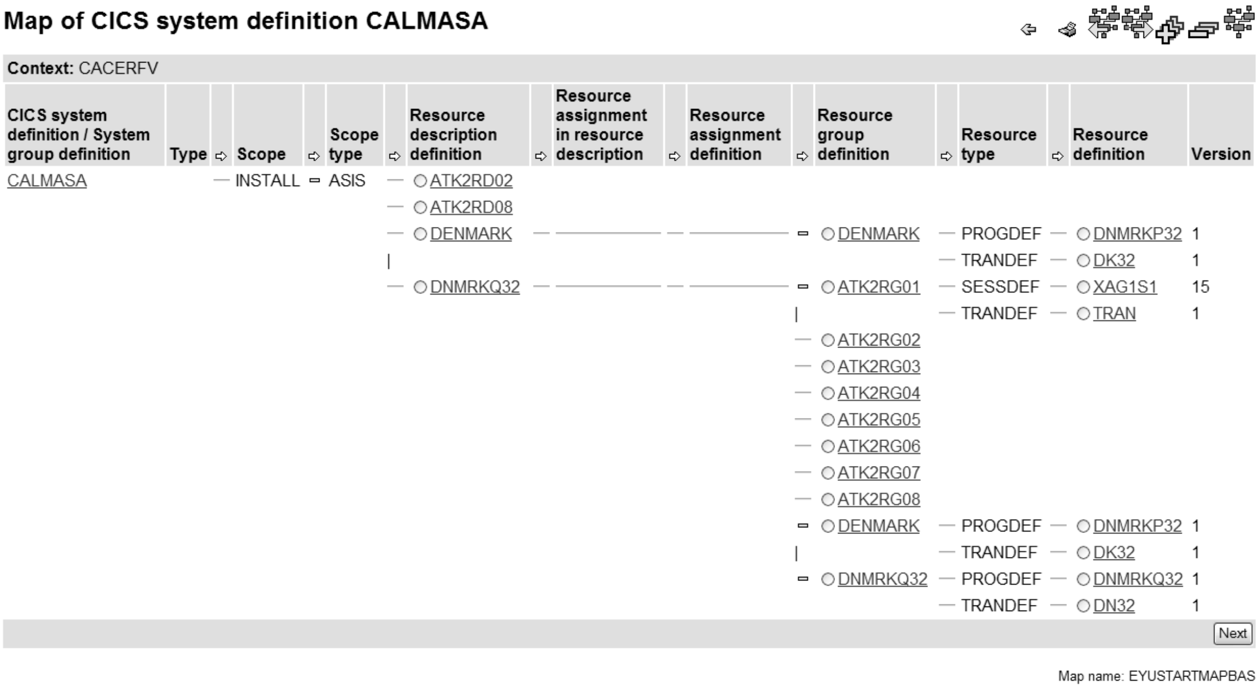
The associations between CICS resource definitions defined to CICSPlex SM can be complex and difficult to visualize. For example, a CICS system can be associated with a specification and a specification might contain one or more groups. In turn there can be definitions within the groups. This type of structure is often portrayed as the branches of the tree and the WUI map function provides a method of generating a visual representation of this tree structure for a selected resource. This representation, called a map, can portray business application services (BAS), resource monitoring (MON), real-time analysis (RTA), or workload management (WLM) definitions. Maps allow you to verify that the relationships between your definitions are what you expect.

Mapping CICSPlex SM definitions in the Web User Interface

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.

By default, the map displays definitions that are referred to by the selected definition, that is, it maps down level. This is called a *map right* operation. For example, if the selected resource is a WLM specification, the map displays associated WLM group definitions and transaction groups. However, if your selected resource is already at the lowest level, for example a WLM transaction group or a BAS resource definition, the map displays all up-level relationships. In the case of a transaction group the map displays associated definitions, groups and specifications. This is called *map left*.

A typical map screen is shown in Figure 7 on page 220.



Map of CICS system definition CALMASA

Context: CACERFV

CICS system definition / System group definition	Type	⇨	Scope	⇨	Scope type	⇨	Resource description definition	⇨	Resource assignment in resource description	⇨	Resource assignment definition	⇨	Resource group definition	⇨	Resource type	⇨	Resource definition	Version
CALMASA		—	INSTALL	▭	ASIS	—	○ ATK2RD02											
						—	○ ATK2RD08											
						—	○ DENMARK	—	———	—	———	▭	○ DENMARK	—	PROGDEF	—	○ DNMRKP32	1
						|								—	TRANDEF	—	○ DK32	1
						—	○ DNMRKQ32	—	———	—	———	▭	○ ATK2RG01	—	SESSDEF	—	○ XAG1S1	15
												|		—	TRANDEF	—	○ TRAN	1
												—	○ ATK2RG02					
												—	○ ATK2RG03					
												—	○ ATK2RG04					
												—	○ ATK2RG05					
												—	○ ATK2RG06					
												—	○ ATK2RG07					
												—	○ ATK2RG08					
												▭	○ DENMARK	—	PROGDEF	—	○ DNMRKP32	1
												|		—	TRANDEF	—	○ DK32	1
												▭	○ DNMRKQ32	—	PROGDEF	—	○ DNMRKQ32	1
														—	TRANDEF	—	○ DN32	1

Map name: EYUSTARTMAPBAS

Figure 7. Diagram of a typical map view

At the top of the map view are icons enabling you to do the following, from left to right:

- Return to the previous detailed view.
- Display a printer-friendly map view
- Switch to a map left view of the selected resource definition
- Switch to a map right view of the selected resource definition
- Expand all of the definitions on the map display
- Collapse all of the definitions on the map display

There is no **Add to favorites** icon on a map screen. You cannot bookmark a map screen as one of your favorites.

If you select more than one resource on a tabular view, you can click **Next** to display a map of the next selected resource.

By default the map is normally displayed in the expanded state. However you can alter this default behavior using the WUI server initialization parameter DEFAULTMAPCOLL, or for particular groups of users, when specifying a user group profile.

The map itself is a table with between 4 and 18 columns. There are two different types of column on the map view; resource columns and connecting columns.

Resource columns contain the resource names of objects. The header shows the type of definition the column relates to, for example WLM specification, WLM group, and so on. Resource names act as hyperlinks. Clicking on one opens a detailed view of that resource.

Instead of resource names, resource columns may contain the following symbols:

Horizontal line (______)
Indicates that there is a higher level resource name that is directly connected. There is no corresponding resource name.

Asterisks (****)**
Indicates that this row is in a collapsed state. This cell is at a lower level to the resource that performed the collapse.

Broken line (- - - - -)
This is displayed only in the RASINDSC column on a map of a BAS resource definition. It indicates that there is an association between the RESDESC and RESGROUP caused by the fields in a RASINDSC object. Without this symbol, it would imply that the association between the RESDESC and the RESGROUP was only a direct connection.

Connecting columns appear between resources on the map view. The header of a connecting column contains an icon showing the direction in which the map should be read, that is, an arrow pointing right for a map right (down level) operation, and an arrow pointing left for a map left (up level) operation. Connecting columns can contain one of the following symbols:

Horizontal bar
Shows that the values in the columns on either side of this bar are connected together. The bar may connect a resource to a blank space and vertical bar character, in which case it means that the resource is related to the first proper resource name encountered when moving up the table. For example, in Figure 7 on page 220 resource group definition ATK2RG02 is connected to a cell with a bar character. This means that ATK2RG02 is associated with the first proper resource encountered when moving up the table, which is resource description DNMRKQ32.

Expand or collapse icon
Shows a connection in the same way as a horizontal bar. However, these icons can also be used to expand and collapse parts of the map. If a resource has more than one lower level resource associated with it; for example, a resource group with more than one resource definition type, the connecting column to the left of this resource contains a collapse icon (assuming it is in an expanded state). This changes to an expand icon if you collapse the row.

Vertical bar (|)
Indicates that a resource has a sibling and has more than one lower level association. A bar is displayed in each cell in the column until the sibling resource name is rendered.

Some columns on a map, such as **Resource definition**, contain more than one resource type in the column. In such cases there is an additional column added to the right containing the resource type of these resources.

You can initiate a map right or map left operation for most resources displayed on a map screen by selecting the adjacent radio button and clicking the map right or map left icon. The exceptions to this are CICS system or CICS group names because performing a further map operation on these resources would not result in any additional useful data: a map right command would redisplay only the current data, while a map left command would produce a map with only the CICS system/CICS group name displayed.

There are also some columns that can appear on a map screen that are not proper resources but provide additional information about relationships (for example **Scope Type** on a BAS map). As these are not true resources, it is not possible to perform a map left or map right operation on them.

Tip: If you are using a screen reader, you need to ensure that it is properly configured in order to correctly interpret the information contained in the map. In particular:

- Set the screen reader's punctuation mode to voice or display all symbols. This is because the vertical bar symbol is used to denote relationships between elements on a map screen when applicable. If the screen reader is not set up to voice or display the vertical bar symbol, the screen reader cannot determine the relationships between map elements.
- Use the table mode option of the screen reader. This should ensure that the map is read in its intended logical sequence.

There are four default map objects supplied with the WUI. These are named as follows:

EYUSTARTMAPBAS for generating maps of business application services definitions

EYUSTARTMAPMON for generating maps of monitoring definitions

EYUSTARTMAPRTA for generating maps of real-time-analysis definitions

EYUSTARTMAPWLM for generating maps of workload management definitions.

Each map object includes links from the resource columns to detailed views of the named resource. In maps you create yourself in the view editor, you can customize links to point to different destinations including to your own customized views.

You can also use the map function on WUI screens you design yourself. The WUI view editor gives you the option of including a map button on tabular or detailed views for definitional objects and to create and edit customized maps.

Changes to CICSPlex SM externals

Changes to CICSPlex SM Web User Interface server initialization parameters

The map function introduces five new optional Web User Interface server initialization parameters: DEFAULTMAPBAS, DEFAULTMAPCOLL, DEFAULTMAPMON, DEFAULTMAPRTA, and DEFAULTMAPWLM.

New Web User Interface server initialization parameters

The following new parameters specify default map objects and the appearance of newly opened maps.

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.

Changes to problem determination

Messages

There are new Web User Interface view editor messages in the range EYUVE1003 to EYUVE1031.

There are new Web User Interface client messages: EYUVC1228E and EYUVC1229E.

There is a new Web User Interface server message: EYUVS1030E.

Chapter 36. Extended CICSPlex SM support for TDQs and CMASs

The CICSPlex SM Web User Interface (WUI) has been improved by the addition of new WUI views providing more information about transient data queues, and help with the management of CMASs. Additionally, API support has been extended to the CPLXCMAS resource table.

There are two new WUI view sets:

Topology data for transient data queues

This is a single view providing tabular information about all intrapartition, extrapartition, and indirect transient data queue (TDQ) resources within the specified context and scope. It identifies the name and types of transient data queues and contains links to the appropriate type-specific TDQ view. It is associated with the CRESTDQ resource.

- To open this view from the WUI main menu, click **CICS operations views** → **Transaction data queue operations views** → **Topology data for transient data queues** .

CMAS in CICSplex definitions

This view set lists CICSplexes and the CMASs associated with them. By setting your context to a specific CMAS you can see all the CMASs that manage the CICSplexes for which the context CMAS is the maintenance point. It is associated with the CPLXCMAS resource. You can use the **Unassign** action to remove CMASs from the management of the CICSplex.

- To open this view from the WUI main menu, click **Administration views** → **CMAS configuration administration views** → **CMAS in CICSplex definitions**.
- Click on a record in the **CMAS** column to open the associated detailed view of the selected CMAS.

The addition of CICSPlex SM API support for the CPLXCMAS resource table enables you to write applications that use the API command UNASSIGN to remove a CMAS from a CICSplex management role.

Chapter 37. 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 this release. 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.

The following EYUPARMS have been removed:

- xxxCONMSG
- xxxTDQMSG

The following messages have been added to support the NLS-enablement of CICSPlex SM messages:

- EYUBM0329I to EYUBM0348I
- EYUBN0013W to EYUBN0017W
- EYUXL0030I to EYUXL0032I

The following messages have been removed:

- EYUBM0322I to EYUBM0327I
- EYUBN0012W
- EYUXL0020I

Changes to CICS externals

Changes to global user exits

Changes to the message global user exit XMEOUT

The message global user exit, XMEOUT, has been changed to enable national language support for CICSPlex SM messages. Four new fields have been added to XMEOUT:

UEPCPID
Address of 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.

For more information see the *CICS Customization Guide*.

Note: CICSPlex SM messages are not available through the CMAC transaction.

Chapter 38. Improved CICSPlex SM history function

The CICSPlex SM MAS history function has been improved so that it is now possible to retrieve additional performance class monitoring data for the resource managers used by your transactions, by specifying RMI=YES in your MCT, and application naming data by specifying APPLNAME=YES in your MCT. Also, it is now possible to use the CICSPlex SM Web User Interface supplied EYUSTARTHTASK tabular and detailed views to retrieve historical task data from the historical data store.

Changes to CICS externals

Changes to the CICSPlex SM programming interface

New history recording fields and base tables

The following new resource manager interface (RMI) and application naming fields have been added to the HTASK base table:

APPLNAMETRAN
Application naming transaction name

APPLNAMEPROG
Application naming program name

RMIOTHERTIME
Total other RMI elapsed time

RMIDB2TIME
DB2 RMI elapsed time

RMIDBCTLTIME
RMI DBCTL RMI elapsed time

RMIEXECDLITM
DL/I RMI elapsed time

RMIMQSERIEST
WebSphere MQ RMI elapsed time

RMICPSMTIME
CICSPlex SM API RMI elapsed time

RMITCPIPTIME
Communications server CICS socket RMI elapsed time

Two new base tables have been added to further improve the MAS history function and to provide additional task RMI information, as follows:

MASHIST
MAS history control

TASKRMI
Task RMI

For more information see the *CICSPlex System Manager Resource Tables Reference* manual.

To gather performance class monitoring data for the resource managers used by your transactions and to turn on application naming support provided by CICS monitoring RMI=YES and APPLNAME=YES, respectively, must be specified in the MCT on the DFHMCT TYPE=INITIAL macro. For more information see the *CICS Resource Definition Guide*.

Changes to CICSPlex SM views and menus

New historical data viewsets

The following new viewsets have been added to the CICSPlex SM Web User Interface starter set so that historical information regarding tasks may be retrieved using the Web User Interface.

EYUSTARTHTASK
Viewset for history task

EYUSTARTMASHIST
Viewset for MAS history control

EYUSTARTTASKRMI
Viewset for task RMI

Chapter 39. Other changes to CICSPlex SM

A number of changes have been made to the CICSPlex SM Web User Interface (WUI) to make it more functional and enhance its usability and serviceability. Several new resource tables and view sets have also been added.

Sorting and summarizing on CICS system name

New icons have been added to appropriate CICSPlex SM Web User Interface tabular views to enable sorting and summarizing of the CICS system name column.

Terminology improvements

The terminology used in WUI views and menus has been simplified in order to improve consistency and reduce the length of some titles and phrases. The use of shorter titles has led to a reduction in the width of some columns enabling views to display more data. No new terms have been introduced.

New resource tables and WUI view sets

A number of new task-related resource tables and associated WUI view sets have been added to CICSPlex SM.

To access these views from the WUI main menu, click **CICS operations views** → **Task operations views**

Table 20. New resource tables and view sets

Resource table	WUI view set	Description
TASKESTG	Task element storage EYUSTARTTASKESTG	Information about CICS storage elements for tasks.
TASKFILE	File usage by task EYUSTARTTASKFILE	Information about tasks and the CICS files they have used.
TASKRMI	RMI usage by an individual task EYUSTARTTASKRMI	Information about the use tasks have made of the CICS Resource Manager Interface (RMI).
TASKTSQ	TS queue usage by task EYUSTARTTASKTSQ	Information about tasks and their associated CICS temporary storage queues.

Using the WUI to control CMAS and MAS tracing

You can use the Web User Interface (WUI) to control the tracing that occurs in an active CMAS and MAS. Two new views are provided: the **MASs known to CICSplex** trace view and the **CMAS detail** trace view. The trace flags are displayed as strings of bits in the range 1 through 32, separated by commas. You change the trace flag settings by editing the display.

Changes to the WUI data repository import function

The WUI server repository import function has been improved to make it easier to update WUI views and menus as a result of program temporary fixes (PTFs) from IBM service teams. It is now possible to import menus and view sets singly or in groups. In previous releases the whole set of supplied views and menus needed to be regenerated in order to implement any change. The import function now allows you to update the supplied views and menus without having to shut down the WUI server.

The new function, uses the import panel of the CICS COVC transaction. As well as allowing you to import a transient data queue containing a complete set of views and menus, you can now import data set members containing view sets or menus. To facilitate this there are two new fields on the panel; **Input Data set Name**, and the **Input Data set member name**. You can choose either to specify a TDQ name to import a complete set of supplied or customized view and menu definitions as in previous releases, or a data set and member name to import specific views or menus. You can use the asterisk as a trailing wildcard character on the data set member name field to specify a group of views or menus. You cannot import both a TDQ and a data set at the same time using COVC.

The supplied set of WUI view and menu definitions is currently located in the SEYUVIEW data set. The composition of this data set has changed to facilitate the new function. Formerly SEYUVIEW contained three members, one each for the English, Japanese and simplified Chinese versions of the definitions. Now the data set includes one member for each view set and menu in each of the three supplied languages. These data set members are named **EYU**`ltccc`, where:

- `l` specifies the language; **E** for English, **S** for simplified Chinese and **K** for Japanese.
- `t` identifies a set of views. The current supplied WUI views and menus are all identified by the letter **A**.
- `nnn` identifies the resource with which the views are associated.

As an alternative to the COVC transaction, you can configure a WUI server to import automatically menus and view sets from a specified data set or data set and member at startup. To facilitate this there are two new optional WUI server initialization parameters:

AUTOIMPORTDSN()
Identifies the name of a data set to import.

AUTOIMPORTMEM()
Identifies the name of a data set member to import. You can use an asterisk as a trailing wildcard character to specify a group of views or menus.

You can also use the AUTOIMPORTTDQ parameter to automatically import a specified TDQ when you start a WUI server.

The following WUI server messages are introduced in support of the new import function:

EYUVS0929E
EYUVS0930W
EYUVS0931E
EYUVS0113W
EYUVS0114E

EYUVS1050E
EYUVS1051E
EYUVS1052E
EYUVS1053E
EYUVS1054E
EYUVS1055E

Change to the WUI data repository export function

The COVC export function is used to export WUI definitions so that you can back up or distribute definitions to other WUI servers, or migrate definitions to other releases. WUI data repository definitions consist of view sets, menus, map objects, user objects and user group profiles. In previous releases you could specify only one resource type in an export operation. It is now possible to export all data repository definition types at the same time. This makes it easier to export your definitions to a single TDQ.

To facilitate this the COVC Export panel has been changed to allow a value of `All` in the **Type** field. You can use the All in conjunction with the **Name** field that identifies specific or generic name of the objects to be exported. This field utilizes an asterisk as a trailing wildcard character. Specifying `All` with an asterisk in the **Name** field results in all the definitions being exported from the repository. If you use `All` with, for example `TEST*` , COVC will export all of definitions that have a name starting with TEST, whatever their type.

Change to the codepage conversion table (DFHCNV)

The default codepage conversion table (DFHCNV) has been changed so that CICSPlex SM codepages are included automatically. That is, it is no longer necessary to include a copy statement for EYU$CNV1 in the DFHCNV source.

Part 5. Discontinued function

Some functions which were supported in CICS Transaction Server for z/OS, Version 2 have been discontinued, or reduced in scope in CICS Transaction Server for z/OS, Version 3 Release 2.

Chapter 40. Removal of the CICSPlex SM TSO end user interface

With the new enhancements to the CICSPlex SM Web User Interface (WUI) functionality and provision of the EYU9XDBT definition utility, 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.

It has not been possible to use the EUI to manage the more modern features of CICS since the EUI was stabilized at the CICS Transaction Server for z/OS, Version 2 Release 2 level of functionality. Its removal in this release:

- Improves and streamlines the installation of CICSPlex SM.
- Makes migration scenarios more straightforward.
- Reduces the complexity of system configuration by reducing the number of address spaces that have to be managed.

The EUI is replaced by the WUI, which is:

- Customizable to your business needs.
- Easier to learn and use.
- Accessible to authorized users from any location that can launch a web browser.
- Fully accessible to those with restricted vision or mobility.
- National language support (NLS) enabled.

Changes to CICS externals

The removal of the CICSPlex SM EUI results in a number of changes to CICS externals.

Changes to installation

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.

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.

Changes to system initialization parameters

The CICSPlex SM system parameter CASNAME is discontinued.

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 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.

Changes to CICSPlex SM views and menus

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.

Changes to problem determination

With the removal of the EUI, any attempt to run EYUCAS JCL to start a CAS results in an abend. A large number of CICSPlex SM messages are no longer valid and cannot be issued.

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.

Chapter 41. Removal of resettable mode for JVMs (Java virtual machines)

Continuous JVMs, which are not reset between each use, generally perform better than resettable JVMs and are more consistent with other versions of Java. Resettable JVMs are no longer supported, and migration to continuous JVMs is required in this CICS release. The CICS JVM Application Isolation Utility, a code checking and reporting utility, is provided to help identify areas where you should check the behavior of Java programs that were designed to run in resettable JVMs, before migrating them to run in continuous JVMs. Configuration and tuning for continuous JVMs is simpler than it was for resettable JVMs, and some unnecessary JVM profile options are now deprecated.

Continuous JVMs have several advantages over resettable JVMs:

- They have a lower CPU cost per transaction, because they do not require a reset between each use.
- They are simpler to set up and tune, primarily because they have fewer different storage heaps than resettable JVMs.
- Certain constraints placed on programs in resettable JVMs do not apply for continuous JVMs, enabling developers to maximize the performance of their applications.
- They are compatible with future versions of Java, and are more like the standard JVMs used by other products.

Withdrawal of resettable JVMs

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`).

Resettable JVMs were reset after each Java program had completed. The JVM reset prevented applications from performing unresettable actions such as modifying the state of a JVM or leaving cross-heap references in scope. If unresettable events were detected during the execution of a user's Java program, the JVM was marked unresettable, and CICS destroyed the JVM when the Java program had finished using it. The JVM reset also cleaned up the JVM's storage heaps after each use, meaning that state could not persist from one program invocation to the next.

Although this process enforced serial isolation for programs running in the JVM, the time and CPU usage required for a JVM reset reduced the performance of a resettable JVM compared to the performance of a continuous JVM. Resettable JVMs were also incompatible with future versions of Java, whereas continuous JVMs are compatible with future versions of Java.

An application that has been coded with attention to the state of the JVM and to the items in static storage can operate safely in a continuous JVM without the JVM reset. If you need to police the use of any APIs in the continuous JVM, the Java security manager can be used to do this.

The migration process for Java programs that ran in a resettable JVM involves checking that the 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 CICS JVM Application Isolation Utility, a code checking and reporting utility, is provided with CICS Transaction Server for z/OS, Version 3 Release 2 to help identify areas where you should check the behavior of Java programs that were designed to run in resettable JVMs.

Configuration and tuning for continuous JVMs is simpler than it was for resettable JVMs. Your choice of class path is more straightforward, and there are fewer storage settings to tune. When you migrate an application to run in a continuous JVM, you will probably need to merge some of your existing storage settings. Your existing class path options are accepted for migration purposes, and CICS issues a warning message about those options which are obsolete.

Removal of middleware classes and the trusted middleware class path

Because resettable JVMs are no longer supported in CICS Transaction Server for z/OS, Version 3 Release 2, and continuous JVMs are used to run Java applications, there is no longer any need to distinguish between middleware classes and user application classes.

In a resettable JVM, middleware classes were classes trusted by the JVM to manage their own state between one use of a JVM and the next, resetting themselves correctly and reinitializing if necessary, and also trusted to make changes to the JVM environment. User application classes, on the other hand, were not trusted to perform these actions. The JVM reset process handled these actions on behalf of user application classes.

The classes treated as middleware classes were normally those classes supplied by IBM or by another vendor to provide services that access resources, such as the JCICS interface classes or the DB2-supplied JDBC drivers. Although classes like these provide services which can be used by multiple user applications, they are not included in the standard JVM setup for CICS, so they must be placed on an appropriate class path in the JVM profile.

In resettable JVMs, these classes were placed on the trusted middleware class path, so that resettable JVMs could identify them as middleware and allow them freedom of action. The trusted middleware class path was built automatically from the paths specified by the CICS_DIRECTORY (now changed to CICS_HOME), TMPREFIX, and TMSUFFIX options in the JVM profile. User application classes were placed on different class paths so that resettable JVMs could police their activities.

In a continuous JVM, all classes have the same freedom of action, and are all responsible for managing their own state and policing any changes to the JVM environment to maintain the correct level of isolation between successive programs running in the JVM. There are no special restrictions on user application classes. This means that the classes formerly treated as middleware classes must now be placed on the same class path as user application classes. The classes formerly treated as middleware classes still continue to manage their own state and the JVM environment correctly, just as they did when they were used in a resettable JVM. The difference is that the same level of care is now required from user application classes as well.

In CICS Transaction Server for z/OS, Version 3 Release 2, both the classes formerly treated as middleware classes, and user application classes, are all referred to simply as **application classes**.

Removal of the application-class system heap, middleware heap, and transient heap

Resettable JVMs, which had special storage heaps known as the application-class system heap, middleware heap and transient heap, are no longer supported in CICS Transaction Server for z/OS, Version 3 Release 2. Continuous JVMs, which are now used to run Java applications, do not have these storage heaps.

The storage heaps which were in resettable JVMs, but are not in continuous JVMs, are:

Middleware heap

- In a resettable JVM, the middleware heap was a special subset of the nonsystem heap. It was mainly used for objects and static data relating to middleware classes (on the trusted middleware class path).The objects in this storage heap were kept across JVM resets. The middleware heap's initial storage allocation was set by the `Xms` option in a JVM profile. Storage for the middleware heap was taken from the nonsystem heap, that is, from the storage delimited by the `Xmx` option.
- In a continuous JVM, the nonsystem heap is used for items that would be contained in the middleware heap for a resettable JVM. There is no longer any need to distinguish between middleware classes and user application classes, so there is no need to identify this subset of the nonsystem heap. The nonsystem heap's initial storage allocation is set by the `Xms` option in a JVM profile, the same option that was used to specify the middleware heap's initial storage allocation in a resettable JVM.

Transient heap

- In a resettable JVM, the transient heap was another special subset of the nonsystem heap. It was used for objects and static data relating to user-written application classes. The objects in this storage heap had a lifetime that was the same as the program using the JVM, and the transient heap was completely deleted when the JVM reset took place. The transient heap's initial storage allocation was set by the `Xinitth` option in a JVM profile. Storage for the transient heap was taken from the nonsystem heap, that is, from the storage delimited by the `Xmx` option.
- In a continuous JVM, the nonsystem heap is used for items that would be contained in the transient heap for a resettable JVM. This means that the items are kept intact from one JVM reuse to the next. The nonsystem heap's initial storage allocation is set by the `Xms` option in a JVM profile, and the `Xinitth` option is no longer used.

Application-class system heap

- In a resettable JVM, the application-class system heap, or ACS heap, was a separate heap within the Language Environment enclave for the JVM. It was not part of the system heap or the nonsystem heap. It was used for class definitions and class objects relating to user-written application classes on the shareable application class path. The objects in this storage heap persisted for the lifetime of the JVM (that is, they were kept across JVM reuses) and were reinitialized if the JVM was reset. The application-class system heap's initial storage allocation was set by the `Xinitacsh` option in a JVM profile.

- In a continuous JVM, the system heap is used for items that would be contained in the application-class system heap for a resettable JVM. The system heap's initial storage allocation is set by the `Xinitsh` option in a JVM profile.

Changes to CICS externals

Changes to options in JVM profiles and JVM properties files

There are a number of changed options in JVM profiles and JVM properties files as a result of the enhancements to Java 1.4.2 support.

Deprecated or unusable options due to withdrawal of resettable mode

These options in JVM profiles and JVM properties files should no longer be used, because they applied to resettable JVMs. In some cases CICS issues a warning message and uses the value of the option in the correct way. In some cases the option is ignored, and in some cases it prevents the JVM from being started.

The following options are no longer accepted in JVM profiles:

REUSE=RESET
If a JVM profile includes REUSE=RESET, which specifies a resettable JVM, the JVM does not start, and CICS issues the error message DFHSJ0524. The settings REUSE=YES (a continuous JVM) and REUSE=NO (a single-use JVM) are accepted.

Xresettable={YES|*NO*}
This is an older option which was used before the introduction of continuous JVMs to specify whether a JVM was resettable or single-use. If a JVM profile includes `Xresettable=YES`, which specifies a resettable JVM, the JVM does not start, and CICS issues the error message DFHSJ0524. If a JVM profile includes `Xresettable=NO`, which specifies a single-use JVM, the option is ignored and CICS issues the error message DFHSJ0525. The JVM starts, but it will be a continuous JVM, which is the default.

The following options are **ignored** by the Java launcher and have no effect:

ibm.jvm.crossheap.events
Enabled the logging of cross-heap references in a resettable JVM.

ibm.jvm.events.output
Enabled event logging in a resettable JVM.

ibm.jvm.reset.events
Suppressed JVM reset messages in a resettable JVM.

ibm.jvm.resettrace.events
Enabled the logging of reset trace events in a resettable JVM.

ibm.jvm.unresettable.events.level
Enabled the logging of unresettable events in a resettable JVM.

Xinitacsh
Specified the initial size of the application-class system heap for resettable JVMs.

Xinitth
Specified the initial size of the transient heap for resettable JVMs.

If any of the options relating to events are found, CICS issues the warning message DFHSJ0526.

The following options are deprecated, but CICS accepts them and handles their values in an appropriate way:

TMPREFIX
Specified paths to be inserted at the beginning of the trusted middleware class path. If this option is found, CICS issues the warning message DFHSJ0521, and inserts the paths at the beginning of the shareable application class path.

Note: The warning message for this option says that it should only be used under the guidance of IBM support. If you have used this option to specify paths, you should migrate its value to the `-Dibm.jvm.shareable.application.class.path` system property in the JVM properties file as soon as possible.

TMSUFFIX
Specified paths to be inserted at the end of the trusted middleware class path. If this option is found, CICS issues the warning message DFHSJ0522, and inserts the paths on the shareable application class path, after the directory containing the CICS-supplied classes but before the directories specified on the `-Dibm.jvm.shareable.application.class.path` system property in the JVM properties file.

Changes to the system programming interface

INQUIRE CLASSCACHE command

The REUSEST option on the INQUIRE CLASSCACHE command no longer returns a value of RESET. 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 and INQUIRE JVMPROFILE commands

The REUSEST option on the INQUIRE JVM and INQUIRE JVMPROFILE commands no longer returns a value of RESET.

Changes to CEMT

INQUIRE CLASSCACHE command

The value Reset is no longer displayed for the Reusest field. 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 command

The value Reset is no longer displayed for the Reusest field.

Changes to the CICSPlex SM programming interface

Changes to resource tables

In the JVMPOOL resource table, the field SJGREQSRESET (Number of JVM requests with JVM reset) returns ″Not Applicable″ for CICS TS 3.2 regions.

In the JVMPROF resource table, the fields CJVMSUNRESET (Number of CICS key JVMs not resettable) and UJVMSUNRESET (Number of USER key JVMs not resettable) return ″Not Applicable″ for CICS TS 3.2 regions.

In the JVM, JVMPROF and CLCACHE resource tables, the value ″Reset″ for the REUSEST (JVM reuse status) field is now obsolete.

Changes to CICSPlex SM views and menus

In the ″Java virtual machine (JVM) pool″ view, the field ″Number of JVM requests with JVM reset″ is now displayed as ″Not applicable″ for CICS TS 3.2 regions.

In the ″Java virtual machine (JVM) profile″ view, the fields ″Number of CICS key JVMs not resettable″ and ″Number of USER key JVMs not resettable″ are displayed as ″Not applicable″ for CICS TS 3.2 regions.

In the ″Java virtual machine (JVM) status″, ″Java virtual machine (JVM) profile″ and ″JVM Class Cache status″ views, the value ″Reset″ for the JVM reuse status field is now obsolete.

To access any of these views from the main menu, select **CICS operations views** → **Enterprise Java component operations views**.

Changes to CICS utilities

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.

Changes to monitoring

Performance data group DFHTASK

Field 164, ’TRANFLAG’

The flag for an unresettable JVM (which was byte 6, bit 0) is no longer set.

Field 275, ’JVRMTIME’

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.

Changes to statistics

CICS no longer collects statistics for JVMs that cannot be reset, and for requests to run a program in a resettable JVM.

In the DFHSJRDS DSECT (JVM Profile Statistics), the field SJR_JVMS_UNRESETTABLE is replaced by an unnamed reserved field.

In the DFHSJGDS DSECT (JVM Pool Global Statistics), the field SJG_JVM_REQS_RESET is replaced by an unnamed reserved field.

The statistics were shown in the DFHSTUP reports ″JVM profile statistics″ and ″JVM Pool statistics″, and in the DFH0STAT reports ″JVM Profiles Report″ and ″JVM Pool and Class Cache Report″.

Also in the DFH0STAT report ″JVM Pool and Class Cache Report″, the class cache reuse status is no longer displayed, because the master and worker JVMs must now always be continuous JVMs.

Changes to sample JVM profiles and JVM properties files

The sample JVM profiles and JVM properties files are updated to take into account the changes to options.

In particular, note that all the 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 sample JVM profiles and JVM properties files are described in *Java Applications in CICS*.

Changes to problem determination

Messages

New messages are issued if CICS is attempting to start a JVM, and finds an unusable, ignored or deprecated option in a JVM profile or properties file, or encounters a problem with accessing one of the required directories. Messages relating to deprecated options are sent to CDEP, and other messages are sent to CSMT. Messages DFHSJ0521, DFHSJ0522, DFHSJ0524, DFHSJ0525, DFHSJ0526, and DFHSJ0527 relate to JVM options that were withdrawn with the resettable mode.

Trace

Trace points SJ 0222, SJ 0519, and SJ 051A, which related to the resettable mode, are deleted.

Trace points SJ 0509, SJ 050C, and SJ 050F no longer provide data items relating to the trusted middleware class path.

Abends

Abend ASJR is issued if an attempt is made to start a resettable mode JVM by specifying REUSE=RESET or Xresettable=YES in the JVM profile. The abend is preceded by message DFHSJ0524.

Chapter 42. Removal of DFH$MOLS support for data for earlier CICS releases

The CICS Transaction Server for z/OS, Version 3 Release 2 release of DFH$MOLS does not process monitoring data for releases earlier than CICS Transaction Server for OS/390, Version 1 Release 3. 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 43. Removal of the DFHLSCU utility

The log stream sizing utility DFHLSCU has been removed from CICS.

Note: The utility is still available as SupportPac CD14 to assist in the migration of CICS MVS/ESA regions to CICS Transaction Server. The CICS SupportPacs are available from the following IBM Web site:

http://www-1.ibm.com/support/docview.wss?rs=1083&uid=swg27007241

Part 6. General Information

Chapter 44. The CICS operating environment

This topic gives some information about related products that you need in order to use the CICS and CICSPlex SM elements of CICS Transaction Server for z/OS.

Hardware requirements

Processors

The basic requirement is for a z/Architecture® processor that supports the prerequisite operating system and has sufficient processor storage to meet the requirements of z/OS, Version 1 Release 7; CICS TS for z/OS, Version 3.2; the application programs; the access methods; and all other software being run.

Parallel Sysplex support

A Parallel Sysplex® environment is required by each of the data-sharing facilities supported by CICS, and by the MVS system logger's log stream merging facility. This requires:

- One or more coupling facilities with their associated coupling links installed
- An IBM sysplex timer to provide a common external time source
- Sufficient DASD paths to support the number of central processor complexes (CPCs) in the sysplex. The DASD paths can be provided either by DASD controllers with enough paths to dedicate one to each CPC in the sysplex, or by an ESCON director.

CICS support for data sharing can be used to access data in IMS databases, DB2 databases, VSAM data sets, CICS temporary storage, coupling facility data tables, and named counters.

Cryptographic hardware

zSeries® cryptographic hardware is required:

- To exploit the WS-Security capability.
- To fully benefit from the performance improvements to SSL encryption.

Both functions rely upon the z/OS Integrated Cryptographic Services Facility (ICSF).

For the supported System z™ servers the cryptographic hardware is the CP Assist for Cryptographic 1 Functions (CPACF) and a Crypto Express2 (CEX2) feature.

Katakana Terminal Devices

Because CICS has to issue certain messages in mixed-case, the product is not supported with displays or terminal emulators that are restricted to the non-extended single-byte character set (SBCS) Katakana part of code page 930.

Software Requirements

Note that the *Program Directory* (GI10-6427) will normally contain the most up-to-date information on software requirements.

Operating environment

CICS TS for z/OS, Version 3.2 requires z/OS, Version 1 Release 7, or later. Note that it will not initialize in an environment with a lower level of operating system installed.

- If CICS TS for z/OS, Version 3.2 is used with z/OS V1.7, PTFs for APARs OA14340 and OA19565 are required.
- If CICS TS for z/OS, Version 3.2 is used with z/OS V1.8, PTF for APAR OA19565 is required.
- If CICS TS for z/OS, Version 3.2 is used with z/OS V1.9, PTF for APAR OA19565 is required.
- For EWLM support:
  - The EWLM Managed Server must be active in the MVS image where CICS is running.
  - With z/OS V1.7, z/OS PTF for APAR OA12935 is required. This is UA29986 (Release 720), UA29987 (Release 72J), or UA29988 (Release 72S).
- For TCP/IP support, Communications Server PTFs are needed.
  - For z/OS V1.7, the following PTFs for APAR PK32534 are needed: UK19627 and UK19628.
  - For z/OS V1.8, PTFs for APAR PK40411 are needed..
- To support the RACF-provided entity class and grouping class support for CICS document templates (RCICSRES and WCICSRES), the RACF PTFs for APAR AA20162 are needed. The PTF numbers are UA33762 (for z/OS V1R7) and UA33763 (for z/OS V1R8).
- The IBM XML Toolkit for z/OS (5655-J51) V1.9 is required. This is a no-charge product. It is used by WS-Security, but note that CICS TS for z/OS, Version 3.2 will not install if it is not present.

In order to forward recover an ESDS that has been updated by both CICS TS V3.2 and Transactional VSAM, applicable service is required on Transactional VSAM in z/OS V1.7, or later.

For developing Java programs (including enterprise beans), one of the following Integrated Development Environments (IDEs) is required:

- WebSphere Developer for zSeries V6.0 or WebSphere Developer for System z V7.0
- Rational® Application Developer V6.0 or Rational Application Developer for WebSphere Software V7.0 2
- WebSphere Studio Application Developer Integration Edition V5.1 or WebSphere Integration Developer V6.0

For deployment of enterprise beans, one of the above IDEs can be used, or the packaging application (Application Server Toolkit) provided with WebSphere Application Server V5.1, V6.0, or V6.1 can be used.

JNDI support for enterprise beans can be provided by the LDAP server provided in SecureWay® Security Server and licensed as part of the base z/OS operating system.

CICS TS V3.2 will interoperate with supported levels of WebSphere Application Server (any platform) V5.1, and later. This applies directly for customers using RMI/IIOP or SOAP, and applies via CICS Transaction Gateway (CICS TG) V5.1 or later for those using JCA. It includes use of the SOAP for CICS feature.

Other supported products

The following levels of other products are supported for use with CICS TS for z/OS, Version 3.2:

- IMS Database Manager V8 (5655-C56)
- IMS Database Manager V9 (5655-J38)
- IMS Database Manager V10 (5635-A01)
- DB2 Universal Database™ Server for OS/390 and z/OS V7.1 (5675-DB2).
- DB2 Universal Database Server for z/OS V8.1 (5625-DB2)
- DB2 V9.1 for z/OS (5635-DB2)
- WebSphere MQ for z/OS V5.3 (5655-F10)
- WebSphere MQ for z/OS V6.0 (5655-L82)
-
- Tivoli Business Systems Manager V3.3 (toleration support only)
- Tivoli Federated Identity Manager V6.1.1
- Tivoli Composite Application Manager for SOA V6.1 3
- Tivoli Composite Application Manager for WebSphere V6.1

Note: Tivoli Decision Support for z/OS (5698-A27) V1.6 and V1.7 do not support CICS TS V3.2.

- CICS Universal Client Version 5.1, or later
- CICS Transaction Gateway Version 5.1, or later

  Note: From V6.0 onwards, this is two products: CICS TG for Multiplatforms and CICS TG for z/OS.

CICSPlex SM Web User Interface

The CICSPlex SM Web User Interface can be used with all browsers that support HTML V4. IBM has validated the use of the CICSPlex SM Web User Interface with these browsers:

- Internet Explorer 6.0 and 7.0
- Firefox 2.0

Information Center environment

The Information Center can be installed on a workstation or a server on the following platforms:

- Windows Server 2003 (32-bit)
- Windows XP (32-bit)
- Windows Vista
- RedHat Enterprise Linux® 4.0 (Intel®) (32-bit)
- SUSE Linux Enterprise 8, 9, and 10 (Intel) (32-bit)
- AIX® V5.2 and V5.3 (32-bit)

The Information Center can be used as a server only on the following platforms:

- z/OS V1.7, or later
- Red Hat Enterprise Linux 4.0 for zSeries
- SUSE Linux Enterprise 8 and 9 for zSeries

For best results, view the Information Center using one of the following browsers:

- Internet Explorer 6.0, and later
- Mozilla-based browsers 1.7, and later
- Firefox 1.0, and later

To read PDF files shipped with the Information Center, you will need Adobe Acrobat Reader 5.0 or 6.0. The files have been generated using Adobe Acrobat Distiller 6.0 at the Acrobat 6.0 (PDF 1.5) level. They can be read using Adobe Acrobat Reader 5.0, but Reader 6.0 is necessary if you need the accessibility features of Distiller 6.0.

Support for CICS Tools and related products

The following can be used with CICS TS for z/OS, Version 3.2:

Application Performance Analyzer for z/OS V7.1
Asset Transformation Workbench V2.1
CICS Batch Application Control for z/OS V1.1.1 (with service applied)
CICS Configuration Manager for z/OS V1.2 (with service applied)
CICS Interdependency Analyzer for z/OS V2.1, with service applied
CICS Performance Analyzer for z/OS V2.1
CICS VSAM Recovery for z/OS V4.2
CICS VSAM Transparency for z/OS V1.1
CICS Online Transmission Time Optimizer for z/OS V1.2
Debug Tool Utilities and Advanced Functions for z/OS V7.1
Fault Analyzer for z/OS V7.1
File Manager for z/OS V7.1
IBM Session Manager for z/OS V1.3
Tivoli OMEGAMON® XE for CICS on z/OS V3.1
Tivoli OMEGAMON XE for CICS on z/OS V4.1
WebSphere Developer for System z V7.0
WebSphere Host Access Transformation Services
WebSphere Studio Asset Analyzer V5.1
Workload Simulator for z/OS V1.1

Compatibility

z/OS conversion services

Unlike previous levels of CICS Transaction Server, CICS TS V3 can use z/OS services to perform conversions beyond those supported by CICS TS in previous releases. An example is conversions to and from Unicode, which might be required to support Web services. This requires z/OS to have the initial conversion image installed, which can only be done on a system IPL. If it is wished to install CICS TS V3 without a re-IPL of z/OS, this can be done provided the initial conversion image is installed during a previous system IPL. The conversion image does not include any code from CICS TS; it can also be refreshed without any need for a further IPL.

JVM modes in CICS

Customers using Java programs in CICS TS V3.1 are recommended to use continuous mode. Support for continuous mode was introduced in CICS TS V2.3; in order to bring CICS use of Java into line with standard practices, support for resettable mode will be removed in a future release of CICS TS.

SOAP for CICS feature

The SOAP for CICS feature, orderable with CICS TS V2.2 and V2.3, is not orderable with CICS TS for z/OS, Version 3 releases. However, to assist migration for customers who already have this feature, the feature may be used and is supported with CICS TS for z/OS, Version 3, and applications will continue to run. However, customers are recommended to migrate to the Web services support capabilities of CICS TS for z/OS, Version 3 releases.

Common Connector Framework (CCF)

The Common Connector Framework (CCF), which was the predecessor interface to the Common Client Interface (CCI), is not supported by CICS TS V3.1. The intention to remove this support was indicated in the announcement of CICS TS V2.3.

ECI Base Classes (ECIREQUEST)

The ECI Base Classes (ECIREQUEST, which were introduced for compatibility with the CICS Transaction Gateway), are not included in CICS TS V3.1. The recommended replacement is the COMMON CLIENT INTERFACE CONNECTOR FOR CICS TS (CCI Connector for CICS TS), introduced in CICS TS V2.3, when it was announced that ECIREQUEST would be removed.

Transaction Affinities utility

CICS TS for z/OS, Version 3.2 does not include the detector and reporter components previously provided as part of the CICS Transaction Affinities utility. These components, as well as the load library scanner component, are now incorporated in the IBM CICS Interdependency Analyzer for z/OS, which has the capability of analyzing both interdependencies and affinities. The load library scanner alone remains in CICS TS for z/OS, Version 3.2, and can produce reports on application programs which have potential affinities.

Chapter 45. Threadsafe API and SPI commands

Most new application programming interface (API) and system programming interface (SPI) commands in CICS Transaction Server for z/OS, Version 3 Release 2 are threadsafe. Additionally, some existing commands have been made threadsafe in this release.

New API commands that are threadsafe

DOCUMENT DELETE

New SPI commands that are threadsafe

INQUIRE ASSOCIATION
INQUIRE ASSOCIATION LIST
INQUIRE IPCONN
INQUIRE LIBRARYINQUIRE LIBRARY
SET IPCONN
PERFORM JVMPOOL
SET DOCTEMPLATE

Existing API commands that are now threadsafe

WAIT JOURNALNAME
WAIT JOURNALNUM
WRITE JOURNALNAME
WRITE JOURNALNUM

API commands that are now threadsafe for local VSAM or RLS files

DELETE
ENDBR
READ
READNEXT
READPREV
RESETBR
REWRITE
STARTBR
UNLOCK
WRITE

These commands are threadsafe if the file to which they refer is defined as either local VSAM or RLS. If the file is defined as remote, or is a shared data table, a coupling facility data table, or a BDAM file the commands are not threadsafe.

Existing SPI commands that are now threadsafe

INQUIRE FILE

New SPI commands that are not threadsafe

CREATE IPCONN
CREATE LIBRARY
DISCARD IPCONN
DISCARD LIBRARY
SET LIBRARY

Chapter 46. High-level language support

This reference topic describes the high-level programming languages supported by CICS Transaction Server for z/OS, Version 3 Release 2.

COBOL

Product name	PID	Translator support	Run time support
Compilers that have been withdrawn from service on z/OS:			
OS/VS COBOL	5740-CB1 5734-CB4 5740-LM1	No	No
VS COBOL II	5668-022 5668-023 5668-958	Yes	Provided by Language Environment
COBOL/370™	5688-197	Yes	Provided by Language Environment
COBOL for OS/390 and VM V2.1	5688-197	Yes	Provided by Language Environment
COBOL for OS/390 and VM V2.2	5648-A25	Yes. The compiler provides support for the CICS integrated translator	Provided by Language Environment
Enterprise COBOL for z/OS and OS/390 (V3.1 and V3.2)	5655-G53	Yes. The compiler provides support for the CICS integrated translator	Provided by Language Environment
Compilers that are in service:			
Enterprise COBOL for z/OS and OS/390 (V3.3 and V3.4)	5655-G53	Yes. The COBOL compiler provides support for the CICS integrated translator	Provided by Language Environment

PL/I

Product name	PID	Translator support	Run time support
Compilers that have been withdrawn from service on z/OS:			
OS PL/I Optimizing Compiler V1	5724-PLI	Yes	No
OS PL/I Optimizing Compiler V2	5668-909 5668-910 5668-911	Yes	No
SAA® AD/Cycle® PL/I for MVS and VM	5688-235	Yes. The compiler provides support for the CICS integrated translator	Provided by Language Environment

Product name	PID	Translator support	Run time support
PL/I for MVS and VM V1	5688-235	Yes. The compiler provides support for the CICS integrated translator	Provided by Language Environment
VisualAge® PL/I for OS/390 V2	5655-B22	Yes. The compiler provides support for the CICS integrated translator	Provided by Language Environment
Compilers that are in service:			
Enterprise PL/I for z/OS and OS/390 V3 including Enterprise PL/I for z/OS V3.6	5655-H31	Yes. The compiler provides support for the CICS integrated translator	Provided by Language Environment

C and C++

Product name	PID	Translator support	Run time support
Compilers that have been withdrawn from service on z/OS:			
C/370™ V1	5688-040	Yes	No
C/370 V2	5688-187 5688-188	Yes	No
SAA AD/Cycle C/370	5688-216	Yes	Provided by Language Environment
C/C++ for MVS/ESA	5655-121	Yes	Provided by Language Environment
C/C++ for OS/390	Component of OS/390, (5647-A01)	Yes	Provided by Language Environment
Compilers that are in service:			
C/C++ for z/OS and OS/390	Components of z/OS (5694-A01) and z/OS.e (5655-G53)	Yes	Provided by Language Environment
z/OS V1.4 C/C++			
z/OS V1.5 C/C++			
z/OS V1.6 C/C++			
z/OS V1.7 XL C/C++		Yes. The compiler provides support for the CICS integrated translator.	
z/OS V1.8 XL C/C++			
z/OS V1.9 XL C/C++			

Java

Product name	PID	Run time support
Java products that have been withdrawn from service on z/OS:		

Product name	PID	Run time support
Java for OS/390 V1.1.8	5655-A46	Application bytecode. Most Java application bytecode should run on the IBM SDK for z/OS Java 2 Technology Edition, SDK 1.4 (5655-I56) unchanged provided the application does not used deprecated APIs.
VisualAge for Java, Enterprise Edition for OS/390	5655-JAV	Application bytecode. Most Java application bytecode should run on the IBM SDK for z/OS Java 2 Technology Edition, SDK 1.4 (5655-I56) unchanged provided the application does not used deprecated APIs.
Java products that remain in service on z/OS:		
The IBM Developer Kit for OS/390, Java 2 Technology Edition, SDK 1.3	5655-D35	Application bytecode. Most Java application bytecode should run on the IBM SDK for z/OS Java 2 Technology Edition, SDK 1.4 (5655-I56) unchanged provided the application does not used deprecated APIs.
The IBM SDK for z/OS, Java 2 Technology Edition, SDK 1.4	5655-I56	Yes
The IBM 64-bit SDK for z/OS, Java 2 Technology Edition, SDK 1.4	5655-M30	No
The IBM 31-bit SDK for z/OS, Java 2 Technology Edition, SDK 1.5	5655-N98	No
The IBM 64-bit SDK for z/OS, Java 2 Technology Edition, SDK 1.5	5655-N99	No

Part 7. Publications

Chapter 47. Improvements to the CICS Information Center

The CICS Information Center has some new functions to help you navigate and search more effectively, and to provide more information about topics. You can also refresh the documentation more conveniently.

New facility for updating documentation

If you have the information center installed locally on your workstation, or installed on a server, the new Update function for the information center can be used to list and download available documentation updates. You no longer need to download and install the complete information center each time you want to refresh the documentation.

If you have the information center installed locally on your workstation, you can access the Update function from within the information center. You need to have an Internet connection to obtain the updates. If the information center is installed on a server, your administrator can use the Update function through the command line, but it cannot be accessed from within the information center.

The Update function lists the updates that are available for your existing documentation, and any relevant new documentation that you have not installed. Each item in the list is a *feature*, which is a package that can contain one or more document plug-ins. The CICS TS 3.2 documentation has been divided into multiple document plug-ins, so you can choose to refresh only those features which contain the information that you need.

When you have selected the updates that you want, the Update function installs them. You do not need to restart the help system. The documentation in your information center refreshes automatically.

Topic headers and footers with navigation functions

Each topic in the CICS Information Center now has a header and footer, which contain information about the topic and some navigation choices.

Topic information

The header and footer for each topic show:

- The CICS product name and release. This information is in the header for the topic.
- The type of topic: task (tells you how to do something), concept (helps you to understand something), or reference (gives you more detailed technical information). This information is in the footer for the topic. If a topic does not fall into one of these categories, the topic type information in the footer does not appear.
- The date and time when the topic was updated in the information center. This information is in the footer for the topic.
- A full URL for the topic. This is in the footer for the topic. You can send this URL to someone to give them a link that takes them directly to a topic in the information center. (The URL displayed in the browser's location bar, or address bar, when you navigate around the information center is normally for the whole information center, and does not take you directly to the topic.) The URL refers to

the location from which you are currently using the information center. If you are using a local installation of the information center on your workstation, or over your organization's intranet, the URL directs the recipient to this location. If you are using the information center from the Internet location where it is provided by IBM, anyone with an Internet connection can make use of the URL.

Navigation

The topic footers contain some functions to help you with navigation:

- The 'Can also be found in' link opens the PDF that contains the topic.
- The Feedback link takes you to the CICS Information Center feedback form. You can use this form to contact IBM if you have any comments about a topic.

Search and print by topic group

You can now search in, or print, a specific group of HTML topics. Use these functions when you have already identified the area of documentation in which you are interested, and want to interact with it more efficiently.

These functions are both provided by the Quick Menu. To open the Quick Menu, in the navigation tree, hover over the title of a topic or topic group in which you are interested, and click the round button that appears next to the title.

Searching by topic group can give you a much narrower search scope than the use of a search list. Search lists can only include or exclude document plug-ins or high-level navigation categories. When you search by topic group, you can select a group at any level of the navigation, or a single topic. For example, you could search only in the listing of CICS Application programming commands.

Printing by topic group enables you to print multiple topics in HTML format with a single instruction. You can select a group at any level of the navigation for printing, and a Print Preview window enables you to confirm your selection of topics. Up to 100 topics at a time can be printed in this way.

Topic summary for search results

In your list of search results, you can now display a short description of each topic in addition to the title.

If you want to see this additional information to help you choose between the topics, select the **Show Summary** link at the top of the list of search results. Select **Hide Summary** to remove the descriptions.

Support for customer-supplied information in the information center

In CICS Transaction Server for z/OS, Version 3 Release 2, you can add your own information at a number of points (known as *anchors)* in the CICS Information Center.

The CICS Information Center runs within an Eclipse-based framework, known as the help system. The information that is displayed in the information center is organized in one or more *document plug-ins*. The Information Center for CICS Transaction Server for z/OS, Version 3 Release 2 provides a number of anchors at which you can insert your own document plug-ins. The contents of each plug-in that you supply is displayed alongside the IBM-supplied content in the information center.

A document plug-in contains the following:

- One or more files containing the content that you wish to display in the information center (for example HTML files).
- One or more XML files that define the table of contents for the plug-in. These files provide the data for the hierarchical information in the left pane of the Eclipse help window, and specify the content files at each point in the navigation.
- A plug-in *manifest* file (`plugin.xml`) that declares the tables of contents files used in the plug-in.

Anchors for customer-supplied information

You can add your own information at a number of points in the CICS Information Center. The points at which you can add information are known as *anchors*.

Anchors are provided within each of the following sections in the Information Center navigation. Within each section, the anchors follow the IBM-supplied topics.

To use the anchors, you must create an Eclipse documentation plug-in that refers to the appropriate anchor point identifier. The anchor point identifier has the following format:

```
plugin_id/navigation_file#anchor_id
```

Important: The location and identifiers of the anchor points described apply to this release only; they are not guaranteed to remain unchanged from release to release.

Location in navigation structure	Anchor point identifier
Product Overview	`com.ibm.cics.ts.productoverview.doc/ProductOverview.xml#useranchor`
What's new	`com.ibm.cics.ts.whatsnew.doc/MainNavigation_toc.xml#useranchor`
Learning Path	`com.ibm.cics.ts.doc/LearningPaths_toc.xml#useranchor`
Information Roadmaps	`com.ibm.cics.ts.doc/InformationRoadmaps_toc.xml#useranchor`
Planning	`com.ibm.cics.ts.doc/Planning_toc.xml#useranchor`
Installing	`com.ibm.cics.ts.installation.doc/installation_navigation.xml#useranchor`
Migrating	`com.ibm.cics.ts.migration.doc/migrating_navigation.xml#useranchor`
Setting up your system	`com.ibm.cics.ts.doc/Configuring_toc.xml#useranchor`
Customizing your system	`com.ibm.cics.ts.doc/Customizing_toc.xml#useranchor`
Administering	`com.ibm.cics.ts.doc/Administering_toc.xml#useranchor`
Connecting CICS to the Web	`com.ibm.cics.ts.doc/WebInterfaces_toc.xml#useranchor`

Location in navigation structure	Anchor point identifier
The CICSPlex	`com.ibm.cics.ts.doc/CICSPlex_toc.xml#useranchor`
Application programming	`com.ibm.cics.ts.doc/Programming_toc.xml#useranchor`
Security	`com.ibm.cics.ts.doc/Security_toc.xml#useranchor`
Improving performance	`com.ibm.cics.ts.performance.doc/improving_performance.xml#useranchor`
Diagnosing problems	`com.ibm.cics.ts.doc/Troubleshooting_toc.xml#useranchor`
Intercommunication	`com.ibm.cics.ts.doc/Intercommunication_toc.xml#useranchor`
Database services	`com.ibm.cics.ts.doc/DatabaseServices_toc.xml#useranchor`
External interfaces	`com.ibm.cics.ts.doc/ExternalInterfaces_toc.xml#useranchor`
Recovery and restart	`com.ibm.cics.ts.doc/Recovery_toc.xml#useranchor`
Reference → CICS supplied transactions	`com.ibm.cics.ts.doc/ReferenceTopics_toc.xml#useranchor01`
Reference → Application programming	`com.ibm.cics.ts.doc/ReferenceTopics_toc.xml#useranchor02`
Reference → System definition	`com.ibm.cics.ts.doc/ReferenceTopics_toc.xml#useranchor03`
Reference → Resource definition	`com.ibm.cics.ts.doc/ReferenceTopics_toc.xml#useranchor04`
Reference → Web User Interface views	`com.ibm.cics.ts.doc/ReferenceTopics_toc.xml#useranchor11`
Reference → Resource table descriptions	`com.ibm.cics.ts.doc/ReferenceTopics_toc.xml#useranchor12`
Reference → CICS Web support	`com.ibm.cics.ts.doc/ReferenceTopics_toc.xml#useranchor05`
Reference → System programming	`com.ibm.cics.ts.doc/ReferenceTopics_toc.xml#useranchor06`
Reference → Customization	`com.ibm.cics.ts.doc/ReferenceTopics_toc.xml#useranchor07`
Reference → Statistics	`com.ibm.cics.ts.doc/ReferenceTopics_toc.xml#useranchor08`
Reference → Trace	`com.ibm.cics.ts.doc/ReferenceTopics_toc.xml#useranchor09`
Reference → CICS utilities	`com.ibm.cics.ts.doc/ReferenceTopics_toc.xml#useranchor10`
Reference → CICS Transaction Server messages → CICS messages	`com.ibm.cics.ts.messages.doc/MessageReference.xml#dfhmessages`
Reference → CICS Transaction Server messages → CICSPlex SM messages	`com.ibm.cics.ts.messages.doc/MessageReference.xml#eyumessages`

Migrating to the CICS TS 3.2 Information Center

The CICS TS 3.2 Information Center uses Version 3.1 of the IBM Eclipse Help System. The information centers for previous versions and releases of CICS TS used Version 3.0.

If you install the information center locally on your workstation, or on a server to run on your organization's intranet, you need to install the new version of the IBM Eclipse Help System, as well as the document plug-ins for the CICS TS 3.2 Information Center.

- If you do not have any existing installations of the CICS TS Information Center on the workstation or server where you want to install, just follow the instructions in the readme file for the information center. The readme file is provided when you receive the information center on CD or download it from the IBM Publications Center.
- If you have an existing installation of the CICS TS Information Center, follow the instructions in the readme file, but be aware of these points:
  - Install the new version of the IBM Eclipse Help System and the CICS TS 3.2 Information Center to a different location from your existing IBM Eclipse Help System installation.
  - Document plug-ins created for Version 3.1 of the Help System cannot be installed on an earlier version of the Help System, because they are packaged in a different way.
  - Document plug-ins created for Version 3.0 of the Help System can be used with Version 3.1 of the Help System, although they will not exploit all the new Version 3.1 functions. This applies to any document plug-ins issued before the availability of CICS TS 3.2 for previous versions and releases of CICS TS and its related family products.
  - If you want to continue accessing old document plug-ins, you can either keep your old IBM Eclipse Help System installation to run these, or copy them into your Version 3.1 Help System. Be aware that if you have documentation for multiple releases of the same product installed in the same Information Center, it is important to pay attention to your search scope when carrying out searches, to avoid confusion.
  - If you do not want to continue accessing old document plug-ins any more, or if you have copied them all into your Version 3.1 Help System, you can delete your old Help System installation after installing Version 3.1 of the Help System.

When you have installed the CICS TS 3.2 Information Center, see the section **Using the information center** in the navigation tree, for help with using the new functions that are available.

Chapter 48. The CICS Transaction Server for z/OS library

The CICS Transaction Server for z/OS Information Center is the primary source of user information for CICS Transaction Server. A small subset (the *entitlement set*) of the CICS TS publications is available as hardcopy.

The Information Center contains:

- Information for CICS Transaction Server in HTML format.
- CICS Transaction Server books provided as Adobe Portable Document Format (PDF) files. You can use these files to print hardcopy of the 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. You will also receive a small set of essential hardcopy books.

Further copies of the Information Center on CD-ROM are available through the publications ordering system, order number SK4T-2578. Alternatively, you can download a copy of the Information Center from the IBM Publications Center free of charge.

Information provided in HTML only

Some information for CICS Transaction Server is provided in the Information Center in HTML format only.

General product information that is provided in this way includes:

- The product overview
- Learning paths
- Information Roadmaps
- The CICS glossary

Detailed product information provided in this way includes:

- CICS integration with WebSphere MQ

Books available as hardcopy

When you order CICS Transaction Server for z/OS, Version 3 Release 2, you will receive a small number of hardcopy books.

The hardcopy books are:

Memo to Licensees, GI13-0514
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 Licensed Program Specification, GC34-6810

You can order further copies of the following books, 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

CICS Transaction Server books are provided in the CICS Information Center as Adobe Portable Document Format (PDF) files. You can use these files to print hardcopy of the books.

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 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 Migration from CICS TS Version 2.3, GC34-6857
CICS Transaction Server for z/OS Migration from CICS TS Version 3.2, GC34-6858
CICS Transaction Server for z/OS Installation Guide, GC34-6812

Access to CICS

CICS Internet Guide, SC34-6831

CICS Web Services Guide, SC34-6838

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 RACF Security Guide, SC34-6835
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, SC34-6423
CICS Business Transaction Services, SC34-6824
Java Applications in CICS, SC34-6825

Diagnosis

CICS Problem Determination Guide, SC34-6826
CICS Performance Guide, SC34-6833
CICS Messages and Codes, SC34-6827
CICS Diagnosis Reference, GC34-6862
CICS Recovery and Restart Guide, SC34-6832
CICS Data Areas, GC34-6863
CICS Trace Entries, SC34-6828
CICS Supplementary Data Areas, GC34-6864
CICS Debugging Tools Interfaces Reference, GC34-6865

Communication

CICS Intercommunication Guide, SC34-6829
CICS External Interfaces Guide, SC34-6830

Databases

CICS DB2 Guide, SC34-6837

CICS IMS Database Control Guide, SC34-6834

CICS Shared Data Tables Guide, SC34-6836

CICSPlex SM books for CICS Transaction Server for z/OS

General

CICSPlex SM Concepts and Planning, SC34-6839
CICSPlex SM Web User Interface Guide, SC34-6461

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 System/390, SC34-6854

Delicensing of licensed publications

None of the publications for CICS Transaction Server for z/OS, Version 3 Release 2 is licensed. The following publications which are licensed in earlier releases are unlicensed in this release:
CICS Diagnosis Reference
CICS Data Areas
CICS Supplementary Data Areas
CICS Debugging Tools Interfaces Reference

Appendix A. New system initialization parameters

CICS Transaction Server for z/OS, Version 3 Release 2 provides new system initialization parameters.

XCFGROUP

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

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".

You can specify **XCFGROUP** on the SIT macro or as a SYSIN override. You cannot specify it as a console override.

Each CICS region can join only one XCF group, which happens when it signs on to CICS interregion communication (IRC). The default XCF group is DFHIR000.

XCF groups allow CICS regions in different MVS images within the same sysplex to communicate with each other across multi-region operation (MRO) connections.

Note: Regions in the same MVS image too, can communicate with each other using MRO, but this does not require a coupling facility. The only situation in which CICS regions in the same MVS image cannot communicate via MRO is when they are members of different XCF groups.

For introductory information about XCF/MRO, and instructions on how to set up XCF groups, see the *CICS Intercommunication Guide*.

XHFS

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

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.

Appendix B. New application programming commands

CICS Transaction Server for z/OS, Version 3 Release 2 extends the CICS application programming interface with new commands.

DOCUMENT DELETE

Delete a document.

DOCUMENT DELETE

```
►►──DOCUMENT──DELETE──DOCTOKEN(data-area)─────────────────────────────────────────►◄
```

Conditions: NOTFND

This command is threadsafe.

Description

The DOCUMENT DELETE command enables you to delete documents that are no longer required during a transaction. The command allows the application to request deletion of a document and all storage related to the document. On execution of this command, the storage allocated to the document is released immediately. If the DOCUMENT DELETE command is not invoked, the document exists until the application ends.

Options

DOCTOKEN(data-area)
specifies the 16-byte binary token of the document to be deleted.

Conditions

NOTFND
RESP2 values:

1 The document has not been created, or the name is incorrectly specified.

Appendix C. New RDO resources

There are new RDO resources in CICS Transaction Server for z/OS, Version 3 Release 2.

IPCONN resource definitions

An IPCONN (also known as an IPIC connection) is a CICS resource that represents an outbound 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. See also CONNECTION resource definitions. Like an IPCONN, a CONNECTION defines a communication link to a remote system, but in this case the connection uses the APPC or LUTYPE6.1 communication protocol (intersystem communication), or the IRC, XM, or XCF/MRO access method (multiregion operation).

The REMOTESYSTEM name on a PROGRAM definition can refer to an IPCONN definition through its IPCONN name (or to a CONNECTION definition through its CONNECTION name). This attribute is used for distributed program link.

For guidance on defining IPCONNs, see the *CICS Intercommunication Guide*.

Defining IPCONN resources

You can define IPCONN resources in the following ways:

- With the CEDA transaction.
- With the DFHCSDUP utility. For more information, see the *CICS Operations and Utilities Guide*.
- With the CREATE system programming (SPI) command. For more information, see the *CICS System Programming Reference*.
- Using autoinstall. For information about writing a program to control autoinstall of IPCONN resources, see the *CICS Customization Guide*.

Installing IPCONN definitions

To install new IPCONN definitions, put them in a group of their own which does not contain IPCONN definitions that have already been installed, then use CEDA INSTALL to install the group. (You can also install IPCONN definitions singly.)

Bear in mind that, for connectivity to be achieved when you install the IPCONN definition:

1. The TCPIPSERVICE definition named on the TCPIPSERVICE option of this IPCONN definition must also be installed in this region and must specify PROTOCOL(IPIC).
2. Corresponding IPCONN and TCPIPSERVICE definitions must be installed in the remote region.“Corresponding” means that:
   a. The HOST option of the IPCONN definition on the remote region must specify this region.

b. The PORT option of the IPCONN definition on the remote region must specify the same port number as that specified on the PORTNUMBER option of the local TCPIPSERVICE definition named by this IPCONN.

c. The TCPIPSERVICE definition on the remote region (named by the IPCONN definition on the remote region) must specify PROTOCOL(IPIC) and, on its PORTNUMBER option, the same port number as that specified by the PORT option of this IPCONN.

IPCONN attributes

Syntax

```
►►──IPCONN(IPCONNname)──GROUP(groupname)──┬──────────────────────┬──────────────────►
                                          └─DESCRIPTION(text)─┘


   ┌─APPLID(IPCONNname)─┐  ┌─AUTOCONNECT(NO)──┐          ┌─INSERVICE(YES)─┐
►──┼────────────────────┼──┼──────────────────┼─HOST(name)─┼────────────────┼────────►
   └─APPLID(applid)─────┘  └─AUTOCONNECT(YES)─┘          └─INSERVICE(NO)──┘


                        ┌─LINKAUTH(SECUSER)─┬───────────────────────┬─┐
   ┌─MAXQTIME(NO)──────┐ │                   └─SECURITYNAME(name)─┘ │
►──┼───────────────────┼─┼──────────────────────────────────────────┼──────────────►
   └─MAXQTIME(seconds)─┘ └─LINKAUTH(CERTUSER)───────────────────────┘

                                               ┌─QUEUELIMIT(NO)───────┐
►──┬─────────────────────────┬─┬─PORT(NO)─────┬─┼──────────────────────┼─────────────►
   └─NETWORKID(networkID)─┘ └─PORT(number)─┘ └─QUEUELIMIT(number)─┘


   ┌─RECEIVECOUNT(1)──────┐  ┌─SENDCOUNT(1)──────┐
►──┼──────────────────────┼──┼───────────────────┼─TCPIPSERVICE──(──name──)──────────►
   └─RECEIVECOUNT(number)─┘  └─SENDCOUNT(number)─┘


   ┌─XLNACTION(KEEP)──┐
►──┼──────────────────┼──────────────────────────────────────────────────────────────►
   └─XLNACTION(FORCE)─┘


   ┌─SSL(NO)─────────────────────────────────────────────────┐
►──┼─────────────────────────────────────────────────────────┼───────────────────────►
   └─SSL(YES)─┬──────────────────────┬─┬─────────────────┬─┘
              └─CERTIFICATE(label)─┘ └─CIPHERS(value)─┘


   ┌─USERAUTH(LOCAL)───────┐
►──┼───────────────────────┼─────────────────────────────────────────────────────────►◄
   ├─USERAUTH(IDENTIFY)────┤
   ├─USERAUTH(VERIFY)──────┤
   └─USERAUTH(DEFAULTUSER)─┘
```

Attributes

APPLID(*applid*)
specifies the application identifier (applid) of the remote system. (If the remote

system is a CICS region, its applid is defined on the APPLID parameter of its system initialization table (SIT)). The *applid* can be up to eight characters in length and must start with an alphabetic character.

Acceptable characters:

```
A-Z 0-9 $ @ #
```

Unless you are using the CREATE command, any lowercase characters you enter are converted to uppercase.

For connections to an extended recovery facility (XRF) CICS region, specify the *generic* applid of the remote region.

If you do not supply an APPLID, CICS uses the IPCONN name.

There are some rules about duplicate APPLIDs:

- You cannot install two or more IPCONN definitions that specify the same APPLID *and* the same NETWORKID. (The combination of APPLID and NETWORKID can be used to ensure unique naming of systems across the network. See the description of the NETWORKID option, below.)
- You *can* install an IPCONN definition that specifies the same APPLID as the NETNAME of an installed MRO, APPC, or LUTYPE6.1 CONNECTION definition.
- If an installed IPCONN definition has the same name as an installed CONNECTION definition, the APPLID of the IPCONN definition must match the NETNAME of the CONNECTION definition. If they do not, the message that results depends on the situation:
  - DFHIS3009 if the error is detected during IPCONN autoinstall
  - DFHAM4913 if the error is detected during IPCONN install
  - DFHZC6312 if the error is detected during CONNECTION install or autoinstall

  The IPCONN definition takes precedence over the CONNECTION definition: that is, if an IPCONN and a CONNECTION have the same name, CICS uses the IPCONN.
- a CONNECTION and an IPCONN with the same NETNAME and APPLID do not have to have the same name.

  This allows the possibility to use a distinct sysid for communication over TCP/IP rather than relying on the CICS default of routing all supported function via the IPCONN, if it exists.

The above rules are validated at install time.

AUTOCONNECT({**NO**|**YES**})

specifies whether sessions are to be established when the IPCONN definition is installed (which can happen during CICS initialization, when you issue a subsequent CEDA INSTALL command, or when you use the CEMT or EXEC CICS SET TCPIP OPEN command to start communication with TCP/IP). If the connection cannot be made at these times because the remote system is unavailable, you can subsequently acquire the link by using the CEMT or EXEC CICS SET IPCONN(*name*) INSERVICE ACQUIRED command, unless the remote system becomes available in the meantime and initiates communications.

NO CICS does not try to establish sessions when the IPCONN is installed.

YES CICS tries to establish sessions when the IPCONN is installed.

Bear in mind that, for connectivity to be achieved when you install the IPCONN definition:

1. The TCPIPSERVICE definition named on the TCPIPSERVICE option of this IPCONN definition must also be installed in this region and must specify PROTOCOL(IPIC).
2. Corresponding IPCONN and TCPIPSERVICE definitions must be installed in the remote region. “Corresponding” means that:
   a. The HOST option of the IPCONN definition on the remote region must specify this region.
   b. The PORT option of the IPCONN definition on the remote region must specify the same port number as that specified on the PORTNUMBER option of the local TCPIPSERVICE definition named by this IPCONN.
   c. The TCPIPSERVICE definition on the remote region (named by the IPCONN definition on the remote region) must specify PROTOCOL(IPIC) and, on its PORTNUMBER option, the same port number as that specified by the PORT option of this IPCONN.

You cannot specify AUTOCONNECT(YES) when PORT(NO) is specified.

CERTIFICATE(*label*)
specifies the label of an X.509 certificate to be used as a client certificate during the SSL handshake when the IPCONN is acquired, if the TCPIPSERVICE identified by the HOST and PORT is defined with SSL(CLIENTAUTH). If this attribute is omitted, the default certificate defined in the key ring for the CICS region user ID is used.

Certificate labels can be up to 32 bytes long.

The certificate must be stored in a key ring in the external security manager’s database. For more information, see the *CICS RACF Security Guide*.

If you specify this attribute you must also specify SSL(YES).

CIPHERS(*value*)
specifies a string of up to 56 hexadecimal digits that is interpreted as a list of up to 28 2-digit cipher suite codes. When you use CEDA to define the resource, CICS automatically initializes the attribute with a default list of acceptable codes, depending on the level of encryption that is specified by the ENCRYPTION system initialization parameter.

- For ENCRYPTION=WEAK, the default value is `03060102`.
- For ENCRYPTION=MEDIUM, the initial value is `0903060102`.
- For ENCRYPTION=STRONG, the initial value is `050435363738392F303132330A1613100D0915120F0C03060201`.

You can reorder the cipher codes or remove them from the initial list. However, you cannot add cipher codes that are not in the default list for the specified encryption level. To reset the value to the default list of codes, delete all of the cipher suite codes: the field is automatically repopulated with the default list.

See the *CICS RACF Security Guide* for more information.

DESCRIPTION(*text*)
You can provide a description of the resource you are defining in this field. The description text can be up to 58 characters in length. There are no restrictions

on the characters that you can use. However, if you use parentheses, ensure that for each left parenthesis there is a matching right one. If you use the CREATE command, for each single apostrophe in the text, code two apostrophes.

GROUP(*groupname*)

Every resource definition must have a GROUP name. The resource definition becomes a member of the group and is installed in the CICS system when the group is installed.

Acceptable characters:

```
A-Z 0-9 $ @ #
```

Any lower case characters you enter are converted to upper case.

The GROUP name can be up to eight characters in length. Lowercase characters are treated as uppercase characters. Do not use group names beginning with DFH, because these characters are reserved for use by CICS.

HOST(*hostname*)

specifies the host name of the target system: for example, `abc.example.com`. The name can be up to 116 characters long.

Acceptable characters:

```
a-z 0-9 - .
```

Any upper case characters you enter are converted to lower case.

The HOST attribute must contain only alphanumeric characters, hyphens (-) or periods (.). The HOST attribute must not contain hexadecimal escape sequences.

You can specify a dotted-decimal IPv4 address as a host name, but IPv6 addresses are not supported.

CICS validates the *hostname* at define time.

This parameter is optional when SENDCOUNT is zero. It is a required parameter when SENDCOUNT is greater than zero.

INSERVICE({**NO**|**YES**})

specifies the status of the IPCONN when it is installed.

NO The connection can neither receive messages nor transmit output.

YES The connection is available for use.

IPCONN(*name*)

specifies the name of this IPCONN definition. The name can be up to eight characters in length.

Acceptable characters:

```
A-Z 0-9 $ @ #
```

Unless you are using the CREATE command, any lowercase characters you enter are converted to uppercase.

If this IPCONN is to be used for distributed program link (DPL), its name must match the 4-character “local name” (SYSID) by which CICS knows the remote system, padded with four trailing blanks.

Note: The name (SYSID) of the remote, target region, of a DPL request may be specified by any of the following:

1. The REMOTESYSTEM option of the installed PROGRAM definition
2. The SYSID option of the EXEC CICS LINK PROGRAM command
3. The dynamic routing program

The IPCONN name can be the same as the name of an installed MRO or APPC CONNECTION definition.

LINKAUTH({**CERTUSER**|**SECUSER**})
Specifies how the user ID for link security is established in a CICS system with security initialized (SEC=YES).

CERTUSER
TCP/IP communication with the partner system must be configured for SSL and a certificate must be received from the partner system during SSL handshake.

The IPCONN must refer to a TCPIPSERVICE that is defined with SSL(CLIENTAUTH).

The received certificate must be defined to the external security manager so that it is associated with a user ID, which is used to establish link security.

SECUSER
Specifies that the user ID specified in SECURITYNAME is used to establish link security.

If you do not specify a value for SECURITYNAME, CICS uses the default user ID.

MAXQTIME({**NO**|*seconds*})
specifies the maximum time that queued allocate requests, waiting for free sessions on this connection, can wait before the queue is purged. Note that:

1. The maximum queuing time is used only if a limit to the length of the queue is specified on the QUEUELIMIT option.
2. The time limit is applied only when the queue length has reached the QUEUELIMIT value.

NO CICS maintains the queue of allocate requests that are waiting for a free session. No limit is set on the length of time that requests can remain queued (though the DTIMOUT mechanisms can apply to individual requests).

seconds
The approximate maximum time, in seconds, that allocate requests waiting for a free session can be queued, when this connection appears to be unresponsive; *seconds* must be in the range 0 through 9999.

When the queue of allocate requests reaches its maximum length (specified by QUEUELIMIT), and a new allocate request is received for the connection, if the rate of processing for the queue indicates that, on average, the new allocate would take more than the maximum queue time, the queue is purged, and message DFHIS500 is issued. When the queue is purged, queued allocate requests return SYSIDERR.

No further queuing takes place until the connection has successfully freed a session. At this point, CICS issues DFHIS5001 and resumes normal queuing.

NETWORKID(*networkID*)

specifies the network ID of the remote system. (The remote system's network ID is either its VTAM NETID or, for VTAM=NO systems, the value of its UOWNETQL system initialization parameter.)

If NETWORKID is not specified, CICS assumes that the remote system is in the same network as the local system. In this instance, CICS uses the VTAM NETID, or the value of the UOWNETQL system initialization parameter, of this CICS (that is, the CICS on which this definition is installed).

Specify NETWORKID if you want to connect to a remote system that is in a different network, and so has a different VTAM NETID or UOWNETQL value. In this instance, it could be possible for two or more remote systems to have the same APPLID. (Although CICS APPLIDs must be unique within a sysplex, you may, for example, want to connect to a system outside the sysplex or in a different sysplex.) The combination of APPLID and NETWORKID ensures that the remote system is referred to by a unique name.

NETWORKID must match the remote system's network ID.

When not specified, the NETWORKID value is derived when the IPCONN is first installed and is not changed between warm starts, even if the local NETID changes.

The name can be up to eight characters in length and follows assembler language rules. It must start with an alphabetic character.

Acceptable characters:

```
A-Z 0-9 $ @ #
```

Unless you are using the CREATE command, any lowercase characters you enter are converted to uppercase.

PORT(**NO**|*port*)

specifies, in the range 1 through 65535, the decimal number of the port that the remote region will listen on. The port number is combined with the HOST value to determine the destination for outbound requests on this IPCONN.

NO is not valid for CICS to CICS IPCONNs.

NO forces the value of AUTOCONNECT to NO.

Specify NO for a non-CICS client if this IPCONN is not used for outbound requests (that is, it has no send sessions).

QUEUELIMIT({**NO**|*number*})

specifies the maximum number of allocate requests that CICS is to queue while waiting for free sessions:

NO
There is no limit to the number of allocate requests that CICS can queue while waiting for a free session.

number
The maximum number of allocate requests, in the range 0 through 9999, that CICS can queue on the connection while waiting for a free session. When the number of queued allocate requests reaches this limit, subsequent allocate requests fail, returning SYSIDERR, until the queue drops below the limit.

RECEIVECOUNT(*number*)

specifies, in the range 1-999, the number of receive sessions; that is, sessions that receive incoming requests. The actual number of receive sessions that are used depends also on the number of send sessions defined in the remote system. When the connection is established, these values are exchanged and the lower value is used.

SECURITYNAME(*user ID*)
specifies the security name of the remote system, to be used for link security.

In a CICS system with security initialized (SEC=YES), and with LINKAUTH(SECUSER) in use, the security name is used to establish the authority of the remote system.

The security name must be a valid RACF user ID on this region. Access to protected resources on this region is based on the RACF user profile and its group membership.

The default value is the default user ID.

SENDCOUNT(*number*)
specifies, in the range 0-999, the number of send sessions; that is, sessions that send outgoing requests. The actual number of send sessions that are used depends also on the number of receive sessions defined in the remote system. When the connection is established, these values are exchanged and the lower value is used. If 0 is specified, then this IPCONN can only process incoming work. It cannot send requests to the connected system.

SENDCOUNT(0) is not valid for CICS to CICS IPCONNs.

SENDCOUNT(0) forces PORT(NO). A SENDCOUNT value greater than zero requires PORT to have a numeric value.

An attempt to acquire an IPCONN that has SENDCOUNT(0) will fail because there is no port defined.

SSL({**NO**|**YES**})
whether Secure Sockets Layer (SSL) is used for encrypting the transmitted data.

NO The Secure Sockets Layer (SSL) will not be used. No security checks are applied when the connection is being acquired. No encryption is applied to outbound messages.

YES If the SEC system initialization parameter is set to ″YES″, the Secure Sockets Layer (SSL) is used. If the TCPIPSERVICE identified by the HOST and PORT is defined with SSL(CLIENTAUTH), CICS extracts the client certificate named in the CERTIFICATE parameter, and uses it when acquiring the IPCONN to the partner system. SSL encryption processing is applied to all messages sent from this IPCONN. The level of encryption depends on the value of the CIPHERS option.

TCPIPSERVICE(*name*)
specifies the name of a TCPIPSERVICE definition, with PROTOCOL(IPIC), that defines the attributes of the inbound processing for this IPCONN.

USERAUTH({**LOCAL**|**IDENTIFY**|**VERIFY**|**DEFAULTUSER**})
Specifies how the user ID for attach-time user security is established in a CICS system with security initialized (SEC=YES).

LOCAL
CICS does not accept a user ID or password from clients. All requests will run under the link user ID, or the default user ID if there is no link user ID.

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

VERIFY
Incoming attach requests must specify a user identifier and a user password. Specify VERIFY when connecting systems are unidentified and cannot be trusted.

DEFAULTUSER
CICS will not accept a user ID and password from the partner system. All requests run under the default user ID.

XLNACTION({**KEEP**|**FORCE**})
specifies the action to be taken when a new logname is received from the partner, system. (Receipt of a new logname indicates that the partner has deleted its recovery information.)

FORCE
Before any new work with the new logname is started, the predefined decisions for indoubt units of work (UOWs), as defined by the indoubt attributes of the TRANSACTION definition, are implemented. CICS also deletes any information retained for possible resolution of UOWs that were indoubt on the partner system.

Attention: Data integrity may be compromised if you use this option.

KEEP Recovery information is kept, and no predefined actions are taken for indoubt units of work.

The connection is unable to perform new work that requires sync level 2 protocols until all outstanding recoverable work with the partner (that is, indoubt UOWs, or information relevant to UOWs that were indoubt on the partner system under the old logname) is completed, using the CEMT or SPI interface. This means that if the connection is being used for CICS-to-CICS communication (which always uses the synclevel 2 protocols) the connection cannot be acquired until all outstanding recoverable work with the partner has completed.

Note: When an IPCONN to a Java client is autoinstalled by the ECI Resource Adapter, CICS is unaware of any outstanding work until the Adapter sends a resynchronization flow. For such connections, lognames are not used and the XLNACTION attribute is ignored.

For information about the ECI Resource Adapter, see *Java Applications in CICS*.

Appendix D. New system programming commands

CICS Transaction Server for z/OS, Version 3 Release 2 extends the CICS system programming interface with new commands.

CREATE IPCONN

Define and install an IPCONN in the local CICS region.

CREATE IPCONN

```
►►──CREATE IPCONN(data-value)─────────────────────────────────────────────────►

                                                ┌─LOG────────────────┐
►──ATTRIBUTES(data-value)──┬────────────────────┬┼────────────────────┼───────►◄
                           └─ATTRLEN(data-value)┘├─NOLOG──────────────┤
                                                 └─LOGMESSAGE(cvda)───┘
```

CREATE IPCONN attribute values:

```
                              ┌─APPLID(IPCONN name)─┐ ┌─AUTOCONNECT(NO)──┐
├──┬──────────────────────┬───┼─────────────────────┼─┼──────────────────┼────►
   └─DESCRIPTION(char58)──┘   └─APPLID(applid)──────┘ └─AUTOCONNECT(YES)─┘

            ┌─INSERVICE(YES)─┐
►──HOST(name)──┼────────────────┼─────────────────────────────────────────────►
            └─INSERVICE(NO)──┘

   ┌─LINKAUTH(SECUSER)──┬────────────────────┬─┐
   │                    └─SECURITYNAME(name)─┘ │  ┌─MAXQTIME(NO)──────┐
►──┼────────────────────────────────────────────┼──┼───────────────────┼─────►
   └─LINKAUTH(CERTUSER)─────────────────────────┘  └─MAXQTIME(seconds)─┘

                              ┌─PORT(NO)─────┐ ┌─QUEUELIMIT(NO)─────┐
►──┬──────────────────────┬───┼──────────────┼─┼────────────────────┼─────────►
   └─NETWORKID(networkID)─┘   └─PORT(number)─┘ └─QUEUELIMIT(number)─┘

   ┌─RECEIVECOUNT(1)──────┐   ┌─SENDCOUNT(1)──────┐
►──┼──────────────────────┼───┼───────────────────┼──TCPIPSERVICE(name)───────►
   └─RECEIVECOUNT(number)─┘   └─SENDCOUNT(number)─┘

   ┌─XLNACTION(KEEP)──┐
►──┼──────────────────┼───────────────────────────────────────────────────────►
   └─XLNACTION(FORCE)─┘

   ┌─SSL(NO)──────────────────────────────────────────────────┐
►──┼──────────────────────────────────────────────────────────┼───────────────►
   └──┬─────────┬──┬────────────────────┬──┬───────────────┬──┘
      └─SSL(YES)┘  └─CERTIFICATE(label)─┘  └─CIPHERS(value)┘

   ┌─USERAUTH(LOCAL)───────┐
►──┼───────────────────────┼──────────────────────────────────────────────────┤
   ├─USERAUTH(DEFAULTUSER)─┤
   ├─USERAUTH(IDENTIFY)────┤
   └─USERAUTH(VERIFY)──────┘
```

Conditions: ILLOGIC, INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use

ATTRIBUTES(*data-area*)
instead of
ATTRIBUTES(*data-value*)

Description

The CREATE IPCONN command installs an IPCONN definition with the attributes specified on the command. It does not use a resource definition stored in the CSD. If there is already an IPCONN with the name you specify in the local CICS region, the new definition replaces the old one; if not, the new definition is added.

Note: CREATE IPCONN creates a TCP/IP communication link to a remote system. See also CREATE CONNECTION. Like an IPCONN, a CONNECTION defines a communication link to a remote system, but in this case the connection uses the APPC or LUTYPE6.1 communication protocol (intersystem communication), or the IRC, XM, or XCF/MRO access method (multiregion operation).

Bear in mind that, for connectivity to be achieved when you install the IPCONN definition:

1. The TCPIPSERVICE definition named on the TCPIPSERVICE option of this IPCONN definition must also be installed in this region and must specify PROTOCOL(IPIC).
2. Corresponding IPCONN and TCPIPSERVICE definitions must be installed in the remote region. By “corresponding” we mean that:
    a. The HOST option of the IPCONN definition on the remote region must specify this region.
    b. The PORT option of the IPCONN definition on the remote region must specify the same port number as that specified on the PORTNUMBER option of the local TCPIPSERVICE definition named by this IPCONN.
    c. The TCPIPSERVICE definition on the remote region (named by the IPCONN definition on the remote region) must specify PROTOCOL(IPIC) and, on its PORTNUMBER option, the same port number as that specified by the PORT option of this IPCONN.

If this IPCONN is to be used for distributed program link (DPL), its name must match the 4-character “local name” (SYSID) by which CICS knows the remote system, padded with four trailing blanks.

Note: The name (SYSID) of the remote, target region, of a DPL request may be specified by any of the following:
1. The REMOTESYSTEM option of the installed PROGRAM definition
2. The SYSID option of the EXEC CICS LINK PROGRAM command
3. The dynamic routing program

For details of the attributes of IPCONN and TCPIPSERVICE definitions, see the *CICS Resource Definition Guide*. For guidance on defining IPIC connections, see the *CICS Intercommunication Guide*.

A syncpoint is implicit in CREATE IPCONN processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the CREATE executes successfully, and rolled back if not.

See Creating resource definitions for other general rules governing CREATE commands.

Options

ATTRIBUTES(*data-value***)**
specifies the attributes of the IPCONN being added. The list of attributes must be coded as a single character string using the syntax shown in **IPCONN attributes**. See The ATTRIBUTES option for general rules for specifying attributes, and the *CICS Resource Definition Guide* for details about specific attributes.

ATTRLEN(*data-value***)**
specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length may not exceed 32767 bytes.

LOGMESSAGE(*cvda*)
specifies whether CICS should log the attributes used for the resource that is created. CVDA values are:

LOG
The resource's attributes are logged to the CSDL transient data queue.

NOLOG
The resources attributes are not logged.

IPCONN(*data-value***)**
specifies the 8-character name of the connection to the remote system (that is, the name of the IPCONN definition to be created).

Conditions

ILLOGIC
RESP2 values:

2 The command cannot be executed because an earlier CONNECTION or TERMINAL pool definition has not yet been completed.

INVREQ
RESP2 values:

n There is a syntax error in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. See EXEC CICS CREATE RESP2 values for information on RESP2 values.

7 The LOGMESSAGE cvda value is not valid.

200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET, or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LENGERR
RESP2 values:

1 The length you have specified in ATTRLEN is negative.

NOTAUTH
RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

DISCARD IPCONN

Remove an IPCONN definition.

DISCARD IPCONN

```
►►──DISCARD IPCONN(data-value)─────────────────────────────────────────────►◄
```

Conditions: INVREQ, NOTAUTH, SYSIDERR

Description

The DISCARD IPCONN command removes an IPCONN definition from the local CICS system.

You cannot discard an IPCONN unless it is in OUTSERVICE status.

See Discarding resource definitions for general information about DISCARD commands.

Options

IPCONN(*data-value*)
specifies the 8-character name of the IPCONN definition to be discarded.

Conditions

INVREQ
RESP2 values:

9 The IPCONN is not out of service.

NOTAUTH
RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

SYSIDERR
RESP2 values:

9 The IPCONN name was not found.

INQUIRE ASSOCIATION

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

INQUIRE ASSOCIATION

```
►►──INQUIRE ASSOCIATION(data-value)──┬─────────────┬──────────────►◄
                                     └─┤ options ├─┘
```

Conditions: NOTAUTH, TASKIDERR

This command is threadsafe.

Options:

```
├──┬──────────────────────────┬──────────────────────────────────────┤
   ├─APPLDATA(data-area)──────┤
   ├─APPLID(data-area)────────┤
   ├─CLIENTIPADDR(data-area)──┤
   ├─CLIENTPORT(data-area)────┤
   ├─FACILNAME(data-area)─────┤
   ├─FACILTYPE(cvda)──────────┤
   ├─INITUSERID(data-area)────┤
   ├─IPCONN(data-area)────────┤
   ├─IPFAMILY(cvda)───────────┤
   ├─LUNAME(data-area)────────┤
   ├─MVSIMAGE(data-area)──────┤
   ├─NETID(data-area)─────────┤
   ├─ODAPPLID(data-area)──────┤
   ├─ODCLNTIPADDR(data-area)──┤
   ├─ODCLNTPORT(data-area)────┤
   ├─ODFACILNAME(data-area)───┤
   ├─ODFACILTYPE(cvda)────────┤
   ├─ODIPFAMILY(cvda)─────────┤
   ├─ODLUNAME(data-area)──────┤
   ├─ODNETID(data-area)───────┤
   ├─ODNETWORKID(data-area)───┤
   ├─ODSTARTTIME(data-area)───┤
   ├─ODTASKID(data-area)──────┤
   ├─ODTRANSID(data-area)─────┤
   ├─ODUSERID(data-area)──────┤
   ├─PROGRAM(data-area)───────┤
   ├─SERVERIPADDR(data-area)──┤
   ├─SERVERPORT(data-area)────┤
   ├─STARTTIME(data-area)─────┤
   ├─TCPIPJOB(data-area)──────┤
   ├─TCPIPSERVICE(data-area)──┤
   ├─TCPIPZONE(data-area)─────┤
   ├─TRNGRPID(data-area)──────┤
   ├─TRANSACTION(data-area)───┤
   ├─USERCORRDATA(data-area)──┤
   └─USERID(data-area)────────┘
```

For more information about the use of CVDAs, see CICS-value data areas (CVDAs).

Description

The INQUIRE ASSOCIATION command retrieves information about how a task was started, based on a task number.

Association records are identified by task numbers. Thus the input data (specified on the ASSOCIATION option of the INQUIRE command) is the task number. The association data is retrieved from the specified task's associated data control block (ADCB).

The associated data control block is built during task attach processing. It might contain information about another CICS task that acted as the point of origin for this task.

INQUIRE ASSOCIATION enables you to inquire about a single task's association data in the local region. Browsing is not supported.

Options

APPLDATA(*data-area*)
returns the 40-character value of the application data associated by CICS with the socket that received the request that started this task. If the task was not started through a socket then APPLDATA is blank.

The 40-character application data consists of:

A 24-byte prefix owned by the Sockets domain

Bytes 01-03
"DFH"

Byte 04

I	Inbound (listen and accept)
O	Outbound (connect)

Bytes 05-12
The APPLID of this region

Bytes 13-16
The ID of the transaction that created the socket:

CIEP	ECI inbound
CIRR	IIOP inbound
CISC	IPIC outbound
CISS	IPIC inbound
CWXN	HTTP inbound
CWXU	USER inbound
xxxx	HTTP outbound
xxxx	IIOP outbound

Bytes 17-24
The network protocol: one of ECI, HTTP, IIOP, IPIC, or USER

A 16-byte suffix owned by the using domain
The contents of the suffix depends on the state of the connection:

The TCPIPSERVICE is listening on the socket

Bytes 25-32
The TCPIPSERVICE name

Bytes 33-40
The first 8 bytes of the TCPIPSERVICE description

After the IPCONN has been acquired

Bytes 25-32
The IPCONN name

Bytes 33-40
The APPLID of the partner region

Default for outbound connections

Bytes 25-40
Blank

This data can be used to correlate CICS connection information with z/OS Communication Server connection information.

APPLID(*data-area***)**
returns the 8 character APPLID of the CICS region this task is running in.

ASSOCIATION(*data-value***)**
specifies the 4-byte number of the task for which you want to retrieve association data.

CLIENTIPADDR(*data-area***)**
returns, into a 39-character area, the IP address of the TCP/IP client that requested this task to start. When the IPFAMILY option returns “IPV4”, the returned address is a 15-character, dotted-decimal, IP Version 4 address, padded with blanks. If this task was not started from a TCP/IP client, CLIENTIPADDR returns blanks.

CLIENTPORT(*data-area***)**
returns, in fullword binary form, the number of the port that the TCP/IP stack used to send the request that resulted in this task being attached. If the task was not started in this way, CLIENTPORT returns zero.

FACILNAME(*data-area***)**
returns the 8-character name of the facility associated with the initiation of this task. If the task was started by an unnamed facility, FACILNAME returns blanks.

FACILTYPE(*cvda***)**
returns a CVDA value identifying the type of facility that initiated this task. CVDA values are:

APPC APPC connection

BRIDGE
3270 bridge

ECIIP ECI over TCP/IP

IIOP IIOP request

IPIC IP interconnectivity connection (IPCONN)

LU61 LU6.1 connection

MRO MRO connection

NONE Not started by any of the listed facility-types

RRSUR

RZINSTOR

SCHEDULER
Scheduled task

SOCKET
TCP/IP socket request

START
Non-terminal-related START command

STARTTERM
Terminal-related START command

TERMINAL
User terminal

TRANDATA
Transaction data

UNKNOWN
The facility-type is unknown

WEB Web service request

XMRUNTRAN

INITUSERID(*data-area***)**
returns the 8-character user ID of the initiating task (the task that caused this one to be attached).

IPCONN(*data-area***)**
returns the 8-character name of any IPCONN that was used to receive a request that resulted in this task starting. If the task was not started in this way, IPCONN returns blanks. This field contains a non-blank value only when the FACILTYPE is IPIC.

IPFAMILY(*cvda***)**
returns a CVDA value indicating the form of TCP/IP addressing used by this task. In this release, only IP Version 4 addressing is supported. CVDA values are:

IPV4 The request that caused CICS to initiate this task arrived at a TCPIPSERVICE that made use of an IP Version 4 address.

NOTAPPLIC
There is no TCP/IP client associated with this task.

LUNAME(*data-area***)**
returns the 8-character fully-qualified network name of the terminal from which this task was started. If the task was started from an IPIC (IPCONN), ISC over SNA (APPC), or MRO session, LUNAME returns the network name of the remote region. If the task was not started from a terminal, nor from an IPCONN, APPC, or MRO session, LUNAME returns blanks. For OTS transactions, LUNAME returns blanks.

MVSIMAGE(*data-area***)**
returns the 8-character name of the MVS image associated with the TCPIPSERVICE used to receive a request that resulted in this task starting. If the task was not started in this way, MVSIMAGE returns blanks.

Note: This function is dependent on Communication Server TCP/IP Network Access Control support being activated and the CLIENTIPADDRESS being configured into a Network Security Zone.

NETID(*data-area***)**
returns the 8-character network ID of the terminal from which this task was started.

ODAPPLID(*data-area***)**
returns the 8-character APPLID taken from the Origin Descriptor associated with this task.

ODCLTIPADDR(*data-area***)**
returns, into a 39-character area, the IP address of the TCP/IP client that

requested the originating task to start. (The originating task is the one that forms the root of a distributed transaction.) When the ODIPFAMILY option returns “IPV4”, the returned address is a 15-character, dotted-decimal, IP Version 4 address, padded with blanks. If the originating task was not started from a TCP/IP client, ODCLNTIPADDR returns blanks.

ODCLNTPORT(*data-area***)**
returns, in fullword binary form, the number of the port which the TCP/IP stack used to send the request that resulted in the originating task being attached. If the originating task was not started in this way, ODCLNTPORT returns zero.

ODFACILNAME(*data-area***)**
If the facility associated with the initiation of the originating task is a transient data queue, a terminal, or a system, ODFACILNAME returns the 4-character name of the facility. If the originating task was not started in any of these ways, ODFACILNAME returns blanks.

ODFACILTYPE(*cvda***)**
returns a CVDA value identifying the type of facility that initiated the originating task that is associated with this task. CVDA values are:

APPC APPC connection

BRIDGE
3270 bridge

ECIIP ECI over TCP/IP

IIOP IIOP request

IPIC IP interconnectivity connection (IPCONN)

LU61 LU6.1 connection

MRO MRO connection

NONE Not started by any of the listed facility-types

RRSUR

RZINSTOR

SCHEDULER
Scheduled task

SOCKET
TCP/IP socket request

START
Non-terminal-related START command

STARTTERM
Terminal-related START command

TERMINAL
User terminal

TRANDATA
Transaction data

UNKNOWN
The facility-type is unknown

WEB Web service request

XMRUNTRAN

ODIPFAMILY(*cvda***)**
returns a CVDA value indicating form of TCP/IP addressing used by the originating task. In this release only IP Version 4 addressing is supported. CVDA values are:

IPV4 The request that caused CICS to initiate the originating task arrived at a TCPIPSERVICE that made use of an IP Version 4 address.

NOTAPPLIC
There is no TCP/IP client associated with this task.

ODLUNAME(*data-area***)**
returns the 8-character network logical unit name of the terminal from which the originating task was started. If the originating task was started from an IPIC (IPCONN), ISC over SNA (APPC), or MRO session, ODLUNAME returns the network name of the remote region. If the originating task was not started from a terminal, nor from an IPCONN, APPC, or MRO session, ODLUNAME returns blanks. For OTS transactions, ODLUNAME returns blanks.

ODNETID(*data-area***)**
returns the 8-character network ID of the terminal (terminal, APPC peer, and so on) from which the originating task was started.

ODNETWORKID(*data-area***)**
returns the 8-character network qualifier for the origin region APPLID on which the task ran.

ODSTARTTIME(*data-area***)**
returns a 21-character representation of the time when the originating task was started. The time is in the form `yyyymmddhhmmss.ssssss`.

ODTASKID(*data-area***)**
returns the 4-byte packed decimal identifier of the originating task that is associated with this task.

ODTRANSID(*data-area***)**
returns the 4-character name of the transaction under which the originating task ran.

ODUSERID(*data-area***)**
returns the 8-character user ID under which the originating task ran.

PROGRAM(*data-area***)**
returns the 8-character name of the first program invoked by a task executing this transaction.

SERVERIPADDR(*data-area***)**
returns, into a 39-character area, the IP address of the TCP/IP server that scheduled this task. When the IPFAMILY option returns “IPV4”, the returned address is a 15-character, dotted-decimal, IP Version 4 address, padded with blanks. If this task was not started from a TCP/IP server, SERVERIPADDR returns blanks.

SERVERPORT(*data-area***)**
returns, in fullword binary form, the number of the port on which the TCPIPSERVICE that received the request that resulted it this task being attached, is listening. If the task was not started in this way, SERVERPORT returns zero.

STARTTIME(*data-area***)**
returns a 21-character representation of the time when this task was started. The time is in the form `yyyymmddhhmmss.ssssss`.

TCPIPJOB(*data-area***)**
returns the 8-character name of the TCP/IP job associated with the IPCONN that received the request that resulted in this task starting. If the task was not started in this way, TCPIPJOB returns blanks.

Note: This function is dependent on Communication Server TCP/IP Network Access Control support being activated and the CLIENTIPADDRESS being configured into a Network Security Zone.

TCPIPSERVICE(*data-area*)
returns the 8-character name of the TCPIPSERVICE associated with the IPCONN that received the request that resulted in this task starting. If the task was not started in this way, TCPIPSERVICE returns blanks.

TCPIPZONE(*data-area*)
returns the 8-character name of the TCP/IP network security zone, if any, associated with the IPCONN that received the request that resulted in this task starting. If there is no TCP/IP network security zone, or the task was not started in this way, TCPIPZONE returns blanks.

Note: This function is dependent on Communication Server TCP/IP Network Access Control support being activated and the CLIENTIPADDRESS being configured into a Network Security Zone.

TRNGRPID(*data-area*)
returns, in a 28-byte area, a mixture of hexadecimal and character data that represents the transaction group ID of the origin transaction.

TRANSACTION(*data-area*)
returns the 4-character name of the transaction that this task is executing.

USERCORRDATA(*data-area*)
returns, in a 64-byte area, the user correlator data that was added to the associated data origin descriptor by means of an XAPADMGR global user exit program. This field is created when the originating task is started. If the global user exit program is not driven at that point, USERCORRDATA returns blanks.

USERID(*data-area*)
returns the 8-character user ID associated with this task.

Conditions

INVREQ
RESP2 values:

2 The command was specified with no arguments.

NOTAUTH
RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

TASKIDERR
RESP2 values:

1 The task specified on the ASSOCIATION option could not be found.

INQUIRE ASSOCIATION LIST

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

INQUIRE ASSOCIATION LIST

```
►►──INQUIRE ASSOCIATION LIST LISTSIZE(data-area)──────────────────────────────────►

►──┬──────────────────────────────────────────┬───────────────────────────────────►◄
   └─USERCORRDATA(data-value)──SET(ptr-ref)───┘
```

Conditions: NOTAUTH

This command is threadsafe.

Description

The INQUIRE ASSOCIATION LIST command returns a list of tasks, in the local region, that have matching correlation information contained within their associated data control blocks (ADCBs).

Tracking of tasks that communicate using IP connections (IPCONNs) is superficially not unlike tracking the various tasks of a distributed unit of work (UOW). However, because the tracking applies to components communicating over a TCP/IP network, you need different tools to manage it.

INQUIRE ASSOCIATION LIST allows you to filter tasks on user correlation data that has been added to the tasks' associated data origin descriptors by an XAPADMGR global user exit program. You can also search on any of the CICS-provided fields in the origin data portion of the Association Data to find those tasks and transaction group IDs that share a set of common values.

The command returns, in SET, a list of task numbers that are associated with the user correlation data specified in USERCORRDATA. The number of items in the list is returned in LISTSIZE.

Options

LISTSIZE(*data-area***)**
returns, as a fullword binary number, the number of items in the list addressed by the SET option. If the USERCORRDATA filter produces no results (that is, there are no tasks in the category requested), LISTSIZE returns zero.

SET(*ptr-ref***)**
returns the address of a list of 4-byte, packed-decimal, task numbers. Each entry in the list identifies a task that has user correlation data in its associated data control block that matches that specified on the USERCORRDATA filter. If there are no tasks in the category requested, the SET pointer contains a null value.

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

USERCORRDATA(*data-value***)**
specifies a subset (up to 64 bytes) of the user correlation data added to the associated data origin descriptor by an XAPADMGR global user exit program. This data is used as a filter to return a list of task numbers that match this request.

The filter can contain the following “wildcard” characters:

? matches exactly one arbitrary character.

***** matches zero or more arbitrary characters.

If USERCORRDATA is omitted or left blank, the list contains all the tasks in the region.

If the USERCORRDATA filter produces no results, the LISTSIZE returned is zero and the pointer returned in SET is NULL.

Conditions

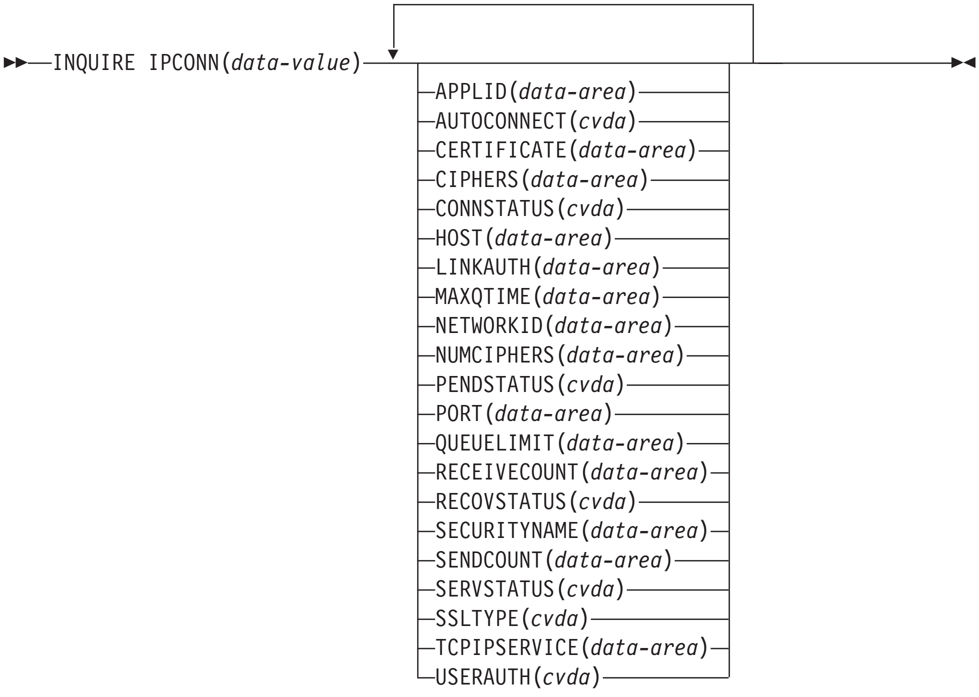
NOTAUTH
RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

INQUIRE IPCONN

Retrieve information about an IPCONN.

INQUIRE IPCONN



```
                                   ┌────────────────────────────┐
                                   ▼                            │
►►──INQUIRE IPCONN(data-value)──────┬──────────────────────────┬─┴──────────►◄
                                    ├─APPLID(data-area)────────┤
                                    ├─AUTOCONNECT(cvda)────────┤
                                    ├─CERTIFICATE(data-area)───┤
                                    ├─CIPHERS(data-area)───────┤
                                    ├─CONNSTATUS(cvda)─────────┤
                                    ├─HOST(data-area)──────────┤
                                    ├─LINKAUTH(data-area)──────┤
                                    ├─MAXQTIME(data-area)──────┤
                                    ├─NETWORKID(data-area)─────┤
                                    ├─NUMCIPHERS(data-area)────┤
                                    ├─PENDSTATUS(cvda)─────────┤
                                    ├─PORT(data-area)──────────┤
                                    ├─QUEUELIMIT(data-area)────┤
                                    ├─RECEIVECOUNT(data-area)──┤
                                    ├─RECOVSTATUS(cvda)────────┤
                                    ├─SECURITYNAME(data-area)──┤
                                    ├─SENDCOUNT(data-area)─────┤
                                    ├─SERVSTATUS(cvda)─────────┤
                                    ├─SSLTYPE(cvda)────────────┤
                                    ├─TCPIPSERVICE(data-area)──┤
                                    └─USERAUTH(cvda)───────────┘
```

Conditions: END, ILLOGIC, NOTAUTH, SYSIDERR

For more information about the use of CVDAs, see CICS-value data areas (CVDAs).

This command is threadsafe.

Description

The INQUIRE IPCONN command retrieves information about an IPIC connection (also known as an "*IPCONN*"). An IPCONN is a Transport Control Protocol/Internet Protocol (TCP/IP) communication link from your local CICS region to another CICS region or another system.

Note:

- See also INQUIRE CONNECTION. The INQUIRE CONNECTION command, which returns information about MRO and ISC over SNA connections.

  For information about the different kinds of intercommunication connections, see the *CICS Intercommunication Guide*.
- The *outbound* attributes of the IPCONN are specified by an IPCONN definition. The *inbound* attributes of the connection are specified by the TCPIPSERVICE definition named on the TCPIPSERVICE option of the IPCONN definition.

Browsing

You can also browse through all of the IPCONN definitions installed in your system by using the browse options (START, NEXT, and END) on INQUIRE IPCONN commands. See Browsing resource definitions for general information about browsing, including syntax, exception conditions, and examples.

Options

APPLID(*data-area*)
returns the 8-character name by which the remote system is known to the network (taken from the APPLID option of the IPCONN definition). This is the application identifier (*applid*) of the remote system, as specified on the APPLID option of its system initialization table. For XRF systems it is the generic applid.

AUTOCONNECT(*cvda*)
returns a CVDA value identifying which AUTOCONNECT option has been specified in the IPCONN definition. CVDA values are:

AUTOCONN
AUTOCONNECT(YES) has been specified on the IPCONN definition.

NONAUTOCONN
AUTOCONNECT(NO) has not been specified for the IPCONN definition.

CERTIFICATE(*data-area*)
returns a 32-character area containing the label of the certificate, within the key ring, that is used as a client certificate in the SSL handshake for outbound IPCONN connections. If the label is blank, the certificate nominated as the default for the key ring is used.

CIPHERS(*data-area*)
returns a 56-character area containing the list of cipher suites that is used to negotiate with clients during the SSL handshake. The list is set by the ENCRYPTION system initialization parameter, but you can edit the list to remove or change the order of cipher suites. See the *CICS RACF Security Guide*.

CONNSTATUS(*cvda*)
returns a CVDA value identifying the state of the IPCONN between CICS and the remote system. CVDA values are:

ACQUIRED
The IPCONN is acquired. The criterion for ACQUIRED is that the capabilities exchange is complete. (The capabilities exchange is how two connected CICS regions discover the levels of service that they can collectively support; for example, the syncpoint level, and security protocols such as SSL.)

FREEING
The IPCONN is being released.

OBTAINING
The IPCONN is being acquired. The connection remains in the OBTAINING state until all the criteria for ACQUIRED have been met.

RELEASED
The IPCONN is RELEASED. Although it may also be in INSERVICE status, it is not usable.

The RELEASED status can be caused by any one of a number of general conditions:

- The remote system has not yet initialized.
- No IPCONN definition exists on the remote system and autoinstall was not active or not successful.
- The IPCONN on the remote system has been set out of service.
- AUTOCONNECT(NO) has been specified on the IPCONN definition.
- The IPCONN had been acquired but has since been released by an explicit operator command.

HOST(*data-area*)
returns the 116-character host name of the remote system (for example, `abc.example.com`), or its dotted decimal IP address (for example, `9.20.181.3`).

IPCONN(*data-value*)
specifies the 8-character identifier of the remote system or region about which you are inquiring (that is, the name assigned to its IPCONN definition).

LINKAUTH(*data-value*)
returns a CVDA value that specifies how the user ID for link security is established in a CICS system with security initialized (SEC=YES).

CERTUSER
TCP/IP communication with the partner system must be configured for SSL and a certificate must be received from the partner system during SSL handshake.

The IPCONN must refer to a TCPIPSERVICE that is defined with SSL(CLIENTAUTH).

The received certificate must be defined to the external security manager so that it is associated with a user ID, which is used to establish link security.

SECUSER
Specifies that the user ID specified in SECURITYNAME is used to establish link security.

This is the default value.

MAXQTIME(*data-area***)**
returns a fullword binary value giving the maximum time, in seconds, for which allocate requests may be queued. The value is in the range 0-9999, or will have the standard null value of -1 if MAXQTIME(NO) is specified on the IPCONN definition.

NETWORKID(*data-area***)**
returns the network ID of the remote system. The value retruned is an 8-byte character string, which is is the value of the NETWORKID option of the IPCONN definition. If NETWORKID is not specified on the IPCONN definition, the value returned is the VTAM NETID or, for VTAM=NO systems, the value of the UOWNETQL system initialization parameter, of this CICS (that is, the CICS on which the IPCONN definition is installed).

The NETWORKID is used in combination with the APPLID to ensure unique naming for connecting systems.

NUMCIPHERS(*data-area***)**
returns a binary halfword data area that contains the number of cipher suites that are specified in the CIPHERS attribute.

PENDSTATUS(*cvda***)**
shows whether there are any pending units of work for this IPCONN. The values are:

Notpending
There has been no mismatch of lognames with the partner.

Pending
There is resynchronization work outstanding for the connection but the partner system has performed an initial start, preventing completion of the resynchronization process. You can use the SET IPCONN NOTPENDING command to unilaterally commit or back out the units of work associated with the connection, according to their associated transaction definitions. You can also investigate the units of work individually and force them to commit or back out, in which case you must also complete the recovery activity by using a SET IPCONN NOTPENDING command to clear the PENDING condition.

If this is a CICS-to-CICS IPCONN, no new syncpoint work (that is, work involving sync level 2 protocols) can be transmitted across the connection until a SET IPCONN NOTPENDING command has been issued.

If you are not concerned by the loss of synchronization caused by the initial (or cold) start of the partner, you can cause the SET IPCONN NOTPENDING command to be issued automatically by specifying XLNACTION(FORCE) on the IPCONN definition.

For further information about pending units of work, see the *CICS Intercommunication Guide*.

PORT(*data-area***)**
returns a fullword binary value, in the range 1 through 65535, containing the port number to be used for outbound requests on this IPCONN; that is, the number of the port on which the remote system will be listening.

If the IPCONN is defined with PORT(NO), the value is -1.

QUEUELIMIT(*data-area***)**
returns a fullword binary value giving the maximum number of allocate requests

that can be queued for this IPCONN. The value is in the range 0-9999, or will have the standard null value of -1 if QUEUELIMIT(NO) is specified on the IPCONN definition.

RECEIVECOUNT(*data-area*)
returns a fullword binary value giving the number of RECEIVE sessions defined for this IPCONN.

RECOVSTATUS(cvda)
returns a CVDA value indicating whether there is resynchronization work outstanding for the IPCONN. The connection may never have been connected, have been quiesced and all resynchronization work completed, or disrupted without quiesce—in which case resynchronization may be necessary. CVDA values are:

NORECOVDATA
Neither side has recovery information outstanding.

NRS CICS does not have recovery outstanding for the connection, but the partner may have.

RECOVDATA
There are in-doubt units of work associated with the connection, or there are outstanding resynchronization tasks awaiting FORGET on the connection. Resynchronization takes place when the connection next becomes active, or when the UOW is unshunted.

If there is recovery outstanding, on completion of exchange lognames either resynchronization takes place or, in the case of a cold exchange, the PENDING condition is created.

SECURITYNAME(*data-area*)
returns the security name of the remote system.

In a CICS system with security initialized (SEC=YES), and for an IPCONN defined with LINKAUTH(SECUSER), the security name is used to establish the authority of the remote system.

The security name must be a valid RACF user ID on this region. Access to protected resources on this region is based on the RACF user profile and its group membership.

SENDCOUNT(*data-area*)
returns a fullword binary value giving the number of SEND sessions defined for this IPCONN.

SERVSTATUS(*cvda*)
returns a CVDA value indicating whether data can be sent and received on the IPCONN. CVDA values are:

INSERVICE
Data can be sent and received.

OUTSERVICE
Data cannot be sent or received.

SSLTYPE(*cvda*)
returns a CVDA value specifying the level of secure sockets support being used for this service. CVDA values are:

NOSSL
The Secure Sockets Layer is not being used for this service.

SSL The Secure Sockets Layer without client authentication is being used for this service.

TCPIPSERVICE(*data-area*)
returns the 8-character name of a PROTOCOL(IPIC) TCPIPSERVICE definition that defines the attributes of the inbound processing for this IPCONN.

USERAUTH(*cvda*)
returns a CVDA value that specifies the level of attach-time user security required for the connection.

LOCAL
CICS does not accept a user ID or password from clients. All requests will run under the link user ID.

IDENTIFY
Incoming attach requests must specify a user identifier.

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

DEFAULTUSER
CICS will not accept a user ID and password from the partner system. All requests run under the default user ID.

Conditions

END
RESP2 values:

2 There are no more resource definitions of this type.

ILLOGIC
RESP2 values:

1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

NOTAUTH
RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

SYSIDERR
RESP2 values:

1 The IPCONN cannot be found.

INQUIRE MVSTCB

Retrieve addresses and storage usage information for MVS TCBs.

INQUIRE MVSTCB

```
►►──INQUIRE MVSTCB──(──data-area──)─────────────────────────────────────►

►──┬─────────────────────────────────────────────────┬──────────────────►◄
   └─SET──(──ptr-ref──)──NUMELEMENTS──(──data-area──)─┘
```

Conditions: END, ILLOGIC, NOTAUTH, NOTFND

This command is threadsafe.

Description

The INQUIRE MVSTCB command can only be used in browse mode. It returns addresses and storage information for the MVS TCBs in the CICS address space. The information for each TCB shows the addresses, lengths and MVS subpools for the storage elements owned by the TCB, the storage key for each element, and the number of bytes actually in use (GETMAINed by the task) for each element.

The correct syntax for this command for all new applications is shown above. The options ELEMENTLIST, LENGTHLIST and SUBPOOLLIST in the listing of options below are obsolete, but are supported for compatibility with applications developed in releases before CICS Transaction Server for z/OS, Version 3 Release 2. You cannot use these options in combination with the SET option.

The NUMELEMENTS option has a role in both the old syntax and the new syntax. Where the ELEMENTLIST, LENGTHLIST and SUBPOOLLIST options are used, the NUMELEMENTS option specifies the number of entries in each of these lists (which is the same for each list). NUMELEMENTS is also used in combination with the SET option, to give the number of addresses in the pointer list returned by the SET option.

Browsing

This command can only be used in browse mode. Browse through all of the MVS TCBs in the CICS address space by using the browse options (START, NEXT, and END) on the command. See Browsing resource definitions for general information about browsing, including syntax, exception conditions, and examples.

Options

ELEMENTLIST(*ptr-ref***)**
returns the address of a list of the addresses of all areas of private storage allocated to this TCB. This option is obsolete, but it is supported for compatibility with applications developed in earlier CICS releases.

LENGTHLIST(*ptr-ref***)**
returns the address of a list of fullword binary lengths of the storage areas listed in the ELEMENTLIST list. This option is obsolete, but it is supported for compatibility with applications developed in earlier CICS releases.

NUMELEMENTS(*data-area***)**
A fullword binary field which is set to the number of storage elements owned by this TCB. This value is the number of addresses listed in the pointer list returned by the SET option, where each address indicates one storage element.

MVSTCB(*data-area*)

returns the address of the MVS TCB in the CICS address space. The TCB address that is returned can be used as input to the EXEC CICS COLLECT STATISTICS MVSTCB command to retrieve storage and CPU time statistics for the TCB.

SET(*ptr-ref*)

returns the address of a list of four-byte addresses. Each address points to a descriptor containing details of one storage element owned by this TCB. The number of addresses in the list is the value returned by the NUMELEMENTS option.

CICS obtains the storage for the list and descriptors. It is freed when the inquiring task ends or issues another INQUIRE MVSTCB command with one of the command options. The task cannot free the storage itself.

The format of the descriptor for each storage element is shown in Table 21:

Table 21. INQUIRE MVSTCB, SET option: Descriptor for each storage element

Offset (decimal)	Length	Contents
0	4	Address of the storage
4	4	Length
8	4	MVS subpool number
12	4	MVS storage key (for example, 8)
16	4	Number of bytes in use
Note: “Number of bytes in use” is the amount of storage actually GETMAINed by the task. This might be less than the amount of storage allocated to the TCB, because storage is always allocated to a TCB in page multiples (4096 bytes).		

SUBPOOLLIST(*ptr-ref*)

returns the address of a list of fullword binary subpool numbers of the MVS subpools for the storage areas listed in the ELEMENTLIST list. This option is obsolete, but it is supported for compatibility with applications developed in earlier CICS releases.

Conditions

END

RESP2 values:

2 All authorized resources have been retrieved. All data areas specified on this command are left unchanged.

ILLOGIC

RESP2 values:

1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 values:

1 The TCB specified on the command was not found.

PERFORM JVMPOOL

Start and terminate JVMs in the JVM pool.

PERFORM JVMPOOL

```
►►──PERFORM JVMPOOL──────────────────────────────────────────────────────────────────────►

►──┬─┬─INITIALIZE(cvda)─┬──JVMCOUNT(data-value)──JVMPROFILE(data-value)──┬─EXECKEY(cvda)─┬──┬──────►◄
   │ └─START────────────┘                                                 ├─CICSEXECKEY───┤  │
   │                                                                      └─USEREXECKEY───┘  │
   └─┬─TERMINATE(cvda)─┬──┬────────────────────────┬─────────────────────────────────────────┘
     ├─PHASEOUT────────┤  └─JVMPROFILE(data-value)─┘
     ├─PURGE───────────┤
     └─FORCEPURGE──────┘
```

Conditions: INVREQ, NOTAUTH

This command is threadsafe.

Description

You can use the PERFORM JVMPOOL command to start JVMs with your chosen JVM profile and execution key. You can also use the command to terminate all or some of the JVMs in the pool, in order to implement changes to JVM profiles, or to add new application classes.

Options

EXECKEY(*cvda*)
Specifies the execution key for the JVMs. CVDA values are:

CICSEXECKEY
The JVMs are to execute in CICS key.

USEREXECKEY
The JVMs are to execute in user key.

INITIALIZE(*cvda*)

specifies a number of JVMs to be started with the JVM profile named by the JVMPROFILE option, and in the execution key named by the EXECKEY option. CICS starts the JVMs asynchronously, so you might not receive notifications of all failures.

The only permitted CVDA value is **START**.

You cannot start up JVMs when the JVM pool's status is set to disabled.

JVMCOUNT(*data-value*)
Specifies, as a fullword binary value, the number of JVMs to be started. If the number of JVMs you request, added to the number of JVMs that already exist, would mean exceeding the MAXJVMTCBS limit for the CICS region, CICS returns an error and does not start any JVMs.

JVMPROFILE(*data-value*)

Specifies the 8-character name of the JVM profile to be used for the JVMs, padded with trailing spaces if necessary. The name is case-sensitive, and you must enter it using the same combination of upper and lower case characters that is present in the z/OS UNIX file name.

This option is required with the INITIALIZE option. It is optional with the TERMINATE option; if it is not specified, all the JVMs in the pool are terminated.

TERMINATE(*cvda*)

specifies that all or some of the JVMs in the JVM pool are to be terminated. The new JVMs that are started to handle incoming requests incorporate any changes that you made to their JVM profiles, or new application classes that you added. If the CICS region does not have a shared class cache, the new JVMs also incorporate any changes made to classes on the shareable application class path. You can start new JVMs manually using the PERFORM JVMPOOL STARTUP command, or let CICS start them automatically.

The PERFORM JVMPOOL TERMINATE command does not terminate the shared class cache and the master JVM. If the CICS region has a shared class cache, and you want to terminate the shared class cache in order to update classes on the shareable application class path, use the PERFORM CLASSCACHE command to do this.The CVDA values for the TERMINATE option are:

PHASEOUT

JVMs with the selected profile, or all JVMs in the pool, are marked for deletion. The JVMs are actually deleted when they finish running their current Java program.

PURGE

All tasks using JVMs with the selected profile, or using all JVMs in the pool, are terminated using the SET TASK PURGE mechanism, and the JVMs are terminated.

FORCEPURGE

All tasks using JVMs with the selected profile, or using all JVMs in the pool, are terminated by the SET TASK FORCEPURGE mechanism, and the JVMs are terminated.

Conditions

INVREQ

RESP2 values:

1 TERMINATE is specified with an invalid CVDA value.

2 INITIALIZE is specified with an invalid CVDA value.

3 A task purge failed on a TERMINATE request.

4 EXECKEY is specified with an invalid CVDA value.

5 JVMCOUNT is out of range (1-999).

6 The number of JVMs specified for JVMCOUNT would cause the total number of JVMs in the CICS region to exceed the MAXJVMTCBS limit.

7 INITIALIZE is specified when the status of the JVM pool is DISABLED.

10 INITIALIZE is specified without JVMPROFILE.

26 The JVM profile name specified by JVMPROFILE contains invalid characters or embedded blanks.

NOTAUTH
RESP2 value:

100 The user associated with the issuing task is not authorized to use this command.

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.

SET DOCTEMPLATE

```
►►──SET DOCTEMPLATE(data-value)──COPY(NEWCOPY)──────────────────────────►◄
```

Conditions: INVREQ, NOTFND, NOTAUTH

This command is threadsafe.

Description

The SET DOCTEMPLATE command operates on the specified CICS document template. The COPY(NEWCOPY) option is the only option available on this command.

For document templates in a partitioned data set, CICS file, z/OSUNIX System Services HFS file, temporary storage queue, or transient data queue, the command deletes the copy of the document template which is currently cached by CICS, and replaces it with a new copy. (For templates in a partitioned data set, CICS first performs a BLDL (build list) to obtain the most current directory information, and then rereads the member.)

For document templates that reside in CICS programs (with PROGRAM specified in the DOCTEMPLATE resource definition), the command refreshes the program. It is equivalent to SET PROGRAM PHASEIN for the specified program. Document templates retrieved from programs are not cached by CICS.

For document templates generated by exit programs (with EXITPGM specified in the DOCTEMPLATE resource definition), the command refreshes the exit program. It is equivalent to SET PROGRAM PHASEIN for the specified exit program. When you issue the command, CICS deletes any cached copy of the document template, phases in the new copy of the program, and creates a new cached copy of the document template if the exit program specifies caching. The refreshed exit program can specify a different setting for whether or not caching should take place, and CICS honors the change.

Options

COPY(*cvda*)
refreshes the document template. The CVDA value is:

NEWCOPY
If a cached copy of the document template exists, it is to be deleted. If the document template resides in a CICS program or exit program, a new copy of the program is to be phased in. If caching is required for the document template, a new copy of the document template is to be loaded into the cache.

DOCTEMPLATE(*data-value*)
specifies the 1 to 8-character name of the DOCTEMPLATE resource definition which defines the document template.

Conditions

INVREQ
RESP2 values:

2 COPY is specified with an invalid CVDA value.

4 The new copy of the document template could not be loaded into the cache.

NOTFND
RESP2 values:

1 The DOCTEMPLATE resource definition was not found.

3 The member of the partitioned data set specified by the DOCTEMPLATE resource definition was not found.

5 The resource specified by the DOCTEMPLATE resource definition was not found.

NOTAUTH
RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

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

INQUIRE MVSTCB

Retrieve addresses and storage usage information for MVS TCBs.

INQUIRE MVSTCB

```
►►──INQUIRE MVSTCB──(──data-area──)──────────────────────────────────►

►──┬───────────────────────────────────────────────────┬─────────────►◄
   └─SET──(──ptr-ref──)──NUMELEMENTS──(──data-area──)──┘
```

Conditions: END, ILLOGIC, NOTAUTH, NOTFND

This command is threadsafe.

Description

The INQUIRE MVSTCB command can only be used in browse mode. It returns addresses and storage information for the MVS TCBs in the CICS address space. The information for each TCB shows the addresses, lengths and MVS subpools for the storage elements owned by the TCB, the storage key for each element, and the number of bytes actually in use (GETMAINed by the task) for each element.

The correct syntax for this command for all new applications is shown above. The options ELEMENTLIST, LENGTHLIST and SUBPOOLLIST in the listing of options below are obsolete, but are supported for compatibility with applications developed in releases before CICS Transaction Server for z/OS, Version 3 Release 2. You cannot use these options in combination with the SET option.

The NUMELEMENTS option has a role in both the old syntax and the new syntax. Where the ELEMENTLIST, LENGTHLIST and SUBPOOLLIST options are used, the NUMELEMENTS option specifies the number of entries in each of these lists (which is the same for each list). NUMELEMENTS is also used in combination with the SET option, to give the number of addresses in the pointer list returned by the SET option.

Browsing

This command can only be used in browse mode. Browse through all of the MVS TCBs in the CICS address space by using the browse options (START, NEXT, and END) on the command. See Browsing resource definitions for general information about browsing, including syntax, exception conditions, and examples.

Options

ELEMENTLIST(*ptr-ref***)**
returns the address of a list of the addresses of all areas of private storage allocated to this TCB. This option is obsolete, but it is supported for compatibility with applications developed in earlier CICS releases.

LENGTHLIST(*ptr-ref***)**
returns the address of a list of fullword binary lengths of the storage areas listed in the ELEMENTLIST list. This option is obsolete, but it is supported for compatibility with applications developed in earlier CICS releases.

NUMELEMENTS(*data-area***)**
A fullword binary field which is set to the number of storage elements owned by this TCB. This value is the number of addresses listed in the pointer list returned by the SET option, where each address indicates one storage element.

MVSTCB(*data-area***)**
returns the address of the MVS TCB in the CICS address space. The TCB address that is returned can be used as input to the EXEC CICS COLLECT STATISTICS MVSTCB command to retrieve storage and CPU time statistics for the TCB.

SET(*ptr-ref***)**
returns the address of a list of four-byte addresses. Each address points to a descriptor containing details of one storage element owned by this TCB. The number of addresses in the list is the value returned by the NUMELEMENTS option.

CICS obtains the storage for the list and descriptors. It is freed when the inquiring task ends or issues another INQUIRE MVSTCB command with one of the command options. The task cannot free the storage itself.

The format of the descriptor for each storage element is shown in Table 21 on page 311:

Table 22. INQUIRE MVSTCB, SET option: Descriptor for each storage element

Offset (decimal)	Length	Contents
0	4	Address of the storage
4	4	Length
8	4	MVS subpool number
12	4	MVS storage key (for example, 8)
16	4	Number of bytes in use
Note: "Number of bytes in use" is the amount of storage actually GETMAINed by the task. This might be less than the amount of storage allocated to the TCB, because storage is always allocated to a TCB in page multiples (4096 bytes).		

SUBPOOLLIST(*ptr-ref***)**
returns the address of a list of fullword binary subpool numbers of the MVS subpools for the storage areas listed in the ELEMENTLIST list. This option is obsolete, but it is supported for compatibility with applications developed in earlier CICS releases.

Conditions

END
RESP2 values:

2 All authorized resources have been retrieved. All data areas specified on this command are left unchanged.

ILLOGIC
RESP2 values:

1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

NOTAUTH
RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

NOTFND
RESP2 values:

1 The TCB specified on the command was not found.

SET IPCONN

Change the attributes of an IPCONN or cancel outstanding AIDs.

SET IPCONN

```
►►──SET IPCONN(data-value)──┬──────────────────┬──┬─────────────────┬──────────►
                            ├─CONNSTATUS(cvda)─┤  ├─PENDSTATUS(cvda)─┤
                            ├─ACQUIRED─────────┤  └─NOTPENDING──────┘
                            └─RELEASED─────────┘

►──┬─────────────────┬──┬──────────────────┬──┬─────────────────┬──────────►
   ├─PURGETYPE(cvda)─┤  ├─RECOVSTATUS(cvda)─┤  ├─SERVSTATUS(cvda)─┤
   ├─CANCEL──────────┤  └─NORECOVDATA──────┘  ├─INSERVICE───────┤
   ├─FORCECANCEL─────┤                        └─OUTSERVICE──────┘
   ├─FORCEPURGE──────┤
   ├─KILL────────────┤
   └─PURGE───────────┘

►──┬─────────────────┬──────────────────────────────────────────────────►◄
   ├─UOWACTION(cvda)─┤
   ├─BACKOUT─────────┤
   ├─COMMIT──────────┤
   ├─FORCEUOW────────┤
   └─RESYNC──────────┘
```

Conditions: INVREQ, IOERR, NORMAL, NOTAUTH, SYSIDERR

For more information about the use of CVDAs, see CICS-value data areas (CVDAs).

This command is threadsafe.

Description

The SET IPCONN command allows you to change some of the attributes that define an IPCONN. Control returns to the issuing program when the required operation has been scheduled. To get the operation started, it is necessary to relinquish control to CICS.

Note: SET IPCONN is used to change the attributes of IPIC connections (also known as "*IPCONNs*"). See also SET CONNECTION. The SET CONNECTION command is used to change the attributes of MRO and ISC over SNA connections.

For information about the different kinds of intercommunication connections, see the *CICS Intercommunication Guide*.

Options

CONNSTATUS(*cvda*)

specifies whether to acquire or release sessions with the system represented by the IPCONN name. An IPCONN cannot be both ACQUIRED and OUTSERVICE.

CVDA values are:

ACQUIRED

Sessions are to be acquired.

RELEASED

Sessions are to be released.

For further information about managing IPCONNs, see the *CICS Intercommunication Guide*.

IPCONN(*data-value***)**

specifies, as an 8-character field, the name of the IPCONN to be modified. This is the name of the remote system or region specified on the IPCONN option of the IPCONN definition.

PENDSTATUS(*cvda***)**

specifies, for an IPCONN to a CICS Transaction Server for z/OS partner that has performed an initial start, that the normal resynchronization process is to be overridden:

The CVDA value is:

NOTPENDING

Forces all in-doubt units of work (that were created by the IPCONN before the initial start of the partner) to either commit or back out, as specified by the ACTION option of the TRANSACTION definition. It also forgets any resyncs (waitforget UOW-links) that are outstanding for the connection, and created before the initial start of the partner.

The PENDING condition indicates the existence of recovery information (either shunted UOWs or decisions remembered for the partner) on a connection that has experienced a lognames mismatch with its partner. This indicates that the partner has performed an initial start and that the recovery protocol has been corrupted by a loss of log data at the partner.

It is not possible to set a connection to NOTPENDING state (forcing in-doubt and erasing NOFORGET UOWs) until CICS has made contact with the partner and received a new logname from it.

Decisions for a whole connection can be forgotten, but that does not affect the memory of a decision for any other connection involved in the UOW.

Note: SET IPCONN NOTPENDING, SET IPCONN NORECOVDATA, and SET IPCONN UOWACTION are mutually exclusive. For advice on which command to use, see the notes following the description of the UOWACTION option.

The exchange lognames function and the resynchronization function are described in the *CICS Intercommunication Guide*.

PURGETYPE(*cvda***)**

specifies how associated transactions are to be purged. CVDA values are:

CANCEL

specifies that queued requests by transactions to use this IPCONN are to be canceled.

Queued requests to use this IPCONN by CICS system transactions that manage communications across this IPCONN are not purged unless FORCECANCEL is specified.

Message DFHISnnnn is written to CSMT to indicate how many queued requests to use this IPCONN have been deleted and how many remain.

A “QUEUED REQUESTS CANCELED” message appears on the CEMT panel whenever queued requests to use this IPCONN are deleted using the CANCEL option of the CEMT SET IPCONN command.

FORCECANCEL

specifies that all queued requests by transactions to use this IPCONN are to be canceled, including requests by CICS system transactions that manage communications across this IPCONN. This can lead to unpredictable results and should be used only in exceptional circumstances.

A “QUEUED REQUESTS CANCELED” message appears on the CEMT panel whenever queued requests to use this IPCONN are deleted using the FORCECANCEL option of the CEMT SET IPCONN command.

FORCEPURGE

specifies that all transactions running on sessions to the connected system are to be abnormally terminated immediately. This can lead to unpredictable results and should be used only in exceptional circumstances.

In some extreme cases (for example, if an error occurs during backout processing), CICS might terminate abnormally.

KILL specifies that the task is to be terminated. System and data integrity is not guaranteed. The KILL option extends the PURGE and FORCEPURGE options. You should use it only after an attempt has been made to PURGE or FORCEPURGE a task. The KILL option does not guarantee integrity of any kind but in some situations it allows you to free up a stalled region, enabling the region to continue processing. In some cases, for example if a task is killed during backout processing, CICS terminates abnormally.

PURGE

specifies that transactions running on the connected system are to be abnormally terminated. Transactions are terminated only if system and data integrity can be maintained. A transaction is not purged if its definition specifies SPURGE=NO.

RECOVSTATUS(*cvda*)

specifies that the normal resynchronization process is to be overridden. The CVDA value is:

NORECOVDATA

Forces all in-doubt units of work (according to the transaction definitions), targets any resyncs that were outstanding for the IPCONN, and erases the logname previously received from the partner system. The state of the connection is reset.

Attention: You should use SET IPCONN NORECOVDATA only in exceptional circumstances. It erases recovery information and may compromise data integrity for units of work that have updated resources on remote systems.

Examples of circumstances in which you might need to use it are:

- You need to discard an IPCONN, and it is not possible for the quiesce protocols with the partner system to be completed.
- An operational or logic error results in a logname mismatch for the connection. The connection state must be reset to allow the exchange lognames process to complete.

Note: SET IPCONN NORECOVDATA, SET IPCONN NOTPENDING, and SET IPCONN UOWACTION are mutually exclusive.

SERVSTATUS(*cvda*)

specifies whether the IPCONN is to be placed in service or out of service. CVDA values are:

INSERVICE

The IPCONN is to be placed in service. This allows it to be acquired.

OUTSERVICE

The IPCONN is to be placed out of service; that is, not available for use.

The following occurs:

- If the connection is currently ACQUIRED and you specify OUTSERVICE, the command fails with INVREQ and a RESP2 of 2. You must RELEASE the connection before setting OUTSERVICE.
- If the connection is currently RELEASED, the status of the connection is set OUTSERVICE and it cannot be used until it is INSERVICE again.

UOWACTION(*cvda*)

specifies that the normal resynchronization process is to be partially overridden: decisions are taken for any units of work that are in-doubt because of a failure of the IPCONN; but the decisions are recorded and any data inconsistencies are reported when the connection is next acquired.

The operation is synchronous with setting the state of the UOW; that is, an INQUIRE UOW following a SET IPCONN UOWACTION returns the new UOW states. CVDA values are:

BACKOUT

All UOWs shunted because of the failure of this IPCONN are to be backed out.

COMMIT

All UOWs shunted because of the failure of this IPCONN are to be committed.

FORCE

All UOWs shunted because of the failure of this IPCONN are to be forced to BACKOUT or COMMIT, as specified on the ACTION option of the TRANSACTION definition.

RESYNC

Any UOWs shunted because of the failure of this IPCONN are to be retried (that is, exchange lognames resynchronization for this connection is to be attempted). This process should normally be started automatically when a connection is acquired or when a UOW is unshunted.

Notes:

1. SET IPCONN UOWACTION unshunts all units of work that have failed in-doubt because of a failure of the IPCONN. Before issuing SET IPCONN FORCE, you may want to use the SET UOW command to specify commit or backout for each in-doubt unit of work explicitly, rather than letting it default. Local procedures will determine the importance of the data and the method of using the

INQUIRE UOW, INQUIRE UOWENQ, and INQUIRE UOWLINK commands to establish the correct actions.

2. As far as shunted units of work are concerned, you may use only one of SET IPCONN UOWACTION, SET IPCONN NOTPENDING, and SET IPCONN NORECOVDATA. SET IPCONN NORECOVDATA should be used only in exceptional circumstances.
3. To force all in-doubt units of work caused by a failure of the IPCONN in the same direction, use SET IPCONN COMMIT or SET IPCONN BACKOUT.
4. Neither SET IPCONN UOWACTION nor the SET UOW UOWACTION command clears resync information. If you want to do this, you must use SET IPCONN NOTPENDING or SET IPCONN NORECOVDATA.
5. You can issue SET UOW UOWACTION commands *before* issuing SET IPCONN NOTPENDING or SET IPCONN NORECOVDATA.

Conditions

INVREQ

RESP2 values:

2 ACQUIRED and OUTSERVICE are specified inconsistently in any of the following ways:
1. ACQUIRED specified with OUTSERVICE
2. ACQUIRED specified for OUTSERVICE IPCONN
3. RELEASED and OUTSERVICE specified in the same command for an ACQUIRED IPCONN.

3 CONNSTATUS has an invalid CVDA value.

4 SERVSTATUS has an invalid CVDA value.

7 PURGETYPE has an invalid CVDA value.

8 PENDSTATUS has an invalid CVDA value.

18 NOTPENDING cannot be set for an IPCONN that has successfully completed exchange lognames processing.

19 CONNSTATUS cannot be set to ACQUIRED when in the FREEING state.

20 An attempt was made to acquire a one-way IPCONN.

21 BACKOUT or FORCE was specified, but was unsuccessful. Some UOWs remain shunted for this IPCONN.

22 Other SET parameters were included with the CANCEL or FORCECANCEL option.

25 IPCONN is still in service.

26 RECOVSTATUS does not have a value of NORECOVDATA.

27 The CVDA value specified on the UOWACTION option is invalid.

45 NORECOVDATA cannot be set for an IPCONN that is in service.

IOERR

RESP2 values:

10 Unexpected error.

NORMAL

RESP2 values:

58 AIDs have been successfully canceled.

59 No AIDs have been canceled.

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

SYSIDERR
RESP2 values:

9 The named IPCONN could not be found.

Appendix E. New CEMT commands

CICS Transaction Server for z/OS, Version 3 Release 2 provides new CEMT commands.

CEMT 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.

Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET DOCTEMPLATE` followed by one or more DOCTEMPLATE resource definition names or ALL. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE DOCTEMPLATE.
- Type `CEMT SET DOCTEMPLATE`, followed by one or more DOCTEMPLATE resource definition names or ALL, followed by NEWCOPY.

Typing `?` at the beginning of either the first or second line gives a syntax prompt.

CEMT SET DOCTEMPLATE

```
►►──CEMT Set DOctemplate──Newcopy──────────────────────────────────────────►◄
```

Options

ALl
The action you request is performed for all resources of the specified type that you are authorized to access.

(*value***)**
specifies the 1–8 character name of the DOCTEMPLATE resource definition.

Newcopy
specifies that if a cached copy of the document template exists, it is to be deleted. If the document template resides in a CICS program or exit program, a new copy of the program is to be phased in. If caching is required for the document template, a new copy of the document template is to be loaded into the cache.

CEMT INQUIRE IPCONN

Function

Retrieve information about IPCONNs.

Description

CEMT INQUIRE IPCONN returns information about the status of IPCONNs to a remote system or to another CICS region.

An IPCONN (or IPIC connection) is a TCP/IP communication link from your local CICS region to another CICS region, or another system.

You can reset the options either by typing the SET command or by overtyping at the appropriate place on the INQUIRE screen.

If you want to install a new IPCONN definition when one is already installed, you must set the connection OUTSERVICE before using the CEDA INSTALL command for your new definition.

Note: CEMT INQUIRE IPCONN returns information about IPIC connections. See also CEMT INQUIRE CONNECTION. The CEMT INQUIRE CONNECTION command returns information about MRO and ISC over SNA connections.

For information about the different kinds of intercommunication connections, see the *CICS Intercommunication Guide*.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

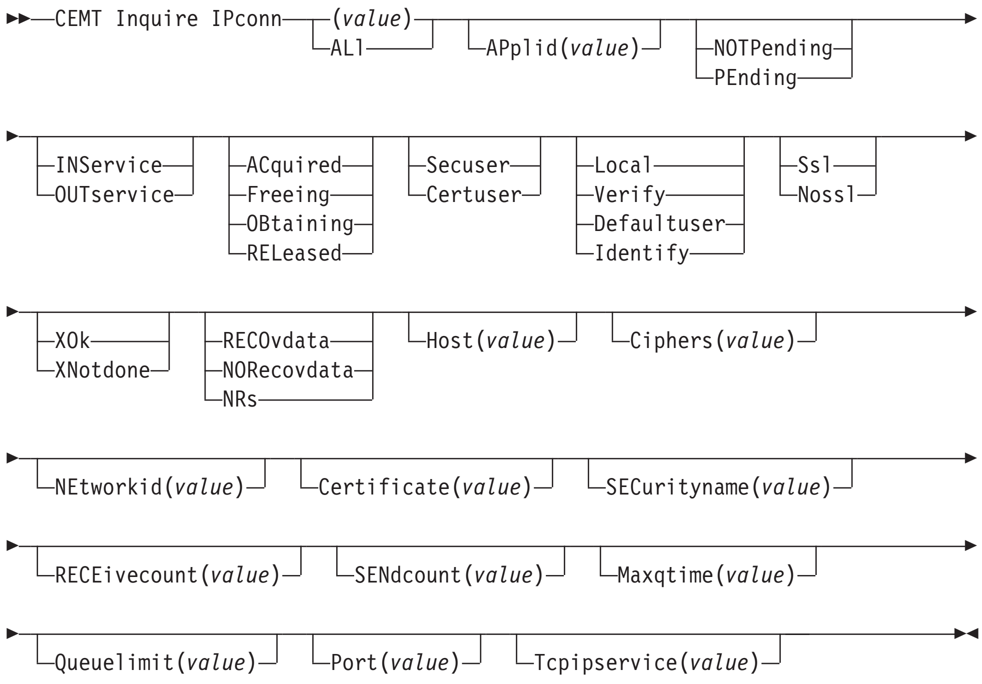
- Type `CEMT INQUIRE IPCONN` (the minimum abbreviation is `CEMT I IP`). You get a display that lists the current status.
- Type `CEMT INQUIRE IPCONN` (`CEMT I IP`) followed by as many of the other attributes as are necessary to limit the range of information that you require.

You can then tab to the highlighted or blank fields and overtype them with the required values.

(*value***)**
specifies one or more names (1–8 characters) defined for an IPCONN.

All
is the default.

CEMT INQUIRE IPCONN



Displayed fields

Applid(*value*)
displays the name by which the remote system is known to the network (taken from the APPLID option of the IPCONN definition). This is the application identifier (*applid*) of the remote system, as specified on the APPLID option of its system initialization table. For XRF systems it is the generic applid.

Connstatus
indicates the state of the IPCONN between CICS and the remote system. The values are:

Acquired
The IPCONN is acquired. The criteria for ACQUIRED are:

- The capabilities exchange is complete.

Freeing
The IPCONN is being released.

Obtaining
The IPCONN is being acquired. The IPCONN remains in the OBTAINING state until all the criteria for ACQUIRED have been met.

Released
The IPCONN is RELEASED. Although it may also be in INSERVICE status, it is not usable.

The RELEASED status can be caused by any one of a number of general conditions:

- The remote system has not yet initialized.
- No IPCONN definition exists on the remote system and autoinstall was not active or not successful.
- The IPCONN on the remote system has been set out of service.
- AUTOCONNECT(NO) has been specified on the IPCONN definition.

- The IPCONN had been acquired but has since been released by an explicit operator command.

Certificate(*value*)
displays a 32-character area containing the label of the certificate, within the key ring, that is used as a client certificate in the SSL handshake for outbound IPCONN connections. If the label is blank, the certificate nominated as the default for the key ring is used.

Ciphers(*value*)
displays a 56-character area containing the list of cipher suites that is used to negotiate with clients during the SSL handshake. The list is set by the ENCRYPTION system initialization parameter, but you can edit the list to remove or change the order of cipher suites. See the *CICS RACF Security Guide*.

Host(*value*)
displays the host name of the remote system (for example, `abc.example.com`), or its dotted decimal IP address (for example, `9.20.181.3`).

Linkauth
displays a CVDA value that specifies how the user ID for link security is established in a CICS system with security initialized (SEC=YES).

CERTUSER
TCP/IP communication with the partner system must be configured for SSL and a certificate must be received from the partner system during SSL handshake.

The IPCONN must refer to a TCPIPSERVICE that is defined with SSL(CLIENTAUTH).

The received certificate must be defined to the external security manager so that it is associated with a user ID, which is used to establish link security.

SECUSER
Specifies that the user ID specified in SECURITYNAME is used to establish link security.

Maxqtime(*value*)
displays the maximum time, in seconds, for which allocate requests may be queued. The value is in the range 0-9999, or will have the standard null value of -1 if MAXQTIME(NO) is specified on the IPCONN definition.

Networkid
displays the network ID of the remote system. The value retruned is an 8-byte character string, which is is the value of the NETWORKID option of the IPCONN definition. If NETWORKID is not specified on the IPCONN definition, the value returned is the VTAM NETID or, for VTAM=NO systems, the value of the UOWNETQL system initialization parameter, of this CICS (that is, the CICS on which the IPCONN definition is installed).

The `Networkid` value is used in combination with the `Applid` value to ensure unique naming for connecting systems.

Pendstatus
displays whether there are any pending units of work for this IPCONN. The values are:

Notpending
There has been no mismatch of lognames with the partner.

Pending

There is resynchronization work outstanding for the connection but the partner system has performed an initial start, preventing completion of the resynchronization process. You can use the SET IPCONN NOTPENDING command to unilaterally commit or back out the units of work associated with the connection, according to their associated transaction definitions. You can also investigate the units of work individually and force them to commit or back out, in which case you must also complete the recovery activity by using a SET IPCONN NOTPENDING command to clear the PENDING condition.

If this is a CICS-to-CICS IPCONN, no new syncpoint work (that is, work involving sync level 2 protocols) can be transmitted across the connection until a SET IPCONN NOTPENDING command has been issued.

If you are not concerned by the loss of synchronization caused by the initial (or cold) start of the partner, you can cause the SET IPCONN NOTPENDING command to be issued automatically by specifying XLNACTION(FORCE) on the IPCONN definition.

For further information about pending units of work, see the *CICS Intercommunication Guide*.

Port(*value***)**

displays the port number to be used for outbound requests on this connection; that is, the number of the port on which the remote system will be listening. This may be set to NOTAPPLIC if the connection is never used for outbound traffic, as is usually the case for autoinstalled IPCONNs.

Queuelimit(*value***)**

displays the maximum number of allocate requests that can be queued for this connection. The value is in the range 0-9999, or will have the standard null value of -1 if QUEUELIMIT(NO) is specified on the IPCONN definition.

Receivecount(*value***)**

displays the number of RECEIVE sessions defined for this connection.

Recovstatus

indicates whether there is resynchronization work outstanding for the IPCONN. The IPCONN may never have been connected, have been quiesced and all resynchronization work completed, or disrupted without quiesce: in which case resynchronization may be necessary. The values are:

Norecovdata

Neither side has recovery information outstanding.

Nrs CICS does not have recovery outstanding for the IPCONN, but the partner may have.

Recovdata

There are in-doubt units of work associated with the IPCONN, or there are outstanding resynchronization tasks awaiting FORGET on the connection. Resynchronization takes place when the IPCONN next becomes active, or when the UOW is unshunted.

If there is recovery outstanding, on completion of exchange of lognames either resynchronization takes place or, in the case of a cold exchange, the PENDING condition is created.

Securityname(*value***)**

displays the security name of the remote system.

In a CICS system with security initialized (SEC=YES), the security name is used to establish the authority of the remote system.

The security name must be a valid RACF user ID on this region. Access to protected resources on this region is based on the RACF user profile and its group membership.

Sendcount(*value*)
displays the number of SEND sessions defined for this IPCONN.

Servstatus
indicates whether data can be sent and received on the IPCONN. The values are:

Inservice
Data can be sent and received.

Outservice
Data cannot be sent or received.

Ssltype
displays a CVDA value specifying the level of secure sockets support being used for this service. CVDA values are:

NOSSL
The Secure Sockets Layer is not being used for this service.

SSL The Secure Sockets Layer is being used for this service.

Tcpipservice(*value*)
displays the name of a PROTOCOL(IPIC) TCPIPSERVICE definition that defines the attributes of the inbound processing for this IPCONN.

Userauth
displays a CVDA value that specifies how the user ID for attach-time user security is established in a CICS system with security initialized (SEC=YES)

DEFAULTUSER
CICS does not accept a user ID and password from the partner system. All requests run under the default user ID.

IDENTIFY
Incoming attach requests must specify a user identifier but not a password.

LOCAL
CICS does not require a user ID or password from clients. All requests will run under the link user ID.

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

CEMT PERFORM JVMPOOL

Start and terminate JVMs in the JVM pool.

Description

You can use the CEMT PERFORM JVMPOOL command to start JVMs with your chosen JVM profile and execution key. You can also use the command to terminate all or some of the JVMs in the pool, in order to implement changes to JVM profiles, or to add new application classes. If the CICS region does not have a shared class cache, terminating the JVMs also implements any changes made to classes on the shareable application class path.

The CEMT PERFORM JVMPOOL TERMINATE command does not terminate the shared class cache and the master JVM. If the CICS region has a shared class cache, and you want to terminate the shared class cache in order to update classes on the shareable application class path, use the CEMT PERFORM CLASSCACHE command to do this.

Syntax

CEMT PERFORM JVMPOOL

```
►►──CEMT Perform Jvmpool──────────────────────────────────────────────────►

►──┬─Start──JVMCount(number)──JVMProfile(name)──┬─Cexeckey─┬─┬───────────►◄
   │                                            └─Uexeckey─┘ │
   └──┬─PHaseout───┬──┬───────────────────┬─────────────────┘
      ├─PUrge──────┤  └─JVMProfile(name)──┘
      └─Forcepurge─┘
```

Options

Cexeckey
Specifies that the JVMs to be started are to run in CICS key.

Forcepurge
All tasks using JVMs with the profile specified by the Jvmprofile option, or (if Jvmprofile is not specified) using all JVMs in the pool, are terminated by the SET TASK FORCEPURGE mechanism, and the JVMs are terminated.

Jvmcount(*number*)
Specifies the number of JVMs to be started. If the number of JVMs you request, added to the number of JVMs that already exist, would mean exceeding the MAXJVMTCBS limit for the CICS region, CICS does not start any JVMs.

Jvmprofile(*name*)
Specifies the 8-character name of a JVM profile. The name is case-sensitive, and you must enter it using the same combination of upper and lower case characters that is present in the z/OS UNIX file name. If you need to enter the name of a JVM profile in mixed case when you are using the CEMT transaction, ensure that the terminal you use is correctly configured, with upper case translation suppressed.

Jvmprofile is required with Start, to specify the profile for the JVMs that are to be started.

Jvmprofile can optionally be specified with Phaseout, Purge, or Forcepurge, to mean that only JVMs with this profile should be terminated. Without the Jvmprofile option, these options act on all the JVMs in the pool.

Phaseout
JVMs with the profile specified by the Jvmprofile option, or (if Jvmprofile is not specified) all JVMs in the pool, are marked for deletion. The JVMs are actually deleted when they finish running their current Java program.

Purge
All tasks using JVMs with the profile specified by the Jvmprofile option, or (if Jvmprofile is not specified) using all JVMs in the pool, are terminated by the SET TASK PURGE mechanism, and the JVMs are terminated.

Start(*number*)
Specifies a number of JVMs to be started. Jvmcount, Jvmprofile and either Cexeckey or Uexeckey are required with Start, to specify the number, profile and execution key for the JVMs. CICS starts the JVMs asynchronously.

You cannot specify Start when the JVM pool's status is set to Disabled.

Uexeckey
Specifies that the JVMs to be started are to run in user key. If the system initialization parameter STGPROT=NO is in effect, this is ignored, and the JVMs are started in CICS key.

CEMT SET IPCONN

Change the attributes of an IPCONN or cancel outstanding AIDs.

Description

SET IPCONN allows you to change some of the attributes of an IPCONN.

Note: See also CEMT SET CONNECTION. The CEMT SET CONNECTION command is used to change the attributes of MRO and ISC over SNA connections.

For information about the different kinds of intercommunication connections, see the *CICS Intercommunication Guide*.

If you want to install a new IPCONN definition when one is already installed, you must set the connection OUTSERVICE before using the CEDA INSTALL commands for your new definition. See the *CICS Resource Definition Guide* for further information about connections.

Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET IPCONN` (the minimum abbreviation is `CEMT S IP`) followed by one or more connection identifiers or ALL. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE IPCONN. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET IPCONN` (`CEMT S IP`) followed by one or more connection identifiers or ALL, followed in turn by one or more attribute settings that you wish to change. For example, `cemt s ip al i` resets the values for all connections to make them available for use (inservice).

Typing `?` at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET IPCONN

```
►►──CEMT Set IPconn──┬─(value)─┬──┬──────────┬──┬─────────────┬──────────►
                     └─ALl─────┘  ├─ACquired─┤  └─NOTpending──┘
                                  └─RELeased─┘

►──┬──────────────┬──┬──────────────┬──┬────────────┬──┬──────────┬──────────►◄
   ├─CAncel───────┤  └─NORecovdata──┘  ├─Inservice──┤  ├─Backout──┤
   ├─FORCECancel──┤                    └─Outservice─┘  ├─COmmit───┤
   ├─FORCEPurge───┤                                    ├─FORCEUow─┤
   ├─Kill─────────┤                                    └─RESync───┘
   └─Purge────────┘
```

Options

(*value*)
specifies the name (1–8 characters) of the connection to be modified. This is the name of the remote system or region specified on the IPCONN option of the IPCONN definition. You can specify more than one name.

ACquired
specifies that CICS is to acquire a session with the system represented by the IPCONN name. A connection cannot be both ACQUIRED and OUTSERVICE.

ALl
specifies that any changes you request are to be made to all resources of the specified type that you are authorized to access.

Backout
specifies that all UOWs shunted due to the failure of this connection are to be backed out. The normal resynchronization process is partially overridden: decisions are taken for any units of work that are in-doubt due to a failure of the connection; but the decisions are recorded and any data inconsistencies are reported when the connection is next acquired.

CAncel

specifies that queued requests by transactions to use this IPCONN are to be canceled.

Queued requests to use this IPCONN by CICS system transactions that manage communications across the IPCONN are not purged unless FORCECANCEL is specified.

Message DFHISnnnn is written to CSMT to indicate how many queued requests to use this IPCONN have been deleted for the IPCONN and how many remain.

A "QUEUED REQUESTS CANCELED" message appears on the CEMT panel whenever queued requests to use this IPCONN are deleted using the CANCEL option of the CEMT SET IPCONN command.

COmmit
specifies that all UOWs shunted due to the failure of this connection are to be committed. The normal resynchronization process is partially overridden: decisions are taken for any units of work that are in-doubt due to a failure of the connection; but the decisions are recorded and any data inconsistencies are reported when the connection is next acquired.

FORCECancel

specifies that all queued requests by transactions to use this IPCONN are to be canceled, including requests by CICS system transactions that manage communications across this IPCONN. This can lead to unpredictable results and should be used only in exceptional circumstances.

A “QUEUED REQUESTS CANCELED” message appears on the CEMT panel whenever queued requests to use this IPCONN are deleted using the FORCECANCEL option of the CEMT SET IPCONN command.

FORCEPurge

specifies that all transactions running on sessions to the connected system are to be abnormally terminated immediately. This can lead to unpredictable results and should be used only in exceptional circumstances.

In some extreme cases (for example, if an error occurs during backout processing), CICS might terminate abnormally.

FORCEUow

specifies that all UOWs shunted due to the failure of this connection are to be forced to back out or commit, as specified in the ACTION option of the TRANSACTION definition. The normal resynchronization process is partially overridden: decisions are taken for any units of work that are in-doubt due to a failure of the connection; but the decisions are recorded and any data inconsistencies are reported when the connection is next acquired.

Inservice

specifies that the system is to be placed in service; that is, available for use.

Kill

specifies that the task is to be terminated. System and data integrity is not guaranteed. The KILL option extends the PURGE and FORCEPURGE options. You should use it only after an attempt has been made to PURGE or FORCEPURGE a task. The KILL option does not guarantee integrity of any kind but in some situations it allows you to free up a stalled region, enabling the region to continue processing. In some cases, for example if a task is killed during backout processing, CICS terminates abnormally.

NORecovdata

specifies that the normal resynchronization process is to be overridden. NORECOVDATA forces in-doubt units of work (according to the transaction definitions), targets any resynchronization tasks that are outstanding for the connection, and erases the logname previously received from the partner system. The state of the connection is reset.

Note: You should use SET IPCONN NORECOVDATA only in exceptional circumstances. It erases recovery information and may compromise data integrity for units of work that have updated resources on remote systems.

Examples of circumstances in which you might need to use it are:

- You need to discard a connection and it is not possible for the quiesce protocols with the partner system to be completed.
- An operational or logic error results in a logname mismatch for the connection. The connection state must be reset to allow the exchange lognames process to complete.

Note: If you specify NORECOVDATA you cannot specify COMMIT, BACKOUT, FORCEUOW, RESYNC, or NOTPENDING.

NOTpending

For a connection to a CICS Transaction Server for z/OS partner that has performed an initial start, specifies that the normal resynchronization process is to be overridden.

NOTPENDING forces all in-doubt units of work (according to the transaction definition) that were created by the connection before the initial start of the partner. It also forgets any resynchronization tasks (waitforget UOW-links) that are outstanding for the connection, and created before the initial start of the partner.

The PENDING condition indicates the existence of recovery information (either shunted UOWs or decisions remembered for the partner) on a connection that has experienced a lognames mismatch with its partner. This indicates that the partner has performed an initial start and that the recovery protocol has been corrupted by a loss of log data at the partner.

It is not possible to set a connection to NOTPENDING state (forcing in-doubt and erasing NOFORGET UOWs) until CICS has made contact with the partner and received a new logname from it.

Decisions for a whole connection can be forgotten, but that does not affect the memory of a decision for any other connection involved in the UOW.

Note: If you specify NOTPENDING you cannot specify COMMIT, BACKOUT, FORCEUOW, RESYNC, or NORECOVDATA. For advice on which option to use, see the notes following the description of the RESYNC option.

The exchange lognames function and the resynchronization function are described in the *CICS Intercommunication Guide* and the *Systems Network Architecture—LU6.2 Reference: Peer Protocols* manual.

Outservice

Place the system out of service; that is, unavailable for use.

Purge

specifies that transactions running on the connected system are to be abnormally terminated. Transactions are terminated only if system and data integrity can be maintained. A transaction is not purged if its definition specifies SPURGE=NO.

RELeased

specifies that CICS is to release a session with the system represented by the IPCONN name.

RESync

specifies that any UOWs shunted due to the failure of this connection are to be retried. (That is, exchange lognames resynchronization for this connection is to be attempted.) This process should normally be started automatically when a connection is acquired or when a UOW is unshunted. The normal resynchronization process is partially overridden: decisions are taken for any units of work that are in-doubt due to a failure of the connection; but the decisions are recorded and any data inconsistencies are reported when the connection is next acquired.

Note:

1. The COMMIT, BACKOUT, FORCEUOW and RESYNC operations are synchronous with setting the state of the UOW; that is, an INQUIRE UOW following SET IPCONN BACKOUT, COMMIT, FORCEUOW, or RESYNC returns the new UOW states.

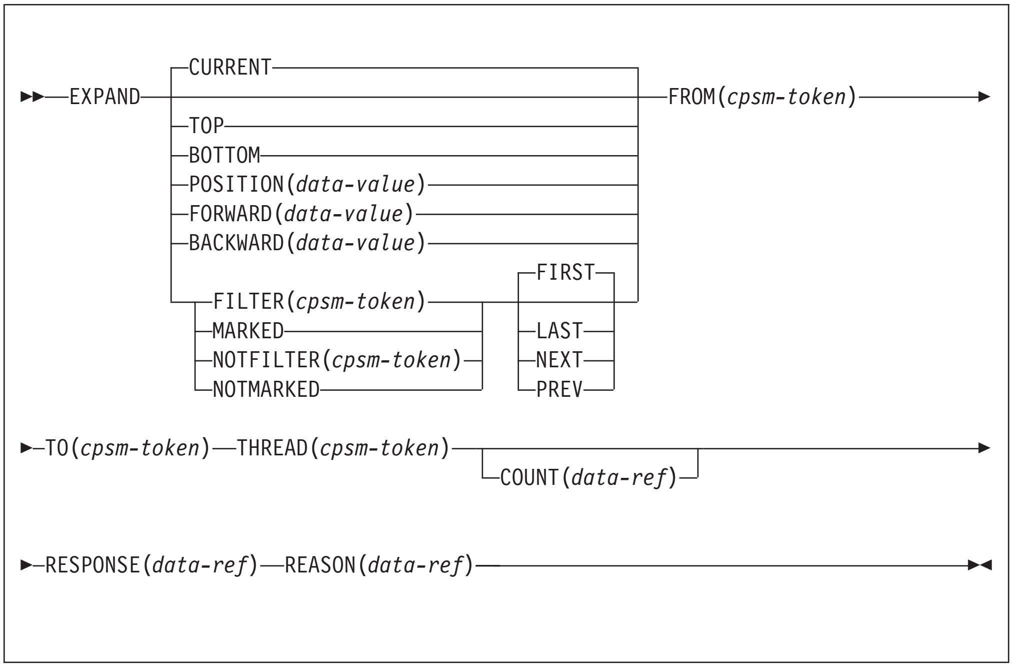
2. Specifying one of these options unshunts all units of work that have failed due to a failure of the connection. Before issuing SET IPCONN FORCEUOW, you may want to use the SET UOW command to specify commit or backout for each in-doubt unit of work explicitly, rather than letting it default, Local procedures determine the importance of the data and the method of using the INQUIRE UOW, INQUIRE UOWENQ, and INQUIRE UOWLINK commands to establish the correct actions.
3. You can specify only one of the BACKOUT, COMMIT, FORCEUOW, RESYNC, NOTPENDING, and NORECOVDATA options. SET IPCONN NORECOVDATA should be used only in exceptional circumstances.
4. To force all in-doubt units of work caused by a failure of the connection in the same direction, use SET IPCONN COMMIT or SET IPCONN BACKOUT.
5. The BACKOUT, COMMIT, FORCEUOW, or RESYNC options of SET IPCONN and SET UOW do not clear resync information. If you want to do this, you must use SET IPCONN NOTPENDING or SET IPCONN NORECOVDATA.
6. You can issue BACKOUT, COMMIT, FORCEUOW, or RESYNC commands *before* issuing SET IPCONN NOTPENDING or SET IPCONN NORECOVDATA.

Appendix F. New CICSPlex SM application programming commands

CICS Transaction Server for z/OS, Version 3 Release 2 extends the CICSPlex SM application programming interface with new commands.

EXPAND

The EXPAND command returns a result set containing all of the records summarized in a summary record.



Description

This command supports the expansion of summary result sets. The command accepts a token from a summarized result set produced by the GROUP command, and a selected record identified by the position of the record pointer in the result set to be expanded. The position of the record pointer depends on the options that you specify on the command. It creates a new result set that contains all the records that are summarized in a summary record.

Related commands

FETCH, GET, GROUP, LOCATE, MARK, ORDER, QUERY, REFRESH, SPECIFY FILTER, UNMARK

Options

BACKWARD*(data-value)*
Expands the record at the position arrived at by moving backwards from the

current pointer position for *(data-value)*number of records. If the *(data-value)* value is greater than the remaining number of records, the first record in the summary result set is expanded.

BOTTOM
Expands the last record in the summary result set.

COUNT*(data-ref)*
The number of resource table records in the TO result set after this operation is complete. This is an output-only parameter.

CURRENT
Expands the current record in the FROM result set.

FILTER*(cpsm-token)*
Identifies the filter that is to be used for this operation and performs an EXPAND operation on the record or records that match the filter criteria. It is used in conjunction with the FIRST, LAST, NEXT and PREV options.

FIRST
Expands either the first marked record in the result set, or the first record that matches the filter criteria. If no record is found, a NODATA code is returned.

FORWARD*(data-value)*
Expands the record at the position arrived at by moving forwards from the current pointer position for *(data-value)* number of records. If the *(data-value)* value is greater than the remaining number of records in the summary result set, the last record is expanded.

FROM *(cpsm-token)*
The summary result set on which the EXPAND command is to operate. If no matching result set can be found, an INVALIDPARM return code is issued with a reason code of FROM.

LAST
Expands either the last marked record in the result set, or the last record that matches the filter criteria. If no record is found, a NODATA code is returned.

MARKED
Expands one or more records that have been selected using the MARK command. It is used in conjunction with the FIRST, LAST, NEXT and PREV options.

You can mark resource table records by using the MARK and UNMARK commands.

NEXT
Starting at the record currently selected and moving forward through the result set, NEXT expands either the next marked record, or the next record that matches the filter criteria. If no record is found, a NODATA code is returned.

NOTFILTER*(cpsm-token)*
Identifies the filter that is to be used for this operation and performs an EXPAND operation on the record or records that do not match the filter criteria. It is used in conjunction with the FIRST, LAST, NEXT and PREV options.

NOTMARKED
Expands one or more records that have been left unselected by the MARK command. It is used in conjunction with the FIRST, LAST, NEXT and PREV options.

You can mark resource table records by using the MARK and UNMARK commands.

POSITION*(data-value)*
Expands the record at a position in the summary result set indicated by the supplied value.

PREVIOUS
Starting at the record currently selected and moving backwards through the result set, PREVIOUS expands either the next marked record in the result set, or the next record that matches the filter criteria. If no record is found, a NODATA code is returned.

REASON*(data-ref)*
Names a variable to receive the fullword reason value returned by this command.

RESPONSE*(data-ref)*
Names a variable to receive the fullword response value returned by this command.

THREAD*(cpsm-token)*
The API thread to be used for the EXPAND operation. The *cpsm-token* value that identifies a thread is returned by the CONNECT command.

TO *(cpsm-token)*
Identifies the summary result set to contain the expanded records on which the EXPAND command is to operate. If this result set already exists, any existing resource table records that relate to it are replaced by the resource table records produced by this EXPAND command.

TOP
Expands the first record in the summary result set.

Conditions

The following is a list of the RESPONSE values that can be returned by the EXPAND command. The description of each RESPONSE includes a list of associated REASON values, if appropriate.

OK The command completed processing successfully.

NODATA
No records were found that matched the specified criteria, for one of the following reasons.

BACKWARD
There are no more records that satisfy the search criteria in the backward direction.

FORWARD
There are no more records that satisfy the search criteria in the forward direction.

BUSY A busy condition occurred for one of the following reasons:

FROM The result set specified on the FROM option is being processed by another command.

TO The result set specified on the TO option is being processed by another command.

ENVIRONERROR
An environment error occurred for one of the following reasons:

NOSERVICE
The application stub program could not load the API service module.

NOSTORAGE
The application stub program could not obtain the necessary storage in the address space where the processing thread is running.

FAILED
The command failed for one of the following reasons:

ABENDED
Command processing terminated abnormally.

EXCEPTION
Command processing encountered an exceptional condition.

INVALIDPARM
An invalid parameter was detected. The parameter that is invalid is returned as the reason value:

- BACKWARD
- FORWARD
- POSITION
- FILTER
- NOTFILTER
- FROM
- TO
- THREAD
- COUNT

Check the command description for valid parameter syntax.

NOTAVAILABLE
A not available condition occurred for one of the following reasons:

APITASK
The API control subtask is not active.

CPSMAPI
The CMAS to which the processing thread is connected is not available for API processing.

SERVERGONE
The CMAS to which the processing thread was connected is no longer active.

VERSIONINVL
A version conflict occurred for one of the following reasons:

NOTSUPPORTED
The version of the application stub program used for this command is not supported.

NOTVSNCONN
The version of the application stub program used for this command is not the same as the version used with the CONNECT command.

Appendix G. New pipeline configuration elements

CICS Transaction Server for z/OS, Version 3 Release 2 defines new XML elements that can be coded in a pipeline configuration file.

The <auth_token_type>element

Specifies what type of identity token is required.

This element is mandatory when you specify the `<sts_authentication>` element in a service requester pipeline, and optional in a service provider.

- In a service requester pipeline, the `<auth_token_type>` element indicates what type of token the STS should issue when CICS sends it the user ID contained in the DFHWS-USERID container. The token that CICS gets back from the STS is placed in the header of the outbound message.
- In a service provider pipeline, the `<auth_token_type>` element is used to determine which identity token CICS should take from the message header and send to the STS to exchange or validate. CICS uses the first identity token of the specified type in the message header. If you don't specify this element, CICS uses the first identity token that it finds in the message header. CICS does not consider the following as identity tokens:
  - `wsu:Timestamp`
  - `xenc:ReferenceList`
  - `xenc:EncryptedKey`
  - `ds:Signature`

Used in:

- Service provider
- Service requester

Contained by:

`<sts_authentication>`

Contains:

1. A `<namespace>` element. This element contains the namespace of the token type that should be validated or exchanged.
2. An `<element>` element. This element contains the local name of the token type that should be validated or exchanged.

The values of these elements form the Qname of the token.

Example

```
<auth_token_type>
   <namespace>http://example.org.tokens</namespace>
   <element>UsernameToken</element>
</auth_token_type>
```

The <cics_mtom_handler>element

Enables the CICS-supplied MTOM handler program, that provides support for MTOM MIME multipart/related messages that contain XOP documents and binary attachments. MTOM support is enabled for all inbound messages that are received in the pipeline, but MTOM support for outbound messages is conditionally enabled subject to further options.

Used in:

- Service provider
- Service requester

Contained by:

<provider_pipeline>

<requester_pipeline>

In a provider pipeline configuration file, the <cics_mtom_handler> element should be defined before the <transport> element. At run time, the MTOM handler program needs to unpackage the inbound MTOM message before other handlers including the transport handler process it. It will also then be invoked as the last handler for the response message, to package an MTOM message to send to the Web service requester.

In a requester pipeline configuration file, <cics_mtom_handler> element should be defined after the <transport> element. At run time, the outbound request message is not converted into MTOM format until all other handlers have processed it. It will then also be invoked as the first handler for the inbound response message to unpackage the MTOM message before other handlers process it and return to the requesting program.

Contains:

<dfhmtom_configuration> element

Default options can be changed using configuration options specified in the <dfhmtom_configuration> element. If you do not want to change the default options, you can use an empty element.

Example

For a provider mode pipeline, you could specify:

```
<provider_pipeline>
    <cics_mtom_handler/>
    <transport>
    ....
    </transport>
    <service>
    ....
    </service>
</provider_pipeline>
```

The <dfhmtom_configuration>element

Specifies configuration information for the CICS-supplied MTOM handler program, which provides support for MIME messages that contain XOP documents and binary attachments. If you do not specify any configuration for MTOM, CICS assumes default values.

Used in:

- Service provider
- Service requester

Contained by:

`<cics_mtom_handler>`

Attributes:

Name	Description
version	An integer denoting the version of the configuration information. The only valid value is 1.

Contains:

1. An optional `<mtom_options>` element
2. An optional `<xop_options>` element
3. An optional `<mime_options>` element

Example

```
<dfhmtom_configuration version="1">
  <mtom_options send_mtom="same" send_when_no_xop="no"/>
  <xop_options apphandler_supports_xop="yes"/>
  <mime_options content_id_domain="example.org"/>
</dfhmtom_configuration>
```

The <mime_options>element

Specifies the domain name that should be used when generating MIME content-ID values, that are used to identify binary attachments.

Used in:

- Service provider
- Service requester

Contained by:

`<dfhmtom_configuration>`

Attributes:

Attribute	Description
content_id_domain	The syntax to use is *domain.name*. To conform to Internet standards, the name should be a valid internet host name and should be unique to the CICS system where the pipeline is installed. Note that this is not checked by CICS. If this element is omitted, CICS uses the value `cicsts`.

Example

```
<provider_pipeline>
<dfhmtom_configuration version="1">
  <mime_options content_id_domain="example.org"/>
</dfhmtom_configuration>
...
</provider_pipeline>
```

In this example, references to binary attachments are created using `cid:`*`unique_value`*`@example.org`.

The <mtom_options>element

Specifies when to use MTOM for outbound SOAP messages.

Used in:

- Service provider
- Service requester

Contained by:

`<dfhmtom_configuration>`

Attributes:

Attribute	Description
send_mtom	Specifies if MTOM should be used to convert the outbound SOAP message into a MIME message: **no** MTOM is not used for outbound SOAP messages. **same** In service provider mode, MTOM is used for SOAP response messages whenever the requester uses MTOM. This is the default value in a service provider pipeline. In service requester mode, specifying this value is the same as when you specify send_mtom=″yes″. **yes** MTOM is used for all outbound SOAP messages. This is the default value in a service requester pipeline.
send_when_no_xop	Specifies if an MTOM message should be sent, even when there are no binary attachments present in the message. **no** MTOM is only used when binary attachments are being sent with the message. **yes** MTOM is used for all outbound SOAP messages, even when there are no binary attachments to send in the message. This is the default value, and is primarily used as an indicator to the receiving program that the sender supports MTOM/XOP. This attribute can be combined with any of the send_mtom attribute values, but has no effect if you specify `send_mtom="no"`.

Example

```
<provider_pipeline>
 <cics_mtom_handler>
  <dfhmtom_configuration version="1">
    <mtom_options send_mtom="same" send_when_no_xop="no"/>
  </dfhmtom_configuration>
 </cics_mtom_handler>
...
</provider_pipeline>
```

In this provider pipeline example, SOAP messages are converted into MTOM messages only when binary attachments need to be sent with the message and the service requester sent an MTOM message.

The <xop_options>element

Specifies whether XOP processing can take place in direct or compatibility mode.

Used in:

- Service provider
- Service requester

Contained by:

```
<dfhmtom_configuration>
```

Attributes:

Attribute	Description
apphandler_supports_xop	In provider mode, specifies if the application handler is capable of handling XOP documents in direct mode: **no** The application handler cannot handle XOP documents directly. This is the default value if the `<apphandler>` element does not specify DFHPITP. Compatibility mode is used in the pipeline to handle any inbound or outbound messages that are received or sent in MTOM format. **yes** The application handler can handle XOP documents. This is the default value if the `<apphandler>` element specifies DFHPITP. Direct mode is used in the pipeline to handle any inbound or outbound messages that are received or sent in MTOM format. This is subject to restrictions at run time. For example, if you have specified WS-Security related elements in the pipeline configuration file, the MTOM handler determines that the pipeline should use compatibility mode rather than direct mode for processing XOP documents. In requester mode, specifies if service requester applications use the CICS Web services support to create and handle XOP documents in direct mode. **no** Service requester applications do not use the CICS Web services support. Specify this value if your requester application links to DFHPIRT to drive the pipeline, and is therefore not capable of creating and handling XOP documents in direct mode. **yes** Service requester applications do use the CICS Web services support. Specify this value if your requester application uses the EXEC CICS INVOKE WEBSERVICE command.

Example

```
<provider_pipeline>
 <cics_mtom_handler>
  <dfhmtom_configuration version="1">
    <xop_options apphandler_supports_xop="no"/>
  </dfhmtom_configuration>
 </cics_mtom_handler>
 ...
</provider_pipeline>
```

In this provider pipeline example, inbound MTOM messages and outbound response messages are processed in the pipeline using compatibility mode.

The <sts_authentication>element

Specifies that a Security Token Service (STS) should be used for authentication and determines what type of request is sent.

Used in:

- Service provider
- Service requester

Contained by:

`<dfhwsse_configuration>`

Attributes:

Name	Description
action	Specifies what type of request CICS should send to the STS when a message is received in the service provider pipeline. Valid values are: **issue** The STS issues an identity token for the SOAP message. **validate** The STS validates the provided identity token and returns whether the token is valid to the security handler. If you do not specify this attribute, CICS assumes that the action is to request an identity token. In a service requester pipeline, you do not need to specify this attribute because CICS always requests that the STS issues a token.

Contains:

1. An `<auth_token_type>` element. This element is required when you specify a `<sts_authentication>` element in a service requester pipeline and optional in a service provider pipeline.
   - In a service requester pipeline, the `<auth_token_type>` element indicates what type of token the STS should issue when CICS sends it the user ID contained in the DFHWS-USERID container. The token that CICS gets back from the STS is placed in the header of the outbound message.
   - In a service provider pipeline, the `<auth_token_type>` element is used to determine which identity token CICS should take from the message header and send to the STS to exchange or validate. CICS uses the first identity token of the specified type in the message header. If you don't specify this element, CICS uses the first identity token that it finds in the message header. CICS does not consider the following as identity tokens:
     - `wsu:Timestamp`
     - `xenc:ReferenceList`
     - `xenc:EncryptedKey`
     - `ds:Signature`
2. In a service provider pipeline only, an optional, empty `<suppress/>` element. If this element is specified, the handler will not attempt to use any security tokens in the message to determine under which user ID the work will run. This includes the identity token that is returned by the STS.

Example

The following example shows a service provider pipeline, where the security handler requests a token from the STS.

```
<sts_authentication action="issue">
   <auth_token_type>
      <namespace>http://example.org.tokens</namespace>
      <element>UsernameToken</element>
   </auth_token_type>
   <suppress/>
</sts_authentication>
```

The <sts_endpoint>element

Specifies the location of the Security Token Service (STS).

Used in:

- Service provider
- Service requester

Contained by:

<dfhwsse_configuration>

Contains:

- An <endpoint> element. This element contains a URI that points to the location of the Security Token Service (STS) on the network. It is recommended that you use SSL or TLS to keep the connection to the STS secure, rather than using HTTP.

  You can also specify a WebSphere MQ endpoint using the JMS format of URI.

Example

In this example, the endpoint is configured to use a secure connection to the STS that is located at the specified URI.

```
<sts_endpoint>
   <endpoint>https://example.com/SecurityTokenService</endpoint>
</sts_endpoint>
```

Appendix H. New global user exits

CICS Transaction Server for z/OS, Version 3 Release 2 introduces new global user exits (GLUEs).

XISQUE exit for managing IPIC intersystem queues

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

Note:

- Queued requests for sessions are known as "*allocate queues*".
- IPIC connections are also known as "*IPCONNs*".
- The equivalent global user exit to control the number of queued requests for sessions on MRO and APPC connections is XZIQUE: see XZIQUE exit for managing MRO and APPC intersystem queues.

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.

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.

Some accessibility features may not be available when using the application assembly tools for enterprise beans (ATK and AAT), which are components of WebSphere Application Server. You should consult the documentation that comes with WebSphere Application Server to determine which accessibility features are available when using these tools.

If you use the resource manager for enterprise beans to work with EJB resources, the accessibility features are those that your Web browser provides. In particular, note that the help that is presented when you allow the mouse pointer to hover over part of the display, is also available through the help function on that panel.

Index

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply in the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM United Kingdom Laboratories, MP151, Hursley Park, Winchester, Hampshire, England, SO21 2JN. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

Trademarks

The following terms are trademarks, or registered trademarks, of International Business Machines Corporation in the United States, or other countries, or both:

AD/Cycle
AIX
C/370
CICS
CICS/ESA
CICSPlex
COBOL/370
DB2
DB2 Universal Database
ESCON
IBM
IMS
Language Environment
MVS
MVS/ESA
OMEGAMON
OS/390
Parallel Sysplex
RACF
Rational
SAA
SecureWay
SupportPac
System z
Tivoli
VisualAge
VSE/ESA
WebSphere
z/Architecture
z/OS
zSeries

Intel is a trademark of Intel in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries or both.

Linux is a trademark of Linus Torvalds 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, 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
Release Guide
Version 3 Release 2

Publication No. GC34-6811-01

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

GC34-6811-01

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

GC34-6811-01

Spine information:

CICS Transaction Server for z/OS **Release Guide**

Version 3
Release 2