

CICS Transaction Server for z/OS
Version 4 Release 2



What's New

CICS Transaction Server for z/OS
Version 4 Release 2



What's New

Note

Before using this information and the product it supports, read the information in "Notices" on page 341.

This edition applies to Version 4 Release 2 of CICS Transaction Server for z/OS (product number 5655-S97) and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 2011, 2012.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Preface	vii
What this book is about	vii
Who this book is for	vii
What you need to know to understand this book	vii
Notes on terminology	vii
Syntax notation	viii

Part 1. Events 1

Chapter 1. Improvements to event processing	3
Terminology	4
Changes to external interfaces	5
Changes to the CICS application programming interface	5
Changes to resources and attributes	5
Changes to supplied transactions	10
Changes to global user exits	11
Changes to user-replaceable programs	11
Changes to monitoring data	12
Changes to statistics	12
Changes to sample programs	12
Changes to problem determination	13
Changes to security	15

Part 2. Java 17

Chapter 2. Support for Java applications in a JVM server	19
Terminology	19
Changes to external interfaces	20
Changes to resources and attributes	20
Changes to global user exits	22
Changes to statistics	22
Changes to the JCICS packaging	23
Changes to sample programs and profiles	24
Changes to problem determination	25

Chapter 3. Developing Java applications in the CICS Explorer SDK	27
---	-----------

Chapter 4. Java runtime support for JZOS and J2C copybook importers	29
--	-----------

Chapter 5. Support for Java-based SOAP pipelines	31
Changes to external interfaces	31
Changes to the pipeline configuration	31
Changes to other programming interfaces	32
Changes to sample profiles	32
Changes to sample pipeline files	32
Changes to problem determination	33

Chapter 6. Enhancements to JVM support	35
Changes to installation procedures	35
Changes to external interfaces	35
Changes to user-replaceable programs	35
Changes to sample profiles and programs	36
Changes to problem determination	36

Part 3. Connectivity 39

Chapter 7. Support for web services discovery	41
Changes to external interfaces	41
Changes to resources and attributes	41
Changes to statistics	42
Changes to problem determination	42

Chapter 8. HTTP connection management	43
Changes to external interfaces	44
Changes to the CICS application programming interface	44
Changes to resources and attributes	45
Changes to statistics	47
Changes to problem determination	47

Chapter 9. IPIC support for function shipping	49
Terminology	49
Changes to external interfaces	50
Changes to system initialization parameters	50
Changes to resources and attributes	50
Changes to global user exits	50
Changes to problem determination	51
Changes to statistics	52

Chapter 10. Recovery support for WebSphere MQ	53
Changes to external interfaces	53
Changes to resources and attributes	54
Changes to statistics	55
Changes to problem determination	55

Chapter 11. Enhancements to CICS DB2 thread reuse	57
Changes to external interfaces	57
Changes to resources and attributes	57
Changes to statistics	58

Chapter 12. Enhancements to Atom feeds	59
Changes to external interfaces	59

Changes to the Atom configuration file	59
Changes to resources and attributes	60
Changes to sample programs	61
Changes to problem determination	61

Part 4. Management 63

Chapter 13. Enhancements to transaction tracking 65

Terminology	65
Changes to external interfaces	66
Changes to monitoring data	66
Changes to interfaces	68
Changes to task-related user exits	69

Chapter 14. Improvements to dynamic workload management 71

Changes to external interfaces	71
Changes to CICSplex SM views and menus	71
Changes to CICSplex SM resource tables	73
Changes to user-replaceable programs	74

Chapter 15. CICS support for password phrases 77

Terminology	78
Changes to external interfaces	78
Changes to the CICS application programming interface	78
Changes to supplied transactions	78
Changes to global user exits	79
Changes to problem determination	79

Chapter 16. Dynamic transaction-level control for workload routing 81

Changes to external interfaces	81
Changes to resources and attributes	81

Chapter 17. New WLM routing algorithms to exclude link weighting 83

Changes to external interfaces	83
Changes to resources and attributes	83

Chapter 18. Modified WLM routing for IPIC connections 85

Chapter 19. Enhancement to system availability monitoring 87

Chapter 20. CMCI support for task history recording 89

Changes to external interfaces	89
Changes to CMCI	89

Chapter 21. CMCI support for multi-field sorting of returned records 91

Changes to external interfaces	91
--	----

Changes to the CICS application programming interface	91
Changes to problem determination	91

Chapter 22. Addition of MVS job ID information to CICSplex SM 93

Chapter 23. CICSplex SM API improvements: Discovering system initialization parameters 95

Changes to external interfaces	95
Changes to resources and attributes	96

Chapter 24. Enhancements to main and auxiliary temporary storage queues 97

Changes to external interfaces	97
Changes to system initialization parameters	97
Changes to resources and attributes	98
Changes to statistics	99
Changes to problem determination	100

Chapter 25. Enhancements to CECI and EDF 101

Part 5. Scalability 103

Chapter 26. Threadsafe extensions 105

Threadsafe API and SPI commands	105
Threadsafe programs	107

Chapter 27. CICS use of 64-bit storage 109

Terminology	110
Changes to external interfaces	110
Changes to resources and attributes	111
Changes to statistics	112
Changes to problem determination	113

Chapter 28. Changes to storage use for CICS trace 115

Changes to external interfaces	115
Changes to system initialization parameters	115
Changes to statistics	116
Changes to problem determination	116

Chapter 29. Changes to the overall limits for the CICS extended dynamic storage areas 119

Changes to external interfaces	119
Changes to system initialization parameters	119

Chapter 30. Changes to the number of LSR pools 121

Changes to external interfaces	121
Changes to system initialization parameters	121
Changes to resources and attributes	122
Changes to statistics	123

Changes to sample programs	123
Changes to problem determination	124
Chapter 31. Enhancements to IMS Database Control (DBCTL).	125
Changes to external interfaces	125
Changes to task-related user exits	125
Chapter 32. Concurrency in program resource definitions	127
Changes to external interfaces	127
Changes to resources and attributes	127
Changes to user-replaceable programs	128
Changes to statistics	128
Part 6. Discontinued functions	131
Chapter 33. Removal of support for CICS Events for WebSphere Business Events SupportPac CB11	133
Part 7. Changes to the CICS Explorer	135
Part 8. General information	139
Chapter 34. High-level language support	141
Compilers and high-level language versions withdrawn from service	142
Chapter 35. Where is the migration information?	147
Part 9. Publications	149
Chapter 36. The CICS Transaction Server for z/OS library	151
Information provided in HTML only	151
Books available as hardcopy	151
Publications available in Portable Document Format (PDF)	152
CICS books for CICS Transaction Server for z/OS	152
CICSplex SM books for CICS Transaction Server for z/OS	153
Licensed publications	153
CICS family books	153
Part 10. Appendixes	155
Appendix A. New application programming commands	157
BIF DIGEST	157
INVOKE SERVICE	158

SIGNAL EVENT	162
TRANSFORM DATATOXML	164
TRANSFORM XMLTODATA	167
WEB ENDBROWSE QUERYPARM	170
WEB READ QUERYPARM	170
WEB READNEXT QUERYPARM	172
WEB STARTBROWSE QUERYPARM	174
WSACONTEXT BUILD	175
WSAEPR CREATE	180
WSACONTEXT DELETE	182
WSACONTEXT GET	183

Appendix B. New RDO resources	189
ATOMSERVICE resources	189
Installing ATOMSERVICE resource definitions	189
ATOMSERVICE attributes	190
BUNDLE resources	192
BUNDLE attributes	193
JVMSERVER resources	195
JVMSERVER attributes	195
MQCONN resources	197
MQCONN attributes	197

Appendix C. New system programming commands	201
CREATE ATOMSERVICE	201
CREATE BUNDLE	203
CREATE JVMSERVER	204
CREATE MQCONN	206
CSD ADD	208
CSD ALTER	209
CSD APPEND	212
CSD COPY	214
CSD DEFINE	216
CSD DELETE	220
CSD DISCONNECT	222
CSD ENDBRGROUP	222
CSD ENDBRLIST	223
CSD ENDBRRSRCE	224
CSD GETNEXTGROUP	225
CSD GETNEXTLIST	226
CSD GETNEXTRSRCE	226
CSD INQUIREGROUP	229
CSD INQUIRELIST	230
CSD INQUIRERSRCE	230
CSD INSTALL	233
CSD LOCK	236
CSD REMOVE	237
CSD RENAME	239
CSD STARTBRGROUP	242
CSD STARTBRLIST	243
CSD STARTBRRSRCE	244
CSD UNLOCK	245
CSD USERDEFINE	246
DISCARD ATOMSERVICE	249
DISCARD BUNDLE	250
DISCARD JVMSERVER	251
DISCARD MQCONN	252
INQUIRE ATOMSERVICE	253
INQUIRE BUNDLE	257

INQUIRE BUNDLEPART	260
INQUIRE CAPTURESPEC	262
INQUIRE EPADAPTER	268
INQUIRE EVENTBINDING	273
INQUIRE EVENTPROCESS	275
INQUIRE JVMSERVER	276
INQUIRE MQCONN	280
INQUIRE MQINI	283
INQUIRE XMLTRANSFORM	286
SET ATOMSERVICE	290
SET BUNDLE	290
SET EPADAPTER	291
SET EVENTBINDING	292
SET EVENTPROCESS	293
SET JVMSERVER	295
SET MQCONN	297
SET XMLTRANSFORM	300

Appendix D. New CEMT commands 303

CEMT DISCARD	303
CEMT INQUIRE ATOMSERVICE.	308
CEMT INQUIRE BUNDLE	312

CEMT INQUIRE EPADAPTER	314
CEMT INQUIRE EVENTBINDING	319
CEMT INQUIRE EVENTPROCESS	321
CEMT INQUIRE JVMSERVER.	321
CEMT INQUIRE MQCONN	325
CEMT INQUIRE MQINI	328
CEMT SET ATOMSERVICE.	330
CEMT SET BUNDLE	331
CEMT SET EPADAPTER	332
CEMT SET EVENTBINDING	333
CEMT SET EVENTPROCESS	333
CEMT SET JVMSERVER.	334
CEMT SET MQCONN	335

Appendix E. New global user exits 339

Exit XISQLCL	339
------------------------	-----

Notices 341

Trademarks	342
----------------------	-----

Accessibility. 343

Preface

What this book is about

This book provides information about new and changed function in CICS® Transaction Server for z/OS®, Version 4 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® System Manager, 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.

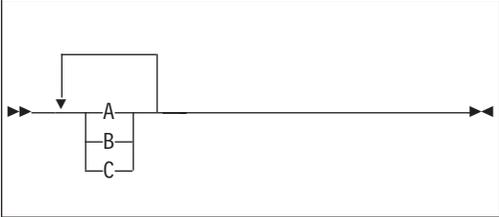
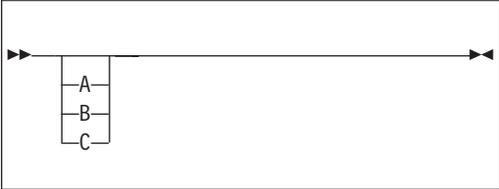
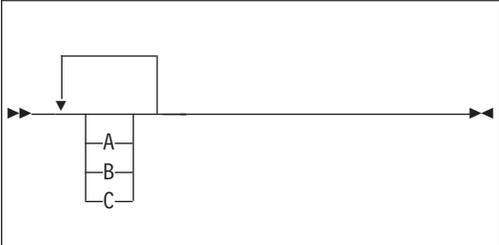
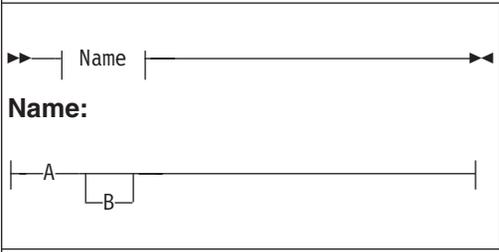
“CICSplex SM” is used for the CICSplex System Manager element of IBM CICS TS.

“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
	<p>Denotes a set of required alternatives. You must specify one (and only one) of the values shown.</p>
	<p>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.</p>
	<p>Denotes a set of optional alternatives. You can specify none, or one, of the values shown.</p>
	<p>Denotes a set of optional alternatives. You can specify none, one, or more than one of the values shown, in any sequence.</p>
	<p>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.</p>
 <p>Name:</p>	<p>A reference to a named section of syntax notation.</p>
	<p>A= denote characters that should be entered exactly as shown.</p> <p><i>value</i> denotes a variable, for which you should specify an appropriate value.</p>

Part 1. Events

CICS Transaction Server for z/OS, Version 4 Release 2 delivers a set of capabilities represented in the following themes:

- Events
- Java
- Connectivity
- Management
- Scalability

The *Events* theme provides the capability for system events to be emitted using the CICS business event infrastructure, providing monitoring information about the CICS system. Enhanced CICS bundle support reduces setup time and complexity by separating system programmer and application developer activities, with common technology for both system- and application-related events.

Chapter 1. Improvements to event processing

The scope of event processing is expanded to include everything from events that check the health of your CICS system, to the assured emission of events, to capture specification improvements, and more.

System events

You can use system events to help you understand and manage the health and availability of your CICS system by emitting events that notify you when system health or system resource state changes or when a threshold has been crossed.

Event processing supports the following system events:

- FILE enable or disable status
- FILE open or close status
- DB2CONN connection status
- TASK threshold
- TRANCLASS TASK threshold
- Unhandled transaction abend

For more information, see System events.

A new subsidiary schema is added to the CicsEventBinding schema. Using this schema you can pick event data from named data locations for the event type. For more information about schemas, see Event binding and Schema versions.

Assured event emission

You can use assured event emission to specify that certain CICS events are critical and that their emission is assured. Assured event emission uses synchronous processing that is completed within the unit of work of the capturing transaction. When assured event emission is used with WebSphere® MQ queues, event delivery is also assured.

EPADAPTER resource

Using the CICS event binding editor you can now choose whether to use a predefined EPADAPTER resource, or the adapter defined in the Adapter section of the **Adapter** tab. For more information about a predefined EPADAPTER resource, see Event processing (EP) adapter configuration editor.

When you use a separate EP adapter configuration you can easily share it between event bindings and you have only one resource to manage. When you specify an EP adapter configuration embedded in an event binding, then CICS creates an EPADAPTER resource for you with the same name as the event binding. For more information, see Adapter tab.

Search facility

Using the CICS event binding editor, you can use the **EP Search** tab to search event bindings and EP adapter configurations for named resources. You can also search by variable, structure, and copybook name. You can discover which of the

events that are enabled are affected by changes such as a program structure that is defined in a copybook.

Additional data types

Event processing supports:

- Short and long floating point (binary, decimal, and hexadecimal) numbers. The BINFLOAT, DECFLOAT, and HEXFLOAT data types (for filtering and capturing) and the scientific format data type are recognized by the event binding installation.
- The full set of COBOL zoned decimal data types, including:
 - PIC S9999 DISPLAY SIGN LEADING
 - PIC S9999 DISPLAY SIGN LEADING SEPARATE
 - PIC S9999 DISPLAY SIGN TRAILING SEPARATE
- The CHARZ and HEXZ filter and capture data types for null-terminated character and hexadecimal strings

TSQ EP adapter XML event format

The XML event formats (common base event, common base event REST, and WebSphere Business Events (WBE)), are supported for the temporary storage (TS) queue EP adapter.

Capture specification improvements

New options for the INQUIRE CAPTURESPEC command mean that system programmers can determine information about any primary predicate or application context filters that are enabled for a given capture specification. Programmers can also determine how many application command options, application data predicates, and information sources exist for a given capture specification. Details about the predicates and information sources that are defined for a given capture specification can be seen by using the new INQUIRE CAPDATAPRED, CAPOPTPRED, and CAPINFOSRCE commands.

You can define capture specifications to emit events from file and temporary storage commands issued by the CICS Atom support and from EXEC CICS LINK commands that are issued by the CICS-WebSphere MQ bridge program.

Terminology

Terms used to explain event processing.

application event. A type of business event that contains application data.

assured event emission. A synchronous method of emitting events within the unit of work for the capturing transaction.

business event. An occurrence of significance to a business. Application events and system events can be business events.

deferred filter request. A filter request that is performed under the CEPF system task rather than the requesting task to avoid suspends of critical system tasks.

EP adapter configuration. An XML definition that defines one EP adapter to CICS, and can be deployed to CICS in a CICS bundle.

IBM Business Monitor. WebSphere Business Monitor (WBM) has been renamed IBM Business Monitor from version 7.5.

system event. A business event that is emitted by the system. System events can include resource state changes, thresholds being crossed, unusual system states or actions, or an input event.

Changes to external interfaces

Some external interfaces now support event processing.

Changes to the CICS application programming interface

CICS applications can now query EPADAPTER resource security using the QUERY SECURITY command.

Changed command, QUERY SECURITY

The QUERY SECURITY command is changed to include a new resource type of EPADAPTER.

For more information, see QUERY SECURITY.

Changes to resources and attributes

EVENTPROCESS, EVENTBINDING, and CAPTURESPEC are changed to support system events and assured events, and a new EPADAPTER resource is introduced. CICSplex SM resources EVCSPEC, EVNTBIND, EVNTGBL, HTASK, and TASK are changed and new CICSplex SM resources CRESEPAD, EPADAPT, EVCSDATA, EVCSINFO, and EVCSOPT are added to support new CICS event processing functions.

Support for new resource: CRESEPAD

You can use the new CRESEPAD resource to display information about the event processing adapters.

The new CRESEPAD resource has the following support:

- New CMCI external resource, CICSTopologyEPAdapter.
- New CICSplex SM resource table, CRESEPAD.

Support for new resource: EPADAPT

You can use the new EPADAPT resource to display information about the event processing adapter for a specified event binding.

The new EPADAPT resource has the following support:

- New CMCI external resources, CICSEPADapter and CICSTopologyEPAdapter.
- New CICSplex SM view EYUSTARTEPADAPT, **CICS operations views > Application operations views > Event processing adapters.**
- Updated CICSplex SM view EYUSTARTAPPLICTN, **CICS operations views > Application operations views.**
- New CICSplex SM resource table, EPADAPT.
- New INQUIRE EPADAPTER and CEMT INQUIRE EPADAPTER commands.
- New SET EPADAPTER and CEMT SET EPADAPTER commands.

Support for new resource: EVCSDATA

You can use the new EVCSDATA resource to display information about any application data predicates that are defined for a capture specification.

The new EVCSDATA resource has the following support:

- New Explorer view, **Event Processing > Data predicates**.
- New CMCI external resource CICS CaptureSpecificationDataPredicate.
- New CICSplex SM view EYUSTARTEVCSDATA, **CICS operations views > Application operations views > Event capture specification data predicates**.
- New CICSplex SM resource table, EVCSDATA.
- New INQUIRE CAPDATAPRED command.

Support for new resource: EVCSINFO

You can use the new EVCSINFO resource to display information about any information sources that are defined in the capture specification for a system or application event.

The new EVCSINFO resource has the following support:

- New Explorer view, **Event Processing > Information sources**.
- New CMCI external resource CICS CaptureSpecificationInformationSource.
- New CICSplex SM view EYUSTARTEVCSINFO, **CICS operations views > Application operations views > Event capture specification information source**.
- New CICSplex SM resource table, EVCSINFO.
- New INQUIRE CAPINFOSRCE command.

Support for new resource: EVCSOPT

You can use the new EVCSOPT resource to display information about any application options that are defined in the capture specification for a system or application event.

The new EVCSOPT resource has the following support:

- New Explorer view, **Event Processing > Application operations**.
- New CMCI external resource CICS CaptureSpecificationOptionPredicate.
- New CICSplex SM view EYUSTARTEVCSOPT, **CICS operations views > Application operations views > Event capture specification option predicates**.
- New CICSplex SM resource table, EVCSOPT.
- New INQUIRE CAPOPTPRED command.

Support for updated resource: EVCSPEC

You can use the updated EVCSPEC resource to display information about CAPTURESPEC statistics for system events, assured emission and for new options on the INQUIRE CAPTURESPEC command.

The updated EVCSPEC resource has the following changes:

- Updated Explorer view, **Event Processing > Capture Specifications**.
- Updated CMCI external resources, CICS CaptureSpecification and CICS TopologyCaptureSpecification.

- Updated CICSplex SM view EYUSTARTEVCSPEC, **CICS operations views > Application operations views > Event capture specifications.**
- Updated CICSplex SM resource table, EVCSPEC.
- Updated INQUIRE CAPTURESPEC command.

Support for updated resource: EVNTBIND

You can use the updated EVNTBIND resource to display information about EVENTBINDING statistics for assured emission.

The updated EVNTBIND resource has the following changes:

- Updated Explorer view, **Event Processing > Event Bindings.**
- Updated CMCI external resources, CICSEventBinding and CICSTopologyEventBinding.
- Updated CICSplex SM view EYUSTARTEVNTBIND, **CICS operations views > Application operations views > Event binding.**
- Updated CICSplex SM resource table, EVNTBIND.
- Updated INQUIRE EVENTBINDING and CEMT INQUIRE EVENTBINDING command.
- Updated INQUIRE EVENTPROCESS and CEMT INQUIRE EVENTPROCESS commands.

Support for updated resource: EVNTGBL

You can use the updated EVNTGBL resource to display information about EVENTPROCESS statistics for the HTTP EP adapter, schema level, assured emission, and events with disabled or missing EPADAPTER.

The updated EVNTGBL resource has the following changes:

- Updated Explorer view, **Event Processing > Event Processing.**
- Updated CMCI external resource, CICSEventProcessing.
- Updated CICSplex SM view EYUSTARTEVNTGBL, **CICS operations views > Application operations views > Event processing.**
- Updated CICSplex SM resource table, EVNTGBL.

Support for updated resource: HTASK

You can use the updated HTASK resource to display historical information about completed event processing tasks.

The updated HTASK resource has the following changes:

- Updated CMCI external resource, CICSTaskHistory.
- Updated CICSplex SM view EYUSTARTHTASK, **CICS operations views > Task operations views > Completed tasks.**
- Updated CICSplex SM resource table, HTASK.

Support for updated resource: TASK

You can use the updated TASK resource to display information about currently running event processing tasks.

The updated TASK resource has the following changes:

- Updated CMCI external resource, CICSTask.
- Updated CICSplex SM view EYUSTARTTASK, **CICS operations views > Task operations views > Active tasks.**
- Updated CICSplex SM resource table, TASK.

CAPTURESPEC resource updates

The following interfaces are updated to support the CAPTURESPEC resource:

CICSplex SM

- ☞ The Event capture specifications - EVCSPEC view
- ☞ The EVCSPEC Resource Table

CICS SPI

- ☞ The INQUIRE CAPTURESPEC command

EPADAPTER resource updates

The following interfaces are updated to support the EPADAPTER resource:

CICS Explorer®

- ☞ Operations view

CICSplex SM

- ☞ The EP adapter - EPADAPT view
- ☞ The EPADAPT Resource Table

CEMT

- ☞ The CEMT INQUIRE EPADAPTER command
- ☞ The CEMT SET EPADAPTER command

CICS SPI

- ☞ The INQUIRE EPADAPTER command
- ☞ The SET EPADAPTER command

EVCSDATA resource updates

The following interfaces are updated to support the EVCSDATA resource:

CICSplex SM

- ☞ The Event capture specification data predicate - EVCSDATA view
- ☞ The EVCSDATA Resource Table

CICS SPI

- ☞ The INQUIRE CAPDATAPRED command

EVCSINFO resource updates

The following interfaces are updated to support the EVCSINFO resource:

CICSplex SM

- The Event capture specification information source - EVCSINFO view
- The EVCSINFO Resource Table

CICS SPI

- The INQUIRE CAPINFOSRCE command

EVCSOPT resource updates

The following interfaces are updated to support the EVCSOPT resource:

CICSplex SM

- The Event capture specification option predicate - EVCSOPT view
- The EVCSOPT Resource Table

CICS SPI

- The INQUIRE CAPOPTPRED command

EVENTBINDING resource updates

The following interfaces are updated to support the EVENTBINDING resource:

CICS Explorer

- Operations view

CICSplex SM

- The Event binding - EVNTBIND view
- The EVNTBIND Resource Table

CEMT

- The CEMT INQUIRE EVENTBINDING command

CICS SPI

- The INQUIRE EVENTBINDING command

EVENTPROCESS resource updates

The following interfaces are updated to support the EVENTPROCESS resource:

CICS Explorer

- Operations view

CICSplex SM

- The Event processing - EVNTGBL view
- The EVNTGBL resource table

CEMT

➤ The CEMT INQUIRE EVENTPROCESS command

The CICS SPI

➤ The INQUIRE EVENTPROCESS command

HTASK resource updates

The following interfaces are updated to support the HTASK resource:

CICSplex SM

➤ The Completed tasks (history) - HTASK view

➤ The HTASK Resource Table

TASK resource updates

The following interfaces are updated to support the TASK resource:

CICSplex SM

➤ The Active tasks - TASK view

➤ The TASK Resource Table

Changes to supplied transactions

New options for the INQUIRE CAPTURESPEC command mean that you can determine information about primary predicate or application context filters that are set for a given capture specification.

Note: This topic does not describe the following changes:

- Changes to CEMT commands.
- Changes to resources managed by the CEDA transaction.

New transactions

CECI displays the new options on the EXEC CICS CAPTURESPEC command:

CURRPGM(*data-area*)

Specifies an 8-character data area to receive the value specified by the application context predicate for the current program name. Blanks are returned if no application context predicate for the current program name is defined for this capture specification.

CURRTRANID(*data-area*)

Specifies a 4-character data area to receive the value specified by the application context predicate for the current transaction name.

CURRUSERID(*data-area*)

Specifies an 8-character data area to receive the value specified by the application context predicate for the user ID that is associated with the current transaction.

PRIMPRED(*data-area*)

Specifies a 32-character data area to receive the value of the primary predicate for this capture specification. The primary predicate for a capture specification is the predicate to specify with the EQUALS operator; it helps to avoid a

performance impact as more capture specifications are added for a particular capture point. Blanks are returned if there is no named primary predicate defined for this capture point.

Changes to global user exits

The XEPCAP exit can be used to detect when events are captured. The XRSINDI exit provides support for EPADAPTER resources.

New global user exit, XEPCAP

The XEPCAP exit is invoked just before an event is captured by CICS event processing. Use the XEPCAP exit to detect when events are captured.

Changed global user exit, XRSINDI

New parameters are added to the install and discard global user exit, XRSINDI, to support the EPADAPTER resource and the EPADAPTER resource signature parameter.

UEIDEPAD

An EPADAPTER resource.

UEPDEFTM

The address of a variable-length list, which corresponds to the list in UEPIDNAM, containing the definition time of the individual resource as an 8-character STCK value.

Note: The parameters UEPDEFTM, UEPCHUSR, UEPCHAGT, UEPCHREL, UEPCHTIM, UEPDEFSRC, UEPINUSR, UEPINTIM, and UEPINAGT are valid for the following resources: ATOMSERVICE, BUNDLE, CONNECTION, CORBASERVER, DB2CONN, DB2ENTRY, DB2TRAN, DJAR, DOCTEMPLATE, ENQMODEL, EPADAPTER, EVENTBINDING, FILE, IPCONN, JOURNALMODEL, JVMSERVER, LIBRARY, MQCONN, MQINI, OSGIBUNDLE, PIPELINE, PROFILE, PROCESSTYPE, PROGRAM, REQUESTMODEL, TCPIPSERVICE, TDQUEUE, TRANCLASS, TRANSACTION, TSMODEL, URIMAP, WEBSERVICE, and XMLTRANSFORM. The parameter value is zero for all other resources.

Changes to user-replaceable programs

The sample statistics program, DFH0EPAC, is changed to support the new EPADAPTER resource.

Custom EP adapters

Your custom adapter programs must now honor the EPAP_RECOVER flag in the DFHEP.ADAPTPARM container to support synchronous event emission.

The field **EPCX_Program** in the DFHEP.CONTEXT container used by custom EP adapter programs is not set for system events.

Changed event formats

The CICS Event Processing contextual header which is included in both CICS Flattened Events (CFE) and CICS Container-based Events (CCE) is changed to

increase the accuracy of the date and time stamp field EPFE-DATE-TIME to include milliseconds.

Changes to monitoring data

New performance class data in the DFHCICS group supports event processing.

Changed performance class group, DFHCICS

New performance data field 418 is added to the DFHCICS group. This field provides monitoring data:

418 (TYPE-A, 'ECSEVCCT', 4 BYTES)

The number of synchronous emission events captured by the user task.

Changes to statistics

New statistics fields are added to the CAPTURESPEC and EVENTPROCESS statistics types. The statistics sample program, DFH0STAT, and the statistics utility program, DFHSTUP, now display fields for assured emission and the EPADAPTER resource.

New statistics types

EPADAPTER: resource statistics introduce the following statistics fields:

- The name of the EP adapter.
- The adapter type.
- The EP adapter emission mode. This identifies whether the EP adapter is for synchronous or asynchronous events.
- The number of events passed to EP for emission by this adapter.

Changed statistics types

CAPTURESPEC: resource statistics now include the following statistics fields:

- The number of capture failures, recorded by capture specification. When displayed, this statistic is totaled by event binding.

EVENTBINDING: global statistics now include the following statistics fields:

- The number of system events captured.

EVENTBINDING: resource statistics now include the following statistics fields:

- The 32-character name of an EP adapter.

EVENTPROCESS: global statistics now include the following statistics fields:

- The number of synchronous emission events captured.
- The number of synchronous emission events that were not emitted.
- The number of events that were not emitted because the EP adapter is disabled or not installed.

Changes to sample programs

To help you develop your own custom EP adapter, a sample is provided as source code, and also as a load module.

Changed sample program, DFH0EPAC

The sample custom EP adapter, DFH0EPAC, is changed to include the new DFHEPAP copybook.

Your custom EP adapter programs must now also honor the EPAP_RECOVER flag in the DFHEP.ADAPTPARM container to support synchronous event emission.

For system events, event records are written to the default CICS temporary storage queue (TSQ) *userid.SYSTEM*.

Changed sample program, DFH0STEP

The sample statistics program, DFH0STEP, is changed to collect and print EPADAPTER statistics, and take advantage of new options for the **INQUIRE CAPTURESPEC** command.

Changes to problem determination

CICS provides new information to help you diagnose problems relating to event processing.

New messages

- The following new error messages report EP adapter problems: DFHEC1016, DFHEC1022, DFHEC4006, DFHEC4009, DFHEC4010, DFHEC4113, DFHEC4118, DFHEC4119, DFHEC4120, DFHEC4121, DFHEC4122, and DFHEC4123.
- The following new error messages report problems with the event capture domain: DFHEC1011, DFHEC1012, and DFHEC1013.
- New error message DFHEC1023 reports problems when transactional event emission is defined for system events.
- New error message DFHEC1024 reports problems when synchronous event emission is defined for system events.
- New error message DFHEC1026 reports problems with the CEPF task.
- New error messages DFHEP0120 and DFHEP0121 report problems with the event processing domain.
- New error messages DFHEP0122 and DFHEP0123 report EP adapter problems.
- New error messages in the ranges DFHEP1000 to DFHEP1003 and DFHEP2001 to DFHEP2003, and new error message DFHEP2005, report EP adapter problems.

Changed messages

- Changed error messages DFHEC1001, DFHEC1002, DFHEC1003, DFHEC1009, DFHEC4007, DFHEC4008, DFHEC4111, and DFHEC4117 report problems with the event capture domain.
- Changed error messages DFHEP0114, DFHEP0117, and DFHEP0118 report problems with the event processing domain.
- Changed error messages in the range DFHPI1007 to DFHPI1010 report EP adapter problems.

New abend codes

- Abend codes AECE and AECM are issued when problems occur with event processing deferred filtering task CEPF.

Changed abend codes

- Abend code ASP7 is issued when problems occur with emission of synchronous events.

New trace points

The following new trace points are issued by the event capture domain:

- AP 3524 through AP 3528
- AP 353F
- AP 3571 through AP 3573
- AP 357E
- AP 35AA
- AP 35B0
- AP 35CB through AP 35D7
- AP 35FB through AP 35FF
- AP 363A through AP 363C
- AP 366B through AP 366E
- AP 3670 through AP 3672
- AP 3674 through AP 367B
- AP 367E
- AP 3680 through AP 368B
- AP 36A0 through AP 36A3
- AP 36B0 through AP 36BF
- AP 36C0 through AP 36C6
- AP 36D0 through AP 36D7

The following new trace points are issued by the event processing domain:

- EP 042B through EP 0434
- EP 04F0
- EP 0505 through EP 050C
- EP 0609
- EP 06F0 through EP 06F5
- EP 0700 through EP 0729
- EP 07FF
- EP 0800 through EP 0813
- EP 08FF
- EP 0906 through EP 0907
- EP 0B00 through EP 0B04
- EP 0C00 through EP 0C05

Changed trace points

The following changed trace points are issued by the event capture domain:

- AP 366B through 366D
- EP 35D0 through EP 35D4 are renumbered to EP 35E0 through EP 35E4
- EP 35DA through EP 35EA are renumbered to EP 35EA through EP 35FA
- EP 35C5, EP 366F

The following changed trace points are issued by the event processing domain:

- EP 06FE

Changed dump formatting keywords

The EP dump formatter is updated to format EPADAPTER details.

The EC dump formatter is updated to format:

- A summary of the system event point table
- A summary of all installed system event capture specifications
- A summary of any system event deferred filtering requests

Changes to security

Command, resource, and surrogate user security apply to EPADAPTER resources when these security functions are enabled for the CICS region.

Transaction security categories

The CEPF transaction is the event processing deferred filtering task. The CEPF transaction is a category 1 transaction that is implemented by program DFHECDF.

CEPF is defined internally by the event processing domain.

Command security: INQUIRE CAPDATAPRED

The INQUIRE CAPDATAPRED command uses the new CAPDATAPRED resource.

For more information, see INQUIRE CAPDATAPRED.

Command security: INQUIRE CAPINFOSRCE

The INQUIRE CAPINFOSRCE command uses the new CAPINFOSRCE resource.

For more information, see INQUIRE CAPINFOSRCE.

Command security: INQUIRE CAPOPTPRED

The INQUIRE CAPOPTPRED command uses the new CAPOPTPRED resource.

For more information, see INQUIRE CAPOPTPRED.

Command security: QUERY SECURITY

Command security for EPADAPTER resources uses the EPADAPTER resource in the CCICSCMD class or the VCICSCMD grouping class, or equivalent customer-defined classes specified in the **XCMD** system initialization parameter.

The QUERY SECURITY command is changed to include a new resource type of EPADAPTER.

For more information, see QUERY SECURITY.

Resource security: EPADAPTER

Resource security for EPADAPTER resources uses access resource profiles in the RCICSRES class or the WCICSRES grouping class, or equivalent customer-defined classes specified in the **XRES** system initialization parameter.

Resource security: EVENTBINDING

The new **INQUIRE CAPDATAPRED**, **INQUIRE CAPINFOSRCE**, and **INQUIRE CAPOPTPRED** commands use the existing EVENTBINDING resource.

Part 2. Java

CICS Transaction Server for z/OS, Version 4 Release 2 delivers a set of capabilities represented in the following themes:

- Events
- Java
- Connectivity
- Management
- Scalability

The *Java* theme provides a new 64-bit Java runtime environment, with enhancements to the multithreaded Java Virtual Machine (JVM) server infrastructure. This environment allows CICS to run a greater number of larger Java applications in a single region, with better scalability for Java workloads in CICS. Combined with new support for the deployment of OSGi bundles, you can simplify significantly system administration and portability for Java applications. The provision of a new Java-based Axis2 web services engine improves portability of web services applications. All Java workloads are eligible to run on a zAAP.

Chapter 2. Support for Java applications in a JVM server

You can now run Java applications that comply with the OSGi Service Platform specification in a JVM server. The JVM server is a runtime environment that can handle multiple concurrent requests for different Java applications in a single JVM.

The OSGi Service Platform provides a mechanism for developing applications by using a component model and deploying those applications into a framework as *OSGi bundles*. An OSGi bundle is the unit of deployment for an application and contains version control information, dependencies, and application code. The main benefit of OSGi is that you can create applications from reusable components that are accessed only through well-defined interfaces called *OSGi services*. You can also manage the life cycle and dependencies of Java applications in a granular way.

The JVM server includes an OSGi framework that can run threadsafe Java applications. The JVM server can also run Java-based SOAP pipeline processing for web services. Use this runtime environment for all Java applications where possible. It is the preferred method for running Java workloads in a CICS region and provides the following benefits:

- You can run more than one Java application in a JVM server, saving on the memory and storage usage of having many JVMs running in a CICS region.
- You can run eligible Java workloads on System z Application Assist Processors (zAAPs), reducing the cost of transactions.
- You can manage the life cycle of applications in the OSGi framework without restarting the JVM server.

CICS uses the Equinox implementation of the OSGi framework and supports version 4.2 of the OSGi Service Platform specification. For more information about the OSGi Service Platform, see the OSGi Alliance website.

Terminology

Terms used to explain running Java applications in a JVM server.

bundle. In the OSGi service platform, a Java archive file that contains Java code, resources, and a manifest that describes the bundle and its dependencies. The bundle is the unit of deployment for an application.

bundle activator. An interface implemented by a class in a bundle that is used to start and stop that bundle.

JVM pool. A collection of JVMs that together can handle requests from tasks concurrently. JVMs in the pool can be reused for subsequent requests when a task completes.

JVM server. A runtime environment that uses a single JVM to handle requests from multiple tasks concurrently.

middleware bundle. An OSGi bundle that contains classes to implement system functions. For example, IBM supplies OSGi bundles for connecting to WebSphere MQ.

OSGi Alliance. A consortium of more than 20 companies, including IBM, that creates specifications to outline open standards for the management of voice, data, and multimedia wireless and wired networks.

OSGi framework. A general-purpose, secure, and managed Java framework that supports the deployment of bundles.

OSGi service. An interface registered in the OSGi Service Platform and made available for receiving remote or local invocations.

OSGi Service Platform. A specification that delivers an open, common architecture for service providers, developers, software vendors, gateway operators, and equipment vendors to develop, deploy, and manage services in a coordinated way.

pooled JVM. A JVM that handles only one request at a time from a CICS task. The pool of these JVMs can handle multiple tasks concurrently.

Changes to external interfaces

Some external interfaces now support running Java applications in a JVM server.

Changes to resources and attributes

The BUNDLE, PROGRAM, and JVMSERVER resources have changed to support running Java applications in a JVM server.

Support for updated resource: BUNDLE

Use the updated support in the BUNDLE resource to manage the lifecycle of OSGi bundles. You can inquire on the available OSGi bundles and services using any of the following interfaces:

- New Explorer views, **Operations > Java > OSGi bundles** and **Operations > Java > OSGi services**.
- Updated Explorer view, **Operations > Bundleparts**.
- New CMCI resources, CICSOSGiBundle, CICSOSGiService, CICSOTopologyOSGiBundle, and CICSOTopologyOSGiService.
- New CICSplex SM resource tables, CRESOSGB, CRESOSGS, ERMCOSGB, ERMCOSGS, OSGIBUND, and OSGISERV.
- New **INQUIRE OSGIBUNDLE** and **INQUIRE OSGISERVICE** commands.

Support for updated resource: JVMSERVER

Use the updated JVMSERVER resource to get statistics on the usage of a JVM server. You can view the statistics using any of the following interfaces:

- Updated Explorer view, **Java > JVM servers**.
- Updated CICSplex SM view, **Enterprise Java operations > JVM server**.
- Updated CICSplex SM resource table, JVMSESV.
- Updated **CEMT INQUIRE JVMSERVER** and **INQUIRE JVMSERVER** commands.

Support for updated resource: PROGRAM

Use the new JVMSERVER attribute in the PROGRAM resource to specify the JVM server that contains the OSGi service. Use the JVMCLASS attribute to specify the name of the OSGi service that you want to make available to other applications.

You can define or view the JVMSERVER and JVMCLASS attributes on the PROGRAM resource using any of the following interfaces:

- Updated Explorer Program Definitions view, Programs view, and **Java > JVM status** view.
- Updated CICSplex SM views, **CICS resource definitions > Program definitions** and **Programs**.
- Updated CICSplex SM resource tables, PROGDEF and PROGRAM.
- Updated **CEMT INQUIRE PROGRAM**, **INQUIRE PROGRAM**, and **CREATE PROGRAM** commands.

BUNDLE resource updates

The following interfaces are updated to support the BUNDLE resource:

CICS Explorer

- ☞ Operations view

CICSplex SM

- ☞ CRESOSGB resource table
- ☞ CRESOSGS resource table
- ☞ ERMCOGB resource table
- ☞ ERMCOGS resource table
- ☞ OSGIBUND resource table
- ☞ OSGISERV resource table

The CICS SPI

- ☞ The INQUIRE OSGIBUNDLE command
- ☞ The INQUIRE OSGISERVICE command

JVMSERVER resource updates

The following interfaces are updated to support the JVMSERVER resource:

CICS Explorer

- ☞ Operations view

CICSplex SM

- ☞ The JVM server view
- ☞ JVMSESV resource table

CEMT

- ☞ The INQUIRE JVMSERVER command

The CICS SPI

- ☞ The INQUIRE JVMSERVER command

PROGRAM resource updates

The following interfaces are updated to support the PROGRAM resource:

CICS Explorer

- ☞ Administration view
- ☞ Operations view

CICSplex SM

- ☞ The Program definitions view
- ☞ The Programs view
- ☞ PROGDEF resource table
- ☞ PROGRAM resource table

CEMT

- ☞ The INQUIRE PROGRAM command

The CICS SPI

- ☞ The CREATE PROGRAM command
- ☞ The INQUIRE PROGRAM command

Changes to global user exits

The XRSINDI exit provides support for OSGi bundles.

Support for new resources in global user exit, XRSINDI

The resource management install and discard exit XRSINDI supports the new OSGIBUNLDE resource type in CICS bundles. The **UEPIDTYP** parameter returns a value of UEIDOSGB for an OSGi bundle. The parameters for resource length and name return different information to other resource types:

UEPIDLEN

Address of the length of an individual resource name, as a fullword binary value.

For OSGi bundles, this parameter is the length of the information that uniquely identifies an OSGi bundle in CICS in the **UEPIDNAM** parameter. The maximum possible length is 526 bytes.

UEPIDNAM

Address of a variable-length list containing the names of the individual resources reported by this call.

For OSGi bundles, this parameter contains the information that uniquely identifies an OSGi bundle in CICS. The information is listed in the following order:

1. 8 bytes that contain the JVM server name.
2. A fullword containing the length of the OSGi bundle symbolic name.
3. A fullword containing the length of the OSGi bundle version.
4. A concatenation of the OSGi bundle symbolic name and version as a character string.

For information about the new resource type, see “Changes to resources and attributes” on page 20

Changes to statistics

The CICS statistics for Java programs and JVM servers now provide additional information to manage the performance of Java workloads, including garbage collection and wait times for threads.

Changed statistics types: JVM programs and JVM servers

The resource statistics for JVM programs, mapped by the DFHPGRDS DSECT, now include a field to report if a Java program runs in a JVM server.

The resource statistics for JVM servers, mapped by the DFHSJSDS DSECT, now include the following statistics:

- The total number of system threads
- The total number of tasks that waited for a system thread
- The time that tasks spent waiting for a system thread
- The current number of tasks waiting for a system thread
- The peak number of tasks that waited for a system thread
- The timestamp of when the JVM was created in both GMT and local time
- The status of the JVMSERVER resource
- The size of the initial heap that is allocated to the JVM server
- The maximum size of the heap that can be allocated to the JVM server
- The current heap size that is allocated to the JVM server
- The peak size of the heap
- The heap occupancy (size of the heap after the last garbage collection)
- The garbage collection model that is being used by the JVM
- The number of major garbage collection events
- The total elapsed time spent in major garbage collection
- The total amount of memory that was freed by major garbage collection
- The number of minor garbage collection events
- The total elapsed time spent in minor garbage collection
- The total amount of memory that was freed by minor garbage collection.

The reports produced by the DFHSTUP and DFH0STAT statistics programs show the new statistics.

Changes to the JCICS packaging

The JCICS API is now packaged as a set of OSGi bundles that can run in a JVM server as well as a pooled JVM environment.

The `dfjcics.jar` and `dfjoutput.jar` files that were supplied in previous releases of CICS have been replaced by the following OSGi bundles:

File name	OSGi bundle symbolic name	Description
<code>com.ibm.cics.samples.jar</code>	<code>com.ibm.cics.samples</code>	Samples for redirecting System.out and System.err. Replaces the <code>dfjoutput.jar</code> file.
<code>com.ibm.cics.server.jar</code>	<code>com.ibm.cics.server</code>	JCICS API. Replaces the <code>dfjcics.jar</code> file.

File name	OSGi bundle symbolic name	Description
com.ibm.record.jar	com.ibm.record	The Java API for legacy programs that use IByteBuffer from the Java Record Framework that came with VisualAge. Previously in the dfjcics.jar file.

Changes to sample programs and profiles

The JCICS examples and a new sample JVM profile DFHOSGI support running an OSGi framework in a JVM server.

New examples for OSGi

The existing JCICS examples have been repackaged as OSGi bundles to run in a JVM server environment as well as in a pooled JVM. The JCICS samples are provided in the CICS Explorer SDK to help application developers get started with developing Java applications for CICS. The following OSGi bundles are provided:

File name	OSGi bundle symbolic name	Description
com.ibm.cics.examples.hello	com.ibm.cics.examples.hello	The HelloWorld samples return a simple message when a transaction runs.
com.ibm.cics.examples.jcics	com.ibm.cics.examples.jcics	The program control, TDQ, and TSQ samples return information from programs, a transient data queue, and temporary storage queue when a transaction runs.
com.ibm.cics.examples.web	com.ibm.cics.examples.web	The Web.Sample1 sample returns information to a web browser and write information and error messages to the standard output streams.

New sample profile, DFHOSGI

A new JVM server profile, DFHOSGI, can be used for running Java applications that comply with the OSGi specification in a JVM server. It includes two new options specifically to support OSGi:

OSGI_BUNDLES=middleware_bundles

When you use this option, CICS loads the specified middleware bundles into the OSGi framework. Middleware bundles contain Java classes that implement system functions. For example, middleware OSGi bundles are available for connecting to WebSphere MQ. These OSGi bundles are loaded automatically in the framework when the JVM server is enabled.

OSGI_FRAMEWORK_TIMEOUT={60|number}

When you use this option, CICS waits the specified number of seconds for the OSGi framework to initialize and shut down in the JVM server. If the

framework does not initialize or shut down in time, CICS disables the JVMSERVER resource and terminates the JVM.

Changes to problem determination

CICS provides new information to help you diagnose problems relating to a JVM server.

New messages

The following messages are issued by CICS when problems occur with a JVM server:

- DFHAP1605
- DFHSJ0215
- DFHSJ1007 W
- DFHSJ1008 W

Changed messages

The message DFHAC2016 is now issued if a program requires a JVM server but the named JVM server is not available in the CICS region. The message explanation has been updated.

New abend codes

The following abend codes are issued by CICS when problems occur with a JVM server:

- ASJ7
- ASJS

New trace points

The following new trace points relate to resolving OSGi bundle dependencies during CICS start up:

- RL 0261 to RL 0270

Chapter 3. Developing Java applications in the CICS Explorer SDK

You can now use the CICS Explorer SDK to develop Java applications for any release of CICS. The SDK provides support for OSGi and deploying applications to CICS.

The CICS Explorer SDK is a freely available download for Eclipse-based Integrated Development Environments (IDEs). The SDK provides support for creating new applications or repackaging existing Java applications that comply with the OSGi specification. The SDK also provides support for deploying OSGi bundles to run in CICS. You can create OSGi bundles to run in any supported release of CICS. Because the OSGi bundle is a Java archive, previous releases of CICS ignore the OSGi information and run the Java application in a pooled JVM environment. In this release, you can run OSGi bundles in a JVM server.

The CICS Explorer SDK provides the following benefits to Java developers:

- You can use the target platform feature to develop Java applications that use the correct level of JCICS.
- You can develop new applications that comply with the OSGi specification.
- You can migrate existing Java applications to OSGi.
- You can use the deployment wizards to easily deploy applications to CICS.

The CICS Explorer SDK help provides full details on how to develop and deploy Java applications for CICS.

Chapter 4. Java runtime support for JZOS and J2C copybook importers

You can now reuse existing CICS COBOL and other application programs in Java, by using the copybook importers provided by the JZOS or Rational® J2C tools.

CICS supports copybook importers so that you can use structured data from other programming languages in Java. Supported importers are provided by JZOS tools and by Rational, using the Java EE Connector Architecture (JCA), also known as the J2EE Connector architecture (J2C).

The importers map the data types contained in the source program so that your application can access individual fields in data structures. You can use the JZOS or Rational J2C tools to interact with data to produce a Java class, so that you can pass data between Java and other programs in CICS.

CICS supports Java artifacts from the following importers:

- Data binding beans from the J2C tools in Rational Application Developer (RAD) and Rational Developer for System z®
- Records from the IBM JZOS Batch Toolkit for z/OS SDK

Chapter 5. Support for Java-based SOAP pipelines

The support for web services has changed so that you can create pipelines where the SOAP processing is handled by the Java-based Axis2 technology and runs in a JVM server. The SOAP processing for these pipelines can be offloaded to the IBM System z Application Assist Processor (zAAP).

As a system programmer, you can now create pipelines that use Java SOAP message handlers to perform SOAP processing. These Java message handlers use the Axis2 technology to process SOAP messages in a JVM server. The Axis2 technology supports a range of web service specifications, including WS-Addressing and MTOM/XOP. You can use Axis2 in pipelines that support WS-Addressing and MTOM/XOP. For Java-based service provider pipelines, you can specify the supplied Axis2 application handler in the pipeline configuration file, create provider web services that use Java application handlers that attach to the pipeline using Axis2 MessageContext, or deploy Plain Old Java Objects (POJOs) as provider web services in CICS using the Axis2 style of web service deployment. For Java-based service requester pipelines, the **EXEC CICS INVOKE SERVICE** command is processed using Axis2.

Changes to external interfaces

Some external interfaces have changed to support the enhancements to web services pipelines.

Changes to the pipeline configuration

The pipeline configuration file has new XML elements to support the enhancements to SOAP pipelines and a new container for the Axis2 application handler.

New XML elements and attributes

The following XML elements have been added to the pipeline configuration file:

- A new `<addressing>` element that enables support for WS-Addressing using Axis2. This element contains a `<namespace>` element that you can use to specify the namespace for the WS-Addressing specifications.
- A new `<apphandler_class>` element that specifies the Axis2 application handler for Java-based pipelines.
- A new `<cics_soap_1.1_handler_java>` element that defines the attributes of the SOAP 1.1 message handler program for Java-based SOAP processing using Axis2.
- A new `<cics_soap_1.2_handler_java>` element that defines the attributes of the SOAP 1.2 message handler program for Java-based SOAP processing using Axis2.
- A new `<jvmserver>` element that specifies the name of the JVMSERVER resource.
- A new `<repository>` element that specifies the directory name of the Axis2 repository, which is where the Axis2 configuration files are stored.

New pipeline control container

A new DFHWS-APPHANCLAS container has been added. The container is initialized with the contents of the <apphandler_class> element of the pipeline configuration file.

Changes to other programming interfaces

A new Java interface for Axis2 application handlers is now available.

New Java interface for Axis2 application handlers

The com.ibm.cics.server.pipeline.ApplicationHandler Java interface is now available for Axis2 application handlers. The class specified in the JVMCLASS attribute of the program definition for an Axis2 application handler must implement this interface.

Changes to sample profiles

The sample JVM profile for a JVM server has changed to support running SOAP pipeline processing in a JVM server.

Changed sample profile, DFHJVMAX

The JVM server profile, DFHJVMAX, uses a new option:

JAVA_PIPELINE={YES|NO}

When this option is set to YES, the Axis2 JAR files are added to the class path of the JVM server. The JVM server is configured to process SOAP messages in Java-based SOAP pipelines. This option is set to YES in the profile.

The DFHJVMAX profile also uses a new symbol:

&JVMSERVER;

When you use this symbol, the name of the JVMSERVER resource is substituted at run time. Use this symbol to create unique output or dump files for each JVM server.

The DFHJVMAX profile can also use a new UNIX System Services environment variable:

JVMTRACE={dfhjvmtrc|filename}

When you use this environment variable and switch on level 2 tracing, Java tracing that occurs during the startup, running, and shutdown of a JVM server is written to the specified file. If no value is specified, the trace is written to dfhjvmtrc.

Changes to sample pipeline files

New sample pipeline configuration files for pipelines that support Java are now available.

New sample configuration files

The library /usr/lpp/cicsts/cicsts42/samples/pipelines directory (where /usr/lpp/cicsts/cicsts42 is the default install directory for CICS files on z/OS UNIX) contains two new sample pipeline configuration files for pipelines that support Java. These sample pipeline configuration files can be used as a basis for developing your own.

- The `basicsoap11javaprovider.xml` defines a service provider that uses the `<cics_soap_1.1_handler_java>` element.
- The `basicsoap12javaprovider.xml` defines a service provider that uses the `<cics_soap_1.2_handler_java>` element.

New sample definitions files

Two new sample pipeline definitions files for pipelines that support Java are now available.

- The `EXPIPE03` is the definition of a service provider pipeline.
- The `EXPIPE04` is the definition of a service requester pipeline.

These sample pipeline definition files point to the same pickup directories as the existing sample pipeline definitions `EXPIPE01` and `EXPIPE02`.

Changes to problem determination

CICS provides new information to help you diagnose problems relating to Java-based web services pipelines.

New messages

- New error message `DFHPI0728` reports problems with the repository `Axis2` configuration file defined in the pipeline configuration file.
- New error message `DFHPI0729` reports problems with the CICS Java SOAP handler defined in the pipeline configuration file.
- New error message `DFHPI0734` reports problems with the `JVMSEVER` defined in the pipeline configuration file.

Changed messages

- Changed error message `DFHPI0720` includes elements for pipelines that support Java

New trace points

New trace points that are issued by the pipeline manager domain:

- `PI 2000` to `PI 200F`

Chapter 6. Enhancements to JVM support

All JVMs now run in AMODE(64) instead of AMODE(31), increasing the capacity for running more JVMs in a CICS region.

JVM servers and pooled JVMs use 64-bit storage, significantly reducing the storage constraints in a CICS region for running Java applications. You can therefore reduce the number of CICS regions that run Java to simplify system management and reduce infrastructure costs. You can also use System z Application Assist Processors (zAAPs) to run eligible Java workloads.

CICS uses the IBM 64-bit SDK for z/OS, Java Technology Edition, Version 6.0.1. You must download this version of the SDK to run Java applications in CICS. You can continue to build Java programs using a different version of the SDK.

If you have any programs that use the Java Native Interface (JNI), including other products, you must ensure that these programs can run in a 64-bit environment. For further details see the appropriate Upgrading documentation.

Changes to installation procedures

The default location for Java has changed to support the 64-bit JVM.

The default value for the **JAVADIR** parameter in the installation procedures has changed to `java/J6.0.1_64`. The full path for the default installation directory is now `/usr/lpp/java/J6.0.1_64`.

You must download the IBM 64-bit SDK for z/OS, Java Technology Edition and configure CICS to point to the correct location in z/OS UNIX.

Changes to external interfaces

Some external interfaces now support the 64-bit JVM.

Changes to user-replaceable programs

The user-replaceable programs DFHJVMAT and DFHJVMRO now support 64-bit JVMs only.

Changes to DFHJVMAT

You can no longer use the DFHJVMAT program to override the CICS_HOME option in the JVM profile. The Java launcher uses the value of the **USSHOME** system initialization parameter instead. If you have written your own version, update the program to remove the CICS_HOME option.

Changes to DFHJVMRO

The DFHJVMRO program contains changed options to support the 64-bit Language Environment enclave for pooled JVMs. If you have written your own version, update the program to set the values for 64-bit storage.

Old option	New option
HEAP	HEAP64
LIBHEAP	LIBHEAP64
STACK	STACK64

Changes to sample profiles and programs

The sample JVM profiles and DFHAXRO program have changed to support the 64-bit JVM.

Changed sample JVM profiles

The JVM sample profiles, DFHJVMCD, DFHJVMPR, and DFHJVMAX have changed. The CICS_HOME option has been removed. The JVM launcher uses the value of the **USSHOME** system initialization parameter instead. If you start a JVM that has the CICS_HOME option specified in the profile, CICS issues message DFHSJ0527.

Changed sample program, DFHAXRO

The DFHAXRO program contains changed options to support the 64-bit Language Environment enclave for JVM servers. If you have written your own version, update the program to set the values for 64-bit storage.

Old option	New option
HEAP	HEAP64
LIBHEAP	LIBHEAP64
STACK	STACK64

Changes to problem determination

Messages and abend codes have changed to support the 64-bit JVM.

Changed messages

The following messages have been updated to support the 64-bit JVM:

- DFHSJ0201 to DFHSJ0205
- DFHSJ0911

Changed abend codes

The following abend codes have been updated to remove the CICS_HOME option and use the **USSHOME** system initialization parameter instead.

- ASJ1
- ASJ3
- ASJC

New trace points

The following trace points have been added:

- AP 196A to AP 196D

- SJ 0237 to SJ 0248
- SJ 0D01 to SJ 0D07

Discontinued trace points

The following trace points have been discontinued:

- LE 193A to LE 193F
- LE 1970 to LE 1978

Part 3. Connectivity

CICS Transaction Server for z/OS, Version 4 Release 2 delivers a set of capabilities represented in the following themes:

- Events
- Java
- Connectivity
- Management
- Scalability

The *Connectivity* theme provides an extension to System z IP networking and supports the function shipping of file control, transient data, and temporary storage requests between CICS regions over a TCP/IP network using an IP interconnectivity (IPIC) connection. Potential benefits include improved performance and the ability to simplify network administration when moving from SNA to an IP network.

Chapter 7. Support for web services discovery

The support for Web services is enhanced so that the WSDL documents associated with a Provider mode Web Service are automatically published to the Web.

A convention exists amongst Web service hosting environments that allows the WSDL for a Web Service to be queried by a remote client (typically an Application Developer using a Web browser) using the URI for the Web service suffixed with `?wsdl`. This convention can make it easier to distribute WSDL to interested parties without the need for a formal WSDL repository. This convention is implemented in CICS.

For example, if you have a Web service hosted in CICS and published under the URI:

```
http://www.example.org:1234/example/WebService
```

Then the associated WSDL document could be recovered by requesting the following URI using a Web browser:

```
http://www.example.org:1234/example/WebService?wsdl
```

WSDL documents for service providers can now be published for discovery using URIMAP resources. When you install each PIPELINE resource, CICS scans the directory specified in the WSDIR attribute of the PIPELINE resource (the pickup directory). If this directory contains either a WSDL archive file or WSDL document, a second URIMAP resource is installed. This new URIMAP resource provides CICS with the information to associate the WSDL archive file or a WSDL document with a specific URI so that external requesters can use the URI to discover the WSDL archive file or WSDL document. This URI has the same path as the URI associated with the WEBSERVICE with the suffix `?wsdl` appended. The WSDL archive file can contain one or more WSDL documents. If the pickup directory contains a WSDL archive file and a WSDL document, the URI returns only the WSDL archive. The archive file format that is supported is `.zip` file type. It is also possible to discover the WSDL archive file or WSDL document using SPI and CEMT. The WSDL document in a WSDL archive file can be used for SOAP message validation.

For more information, see CICS resources for web services

Changes to external interfaces

Some external interfaces now support web services discovery.

Changes to resources and attributes

ARCHIVEFILE is a new WEBSERVICE attribute that specifies the 1-255 character fully-qualified file name of an archive that contains one or more WSDL files. The supported format for the archive is `.zip`.

Support for updated resource: WEBSERVICE

You can use the updated WEBSERVICE resource to archive one or more WSDL files in `.zip` file.

The updated WEBSERVICE resource has the following support:

- Updated WEBSERV view: the new ARCHIVEFILE attribute to displays the name and location of an archive file that contains one or more WSDL files associated with the web service, **CICS operations views > TCP/IP service operations views > Web services**.
- Updated WEBSERV resource table: use the new ARCHIVEFILE attribute to define the name and location of an archive file that contains one or more WSDL files associated with the web service.
- Updated INQUIRE WEBSERVICE command: use the new ARCHIVEFILE option to return the name and location of an archive file that contains one or more WSDL files associated with the web service.
- Updated CEMT INQUIRE WEBSERVICE command: use the new ARCHIVEFILE option to display the name and location of an archive file that contains one or more WSDL files associated with the web service.

WEBSERVICE resource updates

The following interfaces are updated to support the WEBSERVICE resource:

CICSplex SM

-  [The Web services - WEBSERV view](#)
-  [The WEBSERV Resource Table](#)

CEMT

-  [The CEMT INQUIRE WEBSERVICE command](#)

The CICS SPI

-  [The INQUIRE WEBSERVICE command](#)

Changes to statistics

The CICS statistics for web services now provides information about the WSDL archive file.

New statistics type, ARCHIVEFILE

The CICS web services resource statistics now includes the ARCHIVEFILE resource. This specifies the file name of the archive file containing one or more Web service description (WSDL) file associated with the WEBSERVICE resource.

Changes to problem determination

CICS provides new information to help you diagnose problems relating to web services discovery.

New messages

- New error message DFHPI0905 reports problems with the length of the WSDL name.
- New error message DFHPI0906 reports problems with the length of the WSDL discovery URI.
- New error message DFHPI0603 reports problems with a SOAP message handlers incorrectly receiving a HTTP GET.

Chapter 8. HTTP connection management

You can now set up connection throttling, where you can limit the number of persistent HTTP connections from web clients that a CICS region accepts. You can also set up connection pooling, where CICS applications can reuse client HTTP connections that have already been opened. Connection pooling is available for CICS web support applications and web services applications that use the HTTP and HTTPS protocols, and for the HTTP EP adapter for CICS event processing.

CICS as an HTTP server: Connection throttling for inbound HTTP connections

If multiple web clients set up long-lived persistent connections to CICS as an HTTP server and use the connections heavily, it is possible for a CICS region handling the connections to become overloaded and experience performance problems. If you experience this problem, you can set up connection throttling to make excess web clients connect to other CICS regions that share the port and provide the same service.

With connection throttling, you can set a limit on the number of persistent HTTP connections that a CICS region accepts for a particular port. If the limit is reached and further web clients send requests, CICS sends Connection: close headers with each response to require the new clients to close their connection. The web clients that already have persistent connections to the CICS region can maintain their persistent connections. When the new clients reconnect, if they connect to another CICS region that shares the port and has not reached its limit, they can maintain a persistent connection there. The CICS region that has reached its limit begins to accept new persistent connections again when the web clients that have persistent connections to it close their connections.

CICS as an HTTP client: Connection pooling for outbound HTTP connections

By default, CICS closes a client HTTP connection after a CICS application has finished using the connection, or a service requester application has made a web service request and received a response, or the HTTP EP adapter has emitted a business event. When you set up connection pooling, instead of closing the connection CICS can place the connection in a pool in a dormant state. The dormant connection can be reused by the same application or by another application that connects to the same host and port. Connection pooling can provide performance benefits where multiple invocations of CICS web support applications, web services applications, or the HTTP EP adapter make connection requests to a particular host and port, or where a web services application makes multiple requests and responses.

To set up connection pooling, you specify the SOCKETCLOSE attribute on a URIMAP resource definition for the client HTTP connection. For a client HTTP connection to be pooled, the CICS application program must specify the URIMAP resource on the INVOKE SERVICE or WEB OPEN command, and CICS web support applications must issue the WEB CLOSE command to explicitly end their use of the connection. If you do not issue the WEB CLOSE command, CICS closes the connection at end of task. The closed connection cannot be placed in a pool. To enable connection pooling, your application must issue the WEB CLOSE command.

A connection cannot be pooled if the server has requested CICS to close the connection, or if the application program has requested the server to close the connection by specifying the `CLOSESTATUS(CLOSE)` option on the **WEB SEND** or **WEB CONVERSE** command. CICS also checks the state of an open connection before placing it in the pool; connections are not pooled if they are found or suspected to be in a poor state, for example, if the last HTTP response code was not 200 OK.

Changes to external interfaces

Some external interfaces now support HTTP connection management.

Changes to the CICS application programming interface

To activate connection pooling, applications that use the **INVOKE SERVICE** or **WEB OPEN** commands to open a client HTTP connection must specify a URIMAP resource with the `SOCKETCLOSE` attribute set. CICS web support and web services application programs, and the HTTP EP adapter for event processing, can use connection pooling.

- For CICS web support applications, you can specify a URIMAP resource on the **WEB OPEN** command to provide the URL for the connection. If this URIMAP resource has the `SOCKETCLOSE` attribute set, connection pooling is activated for the connection. If you code the URL directly in the application, CICS does not access a URIMAP resource and connection pooling is not activated.
- For web services application programs that are service requesters, you can specify a URIMAP resource on the **INVOKE SERVICE** command to provide the URI of the web service. If this URIMAP resource has the `SOCKETCLOSE` attribute set, connection pooling is activated for the connection. If you code the URL directly in the application, or if you use a URI from the web service description, CICS does not access a URIMAP resource and connection pooling is not activated.
- For the HTTP EP adapter, you specify a URIMAP resource in your event binding. If this URIMAP resource has the `SOCKETCLOSE` attribute set, connection pooling is activated for the connection.

For CICS web support applications only, two further steps are required to ensure that connection pooling takes place:

- Ensure that the application does not use the option `CLOSESTATUS(CLOSE)` on any of the **WEB SEND** or **WEB CONVERSE** commands that it issues.
- Ensure that the application issues the **WEB CLOSE** command when it has completed its use of the client HTTP connection.

If your applications already use URIMAP resources when opening outbound connections, and your CICS web support applications issue the **WEB CLOSE** command and do not use the option `CLOSESTATUS(CLOSE)` on any commands, you do not need to make any changes to your applications to implement connection pooling. You can specify the `SOCKETCLOSE` attribute on the URIMAP resources that are already used by your CICS web support and web services applications and by the HTTP EP adapter. The session token used by CICS web support applications does not persist on a pooled connection, so your applications reuse a pooled connection in exactly the same way as they use a new connection.

For more information about implementing connection pooling for connections opened by your applications, see *Implementing connection pooling for client HTTP connections in Upgrading*.

Changes to resources and attributes

The URIMAP resource now supports connection pooling for client (outbound) HTTP connections. The TCPIPSERVICE resource now supports connection throttling for inbound HTTP connections.

Support for updated resource: URIMAP

You can use the updated URIMAP resource to specify the SOCKETCLOSE attribute to enable connection pooling for client HTTP connections.

The updated URIMAP resource has the following support:

- Updated Explorer SM Operations URI Maps view: use the new SOCKETCLOSE attribute to view the expiry limit for dormant client connections, and the SOCKPOOLSIZE attribute to see how many connections are currently held in a dormant state in the pool.
- Updated Explorer SM Definitions URI Mapping Definitions view: use the new SOCKETCLOSE attribute to enable connection pooling for client connections.
- Updated CICSplex SM Operations URI maps - URIMAP view: use the new SOCKETCLOSE field to view the expiry limit for dormant client connections, and the SOCKPOOLSIZE and SOCKPLSZPEAK fields to see the current and peak number of connections held in a dormant state in the pool. Use the SOCKRECLAIMD and SOCKTIMEDOUT fields to see the number of connections that CICS closed in the pool or that timed out without being reused.
- Updated CICSplex SM Administration URI mapping definitions - URIMPDEF view: use the new SOCKETCLOSE field to enable connection pooling for client connections.
- Updated CICSplex SM URIMAP and URIMPDEF resource tables: use the new SOCKETCLOSE attribute to enable connection pooling for client connections, and the SOCKPOOLSIZE attribute to see how many connections are currently held in a dormant state in the pool.
- Updated CREATE URIMAP command: use the new SOCKETCLOSE attribute to enable connection pooling for client connections.
- Updated INQUIRE URIMAP command: use the new SOCKETCLOSE attribute to view the expiry limit for dormant client connections, and the SOCKPOOLSIZE attribute to see how many connections are currently held in a dormant state in the pool.

Support for updated resource: TCPIPSERVICE

You can use the updated TCPIPSERVICE resource to specify the MAXPERSIST attribute to enable connection throttling for inbound HTTP connections.

The updated TCPIPSERVICE resource has the following support:

- Updated Explorer SM Operations TCP/IP Services view: use the new MAXPERSIST field to set the limit for the number of persistent connections from Web clients that the CICS region allows for this port at any one time.
- Updated Explorer SM Definitions TCP/IP Service Definitions view: use the new MAXPERSIST field to set the limit for the number of persistent connections.
- Updated CICSplex SM Operations TCP/IP services- TCPIPS view: use the new MAXPERSIST field to set the limit for the number of persistent connections, and the NONPERSIST field to see how many Web clients were not given a persistent connection.

- Updated CICSplex SM Administration TCP/IP service definitions - TCPDEF view: use the new MAXPERSIST attribute to set the limit for the number of persistent connections.
- Updated CICSplex SM TCPIPS and TCPDEF resource tables: use the new MAXPERSIST attribute to set the limit for the number of persistent connections.
- Updated CREATE TCPIPSERVICE command: use the new MAXPERSIST option to set the limit for the number of persistent connections.
- Updated INQUIRE TCPIPSERVICE command: use the new MAXPERSIST option to view the limit for the number of persistent connections.

URIMAP resource updates

The following interfaces are updated to support the URIMAP resource:

CICS Explorer

- ☞ Definition views
- ☞ Operation view

CICSplex SM

- ☞ URI maps - URIMAP view
- ☞ URI mapping definitions - URIMPDEF view
- ☞ URIMAP resource table
- ☞ URIMPDEF resource table

CEMT

- ☞ INQUIRE URIMAP command

The CICS SPI

- ☞ CREATE URIMAP command
- ☞ INQUIRE URIMAP command

TCPIPSERVICE resource updates

The following interfaces are updated to support the TCPIPSERVICE resource:

CICS Explorer

- ☞ Administration view
- ☞ Operation view

CICSplex SM

- ☞ TCP/IP services - TCPIPS view
- ☞ TCP/IP service definitions - TCPDEF view
- ☞ TCPIPS resource table
- ☞ TCPDEF resource table

CEMT

➤ INQUIRE TCPIPSERVICE command

The CICS SPI

➤ CREATE TCPIPSERVICE command

➤ INQUIRE TCPIPSERVICE command

Changes to statistics

New statistics are added to the resource statistics for URIMAP resources and TCPIPSERVICE resources to provide information about connection pooling and connection throttling.

Changed statistics types

The resource statistics for URIMAP resources, in DFHWBRDS, now include the following statistics:

- The SOCKETCLOSE setting for the URIMAP resource
- The current and peak number of pooled connections in the pool for this URIMAP resource
- The number of dormant connections that CICS reclaimed from the pool because the CICS region had reached the MAXSOCKETS limit
- The number of dormant connections that were not reused and expired while they were in the pool

The resource statistics for TCPIPSERVICE resources, in DFHSORDS, now include the following statistics:

- The MAXPERSIST setting for the TCPIPSERVICE resource (Maximum Persistent Connections field)
- The number of connections where CICS did not permit the Web client to have a persistent connection (Non-Persistent Connections field)

The reports produced by the DFHSTUP and DFH0STAT statistics programs are updated to show the new statistics.

Changes to problem determination

New trace points are provided for connection pooling.

New trace points

Trace points in the range SO 1200 - SO 1209 are for activity in the pools that CICS uses to hold opened client HTTP connections in a dormant state.

Chapter 9. IPIC support for function shipping

IP interconnectivity (IPIC) using TCP/IP now supports function shipping of file control, transient data, and temporary storage requests between CICS TS 4.2 or later regions. Before the introduction of IPIC support of function shipping, these requests could only be accessed over an MRO or ISC connection.

Function shipping file control, transient data, and temporary storage requests with an IPIC connection provides CICS application programs with the ability to run without regard to the location of the requested resources, and uses threadsafe mirror transactions for potentially greater throughput between CICS TS 4.2 regions.

File control requests that are function shipped using IPIC connectivity provide threadsafe file control with significant potential throughput improvements over LU6.2 in CICS regions with multiple processors available.

Transient data requests that are function shipped using IPIC connectivity provide access to an intrapartition or extrapartition transient data queue (TDQ) that is remote on another CICS region.

Temporary storage requests that are function shipped using IPIC connectivity are threadsafe and no longer need to switch to QR before being function shipped.

The MAIN and AUXILIARY options on the WRITEQ TS command have been enhanced with IPIC support for function-shipped requests between CICS TS 4.2 regions or later. Previously, the MAIN and AUXILIARY options were only supported by using the multiregion operation (MRO) facility. APPC does not support the MAIN and AUXILIARY options. Temporary storage queues (TSQs) created as a result of function shipping WRITEQ TS using APPC are stored in auxiliary storage.

The CICS-supplied mirror program DFHMIRS, which is used by all mirror transactions, is now defined as threadsafe. For IPIC connections only, CICS runs the mirror program on an L8 open TCB whenever possible. For threadsafe applications that function ship commands to other CICS regions using IPIC, the resulting reduction in TCB switching improves the performance of the application compared to other intercommunication methods. To gain the performance improvement, you must specify the system initialization parameter **FCQRONLY=NO** in the file-owning region.

The **EXEC CICS LINK** command is now threadsafe when it is used to link to a program in a remote CICS region over an IPIC connection. For other types of connections to remote CICS regions, the command remains nonthreadsafe.

A new option MIRRORLIFE has been added to the IPCONN attributes for function-shipped file control, transient data, and temporary storage requests using an IPIC connection. MIRRORLIFE improves efficiency and provides performance benefits by specifying the lifetime of mirror tasks and the amount of time a session is held.

Terminology

Terminology used to explain support of IP interconnectivity (IPIC) connectivity.

capability exchange. A series of messages that pass between two CICS regions to establish if they can communicate using IP interconnectivity (IPIC). The capability exchange determines the security controls that are applied to the connection, the number of sessions to be made available, and resynchronizes any outstanding work if the connection has previously failed.

Changes to external interfaces

Some external interfaces now support function shipping using IPIC.

Changes to system initialization parameters

Review the system initialization parameter **FCQRONLY** for file-owning regions that use IP interconnectivity (IPIC) connections over TCP/IP for function shipping.

Changed system initialization parameter, **FCQRONLY**

Before CICS TS for z/OS, Version 4.2, you specified **FCQRONLY=YES** to improve the performance of all file-owning regions. However, for file-owning regions in CICS TS for z/OS, Version 4.2, choose an appropriate setting for **FCQRONLY**:

- For FORs where the connections to that region are primarily MRO or ISC connections, these requests run on the QR TCB, and CICS runs the mirror program primarily on the QR TCB. Specify **FCQRONLY=YES** so that all file control requests are processed on the QR TCB. This setting improves performance by avoiding locking, which is unnecessary when all file control requests run on the same TCB.
- For FORs where the connections to that region are primarily IPIC connections, these requests run on open TCBS, and CICS runs the mirror program on an L8 open TCB whenever possible. Specify **FCQRONLY=NO** so that file control requests do not switch to the QR TCB to be processed. This setting improves performance by multi-threading file control requests.

Changes to resources and attributes

You can use the new option **MIRRORLIFE** that has been added to the **IPCONN** attributes for function-shipping support.

Support for updated resource: **IPCONN**

A new option **MIRRORLIFE** has been added to **IPCONN** attributes for function-shipped file control, transient data, and temporary storage requests using an IPIC connection. **MIRRORLIFE** improves efficiency and provides performance benefits by specifying the lifetime of mirror tasks and the amount of time a session is held. It only takes effect when specified on the **IPCONN** on the resource-owning region. For more information about **MIRRORLIFE**, see Long-running mirror tasks for IPIC.

Changes to global user exits

The file control domain exits **XFCFRIN** and **XFCFROUT**, the temporary storage domain exits **XTSQRIN**, **XTSQROUT**, **XTSPTIN**, **XTSPTOUT**, and the managing IPIC intersystem queues exit **XISQUE** support threadsafe mirror transactions over an IPIC connection.

Changed global user exits

The **UEPTERM** parameter is a zero value for temporary storage requests that have been function shipped over an IPIC connection. To use IPIC connections for

temporary storage requests, ensure that XTSQRIN, XTSQRROUT, XTSPTIN, and XTSPTOUT check that the UEPTERM parameter is a non-zero value before trying to use it as an address.

XTSQRIN, XTSQRROUT, XTSPTIN, and XTSPTOUT must be coded to threadsafe standards and declared threadsafe to get the benefits of threadsafe remote temporary storage queue support using an IPIC connection.

The UEPTERM parameter is a zero value for file control requests that have been function shipped over an IPIC connection. To use IPIC connections for function shipping file control requests, ensure that XFCFRIN and XFCFROUT check that the UEPTERM parameter is a non-zero value before trying to use it as an address.

XFCFRIN and XFCFROUT must be coded to threadsafe standards and declared threadsafe to get the benefits of threadsafe remote file support using an IPIC connection.

XISQUE controls the requests and commands that are queued on an IPIC connection. XISQUE must be coded to threadsafe standards and declared threadsafe to get the benefits of threadsafe distributed program link (DPL) support, and threadsafe function shipping file control and temporary storage support, using an IPIC connection.

Changes to problem determination

CICS provides new information to help you diagnose problems relating to function shipping.

Changed messages

The following error messages have changed for requests that have been function shipped or routed over an IPCONN:

- DFHFC0150 - DFHFC0152
- DFHFC0157
- DFHFC0164 - DFHFC0169
- DFHFC0174
- DFHFC0175
- DFHFC0177
- DFHFC0179
- DFHFC0300 - DFHFC0303
- DFHFC0308 - DFHFC0311
- DFHFC3010
- DFHFC7130
- DFHIS1035

The termid displayed in messages might now refer to an IPIC session. IPIC sessions are defined with a prefix of a slash (/). For more information about the standard prefixes for termids, see EIB fields and EIBTRMID.

New abend code

The following abend codes are issued by the mirror program DFHMIRS in the event of a change mode failure.

- AITN

Changes to statistics

The statistics utility program, DFHSTUP, now supports function shipping of file control commands and transient data commands, and serviceability for unsupported functions.

New statistics types

The following statistics have been added to DFHSTUP:

ISR_FS_FC_REQUESTS

The number of file control requests for function shipping on this connection.

ISR_FS_FC_BYTES_SENT

The number of bytes sent by file control requests.

ISR_FS_FC_BYTES_RECEIVED

The number of bytes received by file control requests.

ISR_MIRRORLIFE

The minimum lifetime of the mirror task for function-shipped requests received by this region.

ISR_FS_TD_REQUESTS

The number of transient data requests for function shipping on this connection.

ISR_FS_TD_BYTES_SENT

The number of bytes sent by transient data requests.

ISR_FS_TD_BYTES_RECEIVED

The number of bytes received by transient data requests.

ISR_FS_TS_REQUESTS

The number of temporary storage requests for function shipping on this connection.

ISR_FS_TS_BYTES_SENT

The number of bytes sent by temporary storage requests.

ISR_FS_TS_BYTES_RECEIVED

The number of bytes received by temporary storage requests.

ISR_UNSUPPORTED_REQUESTS

The number of attempts to route requests for unsupported function across this connection.

Chapter 10. Recovery support for WebSphere MQ

CICS provides a new group resynchronization option in the MQCONN resource definition to support WebSphere MQ recovery. You can configure CICS and WebSphere MQ to recover faster and more autonomously without compromising integrity following the failure of a CICS region or a WebSphere MQ queue manager, by using the new option.

You can use the new RESYNCMEMBER(GROUPRESYNC) option on the MQCONN resource definition to connect CICS to any WebSphere MQ queue manager in the queue-sharing group regardless of any outstanding indoubt units of work. A queue manager is eligible for connection to a CICS region if it is currently active on the same LPAR as the CICS region. The main use of the GROUPRESYNC option are with shared queues, but it can be used with private queues. Messages on shared queues are resolved immediately, but resolution of private queues occurs when the owning queue manager restarts. If using private queues, it can be convenient to define the queues with QSGDISP(GROUP). QSGDISP(GROUP) ensures that an instance of the private queue exists on every manager in the queue sharing group, so that in the event of failure and reconnection to a different queue manager, the application can continue using the new queue instance. The application would need to be able to tolerate using a new queue instance.

During startup of the CICS-WebSphere MQ adapter, WebSphere MQ selects the queue manager that CICS connects to and CICS resolves any indoubt units of work on behalf of all eligible queue managers in the queue-sharing group. This function is called *group unit of recovery*. You can use the RESYNCMEMBER(GROUPRESYNC) option only when you are running WebSphere MQ 7.1, or higher, that supports group unit of recovery for CICS, and when the GROUPPUR attribute is enabled in the WebSphere MQ queue managers.

You can use the RESYNCMEMBER(GROUPRESYNC) option to configure faster recovery after a WebSphere MQ queue manager failure by running a minimum of two queue managers in the same queue-sharing group. If the WebSphere MQ queue manager that CICS is connected to fails, CICS immediately reconnects to the other WebSphere MQ queue manager in the queue-sharing group and does not wait for the failed WebSphere MQ queue manager to be restarted.

With the GROUPRESYNC option, you also have more flexible restart options for CICS following a CICS region failure. The failed CICS region can be restarted in any LPAR where a WebSphere MQ queue manager is running in the same queue-sharing group as the queue manager that CICS was previously connected to.

An MQCONN resource definition can be installed with a value of RESYNCMEMBER(GROUPRESYNC), or you can change the RESYNCMEMBER setting to GROUPRESYNC in an installed MQCONN definition. The GROUPRESYNC setting takes effect only when CICS next initiates a connection to WebSphere MQ. If CICS is already connected to WebSphere MQ, or it is not connected, the GROUPRESYNC value is installed but is not used at the time.

Changes to external interfaces

Some external interfaces now support recovery support for WebSphere MQ.

Changes to resources and attributes

You can use the new RESYNCMEMBER(GROUPRESYNC) setting to connect CICS to any member of the WebSphere MQ queue-sharing group and have indoubt units of work resolved.

Support for updated resource: MQCONN

You can use the updated MQCONN resource to specify the new RESYNCMEMBER attribute value GROUPRESYNC. GROUPRESYNC means CICS can recover much faster from a WebSphere MQ failure if the WebSphere MQ queue manager supports group unit of recovery.

The updated MQCONN resource has the following support:

- Updated CICS Explorer **Operations > WebSphere MQ > MQ Connections** view: use the RESYNCMEMBER attribute to view the resynchronization options for WebSphere MQ queue-sharing groups.
- Updated CICS Explorer **Definitions > WebSphere MQ Connections Definitions** view: use the Resyncmember attribute to set the resynchronization options for WebSphere MQ queue-sharing groups.
- Updated CICSplex SM views **Administration > CICS resource definitions > WebSphere MQ Connections definitions - MQCONDEF**: use the RESYNCMEMBER attribute to set the resynchronization options for WebSphere MQ queue-sharing groups.
- Updated CICSplex SM views **CICS operations > DB2, DBCTL and WebSphere MQ operations views > WebSphere MQ connections - MQCON**: use the RESYNCMEMBER attribute to view the resynchronization options for WebSphere MQ queue-sharing groups.
- Updated CICSplex SM tables MQCON and MQCONDEF: use the RESYNCMEMBER attribute value to view the resynchronization options for WebSphere MQ queue-sharing groups.
- Updated CREATE MQCONN, INQUIRE MQCONN, and SET MQCONN commands: use the RESYNCMEMBER attribute to set or view the new option GROUPRESYNC.

MQCONN resource updates

The following interfaces are updated to support the MQCONN resource:

CICS Explorer

 Definition views

 Operations view

CICSplex SM

 WebSphere MQ connections - MQCON

 WebSphere MQ Connections definitions - MQCONDEF

 MQCON Resource Table

 MQCONDEF Resource Table

CEMT

 CEMT INQUIRE MQCONN

☞ CEMT SET MQCONN

The CICS SPI

☞ CREATE MQCONN

☞ INQUIRE MQCONN

☞ SET MQCONN

Changes to statistics

The WebSphere MQ connection statistics for MQCONN resources show whether recovery support is enabled.

Changed statistics types

The resource statistics for the WebSphere MQ connection now include the following value on the statistics field MQG_RESYNCMEMBER:

GROUPRESYNC

CICS connects to any member of the queue-sharing group. The queue manager is chosen by WebSphere MQ and it asks CICS to resolve indoubt units of work on behalf of all eligible queue managers in the queue-sharing group. This function is called group unit of recovery.

Changes to problem determination

CICS provides new information to help you diagnose problems relating to recovery support for WebSphere MQ.

New messages

The following new error messages provide support for the GROUPRESYNC option.

- DFHMQ2065 and DFHMQ2066 report issues when the RESYNCMEMBER option on the MQCONN definition has been changed and CICS has outstanding UOWs for the previous connection.

Changed messages

The following error message has changed to provide diagnostics information for the GROUPRESYNC option.

- DFHMQ0309 is issued if RESYNCMEMBER(GROUPRESYNC) is specified and group unit of recovery for CICS is not supported or not enabled on the WebSphere MQ queue manager.

Chapter 11. Enhancements to CICS DB2 thread reuse

CICS provides new facilities for you to check and limit the number of times a thread can be reused. When a thread reaches its reuse limit CICS terminates it to free up DB2 resources. Long-running CICS DB2 threads can cause resource issues in DB2 particularly in storage. You can now set a reuse limit on the DB2CONN definition to specify this reuse limit.

The new REUSELIMIT parameter of the DB2CONN resource definition specifies the maximum number of times a thread can be reused before it is terminated. This limit applies to all DB2 threads whether they are protected or unprotected.

A value of 0 for REUSELIMIT means that no limit is placed on the number of times a thread can be reused; this is the same behavior as in previous releases. However, in CICS TS release 4.2 the default has been set to a value of 1000. This change has been implemented to reduce the likelihood of DB2 storage constraint issues, while having a negligible effect on the performance of CICS applications.

You can now use the existing PURGECYCLE parameter on the DB2CONN resource definition to specify a new lower value of 5 seconds. The default remains the same: 30 seconds. The parameter controls how long protected threads are allowed to stay dormant before being reused or terminated.

Changes to external interfaces

Some external interfaces now support the enhancements for CICS DB2 thread reuse.

Changes to resources and attributes

You can use the updated DB2CONN resource to limit the number of times a CICS DB2 thread can be reused.

Support for updated resource: DB2CONN

You can use the new REUSELIMIT attribute in the DB2CONN resource to set the maximum number of times a thread can be reused before being terminated. The INQUIRE DB2CONN, CEMT INQUIRE DB2CONN, SET DB2CONN and CEMT SET DB2CONN commands let you query and change how threads are reused.

The updated DB2CONN resource has the following support:

- Updated CICSplex SM views **DB2, DBCTL and WebSphere MQ operations views > Connections - DB2CONN, Entries - DB2ENTRY**: use the new fields to view or set the maximum limit for thread reuse.
- Updated CICSplex SM view **CICS resource definitions > DB2 connection definitions - DB2CDEF**: use the new REUSELIMIT attribute to view or set the maximum limit for thread reuse.
- Updated CICSplex SM tables DB2CONN: use the new REUSELIMIT attribute to view or set the maximum limit for thread reuse.
- Updated INQUIRE DB2CONN, CEMT INQUIRE DB2CONN, SET DB2CONN, CEMT SET DB2CONN, and CREATE DB2CONN commands: use the new REUSELIMIT option to view or set the maximum number of times a thread can be reused.

DB2CONN resource updates

The following interfaces are updated to support the DB2CONN resource:

CICS Explorer

- ☞ Administration view
- ☞ Operations view

CICSplex SM

- ☞ Connections - DB2CONN in the CICSplex SM Operations Views Reference
- ☞ Entries - DB2ENTRY in the CICSplex SM Operations Views Reference
- ☞ DB2 connection definitions - DB2CDEF in CPSM Administration Views

CEMT

- ☞ CEMT INQUIRE DB2CONN in CICS Supplied Transactions
- ☞ CEMT SET DB2CONN in CICS Supplied Transactions

The CICS SPI

- ☞ INQUIRE DB2CONN in CICS System Programming Reference
- ☞ SET DB2CONN in CICS System Programming Reference
- ☞ CREATE DB2CONN in CICS System Programming Reference

Changes to statistics

The CICS DB2 statistics provide additional information about the value of REUSELIMIT.

Changed statistics types

The CICS DB2 global statistics now include the following information:

- The maximum number of times a thread can be reused before being terminated.
- The number of times the reuse limit has been reached by a pool thread.

The CICS DB2 resource statistics now include the following information:

- The number of times the reuse limit has been reached by a thread for the DB2ENTRY.

Chapter 12. Enhancements to Atom feeds

The support for Atom feeds has changed so that you can create Atom feeds in bundles using the CICS Explorer. You can also reuse existing XMLTRANSFORM resources for multiple Atom feeds.

You can now create and manage Atom feeds using the bundle support in CICS. As an application developer, you can create a bundle that contains the XML binding in z/OS UNIX. Using the CICS event binding editor, you can request that events are emitted when Atom support accesses files and temporary storage queues in CICS. As a systems programmer, you can use the CICS Explorer to create an Atom configuration file in a bundle project and deploy the bundle to a CICS region. When you install the BUNDLE resource, CICS creates the ATOMSERVICE, XMLTRANSFORM, and URIMAP resources for you. You can also choose to reuse an existing XMLTRANSFORM resource if one XML binding can support multiple Atom feeds.

You can enable and disable the BUNDLE resource to control the availability of the Atom feed in the CICS region. The BUNDLE manages the life cycle of the other resources for you.

Changes to external interfaces

Some external interfaces have changed to support the enhancements to Atom feeds.

Changes to the Atom configuration file

The Atom configuration file has new and changed XML elements to support the enhancements to Atom feeds.

New XML elements and attributes

The following XML elements and attributes have been added to the Atom configuration file:

- A new `<cics:urimap>` element. Use this element if you want to dynamically create a URIMAP resource for the Atom feed.
- A new optional attribute `xmltransform` is allowed on the `<cics:bind>` element. Provide a value for this attribute to name the XMLTRANSFORM resource that points to the correct XML binding for the Atom feed. This XMLTRANSFORM can be in created by another bundle or in the same bundle as the Atom feed.
- A new mandatory attribute `version` is required on the `<cics:atomservice>` element. You must set a value of 2 if you want to use the Atom feed improvements in this release. The default value is 1.

Changed XML elements and attributes

The root attribute on the `<cics:bind>` element is now optional instead of required.

Changes to resources and attributes

The ATOMSERVICE resource can be dynamically generated by a BUNDLE resource. The resource signature is updated to show the BUNDLE resource and additional information is available when you inquire on ATOMSERVICE resources.

Support for updated resource: ATOMSERVICE

If you dynamically create the URIMAP and XMLTRANSFORM resource, these resources are returned when you inquire on the associated ATOMSERVICE. If the ATOMTYPE attribute is set to CATEGORY or SERVICE, the RESOURCETYPE attribute returns a value of NOTAPPLIC. The resource signature is updated: The INSTALLAGENT returns a value of BUNDLE when the ATOMSERVICE resource is dynamically created by a BUNDLE resource.

You can view additional information about ATOMSERVICE resources using any of the following interfaces:

- Updated Explorer Atom Services view.
- Updated CICSplex SM view, **CICS resource definitions > Atom Services**
- Updated CICSplex SM resource table, ATOMSERV.
- Updated **CEMT INQUIRE ATOMSERVICE** and **INQUIRE ATOMSERVICE** commands.

Support for updated resource: URIMAP

If you dynamically create the URIMAP resource for an Atom feed, the DEFINESOURCE field in the resource signature returns the name of the ATOMSERVICE resource.

You can view the value of the DEFINESOURCE field on a URIMAP resource using any of the following interfaces:

- Updated Explorer URI Maps view.
- Updated CICSplex SM views, **CICS resource definitions > URI maps**
- Updated CICSplex SM resource table, URIMAP
- Updated **CEMT INQUIRE URIMAP** and **INQUIRE URIMAP** commands.

ATOMSERVICE and URIMAP resource updates

The following interfaces are updated to support the ATOMSERVICE and URIMAP resources:

CICS Explorer

 Operations view

CICSplex SM

 The Atom services view

 The URI maps view

 ATOMSERV resource table

 URIMAP resource table

CEMT

 The INQUIRE ATOMSERVICE command

➤ The INQUIRE URIMAP command

The CICS SPI

➤ The INQUIRE ATOMSERVICE command

➤ The INQUIRE URIMAP command

Changes to sample programs

The sample Atom feeds are now available in bundle directories in z/OS UNIX.

New sample programs

The *USSHOME/samples/web2.0/atom/* in z/OS UNIX has a new directory called *xmltransformBundles*. This directory contains two bundle directories that you can use to try out Atom feeds in CICS.

- The *tsqueueRecordsAsBinary* directory is a bundle that you can use to create a sample Atom feed that reads binary data from a temporary storage queue.
- The *tsqueueRecordsasText1047* directory is a bundle that you can use to create a sample Atom feed that reads text data from a temporary storage queue.

The *xmltransformBundles* also contains the following files:

- A *RECORD.cobol* file that describes the format of the temporary storage queue.
- Two JCL files, *TSQBIN.jcl* and *TSQTEXT.jcl*, that describe how to generate an XML binding from the COBOL.

Changes to problem determination

CICS provides new information to help you diagnose problems relating to creating Atom feeds using a bundle.

New messages

CICS issues the following new messages to report warnings and errors with installing an Atom feed using a BUNDLE resource:

- DFHW20134
- DFHW20135
- DFHW20136
- DFHW20137

Changed messages

The message DFHAM4943 E is issued if CICS is unable to install the ATOMSERVICE resource because the dynamically generated URIMAP is invalid.

Part 4. Management

CICS Transaction Server for z/OS, Version 4 Release 2 delivers a set of capabilities represented in the following themes:

- Events
- Java
- Connectivity
- Management
- Scalability

The *Management* theme provides enhancements to simplify the management of CICS, including a new transaction tracking facility, improvements to workload management, support for password phrases to enhance log-on security, enhancements to the CICS Explorer, and the ability to view system initialization parameters.

Chapter 13. Enhancements to transaction tracking

Enhancements to transaction tracking support the CICS strategy of simplifying system operation tasks and providing a standard framework for the tracking and resolution of interrelated CICS transactions across the CICSplex.

Transaction tracking is enhanced with the following support:

- Previous hop data with origin data over an IPIC or MRO connection.
Previous hop data transmitted across an IPIC or MRO connection between CICS TS 4.2 or later regions, including all user-initiated tasks, provides a transaction-tracking solution that makes it easier to trace back the path of an application and use historical records to analyze patterns of work. In previous releases of CICS, the creation of association data was restricted to certain transaction identifiers. This restriction has now been removed.
- Provision for nonterminal START requests from third party adapters (such as WebSphere MQ, CICS sockets, WebSphere Optimized Local Adapter) to be tracked by allowing the adapter to set fields in the origin data of the task that they start within a CICS region.

This support includes the following changes:

- Four new adapter field options added to the origin data to enable identification of a third party resource manager.
- Four new data areas available for an adapter to set when called for context management to be placed in the origin data of the transaction that is started.
- Four new origin data fields available from the INQUIRE ASSOCIATION command. This allows information from other products to be included in the transaction tracking model, which benefits non-CICS environments.
- A new flag field that indicates to the adapter whether any adapter data it sets will be placed in the origin data of the task being started.
- The extension of transaction tracking to WebSphere MQ.

This support includes the following changes:

- Simplification of system operation tasks. The adapter data in the task started by the MQ trigger monitor enables a system programmer to quickly identify a specific MQ listener task and its associated queue at the point of origin of any task (or its child) without requiring changes to CICS applications.
- Greater integration with WebSphere MQ.

Terminology

Terminology used to explain previous data hop support for transaction tracking.

previous hop data. A set of information that identifies the remote sender of a request to attach a task and creates a trail to be followed back into the previous system. This process enables data gathering and monitoring to continue in the region that sent the request. Previous hop data is created for a task when it has been initiated by a task in another CICS system.

hop. One segment of a transmission path between adjacent nodes in a routed network.

hop count. A measure of the links between two systems on a network. A hop count of 5 means that four gateways separate the source and destination machines.

hop data. A packet of data transferred from one system to another that describes context about some earlier processing. Transaction tracking has two types of hop data: origin data and previous hop data.

Changes to external interfaces

Some external interfaces now support passing previous hop data with origin Data.

Changes to monitoring data

New performance class data in the DFHCICS group supports transaction tracking.

Changed performance class group, DFHCICS

The following performance data fields have been added to the DFHCICS group:

351 (TYPE-C, 'OADID', 64 BYTES)

The adapter identifier added to the origin data by the adapter. This field is blank if the task was not started by using an adapter, or if it was and the adapter did not set this value.

352 (TYPE-C, 'OADATA1', 64 BYTES)

The data added to the origin data by the adapter. This field is blank if the task was not started by using an adapter, or if it was and the adapter did not set this value.

353 (TYPE-C, 'OADATA2', 64 BYTES)

The data added to the origin data by using the adapter. This field is blank if the task was not started by using an adapter, or if it was and the adapter did not set this value.

354 (TYPE-C, 'OADATA3', 64 BYTES)

The data added to the origin data by the adapter. This field is blank if the task was not started by using an adapter, or if it was and the adapter did not set this value.

373 (TYPE-C, 'PHNTWKID', 8 BYTES)

The network identifier of the CICS system of an immediately previous task in another CICS system with which this task is associated.

374 (TYPE-C, 'PHAPPLID', 8 BYTES)

The APPLID from previous hop data. This is the APPLID of the CICS system of a previous task in another CICS system with which this task is associated. See Previous hop data characteristics for more information about previous hop data.

375 (TYPE-T, 'PHSTART', 8 BYTES)

The start time of the immediately previous task in another CICS system with which this task is associated.

376 (TYPE-P, 'PHTRANNO', 4 BYTES)

The task number of the immediately previous task in another CICS system with which this task is associated.

377 (TYPE-C, 'PHTRAN', 4 BYTES)

The transaction ID (TRANSID) of the immediately previous task in another CICS system with which this task is associated.

378 (TYPE-A, 'PHCOUNT', 4 BYTES)

The number of times there has been a request from one CICS system to another CICS system to initiate a task with which this task is associated.

Changes to transaction class resource data

The following transaction resource data fields have been added:

MNR_PHD_NTWKID (TYPE-C, 8 BYTES)

The network identifier of the CICS system of an immediately previous task in another CICS region with which this task is associated. For more information, see field 373 (PHNTWKID) in the DFHCICS performance data group.

MNR_PHD_APPLID (TYPE-C, 8 BYTES)

The APPLID from previous hop data. This is the APPLID of the CICS system of a previous task in another CICS system with which this task is associated. For more information, see field 374 (PHAPPLID) in the DFHCICS performance data group. For more information about previous hop data, see Previous hop data characteristics.

MNR_PHD_ATTACH_TIME (TYPE-T, 8 BYTES)

The start time of the immediately previous task in another CICS system with which this task is associated. For more information, see field 375 (PHSTART) in the DFHCICS performance data group.

MNR_PHD_TRANNUM (TYPE-P, 4 BYTES)

The task number of the immediately previous task in another CICS system with which this task is associated. For more information, see field 376 (PHTRANNO) in the DFHCICS performance data group.

MNR_PHD_TRANID (TYPE-C, 4 BYTES)

The transaction ID (TRANSID) of the immediately previous task in another CICS system with which this task is associated. For more information, see field 377 (PHTRAN) in the DFHCICS performance data group.

MNR_PHD_COUNT (TYPE-A, 4 BYTES)

The number of times there has been a request from one CICS system to another CICS region to initiate a task with which this task is associated. For more information, see field 378 (PHCOUNT) in the DFHCICS performance data group.

MNR_ID_TRNGRPID (TYPE-C, 28 BYTES)

The transaction group ID of the originating task.

Changes to Identity class data

The following identity class data fields have been added:

MNI_ID_PHD_NTWKID (TYPE-C, 8 BYTES)

The network identifier of the CICS system of an immediately previous task in another CICS system with which this task is associated. For more information, see field 373 (PHNTWKID) in the DFHCICS performance data group.

MNI_ID_PHD_APPLID (TYPE-C, 8 BYTES)

The APPLID from previous hop data. This is the APPLID of the CICS system of a previous task in another CICS system with which this task is associated. For more information, see field 374 (PHAPPLID) in the DFHCICS performance data group. For more information about previous hop data, see Previous hop data characteristics.

MNI_ID_PHD_START_TIME (TYPE-T, 8 BYTES)

The start time of the immediately previous task in another CICS system with which this task is associated. For more information, see field 375 (PHSTART) in the DFHCICS performance data group.

MNI_ID_PHD_TRANNO (TYPE-P, 4 BYTES)

The task number of the immediately previous task in another CICS system with which this task is associated. For more information, see field 376 (PHTRANNO) in the DFHCICS performance data group.

MNI_ID_PHD_TRANID (TYPE-C, 4 BYTES)

The transaction ID (TRANSID) of the immediately previous task in another CICS system with which this task is associated. For more information, see field 377 (PHTRAN) in the DFHCICS performance data group.

MNI_ID_PHD_COUNT (TYPE-A, 4 BYTES)

The number of times there has been a request from one CICS system to another CICS system to initiate a task with which this task is associated. For more information, see field 378 (PHCOUNT) in the DFHCICS performance data group.

Changes to interfaces

You can use the new options added to the **INQUIRE ASSOCIATION** command for enhanced support of transaction tracking.

Interface support for transaction tracking

Support for transaction tracking includes the following interfaces:

- Updated CICS Explorer **Operations > Task Association** and **Operations > Completed tasks** views: these views have been updated to provide support for previous hop and adapter data fields.
- Updated Explorer SM task search function: the search function has been enhanced to visualize task flows, and to provide search by TCPIP Address or TCPIP Service.
- Updated CICSplex SM views **CICS operations views > Task operations views > Task association information - TASKASSC**: this table has been updated to provide support for previous hop and adapter data fields.
- Updated CICSplex SM views **CICS operations views > Task operations views > Completed tasks - HTASK**: this table has been updated to provide support for previous hop and adapter data fields.
- Updated CICSplex SM Task Associations (TASKASSC) table: this table has been updated with new options that are available on the **INQUIRE ASSOCIATION** command, and the **EYUSTARTTASKASSC.TABULAR** and **EYUSTARTTASKASSC.DETAILED** WUI views have been updated to display these fields.
- Updated CICSplex SM Task History (HTASK) table: this table has been updated to contain additional fields that are provided by DFHMNTDS to provide previous hop and adapter data information, and the **EYUSTARTHTASK.DETAIL17** WUI view has been updated to display these fields.
- Updated **INQUIRE ASSOCIATION** command: this command has been enhanced to extend its support of transaction tracking with previous hop data that can be transmitted across an IPIC or MRO connection and nonterminal START requests from adapters provided by independent software vendors.

HTASK and TASKASSC resource updates

The following interfaces are updated to support the HTASK and TASKASSC resources:

CICS Explorer

 Operations view

CICSplex SM

- ☞ Completed tasks - HTASK
- ☞ Task association information - TASKASSC
- ☞ HTASK Resource Table
- ☞ TASKASSC Resource Table

The CICS SPI

- ☞ INQUIRE ASSOCIATION

Changes to task-related user exits

New adapter fields have been added to the origin data to enable identification of a third party resource manager.

To allow the adapter to pass values to populate the new adapter data fields in the origin data section of the association data of a task that is being started, five new fields are added to the context management parameters (mapped by DFHUECON):

UECON_ADAPTER_ID_PTR

The address of a 64-character area in which the exit program can pass the data to be placed into the origin data adapter identifier field. Use the same value for all instances of the adapter; for example, the product identifier for the owner of the adapter. If an adapter does not specify an identifier in this area then none of the other adapter data is set.

UECON_ADAPTER_DATA1_PTR

The address of a 64-character area in which the exit program can pass the data to be placed into the origin data adapter data 1 field. This field can be used to identify the server to which the adapter instance (which might be one of many) is connected.

UECON_ADAPTER_DATA2_PTR

The address of a 64-character area in which the exit program can pass the data to be placed into the origin data adapter data 2 field. This field can be used to identify the instance of the adapter task that is starting the task with the START command.

UECON_ADAPTER_DATA3_PTR

The address of a 64-character area in which the exit program can pass the data to be placed into the origin data adapter data 3 field. This field can contain details to identify the reason that the adapter instance started this particular task with the START command.

UECON_FLAGS

The address of a single byte with bit definitions that indicate to the adapter whether any adapter data it sets is placed in the origin data of the task being started.

UECON_ADAPTER_DATA_ON

(X'80') indicates that the adapter is permitted to set origin data adapter data fields.

If the adapter returns values for the adapter data when this flag is not set, the values are ignored. Only the first (origin) adapter for a task, or a set of tasks, is permitted to set the origin data adapter fields.

These fields are in the origin data, so they can only be set once, by the origin adapter. If they already exist, it is not possible for a different adapter to change them. UECON_FLAGS addresses a flag that permits the adapter to tell whether adapter data will be accepted.

Chapter 14. Improvements to dynamic workload management

Dynamic workload management now resolves problems associated with the use of multiple dynamic program link requests in a single unit of work.

When using dynamic workload management, problems can occur during the use of multiple dynamic program link (DPL) requests in a single unit of work (UOW). When multiple invocations of the same dynamically-routed program in a UOW access a common resource and one program link locks the resource, if the subsequent program link is routed to a different region, a deadlock can occur.

To prevent this problem, work must not be routed away from the region that locked the resource. CICS TS 4.2 introduces a new type of CICS affinity associated with a UOW, and extends CICSplex SM workload management (WLM) to manage these UOW affinities for DPL requests. The new affinity is restricted to programs that are dynamically linked. Programs with this type of affinity are routed to the same target region for the duration of a unit of work. These affinities are defined with an affinity relation of LOCKED and an affinity lifetime of UOW. To use the new UOW affinity with existing workloads you must restart the workloads with CICS TS 4.2.

In CICSplex SM, you can create transaction groups and WLM specifications incorporating this new type of affinity using the TRANGRP and WLMSPEC resource tables.

Changes to external interfaces

Some external interfaces now support the improvements to dynamic workload management.

Changes to CICSplex SM views and menus

CICSplex SM workload management administration views are updated with new fields and field values to configure UOW affinities. The **Active workloads** detailed view is also improved by the addition of new fields to display extra workload-related object counts.

Changed menu, Active workload views

The **Active workload views** menu is now divided into the following parts:

- Active workloads
- Active workload region data

Changed views

View	Changes
Administration views > Workload manager administration views > Specifications (WLMSPEC)	The following fields are changed: <ul style="list-style-type: none">• Default affinity has a new value of LOCKED.• Default affinity lifetime has a new value of UOW.

View	Changes
Administration views > Workload manager administration views > Transaction group definitions (TRANGRP)	<p>The following fields are changed:</p> <ul style="list-style-type: none"> • Affinity relationship has a new value of LOCKED. • Affinity lifetime has a new value of UOW.
Active workload views > Active workloads (WLMWORK)	<p>The following fields are new:</p> <p>Active workload definitions The number of active workload definitions that are associated with the workload</p> <p>Active dynamic transactions The number of active dynamic transactions that are associated with installed transaction groups</p> <p>Active transaction groups The number of active transactions groups that are associated with the workload</p> <p>Active transaction group affinities The number of active transaction group affinities that are associated with the workload</p> <p>The following fields are changed:</p> <ul style="list-style-type: none"> • Default affinity relation has a new value of LOCKED • Default affinity lifetime has a new value of UOW <p>The hyperlink from the Owning system ID field now links to the CMAS Details view for the system identified by the attribute.</p>
Active workload views > Transaction groups (WLMATGRP)	<p>The following fields are changed:</p> <ul style="list-style-type: none"> • Affinity relationship has a new value of LOCKED. • Affinity lifetime has a new value of UOW. <p>The Workload filter has a new generic value of * to display all active affinities in the current scope.</p>
Active workload views > Transaction group affinities (WLMATAFF)	<p>The following fields are new:</p> <ul style="list-style-type: none"> • Network unit-of-work ID • Local unit-of-work ID <p>The following fields are changed:</p> <ul style="list-style-type: none"> • Affinity relationship has a new value of LOCKED. • Affinity lifetime has a new value of UOW.

View	Changes
Active workload views > Active routing regions (WLMAWTOR)	<p>The following fields are new:</p> <p>Unit of work completes The number of times that the calling program of a dynamic program link has completed a unit of work</p> <p>Unique WLM routing region ID An internal value used by WLM that designates a unique numeric identifier to the current routing region in the CICSplex</p> <p>MVS system name The SMF ID of the MVS system where this CICS is running</p> <p>Operating system level The version of the operating system associated with this routing region</p> <p>Local time at which router was started The time at which this run of CICS started</p> <p>CICS Release The release number of the CICS system</p> <p>Abend compensation status The status of abend compensation for the transaction that the TOR is routing</p> <p>The hyperlink from the Routing region name field on the Active routing regions tabular view now links to the Active workload routing regions detailed view.</p>

Changes to CICSplex SM resource tables

The new LOCKED affinity type is supported by the new WLMAROUT resource table and some new and changed resource table attributes.

New table: WLMAROUT

The new workload management runtime resource table WLMAROUT reports on the state of router regions in a CICSplex. Unlike the existing WLMAWTOR resource table, WLMAROUT returns a router status record from every CMAS in the workload and can therefore display multiple records from the same target region. The WLMAWTOR reports only the state of a router region as seen by the CMAS that owns the workload.

Changed tables

Resource table	Changes
TRANGRP, WLMAROUT, WLMATGRP, WLMASPEC	<p>The following attributes are changed:</p> <ul style="list-style-type: none"> • AFFINITY has a new value of LOCKED. • AFFLIFE has a new value of UOW.

Resource table	Changes
WLMATAFF	<p>The following attributes are new:</p> <ul style="list-style-type: none"> • LOCUOWID: local unit of work ID. • NETUOWID: network unit of work ID. <p>The following attributes are changed:</p> <ul style="list-style-type: none"> • AFFINITY has a new value of LOCKED. • AFFLIFE has a new value of UOW.

Changes to user-replaceable programs

The dynamic transaction routing user-replaceable modules DFHDYPDS, EYURWCOM, and EYURWTRA now support the improvements to dynamic workload management.

Changes to DFHDYPDS

DFHDYPDS is the copy book that defines the communications area for the CICS-supplied sample distributed routing program DFHDSRP.

- The following DFHDYPDS tokens are new:

DYRUOWAF

Indicates that callback is required at the end of the unit of work.

DYRFUNC 7 = End_UOW

Identifies that this call is for end of unit of work processing.

DYRLUOWID

Identifies the local unit of work. This token forms part of the key for the LOCKED affinity type.

DYRNUOWID

Identifies the network unit of work. This token forms part of the key for the LOCKED affinity type.

- The DYRVER token is incremented by 1 to indicate that this module has changed for this CICS TS release.

If you use your own routing program, you might have to make adjustments for these changes.

Changes to EYURWCOM

EYURWCOM is the communication area used by the dynamic routing user-replaceable module EYU9WRAM.

The following EYURWCOM tokens are new:

WCOM_DYRLUOW

Identifies the local unit of work for this request.

WCOM_DYRNUOW

Identifies the network unit of work for this request.

The following EYURWCOM tokens are changed:

- **WCOM_AFF_TYPE** has a new value of **WCOM_AFF_LOCKED**
- **WCOM_AFF_LIFE** has a new value of **WCOM_AFF_UOW**

If you have customized EYU9WRAM, you might have to make adjustments for these changes.

Changes to EYURWTRA

EYURWTRA is the communication area for the dynamic routing program EYU9XLOP.

The following EYURWTRA tokens are new:

WTRA_UOWOPT

Signifies that the exit requires callback processing at the end of the UOW

WTRA_LOCUOWID

Identifies the local UOW token to be used

WTRA_NETUOWID

Identifies the network UOW token to be used

Chapter 15. CICS support for password phrases

CICS now supports the use of password phrases as well as standard passwords to authenticate a user ID when signing on to CICS. Because a password phrase can provide an exponentially greater number of possible combinations of characters than a standard password, the use of password phrases can improve system security and enhance usability.

A password phrase is a character string that can comprise mixed-case letters, numbers, and certain special characters from 9- to 100-characters in length.

The rules and characters that can be used for password phrases are defined and maintained by RACF. These rules can be summarized as follows:

- The length of a password phrase can be between 9 and 100 characters.
- The user ID must not be part of the password phrase.
- At least 2 alphabetic characters must be included.
- At least 2 non-alphabetic characters such as numbers, punctuation marks, and special characters must be included.
- Valid characters are as follows:
 - Alphabetic uppercase and lowercase characters.
 - Numbers 0 through 9.
 - National and special characters such as # , @ , and \$.
 - Punctuation characters.
 - Blank spaces.
- No more than 2 consecutive characters can be the same.

In addition, special restrictions can apply in certain environments, for example, the use of leading and trailing blanks can cause problems, and in command line interfaces, single quotes must be enclosed in two more single quotes. If you use multiple user interfaces for entering password phrases, use the RACF installation exit for password phrases to ensure that only those characters supported by all interfaces are allowed.

A user ID can have both a standard password and a password phrase. When you sign on to CICS, if you enter a password that is up to 8 characters long, it is treated as a standard password and the ESM checks that the character string matches the standard password recorded by the ESM for your user ID. If the length of the password you enter is between 9 and 100 characters, it is treated as a password phrase and the ESM checks that the character string matches the password phrase recorded for your user ID.

The CICS API **SIGNON** command and the two new commands **CHANGE PHRASE** and **VERIFY PHRASE** support the use of password phrases and standard passwords. However you cannot use a standard password to authorize a change of password phrase. Similarly, you cannot use a password phrase to authorize a change of standard password. The existing **CHANGE PASSWORD** and **VERIFY PASSWORD** commands are unchanged and do not support the use of password phrases.

The new supplied transaction CESL (sign-on long) supports both standard passwords and password phrases. The CESN transaction is unchanged and supports only standard passwords. However you can ensure that password phrases are always available by defining CESL as an alias for transaction CESN.

Terminology

Terms used to explain CICS support for password phrases.

password phrase. A string consisting of mixed-case letters, numbers, and special characters, including blanks, that is used to control access to data and systems.

Changes to external interfaces

Several external interfaces now support CICS password phrases.

Changes to the CICS application programming interface

Password phrases are supported by two new API commands: CHANGE PHRASE and VERIFY PHRASE. The SIGNON command is changed to enable the use of password phrases as well as standard passwords when logging on to CICS

New commands

CHANGE PHRASE

The CHANGE PHRASE command changes the password or password phrase recorded by an external security manager (ESM) for a specified user ID.

VERIFY PHRASE

The VERIFY PHRASE command checks that a password or password phrase matches the password or password phrase recorded by an ESM for a user ID. The command returns the values recorded by the ESM for the password or password phrase.

Changed command, SIGNON

The SIGNON command includes new options to support password phrases as well as standard passwords. If authorization is required, you can sign on with either a valid password or a valid password phrase. However, you cannot change a password phrase using a password for authentication, nor can you change a standard password using a password phrase for authentication.

Related topics

 CHANGE PHRASE

 VERIFY PHRASE

 SIGNON

Changes to supplied transactions

CESL is a new supplied transaction that you can use to sign on to CICS using a password or a password phrase as authorization. Transaction CEDF is changed to support password phrases.

New transaction, CESL

With CESL, you can sign on to CICS with a password phrase of 9 to 100 characters or a standard password of up to 8 characters. In other respects CESL operates in the same way as the CESN sign-on transaction. CESN does not support password phrases but you can specify CESL as an alias for CESN. If you enter a password that is 9 to 100 characters, CESL treats the password as a password phrase. CESL is a category 3 transaction.

Changed transaction, CEDF

The CEDF transaction is changed to prevent the viewing of password phrases as well as standard passwords when you debug an application.

Related topics

-  [CESL - Sign-on long](#)
-  [Security checking of transactions running under CEDF](#)

Changes to global user exits

The UEPPSWD parameter of the XWBAUTH global user exit is changed to support password phrases.

Changed global user exit, XWBAUTH

The size of the predefined area of the UEPPSWD (Output supplied by user exit) parameter is increased from 64 bytes to 128 bytes to provide sufficient storage for password phrases.

Changes to problem determination

Several messages are added and changed in support of password phrases.

New messages

The following messages are new:

- DFHCE3554
- EYUVC1019E
- EYUVC1020E
- EYUVC1021E
- EYUVC1023E
- EYUVC1024E
- EYUVC1025E
- EYUVC1026E
- EYUVC1027E

Changed messages

The following messages are changed:

- DFHCE3503
- DFHCE3504

Chapter 16. Dynamic transaction-level control for workload routing

Workload routing is enhanced by the ability to control dynamic routing behavior at the transaction level. You can now specify a dynamic routing algorithm for specific transactions in a workload that is different from the algorithm defined in the parent workload management specification.

A workload specification (WLMSPEC) controls the default routing behavior for all the transactions that workload management (WLM) is called on to evaluate in a routing region or routing system group. You can override certain routing characteristics by supplying associated pairs of WLM definitions and transaction group definitions (WLMDEF and TRANGRP objects) to specify a specialized routing evaluation to be applied to one or more transactions. However, before CICS TS 4.2 you could not specify alternative workload routing characteristics in this way. Workload routing is the directing of transactions among a group of target regions according to the availability and activity levels of those target regions. The criteria used to select a target region is determined by the choice of routing algorithm, each of which contains a different combination of weighting factors.

In CICS TS 4.2, you can now specify one of four routing algorithms in a transaction group. The selected algorithm overrides the default algorithm associated with the workload specification. If you specify an alternative routing algorithm in this way, you can change workload routing characteristics for specific target regions dynamically without stopping your routing region.

Changes to external interfaces

CICSplex SM resources and views are changed to support the new attribute ALGTYPE.

Changes to resources and attributes

The ALGTYPE attribute has been added to the TRANGRP and WLMATGRP resource tables and their associated views.

Support for the resource attribute: ALGTYPE

You can now use the ALGTYPE resource attribute to specify the WLM routing algorithms in a transaction group as well as in a workload specification.

The ALGTYPE attribute has the following support:

- The following CICSplex SM view sets are updated to display the Algorithm type (ALGTYPE) field:
 - Transaction group definitions (**Administration views > Workload manager administration views > Transaction group definitions**)
 - Active workload transaction groups (**Active workload views > Transaction groups**)
- The WLMATGRP and TRANGRP CICSplex SM resource tables are updated to include the ALGTYPE attribute.

Chapter 17. New WLM routing algorithms to exclude link weighting

Workload management (WLM) is enhanced by the introduction of new routing algorithms that exclude the connection type as a weighting factor in decisions to direct work requests to a target region.

Links between router regions and target regions can be of type Local, MRO, MRO/XCF, IPIC, LU6.2, and INDIRECT. In previous CICS releases WLM directed work requests to a target region using one of two weighting algorithms: the queue algorithm or the goal algorithm. In both of these algorithms the type of links between the router and the target are significant weighting factors, which are used along with other factors when distributing work from a router to a target. Link weighting can have a strong impact on the routing behavior in some situations and can prevent CICS transaction routing across system boundaries in a single-site parallel sysplex environment.

To increase your routing options, two new algorithms are introduced: LNQUEUE (link neutral queue) and LNGOAL (link neutral goal). LNQUEUE and LNGOAL correspond to the existing QUEUE and GOAL algorithms but exclude the connection type as a weighting factor.

The LNQUEUE algorithm selects the target region that conforms to the following criteria:

- Has the shortest queue of work waiting to be processed, relative to the maximum number of tasks permitted in the target region.
- Is the least affected by conditions such as short-on-storage, SYSDUMP, and TRANDUMP.
- Is the least likely to cause the transaction to stop.

The LNGOAL algorithm selects the target region that is the most likely to allow the transaction to meet the response time goal set for it and other transactions in its z/OS workload management class. The LNGOAL mode algorithm reverts to LNQUEUE mode when multiple target regions are achieving the same best response time goals, or when the target and routing regions are managed by different CMASs.

You can specify the new algorithms in the same way as GOAL and QUEUE; when defining a workload specification (WLMSPEC), or dynamically in a transaction definition (TRANGRP).

Changes to external interfaces

CICSplex SM resources and views support the new WLM routing algorithms LNGOAL and LNQUEUE.

Changes to resources and attributes

LNGOAL and LNQUEUE are new values for the ALGTYPE attribute in the WLMATGRP, WLMWORK, WLMSPEC and TRANGRP resources and their associated views.

Support for updated resource attribute: ALGTYPE

You can use the updated ALGTYPE resource attribute to specify the new WLM routing algorithms. LNGOAL (link neutral goal) and LNQUEUE (link neutral queue).

The updated ALGTYPE attribute has the following support:

- The following CICSplex SM view sets are updated to display the values LNGOAL and LNQUEUE for the **Algorithm type** field:
 - Active workloads (**Active workload views > Active workloads**)
 - Specifications (**Administration views > Workload manager administration views > Specifications**)
 - Transaction groups (**Active workload views > Transaction groups**)
 - Transaction group definitions (**Administration views > Workload manager administration views > Transaction group definitions**)
- The following CICSplex SM resource tables are updated to include the values LNGOAL and LNQUEUE for the ALGTYPE attribute:

TRANGRP

WLMATGRP

WLMAWORK

WLMSPEC

Chapter 18. Modified WLM routing for IPIC connections

Workload management (WLM) can use the type of connection between a routing region and target regions as a weighting factor when making dynamic routing decisions. The weighting applied to IP connectivity (IPIC) connections is decreased so that target regions that use this type of connection are more attractive than LU6.2 connections in decisions to route work requests.

In CICSplex SM workload management, the dynamic routing process evaluates the viability or health of a target region when making workload routing decisions. These decisions are based on routing algorithms that contain weighting factors. The type of connection between routing and target regions is a significant weighting factor in the QUEUE and GOAL routing algorithms (the connection type is not a weighting factor for the LNQUEUE and LNGOAL routing algorithms).

The weighting applied to IPIC connections is altered so that this type of connection has higher priority relative to LU6.2 connections than in previous releases of CICS TS. Also, a lower weighting is applied to IPIC connections between routing and target regions in the same LPAR than IPIC connections to target regions in a different LPAR. This means that if other factors are equal, local IPIC connections are preferred to remote IPIC connections.

In CICS TS 4.2, the relative connection weights put links in the following order of priority:

- Local
- MRO/IRC and MRO/XM (local LPAR)
- MRO/XCF (remote LPAR)
- IPIC (local LPAR)
- IPIC (remote LPAR)
- LU6.2
- Indirect

CICSplex SM uses the weighting factor of the link as a multiplier against the task load and other factors to determine an overall routing weight. At the end of the evaluation, the region with the lightest weight is usually selected as the target region.

Chapter 19. Enhancement to system availability monitoring

System availability monitoring (SAM) is enhanced by the new predefined event SAMNRM (nonresponsive MAS), which occurs when CICSplex SM detects that an active MAS has lost contact with its managing CMAS. CICSplex SM workload management uses the SAMNRM event in assessing the health of target regions when making dynamic routing decisions.

The CICSplex SM system availability monitoring (SAM) function monitors CICS systems during their planned times of availability. If any of a set of predefined conditions occurs while a CICS system is being monitored, CICSplex SM sends out external notifications at the start of the condition and also when it is resolved. SAMNRM is an addition to this set of events that occurs when a CMAS determines that it has lost contact with a managed CICS system (MAS) agent. This condition typically occurs when a MAS does not respond to a query from the CMAS for a reasonable time. A SAMNRM event does not necessarily mean that the MAS is not functional for user applications but only that the MAS agent code is in a nonresponsive state.

Unlike other SAM conditions, the SAMNRM event is detected only if the CMAS interacts with the MAS. The condition will not be detected if no CMAS to MAS interaction occurs.

When a SAM event is detected, CICSplex SM issues an external message that is written to the MVS console by default. External messages can also be viewed from the CICSplex SM Outstanding events (EVENT) view while a condition is active. Each SAM event has a default severity and priority value that determines the position of the external message, relative to other external messages. The SAMNRM condition has a default severity of HW (High Warning) and a priority of 255. You can customize the type and severity of notifications issued following a SAM event for any CICS system by creating a CICSplex SM action definition. To support the new SAM event the value SAMNRM is added to the NAME attribute of the EVENT resource and the associated Outstanding events (**Real Time Analysis (RTA) views > Outstanding events**) detailed view.

In CICSplex SM workload management, the dynamic routing process makes an evaluation of the viability or health of a target region when making routing decisions. These decisions are based on routing algorithms containing weighting factors. The addition of SAMNRM as one of these weighting factors makes it less likely that a routing region will send work to a target region that has lost contact with its CMAS. However CICSplex SM workload management can act on the SAMNRM event only if queries that need to gather data from the MAS are run during the workload execution.

Chapter 20. CMCI support for task history recording

You can now use the CICS management client interface (CMCI) to set up and configure the recording of data for completed tasks in a CICSplex.

You can already use CMCI to view historical task data. Now CMCI also supports the CICSplex SM monitoring resources necessary to set up and configure the recording of this data. With this additional resource support you can construct CMCI queries incorporating the GET, DELETE, POST, and PUT methods to create, update and delete monitoring-related resources such as monitor definitions, monitor groups and monitor specifications.

Changes to external interfaces

CMCI is changed to support task history recording.

Changes to CMCI

CMCI now supports additional CICSplex SM monitoring resources.

The new supported resources together with their external resource names for use in CMCI queries are as follows:

CICSplex SM resource name	External resource name	Description
LNKSMSCG	CICSMonitorSpecificationsToSystemGroup	Monitor specifications to system group links
LNKSMSCS	CICSMonitorSpecificationsToSystem	Monitor specifications to CICS system links
MASHIST	CICSTaskHistoryCollection	CICS task history collection
MONDEF	CICSMonitorDefinition	Monitor definitions
MONGROUP	CICSMonitorGroup	Monitor groups
MONINGRP	CICSMonitorResourceInGroup	Monitor definitions in groups
MONINSPC	CICSMonitorGroupInSpecification	Monitor groups in monitor specifications
MONSPEC	CICSMonitorSpecification	Monitor specifications

Chapter 21. CMCI support for multi-field sorting of returned records

You can now use the CICS management client interface (CMCI) to return records that are sorted by one or more attributes. This change means that you can now sort data that is viewed using the CICS Explorer.

Changes to external interfaces

CMCI is changed to support multi-field sorting of returned records.

Changes to the CICS application programming interface

The CMCI GET request URI has changed to support sorting records.

Changed command, ORDERBY

The ORDERBY parameter indicates that the result set is sorted by the specified fields. This attribute is specified by a string of logical expressions; the string that makes up the value of the ORDERBY parameter follows the same rules as the in the CICSplex SM application programming interface. For more information, see CICS management client interface GET request URI

Changes to problem determination

CICS provides new information to help you diagnose problems relating to the CMCI domain.

New trace points

- The following new trace points are issued by the CMCI domain:
- AP A2CD through AP A2CF
- AP A40B through AP A40D
- AP A4D0 through AP A4D7
- AP A4DF
- AP A4E0
- AP A4EF

Chapter 22. Addition of MVS job ID information to CICSplex SM

You can now use CICSplex SM interfaces including CMCI and the WUI to retrieve the MVS job number associated with an address space. For a running CMAS or CICS region this information makes it easier to identify job entry subsystem (JES) related information such as the SYSOUT data sets associated with the current CICS execution.

The CICSplex SM resource tables CICSRRGN and CMAS are expanded to include the new attribute JOBID that contains the MVS job ID. This attribute is included in the following CICSplex SM views:

CICS regions - CICSRRGN

CICS operations views > CICS region operations views > CICS regions

CMASs known to local CMAS - CMASLIST

CICSplex SM operations views > CMASs known to local CMAS

CMAS detail

CICSplex SM operations views > MAS status by CMAS > CMAS detail

You can also retrieve job ID information with CMCI using the external resource names CICSRRGN and CICSRRGNManagingAddressSpace.

Chapter 23. CICSplex SM API improvements: Discovering system initialization parameters

You can now use the CICSplex SM API to discover information about CICS system initialization parameters and system initialization parameter overrides. System initialization parameter retrieval is supported by the CICSplex SM command-level interface, the CICS management client interface (CMCI), and the Web User Interface (WUI).

You use system initialization parameters to modify CICS system attributes when you start CICS regions. The primary method of specifying system initialization parameters is in a system initialization table (SIT). SIT parameters supply the system initialization program with the initial set of parameters necessary to initialize the system to suit your requirements. However, you can also specify system initialization parameter values in other ways, which override the values originally coded in the SIT. During startup these overrides are applied in sequence from the following sources:

1. The PARM parameter of the EXEC PGM=DFHSSIP statement
2. The SYSIN data set defined in the startup job stream
3. The system operator's console

When you retrieve parameters you have the following options:

- You can retrieve the current values of the parameters in the SIT including any override values.
- You can retrieve the original SIT values as specified at system startup.
- You can retrieve the values from a single override source.

In common with many other CICSplex SM operations, you can control which CICS regions the retrieval operates on by specifying context and scope.

The new API capability is implemented using the CICSplex SM resource SYSPARM. The SYSPARM resource has two required parameters, PARMSRCE and PARMTYPE, associated with the GET operation. You use these parameters to specify which parameters to retrieve according to their source.

You can implement system initialization parameter discovery in three ways:

- In an API program using the EXEC CPSM GET command operating on the SYSPARM object.
- Using the CMCI GET method operating on the CICSSystemParameter external resource.
- Using the WUI operations view based on the SYSPARM resource table linked from the CICS region view set.

Changes to external interfaces

All CICSplex SM external interfaces now support the retrieval of information about CICS system initialization parameters and their override values.

Changes to resources and attributes

SYSPARM is a new CICS operational resource that displays information about a selected system initialization parameter or system initialization parameter override from a CICS region that is being managed by CICSplex SM.

Support for new resource: SYSPARM

You can use the new SYSPARM resource to retrieve information about CICS system initialization parameters and system initialization parameter overrides.

The new SYSPARM resource has the following support:

- New WUI view set SYSPARM.
- New SYSPARM operational resource table.
- New CMCI external resource: CICSSystemParameter. To retrieve system parameters with CMCI, use the GET method specifying the resource name CICSSystemParameter, and a parameter expression made up of the PARMSRCE and PARMTYPE parameters.

Chapter 24. Enhancements to main and auxiliary temporary storage queues

Main temporary storage queues can now use 64-bit (above-the-bar) storage. CICS provides new facilities so that you can check the storage use of main temporary storage queues and limit that storage use. You can specify that CICS deletes temporary storage queues automatically when they are no longer required by adding an expiry interval to the corresponding temporary storage models.

Main temporary storage is in 64-bit storage rather than 31-bit (above-the-line) storage, depending on the version of the z/OS operating system and whether the CICS region operates with transaction isolation. See CICS facilities that can use 64-bit storage in the Performance Guide.

If your CICS applications use large amounts of main temporary storage, the move to 64-bit storage can increase the available storage elsewhere in your CICS regions.

Auxiliary temporary storage queues and shared temporary storage queues continue to use 31-bit storage.

If your CICS applications currently specify the location of temporary storage, you can review this. If an application specifies that auxiliary temporary storage is used, and you do not require recoverable temporary storage, you can change it to specify main temporary storage. The advantages of this action are that space becomes available in 31-bit storage, and input/output activity to write data to disk is reduced.

You can control how much storage the CICS region makes available to main temporary storage queues by using the **TSMAINLIMIT** system initialization parameter. The default is 64 MB. This limit on storage use does not include auxiliary temporary storage queues and shared temporary storage queues.

Automatic deletion of eligible temporary storage queues reduces the unnecessary use of virtual storage. To use this feature, you set suitable expiry intervals in the temporary storage models (TSMODEL resource definitions). The expiry interval is available for main temporary storage queues and nonrecoverable auxiliary temporary storage queues that match TSMODEL resource definitions in the local CICS region. For more information, see Automatic deletion of temporary storage queues.

Changes to external interfaces

Some external interfaces now support the enhancements to main and auxiliary temporary storage queues.

Changes to system initialization parameters

A new system initialization parameter, **TSMAINLIMIT**, controls the use of storage by main temporary storage queues.

New system initialization parameter, TSMAINLIMIT

The **TSMAINLIMIT** system initialization parameter specifies a limit for the storage that is available for main temporary storage queues to use. You can specify an amount of storage in the range 1 - 32768 MB (32 GB), but this amount must not be greater than 25% of the value of the z/OS parameter **MEMLIMIT**. The default is 64 MB.

Related information:

 [TSMAINLIMIT system initialization parameter in the System Definition Guide](#)

Changes to resources and attributes

You can use the updated TSMODEL resource and the new TEMPSTORAGE functions to manage how temporary storage queues use storage in the CICS region.

Support for updated resource: TSMODEL

You can use the new EXPIRYINT attribute in the TSMODEL resource to set an expiry interval for temporary storage queues. You can use the INQUIRE TEMPSTORAGE and SET TEMPSTORAGE commands to query and change the use of storage by temporary storage queues.

The updated TSMODEL resource has the following support:

- Updated Explorer SM Operations views **TS Models view** and **TS Queues view**: use the new fields to view or set the expiry interval for temporary storage queues.
- Updated CICSplex SM views **Temporary storage queue (TSQ) operations views** > **Temporary storage queues - TSQNAME**, **Shared queues - TSQSHR**, **Temporary storage queues - TSQUEUE**, **Models - TSMODEL**: use the new fields to view or set the expiry interval for temporary storage queues.
- Updated CICSplex SM view **CICS resource definitions** > **Temporary storage model definitions - TSMDEF**: use the EXPIRYINT attribute to add expiry intervals to your temporary storage models.
- Updated CICSplex SM view **Temporary storage queue (TSQ) operations views** > **Global temporary storage statistics - TSQGBL**: use the new fields to see the storage that main temporary storage queues use and to change the storage limit.
- Updated CICSplex SM tables TSQUEUE, TSQNAME, TSQSHR, TSMODEL, and TSMDEF: use the new EXPIRYINT attribute to view or set the expiry interval for temporary storage queues.
- Updated CICSplex SM table TSQGBL: use the new TSMAINLIMIT, TSMAINUSED, and TSMAINPCT attributes to see the storage that main temporary storage queues use and to change the storage limit.
- New INQUIRE TEMPSTORAGE and SET TEMPSTORAGE commands: query the use of storage by temporary storage queues, and change the amount of storage that is available for them.
- Updated INQUIRE TSQUEUE or TSQNAME, INQUIRE TSMODEL, and CREATE TSMODEL commands: use the new EXPIRYINT option to view or set the expiry interval for temporary storage queues.

TSMODEL resource updates

The following interfaces are updated to support the TSMODEL resource:

CICS Explorer

TS Models view

TS Queues view

CICSplex SM

[Temporary storage queues - TSQNAME in the CICSplex SM Operations Views Reference](#)

[TSQNAME Resource Table in the CICSplex SM Resource Tables Reference](#)

[Shared queues - TSQSHR in the CICSplex SM Operations Views Reference](#)

[TSQSHR Resource Table in the CICSplex SM Resource Tables Reference](#)

[TSQUEUE Resource Table in the CICSplex SM Resource Tables Reference](#)

[Models - TSMODEL in the CICSplex SM Operations Views Reference](#)

[TSMODEL Resource Table in the CICSplex SM Resource Tables Reference](#)

[Temporary storage model definitions - TSMDEF in CICS Administration Views](#)

[TSMDEF Resource Table in the CICSplex SM Resource Tables Reference](#)

[Global temporary storage statistics - TSQGBL in the CICSplex SM Operations Views Reference](#)

[TSQGBL Resource Table in the CICSplex SM Resource Tables Reference](#)

CEMT

[CEMT INQUIRE TEMPSTORAGE in CICS Supplied Transactions](#)

[CEMT SET TEMPSTORAGE in CICS Supplied Transactions](#)

[CEMT INQUIRE TSMODEL in CICS Supplied Transactions](#)

[CEMT INQUIRE TSQUEUE in CICS Supplied Transactions](#)

[CEMT INQUIRE TSQNAME in CICS Supplied Transactions](#)

The CICS SPI

[INQUIRE TEMPSTORAGE in CICS System Programming Reference](#)

[SET TEMPSTORAGE in CICS System Programming Reference](#)

[INQUIRE TSMODEL in CICS System Programming Reference](#)

[INQUIRE TSQUEUE in CICS System Programming Reference](#)

[INQUIRE TSQNAME in CICS System Programming Reference](#)

[CREATE TSMODEL in CICS System Programming Reference](#)

Changes to statistics

The CICS statistics for temporary storage queues provide additional information about storage use.

Changed statistics type: Temporary storage statistics

The CICS global statistics for temporary storage, mapped by the DFHTSGDS DSECT, now include the following information:

- The TSMAINLIMIT setting and the number of times that main temporary storage use reached the limit.
- The current and peak amount of virtual storage that was used for data in main temporary storage.
- The number of times that the cleanup task, which deletes eligible temporary storage queues automatically, has run.
- The number of temporary storage queues that CICS has deleted automatically by using the cleanup task.

Changes to problem determination

New messages and trace points that relate to temporary storage queues have been added.

New messages

New messages in the range DFHTS1601 - DFHTS1608 relate to the use of main storage by temporary storage queues, and automatic deletion of queues.

New trace points

- New trace points TS 0106 and TS 0107 relate to the timer for the cleanup function.
- New trace points in the range TS 0D01 - TS 0D08 relate to automatic deletion of queues.
- New trace points in the range TS 0E01 - TS 0E09 relate to the **TSMAINLIMIT** system initialization parameter.

Chapter 25. Enhancements to CECI and EDF

The CECI transaction and Execution Diagnostic Facility (EDF) now display UTF-8 data.

For input data, both CECI and EDF convert EBCDIC to US ASCII in character mode and do no conversion when in hexadecimal mode. For output data, both CECI and EDF convert all US ASCII characters to the EBCDIC equivalent and the unprintable byte values to ' '.

Part 5. Scalability

CICS Transaction Server for z/OS, Version 4 Release 2 delivers a set of capabilities represented in the following themes:

- Events
- Java
- Connectivity
- Management
- Scalability

The *Scalability* theme provides enhanced scalability of the CICS environment, with major extensions to multiprocessor exploitation, including threadsafe access via the CICS-DBCTL interface to IBM Information Management System (IMS™) Databases and function shipping requests between CICS systems over IPIC connections.

Chapter 26. Threadsafe extensions

More areas have been made threadsafe in this release to help increase throughput and reduce CPU usage.

Threadsafe API and SPI commands

Many new system programming interface (SPI) commands in CICS Transaction Server for z/OS, Version 4 Release 2 are threadsafe. Additionally, some existing commands have been made threadsafe in this release.

New API commands that are threadsafe

CHANGE PHRASE
VERIFY PHRASE

New SPI commands that are threadsafe

INQUIRE CAPDATAPRED
INQUIRE CAPINFOSRCE
INQUIRE CAPOPTPRED
INQUIRE EPADAPTER
INQUIRE OSGIBUNDLE
INQUIRE OSGISERVICE
INQUIRE TEMPSTORAGE
SET EPADAPTER
SET TEMPSTORAGE

Existing API commands made threadsafe in this release

BIF DEEDIT
BIF DIGEST

CHANGE PASSWORD

DEFINE COUNTER and DEFINE DCOUNTER

DELETE: before CICS TS for z/OS, Version 4.2, this command was threadsafe when it was used with a file in a local CICS region, but it was not threadsafe when it was used with a file in a remote CICS region. The command is now threadsafe when it is function shipped to a remote CICS region over an IPIC connection only. For other types of connection to remote CICS regions, the command remains nonthreadsafe.

DELETE COUNTER and DELETE DCOUNTER

DELETEQ TS: before CICS TS for z/OS, Version 4.2, this command was threadsafe when it was used with a queue in main storage or auxiliary storage in a local CICS region, but it was not threadsafe when it was used with a queue in a remote CICS region. The command is now threadsafe when it is function shipped to a remote CICS region over an IPIC connection only. For other types of connection to remote CICS regions, the command remains nonthreadsafe. The command also remains nonthreadsafe when it is used with a queue in a shared temporary storage pool in a z/OS coupling facility managed by a temporary storage data sharing server (TS server).

ENDBR: in the same way as DELETE, this command is now threadsafe when it is function shipped to a remote CICS region over an IPIC connection only.

EXEC DLI

EXTRACT CERTIFICATE

EXTRACT TCPIP

GET COUNTER and GET DCOUNTER

LINK: before CICS TS for z/OS, Version 4.2, this command was threadsafe when it was used to link to a program in a local CICS region, but it was not threadsafe when it was used to link to a program in a remote CICS region. The LINK command is now threadsafe when it is used to link to a program in a remote CICS region over an IPIC connection only. For other types of connection to remote CICS regions, the command remains nonthreadsafe.

QUERY COUNTER and QUERY DCOUNTER

QUERY SECURITY

READ: in the same way as DELETE, this command is now threadsafe when it is function shipped to a remote CICS region over an IPIC connection only.

READQ TS: in the same way as DELETEQ TS, this command is now threadsafe when it is function shipped to a remote CICS region over an IPIC connection only.

READNEXT: in the same way as DELETE, this command is now threadsafe when it is function shipped to a remote CICS region over an IPIC connection only.

READPREV: in the same way as DELETE, this command is now threadsafe when it is function shipped to a remote CICS region over an IPIC connection only.

RESETBR: in the same way as DELETE, this command is now threadsafe when it is function shipped to a remote CICS region over an IPIC connection only.

REWIND COUNTER and REWIND DCOUNTER

REWRITE: in the same way as DELETE, this command is now threadsafe when it is function shipped to a remote CICS region over an IPIC connection only.

SIGNOFF

SIGNON

STARTBR: in the same way as DELETE, this command is now threadsafe when it is function shipped to a remote CICS region over an IPIC connection only.

SYNCPOINT: the Recovery Manager now processes this command on an open TCB wherever possible to minimize TCB switching.

SYNCPOINT ROLLBACK: the Recovery Manager now processes this command on an open TCB wherever possible to minimize TCB switching.

UNLOCK: in the same way as DELETE, this command is now threadsafe when it is function shipped to a remote CICS region over an IPIC connection only.

UPDATE COUNTER and UPDATE DCOUNTER

VERIFY PASSWORD

WRITE: in the same way as DELETE, this command is now threadsafe when it is function shipped to a remote CICS region over an IPIC connection only.

WRITEQ TS: in the same way as DELETEQ TS, this command is now threadsafe when it is function shipped to a remote CICS region over an IPIC connection only.

Existing SPI commands made threadsafe in this release

INQUIRE CLASSCACHE

INQUIRE JVM

INQUIRE JVMPOOL

INQUIRE JVMPROFILE

PERFORM CLASSCACHE

PERFORM JVM POOL

RESYNC ENTRYNAME

SET CLASSCACHE

SET JVMPOOL

Invoking DL/I by using the applicable language interface, for example the COBOL statement **CALL CBLTDLI**, is now threadsafe when used with IMS Version 12 or later.

Threadsafe programs

The following programs are new or have been made threadsafe in this release.

DFHMIRS

The supplied mirror program DFHMIRS, which is used by all mirror transactions, is now defined as threadsafe.

For threadsafe applications that function ship commands to other CICS regions by using IPIC, CICS runs the mirror program on an L8 open TCB whenever possible. The resulting reduction in TCB switching improves the performance of the application compared to other intercommunication methods. To gain the performance improvement, you must specify the system initialization parameter FCQRONLY=NO in the file-owning region. For more information about function shipping and improving performance in function shipped file control, see Overview of function shipping and Threadsafe file control applications.

For threadsafe applications that use CICS distributed program link (DPL) functions by using IPIC over TCP/IP, CICS runs the mirror program on an L8 open TCB whenever possible, which can improve performance for threadsafe programs in the server region. For more information about DPL, see Overview of DPL.

For more information about DFHMIRS, see DFHMIRS.

Chapter 27. CICS use of 64-bit storage

In CICS TS for z/OS, Version 4.2, some additional CICS infrastructure and system features can use 64-bit (above-the-bar) storage in the CICS region, in place of 31-bit (above-the-line) storage. The move of these items to 64-bit storage increases the virtual storage available for your existing and new applications to operate in 31-bit storage.

For example, main temporary storage and the internal trace table can now use 64-bit storage. For a more detailed list of CICS facilities that can use 64-bit storage, see CICS facilities that can use 64-bit storage in the Performance Guide.

Applications

You do not need to change any existing applications because of the CICS facilities that now use 64-bit storage.

Where CICS stores data in 64-bit storage, applications do not request or access the 64-bit storage directly. CICS manages the storage for applications in the same way as before, and the location of the storage is not apparent to applications. Your applications do not run in AMODE(64) or use 64-bit instructions.

However, if an application specifies the location of temporary storage, you might want to review this and specify that main temporary storage is used rather than auxiliary temporary storage. See Chapter 24, “Enhancements to main and auxiliary temporary storage queues,” on page 97 and “Temporary storage queues” in the *CICS Application Programming Guide*.

System parameters

Because CICS infrastructure and system features can use 64-bit storage, you must review the following parameters:

- The z/OS **MEMLIMIT** parameter, which specifies the limit for 64-bit storage for a CICS region. A CICS region needs at least 4 GB of 64-bit storage. A suitable **MEMLIMIT** value depends on the facilities that you plan to use. See Estimating, checking, and setting **MEMLIMIT** in the Performance Guide.
- The CICS **TRTRANSZ** system initialization parameter, which specifies the size of the transaction dump trace table. When a transaction dump is produced, this trace table is now created in 64-bit storage. Therefore, you might provide more space for the table, to produce more valuable problem diagnosis information. For more information, see Chapter 28, “Changes to storage use for CICS trace,” on page 115.
- The CICS **TRTABSZ** system initialization parameter, which specifies the size of the CICS internal trace table. The internal trace table can now be in 64-bit storage (see “Conditions for CICS facilities to use 64-bit storage” on page 110). If you specified a small size for the trace table in existing CICS regions because of concerns about the availability of 31-bit storage, and the internal trace table is in 64-bit storage, consider increasing the size of your trace table. For more information, see Chapter 28, “Changes to storage use for CICS trace,” on page 115.

If your CICS applications use main temporary storage, you can control how much storage the CICS region makes available to main temporary storage queues by using the new CICS system initialization parameter **TSMAINLIMIT**. See “Changes to system initialization parameters” on page 97.

Conditions for CICS facilities to use 64-bit storage

In CICS TS for z/OS, Version 4.2, some CICS facilities use 64-bit storage rather than 31-bit storage, depending on the version of the z/OS operating system and whether the CICS region operates with transaction isolation.

CICS operates without transaction isolation when the **TRANISO** system initialization parameter is set to NO.

The following table shows the conditions that affect whether such CICS facilities use 64-bit or 31-bit storage.

Table 1. Conditions that affect whether CICS facilities use 64-bit or 31-bit storage

z/OS operating system version	CICS operates with or without transaction isolation?	Storage
V1R11 V1R12	With	31-bit
V1R11 V1R12	Without (TRANISO=NO)	64-bit
V1R12 with the PTF for APARs PA14311 and PA14312 applied V1R13 and later releases	Without (TRANISO=NO)	64-bit

The CICS facilities that are affected by these conditions are shown in CICS facilities that can use 64-bit storage in the Performance Guide.

Terminology

Terms used to explain CICS use of 64-bit storage.

24-bit storage. Virtual storage below the 16 MB address in a z/OS address space. Addresses below the 16 MB address are accessed by 24-bit addressing, and programs can use this storage when they run in AMODE 24 or higher. The 16 MB address is known as the line, so 24-bit storage is also called storage below the line. 24-bit storage and 31-bit storage are in storage below 2 GB and can together be referred to as storage below the bar.

31-bit storage. Virtual storage above the 16 MB address but below the 2 GB address in a z/OS address space. Addresses above the 16 MB address but below the 2 GB address are accessed by 31-bit addressing, and programs can use this storage when they run in AMODE 31 or higher.

The 16 MB address is known as the line, so 31-bit storage is also called storage above the line. 24-bit storage and 31-bit storage are in storage below 2 GB and can together be referred to as storage below the bar

64-bit storage. Virtual storage above the 4 GB address in a z/OS address space. Addresses above the 4 GB address are accessed by 64-bit addressing, and programs can use this storage when they run in AMODE 64.

The area that separates storage below the 2 GB address from the virtual storage above the 4 GB address is known as the bar, so 64-bit storage is also called storage above the bar.

Changes to external interfaces

Some external interfaces now support changes to CICS use of 64-bit storage.

Changes to resources and attributes

The CICSplex SM resource CICSSTOR is changed to support CICS use of 64-bit storage.

Support for updated resource: CICSSTOR

You can use the updated CICSSTOR resource to display information about CICS use of 64-bit storage.

The following new attributes are provided:

- SMSLVABYTES. Number of bytes allocated from large virtual memory in memory objects.
- SMSLVHBYTES. Number of bytes hidden with large virtual memory objects.
- SMSLVGBYTES. High watermark for number of usable bytes within large virtual memory objects.
- SMSLVNMOMBS. Number of memory objects allocated.
- SMSHVAXSLTS. Number of auxiliary slots used to back 64-bit private storage.
- SMSHVGAXSTS. High watermark for the number of auxiliary slots used to back 64-bit private storage.
- SMSLVSHRBTS. Number of shared bytes allocated from high virtual memory.
- SMSLVSHRGBS. High watermark for number of shared bytes within large virtual memory objects.
- SMSLVSHRNMO. Number of shared memory objects allocated.
- SMSFRMGRDFL. Number of IARV64 CONVERT(FROMGUARD) failures.
- SMSFRMGFLSZ. The largest failing IARV64 CONVERT(FROMGUARD) failure request size (in bytes).
- SMSGDSAALOC. Current GDSA allocated.
- SMSHGDSAALC. Peak GDSA allocated.
- SMSPGSINREAL. Number of real storage frames used to back 64-bit private memory.
- SMSHPGSNREAL. High watermark of real storage frames used to back 64-bit private memory objects.
- SMSLRGMEMOBJ. Number of large memory objects allocated by this address space.
- SMSLRGPGBNRL. Number of large pages (1 MB pages) backed in real storage owned by this address space.
- SMSCSIZE. Current cushion size.
- SMSCREL. Number of cushion releases.
- SMSFSTG. Free storage.
- SMSHWMFSTG. High watermark of free storage (including cushion).
- SMSLWMFSTG. Low watermark of free storage (including cushion).
- SMSLFA. Largest free area in DSA.

The following attributes are removed:

- SMSATBCUSHRE
- SMSATBCUSHLI

The updated CICSSTOR resource has the following changes:

- Updated CICSplex SM view EYUSTARTCICSSTOR, **CICS operations > CICS region operations views > Dynamic storage area global**.
- Updated CICSplex SM resource table, CICSSTOR

Changes to statistics

The CICS storage manager statistics now provide additional information about 64-bit storage.

Changed statistics types: Storage manager

The storage manager global statistics, mapped by the DFHMSDS DSECT, now include the following statistics:

- The number of bytes allocated from large virtual memory in private memory objects
- The number of bytes hidden in large virtual memory private memory objects
- The high watermark of usable bytes in large virtual memory private memory objects
- The number of private memory objects allocated
- The number of auxiliary storage slots that are used to back 64-bit private memory objects
- The high watermark of auxiliary storage slots that are used to back 64-bit private memory objects
- The number of real storage frames that are used to back 64-bit private memory objects
- The high watermark for the number of real storage frames that are used to back 64-bit private memory objects
- The number of large memory objects allocated by this address space.
- The number of large pages (1 MB pages) backed in real storage owned by this address space
- The number of shared bytes allocated from high virtual memory
- The high watermark for number of shared bytes in large virtual memory objects
- The number of shared memory objects allocated
- The number of times that a request for 64-bit storage has failed, where the request uses the z/OS IARV64 macro with the REQUEST=CHANGEGUARD, CONVERT=FROMGUARD parameters
- The size of the largest request for 64-bit storage that has failed, in bytes, where the request uses the z/OS IARV64 macro with the REQUEST=CHANGEGUARD, CONVERT=FROMGUARD parameters
- Total amount of storage currently allocated to the DSAs above the bar
- The peak amount of storage allocated to the DSAs above the bar

The storage manager dynamic storage areas statistics, mapped by the DFHMSDS DSECT, now include the following statistics:

- The size of the cushion in the GCDSA
- The number of times a GETMAIN request caused the storage cushion to be released in the GCDSA
- The peak amount of free storage in the GCDSA
- The smallest amount of free storage in the GCDSA
- The length of the largest contiguous free area in the GCDSA

The reports produced by the DFHSTUP and DFH0STAT statistics programs show the new statistics.

Changes to problem determination

CICS can now use 64-bit (above-the-bar) storage instead of 31-bit (above-the-line) storage for message tables, and several new messages and trace points have been added.

New messages

The following messages relate to the process of loading and using message modules in 64-bit storage.

- DFHME0103
- DFHME0213
- DFHME0215
- DFHME0217
- DFHME0218
- DFHME0220
- DFHME0222
- DFHME0223
- DFHME0225
- DFHME0232
- DFHME0237
- DFHME0240

Changed messages

- Message DFHSM0602 is changed to support the new minimum value for the z/OS **MEMLIMIT** parameter.
- Message DFHSM0606 is changed to provide an updated user response.

Discontinued messages

- Message DFHSM0603 is discontinued.

Changed abend codes

- Abend APGC is changed to support the new limit of 5% of the **MEMLIMIT** value per transaction for channels and containers.

New trace points

The following new trace point relates to the use of 64-bit storage:

- KE 050A

The following new trace points relate to the process of loading and using message modules in 64-bit storage:

- ME 0117
- ME 0701 - 0715
- ME 0736 - 0738
- ME 0742
- ME 0745
- ME 0746

The following new trace points relate to the management of 64-bit storage:

- SM 0C1B
- SM 4C01 - 4C08
- SM 4D01 - 4D12
- SM 4E01 - 4E05

Chapter 28. Changes to storage use for CICS trace

CICS can now use 64-bit (above-the-bar) storage instead of 31-bit (above-the-line) storage for the transaction dump trace table, the internal trace table, and the generalized trace facility (GTF) buffer.

The transaction dump trace table is now created in 64-bit storage.

The internal trace table can be created in 64-bit storage rather than 31-bit storage, depending on the version of the z/OS operating system and whether the CICS region operates with transaction isolation. See CICS facilities that can use 64-bit storage in the Performance Guide.

If you specified a small size for the trace tables before CICS TS for z/OS, Version 4.2 because of concerns about the availability of 31-bit storage, consider reviewing and increasing the size of your trace tables where 64-bit storage is used.

When you set sizes for trace tables in 64-bit storage, check your current setting for the z/OS parameter **MEMLIMIT**. **MEMLIMIT** limits the amount of 64-bit storage that the CICS address space can use. Your trace table sizes must remain within the **MEMLIMIT** setting, and you must also allow for other facilities in the CICS region that use 64-bit storage.

If the total amount of virtual storage required for your CICS regions increases, you might need to review the amount of space allocated for supervisor call (SVC) dumps that are requested by CICS, and the amount of auxiliary storage available. For information about SVC dump data set management, see the *z/OS MVS Diagnosis: Tools and Service Aids*. For information about auxiliary storage management, see the *z/OS MVS Initialization and Tuning Guide*.

The two 4 KB buffers that are used for auxiliary trace data also have a new location. These buffers are now in 31-bit (above-the-line) storage instead of 24-bit (below-the-line) storage.

The GTF buffer can be allocated in 64-bit storage, rather than 31-bit storage, depending on the version of the z/OS operating system and whether the CICS region operates with transaction isolation. See CICS facilities that can use 64-bit storage in the Performance Guide.

To use CICS trace with GTF, you must have the PTF for APAR OA32611 applied to z/OS, Version 1 Release 11 or z/OS, Version 1 Release 12.

Changes to external interfaces

Some external interfaces now support the changes to storage use for CICS trace.

Changes to system initialization parameters

The storage that you specify with the **TRTRANSZ** system initialization parameter is now 64-bit storage. The storage that you specify with the **TRTABSZ** system initialization parameter can be 64-bit storage.

Changed system initialization parameters

- The **TRTABSZ** system initialization parameter specifies the size, in kilobytes, of the internal trace table.

CICS can obtain 64-bit (above-the-bar) storage, rather than 31-bit (above-the-line) storage for the internal trace table, depending on the version of the z/OS operating system, and whether the CICS region operates with transaction isolation. See CICS facilities that can use 64-bit storage in the Performance Guide.

When the internal trace table is in 64-bit storage, check your current setting for the z/OS parameter **MEMLIMIT**. **MEMLIMIT** limits the amount of 64-bit storage that the CICS address space can use. Your setting for **TRTABSZ** must remain within **MEMLIMIT**, and you must also allow for other use of 64-bit storage in the CICS region.

See **TRTABSZ** system initialization parameter in the System Definition Guide.

- The **TRTRANSZ** system initialization parameter specifies the size, in kilobytes, of the transaction dump trace table.

When a transaction dump is taken, CICS obtains MVS storage in 64-bit (above-the-bar) storage for the transaction dump trace table.

When you set this parameter, check your current setting for the z/OS parameter **MEMLIMIT**.

See **TRTRANSZ** system initialization parameter in the System Definition Guide.

Changes to statistics

In the statistics sample program DFH0STAT, the size currently set for the CICS internal trace table is now shown in both the Storage above 16 MB report and the Storage above 2 GB report. The location of the internal trace table is either in 31-bit storage or in 64-bit storage, depending on the version of the z/OS operating system and whether the CICS region operates with transaction isolation.

For further information, see CICS facilities that can use 64-bit storage in the Performance Guide.

Changes to problem determination

The global trap exit DFHTRAP now operates with 64-bit storage, and there are new messages and trace points.

Changes to the global trap exit DFHTRAP

The global trap exit DFHTRAP can be invoked when the CICS trace domain is called to write a trace entry. DFHTRAP is intended for use only with the guidance of IBM Service personnel.

DFHTRAP now runs in AMODE(64). It includes new fields that you use to trace data in 64-bit storage (above the bar). The DFHTRAP work area is now allocated in 64-bit storage.

New messages

The following new messages relate to the process of obtaining storage for the auxiliary trace buffers and the internal trace table:

- DFHTR0119
- DFHTR0122

- DFHTR0123
- DFHTR0124

Changed messages

The explanations for the following messages are changed to reflect the new locations for the auxiliary trace buffers and the 256-byte buffer for writing to GTF trace:

- DFHTR0104
- DFHTR0111

New trace points

The following new trace points relate to the process of obtaining storage for the auxiliary trace buffers and the internal trace table:

- TR 0210
- TR 0211
- TR 0222
- TR 0223
- TR 0224

The following new trace point relates to the process of obtaining storage for the transaction dump trace table:

- DU 0257

New trace selection parameter

For internal trace in a system dump, you can use the following trace selection parameter to select a specified number of blocks at the end of the internal trace table for formatting and printing.

- LAST_BLOCKS

Chapter 29. Changes to the overall limits for the CICS extended dynamic storage areas

The minimum and default **EDSALIM** values have changed to 48 MB to ensure that there is sufficient storage for CICS initialization.

The **EDSALIM** system initialization parameter specifies the upper limit of the total amount of storage within which CICS can allocate the individual extended dynamic storage areas (EDSAs) that reside in 31-bit (above-the-line) storage; that is, above 16 MB but below 2 GB.

Changes to external interfaces

Some external interfaces now support changes to the overall limits for the CICS extended dynamic storage areas (EDSAs).

Changes to system initialization parameters

The minimum and default **EDSALIM** values have changed to 48 MB.

Changed system initialization parameter, **EDSALIM**

The **EDSALIM** system initialization parameter specifies the upper limit of the total amount of storage within which CICS can allocate the individual extended dynamic storage areas (EDSAs) that reside in 31-bit (above-the-line) storage; that is, above 16 MB but below 2 GB.

See **EDSALIM** system initialization parameter in the System Definition Guide.

Chapter 30. Changes to the number of LSR pools

You can now define up to 255 LSR pools using the LSRPOOLNUM attribute in the FILE and LSRPOOL resource definitions.

The existing attribute LSRPOOLID in the FILE and LSRPOOL resource definitions has been replaced with LSRPOOLNUM to increase the maximum number of LSR pools available in a CICS region from 8 to 255.

The LSRPOOL resource defines the size and characteristics of the local shared resource (LSR) pool. Increasing the number of LSR pools available in a CICS region can improve transaction response times for CICS workloads that use VSAM files.

For example an application that processes an entire data set every time the transaction is used can cause excessive I/O operations. Placing the data set into its own LSR pool can help eliminate I/Os and prevent buffer stealing in a shared LSR pool.

You can also slow down transactions that are causing buffer steals in a shared VSAM LSR pool by placing the VSAM data sets in a detuned LSR pool to limit the number of data component buffers. In many cases a dedicated LSR pool performs better than a shared LSR pool. Dedicated LSR pools are also an excellent replacement for CICS data tables, when the capacity of the CICS data tables has been exceeded.

You can optimize CICS workloads that use VSAM files; for more information about tuning VSAM files to improve performance see VSAM tuning: general objectives in the Performance Guide.

CICS provides statistics to help you decide how and when to use LSR pools for the VSAM files. VSAM LSR pools can be monitored with response times, paging rates, and **CICS LSRPOOL** statistics. The **CICS LSRPOOL** statistics show buffer and string usage, data set activity, and buffer lookasides. For more information, see LSRpool statistics in the Performance Guide.

Changes to external interfaces

Some external interfaces now support changes to the maximum number of LSR pools.

Changes to system initialization parameters

The system initialization parameter **CSDLSRNO** has been changed to associate a CSD with a number of an LSR pool in the range 1 - 255.

Changed system initialization parameter, **CSDLSRNO**

Before CICS TS for z/OS, Version 4.2, you specified an LSR pool number in the range 1 - 8. In CICS TS for z/OS, Version 4.2 the number of LSR pools that can be defined has increased to 255.

Changes to resources and attributes

You can use the new LSRPOOLNUM attribute on the FILE and LSRPOOL resource definitions to specify LSR pool numbers in the range 1 - 255.

Support for updated resources: FILE and LSRPOOL

With the **CREATE FILE**, **SET FILE**, **INQUIRE FILE**, and **CREATE LSRPOOL** commands you can set the number of LSR pools on the FILE and LSRPOOL resource definitions.

The updated FILE and LSRPOOL resources have the following support:

- Updated Explorer SM Definitions views **File Definitions view** and **LSR Pool Definitions view**: use the new LSRPOOLNUM attribute to view the LSR pool number.
- Updated Explorer SM Operations view **Files view (local)**: use the new LSRPOOLNUM attribute to view the LSR pool number.
- Updated CICSplex SM views **CICS resource definitions > File definitions - FILEDEF**, and **CICS resource definitions > LSR pool definitions - LSRDEF**: use the changed LSRPOOLID attribute to view the LSR pool number.
- Updated CICSplex SM views **File operations views > Local files - LOCFILE**, **File operations views > VSAM LSR pools - LSRPOOL**, and **File operations views > VSAM LSR pool buffers - LSRPBUF**: use the new LSRPOOLNUM attribute to view the LSR pool number.
- Updated CICSplex SM tables LSRDEF, FILEDEF, CMDT, LOCFILE, LSRPOOL, LSRBUF, MCMDT, MLOCFILE, MLSRPOOL, and MLSRPBUF: view or set the LSR pool number.
- Updated **CREATE FILE**, **INQUIRE FILE**, **SET FILE**, and **CREATE LSRPOOL** commands: use the new LSRPOOLNUM option to view and set the number of LSR pools.

FILE and LSRPOOL resource updates

The following interfaces are updated to support the FILE and LSRPOOL resources:

CICS Explorer

-  Definition views
-  Operations view

CICSplex SM

-  File definitions - FILEDEF in CPSM Administration Views
-  LSR pool definitions - LSRDEF in CPSM Administration Views
-  Local files - LOCFILE in the CICSplex SM Operations Views Reference
-  VSAM LSR pools - LSRPOOL in the CICSplex SM Operations Views Reference
-  VSAM LSR pool buffers - LSRPBUF in the CICSplex SM Operations Views Reference
-  LSRDEF Resource Table in the CICSplex SM Resource Tables Reference
-  FILEDEF Resource Table in the CICSplex SM Resource Tables Reference
-  CMDT Resource Table Reference in the CICSplex SM Resource Tables Reference

- [LOCFILE Resource Table in the CICSplex SM Resource Tables Reference](#)
- [LSRPOOL Resource Table in the CICSplex SM Resource Tables Reference](#)
- [LSRPBUF Resource Table in the CICSplex SM Resource Tables Reference](#)
- [MCMDT Resource Table in the CICSplex SM Resource Tables Reference](#)
- [MLOCFILE Resource Table in the CICSplex SM Resource Tables Reference](#)
- [MLSRPOOL Resource Table in the CICSplex SM Resource Tables Reference](#)
- [MLSRPBUF Resource Table in the CICSplex SM Resource Tables Reference](#)

The CICS SPI

- [CREATE FILE in CICS System Programming Reference](#)
- [SET FILE in CICS System Programming Reference](#)
- [INQUIRE FILE in CICS System Programming Reference](#)
- [CREATE LSRPOOL in CICS System Programming Reference](#)

Changes to statistics

The CICS statistics sample program, DFH0STAT, and the statistics utility program, DFHSTUP, now show values for LSR pool numbers in the range 1 - 255.

Changed statistics type: File control statistics

The CICS resource and summary statistics for file control, mapped by the DFHA17DS DSECT, now show LSR pool numbers up to 255.

Changed statistics type: LSRpool statistics

The CICS resource and summary statistics for LSRpools, mapped by the DFHA08DS DSECT, now show LSR pool numbers up to 255.

Changes to sample programs

The CICS sample statistics program DSF0STAT, the RDO programs DFH0FORC, DFH£FORA, and DFH£FORP, and the DSECTs DFH£SQLT and DFH£DB2T, have changed to support the increase in the number of LSR pools.

Changed sample programs

The sample statistics program DFH0STAT has been enhanced to report on the increased number of LSR pools.

The RDO sample programs DFH0FORC, DFH£FORA, and DFH£FORP have been changed to incorporate the new LSRPOOLNUM keyword on the FILE and the LSRPOOL definitions.

DSECTs DFH£SQLT and DFH£DB2T have been changed to incorporate the new LSRPOOLNUM keyword on the FILE and the LSRPOOL definitions.

Changes to problem determination

CICS provides new information to help you diagnose problems relating to LSR pools.

Changed messages

- Changed error message DFHAM4868W reports problems with LSR pools.

Chapter 31. Enhancements to IMS Database Control (DBCTL)

CICS provides a CICS IMS Database control (CICS-DBCTL) interface for IMS to satisfy DL/I requests that are issued by applications running in a CICS region. In CICS TS 4.2 the CICS-DBCTL interface has been defined as threadsafe and CICS can run the CICS-DBCTL task-related user exit (TRUE) on an L8 open task control block (open TCB).

The open transaction environment (OTE) is supported from IMS version 12 with PTFs for APAR PM31420 applied. IMS indicates to CICS during the connection process that the OTE is supported and consequently CICS defines the CICS-DBCTL TRUE as an open API TRUE.

An open API TRUE is run on an L8 open TCB, which is dedicated for use by the calling CICS task. Running an application on an open TCB improves throughput and performance by reducing the use of the QR TCB. Threadsafe CICS applications that run on an L8 open TCB and use threadsafe CICS-DBCTL commands now avoid up to four TCB switches for each call to IMS. For more information about CICS IMS applications and the OTE, see Enabling CICS IMS applications to use the open transaction environment (OTE) through threadsafe programming.

If your IMS version does not support the OTE, CICS runs the CICS-DBCTL TRUE on the QR TCB.

When using IMS Version 12 or later, you must change the way you calculate CICS and IMS processor times, for more information see, DBCTL monitoring data returned to CICS.

Note: IMS Version 12 is available through a Quality Partnership Program (QPP). For more information about IMS Version 12, visit <http://www.ibm.com/software/data/ims/>.

To use IMS Versions 10 and 11 with CICS TS 4.2 you must apply the following IMS APARs:

- For IMS Version 10, PTFs for APAR PM31730 are needed.
- For IMS Version 11, PTFs for APAR PM31729 are needed.

Changes to external interfaces

Some external interfaces now support enhancements to IMS Database Control (DBCTL).

Changes to task-related user exits

The definitions for the CICS-DBCTL adapter task-related user exit (TRUE) DFHDBAT, and the EXEC DLI TRUE DFHEDP have been changed to specify CONCURRENCY(THREADSAFE). However DFHDBAT only uses the open transaction environment (OTE) when the release level of IMS supports the OTE, so the definition is overridden at run time by ENABLE options.

If IMS supports the OTE then when CICS connects to DBCTL, DFHDBAT is enabled as OPENAPI and uses an open TCB. If IMS does not support the OTE then DFHDBAT is enabled as QUASIRENT meaning DFHDBAT is started on the

QR TCB and the IMS database resource adapter (DRA) switches to use an IMS thread TCB when processing an IMS request.

Chapter 32. Concurrency in program resource definitions

CICS provides a new program concurrency setting **REQUIRED**, which enables a program that is coded to threadsafe standards to start on an open TCB.

Enabling programs to start on an open TCB reduces contention for resources on the CICS QR TCB and reduces TCB switching. With the new **REQUIRED** setting, if CICS switches to the QR TCB to run a CICS command, it switches back to the open TCB before handing control back to the application program.

The type of open TCB that is used depends on the API setting and the language of the program.

- Java programs using pooled JVMs operate like OPENAPI programs and use a J8 TCB if CICS key is set, and a J9 TCB if user key is set. Java programs using the JVMSERVER use a T8 TCB.
- C or C++ XPLink programs operate like OPENAPI programs and use an X8 TCB if CICS key is set, and an X9 TCB if user key is set.
- Cobol, PL/I, non-XPLink C or C++, and assembly language programs defined with **CONCURRENCY(REQUIRED)** and **API(CICSAPI)** start on an L8 open TCB without matching the TCB with the execution key of the program. This setting is suitable for programs that access resource managers like DB2[®] and WebSphere MQ, which also require an L8 TCB. For OPENAPI programs, CICS uses an L9 TCB if **EXECKEY(USER)** is set and an L8 TCB if **EXECKEY(CICS)** is set.

For more information about the different combinations of **CONCURRENCY** and **API** on the **PROGRAM** resource definition and the type of TCB used, see the table at **Multithreading: Reentrant, quasi-reentrant, and threadsafe programs**.

Changes to external interfaces

Some external interfaces now support the new option for concurrency in program resource definitions.

Changes to resources and attributes

You can use the new **CONCURRENCY(REQUIRED)** setting in the **PROGRAM** resource to enable a program coded to threadsafe standards to run on an open TCB from the start of the program.

Support for updated resource: **PROGRAM**

You can use the new **REQUIRED** setting on the **CONCURRENCY** attribute in the **PROGRAM** resource to enable threadsafe programs to start on an open TCB. With the **CREATE PROGRAM**, **INQUIRE PROGRAM**, and **CEMT INQUIRE PROGRAM** commands you can query and set the concurrency on the program resource definition.

The updated **PROGRAM** resource has the following support:

- Updated Explorer SM Definitions views **Program Definitions view**: use the **CONCURRENCY** attribute to view the concurrency setting on the program.

- Updated Explorer SM Operations view **Programs view**: use the CONCURRENCY attribute to view the concurrency on the program resource definition.
- Updated CICSplex SM views **Administration views > CICS resource definitions > PROGRAM definitions - PROGDEF**: use the CONCURRENCY attribute to set the program concurrency.
- Updated CICSplex SM views **CICS operations views > Program operations views > Programs - PROGRAM**: use the CONCURRENCY attribute to view the program concurrency.
- Updated CICSplex SM tables PROGDEF and PROGRAM: use the CONCURRENCY attribute to view the program concurrency.
- Updated CREATE PROGRAM, INQUIRE PROGRAM, and CEMT INQUIRE PROGRAM commands: use the CONCURRENCY attribute to view or set the program concurrency.

PROGRAM resource updates

The following interfaces are updated to support the PROGRAM resource:

CICS Explorer

-  Definition views
-  Operations view

CICSplex SM

-  Program definitions - PROGDEF
-  Programs - PROGRAM
-  PROGDEF Resource Table
-  PROGRAM Resource Table

CEMT

-  CEMT INQUIRE PROGRAM

The CICS SPI

-  EXEC CICS CREATE PROGRAM
-  EXEC CICS INQUIRE PROGRAM

Changes to user-replaceable programs

The new REQUIRED concurrency setting on the program resource definition is applicable to user-replaceable programs.

Changes to statistics

The CICS statistics for program definition now include the new program concurrency setting REQUIRED. The statistics sample program, DFH0STAT, and the statistics utility program, DFHSTUP, also return the new concurrency setting.

Changed statistics types: Program definition statistics

The CICS global statistics for program definition, mapped by the DFHPGDDS DSECT, now include the program concurrency setting REQUIRED.

Part 6. Discontinued functions

Some functions which were supported in CICS Transaction Server for z/OS, Version 4 Release 1 have been discontinued, or reduced in scope, in Version 4 Release 2.

Chapter 33. Removal of support for CICS Events for WebSphere Business Events SupportPac CB11

Support for CICS Events for WebSphere Business Events SupportPac CB11 is withdrawn in CICS TS for z/OS, Version 4.2.

You can migrate events that use the SupportPac over to the non-invasive event specifications that are used in CICS TS for z/OS, Version 4.2. The SupportPac documentation explains the advantages of migrating to the non-invasive event support provided by CICS Transaction Server for z/OS versions 4.1 and 4.2, and Chapter 12, *Migrating to V4.1 event support* describes how to perform the migration. For further information on CICS Events for WebSphere Business Events SupportPac CB11 version 1.1, see CB11: CICS Events for WebSphere Business Events.

Part 7. Changes to the CICS Explorer

Since the release of CICS Transaction Server for z/OS, Version 4 Release 1, there are a number of significant enhancements to the CICS Explorer.

Table 2. CICS Explorer releases and enhancements

CICS Explorer release	Enhancement
V1.1.1.0 - May 2012	<ul style="list-style-type: none">• A new infrastructure to manage system connections• Support for CICS TS for VSE/ESA• Support of the z/OS Management Facility (z/OSMF)• Ability to copy and paste resource data to a spreadsheet• Ability to export a file or folder to a z/OS UNIX file system• Ability to create a z/OS UNIX file• Ability to create a new data set• Ability to create a new resource from an existing definition• Ability to see the contents resource definition group• Support for multiple selection job submission from the z/OS Data Sets view• Support for deleting or purging multiple files in the z/OS Explorer• Ability to create a Job Submission project and JCL files, in the Project Explorer view in the resource perspective, while working offline• You can now use Installation Manager to install the CICS Explorer SDK. However if you do use this method to install CICS Explorer, you must install all future updates using Installation Manager.

Table 2. CICS Explorer releases and enhancements (continued)

CICS Explorer release	Enhancement
V1.1.0.0 - June 2011	<ul style="list-style-type: none"> • Changes to the CICS Explorer menu bar. • New SM Administration perspective. • Improved support for transaction tracking. • Updates to the field names in the Task view. • Support for column sorting in views. • System z/OS perspective is renamed to the z/OS perspective. • Ability to add folders in a z/OS UNIX file system, and to change the access permissions on a z/OS UNIX file or folder. • Addition of a Save as function for z/OS UNIX files. • Support for system events in the CICS event binding editor. • Ability to choose an EPADAPTER configuration in the event binding editor. • Ability to choose an emission mode for all adapter types in the event binding editor. • Addition of an online and offline EP search function. • Ability to copy event and capture specifications in the CICS event binding editor. • New editing functions in the event binding editor: undo, redo, copy, and paste. • Ability to create and edit Atom configuration files. • Ability to disable JVM servers. • CICS Explorer SDK support for creating Java OSGi applications. • Eight new workload management Operations views in the SM Administration perspective. <p>Note: The Active Workloads view is renamed to the Workloads view.</p>
V1.0.1. - September 2010	<ul style="list-style-type: none"> • New z/OS perspective containing new views to help you manage your system z artifacts. • New documentation topic - z/OS Quick Tips. • Ability to export and import system connection details. • New breadcrumb trail in Editor views. • Ability to view SIT parameters. • Ability to choose an EPADAPTER configuration. • Four new Operations views and one new Administration view added
V1.0.0.7 - June 2010	<ul style="list-style-type: none"> • Ability to create and edit Atom configuration files. • New Connection Credentials page in the Preferences view. • Ability to use a proxy server to connect to the internet for software updates. • Changes to the Preference Connections page for the new connection credentials function. • Two new Operations views and three new Administration views.

Table 2. CICS Explorer releases and enhancements (continued)

CICS Explorer release	Enhancement
V1.0.0.6 - April 2010	<ul style="list-style-type: none"> • New CORBASERVER operations view. • Ability to disable JVM Servers.
V1.0.0.5 - February 2010	<ul style="list-style-type: none"> • Enhancements to workload management support. • Ability to select a task that is not responding and search for related tasks. • Ability to create Atom configuration files. • Changes to the CICS Event binding editor to support the HTTP adapter and a choice of synchronous or asynchronous emission mode. • Addition of a new view: Task Association (TASKASSC).
V1.0.0.4 - December 2009	<ul style="list-style-type: none"> • Ability to copy and paste resource definitions, groups, and lists. • Ability to add BAS resource groups to a resource description, or a CSD group to a list. • Ability to append one list to another list. • Ability to delete perspectives. • Inclusion of BAS resource descriptions in the CICSplex repositories view. • Inclusion of a new CICS Explorer Quick Reference Guide. • Improvements to the actions dialog.
V1.0.0.3 - December 2009	Version 1.0.0.3 provided a fix for CICS Explorer APAR IZ65446.
V1.0.0.2 - 25 September 2009	<ul style="list-style-type: none"> • Ability to display groups, lists, and resources in CICS TS V4 CSD files. • Ability to create resource definitions in a single CICS region (CSD). • Ability to use the Shutdown option for a CICS region. • Ability to view the CSDGROUP attribute in resource definition views. • Addition of a new method - String <code>com.ibm.cics.model.ICICSDefinition.getCSDGroup()</code> - in the CICS Explorer SDK API.

Part 8. General information

General information about this release.

System requirements

The minimum required level of operating requirements for CICS Transaction Server are described at <http://www.ibm.com/software/htp/cics/tserver/sysreqs/>.

Chapter 34. High-level language support

This topic lists the high-level programming languages and compilers that are in service on z/OS and have CICS translator support and Language Environment® runtime support with CICS Transaction Server for z/OS, Version 4 Release 2.

All COBOL, PL/I, C, and C++ compilers listed in this topic can use the integrated CICS translator.

COBOL

Product name	PID	Translator	Run time
Enterprise COBOL for z/OS V4.1 and V4.2	5655-S71	Supported	Supported
Enterprise COBOL for z/OS V3.4	5655-G53	Supported	Supported

PL/I

Product name	PID	Translator	Run time
Enterprise PL/I for z/OS V3.7, V3.8, and V3.9	5655-H31	Supported	Supported
Enterprise PL/I for z/OS V4.1 and above	5655-W67	Supported	Supported

C and C++

Product name	PID	Translator	Run time
z/OS V1.10 and above XL C/C++	5694-A01 - Component of z/OS	Supported	Supported

Assembler

Product name	PID	Translator	Run time
High Level Assembler for MVS and VM and VSE V1.6 and above	5696-234	Supported	Supported

Java

Product name	PID	Run time
IBM 64-bit SDK for z/OS, Java Technology Edition, V6.0.1	5655-R32	Supported

Compilers and high-level language versions withdrawn from service

The compilers and high-level programming languages listed here have been withdrawn from service on z/OS. This topic states the level of CICS translator support and Language Environment runtime support that is provided with CICS Transaction Server for z/OS, Version 4 Release 2 for your existing programs that were compiled using these compilers.

For information about the availability and end of service dates for IBM products, see the IBM Software Support Lifecycle Web page at http://www.ibm.com/software/support/lifecycle/index_a_z.html.

Only the more recent COBOL, PL/I, C, and C++ compilers listed in this topic can use the integrated CICS translator. For more information about the integrated CICS translator, see Translation and compilation in CICS Application Programming.

Language Environment runtime required

Language Environment is a base element of z/OS and provides the common runtime environment for COBOL, PL/I, C, and C++. For more information about how CICS uses the services of Language Environment, see Programming languages and Language Environment in CICS Application Programming.

Pre-Language Environment compilers provided native runtime libraries for programming languages, but the native runtime libraries are not supported in CICS Transaction Server for z/OS, Version 4 Release 2. However, Language Environment provides object and load module compatibility for applications that are generated with certain pre-Language Environment compilers. Load modules that are created with these compilers and link-edited with their associated runtime libraries run compatibly with Language Environment without relinking. Also, object modules created with these compilers can be linked and run with Language Environment without recompiling.

COBOL

Product name	PID	CICS translator	Integrated CICS translator	Runtime environment
OS/VS COBOL (pre-Language Environment)	5740-CB1 5740-CB4 5740-LM1		Not supported	
VS COBOL II Release 3 or later (pre-Language Environment)	5668-022 5668-023 5668-958			

Product name	PID	CICS translator	Integrated CICS translator	Runtime environment
VS COBOL II Release 3 or later (with Language Environment runtime)	5668-022 5668-023 5668-958	Supported	Not supported	Supported
COBOL/370	5688-197			
COBOL for MVS & VM V1.2	5688-197			
COBOL for OS/390® & VM V2.1	5648-A25			
COBOL for OS/390 & VM V2.2	5648-A25	Supported		
Enterprise COBOL for z/OS and OS/390 V3.1 and V3.2	5655-G53			
Enterprise COBOL for z/OS V3.3	5655-G53			

PL/I

Product name	PID	CICS translator	CICS integrated translator	Run time
OS PL/I Optimizing Compiler Version 1 (pre-Language Environment)	5734-PL1	Not supported	Not supported	See note
Note: For OS PL/I Version 1, Release 3 (object modules) and Release 5.1 (load modules) are supported.				
OS PL/I Optimizing Compiler Version 2 (pre-Language Environment)	5668-909 5668-910 5668-911	Not supported	Not supported	Supported
SAA AD/Cycle® PL/I for MVS and VM	5688-235	Supported	Not supported	Supported
PL/I for MVS and VM V1	5688-235	Supported	Not supported	Supported
VisualAge® PL/I for OS/390 V2	5655-B22	Supported	Supported	Supported

Product name	PID	CICS translator	CICS integrated translator	Run time
Enterprise PL/I for z/OS V3.1, V3.2, V3.3, V3.4, V3.5, and V3.6	5655-H31	Supported	Supported	Supported

C and C++

Product name	PID	CICS translator	CICS integrated translator	Run time
C/370™ V1 (pre-Language Environment)	5688-040	Supported	Not supported	Supported
C/370 V2 (pre-Language Environment)	5688-187 5688-188	Supported	Not supported	Supported
SAA AD/Cycle C/370	5688-216	Not supported	Not supported	Supported
C/C++ for MVS/ESA	5655-121	Supported	Not supported	Supported
C/C++ for OS/390	5647-A01 - Component of OS/390	Supported	Not supported	Supported
C/C++ for z/OS and OS/390	5694-A01 - Component of z/OS	Supported	Not supported	Supported
z/OS V1.4, V1.5, and V1.6 C/C++	5694-A01 - Component of z/OS	Supported	Not supported	Supported
z/OS V1.7, V1.8, and V1.9 XL C/C++	5694-A01 - Component of z/OS	Supported	Supported	Supported

Assembler

Product name	PID	CICS translator	CICS integrated translator	Run time
High Level Assembler for MVS and VM and VSE V1.5	5694-234	Supported	Not supported	Supported

Java

Product name	PID	Run time
Java for OS/390 V1.1.8	5655-A46	These products are not supported with CICS TS 4.2. Java application bytecode created using these products should run on the supported SDK product, provided the application does not use deprecated APIs and is not affected by any incompatibilities introduced with Java version changes.
VisualAge for Java, Enterprise Edition for OS/390	5655-JAV	
IBM Developer Kit for OS/390, Java 2 Technology Edition, SDK 1.3.1	5655-D35	
IBM SDK for z/OS, Java Technology Edition, V1.4.2	5655-I56	
IBM SDK for z/OS, Java Technology Edition, V5	5655-N98	
IBM 31-bit SDK for z/OS, Java Technology Edition, V6	5655-R31	
IBM 64-bit SDK for z/OS, Java Technology Edition, V6	5655-R32	

Chapter 35. Where is the migration information?

If you are looking for information about moving from a previous release of CICS Transaction Server to this release, you can find this information in the Upgrading section of the navigation tree for the CICS Transaction Server product in the Information Center. This section was previously called Migrating.

In releases before CICS Transaction Server for z/OS, Version 4 Release 1, the information about upgrading from previous releases to the current release of CICS Transaction Server was called the *Migration Guide*. Although the term “migrate” was used in the CICS documentation to refer to the process of replacing an older release of CICS with a newer release, the industry-standard term for this process is “upgrade,” so the CICS documentation has been changed to use this term. “Migrate” is now used only to refer to the processes of moving data or applications to a different program or environment, or of moving from the use of one function or interface in CICS to the use of a different function or interface.

In the information center, you can find the information sets for upgrading from previous releases:

- Upgrading from Version 3 Release 1 overview
- Upgrading from Version 3 Release 2 overview
- Upgrading from Version 4 Release 1 overview

The same information sets are available in PDF format: The CICS library: PDF.

In the information center, CICS Transaction Server for z/OS, Version 3 Release 1 is the oldest release for which information is provided describing how to upgrade to CICS Transaction Server for z/OS, Version 4 Release 2. If you are upgrading from an earlier release of CICS, read the information about upgrading and about changes to functions that is provided in the documentation for any additional intervening releases.

Part 9. Publications

Chapter 36. 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-2624. 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:

- The Discovery Library Adapter for CICS
- Event processing
- CICS integration with WebSphere MQ

Books available as hardcopy

When you order CICS Transaction Server for z/OS, Version 4 Release 2, you will receive a small number of hardcopy books.

The hardcopy books are:

Memo to Licensees, GI13-0537
CICS Transaction Server for z/OS Program Directory, GI13-0536
CICS Transaction Server for z/OS What's New, GC34-6994
CICS Transaction Server for z/OS Licensed Program Specification, GC34-7040

You can order further copies of the following books, using the order number quoted above:

CICS Transaction Server for z/OS What's New

Publications available in Portable Document Format (PDF)

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, GI13-0565
CICS Transaction Server for z/OS What's New, GC34-7192
CICS Transaction Server for z/OS Upgrading from CICS TS Version 3.1, GC34-7188
CICS Transaction Server for z/OS Upgrading from CICS TS Version 3.2, GC34-7189
CICS Transaction Server for z/OS Upgrading from CICS TS Version 4.1, GC34-7190
CICS Transaction Server for z/OS Installation Guide, GC34-7171

Access to CICS

CICS Internet Guide, SC34-7173
CICS Web Services Guide, SC34-7191

Administration

CICS System Definition Guide, SC34-7185
CICS Customization Guide, SC34-7161
CICS Resource Definition Guide, SC34-7181
CICS Operations and Utilities Guide, SC34-7213
CICS RACF Security Guide, SC34-7179
CICS Supplied Transactions, SC34-7184

Programming

CICS Application Programming Guide, SC34-7158
CICS Application Programming Reference, SC34-7159
CICS System Programming Reference, SC34-7186
CICS Front End Programming Interface User's Guide, SC34-7169
CICS C++ OO Class Libraries, SC34-7162
CICS Distributed Transaction Programming Guide, SC34-7167
CICS Business Transaction Services, SC34-7160
Java Applications in CICS, SC34-7174

Diagnosis

CICS Problem Determination Guide, GC34-7178
CICS Performance Guide, SC34-7177
CICS Messages and Codes Vol 1, GC34-7175
CICS Messages and Codes Vol 2, GC34-7176
CICS Diagnosis Reference, GC34-7166
CICS Recovery and Restart Guide, SC34-7180
CICS Data Areas, GC34-7163
CICS Trace Entries, SC34-7187
CICS Debugging Tools Interfaces Reference, GC34-7165

Communication

CICS Intercommunication Guide, SC34-7172
CICS External Interfaces Guide, SC34-7168

Databases

CICS DB2 Guide, SC34-7164
CICS IMS Database Control Guide, SC34-7170
CICS Shared Data Tables Guide, SC34-7182

CICSplex SM books for CICS Transaction Server for z/OS

General

CICSplex SM Concepts and Planning, SC34-7196
CICSplex SM Web User Interface Guide, SC34-7214

Administration and Management

CICSplex SM Administration, SC34-7193
CICSplex SM Operations Views Reference, SC34-7202
CICSplex SM Monitor Views Reference, SC34-7200
CICSplex SM Managing Workloads, SC34-7199
CICSplex SM Managing Resource Usage, SC34-7198
CICSplex SM Managing Business Applications, SC34-7197

Programming

CICSplex SM Application Programming Guide, SC34-7194
CICSplex SM Application Programming Reference, SC34-7195

Diagnosis

CICSplex SM Resource Tables Reference Vol 1, SC34-7204
CICSplex SM Resource Tables Reference Vol 2, SC34-7205
CICSplex SM Messages and Codes, GC34-7201
CICSplex SM Problem Determination, GC34-7203

Licensed publications

None of the publications for CICS Transaction Server for z/OS, Version 4 Release 2 is licensed. Publications which were licensed in the past are unlicensed in this release:

CICS Diagnosis Reference
CICS Data Areas
CICS Supplementary Data Areas
CICS Debugging Tools Interfaces Reference

CICS family books

The CICS family books are not included in the Information Center for CICS Transaction Server for z/OS, Version 4 Release 2.

You can view these publications in earlier versions of the Information Center, or download them from the IBM Publications Center.

The CICS family books are:

CICS Family: Interproduct Communication, SC34-6853
CICS Family: Communicating from CICS on System/390, SC34-6854

Information about code page conversion that was previously included in these publications is now included in the *CICS Intercommunication Guide*

Part 10. Appendixes

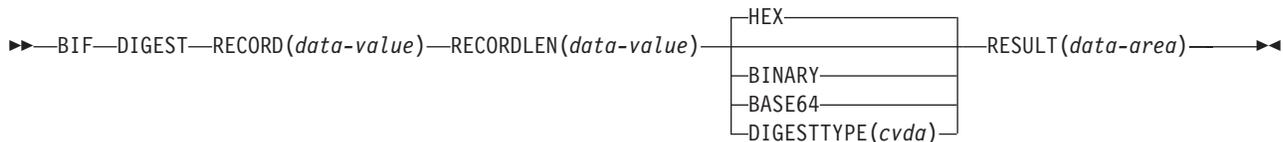
Appendix A. New application programming commands

CICS Transaction Server for z/OS, Version 4 Release 2 extends the CICS application programming interface with new commands.

BIF DIGEST

Calculate the SHA-1 digest of a string of data.

BIF DIGEST



Conditions: INVREQ, LENGERR

This command is threadsafe.

Description

The **BIF DIGEST** command is a CICS built-in function that calculates the SHA-1 digest of a string of data. The result can be returned as binary (20 bytes long), hexadecimal (40 bytes long), or base64-encoded (28 bytes long). The SHA-1 digest is a cryptographically strong checksum of the string, so for practical purposes it is unique for each string.

This command uses z/Architecture message security assist (MSA) functions which require System z cryptographic hardware with CP Assist for Cryptographic Functions (CPACF). For more information, see the *z/OS Cryptographic Services Integrated Cryptographic Service Facility Overview*.

Options

RECORD(*data-value*)

Specifies the string of data for which the digest is to be calculated.

RECORDLEN(*data-value*)

Specifies the length, as a fullword binary value, of the data string.

DIGESTTYPE(*cvda*)

Specifies the format in which the digest is returned.

HEX Hexadecimal, which produces a result 40 bytes long, encoded as hexadecimal characters (0 - 9, A - F).

BINARY

Binary, which produces a result 20 bytes long.

BASE64

Base64 encoding, which produces a result 28 bytes long, using the characters A - Z, a - z, 0 - 9, +, /, =.

RESULT (*data-area*)

Returns the SHA-1 digest of the data string in the format specified by the DIGESTTYPE option. The length of the result depends on the requested format.

Conditions

16 INVREQ

RESP2 values:

- 1 DIGESTTYPE has an invalid CVDA value.
- 3 z/Architecture message security assist (MSA) is not available.

22 LENGERR

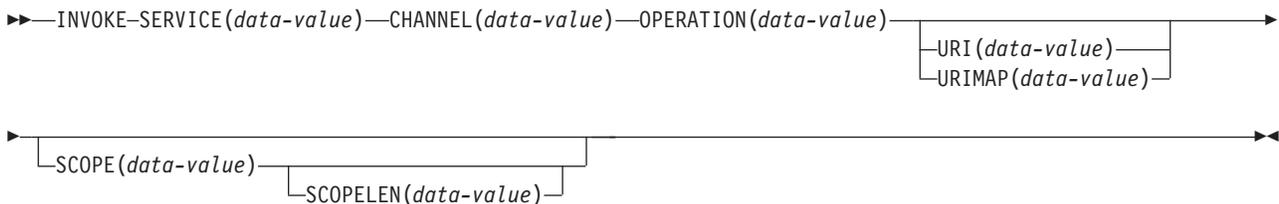
RESP2 values:

- 2 The RECORDLEN value is less than 1.

INVOKE SERVICE

Call a service from a CICS application. The command specifies the name of a service or the CICS resource, such as a WEBSERVICE resource, that contains information about the service to be called.

INVOKE SERVICE



Conditions: INVREQ, LENGERR, NOTFND, TIMEDOUT

This command is threadsafe.

Description

Use the **INVOKE SERVICE** command in CICS applications to call a service; for example, the application can act as a web service requester and call an XML-based service, or the application can call another CICS application that is acting as a channel-based service. For more information about these two types of services, see the *CICS Application Programming Guide*.

Use this command for all new web service applications, rather than the **INVOKE WEBSERVICE** command, which is a synonym. If you use this command for web services, you must supply specific containers to CICS as input. For more information on writing a web service application, see the *CICS Web Services Guide*.

When you invoke the service, you can specify a URIMAP resource that contains the information about the URI of the service. You can specify this information directly on the INVOKE SERVICE command instead of using a URIMAP resource. However, using a URIMAP resource has the following advantages:

- System administrators can manage any changes to the endpoint of the connection, so you do not need to recompile your applications if the URI of a service provider changes.
- You can choose to make CICS keep the connections that were opened with the URIMAP resource open after use, and place them in a pool for reuse by the application for subsequent requests, or by another application that calls the same service. Connection pooling is only available when you specify a URIMAP resource that has the SOCKETCLOSE attribute set. For more information about the performance benefits of connection pooling, see Connection pooling for HTTP client performance in the Internet Guide.

The **INVOKE SERVICE** command drives the XWBOPEN user exit, which can make the connection to the server go through a proxy server, if required.

Options

CHANNEL(*data-value*)

Specifies the name of the channel used to pass the containers that hold the data mapped by the application data structure. On return, the same channel holds the response from the web service, again mapped by the application data structure. The name of the channel can be up to 16 characters. If *name* is a variable, and it contains a name that is less than 16 characters, then the variable must be padded with trailing blanks.

OPERATION(*data-value*)

Specifies a data area containing the name of the operation that is to be invoked. The name of the operation is contained in the WSDL for the target web service. The data area must be 255 characters long; if the operation name is less than 255 characters, then the data area must be padded with trailing blanks.

SERVICE(*data-value*)

Specifies the name of the service:

- If you want to invoke a web service, specify the name of the WEBSERVICE resource that defines the web service. The WEBSERVICE resource specifies the location of the web service description and the web service binding file that CICS uses when it communicates with the web service. The name of the WEBSERVICE resource can be up to 32 characters. The value must be padded with trailing blanks if there are less than 32 characters.
- If you want to invoke a channel-based service, specify the name of the service. The format of the service is a URI. The name can be up to 32 characters. The value must be padded with trailing blanks if there are less than 32 characters.

SCOPE(*data-value*)

Specifies a scope prefix for the service name. Use the scope if you require a service name that is longer than 32 characters.

If you are writing a COBOL program that is translated with the COBOL3 translator option, the length of the data value cannot exceed 160 bytes. If you are using the COBOL2 translator option, you must use a data area instead of a data value.

SCOPELEN(*data-value*)

A fullword binary value that specifies the length of the scope that prefixes the service name.

URI(*data-value*)

Specifies a data area containing the URI of the service to be invoked. If

specified, this option supersedes any URI specified in the WEBSERVICE resource definition. If you omit both this option and the URIMAP option, the WEBSERVICE binding file associated with the resource definition must include either a provider URI or a provider application name. The data area must be 255 characters long; if the URI is less than 255 characters, the data area must be padded with trailing blanks. For information about the format of URIs, see the topic "The components of a URL" in the *CICS Internet Guide*.

Do not specify this option for web services that use WS-Addressing.

URIMAP(*data-value*)

Specifies the name of a URIMAP resource that CICS uses to derive the URI value. Use a URIMAP resource if you want to enable connection pooling, where CICS keeps the client HTTP connection open for this application or another application to reuse. If specified, this option supersedes any URI specified in the WEBSERVICE resource definition. If you omit both this option and the URI option, the WEBSERVICE binding file associated with the resource definition must include either a provider URI or a provider application name.

You must create the URIMAP resource for an HTTP client request with the attribute USAGE(CLIENT). For connection pooling, you must also set the SOCKETCLOSE attribute. The CICS web services assistant does not create the URIMAP resource, so you must define it yourself. For information about creating a URIMAP resource for a client request, see the topic "Creating a URIMAP definition for an HTTP request by CICS as an HTTP client" in the *CICS Internet Guide*.

Do not specify this option for web services that use WS-Addressing.

Conditions

16 INVREQ

RESP2 values:

- 1 The name specified for the CHANNEL option contains an illegal character or combination of characters.
- 2 The name specified for the OPERATION option contains an illegal character or combination of characters.
- 3 The web service binding file associated with the WEBSERVICE is invalid.
- 4 The value specified for the URI contained an illegal character or combination of characters, or the specified host name could not be resolved.
- 5 The PIPELINE used by the WEBSERVICE is defined as a service requester pipeline but is invoked in a service provider or *vice versa*.
- 6 The invoked WEBSERVICE returned a SOAP fault. The description of the fault is available in its XML format in the container DFHWS-BODY.

Note: This condition is not raised for XML-ONLY web service invocations.
- 7 The URI option was not specified on the command, and the WEBSERVICE definition does not specify a URI or a program name.
- 8 The WEBSERVICE is not in service
- 9 A container does not have the correct DATATYPE. This may be the **DFHWS-DATA** container, or another container referenced in the application

data. The **DFHWS-DATA** container and most other application data containers must be populated in BIT mode. Any containers that hold XML markup must be populated in CHAR mode.

- 10 The PIPELINE used by the WEBSERVICE is not enabled.
- 11 CICS could not link to the program specified in the WEBSERVICE definition.
- 12 The containers that the command expects were not on the correct channel.
- 13 An input error was detected either generating a SOAP request message or processing a SOAP response message. A DFHPIxxx message is written to MSGUSR to document the problem in more detail. It is likely that the application data structure contains invalid data that cannot be converted to a SOAP request message. For more information, see the error message in the DFH-XML-ERRORMSG container.
- 14 A conversion error occurred when CICS attempted to convert between the application data structure and the SOAP message. Either the application data structure contains invalid data that cannot be converted to a SOAP request, or data in the SOAP response message cannot be converted into the application's data structure. Some possible causes of this condition are:
- A value in the SOAP response message is larger than the corresponding field in the application's data structure.
 - When building the SOAP request, the web services binding file indicates that a data field contains packed decimal or zoned decimal data, and the contents of the field are invalid for this data type.
- A DFHPIxxx message is written to MSGUSR to document the problem in more detail. For more information, see the error message in the DFH-XML-ERRORMSG container.
- 15 An unhandled error has occurred in the pipeline. Information about the error is in container DFHERROR.
- 16 A locally optimized web service has abended. The underlying unit of work has been backed out.
- 17 A remote web service request did not return a response message.
- 18 The container **DFHWS-BODY** has not been populated by an application for an XML-ONLY WEBSERVICE.
- 19 A URI or a URIMAP has been specified, but this option is not allowed when the WEBSERVICE resource has a default WS-Addressing endpoint reference or the WS-Addressing context has been built using the **WSACONTEXT BUILD** API command.
- 20 The specified URIMAP does not have a valid scheme.
- 21 The specified URIMAP is not client mode.
- 22 The specified URIMAP is not enabled.
- 23 An unspecified transport or link failure occurred when attempting to use the pipeline. CICS issues a message to document the specific problem.
- 101 The container **DFHWS-BODY** does not have the correct DATATYPE. For this container, a DATATYPE of CHAR must be specified.

- 103 The container **DFHWS-BODY** contains no data.
- 104 Either the container **DFHREQUEST** or the container **DFHWS-BODY** is missing.
- 105 A fault was built within the service requester PIPELINE used by the WEBSERVICE, either while the request was being sent, or while the response was being processed. This condition could indicate that a header processing program has issued a fault.
- 106 Either the generated SOAP request message was not well formed, or the SOAP response message was not well formed. This condition could indicate that the XML parser returned a fatal error code.
- 107 Either the generated SOAP request message was not a valid SOAP message, or the SOAP response message was not a valid SOAP message.

22 LENGERR

RESP 2 values:

- 1 Either the SCOPELEN option was not specified or it was not a valid value.

13 NOTFND

RESP2 values:

- 1 The web service binding file associated with the WEBSERVICE specifies the name of a SOAP message parsing program supplied by another product, but the parsing program could not be found.
- 2 The specified CHANNEL could not be located.
- 3 The specified OPERATION was not in the web service binding file.
- 4 The specified WEBSERVICE could not be located.
- 5 A CONTAINER specified in the web service binding file could not be located.
- 6 The specified URIMAP could not be located.

124 TIMEDOUT

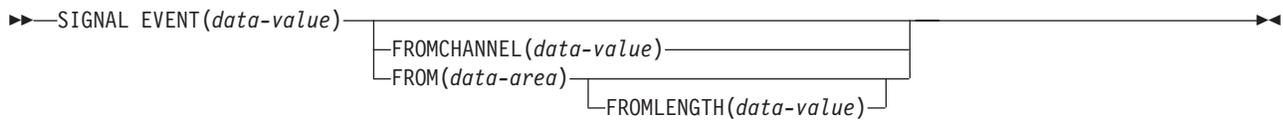
RESP2 values:

- 1 An expected timeout has occurred. When the message exchange pattern specifies an optional error response, and an error response is not returned from the remote web service, the timeout is acceptable.
- 2 An unexpected timeout has occurred. A response was expected from the remote web service, but none was received.
- 62 An unexpected timeout has occurred on socket receive.

SIGNAL EVENT

Identify a place in an application program where one or more events could be emitted.

SIGNAL EVENT



Conditions: CHANNELERR, EVENTERR, LENGERR

This command is threadsafe.

Description

The **SIGNAL EVENT** identifies a place in an application program where one or more events could be emitted. Events are emitted when the following conditions are satisfied:

- Event processing is active.
- There is at least one matching capture specification enabled. A capture specification matches if it has a capture point of **SIGNAL EVENT**, and all its predicates evaluate to true.

SIGNAL EVENT has a primary predicate of **EVENT**, and allows secondary predicates on the **FROM** data-area or the **FROMCHANNEL** and its containers. The data in any CICS Event emitted as a result of **SIGNAL EVENT** is defined in the Business Event that contains the matching capture specification.

Options

EVENT(data-value)

Specifies an identifier (1 – 32 characters) that identifies this **SIGNAL EVENT**.

The acceptable characters are A-Z a-z 0-9 \$ @ # / % & ? ! : | " = , ; < > . - and _ . Leading and embedded blank characters are not permitted. If the name supplied is fewer than 32 characters, it is padded with trailing blanks up to 32 characters.

Event identifiers are always in EBCDIC. The allowable set of characters for event identifiers, listed above, includes some characters that do not have the same representation in all EBCDIC code pages. We therefore recommend that the characters used in naming them should be restricted to A-Z a-z 0-9 & : = , ; < > . - and _ .

FROMCHANNEL(data-value)

Specifies the name (1 – 16 characters) of a channel containing data for this event.

FROM(data-area)

Specifies a data area containing data for this event.

FROMLENGTH(data-value)

Specifies a fullword binary value that is the length in bytes of the **FROM** data area.

Conditions

CHANNELERR

RESP2 values:

- 2 The channel specified on the FROMCHANNEL option could not be found.

EVENTERR

RESP2 values:

- 6 The identifier specified on the EVENT option contains an incorrect character or combination of characters.

LENGERR

RESP2 values:

- 3 The length that you have specified in FROMLENGTH is not greater than zero.

TRANSFORM DATATOXML

Use the **TRANSFORM DATATOXML** command to convert application data to XML.

TRANSFORM DATATOXML

```

▶▶—TRANSFORM DATATOXML—CHANNEL(data-value)—DATCONTAINER(data-value)—————▶
▶┌ELEMNAME(data-area)—ELEMNAMELEN(data-area)┐┌ELEMNS(data-area)—ELEMNSLEN(data-area)┐┌
▶└TYPE(data-area)—TYPELEN(data-area)—TYPENS(data-area)—TYPENSLEN(data-area)┘┘
▶┌XMLCONTAINER(data-value)┐—XMLTRANSFORM(name)—————▶▶

```

Conditions: NOTFND, LENGERR, CHANNELERR, CONTAINERERR, INVREQ

This command is threadsafe.

Description

The **TRANSFORM DATATOXML** command transforms application data to XML using mappings that are defined in the XML binding. The XMLTRANSFORM resource defines where the XML binding and the XML schema are located.

Options

CHANNEL(*data-value*)

Specify the name of the channel used to pass the containers holding the XMLCONTAINER and DATCONTAINER data. The name of the channel must be 16 characters in length. If the channel name is less than 16 characters, you must pad the data value with trailing blanks.

DATCONTAINER(*data-value*)

Specify the name of the container that contains the application data for conversion. This container must be present on the channel. The name of the container must be 16 characters in length. If the container name is less than 16 characters, you must pad the data value with trailing blanks.

CICS reads from this container in BIT mode.

ELEMNAME(*data-area*)

Specify the name of an XML element. CICS returns the local name of the XML element that CICS generates.

ELEMNAMELEN(*data-area*)

Specify the fullword binary length of the XML element in the ELEMNAME option. The maximum value of ELEMNAMELEN is 255.

ELEMNS(*data-area*)

Specify the namespace URI of the XML element that is returned in the ELEMNAME option.

ELEMNSLEN(*data-area*)

Specify the fullword binary length of the namespace in the ELEMNS option. The maximum value of ELEMNSLEN is 255.

TYPENAME(*data-area*)

Specify the xsi:type of the XML element that is returned in the ELEMNAME option.

TYPENAMELEN(*data-area*)

Specify the fullword binary length of the xsi:type that is returned in the TYPENAME option.

TYPENS(*data-area*)

Specify the namespace of the xsi:type attribute of the XML element that is returned in the ELEMNAME option.

TYPENSLEN(*data-area*)

Specify the fullword binary length of the namespace for the xsi:type attribute that is returned in the TYPENS option.

XMLCONTAINER(*data-value*)

Specify the name of the container that contains the output XML. The name of the container must be 16 characters in length. If the container name is less than 16 characters, you must pad the data value with trailing blanks.

You do not need to create the target container before issuing the command. The container is created and populated as part of the command itself. If the container does exist, and has been defined as a data type of BIT, it is deleted and redefined as type CHAR as part of the command.

XMLTRANSFORM(*data-value*)

Specify the name of the XMLTRANSFORM resource that CICS uses to transform the data to XML. The resource defines the XML binding and the XML schema. The name of the XMLTRANSFORM resource must be 32 characters. If the resource name is less than 32 characters, you must pad the value with blanks.

Conditions**NOTFND**

RESP2 values:

- 1 The XMLTRANSFORM was not found.

CHANNELERR

RESP2 values:

- 1 The channel specified by the **CHANNEL** parameter is incorrect.
- 2 The channel specified by the **CHANNEL** parameter was not found.

CONTAINERERR

RESP2 values:

- 1 The container specified by the **XMLCONTAINER** parameter was not found.
- 2 The container specified by the **NSCONTAINER** parameter was not found.
- 3 The container specified by the **DATCONTAINER** parameter was not found.

LENGERR

RESP2 values:

- 1 The data in the container specified by the **DATCONTAINER** parameter is too short for the specified transformation type.
- 2 The **ELEMNAME** buffer is too small.
- 3 The **ELEMNS** buffer is too small.
- 4 The **TYPENAME** buffer is too small.
- 5 The **TYPENS** buffer is too small.
- 6 The **ELEMNAMELEN** value exceeds the maximum value of 255.
- 7 The **ELEMNSLEN** value exceeds the maximum value of 255.

INVREQ

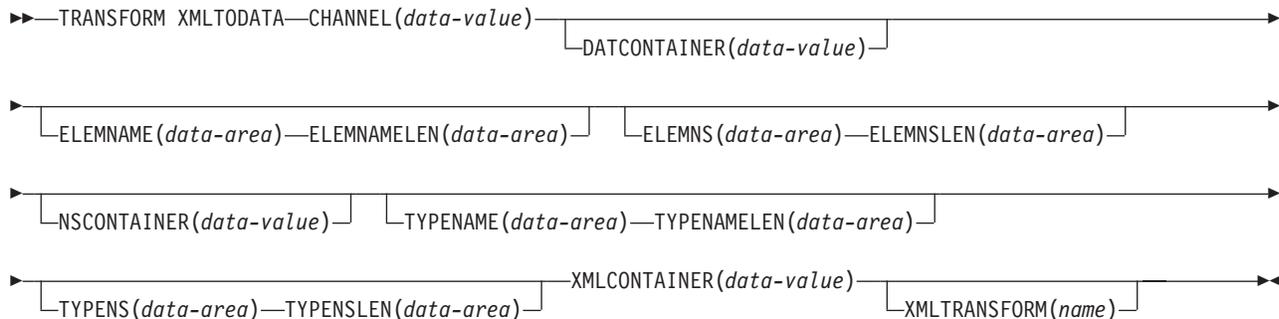
RESP2 values:

- 1 The XMLTRANSFORM resource is not enabled.
- 2 The **XMLCONTAINER** container is empty.
- 3 The XML input data is invalid. For more information, see the error message in the DFH-XML-ERRORMSG container.
- 4 The XML input data cannot be converted. For more information, see the error message in the DFH-XML-ERRORMSG container.
- 5 The application data is invalid. For more information, see the error message in the DFH-XML-ERRORMSG container.
- 6 The application data cannot be converted. For more information, see the error message in the DFH-XML-ERRORMSG container.
- 8 The application data container is not populated in BIT mode
- 9 The XMLTRANSFORM does not support the requested XML element.
- 10 The XMLTRANSFORM does not support the requested XML type.
- 11 There was a problem linking to a vendor-supplied transformer program.
- 13 The **CHANNEL** parameter was not supplied and is required.
- 14 The **ELEMNAME** parameter was not supplied and is required.
- 15 The **ELEMNS** parameter was not supplied and is required.
- 16 The **DATCONTAINER** parameter was not set and is required.
- 17 There is a runtime validation failure.
- 18 There is a container datatype error.
- 101 The user is not authorized to use the XMLTRANSFORM.

TRANSFORM XMLTODATA

Use the **TRANSFORM XMLTODATA** command to convert XML to application data.

TRANSFORM XMLTODATA



Conditions: CHANNELERR, CONTAINERERR, INVREQ, LENGERR, NOTFND,

This command is threadsafe.

Description

The **TRANSFORM XMLTODATA** command can either transform XML to application data or query XML to return information about the XML elements to the application program. The **XMLTRANSFORM** resource defines the location of the XML binding and schema for transforming the XML to application data. If you do not specify an **XMLTRANSFORM** resource on the command, CICS queries the XML instead.

Options

CHANNEL (data-value)

Specify the 16-byte name of the channel used to pass the containers holding the **XMLCONTAINER** and **DATCONTAINER** data. The name of the channel must be 16 characters in length. If the channel name is less than 16 characters, you must pad the data value with trailing blanks.

DATCONTAINER (data-value)

Specify the 16-byte name of the output container that CICS populates with the converted data. The name of the container must be 16 characters in length. If the container name is less than 16 characters, you must pad the data value with trailing blanks.

CICS populates this container in BIT mode.

ELEMNAME (data-area)

Specify an input value to return the name of an XML element. CICS populates the **ELEMNAME** option with the local name of the first XML element that it finds in the **XMLCONTAINER** container. The application must also specify an input value for the **ELEMNAMELEN** option that indicates the maximum length of the data area.

ELEMNAMELEN (data-area)

Specify an input value to return the fullword binary length of the XML

element in the ELEMNAME option. CICS updates the value of the ELEMNAMELEN option to indicate the real length of the element name that it finds.

ELEMNS(*data-area*)

Specify an input value to return the namespace URI of the XML element to which the ELEMNAME option refers. CICS populates this parameter with the namespace URI of the first XML element that it finds in the XMLCONTAINER container. The application must also specify an input value for the ELEMNSLEN option that indicates the maximum length of the data area.

ELEMNSLEN(*data-area*)

Specify an input value to return the length of the ELEMNS option. CICS updates the value of the ELEMNSLEN option to indicate the real length of the namespace URI.

NSCONTAINER(*data-value*)

Specify the 16-byte name of the container that contains a list of XML namespace declarations that are in scope. These XML namespace declarations can be referenced in the body of the XMLCONTAINER container. The container must be populated in CHAR mode.

TYPENAME(*data-area*)

Specify an input value to return the `xsi:type` of the XML element referred to by the ELEMNAME option. This parameter is populated by CICS with the local name of the `xsi:type` attribute of the first XML tag that is found in the XMLCONTAINER container. If the first XML tag does not have an `xsi:type` attribute, this parameter remains empty. If the application supplies a value for the TYPENAME option, this value overrides any element and type information in the supplied XML and CICS attempts the transformation using the type name supplied by this application (together with the associated TYPENS option).

TYPENAMELEN(*data-area*)

Specify an input value to return the length of the TYPENAME option.

TYPENS(*data-area*)

Specify an input value to return the namespace of the `xsi:type` attribute of the XML element to which the ELEMNAME option refers. This parameter is populated by CICS with the namespace of the `xsi:type` attribute of the first XML element that is found in the XMLCONTAINER container.

TYPENSLEN(*data-area*)

Specify an input value to return the length of the TYPENS option.

XMLCONTAINER(*data-value*)

Specify the 16-byte name of the input container that contains the XML to be converted. This container must already exist and be populated in CHAR mode. If it is populated in BIT mode, CICS tries to determine the data encoding.

XMLTRANSFORM(*name*)

Specify the 32-byte name of the XMLTRANSFORM resource that CICS uses to transform the data to XML. The resource defines the XML binding and the XML schema. The name of the XMLTRANSFORM resource must be 32 characters. If the resource name is less than 32 characters, you must pad the value with blanks.

If you do not specify the XMLTRANSFORM option, no data transformation occurs. Instead, the application queries the XML. CICS returns the XML element and type information in the ELEMNAME, ELEMNS, TYPENAME, and TYPENS options.

Conditions

NOTFND

RESP2 values:

- 1 The XMLTRANSFORM was not found.

CHANNELERR

RESP2 values:

- 1 The channel specified by the **CHANNEL** parameter is incorrect.
- 2 The channel specified by the **CHANNEL** parameter was not found.

CONTAINERERR

RESP2 values:

- 1 The container specified by the **XMLCONTAINER** parameter was not found.
- 2 The container specified by the **NSCONTAINER** parameter was not found.
- 3 The container specified by the **DATCONTAINER** parameter was not found.

LENGERR

RESP2 values:

- 1 The data in the container specified by the **DATCONTAINER** parameter is too short for the specified transformation type.
- 2 The **ELEMNAME** buffer is too small.
- 3 The **ELEMNS** buffer is too small.
- 4 The **TYPENAME** buffer is too small.
- 5 The **TYPENS** buffer is too small.

INVREQ

RESP2 values:

- 1 The XMLTRANSFORM resource is not enabled.
- 2 The **XMLCONTAINER** container is empty.
- 3 The XML input data is incorrect. For more information, see the error message in the DFH-XML-ERRORMSG container.
- 4 The XML input data cannot be converted. For more information, see the error message in the DFH-XML-ERRORMSG container.
- 5 The application data is incorrect. For more information, see the error message in the DFH-XML-ERRORMSG container.
- 6 The application data cannot be converted. For more information, see the error message in the DFH-XML-ERRORMSG container.
- 7 Either the XML container or the NAMESPACE container is not CHAR mode.
- 9 The XMLTRANSFORM does not support the requested XML element.
- 10 The XMLTRANSFORM does not support the requested XML type.

Conditions: INVREQ, LENGERR, NOTFND

This command is threadsafe.

Description

The WEB READ QUERYPARM command reads a keyword parameter, consisting of a name and value pair, from a query string in a URL, and returns it in a specified code page. Escaped characters in the query string data are unescaped into the code page.

The WEB READ QUERYPARM command processes query string data for all HTTP methods, including GET, POST, PUT, and DELETE. You can continue to use the existing WEB READ FORMFIELD command for forms (messages with the media types application/x-www-form-urlencoded or multipart/form-data). Although the WEB READ FORMFIELD command can read name and value pairs from a query string, it does so only when the HTTP method is GET, because it assumes that the message is an HTML form.

You can use the query string browsing commands (WEB STARTBROWSE QUERYPARM, WEB READNEXT QUERYPARM, and WEB ENDBROWSE QUERYPARM) to browse through all the query parameters in a URL.

CICS only reads query string data when CICS is the HTTP server. The facility is not available when CICS is an HTTP client.

Options

QUERYPARM(*data-value*)

Specifies the name of the query parameter to extract. Specify a string of text containing the name of the requested parameter. The string of text supplied is not case-sensitive. If you specify the HOSTCODEPAGE option, you must supply the name of the query parameter in the code page that you specify for that option.

HOSTCODEPAGE(*data-value*)

Specifies the 8-character name of the CICS (host) code page required by the application program. This code page is normally an EBCDIC code page. CICS converts the value of the query parameter into this code page before returning it as the VALUE option.

The standard CICS form of a host code page name consists of the code page number (or more generally CCSID) written using 3 to 5 decimal digits as necessary then padded with trailing spaces to 8 characters. For code page 37, which is fewer than 3 digits, the standard form is 037. CICS also accepts any decimal number of up to 8 digits (padded with trailing spaces) in the range 1 to 65535 as a code page name, even if it is not in the standard form.

If the code page is not specified, the data is returned in the EBCDIC code page specified by the LOCALCCSID system initialization parameter (which applies to the local CICS region, and has a default of 037), provided that the specified code page is supported by the CICS web interface. The code page is supported if it is one of a list of EBCDIC code pages that are recognized by CICS as being sufficiently standard to allow successful parsing of the web headers (this includes all SBCS CECP and Euro code pages). Otherwise, CICS returns the data in the default EBCDIC code page 037 instead.

NAMELENGTH(*data-value*)

Specifies the length, as a fullword binary value, of the query parameter name.

SET(*ptr-ref*)

Specifies a pointer reference that is to be set to the address of data received.

VALUE(*data-area*)

Specifies the buffer to contain the value of the named query parameter. CICS unescapes any escaped characters before placing them in the buffer.

VALUELENGTH(*data-area*)

Specifies the length, as a fullword binary value, of the query parameter value. The real length of the value is returned in this data area. If you specify the VALUE option, VALUELENGTH specifies the maximum length of the data that the program accepts. If the value exceeds the length of the buffer, the value is truncated. If the length of the query parameter value is less than the size of the buffer, the query parameter value is placed in the leftmost byte positions.

Conditions**16 INVREQ**

RESP2 values are:

- 1 The command is being issued in a non-CICS Web support application.
- 3 The command is being issued for a non-HTTP request.
- 12 The host code page cannot be found.
- 13 No keyword parameters were supplied in the HTTP request.
- 14 The code page combination for client and server is invalid.
- 17 Invalid keyword parameters found in the HTTP request.

22 LENGERR

RESP2 values are:

- 1 The length in NAMELENGTH or VALUELENGTH is less than or equal to zero.
- 5 The keyword parameter value has been truncated during a read operation because the receiving buffer is too small.

13 NOTFND

RESP2 value is:

- 1 The keyword parameter with the given name could not be found.

WEB READNEXT QUERYPARM

Retrieve next name and value pair in query string data in a URL.

WEB READNEXT QUERYPARM

►►—WEB—READNEXT—QUERYPARM(*data-area*)—NAMELENGTH(*data-area*)—————►
 ►—VALUE(*data-area*)—VALUELENGTH(*data-area*)—————►►

Conditions: ENDFILE, INVREQ, LENGERR

This command is threadsafe.

Description

WEB READNEXT QUERYPARM retrieves the next keyword parameter (name and value pair) in a query string in a URL.

The data is returned in its unescaped form (see Escaped and unescaped data in the *CICS Internet Guide* for an explanation of this).

Options

QUERYPARM(*data-area*)

Specifies the buffer to contain the name of the keyword parameter being retrieved. The case of the name is as it is stored in the keyword parameter.

NAMELENGTH(*data-area*)

Specifies the length, as a fullword binary value, of the keyword parameter name. The actual length of the name is returned in this data area. If the length of the keyword parameter name is less than the size of the buffer, the keyword parameter name is placed in the leftmost byte positions.

VALUE(*data-area*)

Specifies the buffer to contain the value corresponding to the name returned in the QUERYPARM data area. CICS unescapes any escaped characters before placing them in the buffer.

VALUELENGTH(*data-area*)

Specifies the length, as a fullword binary value, of the keyword parameter value. The actual length of the value is returned in this data area. If the value exceeds the buffer length, it is truncated. If the length of the keyword parameter value is less than the size of the buffer, the keyword parameter value is placed in the leftmost byte positions.

Conditions

20 ENDFILE

The end of the list of keyword parameters has been reached.

16 INVREQ

RESP2 values are:

- 1 The command is being issued in a non-CICS Web support application.
- 3 The command is being issued for a non-HTTP request.
- 4 The command is being issued before a WEB STARTBROWSE has been issued.
- 6 A keyword parameter has been found which is not in the format NAME=VALUE.

22 LENGERR

RESP2 values are:

- 1 NAMELENGTH or VALUELENGTH is less than or equal to zero.
- 4 The keyword parameter name has been truncated during a browse operation because the receiving buffer is too small.
- 5 The keyword parameter value has been truncated because the receiving buffer is too small.

includes all SBCS CECP and Euro code pages). Otherwise, CICS returns the data in the default EBCDIC code page 037 instead.

NAMELENGTH (*data-value*)

specifies the length, as a fullword binary value, of the keyword parameter name. This field must be specified if a name data-area is specified with the QUERYPARM option.

Conditions

21 ILLOGIC

RESP2 value is:

- 5 A browse of keyword parameters is already in progress.

16 INVREQ

RESP2 values are:

- 1 The command is being issued in a non-CICS Web support application.
- 3 The command is being issued for a non-HTTP request.
- 12 The host code page cannot be found.
- 13 No keyword parameters found.
- 14 The code page combination for client and server is invalid.
- 17 Invalid keyword parameters found in the HTTP request.

13 NOTFND

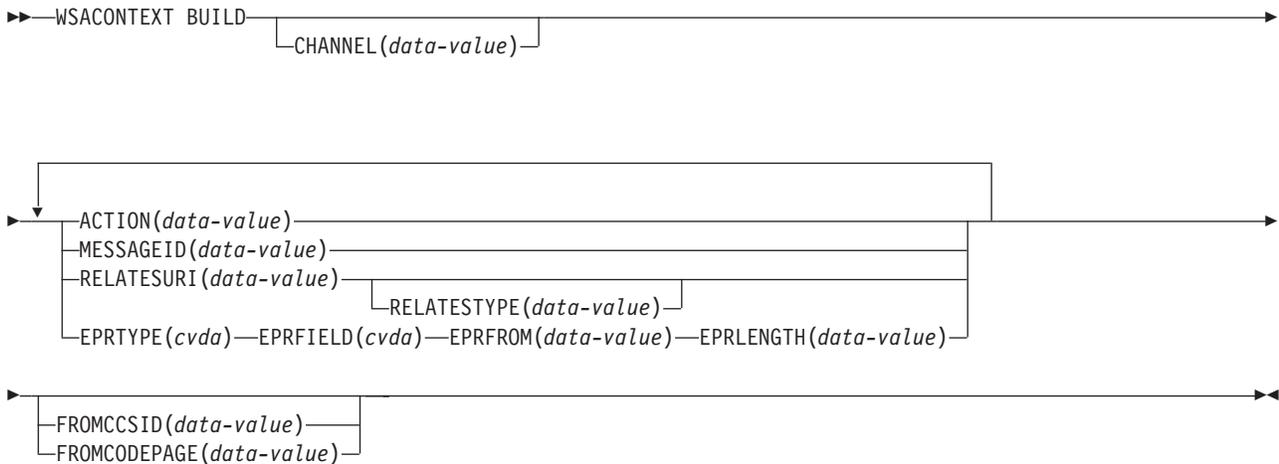
RESP2 value is:

- 1 Keyword parameter not found.

WSACONTEXT BUILD

Use the **WSACONTEXT BUILD** command to insert or replace WS-Addressing message addressing properties (MAPs) in the addressing context.

WSACONTEXT BUILD



Conditions: CCSIDERR, CHANNELERR, CODEPAGEERR, INVREQ, LENGERR

This command is threadsafe.

Description

Use the **WSACONTEXT BUILD** command for any of the following actions:

- To insert or replace the Action or Message ID MAPs
- To insert or replace the To, From, ReplyTo, or FaultTo endpoint reference MAPs.
- To insert the RelatesTo MAPs.

You can use the command repeatedly to supply different data on each call, for example different endpoint references (EPRs). The MAPs are applied to all outbound SOAP messages created by the **INVOKE SERVICE** or **INVOKE WEBSERVICE** commands and to response SOAP messages from a service provider.

Options

ACTION(data-value)

Specifies an input value containing an Action MAP of the request or response SOAP message; for example, `http://example.ibm.com/namespace/bookingInterface/MakeBooking`. Actions are supplied in the WSDL or are calculated by the Web services assistant, but can be overridden by this option. The data value must be 255 characters in length. If the Action MAP is less than 255 characters, you must pad the data value with trailing blanks.

CHANNEL(data-value)

Specifies the name of the channel that holds the addressing context. The name of the channel can be up to 16 characters in length. If the channel name is fewer than 16 characters, you must pad the data value with trailing blanks. If you do not specify this option, the current channel is implied.

Acceptable characters for the channel name are A-Z a-z 0-9 \$ @ # / % & ? ! : | " = , ; < > . - and _ . Leading and embedded blank characters are not permitted. The accepted set of characters for channel names includes some characters that do not have the same representation in all EBCDIC code pages. It is therefore

recommended that, if channels are to be shipped between regions, the characters used in naming them should be restricted to A-Z a-z 0-9 & : = , ; < > . - and _.

This option is required when the **WSACONTEXT BUILD** command is used by a Web service requester application and is the channel name used by subsequent **INVOKE SERVICE** commands.

EPRFIELD(*cvda*)

Specifies the endpoint reference field. You can use this option multiple times to build a full endpoint reference.

ADDRESS

The Address field of the endpoint reference is specified as a URI in the EPRFROM option.

ALL A complete endpoint reference, described in XML, is specified in the EPRFROM option.

METADATA

The Metadata section of the endpoint reference, described in XML, is specified in the EPRFROM option.

REFPARMS

The ReferenceParameters section of the endpoint reference, described in XML, is specified in the EPRFROM option.

EPRFROM(*data-value*)

An input data value that contains a complete or partial endpoint reference that is to be placed in the addressing context. The EPRFIELD option describes what part of the endpoint reference is specified in this option by the application. The endpoint reference can be supplied by a Web service application that uses the **WSAEPR CREATE** command or from another source, such as a service registry.

If the EPRFROM option contains an address, any special characters in the address are automatically escaped or unescaped.

The following characters in an address are replaced with an escape sequence:

" , > , < , ' , &

EPRLENGTH(*data-value*)

A fullword binary input data value used to contain the length of the supplied EPR.

EPRTYPE(*cvda*)

Specifies the type of EPR that is being built:

TOEPR

The destination EPR to which a SOAP message is sent.

REPLYTOEPR

An EPR to which a SOAP response message is returned.

FAULTTOEPR

An EPR to which a SOAP fault message is returned.

FROMEPR

An EPR that represents the sender of the SOAP message.

FROMCCSID(*data-value*)

Specifies the current Coded Character Set Identifier (CCSID) of the character data to be put into the addressing context, as a fullword binary number. If you

want to specify an IANA name for the code page or if you want to specify the CCSID as alphanumeric characters, use the FROMCODEPAGE option instead.

For CICS Transaction Server for z/OS applications, the CCSID is typically an EBCDIC CCSID.

The default CCSID of the region is specified on the **LOCALCCSID** system initialization parameter.

For an explanation of CCSIDs, and a list of the CCSIDs supported by CICS, see the *CICS Intercommunication Guide*.

FROMCODEPAGE(*data-value*)

Specifies an IANA-registered alphanumeric charset name or a Coded Character Set Identifier (CCSID) for the current code page of the character data to be put into the addressing context, 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 Intercommunication Guide*.

MESSAGEID(*data-value*)

Specifies a data value for a URI that uniquely identifies a SOAP message. The data value must be 255 characters in length. If the URI is less than 255 characters, you must pad the data area with trailing blanks.

RELATESTYPE(*data-value*)

Specifies a URI denoting the relationship type between the message to be sent and another message. The value must be 255 characters long. If the URI is less than 255 characters, you must pad the data value with trailing blanks. You are allowed to specify multiple RelatesTo MAPs. Subsequent calls of the RELATESTYPE and RELATESURI options create new RelatesTo MAPs.

If you do not specify a value for this option, the RelatesTo MAP does not have a type attribute in the SOAP message and defaults to a value of <http://www.w3.org/2005/08/addressing/reply>.

RELATESURI(*data-value*)

Specifies a URI denoting the message ID of a message that the message to be sent is related to. The value must be 255 characters long. If the URI is less than 255 characters long, you must pad the data value with trailing blanks.

Conditions

123 CCSIDERR

RESP2 values:

- 1 The CCSID specified on the FROMCCSID option is outside the range of valid CCSID values.
- 2 The CCSID specified on the FROMCCSID option and the CCSID of the addressing context are an unsupported combination.
- 4 One or more characters could not be converted. Each unconverted character has been replaced by a blank in the converted data.
- 5 There was an internal error in the code page conversion of the addressing context data.

- 6 Either the text encoding is not compatible with the specified CCSID on the FROMCCSID option, or one or more characters are truncated.

122 CHANNELERR

RESP2 values:

- 1 The name specified for the CHANNEL option contains an illegal character or combination of characters.
- 2 The specified channel was not located.

125 CODEPAGEERR

RESP2 values:

- 1 The code page specified on the FROMCODEPAGE option is not supported.
- 2 The code page specified on the FROMCODEPAGE option and the CCSID of the addressing context are an unsupported combination.
- 4 One or more characters could not be converted. Each unconverted character has been replaced by a blank in the converted data.
- 5 There was an internal error in the code page conversion of a container.
- 6 Either the text encoding is not compatible with the specified CCSID on the FROMCCSID option, or one or more characters are truncated.

16 INVREQ

The INVREQ RESP2 values and the corresponding messages are shown below. For this command, if the EIBRESP2 value is > 100, the fullword EIBRESP2 field is regarded as a structure containing two halfwords. The low-order halfword always contains the error number. The high-order halfword contains the offset into the XML data where the parsing error occurred.

RESP2 values:

- 4 The CHANNEL option was not specified. There is no current channel because the program that issued the command was not passed the name of a channel.
- 6 The ACTION field does not contain valid URI characters.
- 7 The MESSAGEID field does not contain valid URI characters.
- 8 The RELATESURI field does not contain valid URI characters.
- 9 The RELATESTYPE field does not contain valid URI characters.
- 10 The EPRFROM option does not contain valid XML.
- 13 The EPRFROM option does not contain valid XML. The EPR <Metadata> might not contain valid XML.
- 14 The EPRFROM option does not contain valid XML. The EPR <ReferenceParameters> might not contain valid XML.
- 15 The EPRFROM option might not contain a valid URI.

22 LENGERR

RESP2 values:

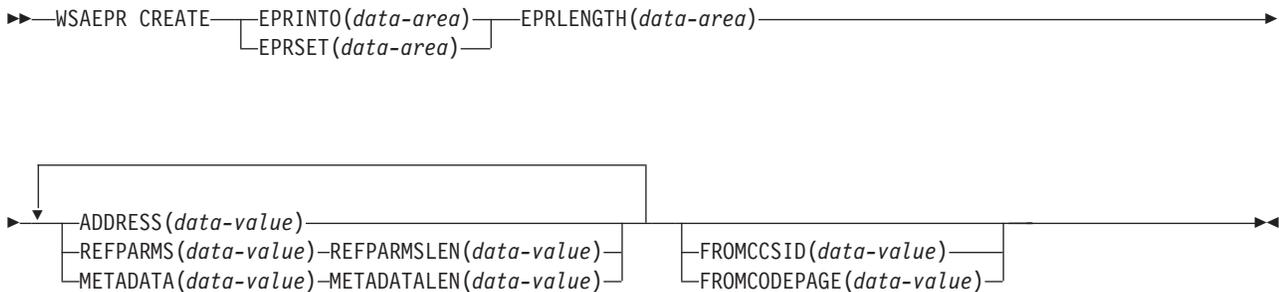
- 20 This condition occurs when the length of the stored data is greater than the value specified by the EPRLENGTH option. This condition only applies to the EPRINTO option and cannot occur with the EPRSET option.

Default action: ends the task abnormally.

WSAEPR CREATE

Use the **WSAEPR CREATE** command to create an endpoint reference (EPR) to represent a Web service or Web service resource.

WSAEPR CREATE



Conditions: CCSIDERR, CODEPAGEERR, INVREQ, LENGERR

This command is threadsafe.

Description

The **WSAEPR CREATE** command creates an endpoint reference, which can represent a Web service or Web service resource. You can send this EPR to a client so that the addressing context is used for requests to the service.

Options

ADDRESS(*data-value*)

Specifies a URI that forms the address of the endpoint reference. The value of this option must be 255 characters in length. If the URI is less than 255 characters, you must pad the data value with trailing blanks.

The default address of <http://www.w3.org/2005/08/addressing/anonymous> returns information to the caller of the command. Use an address of <http://www.w3.org/2005/08/addressing/none> if no request or response is required. If the To EPR contains a URI, the SOAP message is sent to this URI. If the ReplyTo or FaultTo EPRs contain a URI, response messages are sent to the Web service using that URI and not back to the sender of the request message.

EPRINTO(*data-area*)

Specifies the data area used to contain the generated endpoint reference. The EPRINTO and EPRSET options are mutually exclusive.

If the EPRINTO option contains an address, any special characters in the address are automatically escaped or unescaped.

The following characters in an address are replaced with an escape sequence:

" , > , < , ' , &

EPRLENGTH(*data-area*)

Specifies the length, as a halfword binary value, of the endpoint reference.

If you specify the EPRINTO option, you must specify a value for EPRLENGTH unless the length can be generated by the compiler from the EPRINTO option.

The EPRLENGTH option defines the maximum length of data that the program accepts. If the value specified is less than zero, zero is assumed. If the length of the data exceeds the value specified, the data is truncated to that value and the LENGERR condition occurs.

EPRSET(*ptr-ref*)

Specifies the pointer reference that is set to the address of the output data area used to contain the complete or partial endpoint reference. The pointer reference, unless changed by other commands or statements, is valid until the next **WSACONTEXT GET** command or the end of the task. The EPRINTO and EPRSET options are mutually exclusive.

FROMCCSID(*data-value*)

Specifies the current Coded Character Set Identifier (CCSID) of the character data to be read or written 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 FROMCODEPAGE option instead.

For CICS Transaction Server for z/OS applications, the CCSID is typically an EBCDIC CCSID.

The default CCSID of the region is specified on the LOCALCCSID system initialization option.

For an explanation of CCSIDs, and a list of the CCSIDs supported by CICS, see the *CICS Intercommunication Guide*.

FROMCODEPAGE(*data-value*)

Specifies an IANA-registered alphanumeric charset name or a Coded Character Set Identifier (CCSID) for the current code page of the character data to be read or written 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 Intercommunication Guide*.

METADATA(*data-value*)

Specifies metadata that describes the behavior, policies, and capabilities of the endpoint targeted by the endpoint reference. The metadata must be described in XML.

METADALEN(*data-value*)

Specifies the length, as fullword binary, of the METADATA option.

REFPARMS(*data-value*)

Specifies application reference options that form part of the endpoint reference. These options are described in XML.

REFPARMSLEN(*data-value*)

Specifies the length, as fullword binary, of the reference options.

Conditions

123 CCSIDERR

RESP2 values:

- 1 The CCSID specified on the FROMCCSID option is outside the range of valid CCSID values.
- 2 The CCSID specified on the FROMCCSID option and the CCSID of the container are an unsupported combination.
- 5 There was an internal error in the code page conversion of a container.
- 6 Either the text encoding is not compatible with the specified CCSID on the FROMCCSID option, or one or more characters are truncated.

125 CODEPAGEERR

RESP2 values:

- 1 The code page specified on the FROMCODEPAGE option is not supported.
- 2 The code page specified on the FROMCODEPAGE option and the CCSID of the container are an unsupported combination.
- 5 There was an internal error in the code page conversion of a container.
- 6 Either the text encoding is not compatible with the specified CODEPAGE on the FROMCODEPAGE option, or one or more characters are truncated.

16 INVREQ

The INVREQ RESP2 values and the corresponding messages are shown below. For this command, if the EIBRESP2 value is > 100, the fullword EIBRESP2 field is regarded as a structure containing two halfwords. The low-order halfword always contains the error number. The high-order halfword contains the offset into the XML data where the parsing error occurred.

RESP2 values:

- 8 One of the input parameters contains an incorrect value. The ADDRESS option might not contain a valid URI.
- 13 One of the input parameters contains an incorrect value. The METADATA option might not contain valid XML.
- 14 One of the input parameters contains an incorrect value. The REFPARMS option might not contain valid XML.

22 LENGERR

RESP2 values:

- 20 This condition occurs when the length of the stored data is greater than the value specified by the EPRLENGTH option. This condition only applies to the EPRINTO option and cannot occur with the EPRSET option.

Default action: ends the task abnormally.

WSACONTEXT DELETE

Use the **WSACONTEXT DELETE** command to delete the addressing context.

WSACONTEXT DELETE

▶▶—WSADDCONTEXT DELETE—CHANNEL(*data-value*)—————▶▶

Conditions: CHANNELERR, NOTFND, INVREQ

This command is threadsafe.

Description

The **WSACONTEXT DELETE** command deletes the addressing contexts for the request and response SOAP messages from the specified channel. You can use this command only in a Web service requester.

Options

CHANNEL(*data-value*)

Specifies the name of the channel holding the addressing context. The name of the channel can be up to 16 characters in length. If the channel name is less than 16 characters, you must pad the data value with trailing blanks.

Conditions

122 CHANNELERR

RESP2 values:

- 1 The name specified for the CHANNEL option contains an incorrect character or combination of characters.
- 2 The specified channel was not located.

16 INVREQ

RESP2 values:

- 5 This command is not allowed in a Web service provider.

13 NOTFND

RESP2 values:

- 3 The addressing context was not located on the specified channel.

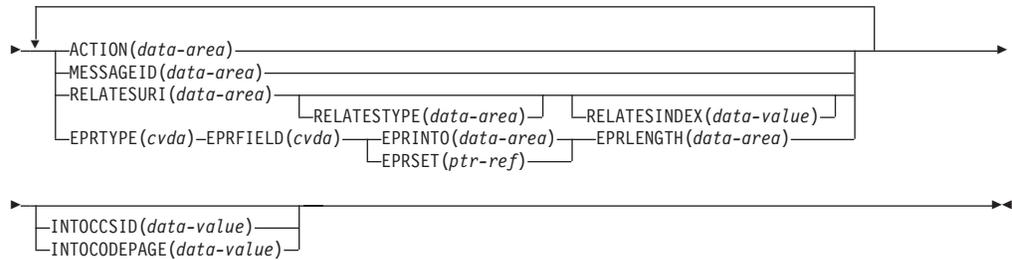
WSACONTEXT GET

Use the **WSACONTEXT GET** command in a service provider to get the message addressing properties (MAPs) sent by the service requester. Use the **WSACONTEXT GET** command in a service requester to get the MAPs of the reply message.

This command cannot be used by Axis2 applications hosted in an Axis2 pipeline in CICS.

WSACONTEXT GET

▶▶—WSACONTEXT GET—CONTEXTTYPE(*cvda*)—————▶▶
└─CHANNEL(*data-value*)┐



Conditions: CCSIDERR, CHANNELERR, CODEPAGEERR, INVREQ, LENGERR, NOTFND

This command is threadsafe.

Description

The **WSACONTEXT GET** command is used in a service provider to get the MAPs of the requester from the addressing context, or in a service requester to get the MAPs of the provider from the response message. The **WSACONTEXT GET** command can be called repeatedly to return different MAPs.

Options

ACTION(*data-area*)

Specifies an output area to contain the Action MAP of the request or response SOAP message. The data area must be 255 characters in length. If the Action MAP is fewer than 255 characters, CICS pads the data area with trailing blanks.

CHANNEL(*data-value*)

Specifies the name of the channel that holds the addressing context. The name of the channel can be up to 16 characters in length. If the channel name is fewer than 16 characters, you must pad the data value with trailing blanks. If you do not specify this option, the current channel is implied.

Acceptable characters for the channel name are A-Z a-z 0-9 \$ @ # / % & ? ! : | " = , ; < > . - and _ . Leading and embedded blank characters are not permitted. The accepted set of characters for channel names includes some characters that do not have the same representation in all EBCDIC code pages. It is therefore recommended that, if channels are to be shipped between regions, the characters used in naming them should be restricted to A-Z a-z 0-9 & : = , ; < > . - and _ .

This option is required when the **WSACONTEXT GET** command is used by a web service requester application and is the channel name used by subsequent **INVOKE SERVICE** commands.

CONTEXTTYPE(*cvda*)

Specifies which type of addressing context to fetch the MAPs from. CVDA values are as follows:

REQCONTEXT

Addressing context containing the request. Either a web service requester, or a web service provider application can access the MAPs in this addressing context.

RESPCONTEXT

Addressing context containing the response. Only a web service requester application can access the MAPs in this addressing context.

EPRFIELD(*cvda*)

Specifies the part of the endpoint reference that is to be returned in the EPRINTO data area. CVDA values are as follows:

ADDRESS

Return the Address field of the endpoint reference.

ALL Return the complete endpoint reference in XML.

METADATA

Return the Metadata section of the endpoint reference in XML.

REFPARMS

Return the ReferenceParameters section of the endpoint reference in XML.

EPRINTO(*data-area*)

An output data area used to contain the complete or partial endpoint reference. The EPRINTO and EPRSET options are mutually exclusive.

If the EPRINTO option contains an address, any special characters in the address are automatically escaped or unescaped.

The following characters in an address are replaced with an escape sequence:

" , > , < , ' , &

EPRLENGTH(*data-area*)

Specifies the length, as a halfword binary value, of the endpoint reference.

If you specify the EPRINTO option, you must specify a value for EPRLENGTH unless the length can be generated by the compiler from the EPRINTO option.

The EPRLENGTH option defines the maximum length of data that the program accepts. If the value specified is less than zero, zero is assumed. If the length of the data exceeds the value specified, the data is truncated to that value and the LENGERR condition occurs.

EPRSET(*ptr-ref*)

Specifies the pointer reference that is set to the address of the output data area used to contain the complete or partial endpoint reference. The pointer reference, unless changed by other commands or statements, is valid until the next **WSACONTEXT GET** command or the end of the task. The EPRINTO and EPRSET options are mutually exclusive.

EPRTYPE(*cvda*)

Specifies the type of endpoint reference that is being requested. CVDA values are as follows:

TOEPR

The destination endpoint reference to which a SOAP message is sent.

REPLYTOEPR

An endpoint reference to which a SOAP response message is returned.

FAULTTOEPR

An endpoint reference to which a SOAP fault message is returned.

FROMEPR

An endpoint reference that represents the sender of the SOAP message.

INTOCCSID(*data-value*)

Specifies the Coded Character Set Identifier (CCSID) into which the character data in the addressing context 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.

For an explanation of CCSIDs, and a list of the CCSIDs supported by CICS, see CICS-supported conversions in the Intercommunication Guide.

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 addressing context 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 CICS-supported conversions in the Intercommunication Guide.

MESSAGEID(*data-area*)

Specifies an output area to contain the MessageID MAP of the request or response SOAP message. The data area must be 255 characters in length. If the MessageID MAP is less than 255 characters, CICS pads the data area with trailing blanks.

RELATESINDEX(*data-value*)

Specifies a numeric value that indicates which RelatesTo MAP to return. If this value is not specified, the first RelatesTo MAP is returned. The index starts at 1. If the value of the index is greater than the number of RelatesTo MAPs, spaces are returned in the RELATESTYPE and RELATESURI data areas.

RELATESTYPE(*data-area*)

Specifies an output area that contains a URI denoting the relationship type between this message and another message. The data area must be 255 characters in length. If the URI is less than 255 characters, CICS pads the data area with trailing blanks.

RELATESURI(*data-area*)

Specifies an output area that contains the RelatesTo MAP relationship URI between this message and another. The data area must be 255 characters in length. If the MessageID MAP is less than 255 characters, CICS pads the data area with trailing blanks.

Conditions

123 CCSIDERR

RESP2 values:

- 1 The CCSID specified on the INTOCCSID option is outside the range of valid CCSID values.

- 2 The CCSID specified on the INTOCCSID option and the CCSID of the addressing context are an unsupported combination.
- 4 One or more characters could not be converted. Each unconverted character has been replaced by a blank in the converted data.
- 5 There was an internal error in the code page conversion of the addressing context data.

122 CHANNELERR

RESP2 values:

- 1 The name specified for the CHANNEL option contains an incorrect character or combination of characters.
- 2 The channel specified was not located.

125 CODEPAGEERR

RESP2 values:

- 1 The code page specified on the INTOCODEPAGE option is not supported.
- 2 The code page specified on the INTOCODEPAGE option and the CCSID of the addressing context are an unsupported combination.
- 4 One or more characters could not be converted. Each unconverted character has been replaced by a blank in the converted data.
- 5 There was an internal error in the code page conversion of a container.

16 INVREQ

RESP2 values:

- 4 The CHANNEL option was not specified. There is no current channel because the program that issued the command was not passed the name of a channel.
- 11 The RELATESINDEX option is not valid.
- 12 The RELATESINDEX option is greater than the number of RelatesTo MAPs.

22 LENGERR

RESP2 values:

- 20 This condition occurs when the length of the stored data is greater than the value specified by the EPRLENGTH option. This condition only applies to the EPRINTO option and cannot occur with the EPRSET option.

Default action: ends the task abnormally.

13 NOTFND

RESP2 values:

- 3 The addressing context was not located.

Appendix B. New RDO resources

There are new RDO resources in CICS Transaction Server for z/OS, Version 4 Release 2.

ATOMSERVICE resources

An ATOMSERVICE resource defines an Atom service, feed, collection, or category document that CICS can deliver to a Web client over HTTP.

To support each ATOMSERVICE resource, you need a URIMAP resource to handle incoming Web client requests and point to the appropriate ATOMSERVICE resource.

Atom feed documents and collections are documents that contain one or more Atom entries. Web clients cannot edit the Atom entries in a feed, but they can edit the Atom entries in a collection. When a Web client makes an HTTP request relating to the Atom feed or collection, CICS assembles the appropriate Atom entries and delivers the document to the Web client. The ATOMSERVICE resource for an Atom feed or collection identifies the Atom configuration file, CICS resource or application program, and XML binding that CICS uses to supply the data and metadata for the Atom entries.

Atom service documents provide information about the collections that are available from CICS, and Atom category documents list the categories that can be applied to Atom entries in a collection. You can create these documents as an Atom configuration file and serve them using an ATOMSERVICE resource, or you can deliver them as a static document through CICS Web support.

Installing ATOMSERVICE resource definitions

This procedure uses the CEMT and CEDA transactions to install an ATOMSERVICE resource definition. If the ATOMSERVICE resource already exists, it has to be disabled before it can be reinstalled.

Procedure

1. If the ATOMSERVICE resource already exists, ensure that it is disabled. Use the following command:

```
CEMT SET ATOMSERVICE(name) DISABLED
```

While the ATOMSERVICE resource is disabled, if a web client makes an HTTP request that requires the resource, CICS returns an HTTP 503 response (Service Unavailable) to the web client.

2. Install the ATOMSERVICE definition. Use the following command:

```
CEDA INSTALL GROUP(groupname) ATOMSERVICE(name)
```

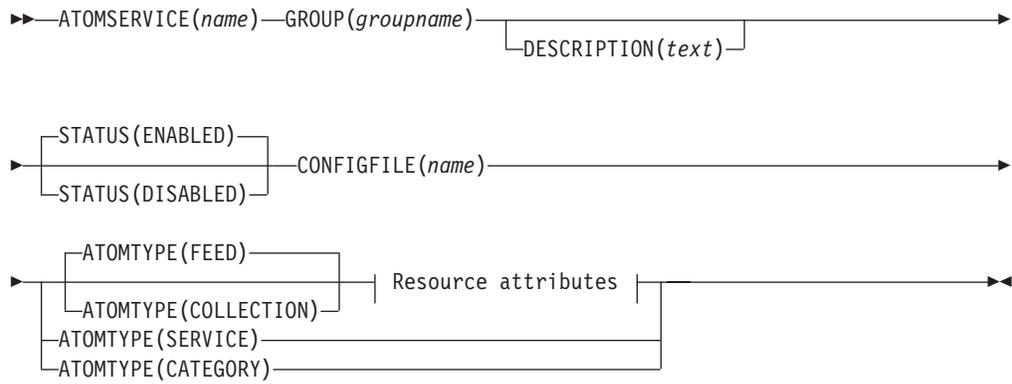
3. Optional: When you have successfully installed the ATOMSERVICE definition, use CEMT to enable the resource. Perform this step only if the ATOMSERVICE resource is not already defined as ENABLED, and you want to make the resource available for web clients. Use the following command:

CEMT SET ATOMSERVICE(*name*) ENABLED

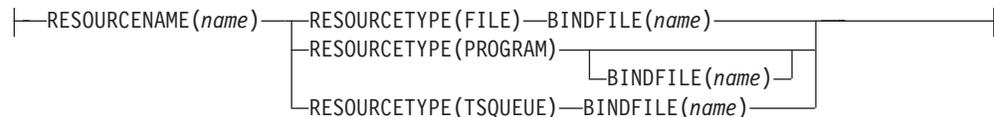
ATOMSERVICE attributes

Describes the syntax and attributes of the ATOMSERVICE resource.

Syntax



Resource attributes:



Attributes

ATOMSERVICE(*name*)

Specifies the 8-character name of this resource definition.

Acceptable characters:

A-Z 0-9 \$ @ #

Unless you are using the CREATE command, any lowercase characters that you enter are converted to uppercase.

ATOMTYPE({ FEED | SERVICE | COLLECTION | CATEGORY })

Specifies the type of Atom document that is returned for this ATOMSERVICE definition.

CATEGORY

An Atom category document, which lists the categories for entries in a collection. You can set up a category document if you want to use the same categories to define multiple collections.

COLLECTION

An Atom collection document, which contains a group of Atom entries that can be edited by Web clients using HTTP POST, PUT, and DELETE requests, as well as being retrieved using HTTP GET requests. The Atom configuration file for a collection must begin with the root element <cics:atomservice type="collection">.

FEED An Atom feed document, which describes the metadata for a feed, and contains Atom entries that provide data for the feed. An Atom feed can

be retrieved using HTTP GET requests, but it cannot be edited by a Web client. The Atom configuration file for an Atom feed must begin with the root element `<cics:atomservice type="feed">`.

SERVICE

An Atom service document, which provides information about the editable collections that are available on a server.

BINDFILE(*name*)

Specifies the fully qualified (absolute) or relative name of an XML binding stored in z/OS UNIX System Services. This attribute is not used for an Atom service or category document. You create an XML binding using the CICS XML assistant program DFHLS2SC.

For resource types FILE and TSQUEUE, the XML binding is required, and it specifies the data structures used by the resource named in RESOURCENAME, which supplies the data for the Atom document.

For resource type PROGRAM, an XML binding is optional, and you create it using the resource that the program accesses to obtain the data for the Atom entries, not the program itself. You must specify an XML binding for resource type PROGRAM if you are using the resource handling parameters in the DFHATOMPARMS container to pass information from the Atom configuration file to the program. If you are not doing this, do not specify an XML binding.

The name of the XML binding can be specified as an absolute path including all directories and beginning with a slash, for example, /u/atom/atomictest.xsdbind. Alternatively, it can be specified as a path relative to the HOME directory of the CICS region user ID; for example, atom/atomictest.xsdbind (with no leading forward slash). Up to 255 characters can be used.

Acceptable characters:

A-Z a-z 0-9 \$ @ # . / - _ % & ? ! : | " = ~ , ; < >

CONFIGFILE(*name*)

Specifies the fully qualified (absolute) or relative name of an Atom configuration file stored in z/OS UNIX System Services. The Atom configuration file contains XML that specifies metadata and field names for the Atom document that is returned for this resource definition. For details, see the *CICS Internet Guide*.

The name can be specified as an absolute path including all directories and beginning with a slash, for example, /u/atom/myfeed.xml. Alternatively, it can be specified as a path relative to the HOME directory of the CICS region user ID, for example, atom/myfeed.xml (with no leading forward slash). Up to 255 characters can be used.

Acceptable characters:

A-Z a-z 0-9 \$ @ # . / - _ % & ? ! : | " = ~ , ; < >

DESCRIPTION(*text*)

You can provide a description of the resource that 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.

RESOURCENAME (*name*)

Specifies the 1 - 16 character name of the CICS resource that provides the data for this Atom feed or collection. If the data for your Atom entries is held in a resource that is accessed by a service routine, specify the name of the service routine here. This attribute is not used for an Atom service or category document.

Acceptable characters:

A-Z a-z 0-9 \$ @ # . / - _ % & ? ! : | " = ~ , ; < >

RESOURCETYPE (**FILE** | **PROGRAM** | **TSQUEUE**)

Specifies the type of CICS resource that holds the data for this Atom feed or collection. This attribute is not used for an Atom service or category document.

FILE A CICS file. A single record in the file provides the data for a single Atom entry. A file that holds Atom entries must have a unique key for the records, and you cannot use an alternate index file that has been defined with the NONUNIQUEKEY attribute. You can use any type of VSAM file to hold Atom entries, but note that ESDS (entry-sequenced data set) files are not suitable for a feed that you might want to set up as an editable collection, because you cannot delete records in an ESDS. You cannot use a BDAM file.

PROGRAM

A service routine, which is a CICS application program written to supply content for Atom entries.

TSQUEUE

A temporary storage queue. A single record in the temporary storage queue provides the data for a single Atom entry.

STATUS ({ **ENABLED** | **DISABLED** })

Indicates whether the Atom document specified by this resource definition is available or unavailable.

BUNDLE resources

A BUNDLE resource defines a *bundle*, a unit of deployment for an application. A bundle is a collection of CICS resources, artifacts, references, and a manifest that you can deploy into a CICS region to represent an application.

The manifest is a file that describes the contents of the bundle, including what resources to create in the CICS region and the location of the supporting artifacts,

what prerequisites are required for the application to run successfully, and any services that the application can offer to other applications.

A bundle is deployed to z/OS UNIX and comprises a directory structure of artifacts. The BUNDLE resource defines where the bundle is deployed on z/OS UNIX and its status. When you enable a BUNDLE resource, CICS reads the manifest and dynamically creates the application resources that are defined in the manifest for you. Other resources that are defined as prerequisites for the application must be present in the CICS region to successfully enable the BUNDLE resource.

If CICS fails to create one or more of the application resources, the BUNDLE installs in a DISABLED state. You can use the IBM CICS Explorer to view the state of each resource. You can try to enable the BUNDLE resource again. However, if one of the resources, for example a WEBSERVICE, installs in an UNUSABLE state, you cannot enable the BUNDLE resource. You must discard the BUNDLE resource and re-create the definition.

If you disable one of the resources that was created by the BUNDLE, for example an EVENTBINDING resource, CICS disables the BUNDLE resource as well. However, any other resources that are part of the bundle remain in an enabled state in the CICS region. If you reenable the resource successfully, the BUNDLE resource also changes to the ENABLED state. If you try to discard a disabled BUNDLE resource when enabled resources that belong to the bundle are in the CICS region, CICS issues a message and the discard fails. You must disable each of the enabled resources before discarding the BUNDLE resource. You can use the **DISABLE BUNDLE** command on a disabled bundle to disable all of the associated resources.

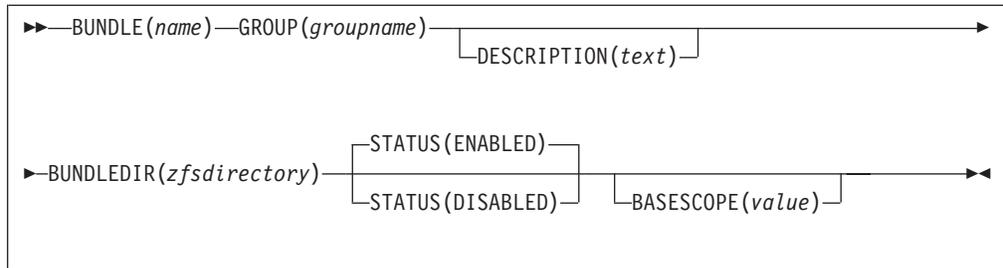
Note: When you have created and deployed your bundle using the CICS Explorer, you should ensure proper management of the bundle source code. The bundle cannot be reconstructed from the exported data in zFS, and a failure of your workstation would cause the data to be lost. You can use the Export function of CICS Explorer to export the bundle and check it in to a source code management system.

Tip: Because a bundle is the unit of deployment for an application, you are recommended to enable, disable, and discard the BUNDLE resource only.

If you disable a resource that is listed as a prerequisite of the bundle, for example a FILE resource, CICS disables the BUNDLE resource. Prerequisites are listed in the imports section of the bundle manifest. Use the IBM CICS Explorer to view the list of imports for a bundle.

BUNDLE attributes

Describes the syntax and attributes of the BUNDLE resource.



BASESCOPE(*value*)

Specifies the 1 - 255 character string that defines the scope for the contents of the bundle. You are recommended to specify a uniform resource identifier (URI). Use the BASESCOPE attribute when you want to group similar bundles together. The default value is empty so that all BUNDLE resources install in the same scope.

Acceptable characters:

A-Z a-z 0-9 \$ @ # . / - _ % & ? ! : | " = ~ , ; < >

BUNDLE(*name*)

Specifies the 1 - 8 character name of the BUNDLE.

Acceptable characters:

A-Z a-z 0-9 \$ @ # . / - _ % & ? ! : | " = ~ , ; < >

Do not use names beginning with DFH, because these characters are reserved for use by CICS.

BUNDLEDIR(*zfsdirectory*)

Specifies the 1 - 255 character fully qualified name of the root directory for the bundle on z/OS UNIX. The first character must be /.

The value specified must be a valid name for a UNIX file:

- It must not contain imbedded space characters.
- It must not contain consecutive instances of the / character.
- It is case-sensitive.

Acceptable characters:

A-Z a-z 0-9 . / _ # @ -

DESCRIPTION(*text*)

In this field, you can provide a description of the resource that you are defining. The description text can be up to 58 characters in length. No restrictions apply to the characters that you may use. However, if you use parentheses, ensure that each left parenthesis has 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 lowercase characters that you enter are converted to uppercase.

The GROUP name can be up to 8 characters in length. Do not use group names beginning with DFH, because these characters are reserved for use by CICS.

STATUS(ENABLED|DISABLED)

Specifies the initial status of the BUNDLE resource when it is installed.

ENABLED

The BUNDLE is available for use. CICS checks that all prerequisites for the bundle are present in the region and attempts to install and enable all of the required resources that are defined in the bundle.

DISABLED

The BUNDLE is not available for use. CICS does not check for prerequisites and installs, but does not enable, the resources that are defined in the bundle.

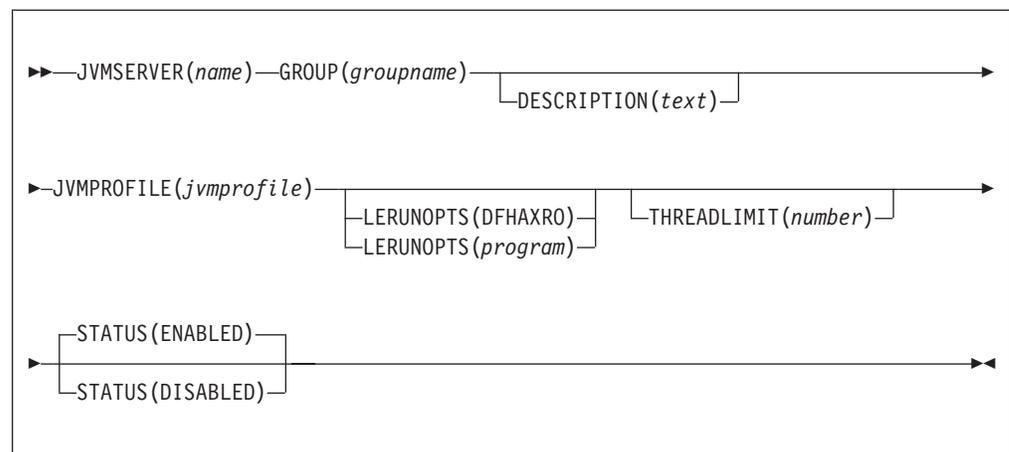
JVMSEVER resources

A JVMSEVER resource defines the runtime environment for a JVM server.

The JVMSEVER resource defines the location of the JVM profile and the Language Environment options that are required to create a Language Environment enclave and a JVM server in a CICS region.

JVMSEVER attributes

Describes the syntax and attributes of the JVMSEVER resource.



JVMSEVER(*name*)

Specifies the 1 - 8 character name of the JVMSEVER resource.

Acceptable characters:

A-Z a-z 0-9 \$ @ # . - _ % ? ! : | = , ;

Do not use names beginning with DFH, because these characters are reserved for use by CICS.

DESCRIPTION(*text*)

In this field, you can provide a description of the resource that you are defining. The description text can be up to 58 characters in length. No restrictions apply to the characters that you may use. However, if you use parentheses, ensure that each left parenthesis has 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 lowercase characters that you enter are converted to uppercase.

The GROUP name can be up to 8 characters in length. Do not use group names beginning with DFH, because these characters are reserved for use by CICS.

JVMPROFILE(*jvmpfile*)

Specifies the 1 - 8 character name of the JVM profile for the JVM server. The JVM profile is a file in the z/OS UNIX directory that is specified by the system initialization parameter **JVMPROFILEDIR**. Alternatively, the file can be in another place in the z/OS UNIX file system and be referenced by a UNIX soft link from the **JVMPROFILEDIR** directory. The profile contains the JVM options for running a JVM server.

Acceptable characters:

A-Z a-z 0-9 \$ @ # . - _ % ? ! : | " = , ; < >

LERUNOPTS(**DFHAXRO**|*program*)

Specifies the 1 - 8 character name of the program that defines the runtime options for the Language Environment enclave. DFHAXRO is a supplied program that provides a set of default values. The source for DFHAXRO is in the *hlq*.SDFHSAMP library if you want to change the defaults for any of the Language Environment runtime options.

If you want to use a different program, put the program in the *hlq*.SDFHLOAD library and specify the program name in uppercase characters.

STATUS(**ENABLED**|**DISABLED**)

Specifies the initial status of the JVMSERVER resource when it is installed.

ENABLED

The JVM server runtime environment is available for use.

DISABLED

The JVM server runtime environment is not available for use.

THREADLIMIT(**15**|*number*)

Specifies the maximum number of threads that are allowed in the Language Environment enclave for the JVM server. Each thread runs under a T8 TCB. You can specify a limit in the range of 1 - 256 threads.

If you specify a thread limit that exceeds the maximum of 1024 threads that is allowed for the CICS region, taking into account all other enabled and disabled JVMSERVER resources, CICS allocates the remaining threads up to 1024 to the resource as the thread limit value. If CICS is already at the maximum number of JVMSERVER threads, the resource installs in a disabled state.

MQCONN resources

An MQCONN resource defines the attributes of the connection between CICS and WebSphere MQ. You can install or discard an MQCONN resource only when CICS is not connected to WebSphere MQ.

Only one MQCONN resource can be installed at a time in a CICS region. When you define an MQCONN resource, CICS checks to ensure that there is only one MQCONN resource defined in the group or list. If more than one is found, even one with a different name, a warning message is issued. If you do install a second MQCONN resource, CICS implicitly discards the existing MQCONN resource before proceeding with the installation, unless you are reinstalling an MQCONN resource with the same name.

When you have installed the MQCONN resource, you can use the CEMT or EXEC CICS SET MQCONN command, CICSplex SM, or the CICS Explorer to start the connection between CICS and WebSphere MQ.

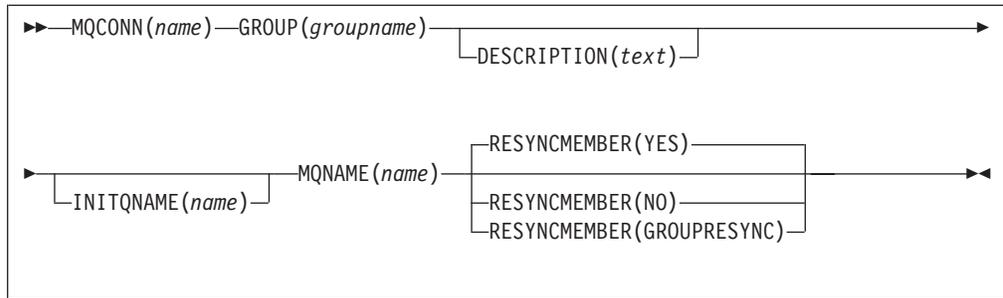
When you install an MQCONN resource that includes a setting for the INITQNAME attribute, CICS also dynamically creates and installs an MQINI resource. The MQINI resource represents the default initiation queue that is specified by the INITQNAME attribute of the MQCONN resource. The name of the MQINI resource is DFHMQINI, and its attribute INITQNAME specifies the initiation queue name.

You can use the EXEC CICS or CEMT INQUIRE MQINI command to inquire on the INITQNAME attribute of the dynamically created MQINI resource. If you want to change it, you must reinstall the MQCONN resource with an appropriate INITQNAME attribute. The MQINI resource is discarded when the MQCONN resource is discarded.

When you discard an MQCONN resource that includes a setting for the INITQNAME attribute, the dynamically created MQINI resource that represents the default initiation queue specified by the INITQNAME attribute is also discarded.

MQCONN attributes

Describes the syntax and attributes of the MQCONN resource.



Attributes

DESCRIPTION(*text*)

In this field, you can provide a description of the resource that you are defining. The description text can be up to 58 characters in length. No restrictions apply to the characters that you may use. However, if you use parentheses, ensure that each left parenthesis has 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 lowercase characters that you enter are converted to uppercase.

The GROUP name can be up to 8 characters in length. Do not use group names beginning with DFH, because these characters are reserved for use by CICS.

INITQNAME(*name*)

Specifies the 1 - 48 character name of the default initiation queue for this CICS-WebSphere MQ connection.

Acceptable characters:

A-Z a-z 0-9 . / _ %

When you install the MQCONN resource definition, if you have specified a non-blank INITQNAME attribute, CICS installs an implicit MQINI resource to represent the default initiation queue. The name of the MQINI resource is DFHMQINI, and its attribute INITQNAME specifies the default initiation queue name. You can inquire on this resource, but you cannot explicitly create, discard, or set the resource. If you want to change it, you must reinstall the MQCONN resource definition with an appropriate INITQNAME attribute.

If you specify the name of a default initiation queue when you start the CICS-WebSphere MQ connection by using the CKQC START command, the setting for the INITQNAME attribute in the installed MQINI resource definition is replaced with the name of the default initiation queue that you specified on the command.

MQCONN(*name*)

Specifies the 8-character name of this resource definition.

Acceptable characters:

A-Z 0-9 \$ @ #

Unless you are using the CREATE command, any lowercase characters that you enter are converted to uppercase.

MQNAME (*name*)

Specifies the 1 - 4 character name of either a single WebSphere MQ queue manager, or a queue-sharing group of WebSphere MQ queue managers.

- If you specify a single queue manager, when you start the CICS-WebSphere MQ connection, CICS connects only to this queue manager.
- If you specify a queue-sharing group, when you start the connection, CICS connects to any active member of this group on the same LPAR. Use the RESYNCMEMBER attribute to specify what happens when CICS is holding outstanding units of work for the last queue manager to which it connected from the queue-sharing group.
- If you specify RESYNCMEMBER(GROUPRESYNC) then MQNAME must be the name of a queue-sharing group.

Acceptable characters:

A-Z 0-9 \$ @ #

The name of the queue manager or queue-sharing group must not start with a numeric character.

WebSphere MQ uses the @ symbol to pad queue-sharing group names that are less than four characters long. When you specify a queue-sharing group that is less than four characters long, do not include the @ symbols at the end; CICS adds the padding before querying the queue-sharing group. For example, if the queue-sharing group is defined to WebSphere MQ as DEV@, specify DEV as the value of the MQNAME attribute.

If you specify the name of an alternative WebSphere MQ queue manager or queue-sharing group on the CEMT or EXEC CICS SET MQCONN command, or specify a queue manager name when you start the CICS-WebSphere MQ connection by using the CKQC START command, CICS connects to that alternative queue manager or queue-sharing group. In addition, your setting for the MQNAME attribute in the installed MQCONN definition is replaced with the name of the queue manager or queue-sharing group that you specified on the command. If you want to revert to the original queue manager or queue-sharing group, set MQNAME again.

RESYNCMEMBER ({YES|NO|GROUPRESYNC})

This attribute applies only if you have used the MQNAME attribute to specify a WebSphere MQ queue-sharing group. RESYNCMEMBER specifies the strategy that CICS adopts if outstanding units of work are being held for the last queue manager to which CICS was connected from the queue-sharing group.

Changing the setting for RESYNCMEMBER must be done only when all resources are in a consistent state; that is, there are no indoubt units of work outstanding otherwise CICS is not able to resynchronize the WebSphere MQ units of work. It is important to ensure that all resources are in a consistent state before changing RESYNCMEMBER to or from GROUPRESYNC.

- YES** CICS connects to the same queue manager, waiting, if necessary, until the queue manager becomes active to resolve the indoubt units of work. YES is the default.
- NO** CICS makes one attempt to connect to the same queue manager. If that attempt fails, CICS connects to any member of the queue-sharing group and issues the warning message DFHMQ2064 about the outstanding units of work.

GROUPRESYNC

CICS connects to any member of the queue-sharing group. The queue manager is chosen by WebSphere MQ and it asks CICS to resolve indoubt units of work on behalf of all eligible queue managers in the queue-sharing group. This function is called *group unit of recovery*. The GROUPRESYNC option can be used only when you are running WebSphere MQ 7.1, or higher, that supports group unit of recovery for CICS and when group unit of recovery is enabled in the queue managers.

Do not change the setting for RESYNCMEMBER when units of work are outstanding in WebSphere MQ, because units of work cannot be resolved. A unit of work held in CICS is identified with a resource manager qualifier. When RESYNCMEMBER(GROUPRESYNC) is used the qualifier is the name of the queue-sharing group, otherwise the qualifier used is the name of the individual queue manager.

Units of work that are shunted indoubt are not included in this process, because CICS itself cannot resolve those units of work at this time. Resynchronization for those UOWs occurs when CICS has resynchronized with its remote coordinator.

Appendix C. New system programming commands

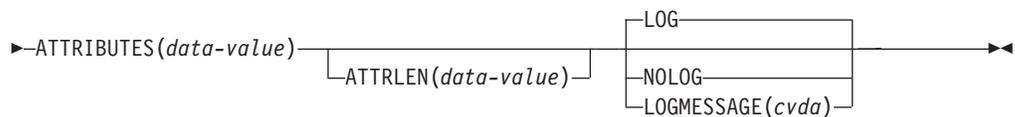
CICS Transaction Server for z/OS, Version 4 Release 2 extends the CICS system programming interface with new commands.

CREATE ATOMSERVICE

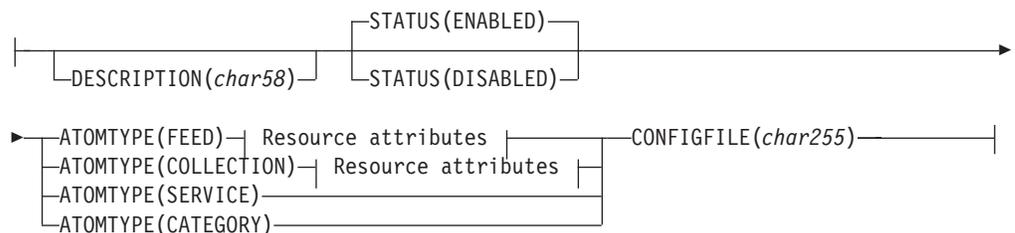
Define an ATOMSERVICE resource definition in the local CICS region.

CREATE ATOMSERVICE

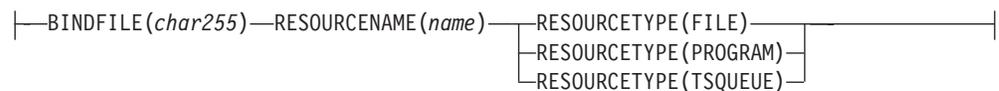
▶—CREATE ATOMSERVICE(*data-value*)—→



CREATE ATOMSERVICE attribute values:



Attributes for FEED or COLLECTION:



Conditions: 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 ATOMSERVICE command builds an ATOMSERVICE definition. It does not use a resource definition stored in the CSD. If an ATOMSERVICE definition already exists with the name that you specify in the local CICS region, the command fails unless the existing ATOMSERVICE definition is disabled, in which case the new definition replaces the old one. If no ATOMSERVICE definition with the name specified exists, the new definition is added.

A syncpoint is implicit in CREATE ATOMSERVICE 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 command is successful, and rolled back if not.

Options

ATTRIBUTES (*data-value*)

Specifies the attributes of the ATOMSERVICE definition being added. The list of attributes must be coded as a single character string using the syntax shown in **ATOMSERVICE definition attributes**. See 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 must not exceed 32767 bytes.

LOGMESSAGE (*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

ATOMSERVICE (*data-value*)

Specifies the 8-character name of the ATOMSERVICE definition to be added to the CICS region.

Conditions

INVREQ

RESP2 values:

- n** The ATTRIBUTES string contains a syntax error, or an error occurred during either the discard or resource definition phase of the processing. The RESP2 value is associated with a message written to the transient data queue CSMT, which identifies more precisely the nature of the error.
- 7** The LOGMESSAGE cvda value is not valid.
- 200** The command ran 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 that you have specified in ATTRLEN is negative.

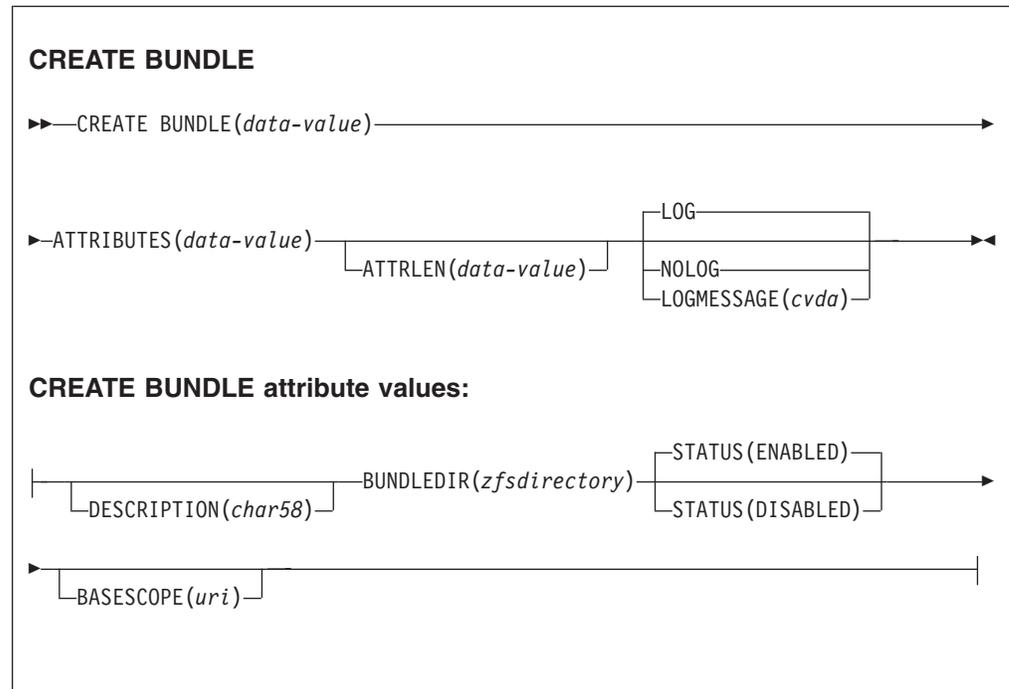
NOTAUTH

RESP2 values:

- 100** The user associated with the issuing task is not authorized to use this command.
- 101** The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

CREATE BUNDLE

Define a BUNDLE resource in the local CICS region.



Conditions: 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 BUNDLE** command installs a **BUNDLE** definition with the attributes specified on the command. It does not use a resource definition stored in the CSD. If there is already a **BUNDLE** with the name that you specify in the local CICS region, and the existing **BUNDLE** is disabled, the new definition replaces the old one; if an existing **BUNDLE** is not disabled, the **CREATE** command fails.

Options

ATTRIBUTES(*data-value*)

Specifies the attributes of the **BUNDLE** being added. The list of attributes must be coded as a single character string using the syntax shown in **BUNDLE attributes**. See 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 must not exceed 32 767 bytes.

BUNDLE(*data-value*)

Specifies the 8-character name of the **BUNDLE** definition to be added to the CICS region.

LOGMESSAGE (*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

Conditions**INVREQ**

RESP2 values:

- n** The ATTRIBUTES string contains a syntax error, or an error occurred during either the discard or resource definition phase of the processing. The RESP2 value is associated with a message written to the transient data queue CSMT, which identifies more precisely the nature of the error.
- 612** Installation of the BUNDLE definition failed because it already exists.
- 632** Installation of BUNDLE resource *resource* failed because the manifest is not valid.
- 633** Installation of BUNDLE resource *resource* failed because it had no manifest.
- 686** Installation of the BUNDLE resource failed because an unexpected resource error occurred.

LENGERR

RESP2 values:

- 1** The length that 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.
- 101** The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

CREATE JVMSERVER

Define a JVMSERVER resource in the local CICS region.

JVMSEVER(*data-value*)

Specifies the 8-character name of the JVMSEVER resource definition that is installed in the CICS region.

Conditions

INVREQ

RESP2 value:

- n** The ATTRIBUTES string contains a syntax error, or an error occurred during either the discard or resource definition phase of the processing. The RESP2 value is associated with a message written to the transient data queue CSMT, which identifies more precisely the nature of the error.

LENGERR

RESP2 value:

- 1** The length that 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.
- 101** The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

CREATE MQCONN

Define an MQCONN resource in the local CICS region.

CREATE MQCONN

►►—CREATE MQCONN(*data-value*)—ATTRIBUTES(*data-value*)—
└──────────────────────────────────┘
└──────────ATTRLEN(*data-value*)──┘

┌──LOG──┐
├──┤
├──NOLOG──┤
├──LOGMESSAGE(*cvda*)──┤
└──┘

CREATE MQCONN attribute values:

┌──┐ ┌──MQNAME(*char4*)──┐ ┌──┐
├──DESCRIPTION(*char58*)──┤ └──INITQNAME(*char48*)──┘
└──┘

┌──RESYNCMEMBER(YES)──┐
├──┤
├──RESYNCMEMBER(NO)──┤
├──RESYNCMEMBER(GROUPRESYNC)──┤
└──┘

Conditions: 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 MQCONN command installs an MQCONN resource definition with the attributes specified on the command. It does not use a resource definition stored in the CSD. If an MQCONN resource definition is already in the local CICS region, the new definition replaces the old one; if not, the new definition is added.

An MQCONN resource definition can be installed only when CICS is not connected to WebSphere MQ.

A sync point is implicit in CREATE MQCONN 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 command is successful, and rolled back if not.

Options

ATTRIBUTES (*data-value*)

Specifies the attributes of the MQCONN resource definition being added. You must code the list of attributes as a single character string by using the syntax shown in the syntax diagram.

See the *CICS Resource Definition Guide* for details about specific attributes. You can assign default values for all attributes of an MQCONN definition by specifying an ATTRLEN value of 0. You must still specify the ATTRIBUTES option, however, even though its value is not used.

ATTRLEN (*data-value*)

Specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length can be 0 - 32767.

LOGMESSAGE (*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

MQCONN (*data-value*)

Specifies the 8-character name of the MQCONN resource definition to be added to the CICS region.

Conditions

INVREQ

RESP2 values:

- n** The ATTRIBUTES string contains a syntax error, or an error occurred during either the discard or resource definition phase of the processing.
- 7** The LOGMESSAGE CVDA value is not valid.
- 200** The command was run in a program defined with an EXECUTIONSET value of DPLSUBSET or a program called from a remote system by a distributed program link without the SYNCONRETURN option.

LENGERR

RESP2 values:

- 1** The length that you have specified in ATTRLEN is negative.

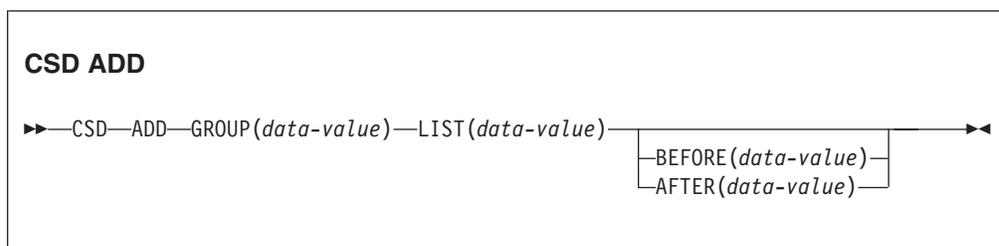
NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

CSD ADD

Add a group to a list in the CSD.



Conditions: CSDERR, DUPRES, INVREQ, LOCKED, NOTAUTH, NOTFND

Description

The **CSD ADD** command adds a group to a list, optionally specifying the position in the list using the BEFORE or AFTER options. If you do not specify BEFORE or AFTER, the group is added to the end of the list.

A syncpoint is implicit in **CSD ADD** 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 command is successful, and rolled back if not.

Options

AFTER(*data-value*)

Specifies the 8-character name of an existing group in the list after which the group is added.

BEFORE(*data-value*)

Specifies the 8-character name of an existing group in the list before which the group is added.

GROUP(*data-value*)

Specifies the 8-character name of the group to be added to the list. You can add a group to a list even if there are no resources in the group.

LIST(*data-value*)

Specifies the 8-character name of the list to which the group is added. If the list does not already exist, a new one is created.

Conditions

CSDERR

RESP2 value:

- 1 The CSD cannot be read
- 2 The CSD is READONLY.

- 3 The CSD is full
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

DUPRES

RESP2 value:

- 1 The group already exists in this list.
- 2 The specified group did not exist but a list of the same name is already present in the CSD. The group could not be added
- 3 The specified list did not exist but a group of the same name is already present in the CSD. The list could not be created.

INVREQ

RESP2 values:

- 2 The GROUP option contains one or more characters that are not valid.
- 3 The LIST option contains one or more characters that are not valid.
- 5 The BEFORE option contains one or more characters that are not valid.
- 6 The AFTER option contains one or more characters that are not valid.
- 200 The command was run in a program defined with an EXECUTIONSET value of DPLSUBSET or in a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LOCKED

RESP 2 values

- 1 The list is locked to another user and cannot be updated.
- 2 The list is IBM-protected.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

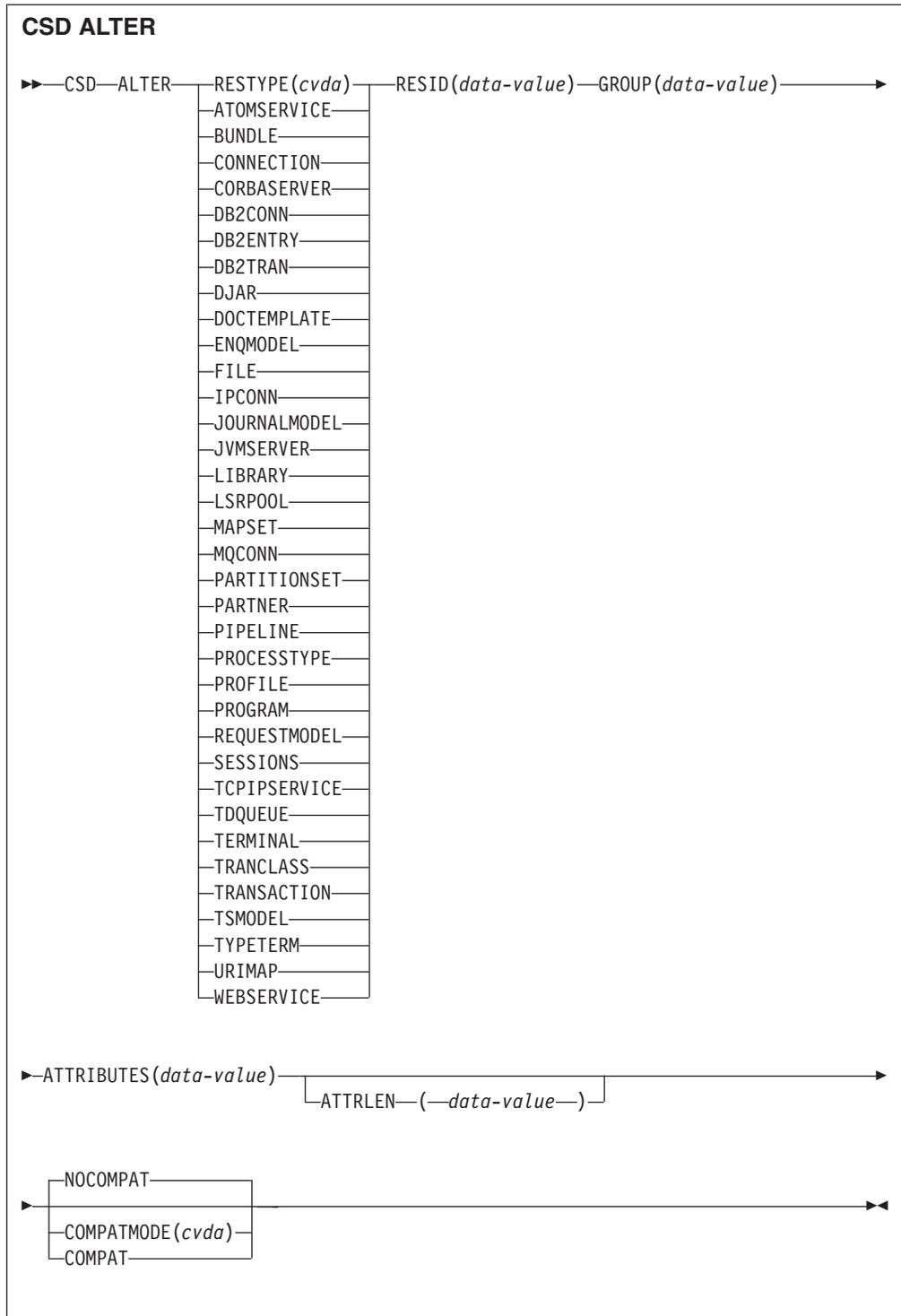
NOTFND

RESP2 value:

- 4 The specified BEFORE or AFTER group does not exist in the list.

CSD ALTER

Change the attributes of an existing resource definition in the CSD.



Conditions: CSDERR, INVREQ, LENGERR, LOCKED, NOTAUTH, NOTFND

Description

The CSD ALTER command changes some or all of the attributes of an existing resource definition in the CSD.

A syncpoint is implicit in **CSD ALTER** 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 command is successful, and rolled back if not.

Options

ATTRIBUTES (*data-value*)

Specifies the attributes to be changed. Code the list of attributes as a single character string.

See 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 fullword binary value.

GROUP (*data-value*)

Specifies the 8-character name of the group containing the resource definition.

RESID (*data-value*)

Specifies the 8-character name of the resource to be altered. Resource names such as TRANSACTION that are only four characters must be padded with four blanks and passed in an 8-character field.

RESTYPE (*cvda*)

Specifies the type of resource to be altered. CVDA values are the resource type names.

COMPATMODE (*cvda*)

Specifies whether obsolete attributes are allowed in the ATTRIBUTES string for this command. Specify one of the following CVDA values:

COMPAT

Obsolete resource attributes are allowed in the ATTRIBUTES string for this command.

NOCOMPAT

Obsolete resource attributes are not allowed in the ATTRIBUTES string for this command.

The default is NOCOMPAT.

Conditions

CSDERR

RESP2 values:

- 1 The CSD cannot be read.
- 2 The CSD is read only.
- 3 The CSD is full.
- 4 The CSD is being used by another CICS system and is not configured for sharing
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

INVREQ

RESP2 values:

- 1 RESTYPE did not specify a valid resource type

- 2 The GROUP option contains one or more characters that are not valid.
- 11 The value of COMPATMODE is not valid.
- 200 The command was run in a program defined with an EXECUTIONSET value of DPLSUBSET or in a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.
- n The value of ATTRIBUTES or RESID is not valid.

LENGERR

RESP2 value:

- 1 The length specified in ATTRLEN is negative.

LOCKED

RESP2 value:

- 1 The group is locked to another user and cannot be updated.
- 2 The group is IBM-protected.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 value:

- 1 The specified resource definition is not in the named group.
- 2 The named group does not exist.

CSD APPEND

Append the groups in one list on the CSD to the end of another list.

CSD APPEND

▶—CSD—APPEND—LIST(*data-value*)—TO(*data-value*)—◀

Conditions: CSDERR, DUPRES, INVREQ, LOCKED, NOTAUTH, NOTFND

Description

The CSD APPEND command appends the groups in one list on the CSD to the end of another list.

A syncpoint is implicit in **CSD APPEND** 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 command is successful, and rolled back if not.

Options

LIST(*data-value*)

Specifies the 8-character name of the list that is appended.

TO(*data-value*)

Specifies the 8-character name of the target list that is appended to. The list is created if it does not exist.

Conditions

CSDERR

RESP2 values:

- 1 The CSD could not be read.
- 2 The CSD is read only.
- 3 The CSD is full.
- 4 The CSD is being used by another CICS system and is not configured for sharing
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

DUPRES

RESP2 value:

- 3 Either the list specified in LIST is present in the CSD as a group, or the list specified in TO did not exist but the list cannot be created because a group of the same name is already present in the CSD.

INVREQ

RESP2 values:

- 3 The LIST option contains one or more characters that are not valid.
- 7 The TO option contains one or more characters that are not valid.
- 200 The command was run in a program defined with an EXECUTIONSET value of DPLSUBSET or in a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LOCKED

RESP2 values:

- 1 The TO list is locked to another user.
- 2 The TO list is IBM protected.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

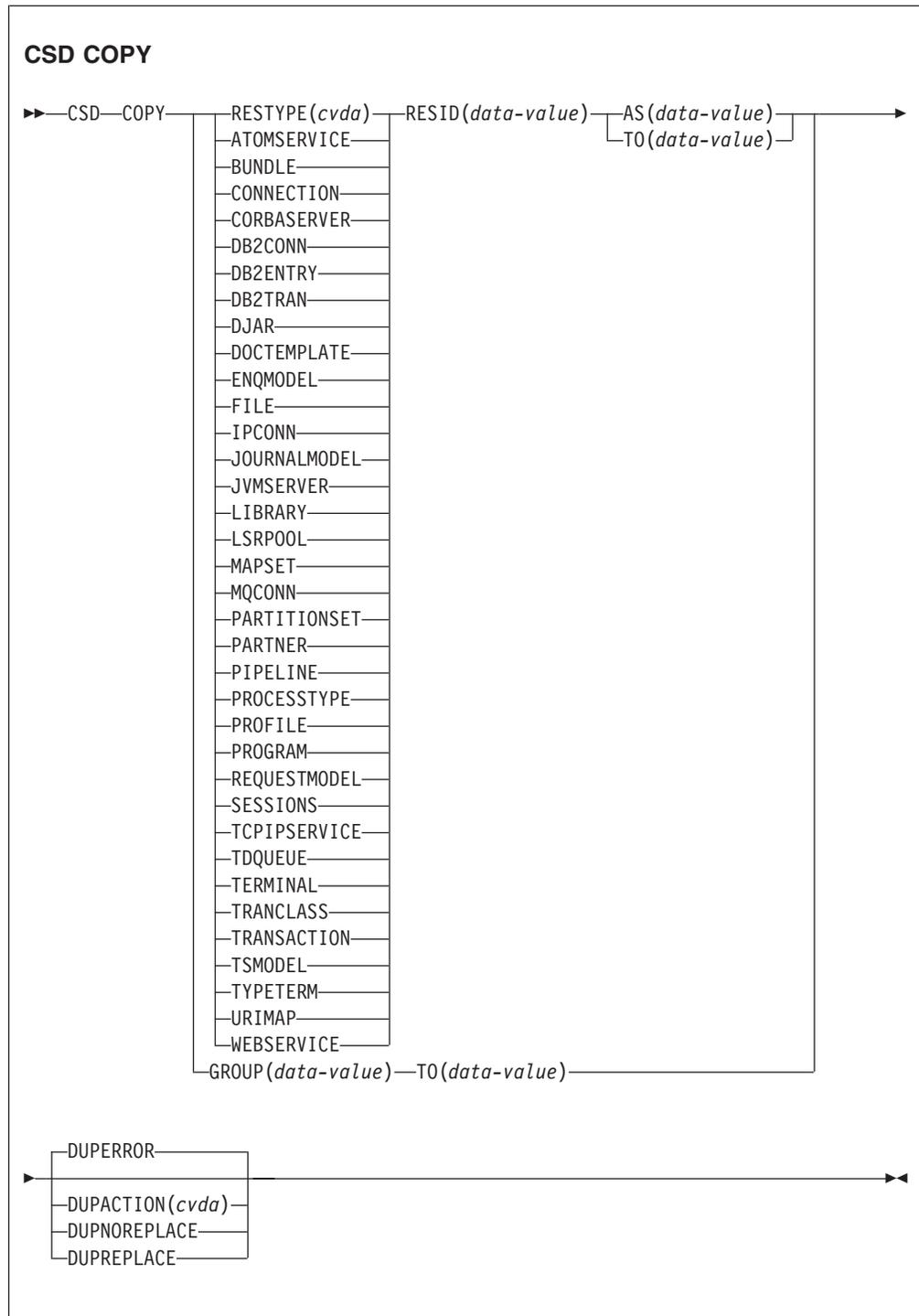
NOTFND

RESP2 value:

- 3 The list specified in the LIST option cannot be found.

CSD COPY

Copy a resource definition in a group to a different group, or copy an entire group.



Conditions: CSDERR, DUPRES, INVREQ, LOCKED, NOTAUTH, NOTFND

Description

The CSD COPY command performs the following operations:

- Copy an individual resource definition to the same group with a new name by using the RESTYPE, RESID and AS options but not TO.
- Copy an individual resource definition to a different group by using the RESTYPE, RESID and TO options.
- Copy an entire group by using the TO option without RESTYPE and RESID.

A syncpoint is implicit in **CSD COPY** 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 command is successful, and rolled back if not.

Options

AS(*data-value*)

Specifies the new 8-character name of an individual resource definition. For resources with 4-character names, the first four characters of this value are used.

GROUP(*data-value*)

Specifies the 8-character name of the group to be copied, or the group that contains the individual resource definition to be copied.

DUPACTION(*cvda*)

Specifies the required action when there are duplicate definitions in the target group. The default value is DUPERROR. CVDA values are as follows:

DUPERROR

Raises the DUPRES condition for duplicate definitions.

DUPNOREPLACE

Specifies that duplicate definitions in the target group are not replaced.

DUPREPLACE

Specifies that duplicate definitions in the target group are replaced.

RESID(*data-value*)

Specifies the 8-character name of the individual resource definition to be copied. Resource names such as TRANSACTION that are only four characters must be padded with four blanks and passed in an 8-character field.

RESTYPE(*cvda*)

Specifies the type of resource definition to be copied. CVDA values are the resource type names.

TO Specifies the 8-character name of the group to which the individual resource definition or whole group is to be copied. If an individual resource definition is specified and the TO option is not, the resource definition is copied in the same group. In this case you must specify the AS option. You must use the TO option if a whole group is to be copied. In all cases, the TO group is created if it does not exist.

Conditions

CSDERR

RESP2 value:

- 1 The CSD could not be read.
- 2 The CSD is read only.

- 3 The CSD is full.
- 4 The CSD is being used by another CICS system and is not configured for sharing
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

DUPRES

RESP2 value:

- 1 The group exists already (for a whole group copy) or one or more of the resource definitions to be created by the **COPY** already exists and **DUPACTION** was set or defaulted to **DUPERROR**.
- 2 The name specified in the **GROUP** or **TO** option is present in the CSD as a list.

INVREQ

RESP2 values:

- 1 The resource type specified for **RESTYPE** is not valid.
- 2 The **GROUP** option contains one or more characters that are not valid.
- 4 The **RESID** option contains one or more characters that are not valid.
- 7 The **TO** option contains one or more characters that are not valid.
- 9 The **DUPACTION** value is not valid.
- 200 The command was run in a program defined with an **EXECUTIONSET** value of **DPLSUBSET** or in a program invoked from a remote system by a distributed program link without the **SYNCONRETURN** option.

LOCKED

RESP2 values:

- 1 The target group is locked to another user.
- 2 The target group is IBM protected.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

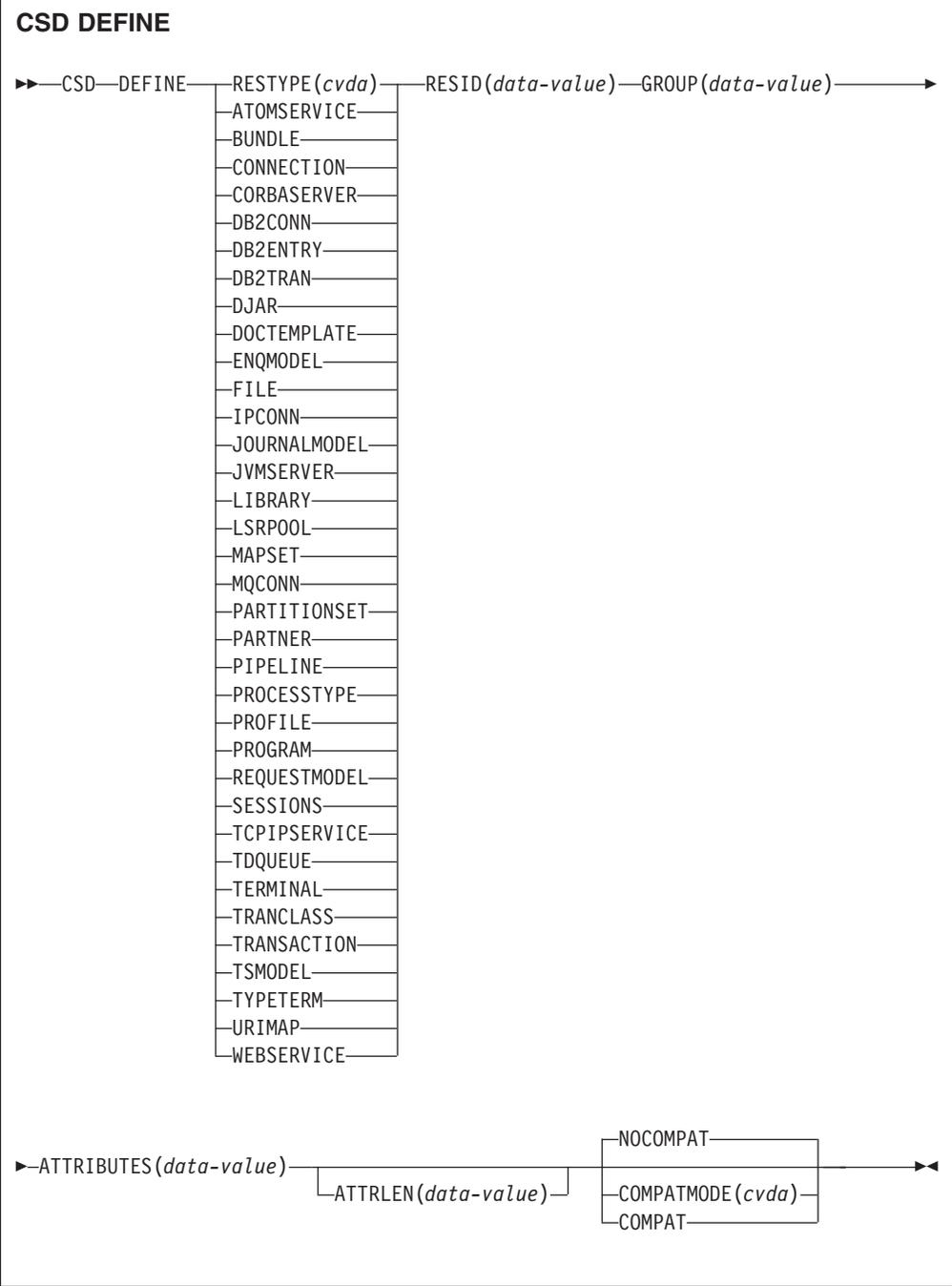
NOTFND

RESP2 values:

- 1 The specified resource definition cannot be found.
- 2 The source group cannot be found.

CSD DEFINE

Create a new resource definition in the CSD.



Conditions: CSDERR, DUPRES, INVREQ, LENGERR, LOCKED, NOTAUTH

Description

The CSD DEFINE command creates a new resource definition on the CSD.

A syncpoint is implicit in **CSD DEFINE** 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 command is successful, and rolled back if not.

Options

ATTRIBUTES (*data-value*)

Specifies the attributes of the new resource. Code the list of attributes as a single character string.

See 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 fullword binary value.

GROUP (*data-value*)

Specifies the 8-character name of the group containing the resource definition.

RESID (*data-value*)

Specifies the 8-character name of the resource to be defined. Resource names such as TRANSACTION that are only four characters in length must be padded with four blanks and passed in an 8-character field.

RESTYPE (*cvda*)

Specifies the type of resource definition to be defined. CVDA values are the resource type names.

COMPATMODE (*cvda*)

Specifies whether obsolete attributes are allowed in the ATTRIBUTES string for this command. Specify one of the following CVDA values:

COMPAT

Obsolete resource attributes are allowed in the ATTRIBUTES string for this command.

NOCOMPAT

Obsolete resource attributes are not allowed in the ATTRIBUTES string for this command.

The default is NOCOMPAT.

Conditions

CSDERR

RESP2 values:

- 1 The CSD cannot be read.
- 2 The CSD is read only.
- 3 The CSD is full.
- 4 The CSD is being used by another CICS system and is not configured for sharing
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

DUPRES

RESP2 values:

- 1 A resource of this name and type already exists in the specified group.
- 2 The specified group did not exist but because a list of the same name is already present in the CSD, the group could not be created.

INVREQ

RESP2 values:

- 1 The resource type specified for RESTYPE is not valid.
- 2 The GROUP option contains one or more characters that are not valid.
- 11 The value of COMPATMODE is not valid.
- 200 The command was run in a program defined with an EXECUTIONSET value of DPLSUBSET or in a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.
- n The ATTRIBUTES string contains a syntax error or RESID contains a character that is not valid.

LENGERR

RESP2 value:

- 1 The length specified in ATTRLEN is negative.

LOCKED

RESP2 values:

- 1 The specified group is locked to another user.
- 2 The group is IBM-protected.

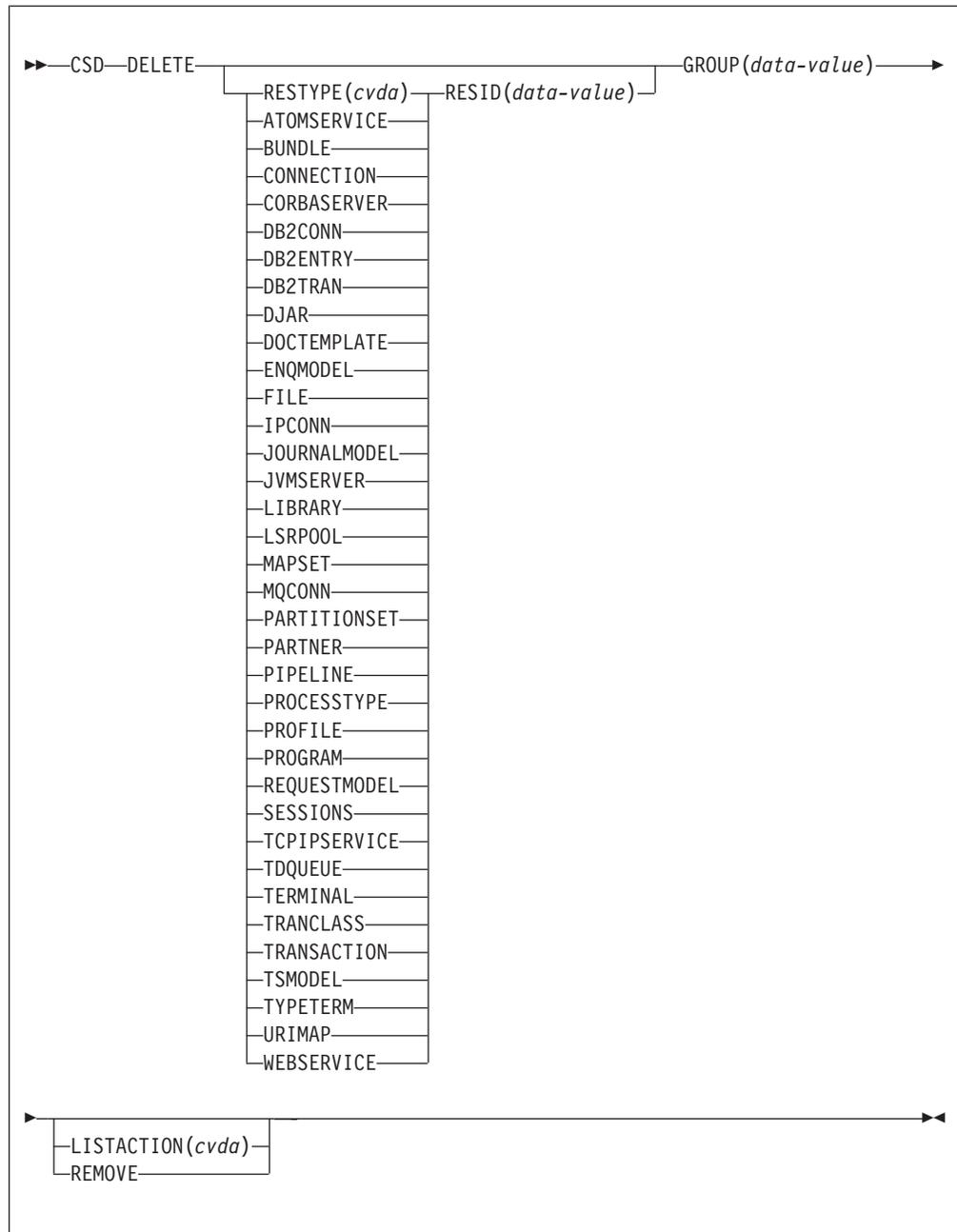
NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

CSD DELETE

Delete a group, or a single resource definition in a group, from the CSD.



Conditions: CSDERR, INVREQ, LOCKED, NOTAUTH, NOTFND

Description

The CSD DELETE command performs the following operations:

- Delete a single resource from a group in the CSD.
- Delete a whole group from the CSD.

A syncpoint is implicit in **CSD DELETE** 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 command is successful, and rolled back if not.

Options

GROUP(*data-value*)

Specifies the 8-character name of the group to be deleted, or the group containing the resource definition to be deleted.

LISTACTION(*cvda*)

Specifies the effect that group delete has on lists that contain the group. CVDA value is:

REMOVE

The group is removed from all lists that contain it.

RESID(*data-value*)

Specifies the 8-character name of the resource definition to be deleted. Resource names such as TRANSACTION that are only four characters in length must be padded with four blanks and passed in an 8-character field.

RESTYPE(*cvda*)

Specifies the type of resource to be deleted. CVDA values are the resource type names.

Conditions

CSDERR

RESP2 values:

- 1 The CSD cannot be read.
- 2 The CSD is read only.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

INVREQ

RESP2 values:

- 1 The resource type specified for RESTYPE is not valid.
- 2 The GROUP option contains one or more characters that are not valid.
- 4 The RESID option contains one or more characters that are not valid.
- 10 The value of LISTACTION is not valid.
- 200 The command was run in a program defined with an EXECUTIONSET value of DPLSUBSET or in a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LOCKED

RESP2 value:

- 1 The specified group is locked to another user.
- 2 The group is IBM-protected.

NOTAUTH

RESP2 value:



Conditions: CSDERR, NOTAUTH

Description

The ENDBRGROUP command stops a browse of groups in the CSD started by a CSD STARTBRGROUP command. The browse can be of all the groups in the CSD, or of all the groups in a specified list.

Options

LIST

Specifies that the browse being ended is of the groups in a list rather than all groups in the CSD.

Conditions

CSDERR

RESP2 value:

- 1 The CSD cannot be accessed.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

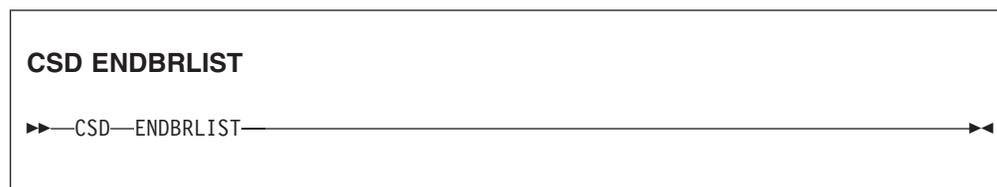
NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

CSD ENDBRLIST

End the current browse of the lists in the CSD.



Conditions: CSDERR, NOTAUTH

Description

The CSD ENDBRLIST command stops a browse of the lists in the CSD started by a CSD STARTBRLIST command.

Conditions

CSDERR

RESP2 value:

- 1 The CSD cannot be accessed.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

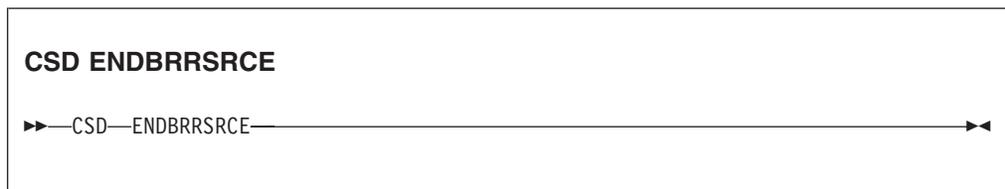
NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

CSD ENDBRRSRCE

End the current browse of the resources in a specified group.



Conditions: CSDERR, NOTAUTH

Description

The **CSD ENDBRRSRCE** command stops a browse of the resource definitions in a group in the CSD that was started by a **CSD STARTBRRSRCE** command.

Conditions

CSDERR

RESP2 value:

- 1 The CSD cannot be accessed.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

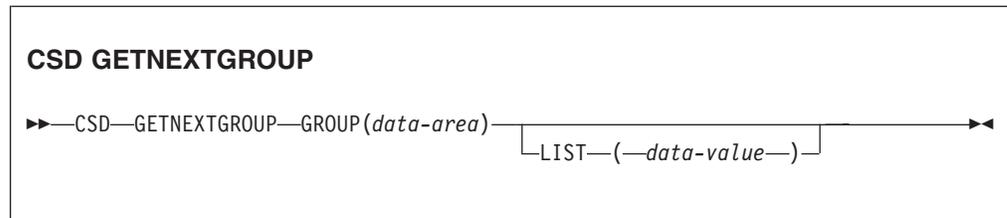
NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

CSD GETNEXTGROUP

Get the next group in a group browse.



Conditions: CSDERR, END, ILLOGIC, NOTAUTH

Description

The **CSD GETNEXTGROUP** command returns the name of the next group in the browse started by a **CSD STARTBRGROUP** command.

Options

GROUP(*data-area*)

Returns the 8-character name of the group.

LIST(*data-value*)

Specifies the 8-character name of the list to which the browse was limited on the STARTBRGROUP command. You must specify a value for LIST if the associated STARTBRGROUP command includes the LIST option.

Conditions

CSDERR

RESP2 value:

- 1 The CSD cannot be read.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

END

RESP2 value:

- 1 The CSD or the list contain no more groups.

ILLOGIC

RESP2 value:

- 1 A group browse is not in progress.

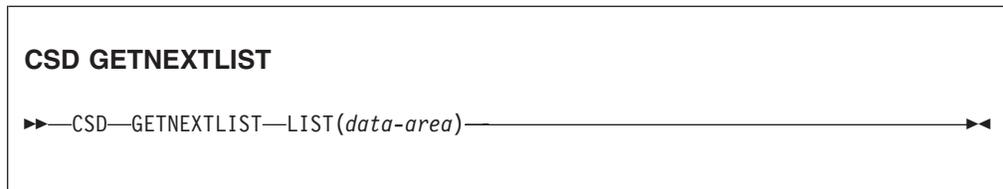
NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

CSD GETNEXTLIST

Get the next list in a list browse.



Conditions: CSDERR, END, ILLOGIC, NOTAUTH

Description

The **CSD GETNEXTLIST** command returns the name of the next list in a browse started by a **CSD STARTBRLIST** command.

Options

LIST (*data-area*)

Returns the 8-character name of the list.

Conditions

CSDERR

RESP2 value:

- 1 The CSD cannot be read.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

END

RESP2 value:

- 1 The CSD contains no more lists.

ILLOGIC

RESP2 value:

- 1 A list browse is not in progress.

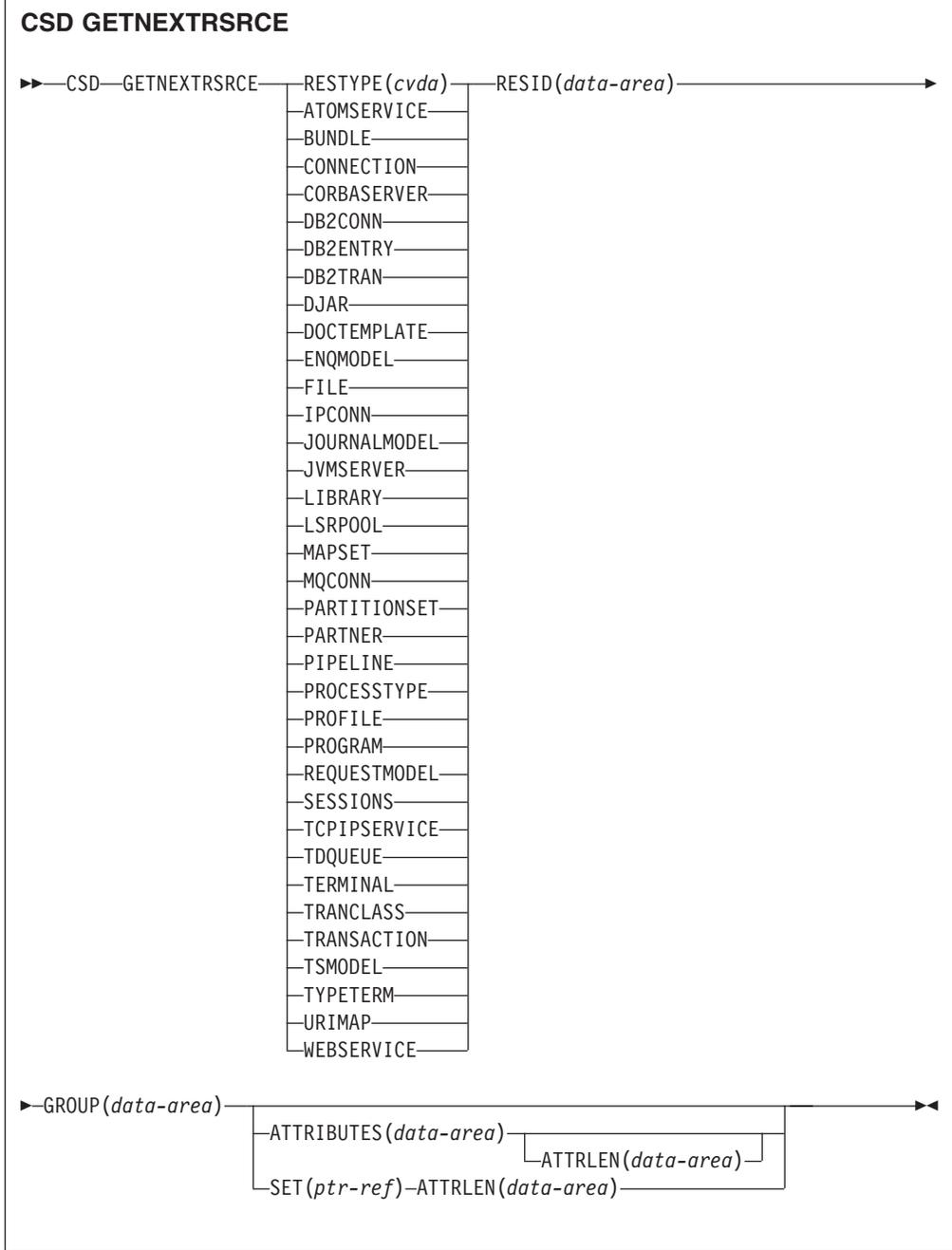
NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

CSD GETNEXTSRCE

Get the details of the next resource in a resource browse.



Conditions: CSDERR, END, ILLOGIC, LENGERR, NOTAUTH

Description

The **CSD GETNEXTRSRCE** command returns the details of the next resource in a browse started by a **CSD STARTBRRSRCE** command.

Options

ATTRIBUTES(*data-area*)

Specifies the data area in which a character string containing a list of attributes of the relevant resource is returned.

See the *CICS Resource Definition Guide* for details about specific attributes.

ATTRLEN(*data-area*)

A fullword binary field containing one of the following values:

- When used with the ATTRIBUTES option:
 - On input ATTRLEN contains the maximum length of the attributes string that the application can accept. You do not need to specify ATTRLEN if the length can be generated by the compiler from the ATTRIBUTES variable.
 - On output, ATTRLEN contains the length of the attributes string returned. The LENGERR condition is raised if the attribute string is longer than the input ATTRLEN value.
- When used with the SET option, ATTRLEN is an output-only field that is set to the length of the attributes string.

GROUP(*data-area*)

Returns the 8-character name of the group being browsed.

RESID(*data-area*)

Returns the 8-character name of the resource definition whose attributes are returned. Resource names such as TRANSACTION that are only four characters are padded with four blanks.

RESTYPE(*cvda*)

Returns the type of the resource definition. CVDA values are the resource type names.

SET(*ptr-ref*)

Specifies a pointer reference that is set to the address of the returned attributes string. The pointer reference is valid until the next CSD GETNEXTRSRCE resource command is issued, or until the end of the task.

Conditions

CSDERR

RESP2 value:

- 1 The CSD cannot be read.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

END

RESP2 value:

- 1 The CSD or the specified list contains no more groups.

ILLOGIC

RESP2 value:

- 1 No resource browse is in progress.

LENGERR

RESP2 value:

- 1 The length of the ATTRIBUTES data area as specified on the ATTRLEN option passed to CICS is negative.
- 2 The length of the ATTRIBUTES data area as specified on the ATTRLEN option passed to CICS is less than the amount of data to be returned.

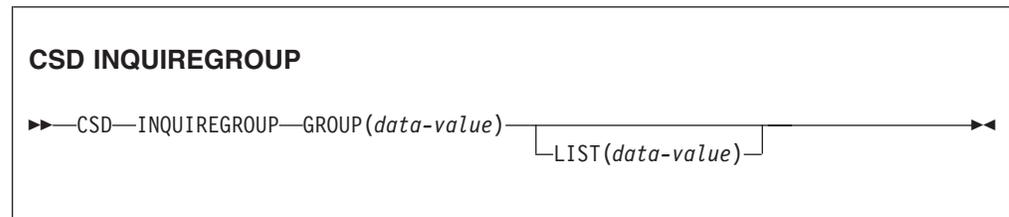
NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

CSD INQUIREGROUP

Inquire on a group in the CSD or on a group in a specified list in the CSD.



Conditions:CSDERR, NOTAUTH, NOTFND

Description

Use the **CSD INQUIREGROUP** command to make a direct inquiry of group names on the CSD. You can limit the scope of the inquiry to a specified list. If the response is *NORMAL*, the specified group exists on the CSD or in the list.

Options**GROUP**(*data-value*)

Specifies the 8-character name of the group being queried.

LIST(*data-value*)

Specifies the 8-character name of a list to which the scope of the query is limited.

Conditions**CSDERR**

RESP2 value:

- 1 The CSD cannot be read.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 values:

- 2 The specified group cannot be found.
- 3 The specified list cannot be found.

CSD INQUIRELIST

Inquire on a list in the CSD.

```
CSD INQUIRELIST
▶—CSD—INQUIRELIST—LIST(data-value)—▶
```

Conditions: CSDERR, NOTAUTH, NOTFND

Description

The **CSD INQUIRELIST** command makes a direct inquiry of list names on the CSD. If the response is **NORMAL**, the specified group exists on the CSD.

Options

LIST (*data-value*)

Specifies the 8-character name of the list being queried.

Conditions

CSDERR

RESP2 value:

- 1 The CSD cannot be read.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

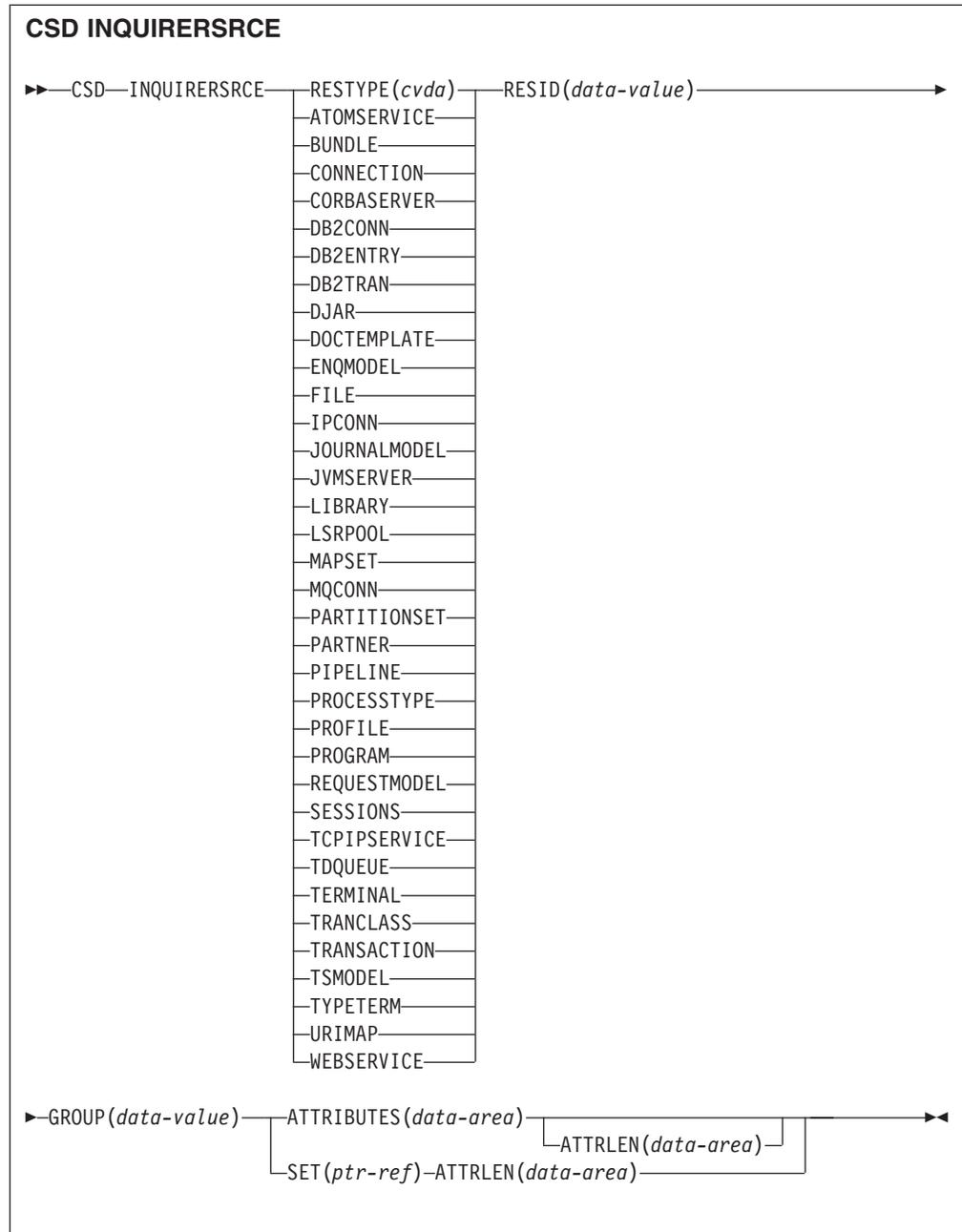
NOTFND

RESP2 value:

- 3 The specified list cannot be found.

CSD INQUIRERSRCE

Inquire on the attributes of a resource in a specified group in the CSD.



Conditions: CSDERR, INVREQ, LENGERR, NOTAUTH, NOTFND

Description

The **CSD INQUIRERSRCE** command queries the attributes of resources in a specified group on the CSD.

Options

ATTRIBUTES (data-area)

Specifies the data area in which a character string containing a list of attributes of the relevant resource is returned.

See the *CICS Resource Definition Guide* for details about specific attributes.

ATTRLEN(*data-area*)

A fullword binary field containing one of the following values:

- When used with the ATTRIBUTES option:
 - On input, ATTRLEN contains the maximum length of the attributes string that the application can accept. You do not have to specify ATTRLEN if the length can be generated by the compiler from the ATTRIBUTES variable.
 - On output, ATTRLEN contains the length of the attributes string returned. The LENGERR condition is raised if the attribute string is longer than the input ATTRLEN value.
- When used with the SET option, ATTRLEN is an output-only field that is set to the length of the attributes string.

GROUP(*data-value*)

Returns the 8-character name of the group containing the resource definition or resource definition attributes being queried.

RESID(*data-value*)

Returns the 8-character name of the resource definition whose attributes are being queried. Resource names such as TRANSACTION that are only four characters in length must be padded with four blanks.

RESTYPE(*cvda*)

Returns the resource type of the resource definition being queried. CVDA values are the resource type names.

SET(*ptr-ref*)

Specifies a pointer reference that is set to the address of the returned attributes string. The pointer reference is valid until the next CSD INQUIRERSRCE resource command is issued or until the end of the task.

Conditions

CSDERR

RESP2 value:

- 1 The CSD cannot be read.
- 2 The CSD is defined as readonly.
- 3 There is no more space available in the CSD.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

INVREQ

RESP2 values:

- 1 The specified RESTYPE is not a valid resource type.
- 4 The RESID option contains one or more characters that are not valid for the specific resource type.

LENGERR

RESP2 value:

- 1 The length of the ATTRIBUTES data area as specified on the ATTRLEN option passed to CICS is negative.

- 2 The length of the ATTRIBUTES data area as specified on the ATTRLEN option passed to CICS is less than the amount of data to be returned.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 value:

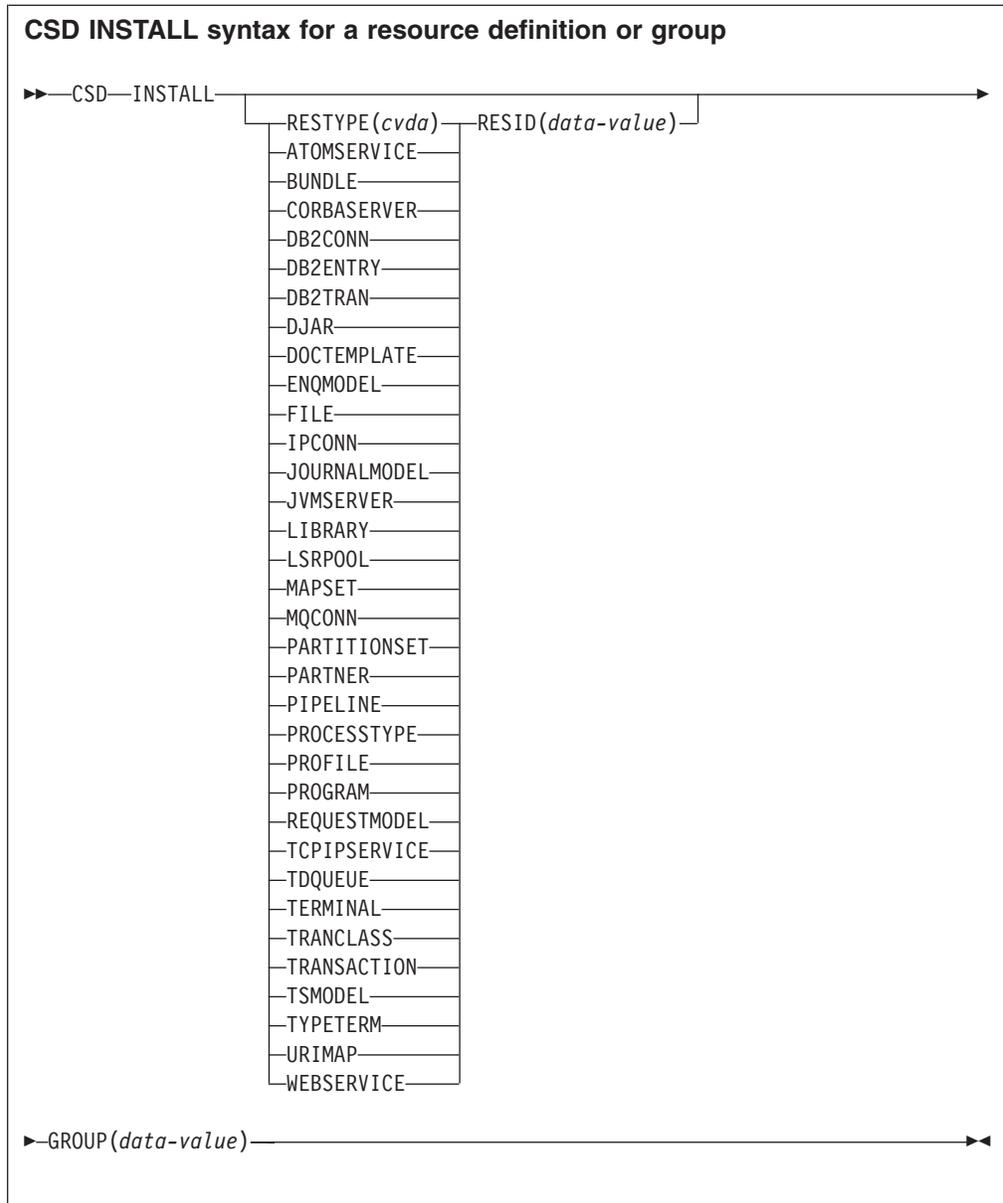
- 1 The specified resource definition cannot be found.
- 2 The specified group cannot be found.

CSD INSTALL

Install a list, a group, or a single resource definition in a group, from the CSD.

CSD INSTALL syntax for a list

►►—CSD—INSTALL—LIST(*data-value*)—————◄◄



Conditions: CSDERR, INCOMPLETE, INVREQ, NOTAUTH, NOTFND

Description

The **CSD INSTALL** command performs the following operations:

- Install a single resource from a group in the CSD.
- Install a whole group from the CSD.
- Install a list from the CSD.

If a whole group or list is installed, some of the individual resources might fail to install. In this case, the INCOMPLETE condition is raised and the relevant warning and error messages are written to the CSDE transient data queue.

Single resource INSTALL is not supported for CONNECTION, SESSIONS or TERMINAL pools. INSTALL GROUP or INSTALL LIST enable a CONNECTION or TERMINAL pool to be installed with one command.

A syncpoint is implicit in **CSD INSTALL** 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 command is successful, and rolled back if not.

Options

GROUP(*data-value*)

Specifies the 8-character name of the group to be installed, or the group containing the individual resource definition to be installed.

LIST(*data-value*)

Specifies the 8-character name of the list to be installed.

RESID(*data-value*)

Specifies the 8-character name of the individual resource definition to be installed. Resource names such as TRANSACTION that are only 4 characters in length must be padded with 4 blanks and passed in an 8-character field.

RESTYPE(*cvda*)

Specifies the type of resource to be installed. CVDA values are the resource type names.

Conditions

CSDERR

RESP2 value:

- 1 The CSD cannot be read.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

INCOMPLETE

RESP2 value:

- 1 The install of a complete group or list was only partially successful.

INVREQ

RESP2 values:

- 1 The resource type specified for RESTYPE is not valid.
- 2 The GROUP option contains one or more characters that are not valid.
- 3 The LIST option contains one or more characters that are not valid.
- 4 The RESID option contains one or more characters that are not valid.
- 200 The command was run in a program defined with an EXECUTIONSET value of DPLSUBSET, or in a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.
- n An error occurred while adding the specified resource to the running system due to the current state of the system.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

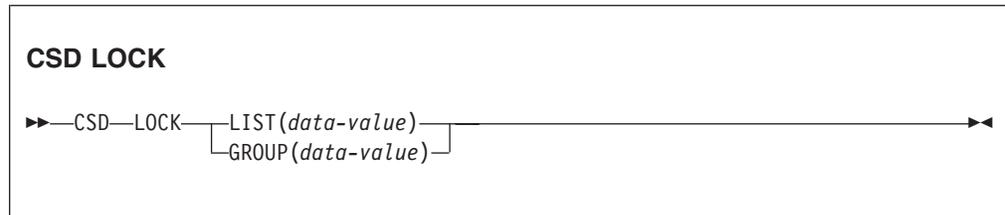
NOTFND

RESP2 values:

- 1 The specified resource definition cannot be found.
- 2 The specified group cannot be found.
- 3 The specified list cannot be found.

CSD LOCK

Restrict update and delete access for a group or list to a single operator identifier.



Conditions: CSDERR, DUPRES, INVREQ, LOCKED, NOTAUTH

Description

When you lock a group or list, other users can view or copy it but they are restricted from changing or deleting it. You can lock a nonexistent group or list to reserve the named group or list for your own future use. The only command that releases a lock is the **UNLOCK** command. No other RDO commands can unlock a group or list. For example, if you delete all the resources in a group, or all the groups in a list, the lock remains.

You use the **LOCK** and **UNLOCK** commands to control update access to a group or list so that only operators with the same operator identifier can make changes.

Users who are not signed on or who have a different operator identifier (OPIDENT) are not allowed to perform any operation that changes the locked list or group. However, any user is allowed to perform the following operations on a locked group or list:

- CHECK (CEDA)
- COPY
- DISPLAY (CEDA)
- INQUIRE/BROWSE
- INSTALL
- VIEW (CEDA)

Only a user on the same system and with the same operator identifier can remove the lock, using the **UNLOCK** command.

A syncpoint is implicit in **CSD LOCK** 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 command is successful, and rolled back if not.

Options

GROUP (*data-value*)

Specifies the 8-character name of the group to be locked.

LIST (*data-value*)

Specifies the 8-character name of the list to be locked.

Conditions

CSDERR

RESP2 values:

- 1 The CSD cannot be read.
- 2 The CSD is read only.
- 3 The CSD is full.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

DUPRES

RESP2 values:

- 2 The name specified in **GROUP** exists in the CSD as a list.
- 3 The name specified in **LIST** exists in the CSD as a group.

INVREQ

RESP2 values:

- 2 The **GROUP** option contains one or more characters that are not valid.
- 3 The **LIST** option contains one or more characters that are not valid.
- 200 The command was run in a program defined with an **EXECUTIONSET** value of **DPLSUBSET** or in a program invoked from a remote system by a distributed program link without the **SYNCONRETURN** option.

LOCKED

RESP2 values:

- 1 The group or list is already locked to another user.
- 2 The group or list is IBM-protected.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

CSD REMOVE

Remove a group from a list in the CSD.

CSD REMOVE

►—CSD—REMOVE—GROUP(*data-value*)—LIST(*data-value*)—◄

Conditions: CSDERR, INVREQ, LOCKED, NOTAUTH, NOTFND

Description

The **CSD REMOVE** command removes a group from a list.

The group and all its resource definitions still exist in the CSD file. When the last group is removed from a list, the list no longer exists in the CSD file.

When a group is deleted, you can request that the group is removed from all lists that contained it. When the last group is removed from a list, the list is deleted.

A syncpoint is implicit in **CSD REMOVE** 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 command is successful, and rolled back if not.

Options

GROUP(*data-value*)

Specifies the 8-character name of the group to be removed.

LIST(*data-value*)

Specifies the 8-character name of the list from which the group is to be removed.

Conditions

CSDERR

RESP2 values:

- 1 The CSD cannot be read.
- 2 The CSD is read only.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

INVREQ

RESP2 values:

- 2 The GROUP option contains one or more characters that are not valid.
- 3 The LIST option contains one or more characters that are not valid.
- 200 The command was run in a program defined with an EXECUTIONSET value of DPLSUBSET or in a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LOCKED

RESP2 values:

- 1 The list is locked to another user and cannot be updated.

2 The list is IBM-protected.

NOTAUTH

RESP2 value:

100 The user associated with the issuing task is not authorized to use this command.

NOTFND

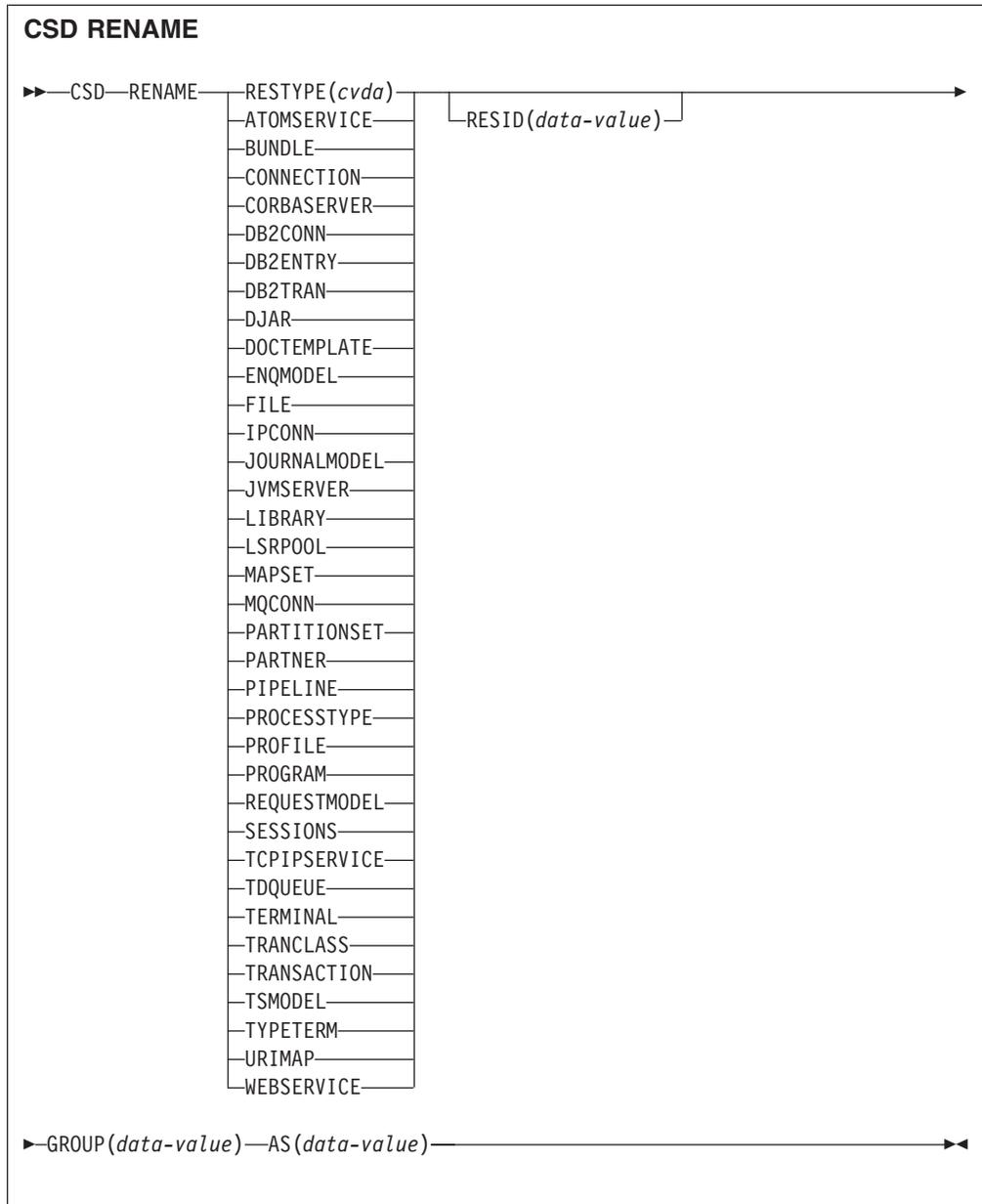
RESP2 values:

2 The specified group cannot be found.

3 The specified list cannot be found.

CSD RENAME

Rename a resource definition in the CSD.



Conditions: CSDERR, DUPRES, INVREQ, LOCKED, NOTAUTH, NOTFND

Description

The **CSD RENAME** command renames an individual resource definition in a specified group.

A syncpoint is implicit in **CSD RENAME** 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 command is successful, and rolled back if not.

Options

AS(*data-value*)

Specifies the new 8-character name of the resource definition. For resources with 4-character names, the first four characters of this value are used.

GROUP(*data-value*)

Specifies the 8-character name of the group containing the resource definition to be renamed.

RESID(*data-value*)

Specifies the 8-character name of the resource to be renamed. Resource names that are only four characters in length must be padded with four blanks and passed in an 8-character field.

Conditions

CSDERR

RESP2 values:

- 1 The CSD cannot be read.
- 2 The CSD is read only.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

DUPRES

RESP2 value:

- 1 The specified resource definition already exists.

INVREQ

RESP2 values:

- 1 The resource type specified for **RESTYPE** is not valid.
- 2 The **GROUP** option contains one or more characters that are not valid.
- 4 The **RESID** or **AS** option contains one or more characters that are not valid.
- 200 The command was run in a program defined with an **EXECUTIONSET** value of **DPLSUBSET** or in a program invoked from a remote system by a distributed program link without the **SYNCONRETURN** option.

LOCKED

RESP2 values:

- 1 The specified group is already locked to another user.
- 2 The group is IBM-protected.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

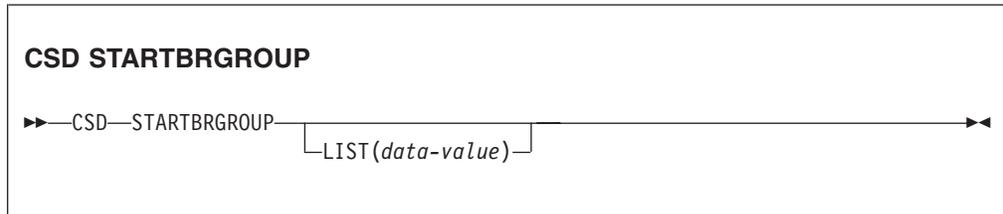
NOTFND

RESP2 values:

- 1 The specified resource definition cannot be found.

CSD STARTBRGROUP

Start a browse of the groups in the CSD or of the groups in a list.



Conditions: CSDERR, ILLOGIC, NOTAUTH, NOTFND

Description

The **CSD STARTBRGROUP** command starts a CSD group browse. The browse can be of all the groups in the CSD, or of all the groups in a specified list.

Rules for concurrent browses

There are four distinct types of browse that can be used on the CSD:

1. LIST browse of all the lists in the CSD.
2. ALL GROUPS browse of all the groups in the CSD.
3. GROUPS IN LIST browse of all the groups in a specified list.
4. RESOURCE browse of all resources in a specified group.

The rules that govern concurrent browses in the same transaction are as follows:

- Browses of the same type are not allowed together.
- The LIST and ALL GROUPS browse are not allowed together.
- The GROUPS IN LIST and RESOURCE browses are not allowed together.

The ILLOGIC condition is raised if these rules are broken.

For example, it is permitted to browse all groups in the CSD and, as each one is returned, browse all of the resources in that group.

Options

LIST(*data-value*)

Specifies the 8-character name of a list to which the browse is to be limited. If you do not specify this option, all groups in the CSD are returned on the browse.

Conditions

CSDERR

RESP2 value:

- 1 The CSD cannot be read.
- 4 The CSD is being used by another CICS system and is not configured for sharing.

- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

ILLOGIC

RESP2 value:

- 2 A browse of the same type or a conflicting type is already in progress.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

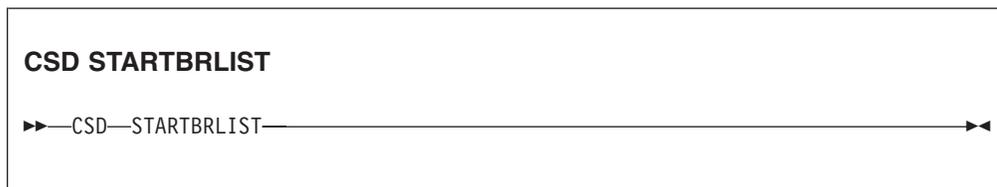
NOTFND

RESP2 value:

- 3 The specified list cannot be found.

CSD STARTBRLIST

Start a browse of the lists in the CSD.



Conditions: CSDERR, ILLOGIC, NOTAUTH,

Description

The **CSD STARTBRLIST** command starts a browse of all the lists in the CSD.

Rules for concurrent browses

There are four distinct types of browse that can be used on the CSD:

1. LIST browse of all the lists in the CSD.
2. ALL GROUPS browse of all the groups in the CSD.
3. GROUPS IN LIST browse of all the groups in a specified list.
4. RESOURCE browse of all resources in a specified group.

The rules that govern concurrent browses in the same transaction are as follows:

- Browses of the same type are not allowed together.
- The LIST and ALL GROUPS browse are not allowed together.
- The GROUPS IN LIST and RESOURCE browses are not allowed together.

The ILLOGIC condition is raised if these rules are broken.

For example, it is permitted to browse all groups in the CSD and, as each one is returned, browse all of the resources in that group.

Conditions

CSDERR

RESP2 value:

- 1 The CSD cannot be read.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

ILLOGIC

RESP2 value:

- 2 A list browse or a conflicting browse is already in progress.

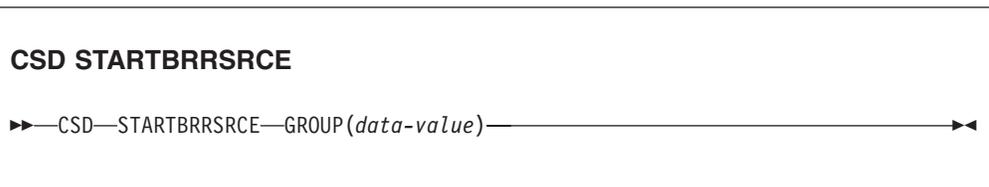
NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

CSD STARTBRRSRCE

Start a browse of the resources in a specified group.



Conditions: CSDERR, ILLOGIC, NOTAUTH, NOTFND

Description

The **CSD STARTBRRSRCE** command starts a browse of all the resource definitions in a specified group.

Rules for concurrent browses

There are four distinct types of browse that can be used on the CSD:

1. LIST browse of all the lists in the CSD.
2. ALL GROUPS browse of all the groups in the CSD.
3. GROUPS IN LIST browse of all the groups in a specified list.
4. RESOURCE browse of all resources in a specified group.

The rules that govern concurrent browses in the same transaction are as follows:

- Browses of the same type are not allowed together.
- The LIST and ALL GROUPS browse are not allowed together.
- The GROUPS IN LIST and RESOURCE browses are not allowed together.

The ILLOGIC condition is raised if these rules are broken.

For example, it is permitted to browse all groups in the CSD and, as each one is returned, browse all of the resources in that group.

Options

GROUP (*data-value*)

Specifies the 8-character name of the group to be browsed.

Conditions

CSDERR

RESP2 value:

- 1 The CSD cannot be read.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

ILLOGIC

RESP2 value:

- 2 A resource browse or a conflicting browse is already in progress.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

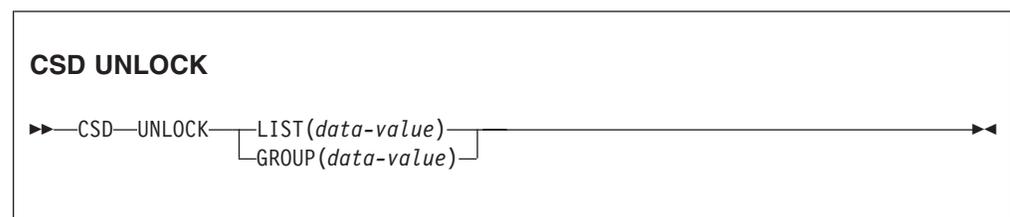
NOTFND

RESP2 value:

- 2 The specified group cannot be found.

CSD UNLOCK

Remove the lock from a group or list of definitions.



Conditions: CSDERR, DUPRES, INVREQ, LOCKED, NOTAUTH

Description

The **UNLOCK** command removes from a group or list a lock previously added by the **LOCK** command.

A syncpoint is implicit in **CSD UNLOCK** 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 command is successful, and rolled back if not.

Options

GROUP (*data-value*)

Specifies the 8-character name of the group to be unlocked.

LIST (*data-value*)

Specifies the 8-character name of the list to be unlocked.

Conditions

CSDERR

RESP2 values:

- 1 The CSD cannot be read.
- 2 The CSD is read only.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

DUPRES

RESP2 values:

- 2 The name specified in **GROUP** exists in the CSD as a list.
- 3 The name specified in **LIST** exists in the CSD as a group.

INVREQ

RESP2 values:

- 2 The **GROUP** option contains one or more characters that are not valid.
- 3 The **LIST** option contains one or more characters that are not valid.
- 200 The command was run in a program defined with an **EXECUTIONSET** value of **DPLSUBSET** or in a program invoked from a remote system by a distributed program link without the **SYNCONRETURN** option.

LOCKED

RESP2 values:

- 1 The specified group or list is already locked to another user.
- 2 The group or list is IBM-protected.

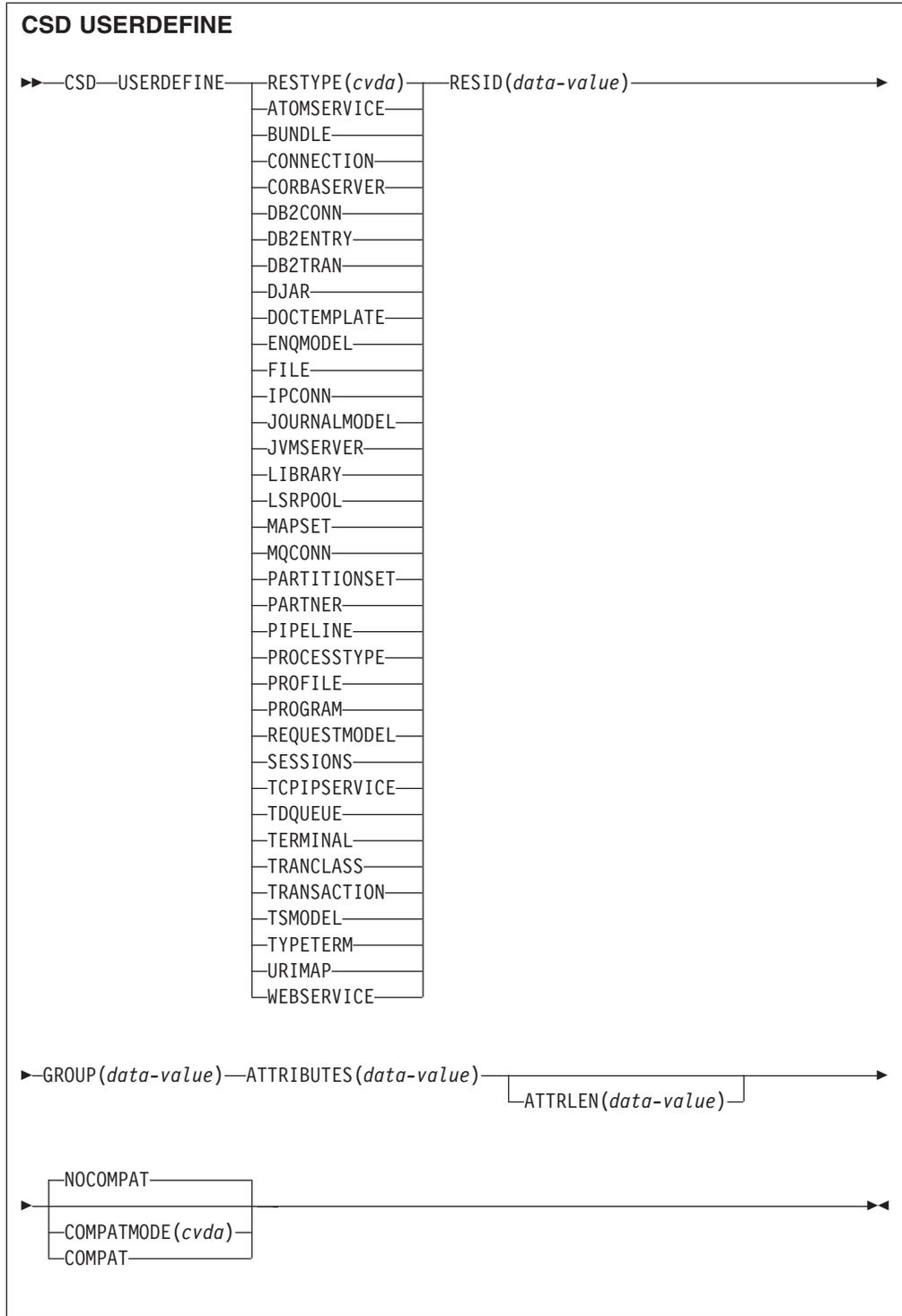
NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

CSD USERDEFINE

Create a new resource definition with user-specified default values in the CSD.



Conditions: CSDERR, DUPRES, INVREQ, LENGERR, LOCKED, NOTAUTH, NOTFND

Description

USERDEFINE is an alternative to the **DEFINE** command. Instead of using CICS-supplied default values, **USERDEFINE** uses your own defaults. Otherwise, it operates in exactly the same way as **DEFINE**.

To set up your own defaults, use **DEFINE** to create a dummy resource definition named **USER** in a group named **USERDEF**. Each dummy resource definition must be complete, for example, a transaction definition must name a program definition, even though you always supply a program name when you **USERDEFINE** a transaction. You need not install the dummy resource definitions before using **USERDEFINE**.

Create a dummy resource definition for each type of resource for which you want to set default values. Each of them is named **USER**, but because they are definitions of different resources, they are unique.

A syncpoint is implicit in **CSD USERDEFINE** 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 command is successful, and rolled back if not.

Options

ATTRIBUTES(*data-value*)

Specifies the attributes of the new resource. Code the list of attributes as a single character string.

See 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 fullword binary value.

COMPATMODE(*cvda*)

Specifies whether obsolete attributes are allowed in the **ATTRIBUTES** string for this command. Specify one of the following **CVDA** values:

COMPAT

Obsolete resource attributes are allowed in the **ATTRIBUTES** string for this command.

NOCOMPAT

Obsolete resource attributes are not allowed in the **ATTRIBUTES** string for this command.

The default is **NOCOMPAT**.

GROUP(*data-value*)

Specifies the 8-character name of the group to which the resource definition is to belong.

RESID(*data-value*)

Specifies the 8-character name of the resource to be altered. Resource names such as **TRANSACTION** that are only four characters must be padded with four blanks and passed in an 8-character field.

RESTYPE(*cvda*)

Specifies the type of resource to be defined. **CVDA** values are the resource type names.

Conditions

CSDERR

RESP2 values:

1 The CSD cannot be read.

- 2 The CSD is read only.
- 3 The CSD is full.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

DUPRES

RESP2 values:

- 1 A resource of this name and type already exists in the specified group.
- 2 The specified group did not exist but because a list of the same name is already present in the CSD, the group could not be created.

INVREQ

RESP2 values:

- 1 The resource type specified for RESTYPE is not valid.
- 2 The GROUP option contains one or more characters that are not valid.
- 11 The value specified for COMPATMODE is not valid.
- 200 The command was run in a program defined with an EXECUTIONSET value of DPLSUBSE, or in a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.
- n The ATTRIBUTES string contains a syntax error or RESID contains a character that is not valid.

LENGERR

RESP2 value:

- 1 The length specified in ATTRLEN is negative.

LOCKED

RESP2 values:

- 1 The specified group is locked to another user.
- 2 The group is IBM-protected.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 values:

- 5 The group USERDEF cannot be found.
- 6 A resource of the required type with name USER cannot be found in group USERDEF.

DISCARD ATOMSERVICE

Remove an ATOMSERVICE resource definition from the system.

DISCARD ATOMSERVICE

►►—DISCARD ATOMSERVICE(*data-value*)—◄◄

Conditions: INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

The DISCARD ATOMSERVICE command removes a disabled ATOMSERVICE definition from the system.

Options

ATOMSERVICE(*data-value*)

Specifies the 8-character name of the ATOMSERVICE definition that is to be removed.

Conditions

INVREQ

RESP2 values are:

- 4 The ATOMSERVICE is not disabled.
- 200 The command ran 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.

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 particular resource in the way required by this command.

NOTFND

RESP2 values:

- 3 The ATOMSERVICE cannot be found.

DISCARD BUNDLE

Remove a BUNDLE resource definition.

DISCARD BUNDLE

►►—DISCARD BUNDLE(*data-value*)—◄◄

Conditions: INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

Use the **DISCARD BUNDLE** command to remove a BUNDLE resource from your CICS region. All the resources that were dynamically created must be disabled or unusable before the BUNDLE resource can be discarded.

Options

BUNDLE(*data-value*)

Specifies the 8-character name of the BUNDLE definition that is to be discarded.

Conditions

INVREQ

RESP2 values:

- 5 The BUNDLE is not disabled.
- 6 The bundle is in an invalid state. You must disable the bundle before it can be discarded.
- 7 CICS failed to link to the registered bundle callback program.

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 particular resource in the way required by this command.

NOTFND

RESP2 values:

- 3 The BUNDLE cannot be found.

DISCARD JVMSERVER

Remove a JVMSERVER resource definition.

DISCARD JVMSERVER

►►—DISCARD JVMSERVER(*data-value*)—◀◀

Conditions: INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

Use the DISCARD JVMSERVER command to remove a JVMSERVER resource from your CICS region. The JVMSERVER resource must be disabled before it can be discarded.

Options

JVMSERVER(*data-value*)

Specifies the 8-character name of the JVMSERVER resource definition that is to be discarded.

Conditions

INVREQ

RESP2 value:

5 The JVMSERVER is not disabled.

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 particular resource in the way required by this command.

NOTFND

RESP2 value:

3 The JVMSERVER cannot be found.

DISCARD MQCONN

Remove an MQCONN resource definition. Any implicit MQINI resource definition is also discarded.

DISCARD MQCONN

▶▶—DISCARD MQCONN—▶▶

Conditions: INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

The DISCARD MQCONN command removes an installed MQCONN resource definition from the local CICS system. Only one MQCONN resource definition can be installed in a CICS system at a time, so no name or identifier is required on this command.

An MQCONN resource definition can be discarded only when CICS is not connected to WebSphere MQ.

If the installed MQCONN resource definition had an INITQNAME attribute, CICS installed an implicit MQINI resource definition to represent the default initiation queue. When you discard the MQCONN resource definition, CICS discards any associated MQINI resource definition.

Options

None

Conditions

INVREQ

RESP2 values:

2 CICS is connected to WebSphere MQ.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

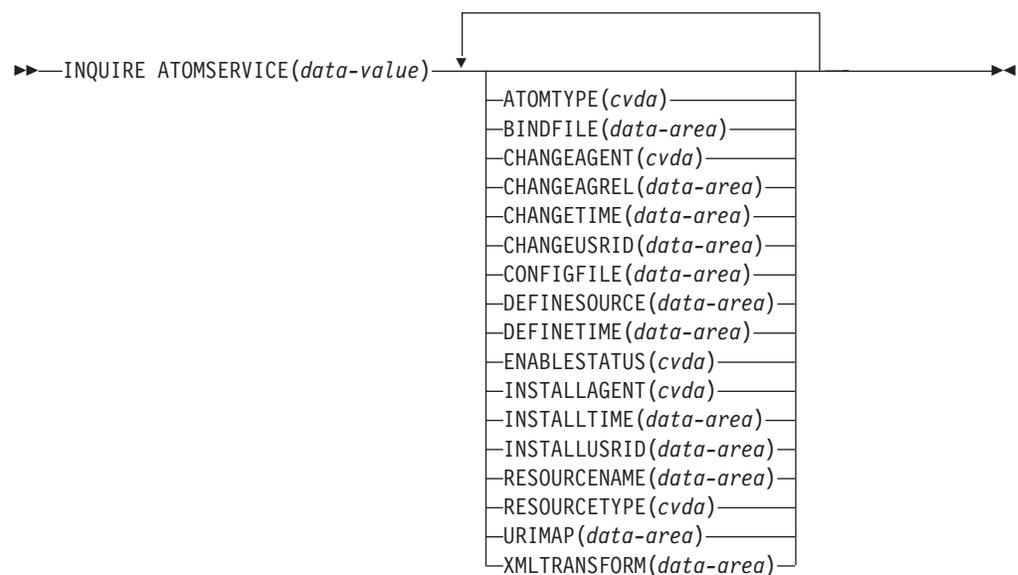
RESP2 values:

- 1 An MQCONN resource definition cannot be found.

INQUIRE ATOMSERVICE

Retrieve information about ATOMSERVICE resources in the local system.

INQUIRE ATOMSERVICE



Conditions: END, ILLOGIC, NOTAUTH, NOTFND

This command is threadsafe.

Browsing

You can browse through all the ATOMSERVICE definitions installed in the region, using the browse options, START, NEXT, and END, on **INQUIRE ATOMSERVICE** commands.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the

Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

ATOMSERVICE (*data-value*)

Specifies the name of the ATOMSERVICE resource about which you are inquiring. The name can be up to 8 characters in length.

ATOMTYPE (*cvda*)

Returns a CVDA value indicating the type of Atom document that is produced by this ATOMSERVICE definition. CVDA values are as follows:

CATEGORY

An Atom category document, which lists the categories of documents in a collection.

COLLECTION

An Atom collection document, which contains a group of entry documents that can be edited.

FEED An Atom feed document, which describes the metadata for a feed and contains entry documents that provide data for the feed.

SERVICE

An Atom service document, which provides information about the collections of entry documents that are available on the server and can be added to or edited.

UNKNOWN

The ATOMTYPE cannot be determined, this is probably because the associated configuration file cannot be read.

BINDFILE (*data-area*)

Returns a 255-character data area containing the fully qualified (absolute) or relative name of the XML binding specified in this ATOMSERVICE definition. The XML binding is stored in z/OS UNIX System Services, and it specifies the data structures used by the CICS resource that supplies the data for the Atom document that is returned for this resource definition. Service and category documents do not use an XML binding, so, if ATOMTYPE is SERVICE or CATEGORY, BINDFILE returns blanks.

CHANGEAGENT (*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL (*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see **FORMATTIME** in CICS Application Programming.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

CONFIGFILE (*data-area*)

Returns a 255-character data area containing the fully qualified (absolute) or relative name of the Atom configuration file specified in this **ATOMSERVICE** definition. The Atom configuration file is stored in z/OS UNIX System Services, and it contains XML that specifies metadata and content for the Atom document that is returned for this resource definition.

DEFINESOURCE (*data-area*)

Returns the 8-character source of the resource definition. The **DEFINESOURCE** value depends on the **CHANGEAGENT** value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

ENABLESTATUS (*cvda*)

Returns a CVDA value indicating the status of this **ATOMSERVICE** definition. CVDA values are as follows:

ENABLED

The **ATOMSERVICE** definition is enabled.

DISABLED

The **ATOMSERVICE** definition is disabled. An **ATOMSERVICE** definition with this status can be discarded.

INSTALLAGENT (*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

BUNDLE

The resource was installed by a bundle deployment.

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID (*data-area*)

Returns the 8-character user ID that installed the resource.

RESOURCENAME(*data-area*)

Returns the 16-character name of the CICS resource that provides the data for this Atom feed or collection. This option does not apply for an Atom service or category document.

RESOURCETYPE(*cvda*)

Returns a CVDA value indicating the type of CICS resource that provides the data for this Atom feed or collection. This option does not apply for an Atom service or category document. CVDA values are as follows:

FILE A CICS file.

PROGRAM

A service routine, which is a CICS application program written to supply content for Atom entries.

TSQUEUE

A temporary storage queue.

NOTAPPLIC

If the value of ATOMTYPE is SERVICE or CATEGORY, the resource type is not applicable.

URIMAP(*data-area*)

Returns the 8-character URIMAP name that indicates the URI associated with this ATOMSERVICE definition. If there is no auto-generated URIMAP associated with this ATOMSERVICE definition, this field is empty.

XMLTRANSFORM(*data-area*)

Returns the 32-character name of the XMLTRANSFORM resource associated with the ATOMSERVICE definition. If the value of ATOMTYPE is SERVICE or CATEGORY, this field is empty.

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.

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

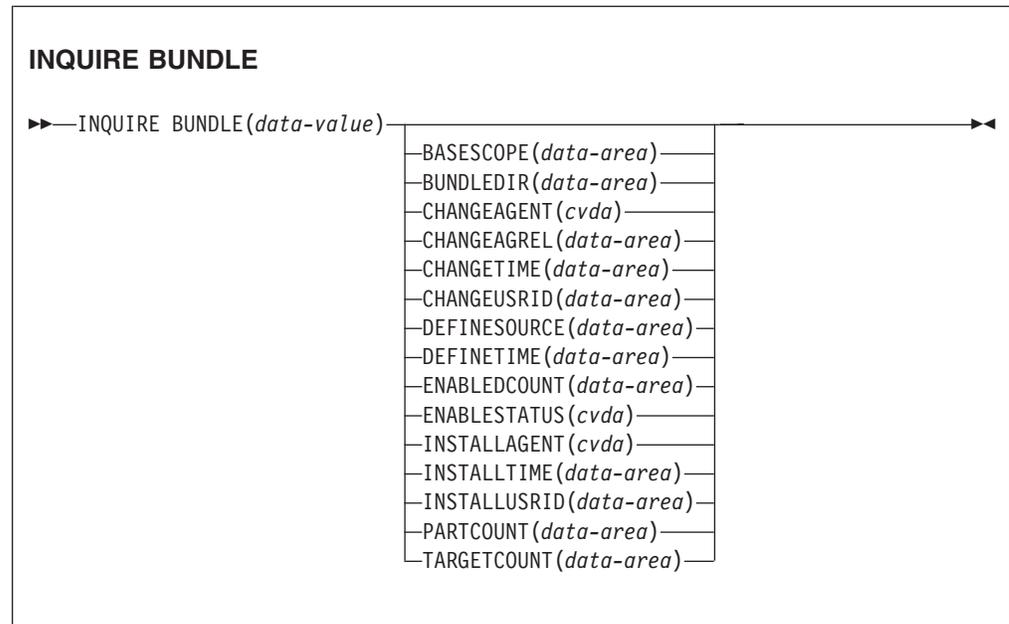
NOTFND

RESP2 values:

3 The ATOMSERVICE cannot be found.

INQUIRE BUNDLE

Retrieve information about an installed BUNDLE.



Conditions: END, ILLOGIC, INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

Use the **INQUIRE BUNDLE** command to retrieve information about an installed BUNDLE resource. The contents of a BUNDLE resource are defined in a manifest that can contain imports, exports, and definitions. The definitions section of the manifest describe the resources that CICS dynamically creates for you when you install the BUNDLE resource. Use this command to find out the location of the bundle on z/OS UNIX, the number of imports, exports, and definitions that are listed in the manifest, and how many of those definitions are currently enabled in the CICS region.

For more detailed information about each import, export, and definition, use the **INQUIRE BUNDLEPART** command.

Browsing

You can browse through all the BUNDLE resources that are installed in your region by using the browse options, START, NEXT, and END, on **INQUIRE BUNDLE** commands.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources

in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

BASESCOPE(*data-area*)

Returns the 1 - 255 character string that defines the root namespace for the contents of the bundle.

BUNDLE(*data-value*)

Specifies the name of the BUNDLE resource about which you are inquiring. The name can be up to 8 characters in length.

BUNDLEDIR(*data-area*)

Returns the 1 - 255 character fully qualified name of the root directory for the bundle on z/OS UNIX.

CHANGEAGENT(*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATC

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL(*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see **FORMATTIME** in CICS Application Programming.

CHANGEUSRID(*data-area*)

Returns the 8-character user ID that ran the change agent.

DEFINESOURCE(*data-area*)

Returns the 8-character source of the resource definition. The **DEFINESOURCE** value depends on the **CHANGEAGENT** value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

ENABLEDCOUNT (*data-area*)

Returns the current number of resources that were dynamically created by the bundle and are enabled in the CICS region.

ENABLESTATUS (*cvda*)

Returns the status of the BUNDLE:

ENABLED

The BUNDLE is ready for use.

DISABLED

The BUNDLE is not available.

ENABLING

The BUNDLE is being initialized. It is creating and enabling the resources that are defined in the bundle manifest file.

DISABLING

The BUNDLE is quiescing before entering DISABLED state. It disables any resources that it enabled.

DISCARDING

A DISCARD command has been issued for the BUNDLE. The BUNDLE is quiescing before being discarded. The bundle discards any resources that it disabled.

INSTALLAGENT (*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID (*data-area*)

Returns the 8-character user ID that installed the resource.

PARTCOUNT (*data-area*)

Returns the total number of imports, exports, and definition statements that are defined in the bundle manifest.

TARGETCOUNT (*data-area*)

Returns the total number of dynamically created resources in the bundle. CICS automatically enables the BUNDLE resource when all of the dynamically created resources are in an enabled state.

Conditions**END**

RESP2 values:

- 2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

- 1 A START command has been issued when a browse of BUNDLE resources is already in progress, or a NEXT or an END command has been issued when a browse of BUNDLE resources is not in progress.

INVREQ

RESP2 values:

- 7 CICS failed to link to the registered bundle callback program.

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 particular resource in the way required by this command.

NOTFND

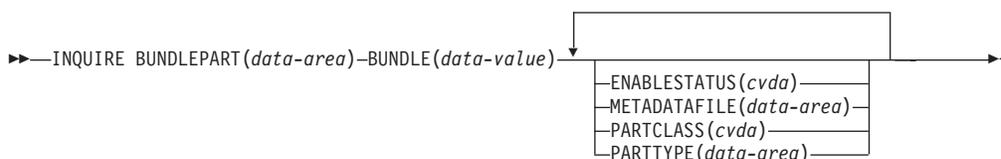
RESP2 values:

- 3 The BUNDLE cannot be found.

INQUIRE BUNDLEPART

Retrieve information about the resources that are contained in an installed BUNDLE resource.

INQUIRE BUNDLEPART



Conditions: END, ILLOGIC, INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

Use the **INQUIRE BUNDLEPART** command to return information about the resources that are contained in an installed BUNDLE resource. You can use the **INQUIRE BUNDLEPART** command only in browse mode.

Browsing

You can browse through all the resources that are installed in your region by a BUNDLE resource by using the browse options (START, NEXT, and END) on the **INQUIRE BUNDLEPART** command.

Options

BUNDLE(*data-value*)

Specify the 1 - 8 character name of the BUNDLE resource that you want to browse when using the START option.

BUNDLEPART(*data-area*)

Returns the 1 - 255 character name of a resource that is contained in the bundle.

ENABLESTATUS(*cvda*)

Returns the status of the resource in the bundle:

ENABLED

The resource is ready for use.

DISABLED

The resource is not available.

ENABLING

The resource is being created as part of the BUNDLE resource installation.

DISABLING

The resource is being disabled.

DISCARDING

A DISCARD command has been issued for the BUNDLE resource. The resource in the bundle is disabled and is being discarded.

UNUSABLE

The resource is in an unusable state.

METADATAFILE(*data-area*)

Returns the 1 - 255 character name of the file on z/OS UNIX that describes the resource, as defined in the manifest. The value is an absolute path from the root of the bundle directory.

PARTCLASS(*cvda*)

Returns the class of the resource that is defined in the manifest. The following values are valid:

DEFINITION

The resource is defined as a definition in the manifest.

EXPORT

The resource is defined as an export in the manifest. The resource is available to other services installed in the CICS region.

IMPORT

The resource is defined as an import in the manifest. The resource is required by the application in the CICS region.

PARTTYPE(*data-area*)

Returns the 1 - 255 character resource type as a URI. For example, the XMLTRANSFORM resource has the URI `http://www.ibm.com/xmlns/prod/cics/bundle/XMLTRANSFORM`.

Conditions

END

RESP2 values:

2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

- 1 A START command has been issued when a browse of BUNDLE resources is already in progress, or a NEXT or an END command has been issued when a browse of BUNDLE resources is not in progress.

INVREQ

RESP2 values:

- 8 A BUNDLE name must be specified on **START BUNDLEPART**.

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 the BUNDLE resource in the way required by this command.

NOTFND

RESP2 values:

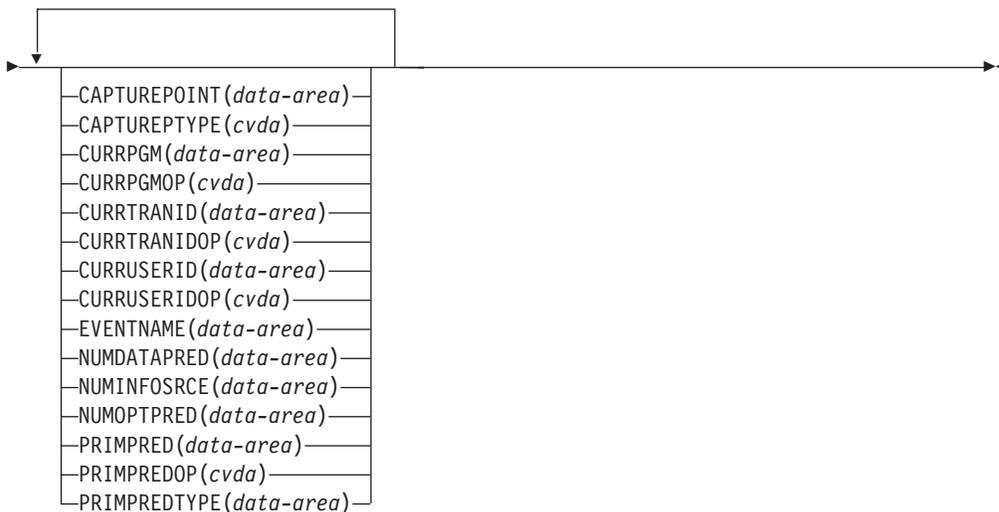
- 3 The BUNDLE cannot be found.

INQUIRE CAPTURESPEC

Retrieve information about a capture specification.

INQUIRE CAPTURESPEC

►►—INQUIRE CAPTURESPEC(*data-area*)—EVENTBINDING(*data-value*)—►►



Conditions: END, ILLOGIC, INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

The INQUIRE CAPTURESPEC command returns the attributes associated with a capture specification.

Browsing

You can browse through all the deployed capture specifications installed in the specified event binding using the browse options (START, NEXT, and END) on INQUIRE CAPTURESPEC commands.

Options

CAPTUREPOINT(*data-area*)

Specifies a 25-character data area to receive the capture point associated with the capture specification. Its contents will match one of the capture point entries in the event binding tooling. Capture point entries consisting of two words or more are separated with an underscore; for example, LINK_PROGRAM or PROGRAM_INITIATION.

CAPTUREEPTYPE(*cvda*)

Specifies a fullword binary data area to receive a CVDA value identifying the type of capture point. Possible CVDA values are as follows:

PRECOMMAND

Capture point is at the start of a CICS API command.

POSTCOMMAND

Capture point is on completion of a CICS API command.

PROGRAMINIT

Capture point is at program initiation.

SYSTEM

Capture point is a system event.

CAPTURESPEC(*data-area*)

On the non-browse form of this command, specifies the name (1-32 characters) of the capture specification. On the browse form of this command, specifies a 32-character data area to receive the name of the capture specification.

CURRPGM(*data-area*)

Specifies an 8-character data area to receive the value specified by the application context predicate for the current program name. Blanks are returned if no application context predicate for the current program name is defined for this capture specification.

CURRPGMOP(*cvda*)

Returns a CVDA value that defines the operator that is used, together with the value in the CURRPGM option, to evaluate the application context predicate on the current program name. Possible CVDA values are as follows:

ALLVALUES

The predicate always evaluates true; that is, there is no filtering based on the name of the current program.

DOESNOTEQUAL

The predicate evaluates true when the name of the current program is not equal to the value of the CURRPGM option.

DOESNOTSTART

The predicate evaluates true when the name of the current program does not start with the value of the CURRPGM option.

EQUALS

The predicate evaluates true when the name of the current program is equal to the value of the CURRPGM option.

GREATERTHAN

The predicate evaluates true when the name of the current program is greater than the value of the CURRPGM option.

ISNOTGREATER

The predicate evaluates true when the name of the current program is equal to or less than the value of the CURRPGM option.

ISNOTLESS

The predicate evaluates true when the name of the current program is equal to or greater than the value of the CURRPGM option.

LESSTHAN

The predicate evaluates true when the name of the current program is less than the value of the CURRPGM option.

STARTSWITH

The predicate evaluates true when the name of the current program starts with the value of the CURRPGM option.

CURRTRANID (*data-area*)

Specifies a 4-character data area to receive the value specified by the application context predicate for the current transaction name.

CURRTRANIDOP (*cvda*)

Returns a CVDA value that defines the operator that is used, together with the value in the CURRTRANID option, to evaluate the application context predicate on the current transaction name. Possible CVDA values are as follows:

ALLVALUES

The predicate always evaluates true; that is, there is no filtering based on the name of the current transaction.

DOESNOTEQUAL

The predicate evaluates true when the name of the transaction that is running is not equal to the value of the CURRTRANID option.

DOESNOTSTART

The predicate evaluates true when the name of the transaction that is running does not start with the value of the CURRTRANID option.

EQUALS

The predicate evaluates true when the name of the current transaction is equal to the value of the CURRTRANID option.

GREATERTHAN

The predicate evaluates true when the name of the current transaction is greater (that is, higher in the collating sequence of possible transaction IDs) than the value of the CURRTRANID option.

ISNOTGREATER

The predicate evaluates true when the name of the current transaction is equal to or less (that is, lower in the collating sequence of possible transaction IDs) than the value of the CURRTRANID option.

ISNOTLESS

The predicate evaluates true when the name of the current transaction is equal to or greater (that is, higher in the collating sequence of possible transaction IDs) than the value of the CURRTRANID option.

LESSTHAN

The predicate evaluates true when the name of the current transaction is less (that is, lower in the collating sequence of possible transaction IDs) than the value of the CURRTRANID option.

STARTSWITH

The predicate evaluates true when the name of the current transaction starts with the value of the CURRTRANID option.

CURRUSERID (*data-area*)

Specifies an 8-character data area to receive the value specified by the application context predicate for the user ID that is associated with the current transaction.

CURRUSERIDOP (*cvda*)

Returns a CVDA value that defines the operator that is used, together with the value in the CURRUSERID option, to evaluate the application context predicate on the user ID. Possible CVDA values are as follows:

ALLVALUES

The predicate always evaluates true; that is, there is no filtering based on the user ID.

DOESNOTEQUAL

The predicate evaluates true when the user ID of the current user is not equal to the value of the CURRUSERID option.

DOESNOTSTART

The predicate evaluates true when the user ID of the current user does not start with the value of the CURRUSERID option.

EQUALS

The predicate evaluates true when the user ID of the current user is equal to the value of the CURRUSERID option.

GREATERTHAN

The predicate evaluates true when the user ID of the current user is greater (that is, higher in the collating sequence of possible user IDs) than the value of the CURRUSERID option.

ISNOTGREATER

The predicate evaluates true when the user ID of the current user is equal to or less (that is, lower in the collating sequence of possible user IDs) than the value of the CURRUSERID option.

ISNOTLESS

The predicate evaluates true when the user ID of the current user is equal to or greater (that is, higher in the collating sequence of possible user IDs) than the value of the CURRUSERID option.

LESSTHAN

The predicate evaluates true when the user ID of the current user is less (that is, lower in the collating sequence of possible user IDs) than the value of the CURRUSERID option.

STARTSWITH

The predicate evaluates true when the user ID of the current user starts with the value of the CURRUSERID option.

EVENTBINDING (*data-value*)

Specifies the name (1-32 characters) of the associated event binding.

EVENTNAME (*data-area*)

Specifies a 32-character data area to receive the associated business event name.

NUMDATAPRED (*data-area*)

Specifies a fullword binary field that is set to the number of application data predicates that are defined for this capture specification.

NUMINFOSRCE (*data-area*)

Specifies a fullword binary field that is set to the number of information sources that are defined for this capture specification.

NUMOPTPRED (*data-area*)

Specifies a fullword binary field that is set to the number of application command option or system event option predicates that are defined for this capture specification. The total number of predicates includes the primary predicate.

PRIMPRED (*data-area*)

Specifies a 32-character data area to receive the value of the primary predicate for this capture specification. The primary predicate for a capture specification is the predicate to specify with the EQUALS operator; it helps to avoid a performance impact as more capture specifications are added for a particular capture point. Blanks are returned if there is no named primary predicate defined for this capture point.

PRIMPREDOP (*cvda*)

Returns a CVDA value that defines the operator that is used, together with the value in the PRIMPRED option, to evaluate the primary predicate. Possible CVDA values are as follows:

ALLVALUES

The predicate always evaluates true; that is, there is no filtering based on the name of the resource for the command.

DOESNOTEQUAL

The predicate evaluates true when the resource that is specified by the command is not equal to the value of the PRIMPRED option.

DOESNOTSTART

The predicate evaluates true when the resource that is specified by the command does not start with the value of the PRIMPRED option.

EQUALS

The predicate evaluates true when the resource that is specified by the command is equal to the value of the PRIMPRED option.

GREATERTHAN

The predicate evaluates true when the resource that is specified by the command is greater than the value of the PRIMPRED option.

ISNOTGREATER

The predicate evaluates true when the resource that is specified by the command is equal to or less than the value of the PRIMPRED option.

ISNOTLESS

The predicate evaluates true when the resource specified by the command is equal to or greater than the value of the PRIMPREP option.

LESSTHAN

The predicate evaluates true when the resource that is specified by the command is less than the value of the PRIMPREP option.

STARTSWITH

The predicate evaluates true when the resource that is specified by the command starts with the value of the PRIMPREP option.

PRIMPREPTYPE(*cvda*)

Returns a CVDA value that identifies the type of the primary predicate for this capture specification. Possible CVDA values are as follows:

CONTAINER

The primary predicate is a container.

CURRENTPGM

The primary predicate is the current program name.

EVENT

The primary predicate is a CICS event.

FILE The primary predicate is a CICS file.

MAP The primary predicate is a CICS basic mapping support (BMS) map.

NONE

The capture specification has no primary predicate.

PROGRAM

The primary predicate is a CICS program name.

SERVICE

The primary predicate is a CICS service or a WEBSERVICE resource.

TDQUEUE

The primary predicate is a CICS transient data queue.

TRANCLASS

The primary predicate is a CICS transaction class name.

TRANSACTION

The primary predicate is a CICS transaction identifier.

TSQUEUE

The primary predicate is a CICS temporary storage queue.

Conditions**END**

RESP2 values:

2 There are no more resource definitions of this type.

8 The event binding has been deleted so browse has been terminated early.

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.

INVREQ

- 4 An EVENTBINDING name has not been specified for the START CAPTURESPEC browse.

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 read the event binding.

NOTFND

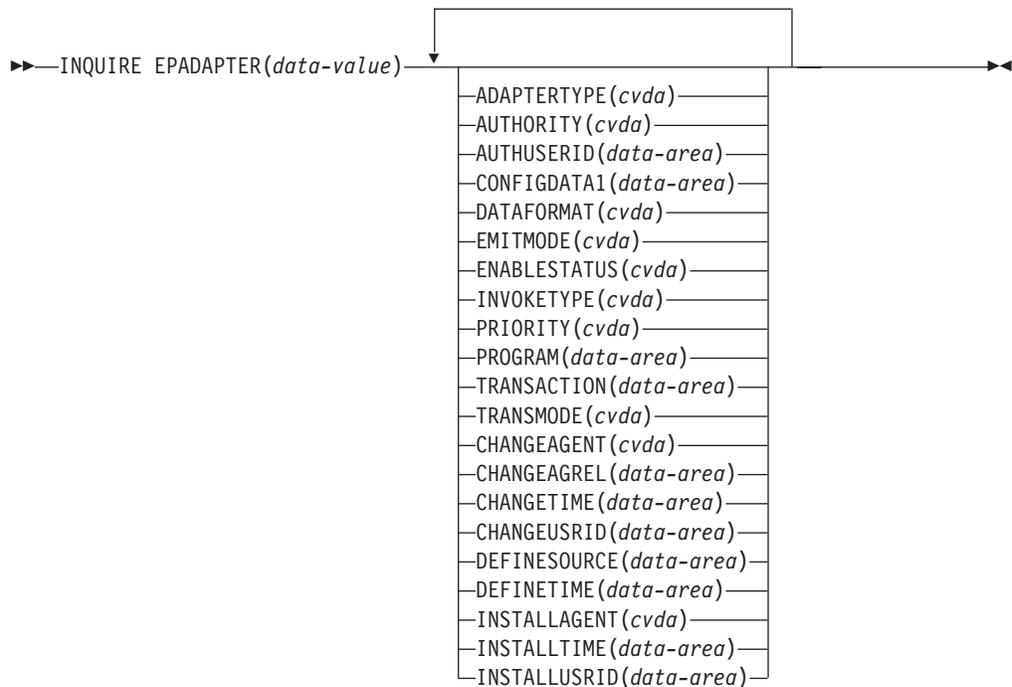
RESP2 values:

- 2 The specified capture specification cannot be found.
- 3 The specified event binding cannot be found.

INQUIRE EPADAPTER

Retrieve information about a specified event processing adapter.

INQUIRE EPADAPTER



Conditions: END, ILLOGIC, NOTAUTH, NOTFND

This command is threadsafe.

Description

The INQUIRE EPADAPTER command returns information about a specified event processing adapter.

Browsing

You can browse through all the event processing adapters that are installed in your region by using the browse options (START, NEXT, and END) on **INQUIRE EPADAPTER** commands.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

ADAPTERTYPE(*cvda*)

Returns a CVDA value indicating the type of this EP adapter. The CVDA values are as follows:

CUSTOM

A user-written EP adapter that emits events in any format that you require to any destination.

HTTP The HTTP EP adapter that emits events to an HTTP server for consumption by products such as WebSphere Business Events and IBM Business Monitor.

TRANSTART

The Transaction Start EP adapter that emits events to a named CICS transaction.

TSQUEUE

The TSQ EP adapter that emits events to a named CICS TS queue.

WMQ The WebSphere MQ EP adapter that emits events to WebSphere MQ for consumption by products such as WebSphere Business Events and IBM Business Monitor.

AUTHORITY(*cvda*)

Returns a CVDA value indicating the authority of the EP adapter. The CVDA values are as follows:

CONTEXT

The EP adapter runs using the user ID of the task that caused the event to be captured. This is always the case when EMITMODE is SYNCHRONOUS or when **Use Context User ID** is specified in the advanced section of the Adapter tab for the EP adapter.

REGION

The EP adapter runs using the CICS region user ID.

USERID

The EP adapter is attached using the identifier specified in the EP adapter user ID and returned in the AUTHUSERID attribute.

AUTHUSERID (*data-area*)

Returns the 8-character identifier to be used to attach the EP adapter transaction. This attribute is only set if AUTHORITY is USERID.

CHANGEAGENT (*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL (*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see **FORMATTIME** in CICS Application Programming.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

CONFIGDATA1 (*data-area*)

A 64-character data area containing the primary configuration data item for the EP adapter. If the primary configuration data item is less than 64 bytes, the field is padded with blanks. The data item returned is dependent upon **ADAPTERTYPE**, as follows:

CUSTOM

Returns the first 64 bytes of custom EP adapter configuration data.

HTTP Returns the 8-character name of the URIMAP definition to be used by an HTTP EP adapter to locate the HTTP server.

TRANSTART

Returns the 4-character name of the event consumer transaction that is started by a Transaction Start EP adapter.

TSQUEUE

Returns the 16-character name of the temporary storage queue for the event emitted by a TSQ EP adapter.

WMQ Returns the 48-character name of the WebSphere MQ queue for event messages emitted by this WebSphere MQ EP adapter. This data is in the code page defined by the **LOCALCCSID** system initialization parameter.

DATAFORMAT (*cvda*)

Returns a CVDA value indicating the format of events emitted by this EP adapter. The CVDA values are as follows:

- USER** The format is user-defined.
- CBER** Common Base Event REST format for the IBM Business Monitor REST HTTP server.
- CBE** Common Base Event format for consumption by products such as IBM Business Monitor.
- CCE** CICS Container Event format.
- CFE** CICS Flattened Event format.
- WBE** WebSphere Business Events XML format.

DEFINESOURCE(*data-area*)

Returns the 8-character source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

EMITMODE(*cvda*)

Returns a CVDA value indicating the event emission attribute of this EP adapter. The CVDA values are as follows:

ASYNCHRONOUS

Event emission and the capturing transaction are asynchronous. Failure to emit an event has no effect on the capturing transaction.

SYNCHRONOUS

Event emission is synchronous with the capturing transaction. The unit of work for the capturing transaction does not complete successfully when the event is not emitted.

ENABLESTATUS(*cvda*)

Returns a CVDA value indicating the status of this EP adapter. CVDA values are as follows:

ENABLED

The EP adapter is enabled.

DISABLED

The EP adapter is disabled.

EPADAPTER(*data-area*)

Specifies the name (1 - 32 characters) of an EP adapter. You must specify this option to retrieve details of a particular EP adapter by name. On the browse form of this command, you must provide a 32-character data area to receive the name of the EP adapter.

INSTALLAGENT(*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. Only one value is possible:

BUNDLE

The resource was installed by a bundle deployment.

INSTALLTIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID(*data-area*)

Returns the 8-character user ID that installed the resource.

INVOKETYPE(*cvda*)

Returns a CVDA value indicating how the EP adapter is started. The CVDA values are as follows:

ATTACH

The EP adapter is attached as a separate task.

LINK The EP adapter program is linked to.

PRIORITY(*cvda*)

Returns a CVDA value indicating the dispatching priority of event emission for this EP adapter. This attribute is ignored when EMITMODE is SYNCHRONOUS. The CVDA values are as follows:

HIGH Events emitted for this EP adapter are high priority.

NORMAL

Events emitted for this EP adapter are normal priority.

PROGRAM(*data-value*)

Returns the 8-character name of the EP adapter program. If the ADAPTERTYPE is CUSTOM, this attribute is only applicable if INVOKETYPE is LINK.

TRANSACTION(*data-value*)

Returns the 4-character name of the transaction definition that is used if the EP adapter transaction is attached. The TRANSACTION attribute is only applicable if INVOKETYPE is ATTACH.

TRANSMODE(*cvda*)

Returns a CVDA value indicating the event transactionality attribute of this EP adapter. CVDA values are as follows:

NONTRANS

Events are not transactional. Events can be emitted regardless of whether the unit of work for the capturing transaction completes successfully.

TRANS

Events are transactional. Events can be emitted only when the unit of work for the capturing transaction completes successfully.

Conditions**END**

RESP2 values:

2 There are no more EP adapters to browse.

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.

101 The user associated with the issuing task is not authorized to read the EP adapter.

NOTFND

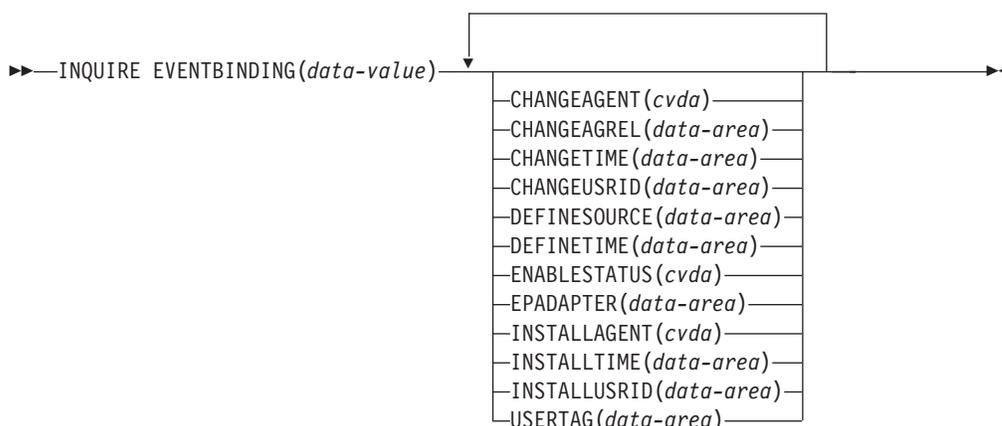
RESP2 values:

3 The specified EP adapter cannot be found.

INQUIRE EVENTBINDING

Retrieve information about a specified event binding.

INQUIRE EVENTBINDING



Conditions: END, ILLOGIC, NOTAUTH, NOTFND

This command is threadsafe.

Description

The `INQUIRE EVENTBINDING` command returns information about a specific event binding.

Browsing

You can browse through all the event bindings that are installed in your region by using the browse options (START, NEXT, and END) on `INQUIRE EVENTBINDING` commands.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

CHANGEAGENT (*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL (*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see **FORMATTIME** in CICS Application Programming.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

DEFINESOURCE (*data-area*)

Returns the 8-character source of the resource definition. The **DEFINESOURCE** value depends on the **CHANGEAGENT** value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

ENABLESTATUS (*cvda*)

Returns a CVDA value indicating the status of this event binding. CVDA values are as follows:

ENABLED

The event binding is enabled.

DISABLED

The event binding is disabled.

EPADAPTER (*data-area*)

Returns the 32-character name of the EP adapter used by this event binding.

EVENTBINDING (*data-value*)

Specifies the 32-character name of an event binding.

INSTALLAGENT (*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. Only one value is possible:

BUNDLE

The resource was installed by a bundle deployment.

INSTALLTIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID(*data-area*)

Returns the 8-character user ID that installed the resource.

USERTAG (*data-area*)

Returns the 8-character user tag of the event binding.

Conditions

END

RESP2 values:

- 2 There are no more event bindings to browse.

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.
- 101 The user associated with the issuing task is not authorized to read the event binding.

NOTFND

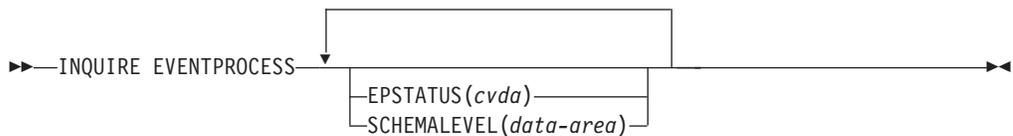
RESP2 values:

- 3 The specified event binding cannot be found.

INQUIRE EVENTPROCESS

Retrieve the status of event processing.

INQUIRE EVENTPROCESS



Conditions: NOTAUTH

This command is threadsafe.

Description

The INQUIRE EVENTPROCESS command returns the status of event processing.

Options

EPSTATUS(*cvda*)

Returns a CVDA value identifying the current status of event processing.

STARTED

CICS is processing events.

DRAINING

CICS event processing is draining.

STOPPED

CICS is not processing events.

SCHEMALEVEL(*data-area*)

Returns a 4-character value (*vrrr*) indicating the highest version and release of event binding schema that is supported by CICS, where *vv* is the version and *rr* is the release; for example, 0201 indicates version 2 release 1 of the event binding schema.

Conditions

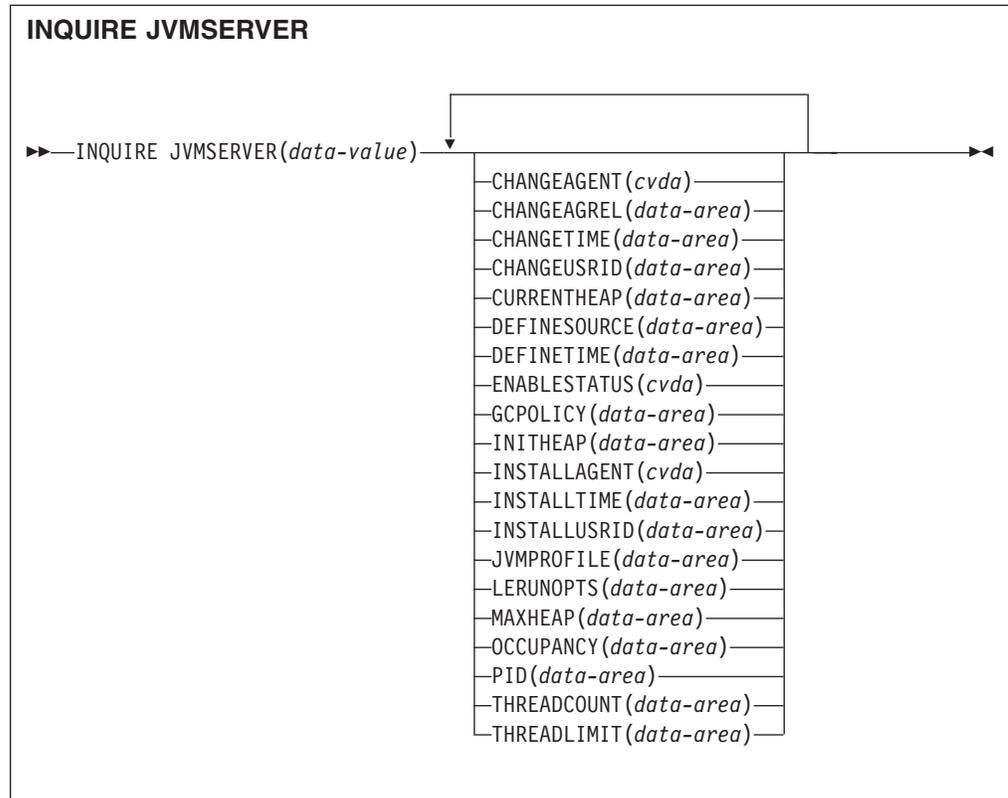
NOTAUTH

RESP2 values are:

100 The user associated with the issuing task is not authorized to use this command.

INQUIRE JVMSERVER

Retrieve information about the JVM server runtime environment in the CICS region.



Conditions: END, ILLOGIC, NOTAUTH, NOTFND

This command is threadsafe.

Description

The **INQUIRE JVMSERVER** command retrieves information about one or more JVM servers that are running in the CICS region.

Browsing

You can browse through all the JVMSERVER definitions installed in the region, using the browse options (START, NEXT, and END) on INQUIRE JVMSERVER commands.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

CHANGEAGENT (*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL (*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see **FORMATTIME** in CICS Application Programming.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

CURRENTHEAP (*data-area*)

Returns a doubleword binary value indicating the current size of the heap in bytes that is allocated to the JVM server.

DEFINESOURCE (*data-area*)

Returns the 8-character source of the resource definition. The **DEFINESOURCE** value depends on the **CHANGEAGENT** value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

ENABLESTATUS (*cvda*)

Returns a CVDA value indicating the overall status of the JVM server. The CVDA values are as follows:

ENABLED

The JVM server has started and is enabled for use.

ENABLING

The JVM server is starting.

DISABLED

The JVM server is stopped and any new requests cannot be processed.

DISABLING

The JVM server is stopping. Threads can still be running if they were started before the JVM server was stopped.

DISCARDING

The JVMSERVER resource is being discarded.

GCPOLICY(*data-area*)

Returns a 32-character value indicating the garbage collection policy that is being used by the JVM server.

INITHEAP(*data-area*)

Returns a doubleword binary value that indicates the initial size of the heap in bytes that is allocated to the JVM server. This value is set by the **-Xms** option in the JVM profile.

INSTALLAGENT(*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID(*data-area*)

Returns the 8-character user ID that installed the resource.

JVMPROFILE(*data-area*)

Returns the 1-8 character profile name of the JVM server.

JVMSERVER(*data-value*)

Specifies the name of the JVMSERVER resource about which you are inquiring. The name can be up to 8 characters in length.

LERUNOPTS(*data-area*)

Returns the 1-8 character name of the program that defines the runtime options for the Language Environment enclave.

MAXHEAP(*data-area*)

Returns a doubleword binary value that indicates the maximum size of the heap in bytes that is allocated to the JVM server. This value is set by the **-Xmx** option in the JVM profile.

OCCUPANCY(*data-area*)

Returns a doubleword binary value that indicates the size of the heap in bytes after the last garbage collection ran in the JVM server.

PID(*data-area*)

Returns a fullword value that indicates the process ID (PID) of the JVM.

THREADCOUNT(*data-area*)

Returns a fullword binary value giving the number of threads that are currently running inside the JVM server.

THREADLIMIT(*data-area*)

Returns a fullword binary value giving the number of threads that are allowed in the Language Environment enclave for the JVM server. Each thread runs under a T8 TCB.

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

NOTFND

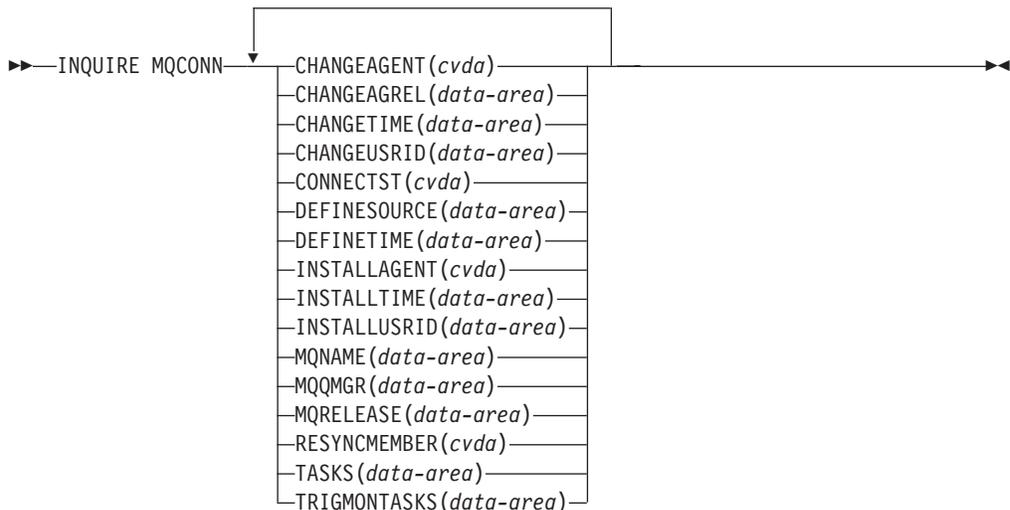
RESP2 values:

- 3 The JVMSERVER resource cannot be found.

INQUIRE MQCONN

Inquire on the attributes and status of the connection between CICS and WebSphere MQ.

INQUIRE MQCONN



Conditions: NOTAUTH, NOTFND

This command is threadsafe.

Description

The **INQUIRE MQCONN** command returns information about attributes of the currently installed MQCONN resource definition, which defines the connection to WebSphere MQ, and about the status of the connection.

Because only one MQCONN resource definition can be installed at a time, the name of the MQCONN resource definition is not required on input.

This command does not inquire on the INITQNAME attribute of the MQCONN resource definition, which specifies the name of the initiation queue. Use the **INQUIRE MQINI** command to inquire on the initiation queue name.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

CHANGEAGENT (*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL (*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see FORMATTIME in CICS Application Programming.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

CONNECTST (*cvda*)

Returns the status of the CICS-MQ connection. CVDA values are as follows:

CONNECTED

CICS is connected to WebSphere MQ.

NOTCONNECTED

CICS is not connected to WebSphere MQ.

CONNECTING

CICS is currently attempting to connect to WebSphere MQ.

DISCONNING

CICS is currently disconnecting from WebSphere MQ.

DEFINESOURCE (*data-area*)

Returns the 8-character source of the resource definition. The **DEFINESOURCE** value depends on the **CHANGEAGENT** value. For more information, see *Summary of the resource signature field values in the Resource Definition Guide*.

DEFINETIME (*data-area*)

Returns an **ABSTIME** value that represents the time stamp when the resource definition was created.

INSTALLAGENT (*cvda*)

Returns a **CVDA** value that identifies the agent that installed the resource. The possible values are as follows:

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a **CEDA** transaction or the programmable interface to **DFHEDAP**.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME (*data-area*)

Returns an **ABSTIME** value that represents the time stamp when the resource was installed.

INSTALLUSRID (*data-area*)

Returns the 8-character user ID that installed the resource.

MQNAME (*data-area*)

Returns the 1 - 4 character name of the WebSphere MQ queue manager or queue-sharing group that you specified in the **MQCONN** resource definition (or using a **SET MQCONN** command) for the CICS region.

MQMGR (*data-area*)

Returns the 1 - 4 character name of the WebSphere MQ queue manager to which CICS is connected, or to which CICS is waiting to connect.

- If CICS is connected to WebSphere MQ, this field shows the name of the queue manager to which CICS is connected. If you specified a queue-sharing group in the **MQCONN** resource definition for the CICS region, the queue manager shown here is the one that was chosen from the group.
- If CICS is not connected to WebSphere MQ, this field usually contains blanks. However, if you specified a queue-sharing group in the **MQCONN** resource definition for the CICS region, and CICS is waiting to reconnect to a specific queue manager in the queue-sharing group because it is holding outstanding units of work for that queue manager, the name of the specific queue manager is shown, and the status of the connection is shown as

CONNECTING. For this situation to arise, the RESYNCMEMBER attribute in the MQCONN resource definition must specify resynchronization.

MQRELEASE (*data-area*)

If CICS is connected to WebSphere MQ, this option returns the 4-digit release number of WebSphere MQ; for example, 0600. When CICS is not connected to WebSphere MQ, MQRELEASE returns blanks.

RESYNCMEMBER (*cvda*)

This option applies only if you have specified a queue-sharing group for the CICS-MQ connection. It shows the strategy that CICS adopts if outstanding units of work are being held for the last queue manager to which CICS was connected from the queue-sharing group. Units of work that are shunted indoubt are not included in this process, because CICS itself cannot resolve those units of work at this time. Resynchronization for those UOWs occurs when CICS has resynchronized with its remote coordinator. CVDA values are as follows:

RESYNC

CICS connects to the same queue manager.

NORESYNC

CICS makes one attempt to connect to the same queue manager. If that attempt fails, CICS connects to any member of the queue-sharing group and issues a warning message about the outstanding units of work.

GROUPRESYNC

CICS connects to any member of the queue-sharing group. The queue manager is chosen by WebSphere MQ and it asks CICS to resolve indoubt units of work on behalf of all eligible queue managers in the queue-sharing group. This function is called *group unit of recovery*.

NOTAPPLIC

A queue-sharing group is not specified for the CICS-MQ connection.

TASKS (*data-area*)

Returns the current number of tasks that are using the CICS-MQ connection, including trigger monitor tasks, as a fullword binary value.

TRIGMONTASKS (*data-area*)

Returns the current number of trigger monitor tasks that are using the CICS-MQ connection, as a fullword binary value.

Conditions

NOTFND

RESP2 values:

1 The MQCONN resource definition cannot be found.

NOTAUTH

RESP2 values:

100 Command authorization failure.

INQUIRE MQINI

Inquire on the name of the default initiation queue to be used for the connection between CICS and WebSphere MQ.

INQUIRE MQINI

►►—INQUIRE MQINI—(*—data-value—*)—CHANGEAGENT(*cvda*)—CHANGEAGREL(*data-area*)—►
►—CHANGETIME(*data-area*)—CHANGEUSRID(*data-area*)—DEFINESOURCE(*data-area*)—►►
►—DEFINETIME(*data-area*)—INITQNAME(*data-area*)—INSTALLAGENT(*cvda*)—►►
►—INSTALLTIME(*data-area*)—INSTALLUSRID(*data-area*)—►►►►

Conditions: NOTAUTH, NOTFND

This command is threadsafe.

Description

The INQUIRE MQINI command inquires on the name of the default initiation queue used for the connection between CICS and WebSphere MQ.

The MQINI resource represents the default initiation queue. MQINI is an implicit resource that exists when you install an MQCONN resource definition in the CICS region with the INITQNAME attribute specified. (Only one MQCONN resource definition can be installed at a time.) The name of the MQINI resource is DFHMQINI. You must specify this resource name on the command.

If you want to change the MQINI resource definition, you must reinstall the MQCONN resource definition with an appropriate MQINI attribute.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

CHANGEAGENT(*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

AUTOINSTALL

The resource was autoinstalled as a result of specifying an initiation queue name on a CKQC START command, and the previously installed MQCONN definition did not specify a value for INITQNAME.

DYNAMIC

The resource was defined as a result of an MQCONN resource definition with INITQNAME specified.

CHANGEAGREL (*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see **FORMATTIME** in CICS Application Programming.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

DEFINESOURCE (*data-area*)

Returns the 8-character source of the resource definition. The **DEFINESOURCE** value depends on the **CHANGEAGENT** value. For more information, see **Summary of the resource signature field values in the Resource Definition Guide**.

DEFINETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

INITQNAME (*data-area*)

Returns the 1- to 48-character name of the default initiation queue that is represented by the **MQINI** resource.

INSTALLAGENT (*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

AUTOINSTALL

The resource was autoinstalled as a result of specifying an initiation queue name on a **CKQC START** command, and the previously installed **MQCONN** definition did not specify a value for **INITQNAME**.

DYNAMIC

The resource was installed as a result of the installation of an **MQCONN** with **INITQNAME** specified.

INSTALLTIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID (*data-area*)

Returns the 8-character user ID that installed the resource.

Conditions**NOTFND**

RESP2 values:

1 The **MQCONN** resource definition that implies the **MQINI** resource cannot be found.

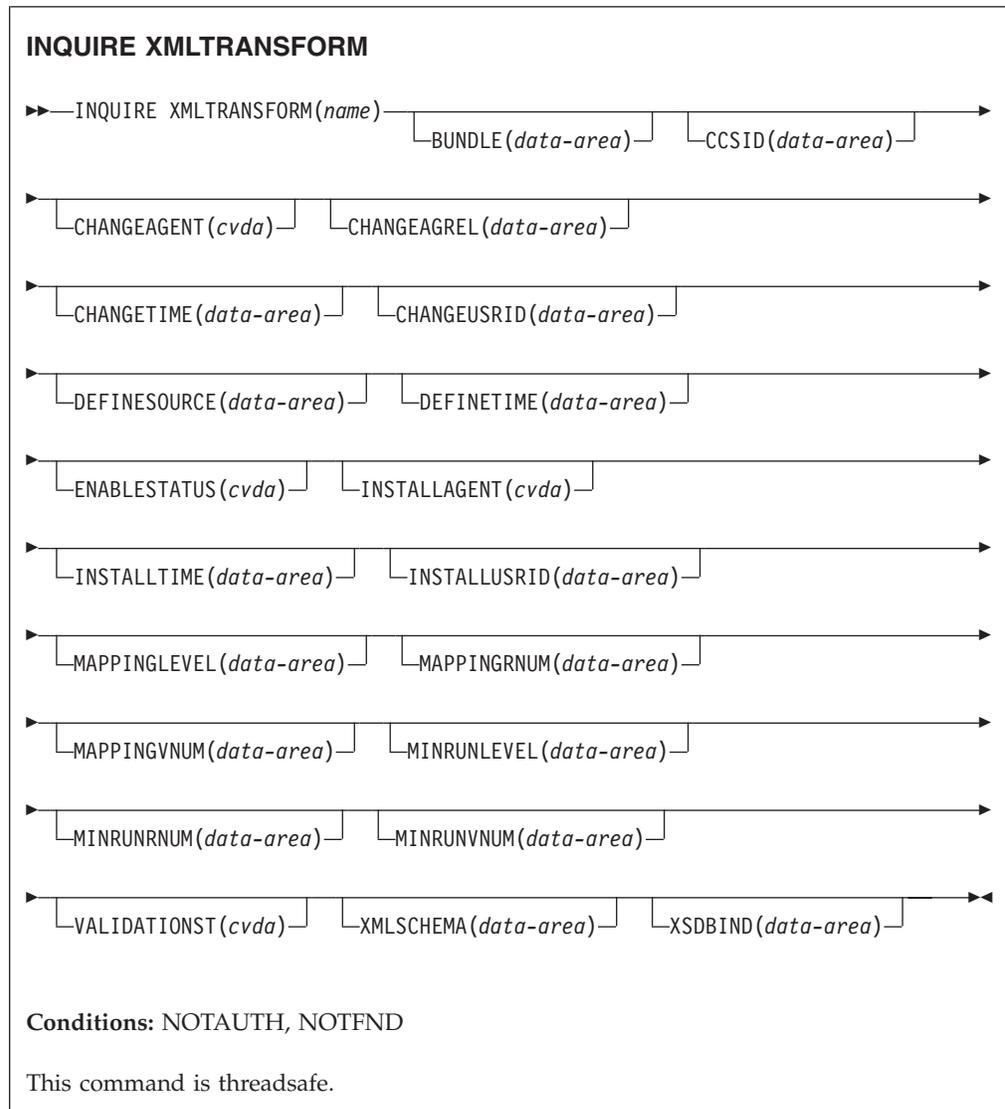
NOTAUTH

RESP2 values:

100 Command authorization failure.

INQUIRE XMLTRANSFORM

Use the **INQUIRE XMLTRANSFORM** command to retrieve information about an installed XMLTRANSFORM resource.



Description

Use the **INQUIRE XMLTRANSFORM** command to retrieve information about an installed XMLTRANSFORM resource. This information can include the state of the XMLTRANSFORM resource and details about the conditions under which the XMLTRANSFORM resource was installed, such as which mapping level was used.

Browsing

You can browse through all the XMLTRANSFORM resources installed in your system by using the browse options, START, NEXT, and END, on the **INQUIRE XMLTRANSFORM** command.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

BUNDLE(*data-area*)

Returns the 8-character name of the bundle from which the XMLTRANSFORM was installed.

CCSID(*data-area*)

Returns the coded character set identifier (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 XML assistant when the XML binding file is generated. The CCSID is a value of up to 8 characters. If CCSID value is not specified, CICS uses the default CCSID that is specified by the **LOCALCCSID** system initialization parameter.

CHANGEAGENT(*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

DYNAMIC

The resource was defined by an ATOMSERVICE resource.

CHANGEAGREL(*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see **FORMATTIME** in CICS Application Programming.

CHANGEUSRID(*data-area*)

Returns the 8-character user ID that ran the change agent.

DEFINESOURCE(*data-area*)

Returns the 8-character source of the resource definition. The **DEFINESOURCE**

value depends on the CHANGEAGENT value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

ENABLESTATUS (*cvda*)

Returns a CVDA indicating the state of the XMLTRANSFORM.

DISCARDING

A DISCARD command has been issued for the XMLTRANSFORM.

ENABLING

The XMLTRANSFORM is in the process of being enabled.

ENABLED

The XMLTRANSFORM is enabled and available for use.

DISABLING

The XMLTRANSFORM is in the process of being disabled. It is not available for further use, but inflight activity will be allowed to complete.

DISABLED

The XMLTRANSFORM is disabled and is not available for use.

INSTALLAGENT (*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

BUNDLE

The resource was installed by a bundle deployment.

DYNAMIC

The resource was installed by an ATOMSERVICE resource.

INSTALLTIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID (*data-area*)

Returns the 8-character user ID that installed the resource.

MAPPINGLEVEL (*data-area*)

Returns an 8-byte character string of the mapping level that was used when the XML binding file was produced. The value of the mapping level is 1.0, 1.1, 1.2, 2.0, 2.1, 2.2, or 3.0.

MAPPINGRNUM (*data-area*)

Returns a fullword binary value of the release number for the mapping level that was used when the XML binding file was produced. 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 was used when the XML binding file was produced. The value of the version number is 1, 2, or 3.

MINRUNLEVEL (*data-area*)

Returns an 8-byte character string of the minimum runtime level that is required to install the XMLTRANSFORM in CICS. The value of the runtime level is 3.0.

MINRUNRNUM(*data-area*)

Returns a fullword binary value for the release number for the minimum runtime level that is required to install the XMLTRANSFORM in CICS. The value of the release number is 0.

MINRUNVNUM(*data-area*)

Returns a fullword binary value for the version number for the minimum runtime level that is required to install the XMLTRANSFORM in CICS. The value of the version number is 3.

VALIDATIONST(*cvda*)

Indicates whether full validation is enabled for the XMLTRANSFORM resource. CVDA values are as follows:

VALIDATION

Full validation is enabled.

NOVALIDATION

Full validation is disabled.

Because validating an XML message against its schema incurs considerable processing overhead, typically you will specify VALIDATIONST(NOVALIDATION). If VALIDATIONST(NOVALIDATION) is specified, checking is performed to ensure that the message contains well-formed XML, but with no guarantee that the XML is valid.

Full validation ensures that the XML in the message is valid with respect to the XML schema; you might want to specify VALIDATIONST(VALIDATION) when you are developing an application.

XMLTRANSFORM(*data-value*)

Specifies the 1- to 32-character name of the XMLTRANSFORM about which you are inquiring.

XMLSCHEMA(*data-area*)

Returns the name of the associated XML schema file. The data area is 255 characters long. If the name is shorter than 255 characters, CICS pads the data area with trailing blanks.

XSDBIND(*data-area*)

Returns the name of the XML binding file. The data area is 255 characters long. If the name is shorter than 255 characters, CICS pads the data area with trailing blanks.

Conditions**NOTAUTH**

RESP2 values are:

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 particular resource in the way required by this command.

NOTFND

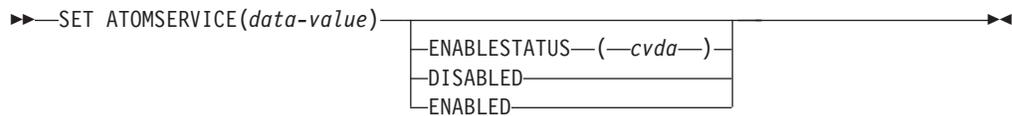
RESP2 values are:

3 The XMLTRANSFORM cannot be found.

SET ATOMSERVICE

Enables or disables an ATOMSERVICE definition.

SET ATOMSERVICE



Conditions: INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Options

ENABLESTATUS(*cvda*)

Sets the ATOMSERVICE definition to enabled or disabled status. CVDA values are:

DISABLED

The ATOMSERVICE definition cannot be accessed by applications. An ATOMSERVICE definition has to be disabled before it can be reinstalled or discarded. If you disable an ATOMSERVICE resource definition, CICS returns an HTTP response to the Web client with a 503 (Service Unavailable) status code.

ENABLED

The ATOMSERVICE definition can be accessed by applications.

Conditions

INVREQ

RESP2 values:

9 Invalid ENABLESTATUS value.

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 particular resource in the way required by this command.

NOTFND

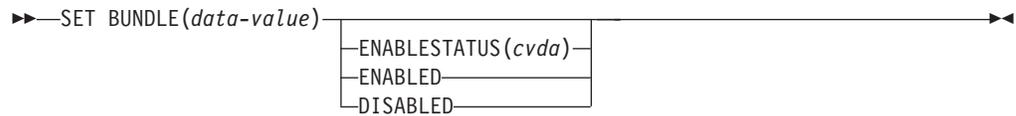
RESP2 values:

3 The ATOMSERVICE cannot be found.

SET BUNDLE

Change the status of an installed BUNDLE resource.

SET BUNDLE



Conditions: INVREQ, NOTAUTH, NOTFOUND

This command is threadsafe.

Description

Use the **SET BUNDLE** command to change the status of an installed BUNDLE resource. If you enable the BUNDLE resource, CICS tries to enable all of the resources that have been dynamically created by the bundle. CICS cannot enable the BUNDLE resource if one or more of these dynamic resources are in an unusable state.

If you disable the BUNDLE resource, CICS tries to disable all of the resources that have been dynamically created by the bundle.

Options

ENABLESTATUS(*cvda*)

Specifies the status of the BUNDLE:

ENABLED

The BUNDLE and its associated resources are available to CICS.

DISABLED

The BUNDLE and its associated resources are not available to CICS.

Conditions

INVREQ

RESP2 values:

- 6 An attempt was made to enable or disable a BUNDLE resource that is in an invalid state.
- 7 CICS failed to link to the registered bundle callback program.
- 9 BUNDLE cannot be enabled.

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 particular resource in the way required by this command.

NOTFND

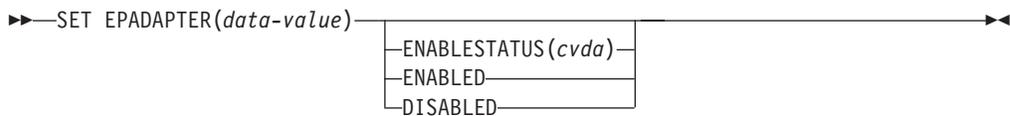
RESP2 values:

- 3 The BUNDLE was not found.

SET EPADAPTER

Set the status of a specified EP adapter to enabled or disabled.

SET EPADAPTER



Conditions: INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

Use the SET EPADAPTER command to change the status of a particular EP adapter. Changing the status of an EPADAPTER has no effect on the status of any related EVENTBINDINGS.

Options

EPADAPTER (*data-value*)

Specifies the 32-character name of an EP adapter.

ENABLESTATUS (*cvda*)

Specifies whether events are to be dispatched to this EP adapter. The CVDA values are as follows:

ENABLED

The event processing dispatcher can dispatch events to this EP adapter. The event processing dispatcher starts dispatching events to this EP adapter immediately.

DISABLED

The event processing dispatcher should stop dispatching events to this EP adapter. The event processing dispatcher stops dispatching events to this EP adapter immediately. Any events already dispatched to the EP adapter are emitted.

Conditions

INVREQ

RESP2 values are:

4 ENABLESTATUS has an invalid CVDA value.

NOTAUTH

RESP2 values are:

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 update the EP adapter.

NOTFND

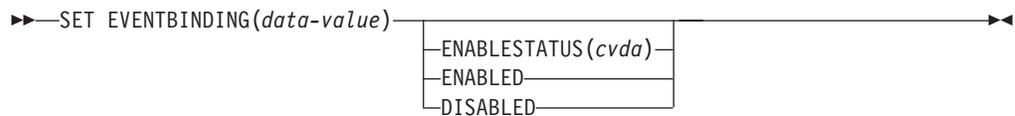
RESP2 values are:

3 The specified EP adapter cannot be found.

SET EVENTBINDING

Set the status of a specified event binding to enabled or disabled.

SET EVENTBINDING



Conditions: INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

Use the SET EVENTBINDING command to change the status of a particular event binding. If installation of the EVENTBINDING caused an EPADAPTER of the same name to also be installed, the state of the related EPADAPTER, if still available, is unaffected.

Options

EVENTBINDING (*data-value*)

Specifies the 32-character name of an event binding.

ENABLESTATUS (*cvda*)

Specifies whether events matching capture specifications in this event binding are captured and emitted. CVDA values are as follows:

ENABLED

The event binding is enabled. Capture of events matching capture specifications in this event binding starts immediately.

DISABLED

The event binding is disabled. Capture of events matching capture specifications in this event binding is stopped immediately. Any events already captured are emitted.

Conditions

INVREQ

RESP2 values are:

4 ENABLESTATUS has an invalid CVDA value.

NOTAUTH

RESP2 values are:

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 update the event binding.

NOTFND

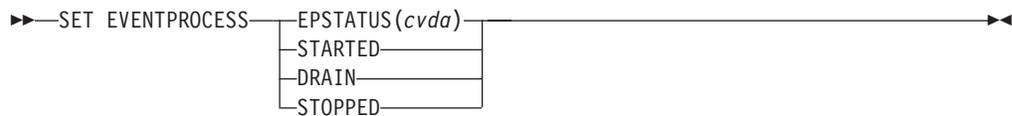
RESP2 values are:

3 The specified event binding cannot be found.

SET EVENTPROCESS

Set the status of event processing.

SET EVENTPROCESS



Conditions: INVREQ, NOTAUTH

This command is threadsafe.

Description

Use the SET EVENTPROCESS command to change the status of event processing.

Note: Do not change the status of event processing (that is, set to start, drain, or stop) while a unit of work that captures synchronous transactional events is in progress because you might cause the events to be backed out and the transaction to end abnormally.

Options

EPSTATUS(*cvda*)

a CVDA value changing the current status of event processing.

STARTED

The EVENTPROCESS state is changed to STARTED. For in-flight transactions, the capture of non-transactional events starts immediately and the capture of transactional events starts at the next sync point.

DRAIN

The EVENTPROCESS state is changed to DRAIN, and event capture is stopped immediately.

Any transactional events on the dispatcher queue will be deleted. Transactional events are not considered captured until a syncpoint occurs and a syncpoint event will not now be captured.

Any non-transactional events will be emitted. When the last event on the queue is emitted, the EVENTPROCESS state changes to STOPPED.

STOPPED

The EVENTPROCESS state is changed to STOPPED, and event capture is stopped immediately.

All events on the dispatcher queue are deleted.

Conditions

INVREQ

RESP2 values are:

- 4 Event processing cannot be started while it is draining.
- 5 EPSTATUS has an invalid CVDA value.

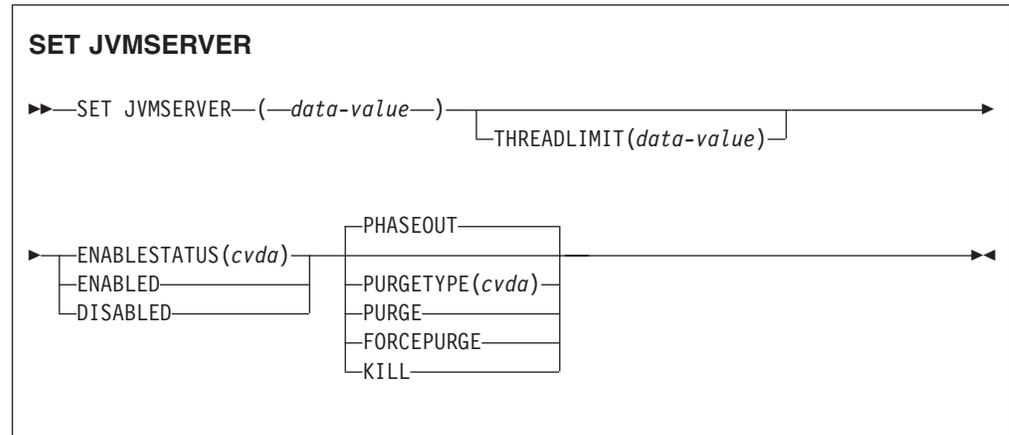
NOTAUTH

RESP2 values are:

- 100 The user associated with the issuing task is not authorized to use this command.

SET JVMSERVER

Change the status of an installed JVMSERVER resource.



Conditions: INVREQ, NORMAL, NOTAUTH, NOTFND

This command is threadsafe.

Description

The **SET JVMSERVER** command enables, disables, and modifies a JVM server.

Options

ENABLESTATUS(*cvda*)

Set the status of the JVMSERVER resource:

ENABLED

Enable the JVMSERVER resource. CICS creates a Language Environment enclave and starts a JVM.

DISABLED

Disable the JVMSERVER resource. CICS finishes processing any requests and then stops the JVM and the Language Environment enclave.

JVMSERVER(*data-value*)

Specify the 8-character name of the JVMSERVER resource that you want to change.

PURGETYPE(*cvda*)

Specifies how tasks associated with the named JVM server are to be purged when you disable the resource. If you do not set a value, CICS uses the PHASEOUT option.

FORCEPURGE

Tasks that are running in the JVM server are force purged by the **SET TASK FORCEPURGE** command. Any threads running in the JVM are stopped. If the JVMSERVER resource remains in the BEING DISABLED state, some tasks could not be force purged.

Use this option only after you have used the PURGE option. Data integrity is not guaranteed.

KILL Tasks that are running in the JVM server are terminated by the **SET TASK KILL** command. Any threads running in the JVM are stopped. The JVMSERVER resource enters the DISABLED state and all work is terminated. However, CICS might be left in an unstable state.

Use this option only after you have used the PURGE and FORCEPURGE options. System and data integrity are not guaranteed. CICS might terminate abnormally.

PHASEOUT

Tasks that are running in the JVM server continue until completion, but no new work is accepted by the JVM server. When all the tasks are finished, the JVMSERVER resource enters the DISABLED state. PHASEOUT is the default value.

PURGE

Tasks that are running in the JVM server are purged by the **SET TASK PURGE** command. Any threads running in the JVM are stopped. CICS purges tasks only when system and data integrity can be maintained. If the JVMSERVER resource remains in the BEING DISABLED state, some tasks could not be purged.

THREADLIMIT (*data-value*)

Set the maximum number of threads that are allowed by CICS when calling the JVM. Each thread is attached using a T8 TCB. The valid range is 1 - 256 threads. If the thread limit exceeds the maximum of 1024 threads allowed for the CICS region, taking into account the other enabled and disabled JVMSERVER resources, CICS adjusts the value accordingly.

Conditions

INVREQ

RESP2 values:

- 1 Insufficient threads available to satisfy the requested maximum number.
- 2 ENABLESTATUS value is not valid.
- 3 THREADLIMIT value specified is not valid because it is 0 or greater than 256.
- 4 The Language Environment enclave was not created successfully.
- 7 The JVMSERVER cannot be disabled because it is in the enabling state.
- 8 You must PURGE the tasks associated with the JVM server before using the FORCEPURGE option.
- 9 The PURGETYPE option has an invalid CVDA value.
- 10 You must FORCEPURGE the tasks associated with the JVM server before using the KILL option.

NORMAL

RESP2 value:

- 1 The number of available threads is less than the THREADLIMIT value requested.

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 particular resource in the way required by this command.

NOTFND

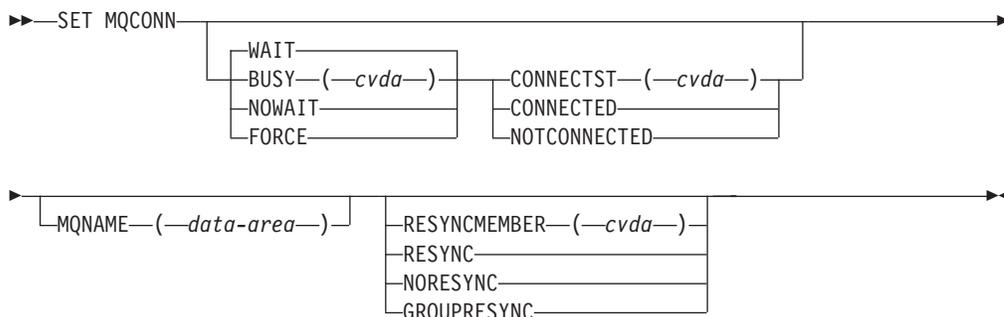
RESP2 value:

- 3 The JVMSERVER resource was not found.

SET MQCONN

Change information about the attributes of the connection between CICS and WebSphere MQ, and start or stop the connection.

SET MQCONN



Conditions: NORMAL, NOTAUTH, NOTFND, INVREQ

This command is threadsafe.

Description

Use the SET MQCONN command to change attributes of the currently installed MQCONN resource definition, which defines the connection to WebSphere MQ, and to start and stop the connection.

Because only one MQCONN resource definition can be installed at a time, the name of the MQCONN resource definition is not required on input.

This command does not set the INITQNAME attribute of the MQCONN resource definition, which specifies the name of the default initiation queue. If you want to change the MQINI resource definition, you must reinstall the MQCONN resource definition with an appropriate MQINI attribute.

Restriction: This command cannot be used in a remote program that is linked by a distributed program link command.

Options

BUSY

This option is valid only with CONNECTST when setting the CICS-WebSphere MQ connection NOTCONNECTED. If you specify CONNECTED, BUSY is ignored. The CVDA values are as follows:

FORCE

Any CICS transactions currently by using WebSphere MQ are abnormally ended, and the connection to WebSphere MQ is stopped. The request is synchronous in nature; that is, control is not returned to the application until the connection is stopped.

NOWAIT

The connection to WebSphere MQ is quiesced. Existing transactions are allowed to finish before the connection is stopped. The request is asynchronous in nature; that is, control is returned to the application before the connection is stopped.

WAIT The connection to WebSphere MQ is quiesced. Existing transactions are allowed to finish before the connection is stopped. The request is synchronous in nature; that is, control is not returned to the application until the connection is stopped. WAIT is the default.

CONNECTST

Starts or stops the connection between CICS and WebSphere MQ. The CVDA values are as follows:

CONNECTED

Starts the CICS-WebSphere MQ connection. This action has the same effect as issuing a CKQC START command to start the CICS-WebSphere MQ adapter. If the requested queue manager is active, control returns when CICS and WebSphere MQ are connected. If the requested queue manager is not active, CICS returns a NORMAL response with RESP2=8, indicating that the CICS-WebSphere MQ adapter is in connecting state and reconnects to WebSphere MQ as soon as the requested queue manager becomes active.

NOTCONNECTED

Stops the CICS-WebSphere MQ connection. The value that you specify for the BUSY option determines whether existing transactions are stopped or allowed to complete, and at what stage control is returned to the application. The default is BUSY(WAIT), allowing existing transactions to finish before the connection is stopped, and not returning control to the application until the connection is stopped.

MQNAME

Specifies the 1 - 4 character name of a WebSphere MQ queue manager or queue-sharing group to which CICS is to connect. CICS attempts to connect to the queue manager or to any active member of the queue-sharing group. You can change MQNAME only when CICS is not connected to WebSphere MQ.

When you specify MQNAME, the queue manager name or queue-sharing group that you specified in the MQNAME attribute of the installed MQCONN resource definition is replaced with the name that you specified on this command. If you want to revert to the original queue manager or queue-sharing group, set MQNAME again.

RESYNCMEMBER

This option applies only if you have specified a queue-sharing group for the CICS-WebSphere MQ connection. RESYNCMEMBER specifies the strategy that CICS adopts if outstanding units of work are being held for the last queue manager to which CICS was connected from the queue-sharing group.

Units of work that are shunted indoubt are not included in this process, because CICS itself cannot resolve those units of work at this time.

Resynchronization for those UOWs occurs when CICS has resynchronized with its remote coordinator. The CVDA values are as follows:

RESYNC

CICS connects to the same queue manager.

NORESYNC

CICS makes one attempt to connect to the same queue manager. If that attempt fails, CICS connects to any member of the queue-sharing group and issues a warning message about the outstanding units of work.

GROUPRESYNC

CICS connects to any member of the queue-sharing group. The queue manager is chosen by WebSphere MQ and it asks CICS to resolve indoubt units of work on behalf of all eligible queue managers in the queue-sharing group. This function is called *group unit of recovery*. The GROUPRESYNC option can be used only when you are running a release of WebSphere MQ that supports group unit of recovery for CICS and when the GROUPUR attribute has been enabled in the WebSphere MQ queue managers.

When an attempt is made to connect CICS to WebSphere MQ by using an **EXEC CICS SET MQCONN CONNECTED** command and RESYNCMEMBER(GROUPRESYNC) is set but WebSphere MQ does not support group unit of recovery, or group unit of recovery is not enabled, then WebSphere MQ rejects the connection attempt. The connection attempt results in the SET command failing with INVREQ and RESP2=9 (connection error).

Do not change the setting for RESYNCMEMBER when units of work are outstanding in WebSphere MQ because this means that units of work cannot be resolved. A unit of work held in CICS is identified with a resource manager qualifier. When RESYNCMEMBER(GROUPRESYNC) is used the qualifier is the name of the queue-sharing group, otherwise the qualifier used is the name of the individual queue manager.

Conditions

NORMAL

RESP2 values:

- 8 Waiting for WebSphere MQ. This situation can occur following a CONNECTST with a CVDA of CONNECT.

NOTAUTH

The user associated with the issuing task is not authorized to use this command, or is not authorized to access this resource.

RESP2 values:

- 100 Command authorization failure.

NOTFND

RESP2 values:

- 1 No MQCONN resource definition is currently installed.

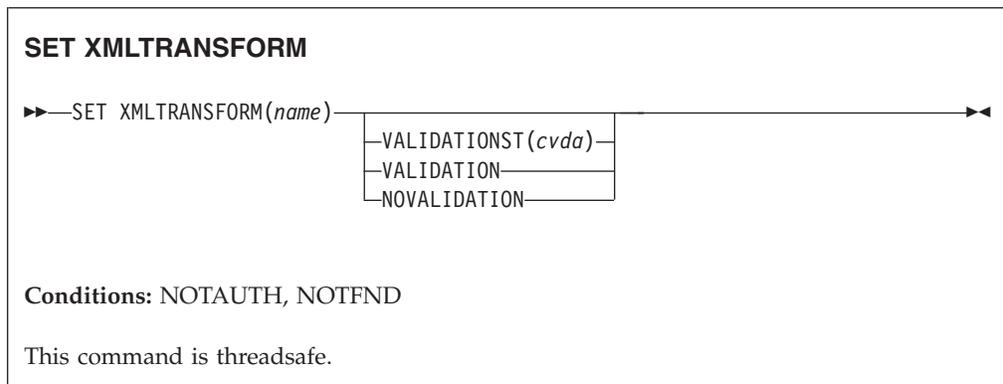
INVREQ

RESP2 values:

- 2 SET NOTCONNECTED with the FORCE or WAIT option has been specified, but this transaction is itself using the CICS-WebSphere MQ interface.
- 3 MQNAME cannot be set because the connection is active.
- 4 BUSY value is not valid.
- 5 MQNAME contains characters that are not valid.
- 6 CONNECTST value is not valid.
- 7 RESYNCMEMBER value is not valid.
- 9 Connection error.
- 10 MQNAME value is not a valid queue manager or queue-sharing group.

SET XMLTRANSFORM

Use the **SET XMLTRANSFORM** command to enable or disable the validation of an installed XMLTRANSFORM resource.



Description

CICS dynamically creates an XMLTRANSFORM resource when you install a BUNDLE or ATOMSERVICE resource. The XMLTRANSFORM resource defines the location of the XML binding and schema in z/OS UNIX. Use the **SET XMLTRANSFORM** command to enable or disable validation. If enabled, CICS checks that the XML is valid against the schema.

Options

VALIDATIONST(*cvda*)

Indicates whether full validation is enabled for the XMLTRANSFORM resource. CVDA values are as follows:

VALIDATION

Full validation is enabled.

NOVALIDATION

Full validation is disabled.

Because validating an XML message against its schema incurs considerable processing overhead, typically you will specify

VALIDATIONST(NOVALIDATION). If VALIDATIONST(NOVALIDATION) is specified, checking is performed to ensure that the message contains well-formed XML, but with no guarantee that the XML is valid.

Full validation ensures that the XML in the message is valid with respect to the XML schema; you might want to specify VALIDATIONST(VALIDATION) when you are developing an application.

XMLTRANSFORM(*name*)

Specifies the 1-32 character name of the XMLTRANSFORM resource.

Conditions

NOTAUTH

RESP2 values are:

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 particular resource in the way required by this command.

NOTFND

RESP2 values are:

3 The XMLTRANSFORM cannot be found.

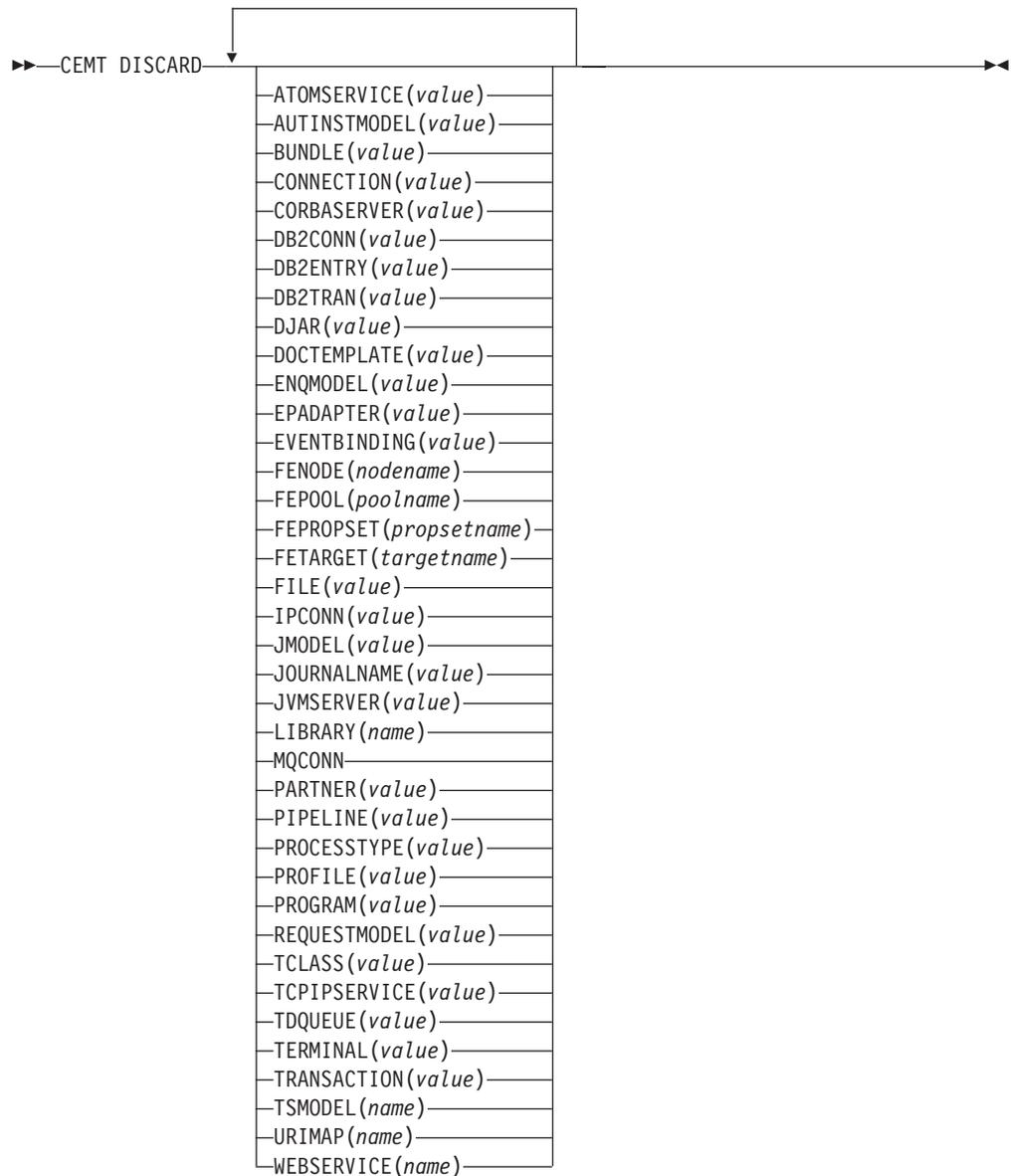
Appendix D. New CEMT commands

CICS Transaction Server for z/OS, Version 4 Release 2 provides new CEMT commands.

CEMT DISCARD

Syntax

CEMT DISCARD



Options

ATOMSERVICE(*value*)

Specifies the name of the ATOMSERVICE definition to be discarded. The name can be up to 8 characters long. You must disable the ATOMSERVICE resource before discarding it.

AUTINSTMODEL(*value*)

Specifies the name of the autoinstall model that is to be removed. The name can be up to 8 characters long. The **DISCARD AUTINSTMODEL** command removes the installed name of an autoinstall model from the CICS system and the CICS catalog.

BUNDLE(*value*)

Specifies the name of the BUNDLE definition to be discarded. The name can be up to 8 characters long. You must disable the BUNDLE resource before discarding it. Discarding the bundle automatically discards any other resources that were created when the bundle was installed.

CONNECTION(*value*)

Specifies the 4-character identifier of the CONNECTION definition to be discarded. When a connection is removed, all its associated sessions are also removed.

For deletion to be successful:

- A non-remote connection must be out of service.
- A remote connection must have no active sessions.
- For MRO connections, the interregion communications facility (IRC) must be closed. (You can use **CEMT SET IRC CLOSED** to close it.)
- If the local system is a member of a z/OS Communications Server generic resource group, APPC synclevel 2 connections must not have any outstanding recovery information.

Other types of connection *can* be discarded, even if there is recovery work outstanding for them. However, it is recommended that you do not discard them if there is. You can use the **INQUIRE CONNECTION RECOVSTATUS** command to check.

In some unusual circumstances, it is possible for deletion of an LU61 connection to fail, even though the connection is out of service, because its sessions are still in-service. If this happens, set the connection **INSERVICE**, then **OUTSERVICE**, then reissue the **DISCARD** command.

CORBASERVER(*value*)

Specifies the 4-character name of the CORBASERVER definition to be discarded.

If any of the beans installed in the named CorbaServer are in use when the **DISCARD** command executes, the state of the CorbaServer turns to **DISCARDING**. No new work is accepted. When all outstanding work has completed, the CORBASERVER definition is removed.

Copies of deployed JAR files are deleted from the CorbaServer's shelf, together with any other files on the shelf that belong to the CorbaServer. The CorbaServer's shelf directory is deleted.

DB2CONN(*value*)

Specifies that the currently installed DB2CONN is to be removed. You can only discard a DB2CONN when the CICS DB2 interface is not active. Note that discarding a DB2CONN also discards all currently installed DB2ENTRYs and DB2TRANS.

DB2ENTRY (*value*)

Specifies the name of an installed DB2ENTRY to be removed. A DB2ENTRY can only be discarded when it is in a disabled state.

DB2TRAN (*value*)

Specifies the name of the DB2TRAN to be removed. Transaction IDs referenced by this DB2TRAN will be routed to the pool after the DB2TRAN is removed.

DJAR (*value*)

Specifies the 1–32 character name of the DJAR definition to be discarded.

As well as the DJAR definition, the shelf copy of the deployed JAR file and the beans it contains are also removed from the system. The current set of JVM instances are marked for termination, so that each JVM instance will not be reused after it terminates. New JVM instances load all their classes and do not use old versions of classes whose bytecodes have been replaced.

Passivated stateful session bean instances are not deleted by this operation because, in a workload-managed CICS EJB server, they could be activated by other AORs.

References to the home interfaces of enterprise beans in the deployed JAR file are not retracted from the namespace because they might still be installed in other AORs. However, before discarding a deployed JAR file from the last AOR in which it is installed, you might want to issue a **PERFORM DJAR RETRACT** command to retract references to home interfaces.

DOCTEMPLATE (*value*)

Specifies the name of the document template to be removed.

ENQMODEL (*value*)

Specifies the 8-character identifier of the resource to be discarded.

EPADAPTER (*value*)

Specifies the name of the EP adapter that is to be removed. You must disable the EP adapter before discarding it.

EVENTBINDING (*value*)

Specifies the name of the event binding that is to be removed. You must disable the event binding before discarding it.

FENode (*nodename*)

The name of the FEPI node to be discarded.

FEP0o1 (*poolname*)

The name of the FEPI pool to be discarded.

FEP0o1 (*poolname*)

The name of the FEPI pool to be discarded.

FEPPropset (*propsetname*)

The name of the FEPI property set to be discarded.

FETarget (*targetname*)

The name of the FEPI target to be discarded.

FILE (*value*)

Specifies the name of the file that is to be removed. The name can be up to 8 characters long. The **DISCARD FILE** command removes the installed name of a file from the CICS system and the CICS catalog.

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.

JMODEL(*value*)

Specifies the name of the installed journal model that you want to remove. The name can be up to 8 characters long. The **DISCARD JMODEL** command removes the journal model name from the CICS system and the CICS catalog. It does not affect any existing journals that have been defined using the journal model name. These continue to use their existing definitions unless they are discarded using the **DISCARD JOURNALNAME** command.

JOURNALNAME(*value*)

Specifies the journal name that you want to remove. The name can be up to 8 characters long. The **DISCARD JOURNALNAME** command removes the journal name from the CICS system and the CICS catalog. It does not affect the VSAM data sets currently using the journal as a forward recovery log or autojournal; they continue to use the existing definition until the data sets are closed.

To discard journals defined with a numeric identifier for auto-journaling or forward recovery logs, specify the journal name as DFHJ*nn*, where *nn* is the journal number in the range 1–99.

Note: Specifying DFHJ01 on this command refers to a user journal, DFHJ01.

Any future reference to the journal name will cause it to be reinstalled using the journal model resource in effect at that time.

JVMSEVER(*value*)

Specifies the name of a JVMSEVER resource that you want to remove. The name can be up to 8 characters long. The **DISCARD JVMSEVER** command removes the JVMSEVER definition from the CICS system and the CICS catalog.

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.

Specifying a LIBRARY name of DFHRPL is invalid and results in the message NOT VALID FOR RPL being displayed.

MQCONN

Specifies that the currently installed MQCONN definition is to be removed. You can discard an MQCONN definition when the connection between CICS and WebSphere MQ is not active. When you discard an MQCONN definition, the associated MQINI definition is also discarded.

PARTNER(*value*)

Specifies the name of the partner that is to be removed. The name can be up to 8 characters long. The **DISCARD PARTNER** command removes the installed name of a partner from the CICS system and the CICS catalog.

PIPELINE(*value*)

Specifies the name of the PIPELINE that is to be removed. The name can be up to 8 characters long. The **DISCARD PIPELINE** command removes the PIPELINE from the CICS system and the CICS catalog.

PROCESSTYPE(*value*)

Specifies the 8-character name of the PROCESSTYPE to be removed. Only disabled processtypes can be discarded. If you are using BTS in a sysplex, it is strongly recommended that you use CPSM to remove them.

PROFILE(*value*)

Specifies the name of the profile that is to be removed. The name can be up to 8 characters long. The **DISCARD PROFILE** command removes the installed name of a profile from the CICS system and the CICS catalog.

PROGRAM(*value*)

Specifies the name of the program, map set, or partition set that is to be removed. The name can be up to 8 characters long. The **DISCARD PROGRAM** command removes the installed name of a program, map set, or partition set from the CICS system and the CICS catalog.

REQUESTMODEL(*value*)

Specifies the name of the REQUESTMODEL resource definition to be removed.

TCLASS(*value*)

Specifies the name of the transaction class that is to be removed. The name can be up to 8 characters long. The **DISCARD TCLASS** command removes the installed name of a transaction class from the CICS system and the CICS catalog.

Note: TRANCLASS is also accepted as a synonym for TCLASS.

TCPIPSERVICE(*value*)

Specifies the name of the TCPIP service to be removed.

TDQUEUE(*value*)

Specifies the name of the transient data queue that is to be removed from a running CICS system. The name can be up to 4 characters long.

Note: When you discard a transient data queue, an implicit **DELETEQ** command is issued to free the control intervals used by the queue if the queue is an intrapartition queue. If the queue is a logically recoverable intrapartition queue, a sync point is taken after the **DELETEQ** request, but before the queue definition is discarded. This ensures that the associated control intervals have been freed before the definition is discarded.

TERMINAL(*value*)

Specifies the name of the terminal to be removed. The name can be up to 4 characters long. The **DISCARD TERMINAL** command removes the installed name of a terminal from the CICS system and the CICS catalog.

Only terminals that use z/OS Communications Server as an access method or consoles are eligible for deletion. Terminals that represent sessions cannot be deleted with **DISCARD TERMINAL**; use the **DISCARD CONNECTION** command instead.

TRANSACTION(*value*)

Specifies the name of the transaction that is to be removed. The name can be up to 8 characters long. The **DISCARD TRANSACTION** command removes the installed name of a transaction from the CICS system and the CICS catalog.

TSMODEL(*name*)

Specifies the name of the TS model that is to be removed. The name can be up to 8 characters long. You cannot discard a TSmodel that is in use, or one which has a name beginning with DFH.

URIMAP(*name*)

Specifies the name of the URIMAP definition that is to be removed. The name can be up to 8 characters long. The **DISCARD URIMAP** command removes the URIMAP definition from the CICS system and the CICS catalog. For a URIMAP definition to be removed, it must be disabled individually (using the **SET URIMAP** command). Disabling a virtual host (using the **SET HOST** command) does not allow the URIMAP definitions that make up the virtual host to be removed.

WEBSERVICE(*name*)

Specifies the name of the web service that is to be removed. The name can be up to 8 characters long. The **DISCARD WEBSERVICE** command removes the web service from the CICS system and the CICS catalog.

CEMT INQUIRE ATOMSERVICE

Retrieve information about ATOMSERVICE resource definitions.

In the CICS Explorer, the Atom services operations view provides a functional equivalent to this command.

The resource signature

You can use this command to display the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Input

Press the Clear key to clear the screen. You can start this transaction in two ways:

- Type **CEMT INQUIRE ATOMSERVICE**. You obtain a display that lists the current status.
- Type **CEMT INQUIRE ATOMSERVICE** followed by the attributes that are necessary to limit the range of information that you require. So, for example, if you enter **CEMT INQUIRE ATOM FEED**, the resulting display shows you the details of only those ATOMSERVICE resource definitions that are for Atom feed documents.

To enable or disable the ATOMSERVICE resource definition:

- Enable or disable the BUNDLE resource if the Atom feed is deployed in a bundle. The resource signature of the ATOMSERVICE resource indicates how it was created.
- Overtyping your changes on the INQUIRE screen after tabbing to the appropriate field.
- Use the **CEMT SET ATOMSERVICE** command.

ALL

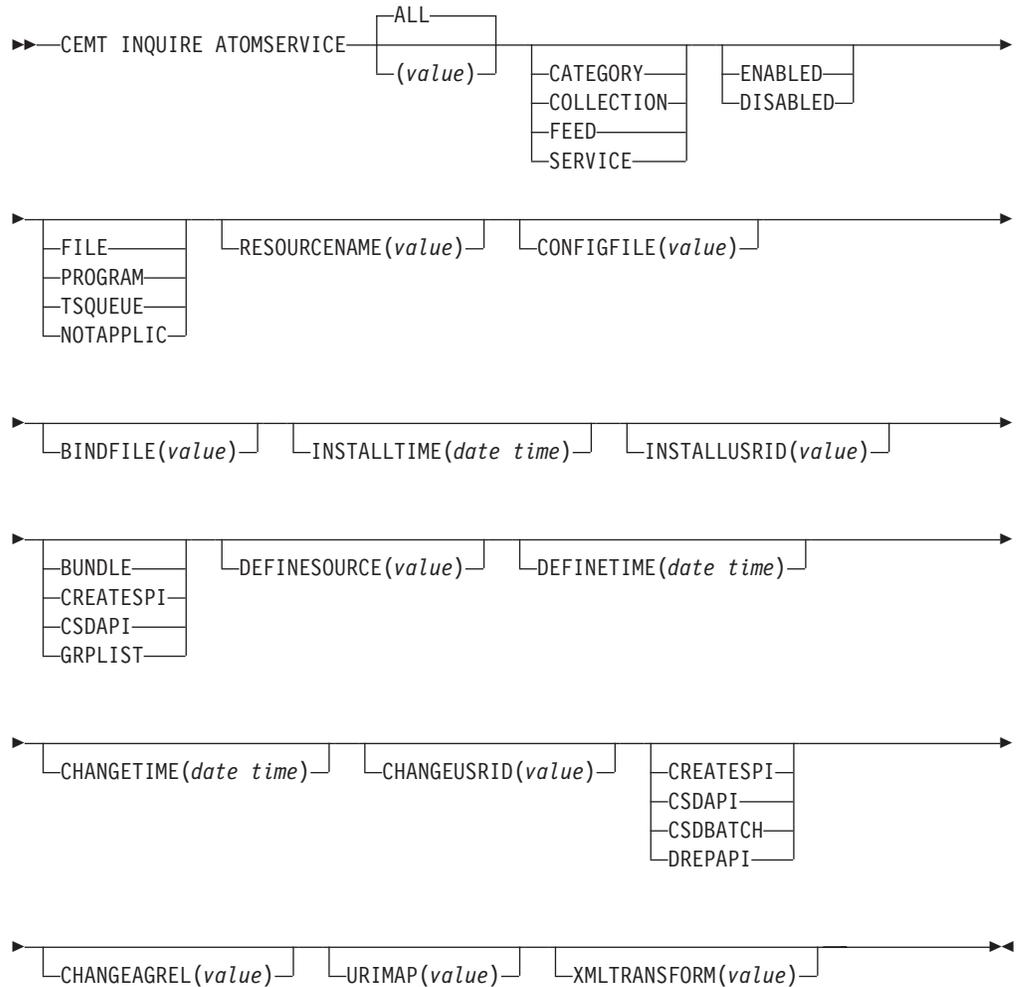
Is the default. Information about all ATOMSERVICE definitions is displayed, unless you specify a selection of definitions to be queried.

(*value*)

Is the name (1 - 8 characters) of an installed ATOMSERVICE definition.

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format of all the attributes for the specified ATOMSERVICE definition.

CEMT INQUIRE ATOMSERVICE



Displayed fields

ATOMTYPE(*value*)

Displays the type of Atom document that is produced by this ATOMSERVICE definition. The possible values are as follows:

CATEGORY

An Atom category document, which lists the categories of documents in a collection.

COLLECTION

An Atom collection document, which contains a group of entry documents that can be edited.

FEED

An Atom feed document, which describes the metadata for a feed, and contains entry documents that provide data for the feed.

SERVICE

An Atom service document, which provides information about the collections of entry documents that are available on the server and can be added to or edited.

BINDFILE(*value*)

Displays the name and file path of the XML binding specified in this ATOMSERVICE resource definition. The XML binding is stored in z/OS UNIX System Services, and it specifies the data structures used by the CICS resource that supplies the data for the Atom document that is returned for this resource definition. Service and category documents do not use an XML binding; if the value of ATOMTYPE is SERVICE or CATEGORY, this field is blank.

CHANGEAGENT(*value*)

Displays a value that identifies the agent that made the last change to the resource definition. You cannot use CEMT to filter on some of these values because they are duplicated. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL(*value*)

Displays the 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME(*date time*)

Displays the date and time when the resource definition was last changed. The format of the date depends on the value that you selected for the DATFORM system initialization parameter for your CICS region. The format of the time is hh:mm:ss.

CHANGEUSRID(*value*)

Displays the 8-character user ID that ran the change agent.

CONFIGFILE(*value*)

Displays the name and file path of the Atom configuration file specified in this ATOMSERVICE resource definition. The Atom configuration file is stored in z/OS UNIX System Services, and it contains XML that specifies metadata and content for the Atom document.

DEFINESOURCE(*value*)

Displays the source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT option. For details, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME(*date time*)

Displays the date and time when the resource was created. The format of the date depends on the value that you selected for the DATFORM system initialization parameter for your CICS region. The format of the time is hh:mm:ss.

ENABLESTATUS(*value*)

Displays the overall status of the ATOMSERVICE resource definition. The possible values are as follows:

ENABLED

The ATOMSERVICE resource definition is enabled.

DISABLED

The ATOMSERVICE resource definition is disabled. An ATOMSERVICE definition with this status can be discarded.

INSTALLAGENT(*value*)

Displays a value that identifies the agent that installed the resource. You cannot use CEMT to filter on some of these values because they are duplicated. The possible values are as follows:

BUNDLE

The resource was installed by a bundle deployment.

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME(*date time*)

Displays the date and time when the resource was installed. The format of the date depends on the value that you selected for the DATFORM system initialization parameter for your CICS region. The format of the time is hh:mm:ss.

INSTALLUSRID(*value*)

Displays the 8-character user ID that installed the resource.

RESOURCENAME(*value*)

Displays the name of the CICS resource that provides the data for this Atom feed or collection. This field does not apply for an Atom service or category document.

RESOURCETYPE(*value*)

Displays the type of CICS resource that provides the data for this Atom feed or collection. This field does not apply for an Atom service or category document. The possible values are as follows:

FILE

A CICS file.

PROGRAM

A service routine, which is a CICS application program written to supply content for Atom entries.

TSQUEUE

A temporary storage queue.

NOTAPPLIC

If the value of ATOMTYPE is SERVICE or CATEGORY, the resource type is not applicable.

URIMAP

Displays the 8-character URIMAP name that indicates the URI associated with this ATOMSERVICE definition. If there is no dynamically generated URIMAP associated with this ATOMSERVICE definition, this field is empty.

XMLTRANSFORM

Displays the 32-character name of the XMLTRANSFORM resource associated with the ATOMSERVICE definition. If the value of ATOMTYPE is SERVICE or CATEGORY, this field is empty.

CEMT INQUIRE BUNDLE

Retrieve information about a BUNDLE resource.

In the CICS Explorer, the Bundles operations view provides a functional equivalent to this command.

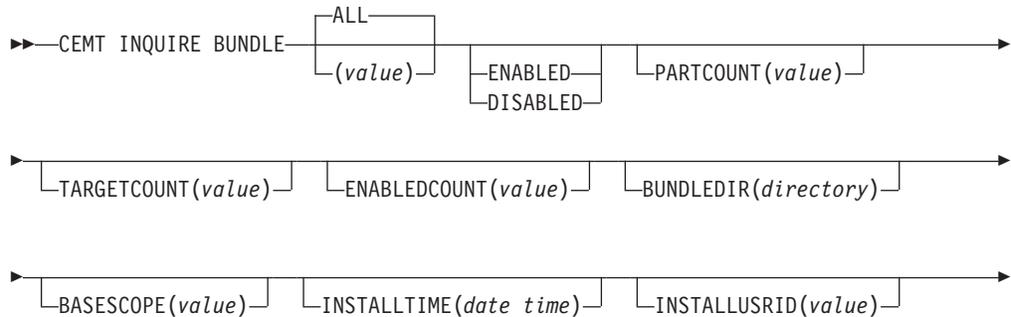
The resource signature

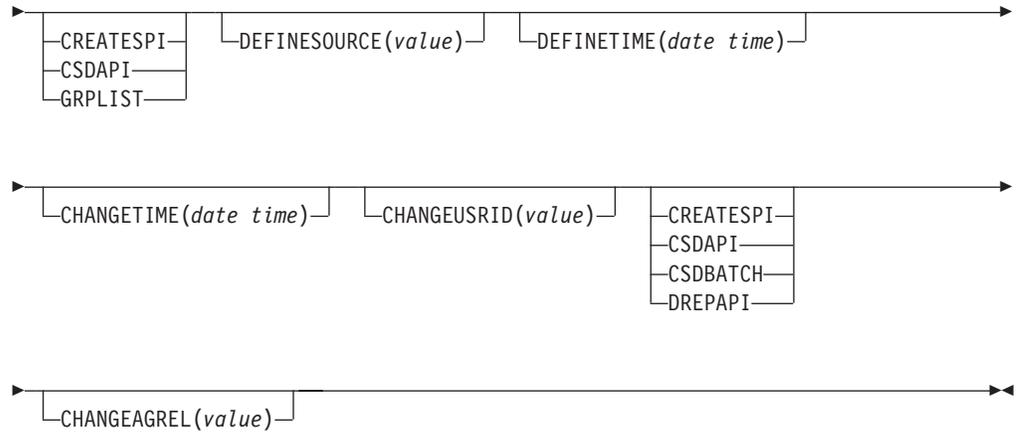
You can use this command to display the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Input

Press the Clear key to clear the screen and then enter CEMT INQUIRE BUNDLE to obtain information about one or more bundles that are installed in the CICS region.

CEMT INQUIRE BUNDLE





Displayed fields

BASESCOPE(*value*)

Displays the URI that defines the root namespace for the contents of the bundle.

BUNDLEDIR(*directory*)

Displays the directory on z/OS UNIX where the bundle is deployed.

CHANGEAGENT(*value*)

Displays a value that identifies the agent that made the last change to the resource definition. You cannot use CEMT to filter on some of these values because they are duplicated. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL(*value*)

Displays the 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME(*date time*)

Displays the date and time when the resource definition was last changed. The format of the date depends on the value that you selected for the DATFORM system initialization parameter for your CICS region. The format of the time is hh:mm:ss.

CHANGEUSRID(*value*)

Displays the 8-character user ID that ran the change agent.

DEFINESOURCE(*value*)

Displays the source of the resource definition. The DEFINESOURCE value

depends on the CHANGEAGENT option. For details, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME (*date time*)

Displays the date and time when the resource was created. The format of the date depends on the value that you selected for the DATFORM system initialization parameter for your CICS region. The format of the time is hh:mm:ss.

ENABLEDCOUNT (*value*)

Displays the current number of resources that were dynamically created by the bundle and are enabled in the CICS region.

ENABLESTATUS (*value*)

Displays the overall status of the bundle. The possible values are as follows:

ENABLED

The bundle is enabled for use.

DISABLED

The bundle is disabled.

INSTALLAGENT (*value*)

Displays a value that identifies the agent that installed the resource. You cannot use CEMT to filter on some of these values because they are duplicated. The possible values are as follows:

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME (*date time*)

Displays the date and time when the resource was installed. The format of the date depends on the value that you selected for the DATFORM system initialization parameter for your CICS region. The format of the time is hh:mm:ss.

INSTALLUSRID (*value*)

Displays the 8-character user ID that installed the resource.

PARTCOUNT (*value*)

Displays the total number of imports, exports, and definition statements that are defined in the bundle manifest.

TARGETCOUNT (*value*)

Displays the total number of dynamically created resources in the bundle. CICS automatically enables the BUNDLE resource when all of the dynamically created resources are in an enabled state.

CEMT INQUIRE EPADAPTER

Retrieve information about an EP adapter.

In the CICS Explorer, the EP adapter operations view provides a functional equivalent to this command.

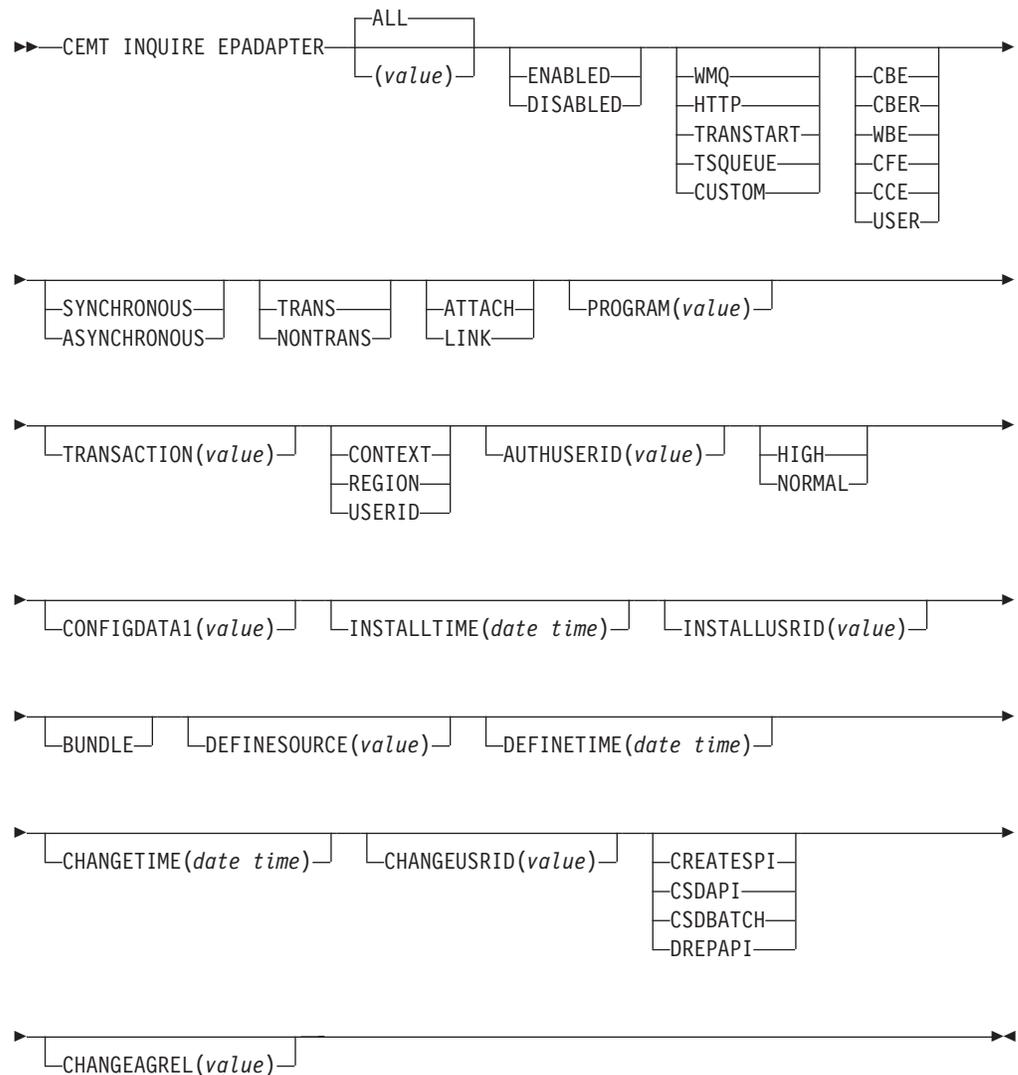
The resource signature

You can use this command to display the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Input

Press the Clear key to clear the screen and enter CEMT INQUIRE EPADAPTER to obtain information about the EP adapter.

CEMT INQUIRE EPADAPTER



Displayed fields

ADAPTERTYPE

Displays the type of this EP adapter. The possible values are as follows:

CUSTOM

A user-written EP adapter that emits events in any format that you require, to any destination.

HTTP The HTTP EP adapter that emits events to an HTTP server for consumption by products such as WebSphere Business Events and IBM Business Monitor.

TRANSTART

The Transaction Start EP adapter that emits events to a named CICS transaction.

TSQUEUE

The TSQ EP adapter that emits events to a named CICS TS queue.

WMQ The WebSphere MQ EP adapter that emits events to WebSphere MQ for consumption by products such as WebSphere Business Events and IBM Business Monitor.

AUTHORITY

Displays the authority of the EP adapter. The possible values are as follows:

CONTEXT

The EP adapter runs using the user ID of the task that caused the event to be captured. This is always the case when EMITMODE is SYNCHRONOUS or when **Use Context user ID** is specified in the advanced section of the Adapter tab for the EP adapter.

REGION

The EP adapter runs using the CICS region user ID.

USERID

The EP adapter is attached using the identifier specified in the EP adapter user ID and returned in the AUTHUSERID attribute.

AUTHUSERID

Displays the 8-character identifier that is used to attach the EP adapter transaction. This attribute is only set if AUTHORITY is USERID.

CHANGEAGENT (*value*)

Displays a value that identifies the agent that made the last change to the resource definition. You cannot use CEMT to filter on some of these values because they are duplicated. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL (*value*)

Displays the 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*date time*)

Displays the date and time when the resource definition was last changed. The format of the date depends on the value that you selected for the DATFORM system initialization parameter for your CICS region. The format of the time is hh:mm:ss.

CHANGEUSRID (*value*)

Displays the 8-character user ID that ran the change agent.

CONFIGDATA1

Displays a 64-character data area containing the primary configuration data item for the EP adapter. If the primary configuration data item is less than 64 bytes, the field is padded with blanks. The data item displayed is dependent upon ADAPTERTYPE, as follows:

CUSTOM

Displays the first 64 bytes of custom EP adapter configuration data.

HTTP Displays the 8-character name of the URIMAP definition to be used by an HTTP EP adapter to locate the HTTP server.

TRANSTART

Displays the 4-character name of the event consumer transaction that is started by a Transaction Start EP adapter.

TSQUEUE

Displays the 16-character name of the temporary storage queue for the event emitted by a TSQ EP adapter.

WMQ Displays the 48-character name of the WebSphere MQ queue for event messages emitted by this WebSphere MQ EP adapter. This data is in the code page defined by the **LOCALCCSID** system initialization parameter.

DATAFORMAT

Displays a CVDA value indicating the format of events emitted by this EP adapter. The possible values are as follows:

CBE Common Base Event format for consumption by products such as IBM Business Monitor.

CBER Common Base Event REST format for the IBM Business Monitor REST HTTP server.

CCE CICS Container Event format.

CFE CICS Flattened Event format.

USER The format is user-defined.

WBE WebSphere Business Events XML format.

DEFINESOURCE (*value*)

Displays the source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT option. For details, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME (*date time*)

Displays the date and time when the resource was created. The format of the

date depends on the value that you selected for the DATFORM system initialization parameter for your CICS region. The format of the time is hh:mm:ss.

EMITMODE (*emitmodevalue*)

Displays the event emission mode of the event binding:

ASYNCHRONOUS

Event emission is asynchronous of the capturing transaction.

SYNCHRONOUS

Event emission is synchronous with the capturing transaction.

ENABLESTATUS

Displays the status of the EP adapter:

DISABLED

The EP adapter is disabled.

ENABLED

The EP adapter is enabled.

INSTALLAGENT (*value*)

Displays a value that identifies the agent that installed the resource. Only one value is possible:

BUNDLE

The resource was installed by a bundle deployment.

INSTALLTIME (*date time*)

Displays the date and time when the resource was installed. The format of the date depends on the value that you selected for the DATFORM system initialization parameter for your CICS region. The format of the time is hh:mm:ss.

INSTALLUSRID (*value*)

Displays the 8-character user ID that installed the resource.

INVOKETYPE

Displays a CVDA value indicating how the EP adapter is started. The CVDA values are as follows:

ATTACH

The EP adapter program is attached as a separate task.

LINK The EP adapter program is linked to.

PRIORITY

Displays a CVDA value indicating the dispatching priority of event emission for this EP adapter. This attribute is ignored when EMITMODE is SYNCHRONOUS. The CVDA values are as follows:

HIGH Events emitted for this EP adapter are high priority.

NORMAL

Events emitted for this EP adapter are normal priority.

PROGRAM

Displays the 8-character name of the EP adapter program. If the ADAPTERTYPE is CUSTOM, this attribute is only applicable if INVOKETYPE is LINK.

TRANSACTION

Displays the 4-character name of the transaction definition that is used if the

EP adapter transaction is attached. The TRANSACTION attribute is only applicable if INVOKETYPE is ATTACH.

TRANSMODE(*transmodevalue*)

Displays the event transactional mode of the event binding:

NONTRANS

Events are not transactional.

TRANS

Events are transactional.

CEMT INQUIRE EVENTBINDING

Retrieve information about an event binding.

In the CICS Explorer, the Events bindings operations view provides a functional equivalent to this command.

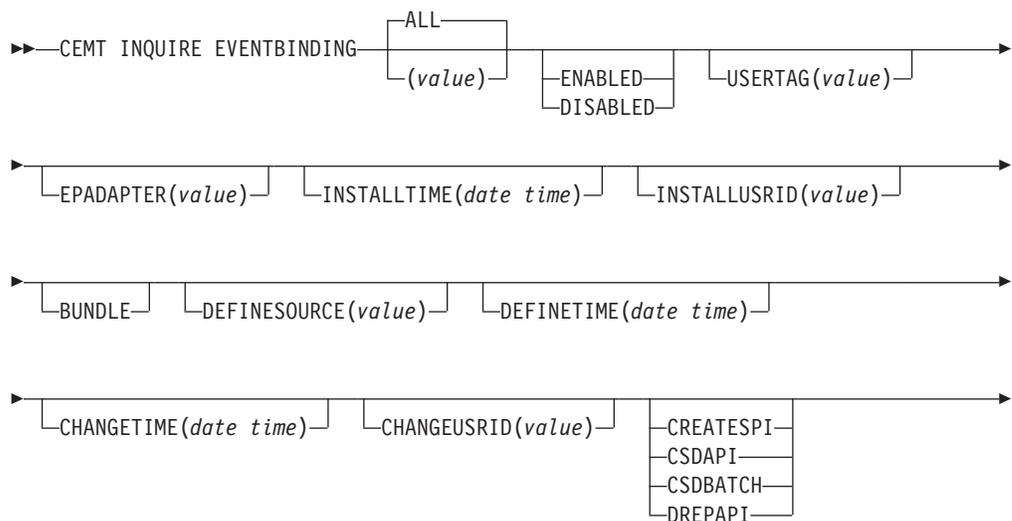
The resource signature

You can use this command to display the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Input

Press the Clear key to clear the screen and enter CEMT INQUIRE EVENTBINDING to obtain information about the event binding.

CEMT INQUIRE EVENTBINDING



CHANGEAGREL(*value*)

Displayed fields

CHANGEAGENT(*value*)

Displays a value that identifies the agent that made the last change to the resource definition. You cannot use CEMT to filter on some of these values because they are duplicated. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL(*value*)

Displays the 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME(*date time*)

Displays the date and time when the resource definition was last changed. The format of the date depends on the value that you selected for the DATFORM system initialization parameter for your CICS region. The format of the time is hh:mm:ss.

CHANGEUSRID(*value*)

Displays the 8-character user ID that ran the change agent.

DEFINESOURCE(*value*)

Displays the source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT option. For details, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME(*date time*)

Displays the date and time when the resource was created. The format of the date depends on the value that you selected for the DATFORM system initialization parameter for your CICS region. The format of the time is hh:mm:ss.

ENABLESTATUS

Displays the status of the event binding:

DISABLED

The event binding is disabled.

ENABLED

The event binding is enabled.

EPADAPTER(*value*)

Displays the 32-character name of an EP adapter.

INSTALLAGENT (*value*)

Displays a value that identifies the agent that installed the resource. Only one value is possible:

BUNDLE

The resource was installed by a bundle deployment.

INSTALLTIME (*date time*)

Displays the date and time when the resource was installed. The format of the date depends on the value that you selected for the DATFORM system initialization parameter for your CICS region. The format of the time is hh:mm:ss.

INSTALLUSRID (*value*)

Displays the 8-character user ID that installed the resource.

USERTAG (*value*)

Displays the 8-character user tag of the event binding.

CEMT INQUIRE EVENTPROCESS

Retrieve the status of event processing.

In the CICS Explorer, the Event processing operations view provides a functional equivalent to this command.

Input

Press the Clear key to clear the screen then enter CEMT INQUIRE EVENTPROCESS to retrieve the status of event processing.

CEMT INQUIRE EVENTPROCESS



Displayed fields

EPSTATUS

Displays the status of the event processing:

DRAINING

Event processing is draining.

STARTED

Event processing is started.

STOPPED

Event processing is stopped.

SCHEMALEVEL (*vvrr*)

Returns a 4-character value (*vvrr*) indicating the highest version and release of event binding schema that is supported by CICS, where *vv* is the version and *rr* is the release; for example, 0201 indicates version 2 release 1 of the event binding schema.

CEMT INQUIRE JVMSERVER

Retrieve information about the status of a JVM server runtime environment.

In the CICS Explorer, the JVM servers operations view provides a functional equivalent to this command.

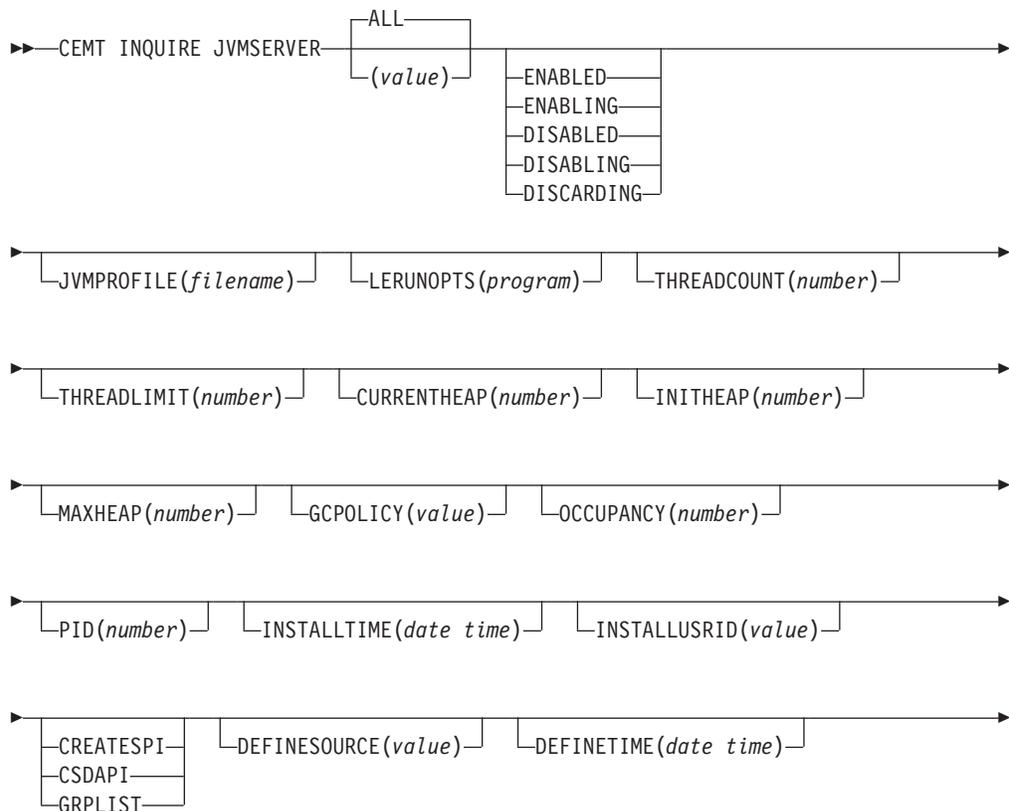
The resource signature

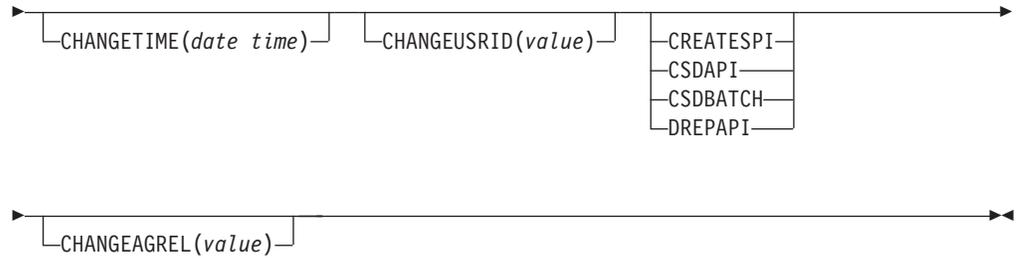
You can use this command to display the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Input

Press the Clear key to clear the screen and then enter CEMT INQUIRE JVMSERVER to obtain information about a JVM server.

CEMT INQUIRE JVMSERVER





Displayed fields

CHANGEAGENT (*value*)

Displays a value that identifies the agent that made the last change to the resource definition. You cannot use CEMT to filter on some of these values because they are duplicated. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL (*value*)

Displays the 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*date time*)

Displays the date and time when the resource definition was last changed. The format of the date depends on the value that you selected for the DATFORM system initialization parameter for your CICS region. The format of the time is hh:mm:ss.

CHANGEUSRID (*value*)

Displays the 8-character user ID that ran the change agent.

CURRENTHEAP (*number*)

Displays the current heap size in bytes that is allocated to the JVM server.

DEFINESOURCE (*value*)

Displays the source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT option. For details, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME (*date time*)

Displays the date and time when the resource was created. The format of the date depends on the value that you selected for the DATFORM system initialization parameter for your CICS region. The format of the time is hh:mm:ss.

GCPOLICY (*value*)

Displays the garbage collection policy that is being used by the JVM server.

INITHEAP(*number*)

Displays the initial heap size in bytes that is allocated to the JVM server.

INSTALLAGENT(*value*)

Displays a value that identifies the agent that installed the resource. You cannot use CEMT to filter on some of these values because they are duplicated. The possible values are as follows:

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME(*date time*)

Displays the date and time when the resource was installed. The format of the date depends on the value that you selected for the DATFORM system initialization parameter for your CICS region. The format of the time is hh:mm:ss.

INSTALLUSRID(*value*)

Displays the 8-character user ID that installed the resource.

JVMPROFILE(*filename*)

Displays the file name of the JVM profile that defines the properties of the JVM server.

LERUNOPTS(*program*)

Displays the name of the program that defines the runtime options for the Language Environment enclave.

MAXHEAP(*number*)

Displays the maximum heap size in bytes that is allowed in the JVM server.

OCCUPANCY(*number*)

Displays the heap size in bytes after the last garbage collection ran in the JVM server.

PID(*number*)

Displays the process ID (PID) of the JVM.

STATUS(*value*)

Displays the overall status of the JVM server runtime environment. The possible values are as follows:

ENABLED

The JVM server is enabled and can perform processing.

ENABLING

The JVM server is starting.

DISABLED

The JVM server is disabled and new requests cannot be processed in the runtime environment.

DISABLING

The JVM server is stopping. The JVM server can complete any work that began before it is disabled.

DISCARDING

The JVMSERVER resource is being discarded.

THREADCOUNT (*number*)

Displays the number of threads that are currently running inside the JVM server.

THREADLIMIT (*number*)

Displays the maximum number of threads that can be used by the JVM server. Each thread runs under a T8 TCB.

CEMT INQUIRE MQCONN

Retrieve information about the connection between CICS and WebSphere MQ.

In the CICS Explorer, the Websphere MQ Connections operations view provides a functional equivalent to this command.

Description

The **INQUIRE MQCONN** command returns information about the attributes of the currently installed MQCONN resource definition, which defines the connection to WebSphere MQ, and about the status of the connection.

Because only one MQCONN resource definition can be installed at a time, the name of the MQCONN resource definition is not required on input.

This command does not inquire on the INITQNAME attribute of the MQCONN resource definition, which specifies the name of the initiation queue. Use the INQUIRE MQINI command to inquire on the initiation queue name.

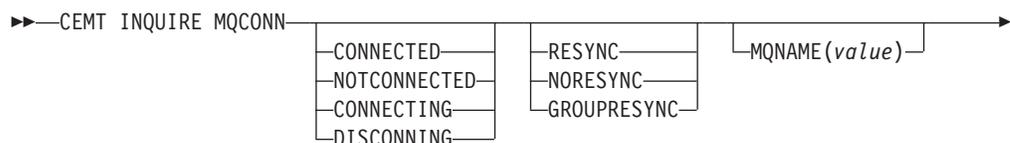
The resource signature

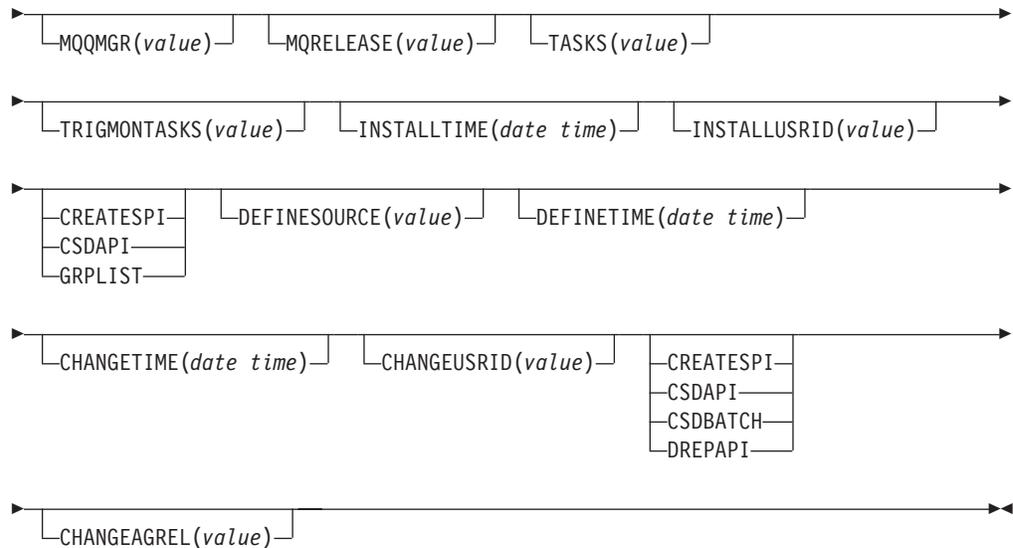
You can use this command to display the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Input

Press the Clear key to clear the screen, and type CEMT INQUIRE MQCONN (or suitable abbreviations for the keywords). The resulting display lists the status. You can tab to the highlighted or blank fields and overtype them with the required values.

CEMT INQUIRE MQCONN





Displayed fields

CHANGEAGENT(*value*)

Displays a value that identifies the agent that made the last change to the resource definition. You cannot use CEMT to filter on some of these values because they are duplicated. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL(*value*)

Displays the 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME(*date time*)

Displays the date and time when the resource definition was last changed. The format of the date depends on the value that you selected for the DATFORM system initialization parameter for your CICS region. The format of the time is hh:mm:ss.

CHANGEUSRID(*value*)

Displays the 8-character user ID that ran the change agent.

CONNECTST(*value*)

Displays the status of the CICS-MQ connection. The possible values are as follows:

CONNECTED

CICS is connected to WebSphere MQ.

NOTCONNECTED

CICS is not connected to WebSphere MQ.

CONNECTING

CICS is currently attempting to connect to WebSphere MQ.

DISCONNING

CICS is currently disconnecting from WebSphere MQ.

DEFINESOURCE(*value*)

Displays the source of the resource definition. The **DEFINESOURCE** value depends on the **CHANGEAGENT** option. For details, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME(*date time*)

Displays the date and time when the resource was created. The format of the date depends on the value that you selected for the **DATFORM** system initialization parameter for your CICS region. The format of the time is hh:mm:ss.

INSTALLAGENT(*value*)

Displays a value that identifies the agent that installed the resource. You cannot use **CEMT** to filter on some of these values because they are duplicated. The possible values are as follows:

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a **CEDA** transaction or the programmable interface to **DFHEDAP**.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME(*date time*)

Displays the date and time when the resource was installed. The format of the date depends on the value that you selected for the **DATFORM** system initialization parameter for your CICS region. The format of the time is hh:mm:ss.

INSTALLUSRID(*value*)

Displays the 8-character user ID that installed the resource.

MQNAME(*value*)

Displays the 1- to 4-character name of the WebSphere MQ queue manager or queue-sharing group that you specified in the **MQCONN** resource definition (or by using a **SET MQCONN** command) for the CICS region.

MQQMGR(*value*)

Displays the name of the WebSphere MQ queue manager to which CICS is connected or to which CICS is waiting to connect.

- If CICS is connected to WebSphere MQ, this field shows the name of the queue manager to which CICS is connected. If you specified a queue-sharing group in the **MQCONN** resource definition for the CICS region, the queue manager shown here is the one that was chosen from the group.
- If CICS is not connected to WebSphere MQ, this field usually contains blanks. However, if you specified a queue-sharing group in the **MQCONN** resource definition for the CICS region, and CICS is waiting to reconnect to a specific queue manager in the queue-sharing group because it is holding outstanding units of work for that queue manager, the name of the specific

queue manager is shown, and the status of the connection is shown as CONNECTING. For this situation to arise, the RESYNCMEMBER attribute in the MQCONN resource definition must specify resynchronization.

MQRELEASE(*value*)

If CICS is connected to WebSphere MQ, this field displays the release number of WebSphere MQ; for example, 0600. When CICS is not connected to WebSphere MQ, the field is blank.

RESYNCMEMBER(*value*)

This option applies only if you have specified a queue-sharing group for the CICS-MQ connection. RESYNCHMEMBER displays the strategy that CICS adopts if outstanding units of work are being held for the last queue manager to which CICS was connected from the queue-sharing group. Units of work that are shunted indoubt are not included in this process, because CICS itself cannot resolve those units of work at this time. Resynchronization for those UOWs occurs when CICS has resynchronized with its remote coordinator. The possible values are as follows:

RESYNC

CICS connects to the same queue manager.

NORESINC

CICS makes one attempt to connect to the same queue manager. If that attempt fails, CICS connects to any member of the queue-sharing group and issues a warning about the outstanding units of work.

GROUPRESINC

CICS connects to any member of the queue-sharing group. The queue manager is chosen by WebSphere MQ and it asks CICS to resolve indoubt units of work on behalf of all eligible queue managers in the queue-sharing group. This function is called *group unit of recovery*.

NOTAPPLIC

A queue-sharing group is not specified for the CICS-MQ connection.

TASKS(*value*)

Displays the current number of tasks that are using the CICS-MQ connection, including trigger monitor tasks.

TRIGMONTASKS(*value*)

Displays the current number of trigger monitor tasks that are using the CICS-MQ connection.

CEMT INQUIRE MQINI

Retrieve the name of the default initiation queue used for the connection between CICS and WebSphere MQ.

In the CICS Explorer, the Websphere MQ Initiation Queues operations view provides a functional equivalent to this command.

Description

The **INQUIRE MQINI** command inquires on the name of the default initiation queue used for the connection between CICS and WebSphere MQ.

The MQINI resource represents the initiation queue. MQINI is an implicit resource that exists when you install an MQCONN resource definition in the CICS region with the INITQNAME attribute specified. Only one MQCONN resource definition can be installed at a time. The name of the MQINI resource is DFHMQINI.

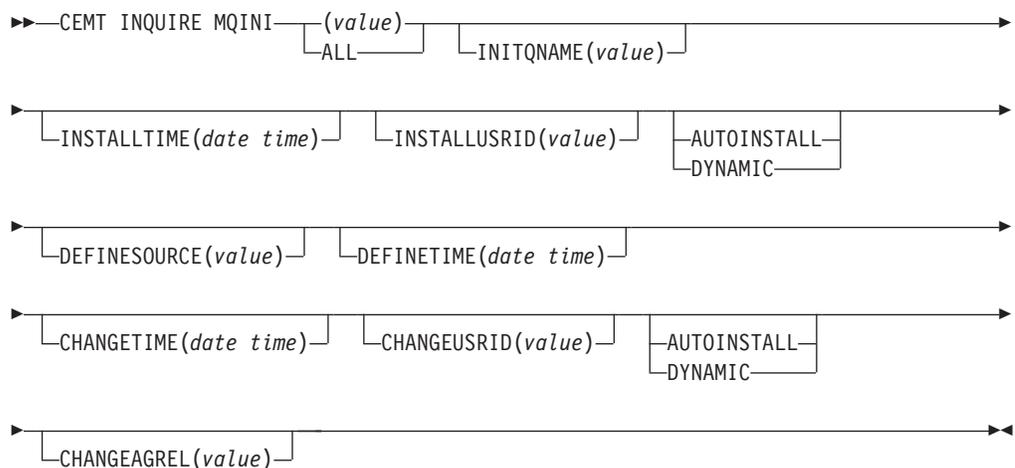
The resource signature

You can use this command to display the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Input

Press the Clear key to clear the screen, and type CEMT INQUIRE MQINI (or suitable abbreviations for the keywords). The resulting display lists the current status. You can tab to the highlighted or blank fields and overwrite them with the required values. To view the display in an expanded format, place the cursor against the resource name and press Enter.

CEMT INQUIRE MQINI



Displayed fields

CHANGEAGENT(value)

Displays a value that identifies the agent that made the last change to the resource definition. You cannot use CEMT to filter on some of these values because they are duplicated. The possible values are as follows:

AUTOINSTALL

The resource was autoinstalled as a result of specifying an initiation queue name on a CKQC START command, and the previously installed MQCONN definition did not specify a value for INITQNAME.

DYNAMIC

The resource was defined as a result of an MQCONN resource definition with INITQNAME specified.

CHANGEAGREL (*value*)

Displays the 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*date time*)

Displays the date and time when the resource definition was last changed. The format of the date depends on the value that you selected for the DATFORM system initialization parameter for your CICS region. The format of the time is hh:mm:ss.

CHANGEUSRID (*value*)

Displays the 8-character user ID that ran the change agent.

DEFINESOURCE (*value*)

Displays the source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT option. For details, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME (*date time*)

Displays the date and time when the resource was created. The format of the date depends on the value that you selected for the DATFORM system initialization parameter for your CICS region. The format of the time is hh:mm:ss.

INITQNAME (*value*)

Displays the 1- 48-character name of the default initiation queue that is represented by the MQINI resource.

INSTALLAGENT (*value*)

Displays a value that identifies the agent that installed the resource. The possible values are as follows:

AUTOINSTALL

The resource was autoinstalled as a result of specifying an initiation queue name on a CKQC START command, and the previously installed MQCONN definition did not specify a value for INITQNAME.

DYNAMIC

The resource was installed as a result of the installation of an MQCONN with INITQNAME specified.

INSTALLTIME (*date time*)

Displays the date and time when the resource was installed. The format of the date depends on the value that you selected for the DATFORM system initialization parameter for your CICS region. The format of the time is hh:mm:ss.

INSTALLUSRID (*value*)

Displays the 8-character user ID that installed the resource.

CEMT SET ATOMSERVICE

Enable or disable an ATOMSERVICE resource definition.

In the CICS Explorer, the Atom services operations view provides a functional equivalent to this command.

If the optional attributes are omitted, the command has no effect.

If you disable an ATOMSERVICE resource definition, CICS returns an HTTP response to the Web client with a 503 (Service Unavailable) status code.

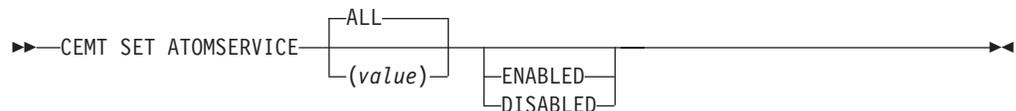
Syntax

Press the Clear key to clear the screen. You can start this transaction in two ways:

- Type CEMT SET ATOMSERVICE (or suitable abbreviations for the keywords), followed by one or more ATOMSERVICE identifiers, or ALL. The resulting display lists the current status, similar to that obtained by using the **CEMT INQUIRE ATOMSERVICE** command. You can tab to the highlighted or blank fields and overwrite them with the required values.
- Type CEMT SET ATOMSERVICE (or suitable abbreviations for the keywords), followed by one or more ATOMSERVICE identifiers or ALL, then followed by the attribute setting that you want to change. For example, `cemt s ato al e` resets the values for all ATOMSERVICE resource definitions to make them available for use (enabled).

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

SET ATOMSERVICE



Conditions: NOTAUTH, NOTFND

Options

ALL

Any changes you request are made to all resources of the specified type that you are authorized to access.

(value)

Specifies the 1- to 8-character name of the ATOMSERVICE resource definition.

ENABLESTATUS

Sets the status of the ATOMSERVICE resource definition:

ENABLED

The ATOMSERVICE resource definition is ready for use.

DISABLED

The ATOMSERVICE resource definition is disabled. An ATOMSERVICE resource definition in this state can be discarded.

CEMT SET BUNDLE

Enable or disable a BUNDLE resource.

In the CICS Explorer, the Bundles operations view provides a functional equivalent to this command.

If the optional attributes are omitted, the command has no effect.

Syntax

Press the Clear key to clear the screen, You can start this transaction in two ways:

- Type CEMT SET BUNDLE followed by one or more BUNDLE identifiers or ALL. CEMT displays a list of BUNDLE resources and the current status of each resource. You can then tab to the highlighted or blank fields and overwrite them with the required values.
- Type CEMT SET BUNDLE followed by one or more BUNDLE identifiers or ALL, followed by the attribute setting that you want to change.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET BUNDLE



Options

ALL

Apply any changes to all of the specified resources that you are authorized to access.

DISABLED

Disable one or more BUNDLE resources. If you disable a BUNDLE resource definition, CICS tries to disable all of the resources that were created and enabled as part of the bundle deployment.

ENABLED

Enable one or more BUNDLE resources. You can enable a BUNDLE resource that is in any state except DISABLING or DISCARDING.

value

Specify the 1-8 character name of the BUNDLE resource definition.

CEMT SET EPADAPTER

Enable or disable an EP adapter.

In the CICS Explorer, the EP adapter operations view provides a functional equivalent to this command.

CEMT SET EPADAPTER



Options

(value)

The 32-character name of an EP adapter.

Any transactional events on the dispatcher queue will be deleted. Transactional events are not considered captured until a syncpoint occurs and a syncpoint event will not now be captured.

Any non-transactional events will be emitted. When the last event on the queue is emitted, the state of event processing changes to stopped.

START

Set the status of event processing to started. For in-flight transactions, the capture of non-transactional events starts immediately and the capture of transactional events starts at the next sync point.

STOP

Set the status of event processing to stopped. Event capture is stopped immediately.

All events on the dispatcher queue are deleted.

CEMT SET JVMSERVER

Change the attributes of a JVM server.

In the CICS Explorer, the JVM servers operations view provides a functional equivalent to this command.

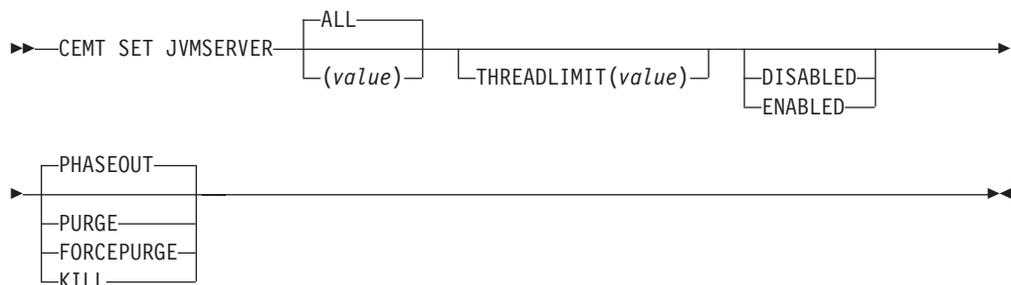
Syntax

Press the Clear key to clear the screen. You can start this transaction in two ways:

- Type CEMT SET JVMSERVER followed by one or more JVMSERVER identifiers or ALL. CEMT displays a list of JVMSERVER resources and the current status of each resource. You can then tab to the highlighted or blank fields and overwrite them with the required values.
- Type CEMT SET JVMSERVER followed by one or more JVMSERVER identifiers or ALL, followed by the attribute setting that you want to change.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET JVMSERVER



Options

ALL

Apply any changes to all of the specified resources that you are authorized to access.

DISABLED

Disable one or more JVMSERVER resources. Disabling a JVMSERVER resource shuts down the JVM server and its Language Environment enclave.

ENABLED

Enable one or more JVMSERVER resources. Enabling a JVMSERVER resource initializes the Language Environment enclave and the JVM server.

Note that any changes to JVMProfile and LERUNOPTS are used to update the JVMServer. If you enable a JVMServer after it has been in the DISABLED state, the JVMServer is completely refreshed.

FORCEPURGE

Force purge tasks that are running in the specified JVM server. CICS force purges the tasks by using the **SET TASK FORCEPURGE** command. Any threads running in the JVM are stopped. If the JVMSERVER resource remains in the BEING DISABLED state, some tasks could not be force purged.

Use this option only after you have used the PURGE option where some tasks could not be purged successfully. Data integrity is not guaranteed.

KILL

Terminate tasks that are running in the specified JVM server. CICS stops the tasks by using the **SET TASK KILL** command. Any threads running in the JVM are stopped. The JVMSERVER resource enters the DISABLED state and all work is ended. However, CICS might be left in an unstable state.

Use this option only after you have used the PURGE and FORCEPURGE options where some tasks could not be purged successfully. System and data integrity are not guaranteed. CICS might terminate abnormally.

PHASEOUT

Shut down the JVM server and its Language Environment enclave. All currently running tasks continue until completion, but no new work is accepted by the JVM server. When all the tasks are finished, the JVMSERVER resource enters the DISABLED state.

PURGE

Purge tasks that are running in the specified JVM server. CICS purges the tasks by using the **SET TASK PURGE** command. Any threads running in the JVM are stopped. CICS purges tasks only when system and data integrity can be maintained. If the JVMSERVER resource remains in the BEING DISABLED state, some tasks could not be purged.

THREADLIMIT(*value*)

Specifies the maximum number of threads that are available to the threaded system JVM. Each thread runs under a T8 TCB. This value can be in the range 1 - 256 threads. If the thread limit exceeds the maximum of 1024 threads allowed for the CICS region, taking into account the other enabled and disabled JVMSERVER resources, CICS adjusts the value accordingly.

value

Specifies the 1 - 8 character name of the JVMSERVER resource.

CEMT SET MQCONN

Change information about the attributes of the connection between CICS and WebSphere MQ, and start or stop the connection.

In the CICS Explorer, the Websphere MQ Connections operations view provides a functional equivalent to this command.

Use the **SET MQCONN** command to change attributes of the currently installed MQCONN resource definition, which defines the connection to WebSphere MQ, and starts and stops the connection.

Because only one MQCONN resource definition can be installed at a time, the name of the MQCONN resource definition is not required on input.

This command does not set the INITQNAME attribute of the MQCONN resource definition, which specifies the name of the initiation queue. If you want to change the MQINI resource definition, you must reinstall the MQCONN resource definition with an appropriate MQINI attribute.

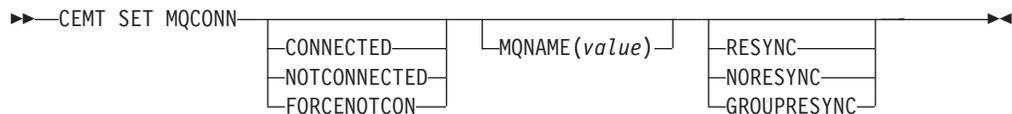
Syntax

Press the Clear key to clear the screen. You can start this transaction in two ways:

- Type CEMT SET MQCONN (or suitable abbreviations for the keywords). The resulting display lists the status, similar to that obtained by using the **CEMT INQUIRE MQCONN** command. You can tab to the highlighted or blank fields and overtype them with the required values.
- Type CEMT SET MQCONN (or suitable abbreviations for the keywords), followed by one or more attribute settings that you want to change.

Typing ? at the beginning of either the first or second line gives a syntax prompt.

SET MQCONN



Conditions: NOTAUTH, NOTFND

Options

CONNECTST (*value*)

Starts or stops the connection between CICS and WebSphere MQ. The possible values are as follows:

CONNECTED

Starts the CICS-WebSphere MQ connection. If the requested queue manager is active, control returns to the application when CICS and WebSphere MQ are connected. If the requested queue manager is not active, CICS connects to WebSphere MQ as soon as the requested queue manager becomes active.

NOTCONNECTED

Stops the CICS-WebSphere MQ connection. Existing transactions are allowed to finish before the connection is stopped. The request is asynchronous in nature; that is, control is returned to the application before the request is complete.

FORCENOTCON

Stops the CICS-WebSphere MQ connection. Any CICS transactions currently using WebSphere MQ are abnormally stopped, and the

connection to WebSphere MQ is stopped. The request is synchronous in nature; that is, control is not returned to the application until the connection is stopped.

MQNAME(*value*)

Specifies the 1 - 4 character name of a WebSphere MQ queue manager or queue-sharing group to which CICS is to connect. CICS attempts to connect to the queue manager or to any active member of the queue-sharing group. You can change MQNAME only when CICS is not connected to WebSphere MQ.

When you specify MQNAME, the queue manager name or queue-sharing group that you specified in the MQNAME attribute of the installed MQCONN resource definition is replaced with the name that you specified on this command. If you want to revert to the original queue manager or queue-sharing group, you must set MQNAME again.

RESYNCMEMBER(*value*)

This option applies only if you have specified a queue-sharing group for the CICS-WebSphere MQ connection. It specifies the strategy that CICS adopts if outstanding units of work are being held for the last queue manager to which CICS was connected from the queue-sharing group. Units of work that are shunted indoubt are not included in this process, because CICS itself cannot resolve those units of work at this time. Resynchronization for those UOWs occurs when CICS has resynchronized with its remote coordinator. The possible values are as follows:

RESYNC

CICS connects to the same queue manager.

NORESINC

CICS makes one attempt to connect to the same queue manager. If that attempt fails, CICS connects to any member of the queue-sharing group and issues a warning about the outstanding units of work.

GROUPRESYNC

CICS connects to any member of the queue-sharing group. The queue manager is chosen by WebSphere MQ and it asks CICS to resolve indoubt units of work on behalf of all eligible queue managers in the queue-sharing group. This function is called *group unit of recovery*. The GROUPRESYNC option can be used only you are when running a release of WebSphere MQ that supports group unit of recovery for CICS and when the GROUPUR attribute has been enabled in the WebSphere MQ queue managers.

When an attempt is made to connect CICS to WebSphere MQ by using an EXEC CICS SET MQCONN CONNECTED command and RESYNCMEMBER(GROUPRESYNC) is set but WebSphere MQ does not support group unit of recovery, or group unit of recovery is not enabled, then WebSphere MQ rejects the connection attempt. The connection attempt results in the SET command failing with INVREQ and RESP2=9 (connection error).

Do not change the settings for RESYNCMEMBER when units of work are outstanding in WebSphere MQ because the units of work cannot then be resolved. A unit of work held in CICS is identified with a resource manager qualifier. When RESYNCMEMBER(GROUPRESYNC) is used, the qualifier is the name of the queue-sharing group, otherwise the qualifier used is the name of the individual queue manager.

Appendix E. New global user exits

CICS Transaction Server for z/OS, Version 4 Release 2 introduces new global user exits (GLUEs).

Exit XISQLCL

Exit XISQLCL is invoked after a function shipping request of a **START NOCHECK** or **START NOCHECK PROTECT** command over IPIC fails because the remote system is not in service, a connection to the remote system cannot be established, or no sessions are immediately available, and your XISQUE exit program specifies that the request is not queued in the issuing region.

Exit-specific parameters

The DSECT, DFHXILDS, is provided for this parameter list.

UEPISQPL

Address of a parameter list that contains the following fields:

UEPPLIST

The address of the parameter list for the command.

UEPQLEN

A halfword binary field containing the number of items currently on the queue.

UEPIPCNM

The eight-byte name of the IPCONN.

UEPTRID

The four-byte identifier of the local transaction name, or blanks if SYSID is specified in the command. Your program can use the transaction manager XPI call, INQUIRE_TRANDEF, to obtain details of the local transaction. .

Return codes

UERCYSYS

Take the system action. This action is determined by the value of the LOCALQ attribute in the local TRANSACTION definition for the remote transaction:

LOCALQ(YES)

The request is queued locally.

LOCALQ(NO)

A SYSIDERR error message is returned to the application program.

UERCQUE

Queue the request locally, overriding the LOCALQ(NO) attribute, if specified.

UERCIGN

Override the LOCALQ(YES) attribute, if specified, and return with a SYSIDERR response.

UERCPURG

Task purged during XPI call.

XPI calls

All can be used.

Important

There is no UERCNORM return code at this exit point, because the exit is invoked after a failure. You must choose whether to take the system default action or to handle the error in some other way.

The sample XISQLCL global user exit program, DFH\$XISL, is provided.

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 want to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM United Kingdom Laboratories, MP151, Hursley Park, Winchester, Hampshire, England, SO21 2JN.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

Adobe and the Adobe logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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.

Readers' Comments — We'd Like to Hear from You

CICS Transaction Server for z/OS
Version 4 Release 2
What's New

Publication No. GC34-7192-02

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 email to: idrctf@uk.ibm.com

If you would like a response from IBM, please fill in the following information:

Name

Address

Company or Organization

Phone No.

Email address



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
United Kingdom
SO21 2JN

Fold and Tape

Please do not staple

Fold and Tape



GC34-7192-02

