

CICS Transaction Server for z/OS
Version 4 Release 1



What's New

CICS Transaction Server for z/OS
Version 4 Release 1



What's New

Note

Before using this information and the product it supports, read the information in "Notices" on page 425.

This edition applies to Version 4 Release 1 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 2009, 2011.**

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

Contents

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

Part 1. Summary of CICS Transaction Server for z/OS, Version 4 Release 1. 1

Chapter 1. Compete	3
Support for event processing	3
Atom feeds from CICS	3
Support for application bundles	4
Improvements to data mappings.	4
Support for Web Services Addressing	5
Support for the IBM SDK for z/OS, Java Technology Edition, Version 6.	6
Support for Service Component Architecture	6
Chapter 2. Comply	9
Improvements in management of resource definitions	9
Support for identity propagation	9
HTTP basic authentication support for Web services	10
Security improvement for DB2 users	10
Interoperability between the Web services assistant and WSRR.	10
Chapter 3. Control	13
The IBM CICS Explorer	13
Extensions to CICS-to-CICS intercommunications over TCP/IP	14
Improvements to system links for IPIC connections	15
Support for IPv6.	15
Support for WebSphere MQ queue-sharing groups	15
Improvements to workload management support.	16
CICS management client interface	16
New system programming commands for managing the CSD	18
The Discovery Library Adapter for CICS	18
Improvements in CICS monitoring	19
Improved CICSplex SM views	19
Chapter 4. Architectural enhancements 21	
Improvements to XML parsing in CICS	21
New option for VTAM persistent sessions support	21
Wild branch diagnosis improvements.	22
Extended z/Architecture MVS linkage support	22
New FLAGSET option on INQUIRE and SET TRACETYPE commands	23
New ENQUEUE_TYPE option on ENQUEUE and DEQUEUE XPI functions	24
New business application manager domain INQUIRE_ACTIVATION XPI function	24

The JVM server runtime environment	24
--	----

Chapter 5. Discontinued functions . . . 27

Removal of support for Version 1.4.2 and Version 1.5.0 of IBM SDK for z/OS, Java Technology Edition	27
Removal of support for the DFHCSDUP MIGRATE command	27

Part 2. Compete 29

Chapter 6. Support for event processing 31

CICS and event processing	31
Terminology	32
Changes to external interfaces	34
Changes to the CICS application programming interface	35
Changes to the JCICS application programming interface	35
Changes to the system programming interface.	35
Changes to resource definitions.	36
Changes to CEMT (master terminal transaction) commands.	36
Changes to CICSplex SM resource tables	36
Changes to CICSplex SM views and menus	37
Changes to supplied transactions	38
Changes to global user exits.	38
Changes to monitoring data	39
Changes to statistics	39
Changes to sample programs	40
Changes to problem determination	40
Changes to security.	41

Chapter 7. Atom feeds from CICS . . . 43

Terminology	43
Atom feeds	43
How CICS supports Atom feeds	44
Changes to external interfaces	46
Changes to system initialization parameters	46
Changes to the CICS application programming interface	46
Changes to the JCICS application programming interface	48
Changes to resource definitions.	48
Changes to the system programming interface.	48
Changes to CEMT (master terminal transaction) commands.	49
Changes to CICSplex SM resource tables	50
Changes to CICSplex SM views and menus	51
Changes to global user exits.	52
Changes to monitoring data	52
Changes to statistics	52
Changes to sample programs	53
Changes to problem determination	53

Changes to security.	55
Chapter 8. Support for application bundles	57
Terminology	57
Application types that support bundles	57
Creating user resource types in bundles	58
The bundle registration interface	59
Changes to external interfaces	59
Changes to resource definitions.	59
Changes to the system programming interface.	59
Changes to CEMT (master terminal transaction) commands.	60
Changes to CICSplex SM resource tables	60
Changes to CICSplex SM views and menus	61
Changes to global user exits.	62
Changes to statistics	62
Changes to problem determination	62
Changes to security.	63
Chapter 9. Improvements to data mappings	65
The CICS XML assistant	65
DFHLS2SC: high-level language to XML schema conversion.	66
DFHSC2LS: XML schema to high-level language conversion.	72
Creating a Web service from multiple containers	80
Changes to the Web services assistant	81
Changes to external interfaces	81
Changes to the CICS application programming interface	81
Changes to the JCICS application programming interface	81
Changes to the system programming interface.	82
Changes to CICSplex SM resource tables	82
Changes to CICSplex SM views and menus	83
Changes to global user exits.	83
Changes to statistics	83
Changes to security.	84
Chapter 10. Support for Web Services Addressing	85
Terminology	86
Changes to external interfaces	86
Changes to the CICS application programming interface	86
Changes to the JCICS application programming interface	86
Changes to monitoring data	87
Changes to problem determination	87
Chapter 11. Support for the IBM SDK for z/OS, Java Technology Edition, Version 6.	89
Changes to CICS externals	89
Changes to installation	89
Changes to JVM profiles	89
Changes to problem determination	91

Chapter 12. Support for Service Component Architecture	93
Terminology	93
Service Component Architecture (SCA)	94
Enhancements to pipeline processing	95
Changes to external interfaces	96
Changes to the CICS application programming interface	97

Part 3. Comply 99

Chapter 13. Improvements in management of resource definitions	101
The definition signature for resource definitions	101
The installation signature for resource definitions	103
Changes to external interfaces	103
Changes to the system programming interface	104
Changes to CEMT.	106
Changes to CICSplex SM resource tables	109
Changes to CICSplex SM views and menus	110
Changes to supplied transactions.	112
Changes to global user exits	113
Changes to CICS utility programs	114
Changes to statistics	115
Changes to sample programs	115
Changes to problem determination	116

Chapter 14. Security updates to monitor RACF Event Notifications (ENF)	117
---	------------

Chapter 15. Support for identity propagation	119
Terminology	119
Changes to external interfaces	120
Changes to system initialization parameters	120
Changes to resource definitions	120
Changes to the system programming interface	120
Changes to CEMT (master terminal transaction) commands	120
Changes to CICSplex SM resource tables	121
Changes to CICSplex SM views and menus	121
Changes to supplied transactions.	121
Changes to global user exits	122
Changes to monitoring data	122
Changes to statistics	122
Changes to sample programs	122
Changes to problem determination	122

Chapter 16. HTTP basic authentication support for Web services	125
Changes to external interfaces	125
Changes to resource definitions	125
Changes to the system programming interface	125
Changes to CEMT (master terminal transaction) commands	125
Changes to CICSplex SM resource tables	125
Changes to CICSplex SM views and menus	125

Changes to statistics	125
Changes to problem determination	126

Chapter 17. Security improvement for DB2 users. 127

Changes to external interfaces	127
Changes to resource definitions	127

Chapter 18. Interoperability between the Web services assistant and WSRR 129

Enhancements to the Web services assistant	129
Changes to external interfaces	131
Changes to problem determination	131

Part 4. Control 133

Chapter 19. The IBM CICS Explorer 135

Changes to installation procedures	136
CICS Explorer planning	136
CICS Explorer installation	136

Chapter 20. Extensions to CICS-to-CICS intercommunications over TCP/IP 137

Changes to external interfaces	137
Changes to resource definitions	137
Changes to the system programming interface	137
Changes to CEMT (master terminal transaction) commands	138
Changes to supplied transactions.	138
Changes to CICSplex SM resource tables	138
Changes to CICSplex SM views and menus	139
Changes to global user exits	139
Changes to monitoring data	139
Changes to statistics	139
Changes to sample programs	139
Changes to problem determination	140

Chapter 21. Improvements to system links for IPIC connections 141

Changes to external interfaces	141
Changes to CICSplex SM resource tables	141
Changes to CICSplex SM views and menus	142

Chapter 22. Support for IPv6. 145

Changes to external interfaces	145
Changes to the CICS application programming interface	145
Changes to the JCICS application programming interface	145
Changes to resource definitions	146
Changes to the system programming interface	146
Changes to CEMT (master terminal transaction) commands	147
Changes to supplied transactions.	148
Changes to CICSplex SM resource tables	148
Changes to CICSplex SM views and menus	149
Changes to global user exits	150

Changes to user-replaceable programs	150
Changes to monitoring data	151
Changes to statistics	151
Changes to sample programs	151
Changes to problem determination	152

Chapter 23. Support for WebSphere MQ queue-sharing groups. 153

Changes to external interfaces	153
Changes to system initialization parameters	153
Changes to resource definitions	154
Changes to the system programming interface	154
Changes to CEMT (master terminal transaction) commands	155
Changes to CICSplex SM resource tables	155
Changes to CICSplex SM views and menus	156
Changes to supplied transactions.	157
Changes to global user exits	158
Changes to statistics	158
Changes to problem determination	158
Changes to security	159

Chapter 24. Improvements to workload management support. 161

Sysplex optimized workload routing overview	161
Support for z/OS workload manager percentile goals	162
Terminology.	163
Changes to external interfaces	163
Changes to CICSplex SM resource tables	163
Changes to CICSplex SM views and menus	165
Changes to problem determination	167
Changes to security	168

Chapter 25. CICS management client interface 169

Setting up the CICS management client interface	170
CICS management client interface DELETE requests	170
CICS management client interface GET requests	171
CICS management client interface POST requests	173
CICS management client interface PUT requests	174
CICS management client interface XML	176
Changes to external interfaces	176
Changes to system initialization parameters	177
Changes to CICSplex SM resource tables	177
Changes to sample programs	177
Changes to problem determination	178

Chapter 26. New system programming commands for managing the CSD . . . 181

Changes to the system programming interface	181
---	-----

Chapter 27. The Discovery Library Adapter for CICS. 185

DLA terminology	186
Running the CICS DLA	186
The CICS DLA package	188
DLA parameters	188

DLA output	192
Security and the DLA	192
DLA problem determination	193

Chapter 28. Improvements in CICS monitoring 195

Changes to external interfaces	195
Changes to the system programming interface	195
Changes to CEMT	196
Changes to CICSplex SM resource tables	197
Changes to CICSplex SM views and menus	197
Changes to supplied transactions	197
Changes to monitoring data	197
Changes to statistics	198
Changes to sample programs	199

Chapter 29. Improved CICSplex SM views 201

Changes to external interfaces	201
Changes to CICSplex SM views	201

Part 5. Architectural enhancements 205

Chapter 30. Improvements to XML parsing in CICS 207

Changes to external interfaces	207
Changes to monitoring data	207
Changes to problem determination	207

Chapter 31. New option for VTAM persistent sessions support 209

Changes to external interfaces	209
Changes to system initialization parameters	209
Changes to the system programming interface	209
Changes to CEMT (master terminal transaction)	
commands	210
Changes to CICSplex SM resource tables	210
Changes to CICSplex SM views and menus	210
Changes to statistics	210

Chapter 32. Wild branch diagnosis improvements 211

Changes to external interfaces	211
Changes to global user exits	211
Changes to sample programs	211

Chapter 33. Extended z/Architecture MVS linkage support 213

Changes to external interfaces	213
Changes to global user exits	213
Changes to the exit programming interface (XPI)	214
Changes to sample programs	214
Changes to problem determination	215

Chapter 34. New FLAGSET option on INQUIRE and SET TRACETYPE commands 217

Chapter 35. New ENQUEUE_TYPE option on ENQUEUE and DEQUEUE XPI functions 219

Chapter 36. New business application manager domain INQUIRE_ACTIVATION XPI function 221

Chapter 37. The JVM server runtime environment 223

Changes to external interfaces	223
Changes to resource definitions	223
Changes to the system programming interface	223
Changes to CEMT (master terminal transaction)	
commands	224
Changes to CICSplex SM resource tables	225
Changes to CICSplex SM views and menus	225
Changes to global user exits	226
Changes to monitoring data	227
Changes to statistics	228
Changes to sample programs	229
Changes to problem determination	230
Changes to security	230

Part 6. Discontinued functions 231

Chapter 38. Removal of support for Version 1.4.2 and Version 1.5.0 of IBM SDK for z/OS, Java Technology Edition 233

Changes to installation procedures	233
Changes to external interfaces	233
Changes to system initialization parameters	233
Changes to options in JVM profiles and the optional JVM properties files	234
Changes to the system programming interface	236
Changes to CEMT (master terminal transaction)	
commands	236
Changes to CICSplex SM resource tables	236
Changes to CICSplex SM views and menus	237
Changes to statistics	237
Changes to security	237

Chapter 39. Removal of support for the DFHCSDUP MIGRATE command 239

Chapter 40. Removal of support for the CICSplex SM WLMLOADCOUNT and WLMLOADTHRSH EYUPARM values 241

Part 7. General information 243

Chapter 41. Threadsafe API and SPI commands 245

Chapter 42. High-level language support 247

Compilers and high-level language versions withdrawn from service 248

Chapter 43. Where is the migration information? 251

Part 8. Publications 253

Chapter 44. The CICS Transaction Server for z/OS library 255

Information provided in HTML only 255
Books available as hardcopy 255
Publications available in Portable Document Format (PDF) 256
 CICS books for CICS Transaction Server for z/OS 256
 CICSplex SM books for CICS Transaction Server for z/OS 257
Licensed publications 257
CICS family books 257

Part 9. Appendixes 259

Appendix A. New application programming commands 261

BIF DIGEST 261
INVOKE SERVICE 262
SIGNAL EVENT 266
TRANSFORM DATATOXML 267
TRANSFORM XMLTODATA 270
WEB ENDBROWSE QUERYPARM 274
WEB READ QUERYPARM 274
WEB READNEXT QUERYPARM 276
WEB STARTBROWSE QUERYPARM 278
WSACONTEXT BUILD 279
WSAEPR CREATE 284
WSACONTEXT DELETE 287
WSACONTEXT GET 287

Appendix B. New RDO resources 293

ATOMSERVICE resources 293
 ATOMSERVICE attributes 293
BUNDLE resources 296
 BUNDLE attributes 297
JVMSEVER resources 298
 JVMSEVER attributes 298
MQCONN resources 300
 MQCONN attributes 301

Appendix C. New system programming commands 305

CREATE ATOMSERVICE 305

CREATE BUNDLE 307
CREATE JVMSEVER 308
CREATE MQCONN 310
CSD ADD 312
CSD ALTER 314
CSD APPEND 316
CSD COPY 317
CSD DEFINE 320
CSD DELETE 324
CSD DISCONNECT 326
CSD ENDBRGROUP 326
CSD ENDBRLIST 327
CSD ENDBRRSRCE 328
CSD GETNEXTGROUP 328
CSD GETNEXTLIST 329
CSD GETNEXTRSRCE 330
CSD INQUIREGROUP 333
CSD INQUIRELIST 334
CSD INQUIRERSRCE 334
CSD INSTALL 337
CSD LOCK 340
CSD REMOVE 341
CSD RENAME 342
CSD STARTBRGROUP 344
CSD STARTBRLIST 346
CSD STARTBRRSRCE 347
CSD UNLOCK 348
CSD USERDEFINE 349
DISCARD ATOMSERVICE 352
DISCARD BUNDLE 353
DISCARD EVENTBINDING 354
DISCARD JVMSEVER 355
DISCARD MQCONN 356
INQUIRE ATOMSERVICE 357
INQUIRE BUNDLE 360
INQUIRE BUNDLEPART 363
INQUIRE CAPTURESPEC 365
INQUIRE EVENTBINDING 367
INQUIRE EVENTPROCESS 370
INQUIRE JVMSEVER 371
INQUIRE MQCONN 374
INQUIRE MQINI 377
INQUIRE XMLTRANSFORM 379
SET ATOMSERVICE 383
SET BUNDLE 384
SET EVENTBINDING 386
SET EVENTPROCESS 387
SET JVMSEVER 388
SET MQCONN 390
SET XMLTRANSFORM 393

Appendix D. New CEMT commands 395

CEMT DISCARD 395
CEMT INQUIRE ATOMSERVICE 400
CEMT INQUIRE BUNDLE 403
CEMT INQUIRE EVENTBINDING 406
CEMT INQUIRE EVENTPROCESS 408
CEMT INQUIRE JVMSEVER 408
CEMT INQUIRE MQCONN 411
CEMT INQUIRE MQINI 414
CEMT SET ATOMSERVICE 416

CEMT SET BUNDLE	417
CEMT SET EVENTBINDING	418
CEMT SET EVENTPROCESS	418
CEMT SET JVMSERVER.	419
CEMT SET MQCONN	421

Appendix E. New global user exits 423

Exit XISQLCL	423
------------------------	-----

Notices 425

Trademarks	426
----------------------	-----

Accessibility. 427

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 1. It gives an overview of the changes to reference information, and points you to the manuals where more detailed reference information is given.

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

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

Who this book is for

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

- Evaluation and planning
- System administration
- Programming
- Customization

What you need to know to understand this book

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

Notes on terminology

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


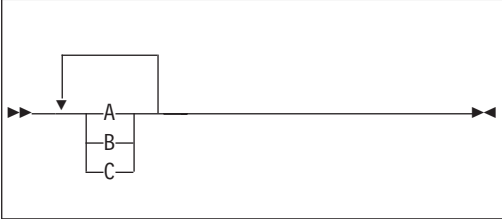

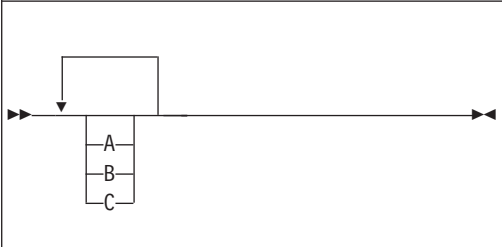
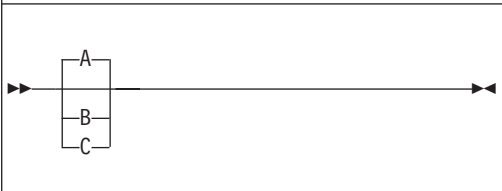
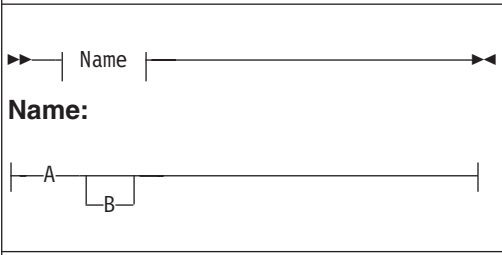

“CICSplex SM” is used for the CICSplex System Manager element of IBM CICS TS.

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

Syntax notation

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

The conventions used in the syntax notation are:

Notation	Explanation
	<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. Summary of CICS Transaction Server for z/OS, Version 4 Release 1

This part contains a brief overview of the major new function in CICS Transaction Server for z/OS, Version 4 Release 1.

Chapter 1. Compete

CICS Transaction Server for z/OS, Version 4 Release 1 delivers a set of capabilities represented in the following themes:

- Compete
- Comply
- Control
- Architectural enhancements

The capabilities represented by the *Compete* theme make it easier to create, extend, and reuse applications quickly, to meet changing business needs.

Support for event processing

You can specify, capture, and emit atomic business events from a CICS application. These business events can be consumed by another CICS application, or placed on a WebSphere MQ queue for consumption in a variety of ways including by a complex event processing engine such as IBM WebSphere Business Events.

A business event is anything that happens that is relevant to your business. An individual stock trade or the placement of an order are both examples of business events. CICS allows business events to be captured in an application and emitted for processing, with typically no change to the emitting CICS application.

The HTTP EP adapter emits events to an HTTP 1.1 compliant server using HTTP POST in XML format for consumption by products such as WebSphere® Business Events and WebSphere Business Monitor without using WebSphere Message Queue as your transport.

WebSphere Message Queue remains the primary method for emitting events from CICS; the HTTP EP adapter can be used as a less robust alternative. You can also use the HTTP EP adapter to send your XML format events to vendor products or any HTTP 1.1 compliant server.

If you want to use the HTTP EP adapter functions in CICS Transaction Server for z/OS, Version 4 Release 1 you must apply APAR PK94205.

Related concepts

Chapter 6, “Support for event processing,” on page 31

Atom feeds from CICS

CICS can now provide Atom feeds using data supplied by CICS resources or by application programs that access other resources. Atom feeds supply Web clients with a series of data items accompanied by metadata for each item in the Atom format. You can also use the HTTP requests defined by the Atom Publishing Protocol to edit a CICS resource that you have made available as a collection of Atom entries.

Related concepts

Chapter 7, “Atom feeds from CICS,” on page 43

Support for application bundles

You can now deploy applications into CICS using bundles. A *bundle* is a collection of CICS resources, artifacts, references, and a manifest that represent an application. Use bundles to more easily manage the availability of an application and the life-cycle of its resources.

Bundles are created by an application developer using a tool such as Rational® Developer for System z®, the CICS XML assistant or the IBM CICS Explorer™. A bundle contains only the resources that are required by the application. The system resources that the application requires might be defined as prerequisites, but they are not included in the bundle. This separation means that you can install the same application into multiple CICS regions without having to repackage or redeploy the bundle.

A bundle is defined in CICS using a BUNDLE resource. For details about how to define this resource and for more information about the format of its contents, see the *CICS Resource Definition Guide*.

The BUNDLE resource is different from an RDO group because it maintains a relationship with all the resources after they are installed, so that you can manage all the related resources as a single entity; for example, if you disable a BUNDLE resource because you want to stop an application from running, CICS disables all of the related application resources for you. To view the contents of a bundle and the state of its resources, use the IBM CICS Explorer.

Related concepts

Chapter 8, “Support for application bundles,” on page 57

Improvements to data mappings

You can now modernize your CICS applications to use XML in channel-based interfaces. Using new utilities and API commands, you can map application binary data to XML and transform it during runtime processing.

The advantage of using this approach to transform application data to and from XML is that CICS goes beyond the capabilities offered by an XML parser. CICS can interpret the XML and perform record-based conversions of application data. Therefore, it is easier and faster for you to create applications that work with XML using this approach.

CICS supports a number of high-level languages and provides an XML assistant to map how the data is transformed during runtime processing. CICS uses the same technology for mapping application data to XML in SOAP messages, as part of the Web services support.

The CICS XML assistant is a supplied utility that helps you to create the required mappings to transform application binary data to XML or transform XML to application binary data. The XML assistant generates an XML binding and optionally a schema in a bundle directory on z/OS UNIX. By creating and installing a corresponding BUNDLE resource, CICS dynamically creates an XMLTRANSFORM resource that defines where the XML binding and schema are located.

If you are using Web services, this support provides a number of benefits:

- You can now process SOAP headers and SOAP faults programmatically.
- You can now create a Web service application that builds a SOAP message from multiple containers on a channel.

Related concepts

Chapter 9, “Improvements to data mappings,” on page 65

Support for Web Services Addressing

Web Services Addressing (WS-Addressing) is a transport-neutral mechanism for passing messaging information between Web services, so that Web services using different protocols can interoperate.

CICS ensures that your existing Web service applications can accept requests from Web services that use WS-Addressing. You can also create new Web services that use endpoint references and message addressing properties in SOAP messages.

WS-Addressing adds addressing information, in the form of Message Addressing Properties (MAPs), to SOAP message headers. MAPs include messaging information, such as a unique message ID and endpoint references that detail where the message came from, where the message is going to, and where reply or fault messages are to be sent. An endpoint reference (EPR) is a specific type of MAP, which includes the destination address of the message, optional reference parameters for use by the application, and optional metadata.

You can use new API commands to create and manipulate addressing contexts. Addressing contexts are used to store WS-Addressing MAPs before and after SOAP requests and responses are performed.

CICS includes the following features to support WS-Addressing:

- Your Web service requester and provider applications can interact with other services that are using WS-Addressing without requiring you to redeploy them. A new message handler, the addressing message handler DFHWSADH, in the pipeline routes messages that contain WS-Addressing information to the specified Web service.
- You can write an application that uses the WS-Addressing API commands to create an endpoint reference and to create, update, delete, and query an addressing context.
- You can route response messages to endpoints other than the requester endpoint; for example, you can route fault messages to a dedicated fault handler.
- You can pass reference parameters to applications as part of the MAPs in the SOAP header.

By default, CICS supports the recommendation specifications:

- W3C WS-Addressing 1.0 - Core
- W3C WS-Addressing 1.0 - SOAP Binding
- W3C WS-Addressing 1.0 - Metadata

These specifications are identified by the <http://www.w3.org/2005/08/addressing> namespace. Unless otherwise stated, WS-Addressing semantics that are described in this documentation refer to the recommendation specifications.

For interoperability, CICS also supports the submission specification:

- W3C WS-Addressing- Submission

This specification is identified by the <http://schemas.xmlsoap.org/ws/2004/08/addressing> namespace. Use the submission specification only if you must interoperate with a client or Web service provider that implements the submission specification.

Related concepts

Chapter 10, “Support for Web Services Addressing,” on page 85

Support for the IBM SDK for z/OS, Java Technology Edition, Version 6

CICS TS 4.1 supports the JVM provided by the 31-bit version of the IBM SDK for z/OS, Java Technology Edition, Version 6. With Java 6, you continue to benefit from the new Java language features, improved garbage collection, and simpler class sharing, first introduced in Java 5.

CICS TS 4.1 does not support the 64-bit version of the IBM SDK for z/OS, Version 6.

Related concepts

Chapter 11, “Support for the IBM SDK for z/OS, Java Technology Edition, Version 6,” on page 89

Support for Service Component Architecture

CICS provides support for applications that conform to the Service Component Architecture (SCA) specifications. SCA supports rapid deployment of new applications to meet changing business requirements by promoting the reuse of existing application assets in a component model.

In SCA, applications are composed of services, where each service is described using one or more components. These components are assembled together into a *composite application* that can participate in a service-oriented architecture (SOA) as part of a business service. A *business service* is a service that is aligned with business processes and models rather than a technical implementation.

The SCA component model is flexible, so you can reuse the same components in different composite applications without needing to change the underlying business logic.

To create a composite application, use the SCA tooling in Rational Developer for System z. The SCA tooling deploys composite applications to CICS as bundles and can be enabled and disabled using the BUNDLE resource.

You can create two types of service from your CICS applications:

Channel-based services

These services use the Service Component Architecture (SCA) support to describe applications as components. The interface for these services is a channel. Channel-based services can be called only by other CICS applications using the **INVOKE SERVICE** API command.

XML-based services

These services are typically Web service provider or requester applications that use XML to interface with other applications and use a binding to transform the data. Web service applications can also be described as components using SCA. The interface for these services is XML with a

binding to transform the data. XML-based services can be called by other CICS applications using the **INVOKE SERVICE** API command or by an external client.

An XML-based service can also be an application that uses the **TRANSFORM** API commands to map application data to and from XML. The XML assistant uses a language structure or XML schema to generate the XML binding and also create a bundle.

Both types of service use the pipeline support in CICS.

Related concepts

Chapter 12, "Support for Service Component Architecture," on page 93

Chapter 2. Comply

CICS Transaction Server for z/OS, Version 4 Release 1 delivers a set of capabilities represented in the following themes:

- Compete
- Comply
- Control
- Architectural enhancements

The capabilities represented by the *Comply* theme help ensure and demonstrate effective management control over business applications and IT facilities.

Improvements in management of resource definitions

The resource signature, which combines the installation and definition signatures, provides specific information for resource definitions that were installed or changed in CICS TS 4.1 or later. You can use these improved details to detect resource modifications for auditing, tracking, or for fixing problems.

The installation signature captures details as to when, how, and by whom the resource was installed. The definition signature captures details in the CICS system definition data set or the CICSplex SM EYUDREP data repository, as to when, how, and by whom the resource was defined and the last change made to the resource definition.

All resource types support the definition signature. The resource types that support the installation signature are: ATOMSERVICE, BUNDLE, CONNECTION, CORBASERVER, DB2CONN, DB2ENTRY, DB2TRAN, DJAR, DOCTEMPLATE, ENQMODEL, EVENTBINDING, FILE, IPCONN, JOURNALMODEL, JVMSERVER, LIBRARY, MQCONN, MQINI, PIPELINE, PROFILE, PROCESSTYPE, PROGRAM, REQUESTMODEL, TCPIPSERVICE, TDQUEUE, TRANCLASS, TRANSACTION, TSMODEL, URIMAP, WEBSERVICE, and XMLTRANSFORM.

Related concepts

Chapter 13, “Improvements in management of resource definitions,” on page 101

Support for identity propagation

If you are using z/OS, Version 1 Release 11, you can now send a user's security identity information (the distributed identity) from a client system across a network, preserving the distributed identity for use during CICS authorization and for subsequent auditing purposes.

Identity propagation depends on the z/OS Identity Propagation function that is provided in z/OS, Version 1 Release 11 and three enabling Program Temporary Fixes (PTF). For more information, see APARs PK95579, PM01622, PK83741, and PK98426. Identity propagation provides improved accountability and a new means of authorizing requests by associating the security information in the distributed environment with the security information used on the server.

CICS provides support for identity propagation in the following situations:

- Inbound requests to CICS from WebSphere Application Server using the CICS ECI resource adapters over an IPIC connection.
- Using a WS-Security Header element in a Web service request. Routed inbound Web service requests do not support identity propagation.
- Using IPIC and MRO connections between CICS systems. The distributed identity is used by CICS only if it is passed to the MRO or IPIC connection from a participating task.
- Function shipping of **START** commands and local **START** commands do not inherit the distributed identity.

Related concepts

Chapter 15, “Support for identity propagation,” on page 119

HTTP basic authentication support for Web services

You can now provide user ID and password information across an HTTP protocol, when a Web service requester sends an outbound request to a Web service provider that requires authentication. This type of authentication is called HTTP basic authentication. You need a URIMAP resource specifying AUTHENTICATE(BASIC) and an enabled XWBAUTH global user exit to perform HTTP basic authentication.

Related concepts

Chapter 16, “HTTP basic authentication support for Web services,” on page 125

Security improvement for DB2 users

In CICS Transaction Server for z/OS, Version 4 Release 1, the CICS interface with DB2 provides more flexibility when you use DB2 and RACF. CICS now passes the address of the access control environment element (ACEE) when AUTHTYPE(SIGN) is specified for a DB2CONN or a DB2ENTRY resource, and the SIGNID attribute specifies the CICS region user ID.

With this change, DB2 can use RACF security when you use the CICS region user ID to control CICS's access to DB2.

In previous releases, the ACEE is passed to DB2 only when AUTHTYPE(USERID) or AUTHTYPE(GROUP) is specified.

Related concepts

Chapter 17, “Security improvement for DB2 users,” on page 127

Interoperability between the Web services assistant and WSRR

The CICS Web services assistant now includes interoperability support for the IBM WebSphere Service Registry and Repository (WSRR). WSRR helps you to manage and govern your services and processes. Use WSRR to find the Web services that you are requesting more quickly and enforce version control of the Web services you are providing.

Both DFHLS2WS and DFHWS2LS include parameters to interoperate with WSRR. DFHLS2WS also includes an optional parameter so that you can add your own customized metadata to the WSDL document in WSRR.

If you want the Web services assistant to communicate securely with WSRR, you can use secure socket level (SSL) encryption. Both DFHLS2WS and DFHWS2LS include parameters for using SSL encryption.

Related concepts

Chapter 18, “Interoperability between the Web services assistant and WSRR,” on page 129

Chapter 3. Control

CICS Transaction Server for z/OS, Version 4 Release 1 delivers a set of capabilities represented in the following themes:

- Compete
- Comply
- Control
- Architectural enhancements

The capabilities represented by the *Control* theme help IT staff to perform their tasks more effectively, while assuming a mixture of skill levels.

The IBM CICS Explorer

The CICS Explorer is a system management tool that is designed to provide a simple, easy to use way of managing one or more CICS systems. Based on an Eclipse platform, it provides functions to view and manage CICS Transaction Server regions and a platform for the integration of CICS Tools.

Eclipse is a platform for building and deploying client applications in which the majority of data manipulation is done by the client application rather than the server. This platform is known as a Rich Client Platform (RCP). The platform provides the ability to deploy native graphical user interface (GUI) applications to a variety of desktop operating systems. The CICS Explorer is an Eclipse RCP based GUI application that runs on the Windows and Linux operating systems.

Today, you manage CICS regions with a variety of stand-alone graphical and non-graphical user interfaces, and with no common way of performing tasks. The CICS Explorer provides a base set of CICSplex SM functions to manage CICS regions, and the CICS event binding editor used by business analysts to define event specifications and by application analysts to define event specifications and dispatcher information. In addition, the CICS Explorer provides a platform on which CICS tools that are integrated in the Eclipse RCP, can perform tasks and present the information in a common way.

CICS Explorer connects to either a CPSM WUI server or a single CICS region. When connected to a CPSM WUI server, CICS Explorer operations views provide a single-system image of all the CICS resources within a CICSplex. When the CICS Explorer is connected to a CICS Transaction Server for z/OS version 3 system, the resources can only be browsed, however using the operations views and CICS Transaction Server for z/OS version 4, you can perform the following actions:

- Enable and disable resources
- Open and close resources
- Acquire and release resources
- Place resources in and out of service
- Purge tasks associated with a resource
- Discard resource definitions from a CICS system where they are installed

The CICS Explorer administration views display details of the CICS and CICSplex SM resource definitions, and with CICS Transaction Server for z/OS version 4 you

can update the resource definitions using the CICS Explorer resource editor. Using the administration views and the resource editor, you can perform the following actions:

- Display resource definitions
- Create and update resource definitions
- Install or remove resource definitions

The CICS Explorer resource editor displays the resource definition attributes in two or more pages in the editor. One page, the Attributes page, displays the attributes in tabular form grouped together in the same way as they appear in CEDA or the CPSM WUI to facilitate the move to CICS Explorer for experienced CICS users. The other page, or pages, display the attributes logically grouped together, but each attribute has more detail provided to simplify the update process for less-experienced CICS users. Every attribute benefits from field-level verification, where the entry is validated in real time. Errors are identified by the **Error** icon, which identifies the field in error and the page on which the field appears.

The CICS Explorer contains context and more detailed help for the resource views and help on using the CICS Explorer and the basic Eclipse functions. Detailed help for CICS and CICSplex SM is found in the CICS Transaction Server for z/OS Information Center. Detailed help for Eclipse is found at the Eclipse homepage.

Related concepts

Chapter 19, “The IBM CICS Explorer,” on page 135

Extensions to CICS-to-CICS intercommunications over TCP/IP

You can now use a TCP/IP network to route transactions and send START and CANCEL commands between CICS TS regions.

IP interconnectivity (IPIC) was introduced in CICS TS for z/OS, Version 3.2, providing an alternative to intersystem communications over SNA (ISC over SNA).

IPIC connections offer similar capabilities and qualities of service to those provided by ISC over SNA, including transactional syncpointing, security controls, encryption, and support for the exchange of either COMMAREAs or channels and containers. Coupled with the latest System z TCP/IP networking enhancements, IPIC enables CICS intercommunications to fully exploit modern high capacity and adaptable TCP/IP communication frameworks.

IPIC supports these intercommunication functions and releases:

- Distributed program link (DPL) calls between CICS TS 3.2, or later regions
- Distributed program link (DPL) calls between CICS TS and TXSeries Version 7.1, or later
- Asynchronous processing of **EXEC CICS START**, **START CHANNEL**, and **CANCEL** commands, between CICS TS 4.1, or later regions
- Traditional transaction routing of 3270 terminals, where the terminal-owning region (TOR) is uniquely identified by an APPLID between CICS TS 4.1, or later regions
- ECI requests from CICS Transaction Gateway Version 7.1 or later

Up to two send and two receive sockets are now available for each IPIC connection, replacing the single send and single receive socket. The increase in the number of sockets can improve your transaction throughput.

Related concepts

Chapter 20, “Extensions to CICS-to-CICS intercommunications over TCP/IP,” on page 137

Improvements to system links for IPIC connections

You can now define and install IPIC connections between CICS regions by using SYSLINK resources in CICSplex SM.

A SYSLINK resource defines the connection between two CICS regions. When you install a SYSLINK resource, CICSplex SM installs the defined resources for you in the correct CICS regions to create the connection.

You can also manage all of your SYSLINK resources together or filter them by connection type.

Related concepts

Chapter 21, “Improvements to system links for IPIC connections,” on page 141

Support for IPv6

With CICS TS 4.1, you can now operate in an Internet Protocol version 6 (IPv6) network without the need for address translation. CICS still operates in an IPv4 network as before.

IPv6 is the protocol designed by the IETF to replace the current version Internet Protocol, IP Version 4 (IPv4). The IPv6 format has a much larger address space than IPv4 (128 bits compared to 32 bits), which provides greater flexibility in allocating addresses and for routing traffic.

CICS operating in a dual-mode environment uses both IPv4 and IPv6 networks and always attempts to communicate using IPv6 before using the IPv4 network. A single-mode environment operates in an IPv4 network only. You need a minimum level of CICS TS 4.1 to communicate using IPv6. The CICS TS 4.1 region must be running in a dual-mode (IPv4 and IPv6) environment and the client or server that CICS is communicating with must also be running in a dual-mode environment.

Related concepts

Chapter 22, “Support for IPv6,” on page 145

Support for WebSphere MQ queue-sharing groups

You can now specify a WebSphere MQ queue-sharing group for the CICS-WebSphere MQ connection, so CICS uses any eligible queue manager in the group when it reconnects to WebSphere MQ, rather than waiting for a single queue manager. Queue-sharing groups increase reliability when you reconnect to WebSphere MQ, and help you standardize this aspect of CICS setup across CICS regions and z/OS images.

Instead of defining default settings for the CICS-WebSphere MQ connection in the DFHMQRPM operand of an INITPARM system initialization parameter, you must now use the new MQCONN resource definition. You can use the MQCONN resource definition to specify a queue-sharing group, or you can use it to specify the name of a single queue manager.

If you have specified a queue-sharing group for the connection, you can select appropriate resynchronization actions for CICS using the RESYNCMEMBER attribute of the MQCONN resource definition. Resynchronization works in the same way as it does for the group attach function for DB2®. Resynchronization takes place when the connection to WebSphere MQ is lost and CICS is holding outstanding units of work for the last queue manager. You can choose whether CICS waits to reconnect to the same queue manager, or whether CICS makes one attempt to reconnect to the same queue manager, but if that attempt fails, connects to a different eligible queue manager in the group. A queue manager is eligible for connection to a CICS region if it is currently active on the same LPAR as the CICS region.

You can upgrade to use the new EXEC CICS and CEMT commands or CICSplex SM to start and stop the CICS-WebSphere MQ connection and change all the attributes of the connection. Alternatively, you can continue to use the existing methods of operating the CICS-WebSphere MQ adapter to initiate and manage connections between CICS and WebSphere MQ. You can use the CKQC transaction from the CICS-WebSphere MQ adapter control panels, or call it from the CICS command line or from a CICS application.

Related concepts

Chapter 23, “Support for WebSphere MQ queue-sharing groups,” on page 153

Improvements to workload management support

Dynamic workload management takes advantage of the Coupling Facility to store current region status information posted directly from CICS. The stored region status is then used to make more efficient routing decisions. CICSplex SM now supports percentile goals when using goal algorithms.

Dynamic workload management provides these benefits:

- Additional WUI views that can assist with problem determination in route selection
- Exploitation of the z/OS coupling facility to improve cross-sysplex routing of distributed workloads
- Dynamic tuning of CICSplex SM workload manager resource consumption in the coupling facility
- Improved recognition of CICS region status for more efficient WLM routing decisions
- Implementation of the new functions requiring minimal configuration effort

When you upgrade from a previous release of CICS TS, existing workloads can be migrated without interrupting them, or modifying their definitions.

Related concepts

Chapter 24, “Improvements to workload management support,” on page 161

CICS management client interface

The CICS management client interface (CMCI) is a system management application programming interface, designed using Representational State Transfer (RESTful) principles, for use by HTTP client applications including IBM CICS Explorer. You can use this interface to develop HTTP client applications that manage installed and definitional CICS and CICSplex SM resources on CICS regions being managed by CICSplex SM or CICS resources on stand-alone CICS regions.

The client initiates an HTTP request to the CICS management client interface. If the interface determines that the request is valid, it constructs a CICSplex SM API command or, in the case of a stand-alone CICS region, a CICS system command. After running the command the CMCI creates an HTTP response. If the request is successful, this takes the form of an HTTP 200 (OK) response and an XML feed containing a result set, which it passes back to the client. If the request is not successful, the response consists of a non-OK HTTP response code with details of the failure.

The format for CICS management client interface HTTP requests and responses is based on the HTTP/1.1 protocol. See the *CICS Internet Guide* for more information about this protocol.

A CICS management client interface request takes the form of an HTTP header followed by a URI (Universal Resource Identifier) and, where appropriate, an XML body containing details of any changes to be made to CICS or CICSplex SM resources.

The header incorporates one of the following HTTP methods:

DELETE

Removes resources from the CICSplex SM data repository, removes resources from the CSD, or discards installed resources.

GET Retrieves information about resources on the CICSplex SM data repository, retrieves information about resources on the CSD, or retrieves information about installed resources.

POST Creates resources on the CICSplex SM data repository or resources in the CSD.

PUT Updates existing resources in the CICSplex SM data repository, updates existing resources in the CSD, or sets attributes and performs actions on installed resources. Also performs actions on CICSplex SM and CSD resources.

The URI includes the name of a CICS or CICSplex SM resource, and specifies a series of parameters that refine the scope and nature of the query to identify one or more instances of the specified resource. In a GET request, the URI also specifies whether the API retains or discards a set of results. If the API retains the results, a new request can act on the retained results without having to repeat the retrieval operation. You can also use subsequent requests to page through the retained results selecting one or more records at a time.

POST and PUT requests include an XML body. In a PUT request the body contains either details of the changes to be made to resource attributes, or the action to be performed on the targeted resources. In a POST request, the body incorporates the attribute values you wish to give to the new resource instance.

GET and DELETE requests do not require an XML body. If additional parameters are required for a DELETE request, those parameters must be included in the URI and can optionally be added to the XML body. For example; if you are deleting a CSD resource definition you must include `PARAMETER=CSDGROUP(csdgrp)` in the URI, and you can optionally add `<parameter name="CSD"/>` to the XML body.

You can use CMCI either in a CICSplex SM environment or in a stand-alone CICS region (SMSS).

If you use CMCI with CICSplex SM, you can manage definitional, operational, and CSD resources in all of the regions managed by CICSplex SM.

If you use CMCI in a stand-alone CICS region, you can manage only the operational and CSD resources associated with that region. If you use CMCI in a stand-alone CICS region, the context is specified as the application ID of that CICS region.

Related concepts

Chapter 25, “CICS management client interface,” on page 169

New system programming commands for managing the CSD

You can now manage CICS resource definitions in the CSD using a set of new CICS system programming interface (SPI) commands. These commands give you a command-level alternative to the CEDA transaction and the DFHCSDUP batch utility for updating the CSD.

Like other SPI commands, you can use the CSD commands in programs written in any CICS-supported language. They are recognized by the command interpreter (CECI), the execution diagnostic facility (EDF), and the CICS translator. The CICS management client interface uses these commands when making requests to a CICS region that is not being managed by CICSplex SM.

Related concepts

Chapter 26, “New system programming commands for managing the CSD,” on page 181

The Discovery Library Adapter for CICS

The Discovery Library Adapter (DLA) for CICS is a utility that collects runtime information about CICS and CICSplex SM resources. It generates XML files that can be used by a discovery library reader to track these resources and their interdependencies across your enterprise.

The CICS DLA uses CICSplex SM services to generate XML files that conform to the Discovery Library IdML XML schema and Common Data Model (CDM). The XML files called discovery library books can be loaded by several products including Tivoli® Change and Configuration Management Database (CCMDB) and Tivoli Business Service Management (TBSM).

The CICS DLA operates against all the CICSplexes that are associated with a specified CMAS. The specified CMAS must be running at the current release level; however, the CICS regions discovered by the DLA can be at earlier release levels, as allowed by the cross-release support rules for CICSplex SM.

To use the CICS DLA you run EYUJXDD0 as a batch job on a z/OS system. Running the DLA creates IdML discovery library books in a PDS data set. The DLA also writes control files and debug information to this data set.

When you run the DLA, it can generate three types of discovery library books:

- One CTSPLEX discovery library book containing summary information about discovered CICSplexes
- One or more CICSplex discovery library books containing information about the CICS system topology in a CICSplex and also the CICSplex SM logical scopes and their associated resources.

- One or more CICSREGION discovery library books containing information about CICS resources associated with a CICS region

The DLA also generates FTP PUT and RENAME statements to facilitate the transfer of the discovery library books to the discovery library file store (DLFS) located on your local system or accessible through a network connection. You control whether the FTP transfer is carried out as part of the DLA run or is done later as an independent operation.

From the DLFS you can load the discovery library books into CCMDB or TBSM using FTP or another transfer mechanism.

You control the output of the DLA using input parameters. These parameters control the range and depth of the discovery and specify which discovery library books are created. The parameters also set general options about the operating environment, including the name of the CMAS under which the discovery takes place. You can supply these parameters in startup JCL or in an input file referred to by the parameter DD statement in the JCL.

You can run the DLA in either create mode or refresh mode:

- Create mode is the default. In this mode, the DLA produces a snapshot of your CICSplexes. The DLA produces discovery library books that might not contain information about all affected resources because some are not currently active.
- In refresh mode, the discovery library books contain everything known about the environment. The reader treats as obsolete anything that was previously loaded from a similar discovery library book and is not in this discovery library book. CCMDB and TBSM support refresh mode by deleting these obsolete resources. Refresh mode can be useful when there has been a significant configuration change but you should use it with caution because of the large amount of processing that might be involved.

Related concepts

The Discovery Library Adapter for CICS

Improvements in CICS monitoring

New performance data metrics improves the performance reporting and analysis of Web and Web service applications. Use the new transaction resource class monitoring data for distributed program link requests to improve the work load management of DPL applications. Also, to help reduce the volume of monitoring output produced, the default now is to compress monitor records.

You can set the monitoring options that limit the number of distributed program link requests, files, and temporary storage queues for which CICS is to perform transaction resource monitoring. Changes to the DPLLIMIT, FILELIMIT, and TSQUEUELIMIT values are made dynamically to the running CICS system.

Related concepts

Chapter 28, "Improvements in CICS monitoring," on page 195

Improved CICSplex SM views

Over 70 CICSplex SM Web User Interface operations view sets have been redesigned to improve their usability and function. The improvements include the reordering of fields into more logical groupings and the addition of new detailed views to make common tasks easier and faster.

The redesigned views include the following improvements:

- Additional hyperlinks to speed up the navigation between often-used views
- New detailed views that display logical groupings of fields on a single screen
- Reordered fields and columns that make it easier to find associated attributes
- Additional filters to quickly display the resources and objects that you want

New techniques to make better use of screen space are introduced. For example, in the Dynamic storage areas (DSA) view (EYUSTARTCICSSTOR.DETAILED1), DSA information is now displayed in a matrix-like arrangement so that you can view more information on one screen.

These changes do not involve the addition or removal of any resource table attributes.

Related concepts

Chapter 29, “Improved CICSplex SM views,” on page 201

Chapter 4. Architectural enhancements

CICS Transaction Server for z/OS, Version 4 Release 1 delivers a set of capabilities represented in the following themes:

- Compete
- Comply
- Control
- Architectural enhancements

The capabilities represented by the *Architectural enhancements* theme provide enhancements, optimizations, and improved performance for CICS workloads.

Improvements to XML parsing in CICS

The performance of XML parsing in CICS has improved with the introduction of the IBM z/OS XML System Services (XMLSS) parser, which can be accessed directly from CICS. The XMLSS parser also allows XML parsing to be offloaded to a System z Application Assist Processor (zAAP), which can reduce the cost of transactions.

The XMLSS parser uses above-the-bar storage, leaving more below-the-bar storage available for user programs.

For more information about zAAP, see the introduction to *System z Application Assist Processor (zAAP)* on the IBM website: <http://www-03.ibm.com/systems/z/advantages/zaap/index.html>.

Related concepts

Chapter 30, “Improvements to XML parsing in CICS,” on page 207

New option for VTAM persistent sessions support

You can increase the number of CICS regions in an LPAR by removing VTAM persistent sessions support where it is not required, such as in a CICS region that is used only for development or testing. You can now disable VTAM persistent sessions support for a CICS region using the new value NOPS for the **PSTYPE** system initialization parameter.

If you have a large number of CICS regions in the same LPAR (around 500), with persistent sessions support available for all the regions, you might reach a z/OS limit on the maximum number of data spaces and be unable to add any more CICS regions. In this situation, when you attempt to start further CICS regions, you see messages IST967I and DFHSI1572, stating that the ALESERV ADD request has failed and the VTAM[®] ACB cannot be opened. However, a region without persistent sessions support does not use a data space and so does not count towards the limit. To obtain a greater number of CICS regions in the LPAR:

1. Identify existing regions that can run without persistent sessions support.
2. Change the **PSTYPE** system initialization parameter for those regions to specify NOPS, and specify a zero value for the **PSDINT** system initialization parameter.
3. Cold start the regions to implement the change.

You can then start further CICS regions with or without persistent sessions support as appropriate, provided that you do not exceed the limit for the number of regions that do have persistent sessions support.

Related concepts

Chapter 31, “New option for VTAM persistent sessions support,” on page 209

Wild branch diagnosis improvements

CICS TS 4.1 supports the Breaking Event Address Register (BEAR). This register captures the address of the instruction that generated the last successful branch before the error occurred in the program. This information helps when investigating wild branch problems in programs. The Breaking Event Address Register is held in the system diagnostic work area.

Wild branch diagnosis is to help identify where a program has taken a branch to a random address because of a computational error. When a program fails, you can use the address of the last successful instruction in the BEAR and the program location in the PSW (program status word) to help identify the sequence of events in the program. Using this information can speed up and improve the quality of problem diagnosis.

Related concepts

Chapter 32, “Wild branch diagnosis improvements,” on page 211

Extended z/Architecture MVS linkage support

CICS TS 4.1 supports the z/Architecture MVS linkage support for the access registers, all the floating point registers, and the 64-bit general registers. This improved linkage support provides execution context for application programs, that follow the linkage conventions, to ensure they can successfully pass control from one to the other while preserving the register contents and parameter data that they need to run successfully. System and transaction dumps display all of the registers.

However, if global user exits or task-related user exits modify the contents of the registers, the registers must be saved by the exit before they are altered and then restored to their original values before the exit passes control to CICS.

The new release sensitive XPI call, RELSENSCALL is an alternative to the CALL XPI call. Using the RELSENSCALL XPI call and assembling against the CICS TS 4.1 libraries, XPI calls execute successfully on all currently supported CICS releases. The release sensitive XPI call alternative applies to all XPI commands.

If a global user exit or task-related user exit is assembled using CICS libraries from a release earlier than CICS TS 4.1 and makes an XPI call on a CICS TS 4.1 system the exit will fail. An error message is issued and the transaction that called the exit might end. You must reassemble all global user exits and task-related user exits, against the CICS TS 4.1 libraries, if they contain *any* XPI calls.

Related concepts

Chapter 33, "Extended z/Architecture MVS linkage support," on page 213

New FLAGSET option on INQUIRE and SET TRACETYPE commands

The new FLAGSET option on the INQUIRE and SET TRACETYPE commands, with its CVDA, replaces the STANDARD and SPECIAL options. Existing programs that use this command will work unchanged. There is no functional change to the command because the CVDA values include STANDARD and SPECIAL.

Changed command, EXEC CICS INQUIRE TRACETYPE

A new FLAGSET option has replaced the SPECIAL and STANDARD options in the **EXEC CICS INQUIRE TRACETYPE** command and a new INVREQ condition has been added.

FLAGSET (*cvda*)

Indicates whether the standard or special flags for the specified component are to be returned. CVDA values are as follows:

SPECIAL

Indicates that CICS returns the trace levels for special tracing.

STANDARD

Indicates that CICS returns the trace levels for standard tracing.

INVREQ

RESP2 values:

1 An incorrect value was specified for FLAGSET.

Changed command, EXEC CICS SET TRACETYPE

A new FLAGSET option has replaced the SPECIAL and STANDARD options in the **EXEC CICS SET TRACETYPE** command and a new INVREQ condition has been added.

FLAGSET (*cvda*)

Indicates whether the standard or special flags, for the specified component, are to be set. CVDA values are as follows:

SPECIAL

Specifies that you want to set levels for special tracing, for the components listed.

STANDARD

Specifies that you want to set levels for standard tracing, for the components listed.

INVREQ

RESP2 values:

1 An incorrect value was specified for FLAGSET.

Related concepts

Chapter 34, “New FLAGSET option on INQUIRE and SET TRACETYPE commands,” on page 217

New ENQUEUE_TYPE option on ENQUEUE and DEQUEUE XPI functions

You can use the new optional ENQUEUE_TYPE parameter, on the XPI ENQUEUE command, to enqueue on the same resource being enqueued on by EXEC CICS ENQ or vice versa. Applications can synchronize processes using EXEC CICS and EXEC XPI commands.

If the ENQUEUE_TYPE option is not specified, XPI is assumed. Existing exits will work unchanged.

Changed functions, ENQUEUE and DEQUEUE

A new ENQUEUE_TYPE option has been added to the ENQUEUE and DEQUEUE functions.

ENQUEUE_TYPE (XPI | EXECSTRN | EXECADDR)

Specifies the type of resource being enqueued on. The XPI option specifies the typical DFHNQEDX behavior. The resource pool used is exclusive to XPI and cannot be accessed by the CICS API. Use EXECSTRN or EXECADDR to indicate that ENQUEUE_NAME1 specifies the name or address respectively of an enqueue resource, located in the same namespace, as the one being used by EXEC CICS ENQ.

Related concepts

Chapter 35, “New ENQUEUE_TYPE option on ENQUEUE and DEQUEUE XPI functions,” on page 219

New business application manager domain INQUIRE_ACTIVATION XPI function

You can use the new INQUIRE_ACTIVATION XPI function to obtain the activity name and process type for the business transaction activity of the current transaction.

New function, INQUIRE_ACTIVATION

The new INQUIRE_ACTIVATION function is provided on the DFHBABRX macro call.

Related concepts

Chapter 36, “New business application manager domain INQUIRE_ACTIVATION XPI function,” on page 221

The JVM server runtime environment

CICS includes a new runtime environment to perform Java processing. The runtime environment is called a *JVM server* and provides a mechanism for CICS to use the same JVM for multiple tasks concurrently.

The JVM server requires a Language Environment enclave and a pool of T8 TCBs. Each task uses a T8 TCB to run as a pthread in the JVM. The JVM server is

represented in CICS by a JVMSERVER resource . The JVMSERVER resource defines the runtime options for the JVM, including the location of the JVM profile and the maximum number of threads that the JVM supports.

The JVM server is not available for use by Java applications. The JVM server is an architectural enhancement that is provided for CICS to perform system processing only, if required by future developments.

Related concepts

Chapter 37, “The JVM server runtime environment,” on page 223

Chapter 5. Discontinued functions

Some functions which were supported in CICS Transaction Server for z/OS, Version 3 Release 2 have been discontinued, or reduced in scope, in Version 4 Release 1.

Removal of support for Version 1.4.2 and Version 1.5.0 of IBM SDK for z/OS, Java Technology Edition

In CICS TS for z/OS, Version 4.1, support for Version 1.4.2 and Version 1.5.0 is withdrawn, and you must upgrade to Version 6 of the SDK.

When you upgrade to Java 6 from Java 1.4.2 you can benefit from the new Java language features, and also from the improvements to runtime technology in the IBM SDK for z/OS, V6. These improvements include better garbage collection and simpler class sharing.

Related concepts

Chapter 38, "Removal of support for Version 1.4.2 and Version 1.5.0 of IBM SDK for z/OS, Java Technology Edition," on page 233

Removal of support for the DFHCSDUP MIGRATE command

Support for the DFHCSDUP MIGRATE command is withdrawn in CICS TS for z/OS, Version 4.1.

In previous versions of CICS, the DFHCSDUP MIGRATE command migrated the eligible DFHDCT, DFHRCT, DFHTCT, and DFHTST macro resource definitions to the CICS system definition data set (CSD).

If you use any of these tables, you must migrate them to the CSD before you upgrade to CICS TS for z/OS, Version 4.1. To do so, you can use the DFHCSDUP MIGRATE command on any supported release up to CICS TS for z/OS, Version 3.2.

Related concepts

Chapter 39, "Removal of support for the DFHCSDUP MIGRATE command," on page 239

Part 2. Compete

CICS Transaction Server for z/OS, Version 4 Release 1 delivers a set of capabilities represented in the following themes:

- Compete
- Comply
- Control
- Architectural enhancements

The capabilities represented by the *Compete* theme make it easier to create, extend, and reuse applications quickly, to meet changing business needs.

Chapter 6. Support for event processing

You can specify, capture, and emit atomic business events from a CICS application. These business events can be consumed by another CICS application, or placed on a WebSphere MQ queue for consumption in a variety of ways including by a complex event processing engine such as IBM WebSphere Business Events.

A business event is anything that happens that is relevant to your business. An individual stock trade or the placement of an order are both examples of business events. CICS allows business events to be captured in an application and emitted for processing, with typically no change to the emitting CICS application.

The HTTP EP adapter emits events to an HTTP 1.1 compliant server using HTTP POST in XML format for consumption by products such as WebSphere Business Events and WebSphere Business Monitor without using WebSphere Message Queue as your transport.

WebSphere Message Queue remains the primary method for emitting events from CICS; the HTTP EP adapter can be used as a less robust alternative. You can also use the HTTP EP adapter to send your XML format events to vendor products or any HTTP 1.1 compliant server.

If you want to use the HTTP EP adapter functions in CICS Transaction Server for z/OS, Version 4 Release 1 you must apply APAR PK94205.

CICS and event processing

You can specify, capture, and emit atomic business events from a CICS application. These business events can be consumed by another CICS application, or placed on a WebSphere MQ queue for consumption in a variety of ways including by a complex event processing engine such as IBM WebSphere Business Events.

A business event is anything that happens that is relevant to your business. An individual stock trade or the placement of an order are both examples of business events. CICS allows business events to be captured in an application and emitted for processing, with typically no change to the emitting CICS application.

CICS Transaction Server for z/OS provides the following comprehensive support for simple business events:

- A CICS application can capture and emit business events with no change to the application itself, using noninvasive capture points, before and after selected EXEC CICS API calls, and at program start.
- For situations in which the noninvasive capture points are not sufficient to capture a specific business event, a new EXEC CICS API call, SIGNAL EVENT, allows events to be captured anywhere in a CICS application.
- The CICS Explorer includes the CICS event binding editor, a tool that helps you to define simple business events and create event bindings for your CICS applications. You deploy event bindings to a CICS system from the CICS Explorer. See Installing the CICS Explorer for further information.
- After CICS has captured and processed a business event, it is passed to an EP (Event Processing) adapter for formatting and routing. CICS provides EP adapters to allow business events to be emitted to WebSphere MQ, either in

XML format for consumption by WebSphere Business Events, in Common Base Event XML format to the Common Event Infrastructure (CEI) for consumption by business monitors, or in a non-XML character format to a CICS transaction or to a CICS TS queue, or in Common Base Event REST format to be consumed by WebSphere Business Monitor when using the WebSphere Business Monitor Event Emitter REST Interface to receive events using HTTP.

You can also write your own custom EP adapter. CICS supports custom EP adapters written in the following languages:

- COBOL
 - PL/I
 - C
 - C++
 - assembler language
- CICS event processing supports interoperability standards with business event consumers such as WebSphere Business Monitor, and others by emitting events in the Common Base Event specification V1.01 format. The common base event is an initial implementation by IBM of the Web Services Distributed Management (WSDM) Web Event Format (WEF) standard.

CICS provides support for the WebSphere Business Events XML format for Events for interoperability with WebSphere Business Events and to use a consistent XML format for events.

You can use event processing in many ways, such as to detect customer trends or to detect abnormalities in patterns of customer behavior to identify potential irregular or fraudulent situations.

Note: When planning to use CICS support for event processing, and in particular, when designing programs that consume CICS events you should consider the following characteristics:

- The order in which events are emitted from a CICS program may vary from the order in which they were captured.
- The order in which events are emitted may vary from one run to another of an event enabled CICS program.
- In the unlikely situation of a CICS system failure after the capture of an event, but before the emission of the event, the event may not be emitted at all, whether or not the EP adapter in the event binding specifies that events are transactional.

Terminology

Terms used to explain event processing.

application context

The predicates in the capture specification used for filtering on the context of the application.

application command options

The predicates in the capture specification used for filtering on the values of options on a CICS command.

application data

The predicates in the capture specification used for filtering on application data specified in a CICS command.

bundle

A directory that contains artifacts, such as event bindings, and a manifest that describes the bundle and its dependencies. The bundle is the unit of deployment for an application.

business event

A business event is anything that happens that is significant to your business.

capture data

The data included for a capture specification (the emitted business information).

capture point

An opportunity in the running of a CICS program for an event to be generated.

capture specification (event capture specification)

Includes the capture point, capture data, and predicates for filtering. The capture specification includes information describing how to obtain the emitted business information from the data available for capture.

CBE format

The common base event format used, for example, to emit events for consumption by WebSphere Business Monitor.

CBER format

The common base event REST format used, for example, to transfer events to the WebSphere Business Monitor REST interface.

CCE Format

The CICS container-based event (CCE) format used, for example, to start the processing in a CICS transaction as a result of the event.

CFE Format

The CICS flattened event (CFE) format used, for example, to write events to a temporary storage queue for testing or to a WebSphere Message Queue, from which they can be read from by a program running in or outside CICS.

context data

A fixed set of data values describing the context at the capture point; for example, the Transaction ID, the User ID, and the program name are all context data.

custom EP adapter

A vendor-written or customer-written customized EP adapter.

dispatcher

The dispatcher routes captured events to the correct EP adapter for processing.

emit Send a formatted event to a consumer from an EP adapter.

emitted business information

A collection of information to be emitted for the business event defined in business terms.

emitted business information assignment

Assigning business information items to information sources. All business information items must be assigned for a capture specification to be complete.

event binding

An XML definition that can be deployed to CICS in a bundle file. The event binding can be enabled, disabled, installed, inquired on, and uninstalled. The event binding contains: event specifications, capture specifications, and adapter information.

event binding file

An event binding is expressed as an XML document saved as a text file with a `.evbind` extension.

event capture

The process of capturing an event in CICS. After a capture point has been successfully filtered, CICS copies all of the capture data and passes it to the dispatcher.

event data

Data included with the emitted event (the emitted business information).

event processing adapter (EP adapter)

An EP adapter formats and routes events emitted by CICS.

event specification

Part of the event binding that represents a business event. The event specification contains the event name, the emitted business information, and one or more capture specifications.

filter The set of predicates connected by AND used to determine whether an event is captured. If all predicates evaluate to TRUE, the event is captured. Predicates that evaluate to FALSE filter out events.

information source

The part of the capture specification which defines where an emitted business information item is obtained. There must be an information source for each item of emitted business information.

predicate (capture predicate)

An expression used as part of a filter, consisting of a data item, an operator, and a value. A predicate is used with data values on the API call or context data, to restrict the occasions when an event is emitted to the occurrences of interest.

In the following example of a predicate:

```
Transaction ID    Equals    EXAM
```

The data items is Transaction ID, the operator is equals, and the value is EXAM.

primary predicate

A predicate for which it is strongly recommended that you provide a value, to avoid a performance impact from too many events being considered as candidates for capture.

WBE format

The WebSphere Business Events (WBE) format used, for example, to emit events for consumption by WebSphere Business Events.

Changes to external interfaces

Some external interfaces have changed to support event processing.

Changes to the CICS application programming interface

The new **SIGNAL EVENT** command identifies a place in an application program where one or more events can be emitted.

For details of the command, see **SIGNAL EVENT**.

Changes to the JCICS application programming interface

Event processing introduces the **SIGNAL EVENT** command to add an event capture point to a CICS program.

The JCICS API provides equivalents for the following new and changed API commands:

SIGNAL EVENT

The support is provided by the following new and existing JCICS classes:

com.ibm.cics.server.Event

This class is the Java implementation of the EXEC CICS **SIGNAL EVENT** command.

For details of the class, see **Event** class.

The following exception classes handle errors:

EventErrorException

This class represents the CICS **EVENTERR** condition.

For details of the exception, see **EventErrorException** class.

Changes to the system programming interface

New commands support the management of event processing and event bindings in CICS.

New commands

DISCARD EVENTBINDING

Removes the definition of an event binding from the CICS system, so that the system no longer has access to the resource.

For details of the command, see **DISCARD EVENTBINDING**.

INQUIRE CAPTURESPEC

Retrieves information about a capturespec specification.

For details of the command, see **INQUIRE CAPTURESPEC**.

INQUIRE EVENTBINDING

Returns information about the specified event binding.

For details of the command, see “**INQUIRE EVENTBINDING**” on page 367.

INQUIRE EVENTPROCESS

Returns the status of event processing.

For details of the command, see “**INQUIRE EVENTPROCESS**” on page 370.

SET EVENTBINDING

Sets the status of the specified event binding to enabled or disabled.

For details of the command, see “SET EVENTBINDING” on page 386.

SET EVENTPROCESS

Sets the status of event processing.

For details of the command, see “SET EVENTPROCESS” on page 387.

Changes to resource definitions

The HTTP EP adapter for event processing in CICS is supported by an additional use for the URIMAP resource definition.

Changed resource definition, URIMAP

When using the HTTP EP adapter, you must specify a URIMAP with USAGE(CLIENT) in your URIMAP definition.

Changes to CEMT (master terminal transaction) commands

Event processing introduces new commands to control event processing and event bindings.

New commands

DISCARD EVENTBINDING

Remove an event binding.

For details of the command, see “CEMT DISCARD” on page 395.

INQUIRE EVENTBINDING

Retrieve information about an installed event binding.

For details of the command, see “CEMT INQUIRE EVENTBINDING” on page 406.

SET EVENTBINDING

Enable or disable an event binding.

For details of the command, see “CEMT SET EVENTBINDING” on page 418.

INQUIRE EVENTPROCESS

Retrieve the status of event processing.

For details of the command, see “CEMT INQUIRE EVENTPROCESS” on page 408.

SET EVENTPROCESS

Change the status of event processing.

For details of the command, see “CEMT SET EVENTPROCESS” on page 418.

Changes to CICSplex SM resource tables

New resource tables support capture specifications, event bindings, and global event processing attributes.

New tables

The following new resource tables support event processing in CICS:

CRESEVBD

A topology table for event bindings.

CRESEVCS

A topology table for capture specifications.

EVCSPEC

A resource table for CICS resource objects that provide information about deployed capture specifications.

EVNTGBL

A resource table for CICS resource objects that provide information about event processing.

EVNTBIND

A resource table for CICS resource objects that provide information about event bindings.

Changed tables, HTASK and TASK

The CICSplex SM resource tables, HTASK and TASK, have new attributes to support event processing monitoring:

Attribute	Description
TMREICTC	Total number of EXEC CICS requests
TMRECSGE	Number of SIGNAL EVENT requests
TMRECFOC	Number of event filter commands
TMRECEVC	Number of EVENTS captured

Changed tables, CICSRRGN and MONITOR

The CICSplex SM resource tables, CICSRRGN and MONITOR, have new attributes to support CAPTURESPEC, EVENTBINDING, and EVENTPROCESS statistics reporting.

Changes to CICSplex SM views and menus

CICSplex SM views and menus now display capture specifications, event bindings, and global event processing attributes.

New menu, Applications

A new operations menu, EYUSTARTAPPLICTN, is available by selecting **CICS operations views > Applications** in the navigation of the Web User Interface. This menu provides a summary of application-related resources. The menu links to the new EYUSTARTEVNTBIND, EYUSTARTEVNTGBL, and EYUSTARTEVCSPEC view sets. Hyperlinks take you to the capture specifications, global event processing attributes, and event bindings data.

New views

New view sets are provided for capture specifications, event bindings, and global event processing attributes:

Event bindings - EVNTBIND

For access from the main menu, click **Application operations views > Event bindings**. The EVNTBIND detail view describes all the attributes of the EVNTBIND base table.

The EYUSTARTEVNTBIND view set contains a tabular view and a detailed view. A hyperlink on the detailed view takes you to the resource signature data for the event binding.

Event processing - EVNTGBL

For access from the main menu, click **Application operations views > Event processing**. The EVNTGBL detail view describes all the attributes of the EVNTGBL base table.

The EYUSTARTEVNTGBL view set contains a tabular view and a detailed view. A hyperlink on the detailed view takes you to the event bindings data.

Event capture specifications - EVCSPEC

For access from the main menu, click **Application operations views > Event capture specifications**. The EVCSPEC detail view describes all the attributes of the EVCSPEC base table.

The EYUSTARTEVCSPEC view set contains a tabular view and a detailed view. A hyperlink on the detailed view takes you to the event bindings data.

Topology data for event bindings - CRESEVBD

The CRESEVBD views display information about event bindings, indicating which managed CICS regions currently have installed EVENTBINDINGS.

Topology data for event bindings - CRESEVCS

The CRESEVCS views display information about event bindings, indicating which managed CICS regions currently have installed CAPTURESPECS.

Changes to supplied transactions

CEPH, the HTTP EP adapter for event processing, is introduced in CICS Transaction Server for z/OS, Version 4 Release 1 by APAR PK94205.

Note: This topic does not describe the following changes:

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

New transaction, CEPH

CEPH is defined by the event processing domain during CICS initialization. It is defined with RESSEC(YES) and CMDSEC(YES). CEPH runs the CICS program DFHECEAH, the HTTP EP adapter for event processing program. You can use an alternative transaction to run DFHECEAH.

CEPH is a RACF® Category 2 transaction.

Changes to global user exits

The resource management install and discard exit XRSINDI supports the new CAPTURESPEC and EVENTBINDING resources. Global user exits XWBAUTH and XWBSNDO now support the event processing HTTP EP adapter.

- For a CAPTURESPEC resource, the **UEPIDTYP** parameter returns a value of UEIDEVCS.
- For an EVENTBINDING resource, the **UEPIDTYP** parameter returns a value of UEIDEVNT.

Changed global user exits

XWBAUTH and XWBSNDO now support the HTTP EP adapter. If your target system requires basic authentication or security policies when using the HTTP EP adapter, you must implement XWBSNDO and XWBAUTH user exits to provide the required credentials.

Changes to monitoring data

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

Changed performance class group, DFHCICS

New performance data fields 402, 415, 416, and 417 are added to the DFHCICS group. These fields provide monitoring data:

402 (TYPE-A, 'EICTOTCT', 4 BYTES)

The total number of EXEC CICS commands issued by the user task.

415 (TYPE-A, 'ECSIGECT', 4 BYTES)

The number of EXEC CICS SIGNAL EVENT commands issued by the user task.

416 (TYPE-A, 'ECEFPCT', 4 BYTES)

The number of event filter operations performed by the user task.

417 (TYPE-A, 'ECEVNTCT', 4 BYTES)

The number of events captured by the user task.

Changes to statistics

There are three new statistics types, CAPTURESPEC, EVENTBINDING, and EVENTPROCESS. The statistics sample program, DFH0STAT, and the statistics utility program, DFHSTUP, now display the new fields for event processing.

New statistics types

The CAPTURESPEC resource, EVENTBINDING resource, and the EVENTPROCESS resource have new statistics.

CAPTURESPEC resource

To generate statistics for this resource type, specify the CAPTURESPEC option on the **CEMT PERFORM STATISTICS** or **EXEC CICS PERFORM STATISTICS RECORD** commands. To retrieve the statistics, use the **EXTRACT STATISTICS** command and specify **RESTYPE(EVENTBINDING)** and **SUBRESTYPE(CAPTURESPEC)**.

The DFHECCDS DSECT maps the global statistics and the DFHECRDS DSECT maps the resource statistics.

EVENTBINDING resource

To generate statistics for this resource type, specify the EVENTBINDING option on the **CEMT PERFORM STATISTICS** or **EXEC CICS PERFORM STATISTICS RECORD** commands. To retrieve the statistics, use the **EXTRACT STATISTICS** command and specify **RESTYPE(EVENTBINDING)**.

The DFHECGDS DSECT maps the global statistics and the DFHECRDS DSECT maps the resource statistics.

EVENTPROCESS resource

To generate statistics about the status of event processing, specify the

EVENTPROCESS option on the **CENT PERFORM STATISTICS** or **EXEC CICS PERFORM STATISTICS RECORD** commands. To retrieve the statistics, use the **EXTRACT STATISTICS** command and specify **RESTYPE(EVENTPROCESS)**.

The DFHEPGDS DSECT maps the global statistics.

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.

New sample program, DFH0EPAC

The sample custom EP adapter is provided in the COBOL language. It is shipped as source code in the **CICSTS41.CICS.SDFHSAMP** library, and also as a load module.

- The source code is named **DFH0EPAC**.
- The load module is named after the source code.
- Group **DFH\$EPAG** is defined in **DFHCURDS.DATA**. The group defines program **DFH0EPAC** and transaction ID **EPAT** to include in your event binding to run it.

Changes to problem determination

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

New component codes

The following component codes have been added to support event processing:

Component code	Component keyword	Description
EC	none	Event capture domain
EP	EVENTPROC	Event processing domain

You can use the component code in the following ways:

- To select the level of standard and special tracing in each component:
 - In the **CETR** transaction.
 - In the **STNTRxx** and **SPCTRxx** system initialization parameters.
 - In the **INQUIRE TRACETYPE** and **SET TRACETYPE** system programming commands. If there is a component keyword, you can use it instead of the component code in these commands.
- To specify the areas of CICS storage that you want to be included in a formatted dump and the amount of data you want formatted.
- To specify the trace entries that you want to be included in a formatted dump and in the output from the trace utility program.

On output, CICS uses the component code to identify messages and trace entries.

New messages

New error messages in the range **DFHEP0001** to **DFHEP9999** report problems with the event processing domain. New error messages in the range **DFHEC0001** to **DFHEC9999** report problems with the event capture domain.

New abend codes

New abend codes in the range AEPA to AEPZ are issued when problems occur during event processing. New abend codes in the range AECA to AECZ are issued when problems occur during event capture.

New trace points

New trace points in the range EP 0100 to EP 0A05 are issued by the event processing domain. New trace points in the range AP 3500 to AP 368B are issued by the event capture domain.

Changes to security

Resource and command security apply to CAPTURESPEC, EVENTBINDING, and EVENTPROCESS resources when these functions are enabled for the CICS region.

Transaction security categories

The CEPD transaction is the event processing dispatcher task. The CEPD transaction is a category 1 transaction that is implemented by program DFHEPDS. The CEPM transaction handles the captured event queue. It distributes events to event processing dispatcher tasks. The CEPM transaction is a category 1 transaction that is implemented by program DFHEPSY. Both CEPD and CEPM are defined internally by the event processing domain.

Resource security

Resource security for EVENTBINDING 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. You must supply a prefix of EVENTBINDING to the name of the EVENTBINDING resource definition. When you start a browse for CAPTURESPEC, it checks whether you have authority to access the EVENTBINDING that contains the CAPTURESPEC.

Command security

Command security for CAPTURESPEC resources uses the CAPTURESPEC resource in the CCICSCMD class or the VCICSCMD grouping class.

Command security for EVENTBINDING resources uses the EVENTBINDING resource in the CCICSCMD class or the VCICSCMD grouping class.

Command security for EVENTPROCESS resources uses the EVENTPROCESS resource in the CCICSCMD class or the VCICSCMD grouping class.

Surrogate user

Surrogate user security for event binding *userid.DFHINSTL* is introduced in CICS Transaction Server for z/OS, Version 4 Release 1.

Chapter 7. Atom feeds from CICS

CICS can now provide Atom feeds using data supplied by CICS resources or by application programs that access other resources. Atom feeds supply Web clients with a series of data items accompanied by metadata for each item in the Atom format. You can also use the HTTP requests defined by the Atom Publishing Protocol to edit a CICS resource that you have made available as a collection of Atom entries.

Terminology

New terms are introduced to explain the support for Atom feeds in CICS.

Ajax (Asynchronous JavaScript And XML)

A design approach and a set of techniques for delivering rich Internet applications (RIAs) using open web formats, for example, HTML, CSS and JavaScript, and rendering using a browser engine.

Atom An XML-based format for syndicated Web content and a protocol for editing and publishing Web resources that is based on the Atom Specification Format.

feed A data format that contains periodically updated content that is available to multiple users, applications, or both.

mashup

A graphical interface that features two or more reusable Web applications (widgets) presenting seemingly disparate data in an understandable combination for a specific purpose.

Really Simple Syndication (RSS)

An XML file format for syndicated Web content that is based on the Really Simple Syndication specification (RSS 2.0). The RSS XML file formats are used by Internet users to subscribe to Web sites that have provided RSS feeds.

service routine

A user program that accesses data to provide an Atom feed to Web clients.

Web 2.0

A second generation of services and applications available on the World Wide Web that enable collaboration, information sharing, dynamic service delivery, and interaction. The concept of Web 2.0 typically includes later-generation Web-based applications such as wikis and weblogs. Web 2.0 applications look more like desktop applications and are often dynamically data driven, rather than comprising static HTML content.

Atom feeds

A Web feed, sometimes just called a "feed", is a series of related items that a content provider publishes on the Internet. An Atom feed is a Web feed that uses the Atom Syndication Format and the Atom Publishing Protocol.

Atom comprises an XML-based format that describes an Atom feed and the items of information in it, and a protocol for publishing and editing Atom feeds. This

format and protocol are described in two Internet Society and IETF (Internet Engineering Task Force) Request for Comments documents (known as RFCs):

RFC 4287, *The Atom Syndication Format*, available from <http://www.ietf.org/rfc/rfc4287.txt>

RFC 5023, *The Atom Publishing Protocol*, available from <http://www.ietf.org/rfc/rfc5023.txt>

Content providers often deliver Web feeds in an earlier format called RSS (Really Simple Syndication). CICS supports Atom, but does not support RSS.

The items of information that make up an Atom feed are known as *Atom entries*. A content provider publishes, or "syndicates", an Atom feed by making it available through a URL on the Internet and updating it with new items. Web pages can display the items in the Atom feed, and Web users can obtain the items from the feed using a feed reader or Web browser. An Atom feed might be used as part of a *mashup*, which is a Web application that merges content from a number of data sources so that users can experience and understand the data in a new way. In a mashup, the data from the Atom feed can be handled by a *widget*, which is a script application that runs in a Web page.

The Atom Publishing Protocol specifies the way that users can add, delete, edit, or view individual Atom entries in an Atom feed by making HTTP requests to a server that stores the entries. A GET request retrieves an entry for viewing, a POST request adds a complete new entry, a PUT request edits an existing entry, and a DELETE request deletes an entry. The server handles the requested changes in an appropriate way and responds to the user's client with confirmation of the changes.

How CICS supports Atom feeds

CICS supports Atom feeds using the HTTP server functions of CICS Web support, and some additional functions to carry out the actions required of a server that supports the Atom format and protocol. You must select or set up a resource that provides the data for your Atom feed, and define the feed to CICS.

Before serving an Atom feed from CICS, you must configure the base components of CICS Web support to set CICS up as an HTTP server.

You can create Atom feeds from data held in or produced by existing resources, such as a temporary storage queue, a file, records in a database application, a Web service, or output produced by an existing application program. A single record in the resource holds the data for a single Atom entry. Alternatively, you can set up a new resource to contain Atom entries.

If your resource is a file or temporary storage queue defined to CICS, and you have a language structure written in COBOL, C, C++, or PL/I that describes the records in the resource, CICS can extract data directly from the resource to produce the Atom feed. You use the language structure as input to the CICS XML assistant program to produce an XML binding that defines the structure of the resource, so that CICS can map the data to the correct elements in the Atom document.

You can also serve any resource as an Atom feed by writing a program, known as a service routine, that extracts data from each record in the resource to form an Atom entry, and supplies the data to CICS in a set of containers. If you are able to

produce an XML binding for your resource, the service routine can make use of the information in the XML binding, but the service routine does not require an XML binding.

When you have identified or created the resource, and produced an XML binding or written a service routine, you define the Atom feed to CICS by creating the following items:

- An ATOMSERVICE resource definition to specify where CICS obtains the data to produce Atom documents in response to a Web client request.
- A URIMAP resource definition to specify how CICS handles HTTP requests from Web clients for the Atom feed. The URIMAP resource references the ATOMSERVICE resource definition. To support your URIMAP resource definition, you must have a TCPIPSERVICE definition that defines an inbound port for CICS Web support, on which CICS can receive HTTP requests.
- An Atom configuration file, which contains the XML syntax for the Atom feed document, together with some elements specific to CICS, such as elements to identify the resource that contains the data for the feed. CICS uses the information in the Atom configuration file to construct an Atom feed document containing a number of Atom entries, which CICS produces using the data from your resource.

If you want to enable Web clients to manage and edit the Atom entries in the feed, you can take further steps to set up the Atom feed as a collection. To set up a collection, you create a new URIMAP definition to make the collection available separately from the feed. You also create a new ATOMSERVICE definition and Atom configuration file by copying the equivalent files for the Atom feed from the same data, redefining them to state that they are for a collection, and making minor changes. You then create an Atom service document and optionally an Atom category document to define your collection, and make those documents available through CICS. If you are using a service routine, you must code it to handle Web client requests to add, edit, and delete Atom entries from the collection.

Interacting with Atom feeds

When you have set up an Atom feed, Web clients can access it to obtain a list of Atom entries. CICS, together with your service routine if used, acts as a server to receive the Web clients' HTTP requests and return Atom feed documents containing a number of Atom entries. Many free or commercially available Web client applications are able to request, receive and display Atom feeds, including most modern Web browsers, dedicated feed readers, and applications that provide further functions, such as applications for creating mashups. Check that the application is described as supporting the Atom format. You can also write your own Web client application to make GET requests for Atom feed data.

If you have also set up your Atom feed as a collection, you or others can manage and edit the entries in the feed through a Web client that supports HTTP POST, PUT, and DELETE requests for Atom feeds, as described in the Atom Publishing Protocol. If you do not have a Web client with this capability, you can use a Web client application that lets you compose and send your own HTTP requests and view the responses. You can also write your own Web client applications to make POST, PUT, and DELETE requests to Atom collections. If CICS is managing your resource directly, CICS applies the Web clients' editing requests to the data that you have made available in the collection, and returns an appropriate response. If you are using a service routine to provide your data, CICS passes the Web clients'

requests to the service routine using the container interface, and you code the service routine to modify the resource in response to the requests.

To learn more about CICS support for Atom feeds and how Web clients interact with them, follow the Web 2.0 scenarios.

Atom feeds from the CA8K SupportPac

If you used the CA8K SupportPac in CICS TS for z/OS, Version 3.1 or CICS TS for z/OS, Version 3.2 to set up Atom feeds, and you want to upgrade to use the support for Atom feeds in CICS TS for z/OS, Version 4.1, you can continue to use your service routines. However, instead of PIPELINE resource definitions, pipeline configuration files, and Resource Layout Mapping structures, you must use ATOMSERVICE resource definitions, Atom configuration files, and XML bindings. You must also make changes to your service routine code to rename the containers and to account for new parameters in one of the containers, then recompile the modules.

Changes to external interfaces

Some CICS external interfaces have changed to support Atom feeds.

Changes to system initialization parameters

A new system initialization parameter, **USSHOME**, specifies the name and path of the root directory for CICS TS 4.1 files on z/OS UNIX.

The existing system initialization parameter **JVMPROFILEDIR** now defaults to the value of **USSHOME**, followed by the subdirectory JVMProfiles.

Changes to the CICS application programming interface

The BIF DIGEST and WEB READ QUERYPARM commands are new. The CONVERTTIME, FORMATTIME, and ASKTIME commands are changed.

New command BIF DIGEST

The BIF DIGEST command calculates the SHA-1 digest of a string of data. You can use the result as a strong entity tag (ETag HTTP header) on HTTP messages sent out from CICS, or to make an HTTP PUT request conditional.

For details of this command, see “BIF DIGEST” on page 261.

New commands WEB READ QUERYPARM and WEB STARTBROWSE, READNEXT, ENDBROWSE QUERYPARM

The WEB READ QUERYPARM command and associated browsing commands read or browse keyword parameters, consisting of name and value pairs, from a query string in a URL. The new commands process query string data for all HTTP methods, including GET, POST, PUT, and DELETE. You can continue to use the existing WEB READ FORMFIELD command and its associated browsing commands for forms (messages with the media types application/x-www-form-urlencoded or multipart/form-data). Although these existing commands can read name and value pairs from a query string, they do so only when the HTTP method is GET, because they assume that the message is an HTML form.

For details of these commands, see the following topics:

“WEB READ QUERYPARM” on page 274
“WEB STARTBROWSE QUERYPARM” on page 278
“WEB READNEXT QUERYPARM” on page 276
“WEB ENDBROWSE QUERYPARM” on page 274

Changed command ASKTIME

The ABSTIME value that is returned by the **EXEC CICS ASKTIME** command is no longer rounded to the nearest 1/100 second. The absolute time returned is the system time-of-day clock, adjusted for leap seconds and the local timezone offset, truncated to the millisecond, and returned as a packed decimal of length 8 bytes. It therefore represents the number of milliseconds since 00:00 on 1 January 1900 in the local timezone and adjusted for daylight saving time.

Changed command CONVERTTIME

This command now recognizes date and time stamp strings in the format specified by RFC 3339:

RFC 3339 format

The XML dateTime datatype, specified in RFC 3339, which is taken from the ISO 8601 standard. An example of a date and time stamp in this format is "2003-04-01T10:01:02.498Z". Date and time stamps in this format are in UTC (Coordinated Universal Time, which differs only slightly from GMT), with the time zone offset (-12:00 to +12:00) indicated at the end of the date and time stamp, or the letter Z for a zero offset (+00:00). The decimal fraction of a second that is shown in the example is optional.

The command now converts all the supported time formats (not just RFC 1123 format) to local time for the ABSTIME that is returned. Also, the ABSTIME is no longer rounded to the nearest 1/100 second.

Changed command FORMATTIME

A new CVDA value RFC3339 for the STRINGFORMAT option specifies the RFC 3339 format for the architected date and time stamp string returned in DATESTRING:

RFC3339

Specifies the RFC 3339 format, also known as the XML dateTime data type. This format is an implementation of a subset of the ISO 8601 standard. An example of a date and time stamp in this format is "2003-04-24T10:01:02+00:00". Date and time stamps in this format are in UTC (Coordinated Universal Time, which differs only slightly from GMT). This date and time stamp string contains the date and the 24-hour clock time. The time zone offset (-12:00 to +12:00) is indicated at the end of the date and time stamp. The FORMATTIME command always returns the time with a zero offset from UTC.

The RFC 3339 specification allows the letter Z to be used for a zero offset (+00:00). A decimal fraction of a second in the 24-hour clock time is optional in the specification, and the FORMATTIME command does not include it. An example of a timestamp showing the decimal fraction of a second and the letter Z for a zero offset is "2003-04-01T10:01:02.498Z". If you want to add the decimal fraction of a second using your application, you can use the MILLISECONDS option to return the number of milliseconds that have also elapsed.

The new option `MILLISECONDS(data-area)` returns the number of milliseconds in the current second that is specified by `ABSTIME`, as a binary integer in the range 0 - 999.

Before CICS TS 4.1, the `EXEC CICS FORMATTIME` command rounded up a returned time if the number of milliseconds was greater than 500, except in the case where the `ABSTIME` argument contained a value representing the half-second before midnight, when no rounding was performed, and the `TIME` option returned 23:59:59. This rounding is no longer carried out, and the returned time (for example, with the `TIME` option) is given with the number of completed seconds. You can use the new `MILLISECONDS` option to obtain the number of milliseconds, and you can perform your own rounding if you need to replicate the former behavior of the command.

Changes to the JCICS application programming interface

The JCICS API supports functions provided by the new `EXEC CICS WEB READ QUERYPARM` command and the associated browsing commands. The support is provided by new methods in the `HttpRequest` class.

```
com.ibm.cics.server.HttpRequest.getQueryParam()
com.ibm.cics.server.HttpRequest.startBrowseQueryParam()
com.ibm.cics.server.HttpRequest.getNextQueryParam()
com.ibm.cics.server.HttpRequest.endBrowseQueryParam()
```

Changes to resource definitions

Delivery of Atom feeds from CICS is supported by the new `ATOMSERVICE` resource definition and changes to the `URIMAP` resource definition.

New resource definition, `ATOMSERVICE`

The new `ATOMSERVICE` resource definition defines an Atom service, feed, collection, or category document, and identifies the Atom configuration file, CICS resource or application program, and XML binding that are used to supply the data for the feed.

For details of this resource, see “`ATOMSERVICE` resources” on page 293.

Changed resource definition, `URIMAP`

`URIMAP` resource definitions handle incoming requests for Atom feeds and point to the appropriate `ATOMSERVICE` resource definition. The `URIMAP` resource definition has a new option `ATOM` for the `USAGE` attribute, and a new attribute `ATOMSERVICE`, which names the associated `ATOMSERVICE` resource.

CICS now supports the use of Internationalized Resource Identifiers (IRIs) in `URIMAP` resource definitions for inbound Web client requests. IRIs are a form of resource identifier for the Internet that permits the use of characters and formats that are suitable for national languages other than English. When you specify a host name or path that contains Unicode characters in a `URIMAP` resource definition, you must use appropriate encoding for those characters.

Changes to the system programming interface

New `CREATE`, `DISCARD`, `INQUIRE` and `SET` commands are available for `ATOMSERVICE` resource definitions.

Support for new resources

The following commands support the new ATOMSERVICE resource:

CREATE
DISCARD
All **CSD** commands

For information about the new resource, see “Changes to resource definitions” on page 48

New command **INQUIRE ATOMSERVICE**

Returns the attributes of an ATOMSERVICE resource definition. For more information, see INQUIRE ATOMSERVICE.

New command **SET ATOMSERVICE**

Enables or disables an ATOMSERVICE resource definition. For more information, see SET ATOMSERVICE.

Changed command, **INQUIRE URIMAP**

The new ATOM value for the USAGE option, and the new ATOMSERVICE option, are added to the INQUIRE URIMAP command.

ATOMSERVICE (*data-area*)

Returns the 1- to 8-character name of an ATOMSERVICE resource definition for an Atom feed. The ATOMSERVICE resource definition defines an Atom service, feed, collection, or category document, and identifies the Atom configuration file, CICS resource or application program, and XML binding that are used to supply the data for the feed. This attribute is for USAGE(ATOM).

USAGE (*cvda*)

returns a CVDA value indicating the purpose of this URIMAP definition.

ATOM

A URIMAP definition for an Atom feed. This type of URIMAP definition is used for an incoming request for data that CICS makes available as an Atom feed. The URIMAP definition maps the request URI to an ATOMSERVICE resource definition, which defines an Atom document.

Changes to **CEMT (master terminal transaction) commands**

New DISCARD, INQUIRE, and SET commands are available for ATOMSERVICE resource definitions.

Support for new resources

The **DISCARD** command supports the new ATOMSERVICE resource.

For information about the new resource, see “Changes to resource definitions” on page 48

New commands

INQUIRE ATOMSERVICE

Display the attributes of an ATOMSERVICE resource definition.

For details of the command, see “CEMT INQUIRE ATOMSERVICE” on page 400

SET ATOMSERVICE

Enable or disable an ATOMSERVICE resource definition.

For details of the command, see “CEMT SET ATOMSERVICE” on page 416

Changed command, **INQUIRE URIMAP**

The new Atom value for the Usage field and the new field Atomservice are added to the INQUIRE URIMAP command.

ATOMSERVICE (*value*)

Displays the 1- to 8-character name of an ATOMSERVICE resource definition for an Atom feed. The ATOMSERVICE resource definition defines an Atom service, feed, collection, or category document, and identifies the Atom configuration file, CICS resource or application program, and XML binding that are used to supply the data for the feed.

Usage

Displays the usage for the URIMAP definition.

ATOM

The URIMAP definition is for an Atom feed. It is used for an incoming request for data that CICS makes available as an Atom feed.

Changes to CICSplex SM resource tables

Some new resource tables apply to ATOMSERVICE resource definitions, and the resource tables that apply to URIMAP resource definitions are changed.

New tables

These new resource tables apply to ATOMSERVICE resource definitions.

ATOMDEF

BAS resource definition for ATOMSERVICE resource definitions. This table includes all the attributes for defining an ATOMSERVICE resource definition.

ATMINGRP

Information about BAS ATOMDEF resources contained in BAS resource groups.

ATOMSERV

Operations resource table for installed ATOMSERVICE resource definitions. This table includes the new statistics for Atom feeds.

CREATOM

Topology resource table for ATOMSERVICE resource definitions.

Changed tables

The USAGE(ATOM) option and the new ATOMSERVICE attribute for the URIMAP resource definition are added to the URIMPDEF and URIMAP resource tables.

The RESDESC resource table has the following new attributes to support ATOMDEF resources:

Attribute	Description
ATMDEFRG	Resource group for Atom service definitions
ATMDEFTS	Target group for Atom service definitions
ATMDEFRS	Related scope for Atom service definitions

The TASK and HTASK resource tables are updated for the new monitoring data for Atom feeds.

Changes to CICSplex SM views and menus

Some new view sets apply to Atom feeds, and some existing menus and views are changed.

Changed menus

ATOMSERVICE definitions are added to the BAS CICS resource definitions menu.

New views

New view sets are provided for Atom feeds:

Atomservice - ATOMSERV

The views in this view set show the attributes of ATOMSERVICE resource definitions and statistics for their use. You can find this view set at **CICS operations views > TCP/IP service operations views > Atomservice definitions**.

Atomservice definitions - ATOMDEF

The views in this view set show the attributes of ATOMSERVICE resource definitions. You can find this view set at **Administration views > Basic CICS resource administration views and Resource definitions > Atomservice definitions**.

ATOMDEF in resource group - ATMINGRP

The views in this view set display information about the membership of an ATOMSERVICE resource definition (ATOMDEF) in a resource group (RESGROUP). You can find this view set at **Administration views > Basic CICS resource administration views > Resource definitions in a resource group**.

Changed views

The views **URI map - URIMAP** and **URI mapping definitions - URIMPDEF** include the new USAGE(ATOM) option and the new ATOMSERVICE attribute. You can find these views at **Administration views > Basic CICS resource administration views**, and **CICS resource definitions > URI mapping definitions**.

The view **Resource descriptions - RESDESC** and the BAS relationships view **EYUSTARTMAPBAS** are changed to support ATOMSERVICE resource definitions. You can find the RESDESC views at **Administration views > Fully functional Business Application Services (BAS) administration views > Resource descriptions**.

The views **Active tasks - TASK** and **Completed tasks - HTASK** include the new monitoring data for Atom feeds. You can find these views at **CICS operations views > Task operations views > Active tasks** and **CICS operations views > Task operations views > Completed tasks**.

Changes to global user exits

The resource management install and discard exit XRSINDI supports the new ATOMSERVICE resource.

For an ATOMSERVICE resource, the **UEPIDTYP** parameter returns a value of UEIDATOM.

For information about the new resource, see “Changes to resource definitions” on page 48

Changes to monitoring data

New performance class data is added to the DFHCICS and DFHWEBB groups to support Atom feeds. Some existing fields in the DFHPROG and DFHWEBB groups are changed.

Group DFHCICS

New performance data fields 405, 406, 408, and 409 are added to this group. These fields provide data about the numbers of EXEC CICS ASKTIME, CONVERTTIME and FORMATTIME commands issued by the user task, and about the numbers of EXEC CICS BIF DEEDIT and DIGEST commands issued by the user task.

Group DFHPROG

Performance data field 071 (PGMNAME) is changed. The field now provides the name of an application program invoked by a Web alias transaction.

Group DFHWEBB

New performance data fields 380, 381, 382, 383, and 385 are added to this group. These fields provide the names of URIMAP, PIPELINE, ATOMSERVICE, and WEBSERVICE resource definitions and programs used by CICS Web support, Atom feeds, and Web service applications.

Performance data fields 224 (WBREADCT), 235 (WBTOTWCT), and 239 (WBBRWCT) are now used to monitor query parameter read and browse requests, in addition to HTTP header and formfield read and browse requests.

Changes to statistics

A new statistics record DFHW2RDS is provided for Atom feeds. The new record includes statistics for the number of Web client GET, POST, PUT, and DELETE requests for the Atom feed.

The statistics utility program DFHSTUP includes the new statistics for Atom feeds.

The sample statistics extract program DFH0STAT formats the new statistics for Atom feeds.

The EXEC CICS EXTRACT STATISTICS and PERFORM STATISTICS RECORD commands and the CEMT PERFORM STATISTICS command support the new ATOMSERVICE resource.

In the URIMAP definition statistics, which are mapped by the DFHWBRDS DSECT, the field WBR_URIMAP_USAGE, which is named URIMAP Usage in the DFHSTUP report, has a new value ATOM.

Changes to sample programs

COBOL and C language sample service routines show you how to handle Atom feeds.

The C language sample service routine, DFH\$W2S1, shows you how to respond to requests for Atom entries by reading the parameters in the DFHATOMPARGS container, updating the character containers, and updating and returning the DFHATOMPARGS container.

DFH0W2F1 is a COBOL sample service routine that shows you how to handle POST, PUT, and DELETE requests for Atom collections. CICS provides sample URIMAP and ATOMSERVICE resources in the DFH\$WEB2 group that you can use to run DFH0W2F1. The resources are both named DFH\$W2P1.

DFH0W2F1 is an updated version of the sample service routine DFH0W2FA that was provided in SupportPac CA8K. The other service routine modules, DFH\$W2FD, DFH\$W2SD, and DFH\$W2TS, that were supplied with SupportPac CA8K, are superseded by the equivalent functions in CICS TS for z/OS, Version 4.1 support for Atom feeds.

Changes to problem determination

CICS provides new information to help you diagnose problems relating to Atom feeds from CICS.

New component codes

The following component code has been added to support Atom feeds from CICS:

Component code	Component keyword	Description
W2	WEB2	Web 2.0 domain

You can use the component code in the following ways:

- To select the level of standard and special tracing in each component:
 - In the CETR transaction
 - In the **STNTRxx** and **SPCTRxx** system initialization parameters
 - In the **INQUIRE TRACETYPE** and **SET TRACETYPE** system programming commands. If there is a component keyword, you can use this instead of the component code in these commands
- To specify the areas of CICS storage that you want to be included in a formatted dump, and the amount of data you want formatted
- To specify the trace entries that you want to be included in a formatted dump and in the output from the trace utility program

On output, CICS uses the component code to identify messages and trace entries.

New messages

New messages beginning DFHW2 relate to the initialization and operation of the W2 (Web 2.0) domain.

Messages are produced for the following error conditions:

- When the Atom configuration file cannot be successfully parsed
- When the HTTP request body for a PUT or POST request cannot be successfully parsed
- When the CICS resource identified by RESOURCENAME and RESOURCETYPE in the ATOMSERVICE resource definition cannot be located
- When I/O errors occur accessing the CICS resource identified by RESOURCENAME and RESOURCETYPE
- When any of the fields named in the <cics:fieldnames> element of the Atom configuration file cannot be located in the XML binding description

The DFHXS1111 security violation message might now be issued when access to an ATOMSERVICE resource definition is denied.

New abend codes

Abend code AALA is issued when CICS detects an unrecoverable error installing an ATOMSERVICE resource definition. Abend code AW2A is issued if you attempt to run the CW2A transaction from a terminal. Abend code AW2B is issued if the transaction work area is too small to contain the responses from the Atom feed manager.

New trace points

New trace points trace entries and exits from the new W2 domain modules, and the contents of parameters for the modules.

Changes to HTTP status codes for Atom feed support

When serving Atom feeds, CICS issues some new HTTP status codes, and some status codes that CICS previously issued are now issued for new reasons. The new status codes that are issued by CICS are as follows:

201 Created	Issued in response to a request with the POST method. A new object has been created. The new URL for the object is returned in the Location header.
409 Conflict	When issued in response to a request with the POST method, this status code means that an existing object already exists with the specified URL, so the new object is not created.
These status codes were previously issued by CICS, but are now issued for new reasons:	
400 Bad Request	When issued in response to a request with the PUT method, this status code might mean that a PUT request without an If-Match header was received. A client that wants to update an object without knowing the current entity tag must specify If-Match: *. The status code is also issued for a markup or data problem in the Atom entry submitted by the Web client for a POST or PUT request.

403 Forbidden	Now issued when the current user is not authorized to access one of the following: <ul style="list-style-type: none"> • The alias transaction specified in the TRANSACTION attribute of the URIMAP resource definition • The ATOMSERVICE resource definition • The CICS resource specified in the ATOMSERVICE resource definition • Any CICS resource or command accessed by a program that is specified in the ATOMSERVICE resource definition
404 Not Found	Now issued when any of the following items cannot be found: <ul style="list-style-type: none"> • The ATOMSERVICE resource definition specified in the URIMAP resource definition • The CICS resource specified in the Atom configuration file • The selected record in the CICS resource
412 Precondition Failed	Now issued in response to a request with the PUT method, when the entity tag value on the If-Match header does not match the entity tag for the object being updated. The current contents of the object are returned in the response body, and the Etag header contains the new entity tag value.
500 Resource Error	Now issued for some errors involving a resource for an Atom feed, such as an error producing XML markup from a resource record for use as Atom entry content.
503 Service Unavailable	Now issued when either a requested ATOMSERVICE resource definition, or the CICS resource that it references, is disabled.

Changes to security

Resource and command security applies to ATOMSERVICE resource definitions if these functions are enabled for the CICS region.

Resource security for ATOMSERVICE 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. You must apply a prefix of ATOMSERVICE to the name of the ATOMSERVICE resource definition.

```
RDEFINE RCICSRES(ATOMSERVICE.atom1, ATOMSERVICE.atom2, ...,
             ATOMSERVICE.atomn) UACC(NONE)
PERMIT ATOMSERVICE.atom1 CLASS(RCICSRES) ID(group1, group2)
ACCESS(READ)
```

To define Atom services as members of a profile in the WCICSRES resource grouping class, with an appropriate access list, use the following commands:

```
RDEFINE WCICSRES(atom_groupname) UACC(NONE) ADDMEM(ATOMSERVICE.atoma,
             ATOMSERVICE.atomb)
PERMIT atom_groupname CLASS(WCICSRES) ID(group_userid) ACCESS(READ)
```

Command security for ATOMSERVICE resources uses the ATOMSERVICE resource in the CCICSCMD class or the VCICSCMD grouping class:

```
RDEFINE CCICSCMD(ATOMSERVICE) UACC(NONE)
PERMIT ATOMSERVICE CLASS(CCICSCMD) ID(group1, group2) ACCESS(READ)
or
RDEFINE VCICSCMD cmdsamp UACC(NONE) ADDMEM(ATOMSERVICE)
PERMIT cmdsamp CLASS(VCICSCMD) ID(operator_group) ACCESS(READ)
```

CW2A transaction

The new RACF Category 2 transaction CW2A is the default alias transaction for Atom feeds. It is defined with RESSEC(YES) and CMDSEC(YES). CW2A is defined in the new CICS-supplied resource definition group DFHWEB2.

Chapter 8. Support for application bundles

You can now deploy applications into CICS using bundles. A *bundle* is a collection of CICS resources, artifacts, references, and a manifest that represent an application. Use bundles to more easily manage the availability of an application and the life-cycle of its resources.

Bundles are created by an application developer using a tool such as Rational Developer for System z, the CICS XML assistant or the IBM CICS Explorer. A bundle contains only the resources that are required by the application. The system resources that the application requires might be defined as prerequisites, but they are not included in the bundle. This separation means that you can install the same application into multiple CICS regions without having to repackage or redeploy the bundle.

A bundle is defined in CICS using a BUNDLE resource. For details about how to define this resource and for more information about the format of its contents, see the *CICS Resource Definition Guide*.

The BUNDLE resource is different from an RDO group because it maintains a relationship with all the resources after they are installed, so that you can manage all the related resources as a single entity; for example, if you disable a BUNDLE resource because you want to stop an application from running, CICS disables all of the related application resources for you. To view the contents of a bundle and the state of its resources, use the IBM CICS Explorer.

Terminology

New terms are introduced to explain the support for bundles.

bundle

A directory that contains artifacts, and a manifest that describes the bundle and its dependencies. The bundle is the unit of deployment for an application.

manifest

A special file that describes the resources, location of supporting artifacts, application prerequisites, and services that are included in a bundle to deploy part or all of an application.

Application types that support bundles

The types of application that you can deploy as bundles include event processing, channel-based services, and XML-based services. Each of these application types is represented by one or more CICS resources and these resources are dynamically created as part of the bundle deployment.

Channel-based services

Channel-based services are CICS applications that are described as components and assembled together using the Service Component Architecture (SCA) tooling in Rational Developer for System z. The SCA tooling deploys the composite application to CICS as a bundle. These services are available only to other CICS applications that use the **INVOKE SERVICE** API command and pass binary data in containers on a channel.

Event processing

If you deploy an application that uses event bindings from the CICS event binding editor, installing the BUNDLE resource generates one or more EVENTBINDING and CAPTURESPEC resources. The resource signatures for each resource indicate that they were created during a bundle deployment and contain the name of the BUNDLE resource.

XML-based services

XML-based services are typically Web service provider or requester applications that use XML to interface with other applications and use a binding to transform the data. XML-based services are available to CICS applications that use the **INVOKE SERVICE** API command or to business services that are on an external network. If you create a Web service using the SCA tooling in Rational Developer for System z, you can deploy the Web service as a bundle.

Installing the BUNDLE resource for a Web service can generate a number of CICS resources, including URIMAP and WEBSERVICE resources.

An XML-based service can also be an application that uses the **TRANSFORM** API commands to map application data to and from XML. The XML assistant uses a language structure or an XML schema to generate the XML binding and can also create a bundle. If you install the BUNDLE resource, CICS dynamically creates an XMLTRANSFORM resource that defines where the XML binding and schema are located.

You can extend the list of supported application types by using the callback interface in the resource life-cycle manager domain. With this interface, vendors can create new user resource types and manage them in BUNDLE resources.

Creating user resource types in bundles

The resource types that are supported in a bundle are extensible, so you can provide your own resource types if required. To support your new resource type in CICS, you must write a program to handle the creation and life cycle of the resource type when the bundle is deployed to CICS.

Before you begin

Before you deploy the bundle to CICS, ensure that the bundle manifest that defines your resource type conforms to the bundle manifest schema.

About this task

CICS provides two interfaces that are channel-based. The first interface registers a program as the manager of a user resource. The second interface is similar to the callbacks between CICS domains. CICS uses the callback interface to pass details of a resource that must be created to the registered program when a BUNDLE resource is installed.

Procedure

1. Write a program to register a callback program as the manager of a resource type in a CICS region. CICS uses this program when the region starts up to recover the resource type in the bundle.
2. Write a callback program to handle the management of the resource type. CICS calls this program using a callback interface when a bundle that contains the resource type is deployed.

3. Create and install PROGRAM resource definitions for the two programs. The program that uses the registration interface must be available during PLTPI processing.

What to do next

Test that a bundle containing your resource type can successfully be deployed into CICS and that an enabled BUNDLE resource can be recovered on a CICS restart.

The bundle registration interface

Use the bundle registration interface to register a program as the owner of a resource type that is deployed in an application bundle. The program is responsible for managing the life cycle of any resources of a particular type that are installed in a CICS region as part of a bundle deployment.

A *bundle* is the unit of deployment for an application. A bundle is defined and installed in a CICS region using the BUNDLE resource. The bundle contains a manifest that describes all of the resources that the application requires as universal resource identifiers (URIs). These resources can be typical CICS RDO resources, artifacts that exist on zFS, or user resources.

To define your own user resources in a bundle, write a callback program to handle the management of the user resources when the bundle deploys into the CICS region. CICS provides a registration interface to register the callback program as the owner of a user resource type. The registration interface is channel-based and provides containers that your registration program must populate.

When a bundle contains a resource type that the callback program is responsible for handling, CICS calls the program to create the resource using a channel-based callback interface.

All the resources associated with the bundle are managed together; for example, when you disable a BUNDLE resource, all the associated resources are also disabled. Your callback program can opt into this life-cycle support if appropriate for the user resource.

Changes to external interfaces

Some changed CICS external interfaces now support the deployment and management of application bundles.

Changes to resource definitions

To manage an application that is deployed as a bundle, use the new BUNDLE resource definition.

New resource definition, BUNDLE

The BUNDLE resource defines a bundle, the unit of deployment for an application.

For details of this resource, see BUNDLE resource definitions.

Changes to the system programming interface

New and changed commands support the management of bundles in CICS.

Support for new resources

The following commands support the new BUNDLE resource:

CREATE
DISCARD
All **CSD** commands

For information about the new resource, see “Changes to resource definitions” on page 59

New commands

INQUIRE BUNDLE

Retrieve information about a BUNDLE resource definition.

For details of the command, see Appendix C, “New system programming commands,” on page 305.

INQUIRE BUNDLEPART

Retrieve information about the contents of an installed BUNDLE resource.

For details of the command, see Appendix C, “New system programming commands,” on page 305.

SET BUNDLE

Change the status of an installed BUNDLE resource.

For details of the command, see Appendix C, “New system programming commands,” on page 305.

Changes to CEMT (master terminal transaction) commands

A number of new CEMT commands help you to manage the bundles that are deployed in CICS.

Support for new resources

The **DISCARD** command supports the new BUNDLE resource.

For information about the new resource, see “Changes to resource definitions” on page 59

New commands

INQUIRE BUNDLE

Retrieve the status of the BUNDLE resource.

For details of the command, see “CEMT INQUIRE BUNDLE” on page 403.

SET BUNDLE

Enable the BUNDLE resource.

For details of the command, see “CEMT SET BUNDLE” on page 417.

Changes to CICSplex SM resource tables

New and changed resource tables support the management of application bundles.

New tables

The following new resource tables support bundles in CICS:

BUNDDEF

A BAS resource definition table for BUNDLE resources. This table includes all the attributes for defining a BUNDLE resource definition.

BUNDLE

An operations table for installed application bundles.

BUNDPART

An operations table for the resources that are dynamically installed and managed by application bundles.

BUNINGRP

A resource table for BAS BUNDDEF resource objects that are contained in BAS resource groups.

CRESBUND

A topology table for installed application bundles.

Changed table, RESDESC

The BAS resource description table (RESDESC) has new attributes to support BUNDDEF resource objects:

Attribute	Description
BUNDEFRG	Resource group for bundle definitions
BUNDEFTS	Target group for bundle definitions
BUNDEFRS	Related scope for bundle definitions

Changes to CICSplex SM views and menus

A number of new and changed view sets and menus support the management of application bundles.

New menu, EYUSTARTAPPLICTN

A new operations menu, EYUSTARTAPPLICTN, is available by selecting **CICS operations views > Applications** in the navigation of the Web User Interface. This menu provides a summary of application-related resources. The menu links to the new EYUSTARTBUNDLE view set.

Changed menus

The following menus now support the BUNDDEF resource object:

- The BAS CICS resource definitions menu, EYUSTARTADMRES, includes a link to the new bundle definitions view set.
- The CICS operations menu, EYUSTARTOPERATE, includes a link to the new EYUSTARTAPPLICTN menu.

New views

Bundle definitions - BUNDDEF

To access this view set from the main menu, click **Administration views > CICS resource definitions > Bundle definitions**.

The views in this view set show the attributes of BUNDLE resource definitions.

Bundles - BUNDLE

To access this view set from the main menu, click **CICS operations views > Application operations views > Bundles**

The views in this view set show the installed bundles in the selected scope. The detailed view of a bundle shows the resource signature and the name of the system on which the BUNDLE resource is installed.

Changed views

The following views now support the BUNDDDEF resource object:

- The resource description, EYUSTARTRESDESC
- Resource selected by resource descriptions, EYUSTARTRDSCPROC
- Mapping of BAS relationships, EYUSTARTMAPBAS

Changes to global user exits

A new value on the XRSINDI global user exit supports application bundles in CICS.

Support for new resources in global user exit, XRSINDI

The resource management installation and discard exit XRSINDI supports the new BUNDLE resource. For a BUNDLE resource, the **UEPIDTYP** parameter returns a value of UEIDBNL.

For information about the new resource, see “Changes to resource definitions” on page 59

Changes to statistics

CICS now collects statistics about application bundles that are deployed in CICS.

New statistics type, BUNDLE

A new option of BUNDLE on the **PERFORM STATISTICS RECORD** and **CEMT PERFORM STATISTICS** commands record statistics for application bundles.

A new RESTYPE value of BUNDLE for the **EXEC CICS EXTRACT STATISTICS** command retrieves statistics relating to application bundles.

The DFHRLRDS DSECT maps the application bundle statistics.

The statistics utility program DFHSTUP includes the new statistics for application bundles and the sample statistics extract program DFH0STAT formats the new statistics.

Changes to problem determination

CICS provides new information to help you diagnose problems relating to application bundles.

New component codes

The following component code supports application bundles:

Component code	Component keyword	Description
RL	RESLIFEMGR	Resource life cycle domain

You can use the component code in the following ways:

- To select the level of standard and special tracing in each component:
 - In the CETR transaction.
 - In the **STNTRxx** and **SPCTRxx** system initialization parameters.
 - In the **INQUIRE TRACETYPE** and **SET TRACETYPE** system programming commands. If there is a component keyword, you can use it instead of the component code in these commands.
- To specify the areas of CICS storage that you want to be included in a formatted dump and the amount of data you want formatted.
- To specify the trace entries that you want to be included in a formatted dump and in the output from the trace utility program.

On output, CICS uses the component code to identify messages and trace entries.

New messages

New messages with the prefix DFHRL relate to the initialization and operation of the resource life-cycle (RL) domain.

New abend code

The new abend code AAM4 is issued when CICS detects an unrecoverable error installing a BUNDLE resource definition.

New trace points

New trace points, in the range RL 0100 to RL 0604, trace entries and exits from the new RL domain modules and also trace the contents of parameters for those modules.

Changes to security

Resource and command security apply to BUNDLE resource definitions if these functions are enabled for the CICS region.

Transaction security categories

The CRLR transaction handles the resolution of BUNDLE resource definitions when a CICS region restarts. The CRLR transaction is a category 1 transaction that is implemented by program DFHRLR. The transaction and the program are in a new CSD group, DFHRL, which is included in DFHLIST.

Resource security

Resource security for BUNDLE 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. You must supply a prefix of BUNDLE to the name of the BUNDLE resource definition.

Command security

Command security for BUNDLE resources uses the BUNDLE resource in the CCICSCMD class or the VCICSCMD grouping class.

Chapter 9. Improvements to data mappings

You can now modernize your CICS applications to use XML in channel-based interfaces. Using new utilities and API commands, you can map application binary data to XML and transform it during runtime processing.

The advantage of using this approach to transform application data to and from XML is that CICS goes beyond the capabilities offered by an XML parser. CICS can interpret the XML and perform record-based conversions of application data. Therefore, it is easier and faster for you to create applications that work with XML using this approach.

CICS supports a number of high-level languages and provides an XML assistant to map how the data is transformed during runtime processing. CICS uses the same technology for mapping application data to XML in SOAP messages, as part of the Web services support.

The CICS XML assistant is a supplied utility that helps you to create the required mappings to transform application binary data to XML or transform XML to application binary data. The XML assistant generates an XML binding and optionally a schema in a bundle directory on z/OS UNIX. By creating and installing a corresponding BUNDLE resource, CICS dynamically creates an XMLTRANSFORM resource that defines where the XML binding and schema are located.

If you are using Web services, this support provides a number of benefits:

- You can now process SOAP headers and SOAP faults programmatically.
- You can now create a Web service application that builds a SOAP message from multiple containers on a channel.

The CICS XML assistant

The CICS XML assistant is a set of batch utilities that can help you to transform XML into high-level language structures and vice versa. The assistant supports rapid deployment of applications that perform XML processing with the minimum amount of programming effort.

Using the XML assistant for CICS reduces the amount of code that you must write to parse or construct XML; CICS transforms data between XML fragments and the data structure of an application program.

The XML assistant can create an XML schema from a simple language structure, or a language structure from an existing XML schema, and supports COBOL, C/C++, and PL/I. It also generates metadata that CICS uses at run time to automatically convert XML data to binary application data or vice versa; the metadata is defined in an XML binding and stored on z/OS UNIX. The schema for the XML binding is in `/usr/lpp/cicsts/cicsts41/schemas/xmltransform/` directory on z/OS UNIX.

The CICS XML assistant comprises two utility programs:

DFHLS2SC

This utility generates an XML schema and binding from a language structure.

DFHSC2LS

This utility generates an XML binding and language structure that you can use in your application programs. You can use either a WSDL document or an XML schema as input.

The JCL procedures to run both programs are in the *hlq.XDFHINST* library, where *hlq* is the high-level qualifier of your CICS installation.

DFHLS2SC: high-level language to XML schema conversion

The DFHLS2SC cataloged procedure generates an XML schema and an XML binding file from a high-level language structure. Use DFHLS2SC when you want to create a CICS program that can parse or create XML.

The job control statements for DFHLS2SC, its symbolic parameters, its input parameters and their descriptions, and an example job help you to use this procedure.

Job control statements for DFHLS2SC

JOB Starts the job.

EXEC Specifies the procedure name (DFHLS2SC).

INPUT.SYSUT1 DD

Specifies the input. The input parameters are usually specified in the input stream. However, you can define them in a data set or in a member of a partitioned data set.

Symbolic parameters

The following symbolic parameters are defined in DFHLS2SC:

JAVADIR=*path*

Specifies the name of the Java directory that is used by DFHLS2SC. The value of this parameter is appended to `/usr/lpp/` giving a complete path name of `/usr/lpp/path`.

Typically, you do not specify this parameter. The default value is the value that was supplied to the CICS installation job (DFHISTAR) in the **JAVADIR** parameter.

PATHPREF=*prefix*

Specifies an optional prefix that extends the z/OS UNIX directory path used on other parameters. The default is the empty string.

Typically, you do not specify this parameter. The default value is the value that was supplied to the CICS installation job (DFHISTAR) in the **JAVADIR** parameter.

SERVICE=*value*

Use this parameter only when directed to do so by IBM support.

TMPDIR=*tmpdir*

Specifies the location of a directory in z/OS UNIX that DFHLS2SC uses as a temporary work space. The user ID under which the job runs must have read and write permission to this directory.

The default value is `/tmp`.

TMPFILE=*tmpprefix*

Specifies a prefix that DFHLS2SC uses to construct the names of the temporary workspace files.

The default value is SC2WS.

USSDIR=*path*

Specifies the name of the CICS TS directory in the z/OS UNIX file system. The value of this parameter is appended to /usr/lpp/cicsts/ giving a complete path name of /usr/lpp/cicsts/*path*.

Typically, you do not specify this parameter. The default value is the value that was supplied to the CICS installation job (DFHISTAR) in the **USSDIR** parameter.

The temporary work space

DFHLS2SC creates the following three temporary files at run time:

```
tmpdir/tmpprefix.in  
tmpdir/tmpprefix.out  
tmpdir/tmpprefix.err
```

where:

tmpdir is the value specified in the **TMPDIR** parameter.

tmpprefix is the value specified in the **TMPFILE** parameter.

The default names for the files (when **TMPDIR** and **TMPFILE** are not specified), are as follows:

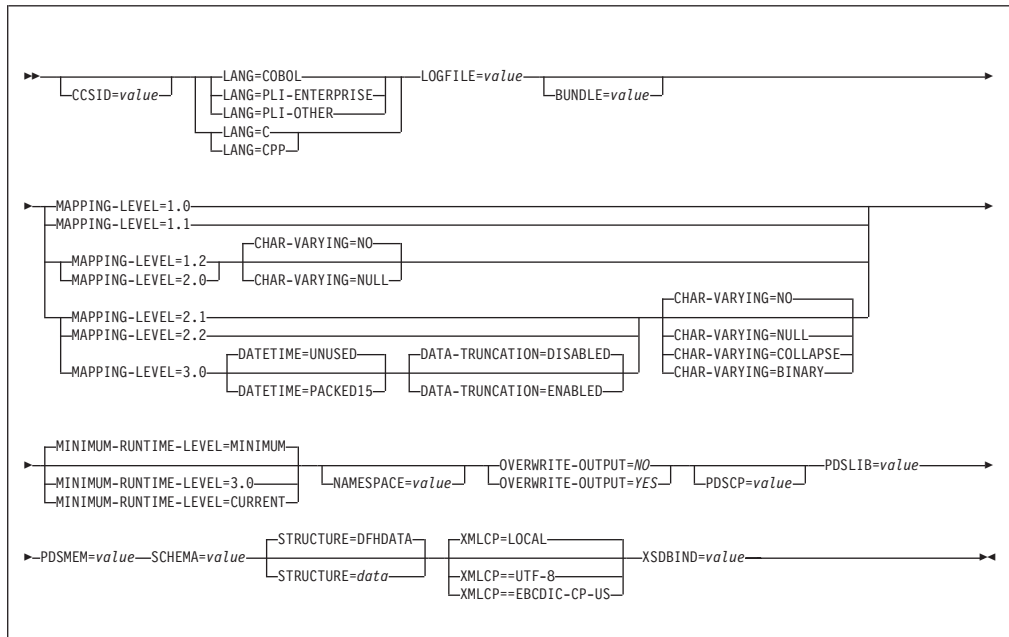
```
/tmp/LS2SC.in  
/tmp/LS2SC.out  
/tmp/LS2SC.err
```

Important: DFHLS2SC does not lock access to the generated z/OS UNIX file names. Therefore, if two or more instances of DFHLS2SC run concurrently and use the same temporary workspace files, nothing prevents one job from overwriting the workspace files while another job is using them, leading to unpredictable failures.

Therefore, you are advised to devise a naming convention and operating procedures that avoid this situation; for example, you can use the system symbolic parameter **SYSUID** to generate workspace file names that are unique to an individual user.

These temporary files are deleted before the end of the job.

Input parameters for DFHLS2SC



Parameter use

- You can specify the input parameters in any order.
- Each parameter must start on a new line.
- A parameter (and its continuation character, if you use one) must not extend beyond column 72; columns 73 to 80 must contain blanks.
- If a parameter is too long to fit on a single line, use an asterisk (*) character at the end of the line to indicate that the parameter continues on the next line. Everything (including spaces) before the asterisk is considered part of the parameter. For example:

```
XSDBIND=xsdbinddir*
/app1
```

is equivalent to

```
XSDBIND=xsdbinddir/app1
```

- A # character in the first character position of the line is a comment character. The line is ignored.

Parameter descriptions

BUNDLE=value

Specifies the path and name of the bundle directory on z/OS UNIX. If you specify this value, the XML assistant generates the XML schema and binding in the bundle directory and creates a bundle manifest for you. The path information on this parameter overrides any path information on the **SCHEMA** and **XSDBIND** parameters.

You can optionally specify an archive file rather than a directory name. The XML assistant supports .zip and .jar archives. However, you must uncompress the archive before trying to install the BUNDLE resource.

If you do not specify this parameter, CICS places the XML schema and binding in the locations specified on those parameters.

CCSID=*value*

Specifies the CCSID that is used at run time to encode character data in the application data structure. The value of this parameter overrides the value of the **LOCALCCSID** system initialization parameter. The *value* must be an EBCDIC CCSID that is supported by Java and z/OS conversion services. If you do not specify this parameter, the application data structure is encoded using the CCSID specified in the system initialization parameter.

You can use this parameter with any mapping level.

CHAR-VARYING={NO | NULL | COLLAPSE | BINARY}

Specifies how character fields in the language structure are mapped when the mapping level is 1.2 or higher. A character field in COBOL is a Picture clause of type X; for example, PIC(X) 10. A character field in C/C++ is a character array. This parameter does not apply to Enterprise and Other PL/I language structures. You can select these options:

NO Character fields are mapped to an `xsd:string` and are processed as fixed-length fields. The maximum length of the data is equal to the length of the field. **NO** is the default value for the **CHAR-VARYING** parameter for COBOL and PL/I at mapping levels 2.0 and earlier.

NULL Character fields are mapped to an `xsd:string` and are processed as null-terminated strings. CICS adds a terminating null character when transforming from an XML schema. The maximum length of the character string is calculated as one character less than the length indicated in the language structure. **NULL** is the default value for the **CHAR-VARYING** parameter for C/C++.

COLLAPSE

Character fields are mapped to an `xsd:string`. Trailing white space in the field is not included in the XML schema. **COLLAPSE** is the default *value* for the **CHAR-VARYING** parameter for COBOL and PL/I at mapping level 2.1 onwards.

BINARY

Character fields are mapped as an `xsd:base64binary` data type and are processed as fixed-length fields. The **BINARY** *value* on the **CHAR-VARYING** parameter is available only at mapping levels 2.1 and onwards.

DATA-TRUNCATION={DISABLED | ENABLED}

Specifies if variable length data is tolerated in a fixed length field structure:

DISABLED

If the data is less than the fixed length that CICS is expecting, CICS rejects the truncated data and issues an error message.

ENABLED

If the data is less than the fixed length that CICS is expecting, CICS tolerates the truncated data and processes the missing data as null values.

DATETIME={UNUSED | PACKED15}

Specifies if `dateTime` fields in the high-level language structure, including CICS `ABSTIME` values, are mapped as time stamps:

PACKED15

Any `dateTime` fields are mapped as time stamps.

UNUSED

Any dateTime fields are not mapped as time stamps. This mapping is the default.

You can set this parameter at a mapping level of 3.0.

LANG=COBOL

Specifies that the programming language of the high-level language structure is COBOL.

LANG=PLI-ENTERPRISE

Specifies that the programming language of the high-level language structure is Enterprise PL/I.

LANG=PLI-OTHER

Specifies that the programming language of the high-level language structure is a level of PL/I other than Enterprise PL/I.

LANG=C

Specifies that the programming language of the high-level language structure is C.

LANG=CPP

Specifies that the programming language of the high-level language structure is C++.

LOGFILE=*value*

The fully qualified z/OS UNIX name of the file into which DFHLS2SC writes its activity log and trace information. DFHLS2SC creates the file, but not the directory structure, if it does not already exist.

Typically, you do not use this file, but it might be requested by the IBM service organization if you encounter problems with DFHLS2SC.

MAPPING-LEVEL={1.0|1.1|1.2|2.0|2.1|2.2|3.0}

Specifies the level of mapping for the assistant to use when generating the XML binding and language structures. You are recommended to use the most recent mapping level that is available. If you are creating an XML binding for an Atom feed, you must use a mapping level of 3.0.

MINIMUM-RUNTIME-LEVEL={MINIMUM|3.0|CURRENT}

Specifies the minimum CICS runtime environment into which the XML binding can be deployed. If you select a level that does not match the other parameters that you have specified, you receive an error message. The options that you can select are as follows:

MINIMUM

The lowest possible runtime level of CICS is allocated automatically based on the parameters that you have specified.

3.0 Specify runtime level 3.0 or above if you want to use the CICS XML assistant and take advantage of advanced data mappings.

CURRENT

Use this runtime level to deploy the generated binding file into a CICS region that has the runtime environment set to the one used to generate the binding file.

NAMESPACE=NO | YES

Controls whether existing CICS BUNDLES on the file system can be overwritten.

If you do not specify this parameter, CICS generates a namespace automatically.

OVERWRITE-OUTPUT=*value*

Specifies the namespace for CICS to use in the generated XML schema. For Atom feeds, CICS provides this namespace in the Atom feed together with the Atom namespace.

NO Any existing BUNDLE is not replaced. If an existing BUNDLE is found DFHLS2SC issues error message DFHPI9689E and terminates.

YES Any existing BUNDLE is replaced. If an existing BUNDLE is found then message DFHPI9683W is issued to inform you that the file has been replaced.

PDSCP=*value*

Specifies the code page that is used in the partitioned data set members, where *value* is a CCSID number or a Java code page number. If you do not specify this parameter, the z/OS UNIX System Services code page is used. For example, you might specify PDSCP=037.

PDSLIB=*value*

Specifies the name of the partitioned data set that contains the high-level language data structures to be processed.

Restriction: The records in the partitioned data set must have a fixed length of 80 bytes.

PDSMEM=*value*

Specifies the name of the partitioned data set member that contains the high-level language structures to be processed.

SCHEMA=*value*

The fully qualified z/OS UNIX name of the file into which the XML schema is written. The XML schema conforms to the WSDL 2.0 specification. DFHLS2SC creates the file, but not the directory structure, if it does not already exist.

STRUCTURE={DFHDATA | *data*}

The name of the top-level data structure in C and C++. The default is DFHDATA.

XMLCP={LOCAL | UTF-8 | EBCDIC-CP-US}

Specifies the code page that is used to generate the XML binding.

LOCAL

This value is the default. It specifies that the XML is generated using the local code page and no encoding tag is generated in the XML schema.

UTF-8 Specifies that the XML is generated using the UTF-8 code page. An encoding tag is generated in the XML schema. If you specify this option, you must ensure that the encoding remains correct when copying the XML schema between different platforms.

EBCDIC-CP-US

Specifies that the XML is generated using the US EBCDIC code page. An encoding tag is generated in the XML schema.

XSDBIND=*value*

The fully qualified z/OS UNIX name of the XSD binding. DFHLS2SC creates the file, but not the directory structure, if it does not already exist. The file extension is `.xsbind`.

DFHSC2LS: XML schema to high-level language conversion

The DFHSC2LS cataloged procedure generates a high-level language data structure and an XML binding from an XML schema or WSDL document. Use DFHSC2LS when you want to create a CICS program that can parse or create XML.

The job control statements for DFHSC2LS, its symbolic parameters, its input parameters and their descriptions, and an example job help you to use this procedure.

Job control statements for DFHSC2LS

JOB Starts the job.

EXEC Specifies the procedure name (DFHSC2LS).

INPUT.SYSUT1 DD

Specifies the input. The input parameters are usually specified in the input stream. However, you can define them in a data set or in a member of a partitioned data set.

Symbolic parameters

The following symbolic parameters are defined in DFHSC2LS:

JAVADIR=*path*

Specifies the name of the Java directory that is used by DFHSC2LS. The value of this parameter is appended to /usr/lpp/ giving a complete path name of /usr/lpp/*path*.

Typically, you do not specify this parameter. The default value is the value that was supplied to the CICS installation job (DFHISTAR) in the **JAVADIR** parameter.

PATHPREF=*prefix*

Specifies an optional prefix that extends the z/OS UNIX directory path used on other parameters. The default is the empty string.

Typically, you do not specify this parameter. The default value is the value that was supplied to the CICS installation job (DFHISTAR) in the **JAVADIR** parameter.

TMPDIR=*tmpdir*

Specifies the location of a directory in z/OS UNIX that DFHSC2LS uses as a temporary work space. The user ID under which the job runs must have read and write permission to this directory.

The default value is /tmp.

TMPFILE=*tmpprefix*

Specifies a prefix that DFHSC2LS uses to construct the names of the temporary workspace files.

The default value is SC2LS.

USSDIR=*path*

Specifies the name of the CICS TS directory in the z/OS UNIX file system. The value of this parameter is appended to /usr/lpp/cicsts/ giving a complete path name of /usr/lpp/cicsts/*path*.

Typically, you do not specify this parameter. The default value is the value that was supplied to the CICS installation job (DFHISTAR) in the **USSDIR** parameter.

SERVICE=value

Use this parameter only when directed to do so by IBM support.

The temporary work space

DFHSC2LS creates the following three temporary files at run time:

```
tmpdir/tmpprefix.in  
tmpdir/tmpprefix.out  
tmpdir/tmpprefix.err
```

where:

tmpdir is the value specified in the **TMPDIR** parameter.

tmpprefix is the value specified in the **TMPFILE** parameter.

The default names for the files (when **TMPDIR** and **TMPFILE** are not specified), are as follows:

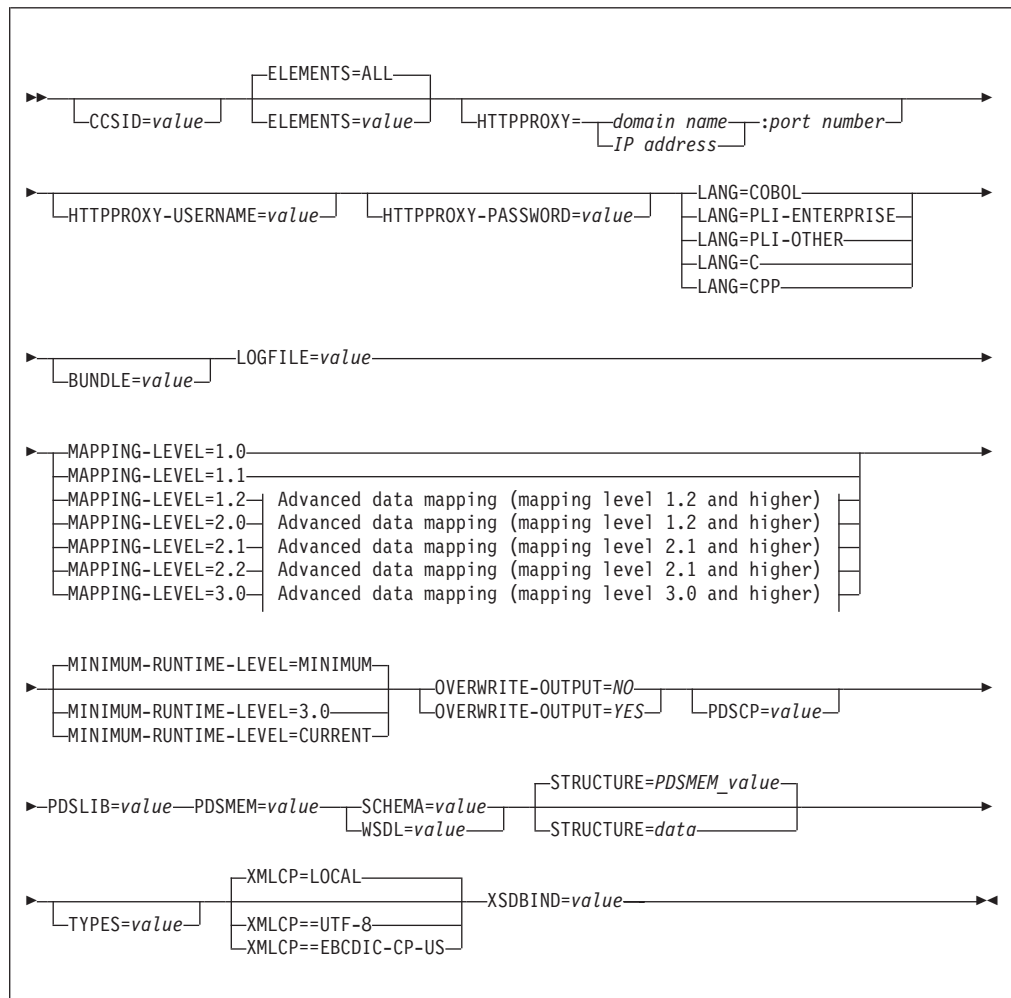
```
/tmp/SC2LS.in  
/tmp/SC2LS.out  
/tmp/SC2LS.err
```

Important: DFHSC2LS does not lock access to the generated z/OS UNIX file names. Therefore, if two or more instances of DFHSC2LS run concurrently, and use the same temporary workspace files, nothing prevents one job from overwriting the workspace files while another job is using them, leading to unpredictable failures.

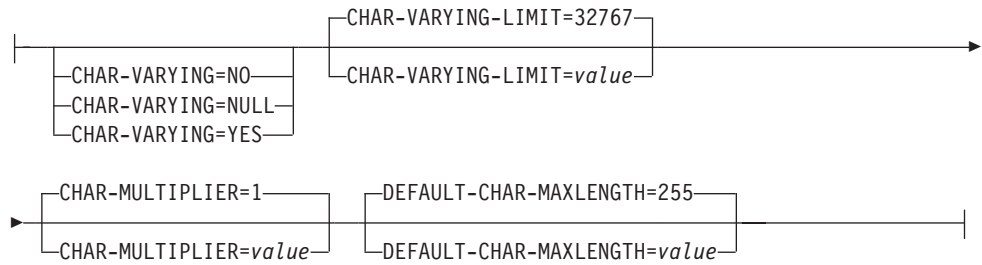
Therefore, you are advised to devise a naming convention and operating procedures that avoid this situation; for example, you can use the system symbolic parameter **SYSUID** to generate workspace file names that are unique to an individual user.

These temporary files are deleted before the end of the job.

Input parameters for DFHSC2LS



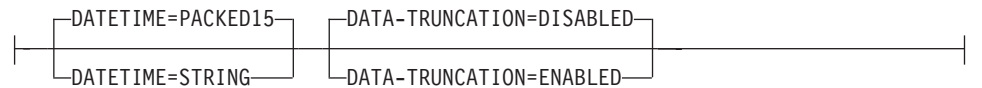
Advanced data mapping (mapping level 1.2 and higher):



Advanced data mapping (mapping level 2.1 and higher):



Advanced data mapping (mapping level 3.0):



Parameter use

- You can specify the input parameters in any order.
- Each parameter must start on a new line.
- A parameter (and its continuation character, if you use one) must not extend beyond column 72; columns 73 to 80 must contain blanks.
- If a parameter is too long to fit on a single line, use an asterisk (*) character at the end of the line to indicate that the parameter continues on the next line. Everything (including spaces) before the asterisk is considered part of the parameter. For example:

```
XSDBIND=xsdbinddir*  
/app1
```

is equivalent to

```
XSDBIND=xsdbinddir/app1
```

- A # character in the first character position of the line is a comment character. The line is ignored.

Parameter descriptions

BUNDLE=value

Specifies the path and name of the bundle directory on z/OS UNIX. If you specify this value, the XML assistant generates the XML binding in the bundle directory and creates a bundle manifest for you. The path information on this parameter overrides any path information on the **XSDBIND** parameter. If you do not specify this parameter, CICS places the XML binding in the specified location.

CCSID=*value*

Specifies the CCSID that is used at run time to encode character data in the application data structure. The value of this parameter overrides the value of the **LOCALCCSID** system initialization parameter. The *value* must be an EBCDIC CCSID that is supported by Java and z/OS conversion services. If you do not specify this parameter, the application data structure is encoded using the CCSID specified in the system initialization parameter.

You can use this parameter with any mapping level. However, if you want to deploy the generated files into a CICS TS 3.1 region, you must apply APAR PK23547 to achieve the minimum runtime level of code to install the Web service binding file.

CHAR-MULTIPLIER={1|*value*}

Specifies the number of bytes to allow for each character when the mapping level is 1.2 or higher. The *value* of this parameter can be a positive integer in the range of 1 to 2 147 483 647. All nonnumeric character-based mappings are subject to this multiplier. Binary, numeric, zoned, and packed decimal fields are not subject to this multiplier.

This parameter can be useful if, for example, you are planning to use DBCS characters where you might opt for a multiplier of 3 to allow space for potential shift-out and shift-in characters around every double-byte character at run time.

CHAR-VARYING={NO|NULL|YES}

Specifies how variable-length character data is mapped when the mapping level is 1.2 or higher. Variable-length binary data types are always mapped to either a container or a varying structure. If you do not specify this parameter, the default mapping depends on the language specified. You can select these options:

NO Variable-length character data is mapped as fixed-length strings.

NULL Variable-length character data is mapped to null-terminated strings.

YES Variable-length character data is mapped to a CHAR VARYING data type in PL/I. In the COBOL, C, and C++ languages, variable-length character data is mapped to an equivalent representation that consists of two related elements: the data-length and the data.

CHAR-VARYING-LIMIT={32767|*value*}

Specifies the maximum size of binary data and variable-length character data that is mapped to the language structure when the mapping level is 1.2 or higher. If the character or binary data is larger than the value specified in this parameter, it is mapped to a container and the container name is used in the generated language structure. The *value* can range from 0 to the default 32 767 bytes.

DATA-TRUNCATION={DISABLED|ENABLED}

Specifies if variable length data is tolerated in a fixed length field structure:

DISABLED

If the data is less than the fixed length that CICS is expecting, CICS rejects the truncated data and issues an error message.

ENABLED

If the data is less than the fixed length that CICS is expecting, CICS tolerates the truncated data and processes the missing data as null values.

DATETIME={PACKED15|STRING}

Specifies that `xsd:dateTime` fields are mapped to CICS ABSTIME data format or to text:

PACKED15

`xsd:dateTime` fields are mapped to CICS ABSTIME format.

STRING

`xsd:dateTime` fields are mapped to text. This mapping is the same as all previous mappings levels.

You can use this parameter at a mapping level of 3.0.

DEFAULT-CHAR-MAXLENGTH={255|*value*}

Specifies the default array length of character data in characters for mappings where no length is implied in the XML schema document or WSDL document, when the mapping level is 1.2 or higher. The *value* of this parameter can be a positive integer in the range of 1 to 2 147 483 647.

ELEMENTS={ALL|*value*}

Defines a list of global element local names to enable. The default value of ALL indicates that all global elements are enabled.

HTTPPROXY={*domain name*|*IP address*}:*port number*

If your XML schema or WSDL document contains references to other XML schema or WSDL files that are located on the Internet, and the system on which you are running DFHSC2LS uses a proxy server to access the Internet, specify either the domain name or IP address and the port number of the proxy server. For example:

```
HTTPPROXY=proxy.example.com:8080
```

In other cases, this parameter is not required.

HTTPPROXY-USERNAME=*value*

Specifies the HTTP proxy user name to be used with **HTTPPROXY-PASSWORD** if the system on which you are running DFHSC2LS uses an HTTP proxy server to access the Internet, and the HTTP proxy server uses basic authentication. You can use this parameter only when you also specify **HTTPPROXY**.

HTTPPROXY-PASSWORD=*value*

Specifies the HTTP proxy password to be used with **HTTPPROXY-USERNAME** if the system on which you are running DFHSC2LS uses an HTTP proxy server to access the Internet, and the HTTP proxy server uses basic authentication. You can use this parameter only when you also specify **HTTPPROXY**.

INLINE-MAXOCCURS-LIMIT={1|*value*}

Specifies whether inline variable repeating content is used based on the `maxOccurs` attribute of the XML attribute.

The **INLINE-MAXOCCURS-LIMIT** parameter is available only at mapping level 2.1 onwards. The *value* of **INLINE-MAXOCCURS-LIMIT** can be a positive integer in the range of 0 to 32 767. A value of 0 indicates that inline mapping is not used. A value of 1 ensures that optional elements are mapped inline. If the *value* of the `maxOccurs` attribute is greater than the *value* of **INLINE-MAXOCCURS-LIMIT**, container based mapping is used; otherwise, inline mapping is used.

When deciding if you want variably repeating lists to be mapped inline, consider the length of a single item of recurring data. If you have few instances of long length, container-based mapping is preferable; if you have many instances of short length, inline mapping is likely to be preferable.

LANG=COBOL

Specifies that the programming language of the high-level language structure is COBOL.

LANG=PLI-ENTERPRISE

Specifies that the programming language of the high-level language structure is Enterprise PL/I.

LANG=PLI-OTHER

Specifies that the programming language of the high-level language structure is a level of PL/I other than Enterprise PL/I.

LANG=C

Specifies that the programming language of the high-level language structure is C.

LANG=CPP

Specifies that the programming language of the high-level language structure is C++.

LOGFILE=*value*

The fully qualified z/OS UNIX name of the file into which DFHSC2LS writes its activity log and trace information. DFHSC2LS creates the file, but not the directory structure, if it does not already exist.

Typically you do not use this file, but it might be requested by the IBM service organization if you encounter problems with DFHSC2LS.

MAPPING-LEVEL={1.0|1.1|1.2|2.0|2.1|2.2|3.0}

Specifies the level of mapping for the assistant to use when generating the XML binding and language structures. You are recommended to use the most recent mapping level that is available; for DFHSC2LS, you are recommended to use a mapping level of 3.0 or higher.

3.0 The xsd:dateTime data type maps to the CICS ASKTIME format.

MINIMUM-RUNTIME-LEVEL={MINIMUM|3.0|CURRENT}

Specifies the minimum CICS runtime environment into which the XML binding can be deployed. If you select a level that does not match the other parameters that you have specified, you receive an error message. The options that you can select are as follows:

MINIMUM

The lowest possible runtime level of CICS is allocated automatically based on the parameters that you have specified.

3.0 Specify runtime level 3.0 or above if you want to use the CICS XML assistant and take advantage of advanced data mappings.

CURRENT

Use this runtime level to deploy the generated XML binding into a CICS region that has the same runtime environment as the region used to generate the XML binding.

OVERWRITE-OUTPUT=*value*

Specifies the namespace for CICS to use in the generated XML schema. For Atom feeds, CICS provides this namespace in the Atom feed together with the Atom namespace.

NO Any existing BUNDLE is not replaced. If an existing BUNDLE is found DFHLS2SC issues error message DFHPI9689E and terminates.

YES Any existing BUNDLE is replaced. If an existing BUNDLE is found then message DFHPI9683W is issued to inform you that the file has been replaced.

PDSCP=*value*

Specifies the code page that is used in the partitioned data set members, where *value* is a CCSID number or a Java code page number. If you do not specify this parameter, the z/OS UNIX System Services code page is used. For example, you might specify PDSCP=037.

PDSLIB=*value*

Specifies the name of the partitioned data set that contains the generated high-level language.

PDSMEM=*value*

Specifies the 1-6 character prefix that DFHSC2LS uses to generate the name of the partitioned data set member that will contain the high-level language structures.

DFHSC2LS generates a partitioned data set member for each operation. It generates the member name by appending a 2-digit number to the prefix.

SCHEMA=*value*

The fully qualified z/OS UNIX name of the file from which the XML schema is read. DFHSC2LS creates the file, but not the directory structure, if it does not already exist.

You can use either an XML schema or a WSDL document as input to DFHSC2LS. You must specify either this parameter or the **WSDL** parameter, but not both, to indicate from where the input comes.

STRUCTURE={PDSMEM_*value*|*data*}

The name of the top-level data structure in C and C++. The default value is the value of the **PDSMEM** parameter.

TYPES=*value*

Defines a list of global type local names to enable. A *value* of ALL indicates that all global types are enabled. By default, global types are not enabled.

WSDL=*value*

The fully qualified z/OS UNIX name of the WSDL document.

You can use either an XML schema or a WSDL document as input to DFHSC2LS. You must specify either this parameter or the **SCHEMA** parameter, but not both, to indicate from where the input comes.

XMLCP={LOCAL|UTF-8|EBCDIC-CP-US}

Specifies the code page that is used to generate the XML binding.

LOCAL

This value is the default. It specifies that the XML is generated using the local code page and no encoding tag is generated in the XML schema.

UTF-8 Specifies that the XML is generated using the UTF-8 code page. An encoding tag is generated in the XML schema. If you specify this option, you must ensure that the encoding remains correct when copying the XML schema between different platforms.

EBCDIC-CP-US

Specifies that the XML is generated using the US EBCDIC code page. An encoding tag is generated in the XML schema.

XSDBIND=*value*

The fully qualified z/OS UNIX name of the XML binding. DFHSC2LS creates the file, but not the directory structure, if it does not already exist. The file extension is `.xsdbind`.

Creating a Web service from multiple containers

In previous releases it was possible to create a Web service provider from a language structure with one container only, usually DFHWS-DATA. Now you can create a Web service provider from an application that uses many containers on a channel.

About this task

Containers can contain different data; for example, structured or unstructured binary content or text. Some containers might be optional and others mandatory. When a Web service uses many containers in its application interface, you must create a channel description document to describe the containers. The channel description document is an XML document that conforms to a schema that is provided by CICS.

When the application sends or receives a SOAP message, CICS uses the channel description to construct the multiple fragments of data into a single XML document and deconstruct the data back into a set of containers.

To create a Web service from an application that uses a channel and many containers, follow these steps:

Procedure

1. Create an XML document that describes the containers on the channel. If the application uses different sets of containers for the request and response, you must create a channel description document to describe the request and another document to describe the response. The channel schema and an example XML document are described in the *CICS Web Services Guide*.
2. Run DFHLS2WS with the following parameters:
 - a. Specify the **REQUEST-CHANNEL** parameter instead of the **PDSLIB** and **REQMEM** parameters. You must indicate where the channel description document for the request is located on z/OS UNIX in this parameter.
 - b. Optional: If the application sends or expects a response, specify the **RESPONSE-CHANNEL** parameter. You must indicate where the channel description document for the response is located on z/OS UNIX in this parameter. If the application uses the same set of containers for the response, specify the same value as the **REQUEST-CHANNEL** parameter.
 - c. Specify 3.0 as the value of the **MAPPING-LEVEL** parameter.
 - d. Specify CHANNEL as the value of the **PGMINT** parameter.

You must not use the **CONTID** parameter. The Web services assistant generates a WSDL document that includes the channel schema.

3. Copy the Web service binding and WSDL document to the pickup directory of the provider pipeline. You must copy the files in binary mode.
4. Create the CICS resources using the **PIPELINE SCAN** command.

Results

When you have successfully created the CICS resources, the creation of your service provider application is complete.

What to do next

Test your application to make sure that it can send and receive SOAP messages successfully.

Changes to the Web services assistant

The Web services assistant has new and changed parameters to support the data mappings enhancements.

The **MAPPING-LEVEL** parameter has a new mapping level of 3.0. Use mapping level 3.0 to take advantage of these improvements:

- DFHLS2WS and DFHWS2LS have a new **DATETIME** parameter to support the mapping of `xsd:dateTime` data types to CICS ABSTIME format.
- DFHLS2WS and DFHWS2LS have a new **DATA-TRUNCATION** parameter to support variable length data in a fixed length field.
- DFHLS2WS has two new parameters, **REQUEST-CHANNEL** and **RESPONSE-CHANNEL**, to support creating a Web service provider application that uses multiple containers on a channel to create the request and response SOAP message.

The **MINIMUM-RUNTIME-LEVEL** parameter also has a new value of 3.0 for deploying Web services into CICS TS for z/OS, Version 4.1 regions.

Changes to external interfaces

Some changed external interfaces now support the improved data mappings provided by the CICS XML assistant.

Changes to the CICS application programming interface

Two new API commands, **TRANSFORM XMLTODATA** and **TRANSFORM DATATOXML**, have been added. Use these commands to convert XML to application data and vice versa.

New commands

TRANSFORM XMLTODATA

Use the **TRANSFORM XMLTODATA** command to convert XML to application data.

For details of the command, see “TRANSFORM XMLTODATA” on page 270.

TRANSFORM DATATOXML

Use the **TRANSFORM DATATOXML** command to convert application data to XML.

For details of the command, see “TRANSFORM DATATOXML” on page 267.

Changes to the JCICS application programming interface

The JCICS API now supports transformations between XML and binary data from a Java application.

The JCICS API provides equivalents for the following new API commands:

TRANSFORM XMLTODATA
TRANSFORM DATATOXML

The support is provided by the following new JCICS classes:

com.ibm.cics.server.XmlTransform

This class represents the XMLTRANSFORM resource.

com.ibm.cics.server.Transform

This class provides the Java interface to the **TRANSFORM** commands.

com.ibm.cics.server.TransformInput

This class represents the input parameters for the **TRANSFORM** commands.

com.ibm.cics.server.TransformOutput

This class represents the output parameters for the **TRANSFORM** commands.

Changes to the system programming interface

New system commands support the data mapping improvements in CICS.

New commands

INQUIRE XMLTRANSFORM

Use INQUIRE XMLTRANSFORM to retrieve information about an installed XMLTRANSFORM resource.

For details of the command, see Appendix C, “New system programming commands,” on page 305.

SET XMLTRANSFORM

Use SET XMLTRANSFORM to change the status of an installed XMLTRANSFORM resource.

For details of the command, see Appendix C, “New system programming commands,” on page 305.

Changes to CICSplex SM resource tables

New and changed resource tables support the data mapping improvements in CICS.

New tables

The following new resource tables support the XMLTRANSFORM resource:

XMLTRANS

An operations table for installed XMLTRANSFORM resources.

CRESEXMLT

A topology table for installed XMLTRANSFORM resources.

Changed tables

The TASK and HTASK resource tables have new attributes for monitoring XMLTRANSFORM resources:

Attribute	Description
TMRMLXTC	The number of times that the XML binding file has been used for data transformation

Attribute	Description
TMRMLTDL	The length of the XML document
TMRMLCTM	The XML system services parser CPU time to process the request

The CICS[®]RG[®]N and MONITOR resource tables have new attributes to support XMLTRANSFORM statistics reporting.

Changes to CICSplex SM views and menus

A new view set and changed views support the improvements to data mappings.

New menu, Applications

A new operations menu, EYU[®]STARTAPPLIC[®]TN, is available by selecting **CICS operations views > Applications** in the navigation of the Web User Interface. This menu provides a summary of application-related resources. The menu links to the new EYU[®]STARTXMLTRANS view set.

New views

XMLTRANSFORM resources - XMLTRANS

To access this view set from the main menu, click **Application operations views > XMLTRANSFORM resources**.

The views in this view set show the installed XMLTRANSFORM resources in the selected scope. The view shows if the XMLTRANSFORM resource was dynamically created by an application bundle and if validation is switched on.

The detailed view of an XMLTRANSFORM resource shows information such as the location of the XML binding and schema, the mapping level and minimum runtime level, as well as the resource signature.

Changed views

The following views now support the XMLTRANSFORM resource:

- The CICS[®]RG[®]N statistics view now includes an option to request XMLTRANSFORM statistics. Navigate to this view as follows: **CICS operations views > CICS region operations views > Request statistics processing**
- The Monitor statistics view now includes an option to request XMLTRANSFORM statistics. From the CICS regions view, click **EYU[®]STARTCICS[®]RG[®]N.DETAILED > Monitoring and statistics details > Statistics details > Request statistics processing**.

Changes to global user exits

The resource management install and discard exit XRSINDI supports the new XMLTRANSFORM resource.

For an XMLTRANSFORM resource, the **UEPIDTYP** parameter returns a value of UEIDXMLT.

Changes to statistics

CICS now collects statistics on applications that transform data between language structures and XML.

New statistics type, XMLTRANSFORM

A new XMLTRANSFORM option on the **PERFORM STATISTICS RECORD** and **CEMT PERFORM STATISTICS** commands records statistics for applications that transform data between language structures and XML.

A new RESTYPE value of XMLTRANSFORM for the **EXEC CICS EXTRACT STATISTICS** command retrieves statistics relating to XML transformations.

The DFHMLRDS DSECT maps the application bundle statistics.

The statistics utility program DFHSTUP includes the new statistics for XML transformations and the sample statistics extract program DFH0STAT formats the new statistics.

Changes to security

Resource and command security apply to XMLTRANSFORM resources if these functions are enabled for the CICS region.

Resource security

Resource security for XMLTRANSFORM 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. You must supply a prefix of XMLTRANSFORM to the name of the XMLTRANSFORM resource.

Command security

Command security for XMLTRANSFORM resources uses the XMLTRANSFORM resource in the CCICSCMD class or the VCICSCMD grouping class.

Chapter 10. Support for Web Services Addressing

Web Services Addressing (WS-Addressing) is a transport-neutral mechanism for passing messaging information between Web services, so that Web services using different protocols can interoperate.

CICS ensures that your existing Web service applications can accept requests from Web services that use WS-Addressing. You can also create new Web services that use endpoint references and message addressing properties in SOAP messages.

WS-Addressing adds addressing information, in the form of Message Addressing Properties (MAPs), to SOAP message headers. MAPs include messaging information, such as a unique message ID and endpoint references that detail where the message came from, where the message is going to, and where reply or fault messages are to be sent. An endpoint reference (EPR) is a specific type of MAP, which includes the destination address of the message, optional reference parameters for use by the application, and optional metadata.

You can use new API commands to create and manipulate addressing contexts. Addressing contexts are used to store WS-Addressing MAPs before and after SOAP requests and responses are performed.

CICS includes the following features to support WS-Addressing:

- Your Web service requester and provider applications can interact with other services that are using WS-Addressing without requiring you to redeploy them. A new message handler, the addressing message handler DFHWSADH, in the pipeline routes messages that contain WS-Addressing information to the specified Web service.
- You can write an application that uses the WS-Addressing API commands to create an endpoint reference and to create, update, delete, and query an addressing context.
- You can route response messages to endpoints other than the requester endpoint; for example, you can route fault messages to a dedicated fault handler.
- You can pass reference parameters to applications as part of the MAPs in the SOAP header.

By default, CICS supports the recommendation specifications:

- W3C WS-Addressing 1.0 - Core
- W3C WS-Addressing 1.0 - SOAP Binding
- W3C WS-Addressing 1.0 - Metadata

These specifications are identified by the <http://www.w3.org/2005/08/addressing> namespace. Unless otherwise stated, WS-Addressing semantics that are described in this documentation refer to the recommendation specifications.

For interoperability, CICS also supports the submission specification:

- W3C WS-Addressing- Submission

This specification is identified by the <http://schemas.xmlsoap.org/ws/2004/08/addressing> namespace. Use the submission specification only if you must interoperate with a client or Web service provider that implements the submission specification.

Terminology

Terms used to explain Web Services Addressing (WS-Addressing) support.

addressing context

An XML document that stores WS-Addressing message addressing properties (MAPs) before they are sent in SOAP request messages and after they are received from SOAP request and response messages.

endpoint reference (EPR)

An XML structure containing addressing information that is used to route a message to a Web service. This addressing information includes the destination address of the message, optional reference parameters for use by the application, and optional metadata.

message addressing property (MAP)

An XML element that conveys addressing information for a specific Web service message, such as a unique message ID, the destination of the message, and the endpoint references of the message.

Changes to external interfaces

Some external interfaces now support Web Services Addressing.

Changes to the CICS application programming interface

Web Services Addressing (WS-Addressing) introduces new commands to manipulate addressing contexts and endpoint references (EPRs).

New commands

WSACONTEXT BUILD

Use the **WSACONTEXT BUILD** command to insert or replace WS-Addressing message addressing properties (MAPs) in the addressing context.

For details of the command, see “WSACONTEXT BUILD” on page 279.

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.

For details of the command, see “WSACONTEXT GET” on page 287.

WSACONTEXT DELETE

Use the **WSACONTEXT DELETE** command to delete the addressing context.

For details of the command, see “WSACONTEXT DELETE” on page 287.

WSAEPR CREATE

Use the **WSAEPR CREATE** command to create an endpoint reference (EPR) to represent a Web service or Web service resource.

For details of the command, see “WSAEPR CREATE” on page 284.

Changes to the JCICS application programming interface

The JCICS API now supports the handling of addressing contexts and endpoint references from a Java application.

The JCICS API provides equivalents for the following new API commands:

WSACONTEXT BUILD
WSACONTEXT DELETE
WSACONTEXT GET
WSAEPR CREATE

The support is provided by the following new JCICS classes:

com.ibm.cics.server.WSAContext

This class implements methods to set and get data from the addressing contexts that are attached to the current Web service requester or provider application.

com.ibm.cics.server.WSAEpr

This class creates a WS-Addressing endpoint reference from supplied data.

Changes to monitoring data

New monitoring fields monitor the number of WS-Addressing requests issued by the user task.

Changed performance class group, DFHWEBB

The DFHWEBB class group contains the following new fields:

420 (TYPE-A, 'WSACBLCT', 4 BYTES)

The number of **EXEC CICS WSACONTEXT BUILD** commands issued by the user task.

421 (TYPE-A, 'WSACGTCT', 4 BYTES)

The number of **EXEC CICS WSACONTEXT GET** commands issued by the user task.

422 (TYPE-A, 'WSAEPCT', 4 BYTES)

The number of **EXEC CICS WSAEPR CREATE** commands issued by the user task.

423 (TYPE-A, 'WSATOTCT', 4 BYTES)

The total number of **EXEC CICS WS-Addressing** commands issued by the user task.

Changes to problem determination

New information helps you to diagnose problems that relate to WS-Addressing.

New messages

New messages, DFHPI9671 to DFHPI9681, provide information about problems in the WSDL with WS-Addressing options when processed by DFHWS2LS.

New trace points

New trace points in the ranges PI 1C00 to PI 1C07, PI 1F00 to PI 1F07, and PI 1F09 to PI 1F0C are issued by the Pipeline Manager domain.

Chapter 11. Support for the IBM SDK for z/OS, Java Technology Edition, Version 6

CICS TS 4.1 supports the JVM provided by the 31-bit version of the IBM SDK for z/OS, Java Technology Edition, Version 6. With Java 6, you continue to benefit from the new Java language features, improved garbage collection, and simpler class sharing, first introduced in Java 5.

CICS TS 4.1 does not support the 64-bit version of the IBM SDK for z/OS, Version 6.

Changes to CICS externals

Some CICS external interfaces have changed to work with Java 6.

Changes to installation

You must upgrade your CICS regions to use Java 6. To do so, change the JAVADIR parameter in the DFHISTAR job.

The default for the JAVADIR parameter, which specifies the default installation location for the IBM Software Developer Kit for z/OS, Java Technology Edition Version 6, is `java/J6.0`.

Changes to JVM profiles

The DFHJVMCC, DFHJVMPC, and DFHJVMPS JVM profiles are not required in Java 6 and samples are not supplied. In Java 6, system properties files are optional and sample properties files are not provided.

JVM profiles: options and samples

CICS provides sample JVM profiles that contain a selection of options for IBM JVMs that are used in a CICS environment. Some of these options are specific to the CICS environment and are not used for JVMs in other environments. Other options are standard or nonstandard Java options, which can be used for IBM JVMs in any environment.

You can specify any JVM option or system property in a JVM profile, and it is passed to the JVM. No central repository of all options and system properties for the JVM exists. Here are some recommended sources of information:

- The documentation for the IBM SDK for z/OS, Java Technology Edition, Version 6.
- The *IBM Developer Kit and Runtime Environment, Java Technology Edition Diagnostics Guide*, which is available to download from www.ibm.com/developerworks/java/jdk/diagnosis/. This guide documents system properties that are used for JVM trace and problem determination.

The Java class libraries include other system properties, and applications might have their own system properties. With all options or system properties available for the IBM JVM that are not specific to the CICS environment, regard the IBM JVM documentation as the primary source of information and the CICS documentation as a secondary source of information.

You can specify any JVM options and system properties in a JVM profile, and it is not necessary to have a separate JVM profile and JVM properties file. From CICS TS 4.1 onwards, sample JVM properties files are not supplied.

The summary table, Table 1, lists options used in the CICS-supplied sample JVM profiles and some further options that you might use to complete tasks described in the CICS documentation. The table indicates the default for each option if it is not specified in the JVM profile.

Version 6 of the SDK does not use a master JVM, so the JVM options for a JVM that uses the shared class cache and a JVM that does not are the same, except for the CLASSCACHE option.

Table 1. JVM options reference table for JVMs in a CICS environment

Option	Default	Comments
<i>JVM type</i>		
CLASSCACHE	NO	YES makes JVM use shared class cache, NO does not
REUSE	YES	YES makes continuous, NO makes single-use
<i>Directories</i>		
CICS_HOME	None	Required, sample profiles include this directory
JAVA_HOME	None	Required, sample profiles include this directory
WORK_DIR	/tmp	
<i>Class paths</i>		
CLASSPATH_SUFFIX	None	
LIBPATH_SUFFIX	None	
<i>Timeout threshold</i>		
IDLE_TIMEOUT	30 minutes	Continuous only
THREAD_TIMEOUT	60 seconds	Used in a JVM server profile only.
<i>Further settings and facilities for the JVM</i>		
JVMPROPS	None	
INVOKE_DFHJVMAT	NO	Single-use only
<i>Initialization classes</i>		
SETUP_CLASSES	None	Used in a JVM server profile only.
TERMINATION_CLASSES	None	Used in a JVM server profile only.
<i>Storage heap sizes</i>		
-Xms		For information on the -Xms default value, see the reference information at http://www.ibm.com/servers/eserver/zseries/software/java/reference

Table 1. JVM options reference table for JVMs in a CICS environment (continued)

Option	Default	Comments
-Xmx		For information on the -Xmx default value, see the reference information at http://www.ibm.com/servers/eserver/zseries/software/java/reference
<i>Garbage collection threshold</i>		
GC_HEAP_THRESHOLD	85%	Continuous only
<i>Output from the JVM</i>		
LEHEAPSTATS	NO	
STDERR	dfhjvmerr	
STDIN	dfhjvmin	
STDOUT	dfhjvmout	
USEROUTPUTCLASS	None	
<i>Problem determination and application debugging</i>		
JAVA_DUMP_OPTS	YES	
USE_LIBJVM_G	NO	
-Xdebug	NO	
PRINT_JVM_OPTIONS	NO	Set YES only temporarily
DISABLE_EJB	NO	Disables CICS provided EJB support

UNIX System Services environment variables

In addition to the JVM options and system properties that are used to construct the JVM, you can specify any UNIX System Services environment variables in a JVM profile. Any name and value pair in a JVM profile that is not recognized as a JVM option or system property is treated as a UNIX System Services environment variable and is exported. UNIX System Services environment variables specified in a JVM profile apply only to JVMs created with that profile.

The JAVA_DUMP_OPTS and JAVA_DUMP_TDUMP_PATTERN options used in the CICS-supplied sample JVM profiles are UNIX System Services environment variables. Another example is the TZ environment variable, which can be specified to change the time zone for the JVM.

UNIX System Services environment variables can be specified only in a JVM profile, not in a JVM properties file.

Changes to the CICS sample JVM profiles.

Sample profile DFHJVMPR now uses the shared class cache of the JVM.

The sample JVM profiles DFHJMCC, DFHJMPC, and DFHJMPS are not required in Java 6 and samples are not supplied. Use DFHJVMPR as the template for your JVM profiles.

Changes to problem determination

There are new trace points for Java 6.

Trace

The new trace points for Java 6 are as follows:

- SJ0213
- SJ0214
- SJ0308
- SJ0507
- SJ0519
- SJ051A
- SJ051D
- SJ060B
- SJ060C

With Java 6, the JVM trace point ID is in the format *TPID_componentId.number*, where *componentId* is the name of the JVM component that issued the trace point, and *number* is the unique identifying number in the component. This format is variable length.

Version 6 of the IBM SDK for z/OS, Java Technology Edition makes other significant changes to the format of JVM trace points and the ways in which you can select them for tracing. The topics about tracing Java applications and the JVM in the Version 6 *IBM Developer Kit and Runtime Environment, Java Technology Edition Diagnostics Guide*, which is available to download from www.ibm.com/developerworks/java/jdk/diagnosis/, has the latest information.

Chapter 12. Support for Service Component Architecture

CICS provides support for applications that conform to the Service Component Architecture (SCA) specifications. SCA supports rapid deployment of new applications to meet changing business requirements by promoting the reuse of existing application assets in a component model.

In SCA, applications are composed of services, where each service is described using one or more components. These components are assembled together into a *composite application* that can participate in a service-oriented architecture (SOA) as part of a business service. A *business service* is a service that is aligned with business processes and models rather than a technical implementation.

The SCA component model is flexible, so you can reuse the same components in different composite applications without needing to change the underlying business logic.

To create a composite application, use the SCA tooling in Rational Developer for System z. The SCA tooling deploys composite applications to CICS as bundles and can be enabled and disabled using the `BUNDLE` resource.

You can create two types of service from your CICS applications:

Channel-based services

These services use the Service Component Architecture (SCA) support to describe applications as components. The interface for these services is a channel. Channel-based services can be called only by other CICS applications using the **INVOKE SERVICE** API command.

XML-based services

These services are typically Web service provider or requester applications that use XML to interface with other applications and use a binding to transform the data. Web service applications can also be described as components using SCA. The interface for these services is XML with a binding to transform the data. XML-based services can be called by other CICS applications using the **INVOKE SERVICE** API command or by an external client.

An XML-based service can also be an application that uses the **TRANSFORM** API commands to map application data to and from XML. The XML assistant uses a language structure or XML schema to generate the XML binding and also create a bundle.

Both types of service use the pipeline support in CICS.

Terminology

New terms are introduced to explain the support for Service Component Architecture.

business service

An abstract representation of a business function, hiding the specifics of the function interfaces.

component

A reusable object or program that performs a specific function and is designed to work with other components and applications.

component name

The external name of a component. It is used, for example, in tooling and commands. Each component requires a name.

composite

A Service Component Architecture (SCA) element that contains components, services, references, and wires that connect them.

composite application

An application that is created by combining more than one existing functions together. A composite application can also include new components.

Service Component Architecture (SCA)

An architecture in which all elements of a business transaction, such as access to Web services, Enterprise Information System (EIS) service assets, business rules, work flows, and databases, are represented in a service-oriented way.

Service Component Architecture (SCA)

Service Component Architecture (SCA) is a set of specifications that describe a programming model for building applications and systems using a Service-Oriented Architecture (SOA). SCA extends and complements previous approaches to implementing services and builds on open standards such as Web services.

The specifications describe how to create *composite applications*. A composite application is created by combining one or more components that together implement the business logic of the new application. A *component* comprises an application program that implements the business logic and configuration information. An application developer can use the same application program with different configurations to form different components. A component offers a service to other components and in turn consumes functions offered by other services using service-oriented interfaces.

An application developer can assemble components together to create a solution for a particular business requirement. A composite application can contain both new components that are created specifically for the business application and existing components that are reused from other applications.

CICS supports the *SCA Assembly Model 1.0* specification, which describes how service components can be assembled to form *composites*. A composite is the unit of deployment in SCA and is described in an XML language called SCDL. Composites can contain components, services, references, property declarations, and the wiring that describes the connections between these elements. Composites can also be used in components with other composites, allowing for a hierarchical construction of composite applications, where high-level services are implemented internally by sets of lower-level services.

Structure of a component

A simple type of component has one service and one reference. A *service* is an addressable interface for the component that can contain one or more operations. A

reference is a dependency on a service that is provided by another component. The bindings for the component can be defined in both the service and the reference:

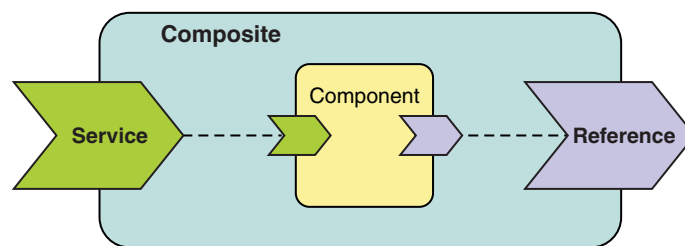
- Component services use bindings to describe the access mechanism that other internal components have to use to call the component.
- Component references use bindings to describe the access mechanism that is used to call other components.

Component services and references are internal and are used only for component-to-component communication. To create an external interface, the component must be deployed inside a composite. A very simple composite has one external service and one external reference:

- Composite services use bindings to describe the access mechanism that external clients must use to call the service.
- Composite references use bindings to describe the access mechanism that is used to call another service.

Composite services and references are not part of the component itself, but are a boundary between the component and other external clients or services.

The following diagram shows a composite that contains a simple component with both internal and external services and references.



Enhancements to pipeline processing

The requester mode pipeline processing now supports channel-based services. All channel-based services use a new URI format that begins with `cics://`. This URI format determines the processing that occurs in the requester pipeline.

When an application calls a channel-based service using the **INVOKE SERVICE** command, CICS uses a requester mode pipeline to handle the request. The URI that the application puts in the container on the channel determines the pipeline processing, unless it is overridden by a message handler. When the service requester pipeline reaches the end of its processing, you have the following options:

Linking to a program

If you change the URI to the format `cics://PROGRAM/program`, where *program* is the name of the target application program, CICS passes the current channel and its containers or `COMMAREA` to the program using an **EXEC CICS LINK** command. The `COMMAREA` length must not exceed 32 763.

This processing is similar to the local optimization that occurs when the service requester and service provider applications are in the same CICS region. However, using this URI format provides the benefit of running the

request through the pipeline and any custom message handlers first. The target application program must be able to handle the contents of the containers or COMMAREA.

Starting another requester mode pipeline

If you change the URI to the format `cics://PIPELINE/pipeline?targetServiceUri=targetServiceUri`, where *pipeline* is the name of a PIPELINE resource and *targetServiceUri* is the URI that you want to put in the DFHWS-URI container, CICS passes the current channel and its containers to the specified requester pipeline. Use this URI when you want to link two or more requester pipelines together before sending the request to the service provider. The number of requester pipelines that you can chain together is not limited.

Sending the request straight to the provider mode pipeline

If you change the URI to the format `cics://SERVICE/service?targetServiceUri=targetServiceUri`, where *service* is the name of the target service and *targetServiceUri* is the path to the service, CICS resolves the request by matching the path to a URIMAP and passes the request to the correct provider pipeline. Use this option when you want to take advantage of processing the request through both the requester and provider pipelines without using the network.

This URI might also be useful where the requester and provider applications are written in different languages, or use different mapping levels, and expect different binary data.

You can also specify parameters on each type of URI that provide CICS with additional information to handle the request; for example, the request can run under a new transaction ID or user ID. See the *CICS Web Services Guide* for details.

CICS does not require you to specify message handlers in a requester mode pipeline for channel-based services. However, you can write message handlers to override the URI in the pipeline if you want the system, rather than the application, to control what processing takes place.

XML-based services and new URI types

XML-based services, such as Web services, can also use URIs that begin with `cics://`. The data transformations to turn the binary data into XML take place as usual, the pipeline runs through the message handlers, and the pipeline transport handler links to another program or starts a pipeline depending on what type of URI is specified.

For example, if a Web service requester and provider application are both in CICS, you can use the `cics://SERVICE/service` URI to avoid using the network, but still run the request through the provider pipeline. This technique might be useful if the requester and provider applications are written in different languages or use different mapping levels and expect different binary data. By using this URI, you have the advantage of the pipeline processing without the overhead of sending the request through the network.

Changes to external interfaces

Some changed external interfaces now support the deployment and management of SCA applications.

Changes to the CICS application programming interface

CICS applications can now call different types of services using the **INVOKE SERVICE** command.

New command, INVOKE SERVICE

Use the **INVOKE SERVICE** command to enable an application as a Web service requester or to call an SCA component.

For details of the command, see **INVOKE SERVICE**.

Changed command, INVOKE WEBSERVICE

The **INVOKE WEBSERVICE** command is retained for compatibility with existing Web service requester applications. It is a synonym of the **INVOKE SERVICE** command and has all of the same options. For all new Web service applications, use the **INVOKE SERVICE** command.

Part 3. Comply

CICS Transaction Server for z/OS, Version 4 Release 1 delivers a set of capabilities represented in the following themes:

- Compete
- Comply
- Control
- Architectural enhancements

The capabilities represented by the *Comply* theme help ensure and demonstrate effective management control over business applications and IT facilities.

Chapter 13. Improvements in management of resource definitions

The resource signature, which combines the installation and definition signatures, provides specific information for resource definitions that were installed or changed in CICS TS 4.1 or later. You can use these improved details to detect resource modifications for auditing, tracking, or for fixing problems.

The installation signature captures details as to when, how, and by whom the resource was installed. The definition signature captures details in the CICS system definition data set or the CICSplex SM EYUDREP data repository, as to when, how, and by whom the resource was defined and the last change made to the resource definition.

All resource types support the definition signature. The resource types that support the installation signature are: ATOMSERVICE, BUNDLE, CONNECTION, CORBASERVER, DB2CONN, DB2ENTRY, DB2TRAN, DJAR, DOCTEMPLATE, ENQMODEL, EVENTBINDING, FILE, IPCONN, JOURNALMODEL, JVMSERVER, LIBRARY, MQCONN, MQINI, PIPELINE, PROFILE, PROCESSTYPE, PROGRAM, REQUESTMODEL, TCPIPSERVICE, TDQUEUE, TRANCLASS, TRANSACTION, TSMODEL, URIMAP, WEBSERVICE, and XMLTRANSFORM.

The definition signature for resource definitions

The definition signature captures details about when, how, and by whom each resource is defined or changed in the CSD file or in the CICSplex SM EYUDREP data repository. The definition signature is updated each time a change is made to the resource. You can use these details to detect resource modifications for auditing or for fixing problems.

The definition signature is displayed in the CICS Explorer views, on CEDA and CEMT panels, CICSplex SM BAS views, EXEC CICS INQUIRE commands, and in DFHCSDUP reports. These are the definition signature fields:

DEFINESOURCE

The source of the resource definition. The **DEFINESOURCE** value depends on the **CHANGEAGENT**.

DEFINETIME

The time stamp when the resource definition was created using the **DEFINE**, **USERDEFINE**, **COPY**, **MOVE**, or **RENAME** commands. When you alter an existing resource using the **ALTER** command, the value specified by **DEFINETIME** does not change. On CEDA panels, the date is displayed in the format that you specified in the **DATFORM** system initialization parameter.

CHANGEAGENT

How the resource was defined or last modified, by using one of these methods:

Autoinstall

Autoinstall

Csdapi

CEDA, the programmable interface to DFHEDAP, or EXEC CICS CSD command

Csdbatch

DFHCSDUP

Drepapi

CICSplex SM BAS API command

Dynamic

The resource was generated by:

A PIPELINE scan (URIMAP or WEBSERVICE).

CICS Web template management, using DFHWBTL or DFHWBMS (DOCTEMPLATE).

The installation of a DB2ENTRY resource definition with transid specified (DB2TRAN).

The installation of an ATOMSERVICE resource definition with XSDBIND specified (XMLTRANSFORM).

The installation of an MQCONN resource definition with INITQNAME specified (MQINI).

The installation of a CORBASERVER resource definition with autopublish specified (DJAR).

System

CICS or CICSplex SM system

Table Table definition

CHANGEAGREL

The level of the CICS system used for the definition of, or last modification to, the resource definition.

CHANGETIME

The time stamp when the resource definition was last modified. When the resource is first defined, the CHANGETIME value is identical to the DEFINETIME value. On CEDA panels, the date is displayed in the format that you specified in the DATFORM system initialization parameter.

CHANGEUSRID

The ID of the user who defined or last modified the resource definition.

To display the definition signature for an individual resource, or a group of resources, in the CEDA DISPLAY and EXPAND GROUP panels press PF2. To return to the previous CEDA command panel, press PF2 again.

To display a summary of the definition signatures for all the specified resources, add the **SIGSUMM** parameter to the **DFHCSDUP LIST** command. The definition signature fields are displayed with the resource attributes when you use the **OBJECTS** option on the command. The **DFHCSDUP EXTRACT** command also extracts the definition signature fields from the CSD file.

Resources defined in CICS releases before CICS TS 4.1 do not have information displayed for the definition signature until they are modified in this CICS release or later. When the resource is modified, the DEFINETIME field remains blank.

The installation signature for resource definitions

The installation signature shows when, how, and by whom each resource is installed.

The installation signature is displayed in the CICS Explorer views, the CICSplex SM Operations views, on the expanded view panel of the **CEMT INQUIRE** command for the resource, or you can use an **EXEC CICS INQUIRE** command. These are the installation signature fields:

INSTALLAGENT

How the resource was installed, by using one of these methods:

Autoinstall

Autoinstall

Bundle

Bundle deployment

Createspi

EXEC CICS CREATE command

Csdapi

CEDA, the programmable interface to DFHEDAP, or EXEC CICS CSD command

Dynamic

The installed resource was generated by:

A PIPELINE scan (URIMAP or WEBSERVICE).

CICS Web template management, using DFHWBTL or DFHWBMS (DOCTEMPLATE).

The installation of a DB2ENTRY resource definition with transid specified (DB2TRAN).

The installation of an ATOMSERVICE resource definition with XSDBIND specified (XMLTRANSFORM).

The installation of an MQCONN resource definition with INITQNAME specified (MQINI).

The installation of a CORBASERVER resource definition with autopublish specified (DJAR).

Grplist

GRPLIST INSTALL

System

CICS or CICSplex SM system

Table Table definition

INSTALLTIME

The time when the resource was installed.

INSTALLUSRID

The ID of the user who installed the resource.

Changes to external interfaces

Some external interfaces now support the resource signature for resource definitions.

Changes to the system programming interface

The INQUIRE command supports the resource signature for a number of resource types.

The resources types that support the resource signature are: ATOMSERVICE, BUNDLE, CONNECTION, CORBASERVER, DB2CONN, DB2ENTRY, DB2TRAN, DJAR, DOCTEMPLATE, ENQMODEL, EVENTBINDING, FILE, IPCONN, JOURNALMODEL, JVMSERVER, LIBRARY, MQCONN, MQINI, PIPELINE, PROFILE, PROCESSTYPE, PROGRAM, REQUESTMODEL, TCPIPSERVICE, TDQUEUE, TRANCLASS, TRANSACTION, TSMODEL, URIMAP, WEBSERVICE, and XMLTRANSFORM.

Changed command, INQUIRE

New options have been added to the **INQUIRE** command to support the definition and installation signatures.

The list of possible values for **CHANGEAGENT** and **INSTALLAGENT** can vary and depends on the resource type. For details of a specific INQUIRE command see the *CICS System Programming Reference*

CHANGEAGENT (cvda)

Returns a CVDA value identifying the agent that made the last change to the resource definition. The possible values are as follows:

AUTOINSTALL

The resource was autoinstalled.

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.

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 definition was last changed by a PIPELINE scan.

DYNAMIC

The resource was defined as a result of the installation of a DB2ENTRY with TRANSID specified.

DYNAMIC

The resource was defined by the CICS scanning mechanism.

DYNAMIC

The resource was defined by the CICS system for a template being used through the CICS template manager, DFHWBTL.

DYNAMIC

The resource was defined by an ATOMSERVICE resource.

DYNAMIC

The resource was defined as a result of a MQCONN resource definition with INITQNAME specified.

SYSTEM

The resource definition was last changed by the CICS or CICSplex system.

TABLE

The resource definition was last changed by a table definition.

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.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

DEFINETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

DEFINESOURCE (*data-area*)

Returns the source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT. For details, see the *CICS Resource Definition Guide*.

INSTALLAGENT (*cvda*)

Returns a CVDA value identifying the agent that installed the resource. The possible values are as follows:

AUTOINSTALL

The resource was autoinstalled.

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.

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.

DYNAMIC

The resource was installed using a PIPELINE scan.

DYNAMIC

The resource was installed as a result of the installation of a DB2ENTRY with TRANSID specified.

DYNAMIC

The resource was installed by the CICS scanning mechanism.

DYNAMIC

The resource was installed by the CICS system for a template being used through the CICS template manager, DFHWBTL.

DYNAMIC

The resource was installed as a result of the installation of a MQCONN with INITQNAME specified.

DYNAMIC

The resource was installed by an ATOMSERVICE resource.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

SYSTEM

The resource was installed by the CICS or CICSplex SM system.

TABLE

The resource was installed using a table definition.

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.

Changes to CEMT

The CEMT INQUIRE command supports the improvements in management of resource definitions for a number of resource types.

The resources types that support the resource signature are: ATOMSERVICE, BUNDLE, CONNECTION, CORBASERVER, DB2CONN, DB2ENTRY, DB2TRAN, DJAR, DOCTEMPLATE, ENQMODEL, EVENTBINDING, FILE, IPCONN, JOURNALMODEL, JVMSERVER, LIBRARY, MQCONN, MQINI, PIPELINE, PROFILE, PROCESSTYPE, PROGRAM, REQUESTMODEL, TCPIPSERVICE, TDQUEUE, TRANCLASS, TRANSACTION, TSMODEL, URIMAP, WEBSERVICE, and XMLTRANSFORM.

New options on INQUIRE command

The definition and installation signatures are displayed on the expanded view panel of the supported resource types. The new fields are displayed at the end of the list of attributes for the installed resource.

The list of possible values for CHANGEAGENT and INSTALLAGENT can vary and depends on the resource type. For details of a specific INQUIRE command see *CICS Supplied Transactions*.

CHANGEAGENT (*value*)

Displays a value identifying 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.

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.

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 definition was last changed by a PIPELINE scan.

DYNAMIC

The resource was defined as a result of the installation of a DB2ENTRY with TRANSID specified.

DYNAMIC

The resource was defined by the CICS scanning mechanism.

DYNAMIC

The resource was defined by the CICS system for a template being used through the CICS template manager, DFHWBTL.

DYNAMIC

The resource was defined as a result of a MQCONN resource definition with INITQNAME specified.

DYNAMIC

The resource was defined by an ATOMSERVICE resource.

SYSTEM

The resource definition was last changed by the CICS or CICSplex system.

TABLE

The resource definition was last changed by a table definition.

TABLE

The resource definition was last changed by a File Control table definition.

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 in the *CICS 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 identifying 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:

AUTOINSTALL

The resource was autoinstalled.

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.

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.

DYNAMIC

The resource was installed using a PIPELINE scan.

DYNAMIC

The resource was installed as a result of the installation of a DB2ENTRY with TRANSID specified.

DYNAMIC

The resource was installed by the CICS scanning mechanism.

DYNAMIC

The resource was installed by the CICS system for a template being used through the CICS template manager, DFHWBTL.

DYNAMIC

The resource was installed as a result of the installation of a MQCONN with INITQNAME specified.

DYNAMIC

The resource was installed by an ATOMSERVICE resource.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

SYSTEM

The resource was installed by the CICS or CICSplex SM system.

TABLE

The resource was installed using a table definition.

TABLE

The resource was installed using a File Control table definition.

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.

Changes to CICSplex SM resource tables

Several CICSplex SM operations base tables support the resource signature.

Changed operations base tables

The resource signature attributes are added to the following operations base tables:

ATOMSERV
BUNDLE
CONNECT
DB2CONN
DB2ENTRY
DB2TRN
DOCTEMP
EJCOSE
EJDJAR
ENQMODEL
EXTRATDQ
INDTDQ
INTRATDQ
IPCONN
JRNLMODL
JVMSERV
LIBRARY
LOCFILE
LOCTRAN
MQCON
MQINI
PIPELINE
PROCTYP
PROFILE
PROGRAM

REMFILE
 REMTDQ
 REMTRAN
 RQMODEL
 TCPIPS
 TRANCLAS
 TSMODEL
 URIMAP
 WEBSERV
 XMLTRANS

Table 2. New fields in views

Field	Attribute name	Description
BAS resource definition version	BASDEFINEVER	The BAS version number of this definition.
Last modification agent	CHANGEAGENT	The change agent identifier that made the last modification.
Last modification agent release	CHANGEAGREL	The CICS release level of the agent that made the last modification to the resource definition.
Last modification userid	CHANGEUSRID	The userid that made the last modification to the resource definition.
Source of the resource definition	DEFINESOURCE	The source of the definition, depending on which agent made the last change.
Creation time	DEFINETIME	The local date and time when the resource definition record was created on DFHCSD or EYUDREP.
Installation agent	INSTALLAGENT	The change agent identifier that made the installation.
Installation time	INSTALLTIME	The local date and time when the definition was installed.
Installation userid	INSTALLUSRID	The userid that installed the resource definition.

For more information, see the *CICSplex System Manager Application Programming Guide*

Changes to CICSplex SM views and menus

Several CICSplex SM views now support the resource signature.

Changes to views

Detailed resource signature information can be viewed in the CICS operations views, listed in the table below. These new fields can also be displayed in the Web User Interface resource administration views.

View set	Navigation
EYUSTARTATOMSERV	CICS operations views → TCP/IP service operations views → Atomservices → EYUSTARTATOMSERV.DETAILED1
EYUSTARTBUNDLE	CICS operations views → Application operations views → Bundles → EYUSTARTBUNDLE.DETAILED1
EYUSTARTCONNECT	CICS operations views → Connection operations views → ISC/MRO connections → EYUSTARTCONNECT.DETAILED4
EYUSTARTDB2CONN	CICS operations views → DB2, DBCTL and WebSphere MQ operations views → Connections → EYUSTARTDB2CONN.DETAILED4
EYUSTARTDB2ENTRY	CICS operations views → DB2, DBCTL and WebSphere MQ operations views → Entries → EYUSTARTDB2ENTRY.DETAILED2
EYUSTARTDB2TRN	CICS operations views → DB2, DBCTL and WebSphere MQ operations views → Entry associated transactions → EYUSTARTDB2TRN.DETAILED1
EYUSTARTDOCTEMP	CICS operations views → Document template operations views → Document template → EYUSTARTDOCTEMP.DETAILED2
EYUSTARTEJCOSE	CICS operations views → Enterprise Java component operations views → CorbaServers → EYUSTARTEJCOSE.DETAILED5
EYUSTARTEJDJAR	CICS operations views → Enterprise Java component operations views → CICS-deployed JAR files → EYUSTARTEJDJAR.DETAILED1
EYUSTARTENQMODEL	CICS operations views → Enqueue model operations views → Enqueue model → EYUSTARTENQMODEL.DETAILED1
EYUSTARTEXTRATDQ	CICS operations views → Transient data queue (TDQ) operations views → Extrapartition → EYUSTARTEXTRATDQ.DETAILED1
EYUSTARTINDTDQ	CICS operations views → Transient data queue (TDQ) operations views → Indirect → EYUSTARTINDTDQ.DETAILED1
EYUSTARTINTRATDQ	CICS operations views → Transient data queue (TDQ) operations views → Intrapartition → EYUSTARTINTRATDQ.DETAILED1
EYUSTARTJRNLMDL	CICS operations views → Journal operations views → Models → EYUSTARTJRNLMDL.DETAILED1
EYUSTARTJVMSERV	CICS operations views → Enterprise Java component operations views → JVM Server → EYUSTARTJVMSERV.DETAILED1
EYUSTARTLIBRARY	CICS operations views → Program operations views → LIBRARYs, including DFHRPL → EYUSTARTLIBRARY.DETAILED1
EYUSTARTLOCFILE	CICS operations views → File operations views → Local files → EYUSTARTLOCFILE.DETAILED3

View set	Navigation
EYUSTARTLOCTRAN	CICS operations views → Transaction operations views → Local or dynamic → EYUSTARTLOCTRAN.DETAIL3
EYUSTARTMQCON	CICS operations views → DB2, DBCTL and WebSphere MQ operations views → WebSphere MQ connections → EYUSTARTMQCON.DETAIL4
EYUSTARTMQINI	CICS operations views → DB2, DBCTL and WebSphere MQ operations views → WebSphere MQ initiation queue → EYUSTARTMQINI.DETAILED
EYUSTARTPIPELINE	CICS operations views → TCP/IP service operations views → Pipelines → EYUSTARTPIPELINE.DETAIL1
EYUSTARTPROCTYP	CICS operations views → CICS Business Transaction Services (BTS) operations views → Process type → EYUSTARTPROCTYP.DETAIL1
EYUSTARTPROFILE	CICS operations views → Connection operations views → Profiles → EYUSTARTPROFILE.DETAIL1
EYUSTARTPROGRAM	CICS operations views → Program operations views → Programs → EYUSTARTPROGRAM.DETAIL1
EYUSTARTREMFIL	CICS operations views → File operations views → Remote files → EYUSTARTREMFIL.DETAIL1
EYUSTARTREMTDQ	CICS operations views → Transient data queue (TDQ) operations views → Remote → EYUSTARTREMTDQ.DETAIL1
EYUSTARTREMTRAN	CICS operations views → Transaction operations views → Remote → EYUSTARTREMTRAN.DETAIL1
EYUSTARTRQMODEL	CICS operations views → Transaction operations views → Request model → EYUSTARTRQMODEL.DETAIL1
EYUSTARTTCPIPS	CICS operations views → TCP/IP service operations views → TCP/IP service → EYUSTARTTCPIPS.DETAIL2
EYUSTARTTRANCLAS	CICS operations views → CICS region operations views → Transaction classes → EYUSTARTTRANCLAS.DETAIL1
EYUSTARTTSMODEL	CICS operations views → Temporary storage queue (TSQ) operations views → Models → EYUSTARTTSMODEL.DETAIL1
EYUSTARTURIMAP	CICS operations views → TCP/IP service operations views → URI map → EYUSTARTURIMAP.DETAIL3
EYUSTARTWEBSERV	CICS operations views → TCP/IP service operations views → Web services → EYUSTARTWEBSERV.DETAIL1
EYUSTARTXMLTRANS	CICS operations views → Application operations views → XML transformation → EYUSTARTXMLTRANS.DETAILED1

Changes to supplied transactions

CEDA now supports the definition signature. CECI supports the changes to the EXEC CICS INQUIRE command for a number of resource types.

Changes to CEDA

To support the definition signature, the PF2 key in the CEDA DISPLAY and EXPAND GROUP panels has a new function. To display the definition signature for an individual resource, or a group of resources, in the CEDA DISPLAY and

EXPAND GROUP panels press PF2. To return to the previous CEDA command panel, press PF2 again. You can switch between the two panels while retaining scroll position.

Changes to global user exits

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

Changed global user exit, XRSINDI

New user exit parameters have been added to the install and discard exit, XRSINDI.

UEPDEFTM

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, EVENTBINDING, FILE, IPCONN, JOURNALMODEL, JVMSERVER, LIBRARY, MQCONN, MQINI, PIPELINE, PROFILE, PROCESSTYPE, PROGRAM, REQUESTMODEL, TCPIPSERVICE, TDQUEUE, TRANCLASS, TRANSACTION, TSMODEL, URIMAP, WEBSERVICE, and XMLTRANSFORM. The parameter value is zero for all other resources.

UEPCHUSR

Address of a variable-length list, which corresponds to the list in UEPIDNAM, containing the 8-character user ID that ran the agent that last changed the individual resource.

UEPCHAGT

Address of a variable-length list, which corresponds to the list in UEPIDNAM, of a 2-byte identifier representing the agent that last changed the individual resource. The possible values are as follows:

UEPUNKAGT

The resource was changed by an unknown agent.

UEPCSDAPI

The resource was changed using the CSD API or CEDA.

UEPCSDBAT

The resource was changed using the CSD batch program, DFHCSDUP.

UEPDRPAPI

The resource was changed using the CICSplex SM BAS API.

UEPAUTOIN

The resource was changed using autoinstall.

UEPSYSTEM

The resource was changed by the running CICS region.

UEPDYNAMC

The resource was changed dynamically.

UEPTABLE

The resource was changed using a table.

UEPCHREL

Address of a variable-length list, which corresponds to the list in UEPIDNAM, containing the 4-character CICS release level that was running when the individual resource was last changed.

UEPCHTIM

Address of a variable-length list, which corresponds to the list in UEPIDNAM, containing the CSD record time stamp change for the individual resource as an 8-character STCK value.

UEPDEFSRC

Address of a variable-length list, which corresponds to the list in UEPIDNAM, containing the 8-character CSD group name or source corresponding to the individual resource.

UEPINUSR

Address of a variable-length list, which corresponds to the list in UEPIDNAM, containing the 8-character user ID that installed the individual resource.

UEPINTIM

Address of a variable-length list, which corresponds to the list in UEPIDNAM, containing the time that the domain was called for the installation of the individual resource as an 8-character STCK value.

UEPINAGT

Address of a variable-length list, which corresponds to the list in UEPIDNAM, of a 2-byte identifier representing the agent that installed the individual resource. The possible values are as follows:

UEPCSDAPI

The resource was installed using the CSD API or CEDA.

UEPCRESPI

The resource was installed using the EXEC CICS CREATE SPI commands.

UEPGRPLST

The resource was installed at startup using GRPLIST install.

UEPAUTOIN

The resource was autoinstalled.

UEPSYSTEM

The resource was installed by the running CICS system.

UEPDYNAMC

The resource was installed dynamically.

UEPBUNDLE

The resource was installed by a bundle deployment.

UEPTABLE

The resource was installed using a table.

Changes to CICS utility programs

DFHCSDUP now supports the improvements in management of resources. The definition signature fields are included in **DFHCSDUP EXTRACT** processing.

New DFHCSDUP LIST parameter, SIGSUMM

A new option is added to **DFHCSDUP LIST** to produce a summary of definition signatures for all of the specified resources.

Sigsumm

Shows the definition signatures for each of the resource definitions in the group specified.

New DFHCSDUP ADD parameters, BEFORE and AFTER

New options are added to **DFHCSDUP ADD** to control where a new group is placed.

After(*groupname2*)

Specify **AFTER** to place the new group name after the existing group name. The group name is added at the end of the list if **BEFORE** or **AFTER** is not specified.

Before(*groupname3*)

Specify **BEFORE** to place the new group name before the existing group name. The group name is added at the end of the list if **BEFORE** or **AFTER** is not specified.

Changes to statistics

A new DFHSTUP extract function sample program, DFH0STXD, provides an installed resource definition report.

New statistics extract sample program, DFH0STXD

The new statistics utility **EXTRACT** sample program, DFH0STXD, provides a report that shows the resource type, the resource ID, and the resource installation signature.

Changes to sample programs

DFH0STXD is a new statistics utility **EXTRACT** program, which produces a summary of the installed resources.

New sample EXTRACT program, DFH0STXD

The new statistics utility **EXTRACT** sample program, DFH0STXD, provides a report, which shows the resource type, the resource ID, and the resource install signature. Here are the JCL and DFHSTUP control statements required to invoke DFH0STXD.

```
//STXDRPT DD SYSOUT=x
EXTRACT USERPROGRAM=DFH0STXD
```

The following example shows these control statements:

```
//xxxxxxx JOB (job accounting)
//stepname EXEC PGM=DFHSTUP,REGION=0M
//STEPLIB DD DSN=....
// DD DSN=....
//DFHSTATS DD DSN=....
//SYSPRINT DD SYSOUT=x
//SYSOUT DD SYSOUT=x
//STXDRPT DD SYSOUT=x
//DFHPRINT DD SYSOUT=x
//SYSIN DD *
```

```
SELECT APPLID=(xxxxxxx)
EXTRACT USERPROGRAM=DFH0STXD
COLLECTION TYPE=ALL
SUMMARY
/*
//
```

Changed sample EXTRACT programs

These sample EXTRACT user programs for the DFHCSDUP utility program support the definition signature fields:

DFH\$CRFA, DFH\$CRFP, and DFH0CRFC
DFH\$FORA, DFH\$FORP, and DFH0FORC
DFH0CBDC
DFH\$DB2T and DFH\$SQLT

Changes to problem determination

New error messages support the positioning of groups using the DFHCSDUP ADD command.

New messages

Two new error messages support the positioning of groups using the DFHCSDUP ADD command:

- DFH5137E
- DFHCA5137E

Chapter 14. Security updates to monitor RACF Event Notifications (ENF)

CICS now monitors for RACF type 71 Event Notifications (ENF) that is issued when a CONNECT or REMOVE command has affected a RACF user profile, or a user ID is revoked using the REVOKE option of the ALTUSER command. With z/OS 1.11, RACF issues an ENF 71 event code and CICS is notified immediately, overriding any setting you have specified in the USRDELAY system initialization parameter.

If you are using RACF with a z/OS 1.11 system, review your USRDELAY settings.

Chapter 15. Support for identity propagation

If you are using z/OS, Version 1 Release 11, you can now send a user's security identity information (the distributed identity) from a client system across a network, preserving the distributed identity for use during CICS authorization and for subsequent auditing purposes.

Identity propagation depends on the z/OS Identity Propagation function that is provided in z/OS, Version 1 Release 11 and three enabling Program Temporary Fixes (PTF). For more information, see APARs PK95579, PM01622, PK83741, and PK98426. Identity propagation provides improved accountability and a new means of authorizing requests by associating the security information in the distributed environment with the security information used on the server.

CICS provides support for identity propagation in the following situations:

- Inbound requests to CICS from WebSphere Application Server using the CICS ECI resource adapters over an IPIC connection.
- Using a WS-Security Header element in a Web service request. Routed inbound Web service requests do not support identity propagation.
- Using IPIC and MRO connections between CICS systems. The distributed identity is used by CICS only if it is passed to the MRO or IPIC connection from a participating task.
- Function shipping of **START** commands and local **START** commands do not inherit the distributed identity.

Terminology

Terms used to explain support for identity propagation.

distinguished name

The name that uniquely identifies an entry in a directory. A distinguished name is made up of *attribute:value* pairs, separated by commas. The format of a distinguished name is defined by RFC4514 at <http://www.ietf.org/rfc/rfc4514.txt>.

distributed identity

User identity information that originates from a remote system. The distributed identity is created in one system and is passed to one or more other systems over a network.

ICRX (Extended Identity Context Reference)

A control block in RACF that contains information about the distinguished name and realm of a user, which are used for identity propagation.

identity propagation

The concept of preserving a user's security identity information (the distributed identity) regardless of where the identity information was created, for use during authorization and for auditing purposes. The distributed identity is part of the identity context that is carried with a request from the external client application to the server, and it is incorporated in the access control of the server as part of the authorization process; for example, using RACF. CICS receives and processes the distributed identity, which is sent by another application; for example, CICS Transaction Gateway.

realm A named collection of users and groups that can be used in a specific security context.

Changes to external interfaces

Some external interfaces now support identity propagation, which depends on the z/OS Identity Propagation function that is provided in z/OS, Version 1 Release 11 and three enabling Program Temporary Fixes (PTF). For more information, see APARs PK95579, PM01622, PK83741, and PK98426.

Changes to system initialization parameters

A new system initialization parameter, MNIDN, specifies whether the monitoring identity class is to be made active during CICS initialization.

Changes to resource definitions

With the new IDPROP attribute in the IPCONN resource, you can specify whether the distributed identity is transmitted outside the sysplex.

Changes to the system programming interface

The **INQUIRE ASSOCIATION**, **INQUIRE ASSOCIATION LIST**, **INQUIRE IPCONN**, **INQUIRE MONITOR**, and **SET MONITOR** commands have new options.

Changed command, **INQUIRE ASSOCIATION**

New options, DNAME and REALM, display the distinguished name and realm name returned for the specified task.

Changed command, **INQUIRE ASSOCIATION LIST**

New options, DNAME and REALM, are search filters for the distinguished name and realm name. When these options are specified, the task IDs for the specified list of tasks are returned. New options, DNAMELEN and REALMLEN, specify the length of the distinguished name search filter and realm name fields.

Changed command, **INQUIRE IPCONN**

A new option, IDPROP, displays whether the sender includes the distributed identity in attach requests over the IPIC connection.

Changed commands, **INQUIRE MONITOR** and **SET MONITOR**

A new option, IDNTYCLASS, specifies whether identity class monitoring is enabled.

Changes to CEMT (master terminal transaction) commands

The **CEMT INQUIRE IPCONN**, **CEMT INQUIRE MONITOR** and **CEMT SET MONITOR** commands have new options.

Changed command, **CEMT INQUIRE IPCONN**

A new option, IDPROP, displays whether the sender includes the distributed identity in attach requests over the IPIC connection.

Changed commands, CEMT INQUIRE MONITOR and CEMT SET MONITOR

A new option, IDNTYCLASS, specifies whether identity class monitoring is enabled.

Changes to CICSplex SM resource tables

Several CPSM resource definition base tables now include identity propagation information.

Changed tables

TASKASSC

The TASKASSC resource table has two new attributes, DNAME and REALM.

CICSRGN and MONITOR

The CICSRGN and MONITOR resource tables have a new attribute, IDNTYCLASS.

IPCONN and IPCONDEF

The IPCONN and IPCONDEF resource tables have a new attribute, IDPROP.

Changes to CICSplex SM views and menus

The CICS operations views now display identity propagation information.

Changed views

EYSTARTCICSRGN operations viewset (CICS operations views > CICS region operations views > CICS regions > click Set attributes)

A new attribute, IDNTYCLASS, is displayed as a monitoring value.

EYSTARTIPCONDEF administration viewset (Administration views > CICS resource definitions)

A new attribute, IDPROP, is displayed.

EYSTARTIPCONN operations viewset (CICS operations views > TCP/IP service operations views)

A new attribute, IDPROP, is displayed.

EYSTARTMONITOR operations viewset (CICS operations views > CICS region operations views > CICS regions > click a region to go to EYSTARTCICSRGN.DETAILED view > click Monitor status > Monitor view EYSTARTMONITOR.DETAILED)

A new attribute, IDNTYCLASS, is displayed.

EYSTARTTASKASSC operations viewset (CICS operations views > Task operations views > Task association information)

Two new attributes, DNAME and REALM, are displayed.

Changes to supplied transactions

Transaction, CEMN, displays a new identity class field, which allows you to specify whether identity class monitoring is active or inactive.

Note: This topic does not describe the following changes:

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

Changes to global user exits

The global user exit, XMNOUT, is now called when an identity class monitoring record is buffered. A record type value of 6 identifies an identity class record.

Changes to monitoring data

A new monitoring Identity Class data record is written by CICS as an SMF 110 subtype 1 record. The new record is created during transaction detach processing for each transaction that has identity propagation data (a distinguished name and realm). The Performance Class record field now includes an indicator in the current Transaction Flags field that indicates that an identity record is available for this task.

Changes to statistics

The statistics sample program, DFH0STAT, and the statistics utility program, DFHSTUP, display new information for the new identity record fields. The System Status report includes the current status of the monitoring field, IDNTYCLASS, and the monitoring statistics record counts. The Monitoring Domain global statistics record now includes counts of identity records and the number of identity records suppressed.

Changes to sample programs

The monitoring sample program, DFH\$MOLS, now provides a report of the new identity class records. The DFH\$MOLS totals report page also includes information about the number of identity records processed.

Changes to problem determination

New information helps you to diagnose problems relating to identity propagation.

New messages

A new message, DFHIS1041, is issued by the Intersystem domain. The message explains that an attempt to send a request across an IPIC connection has failed because the partner region has asked for an ICRX Identity Propagation token, and the ICRX token assigned to the current task is larger than the maximum token size that the IPIC message protocol supports.

A new message, DFHPI0727, is issued by the Pipeline Manager domain. The message explains that a PIPELINE resource cannot be installed because it requires support for ICRX-based identity tokens and these are not supported by the platform.

Message DFHIS1027 is updated to include ICRX information.

Messages DFHPI0119 and DFHPI0515 are updated with the required version of the XML Toolkit for z/OS.

New abend codes

New abend codes, AXFN and AXFV, are issued if the user domain module, DFHUSAD, returns a condition not expected by DFHXFX.

New trace points

New trace points AP 4905 and AP 4906 are issued by the Application Manager domain.

New trace points PI 1717 to PI 1719 and PI1720 to PI 1722 are issued by the Pipeline Manager domain.

New trace points US 0319 to 031B are issued by the User domain.

Chapter 16. HTTP basic authentication support for Web services

You can now provide user ID and password information across an HTTP protocol, when a Web service requester sends an outbound request to a Web service provider that requires authentication. This type of authentication is called HTTP basic authentication. You need a URIMAP resource specifying AUTHENTICATE(BASIC) and an enabled XWBAUTH global user exit to perform HTTP basic authentication.

Changes to external interfaces

URIMAP resources now allow you to send basic authentication information from a Web Services requester to a Web Services provider over an HTTP protocol, using the XWBAUTH global user exit.

Changes to resource definitions

The URIMAP resource has a new attribute, AUTHENTICATE, that allows HTTP basic authentication information to be passed in an outbound request from a Web service requester to a Web service provider, by specifying AUTHENTICATE(BASIC). The new attribute applies when USAGE(CLIENT) is specified.

Changes to the system programming interface

The **INQUIRE URIMAP** command has a new option, AUTHENTICATE, which indicates whether to provide authentication information in an outbound HTTP request to a Web service provider.

Changes to CEMT (master terminal transaction) commands

The **INQUIRE URIMAP** command has a new option, AUTHENTICATE, which indicates whether to provide authentication information in an outbound HTTP request to a Web service provider.

Changes to CICSplex SM resource tables

The URIMAP and URIMPDEF base tables have a new attribute, AUTHENTICATE.

Changes to CICSplex SM views and menus

The URIMAP views display a new attribute, AUTHENTICATE.

Changes to statistics

The statistics sample program, DFH0STAT, and the statistics utility program, DFHSTUP, now display information about whether HTTP basic authentication information is sent on outbound requests to a Web service provider.

Changed statistics type, URIMAP

The URIMAP report displays information for USAGE(CLIENT) about whether authentication information is sent over an HTTP protocol to a Web service provider that requires basic authentication.

Changes to problem determination

New information helps you to diagnose problems that relate to HTTP basic authentication for Web Services.

Changed messages

Message DFHPI0400 has changed to include the XWBAUTH global user exit.

Chapter 17. Security improvement for DB2 users

In CICS Transaction Server for z/OS, Version 4 Release 1, the CICS interface with DB2 provides more flexibility when you use DB2 and RACF. CICS now passes the address of the access control environment element (ACEE) when AUTHTYPE(SIGN) is specified for a DB2CONN or a DB2ENTRY resource, and the SIGNID attribute specifies the CICS region user ID.

With this change, DB2 can use RACF security when you use the CICS region user ID to control CICS's access to DB2.

In previous releases, the ACEE is passed to DB2 only when AUTHTYPE(USERID) or AUTHTYPE(GROUP) is specified.

Changes to external interfaces

Some external interfaces have changed to support the security improvements for DB2 users.

Changes to resource definitions

The behavior of the AUTHTYPE attribute on DB2CONN and DB2ENTRY resources has changed when the attribute has a value of SIGN.

When you specify AUTHTYPE(SIGN) on a DB2CONN or DB2ENTRY resource, and the SIGNID attribute of the DB2CONN specifies the CICS region user ID, CICS now passes the RACF access control environment element (ACEE) for the CICS region user ID to DB2, in addition to the value of the SIGNID attribute. The ACEE is required if you are using DB2 with RACF external security.

Chapter 18. Interoperability between the Web services assistant and WSRR

The CICS Web services assistant now includes interoperability support for the IBM WebSphere Service Registry and Repository (WSRR). WSRR helps you to manage and govern your services and processes. Use WSRR to find the Web services that you are requesting more quickly and enforce version control of the Web services you are providing.

Both DFHLS2WS and DFHWS2LS include parameters to interoperate with WSRR. DFHLS2WS also includes an optional parameter so that you can add your own customized metadata to the WSDL document in WSRR.

If you want the Web services assistant to communicate securely with WSRR, you can use secure socket level (SSL) encryption. Both DFHLS2WS and DFHWS2LS include parameters for using SSL encryption.

Enhancements to the Web services assistant

The Web services assistant batch jobs DFHWS2LS and DFHLS2WS have new parameters to support interoperability with the IBM WebSphere Service Registry and Repository (WSRR) server. You can optionally secure the network connection to WSRR using secure socket layer (SSL) encryption.

Changes to DFHLS2WS

When you create a new Web service from a language structure, you can now decide whether you want to publish it on a WSRR server.

The following new parameters are added to DFHLS2WS:

WSRR-CUSTOM-PropertyName=value

Use this optional parameter to add customized metadata to the WSDL document in the WSRR. The **WSRR-CUSTOM-PropertyName=value** pairs are added into the WSDL document and appear in WSRR without the **WSRR-CUSTOM** prefix.

You can specify a maximum of 255 custom *PropertyName=value* pairs. Avoid duplicate and blank *PropertyName=value* pairs.

Use this parameter only when the **WSRR-SERVER** parameter is specified.

WSRR-DESCRIPTION=value

Use this optional parameter to specify the metadata that describes the WSDL document being published.

Use this parameter only when the **WSRR-SERVER** parameter is specified.

WSRR-ENCODING=value

Use this optional parameter to specify the character set encoding of the WSDL document. If the **WSRR-ENCODING** parameter is not specified, WSRR uses the value specified in the WSDL document.

Use this parameter only when the **WSRR-SERVER** parameter is specified.

WSRR-LOCATION=value

Use this optional parameter to specify the URI that identifies the location of

the WSDL document. If this parameter is not specified, the URI defaults to the filename specified in the **WSDL** parameter. For example, if the value of the **WSDL** parameter is `wrr/example.wsdl`, the value of the **WSRR-LOCATION** parameter defaults to `example.wsdl`.

Use this parameter only when the **WSRR-SERVER** parameter is specified.

WSRR-PASSWORD=*value*

Use this optional parameter if you must enter a password to access WSRR.

If the **WSRR-USERNAME** parameter is specified, you must also specify this parameter.

Use this parameter only when the **WSRR-SERVER** parameter is specified.

WSRR-SERVER={*domain name:port number*}|{*IP address:port number*}

Use this parameter to specify the location of the IBM WebSphere Service Registry and Repository (WSRR) server. If this parameter is specified, WSRR parameter validation is used.

WSRR-USERNAME=*value*

Use this optional parameter if you are required to specify a user name to access WSRR. This user name is used by WSRR to set the owner property.

Use this parameter only when the **WSRR-SERVER** parameter is specified.

WSRR-VERSION=1|*value*

Use this parameter to set the version property of the WSDL document in WSRR.

Use this parameter only when the **WSRR-SERVER** parameter is specified.

Changes to DFHWS2LS

When you create a language structure from a WSDL document, you can now decide whether you want to use a WSDL document that is published on a WSRR server.

The following new parameters are added to DFHWS2LS:

WSRR-NAME=*value*

Specifies the name of the WSDL document to retrieve from WSRR. Use this parameter only when the **WSRR-SERVER** parameter is specified.

WSRR-NAMESPACE=*value*

Specifies the namespace of the WSDL document to retrieve from WSRR. You can optionally use this parameter when the **WSRR-SERVER** parameter is specified to fully qualify the WSDL document name specified in the **WSRR-NAME** parameter.

WSRR-PASSWORD=*value*

Use this optional parameter if you must enter a password to access WSRR.

If the **WSRR-USERNAME** parameter is specified, you must also specify this parameter.

Use this parameter only when the **WSRR-SERVER** parameter is specified.

WSRR-SERVER={*domain name:port number*}|{*IP address:port number*}

Use this parameter to specify the location of the IBM WebSphere Service Registry and Repository (WSRR) server. If this parameter is specified, WSRR parameter validation is used.

WSRR-USERNAME=*value*

Use this optional parameter if you are required to specify a user name to access WSRR. This user name is used by WSRR to set the owner property.

Use this parameter only when the **WSRR-SERVER** parameter is specified.

WSRR-VERSION=*value*

Specifies the version of the WSDL document to retrieve from WSRR. You can optionally use this parameter when the **WSRR-SERVER** parameter is specified.

New parameters to support SSL encryption

The following new parameters are added to DFHWS2LS and DFHLS2WS:

SSL-KEYSTORE=*value*

This optional parameter specifies the fully qualified location of the key store file.

Use this parameter if you want the Web services assistant to use secure sockets layer (SSL) encryption to communicate across a network to an IBM WebSphere Service Registry and Repository (WSRR).

SSL-KEYPWD=*value*

This optional parameter specifies the password for the key store.

Use this parameter if you want the Web services assistant to use secure sockets layer (SSL) encryption to communicate across a network to an IBM WebSphere Service Registry and Repository (WSRR).

SSL-TRUSTSTORE=*value*

This optional parameter specifies the fully qualified location of the trust store file.

Use this parameter if you want the Web services assistant to use secure sockets layer (SSL) encryption to communicate across a network to an IBM WebSphere Service Registry and Repository (WSRR).

SSL-TRUSTPWD=*value*

This optional parameter specifies the password for the trust store.

Use this parameter if you want the Web services assistant to use secure sockets layer (SSL) encryption to communicate across a network to an IBM WebSphere Service Registry and Repository (WSRR).

Changes to external interfaces

Some external interfaces now support interoperability between the Web services assistant and WSRR.

Changes to problem determination

New information helps you to diagnose problems that relate to interoperability between the Web services assistant and WSRR.

New messages

New messages, DFHPI9800 to DFHPI9823, provide information about the failure of the Web services assistant to interoperate successfully with WSRR.

Part 4. Control

CICS Transaction Server for z/OS, Version 4 Release 1 delivers a set of capabilities represented in the following themes:

- Compete
- Comply
- Control
- Architectural enhancements

The capabilities represented by the *Control* theme help IT staff to perform their tasks more effectively, while assuming a mixture of skill levels.

Chapter 19. The IBM CICS Explorer

The CICS Explorer is a system management tool that is designed to provide a simple, easy to use way of managing one or more CICS systems. Based on an Eclipse platform, it provides functions to view and manage CICS Transaction Server regions and a platform for the integration of CICS Tools.

Eclipse is a platform for building and deploying client applications in which the majority of data manipulation is done by the client application rather than the server. This platform is known as a Rich Client Platform (RCP). The platform provides the ability to deploy native graphical user interface (GUI) applications to a variety of desktop operating systems. The CICS Explorer is an Eclipse RCP based GUI application that runs on the Windows and Linux operating systems.

Today, you manage CICS regions with a variety of stand-alone graphical and non-graphical user interfaces, and with no common way of performing tasks. The CICS Explorer provides a base set of CICSplex SM functions to manage CICS regions, and the CICS event binding editor used by business analysts to define event specifications and by application analysts to define event specifications and dispatcher information. In addition, the CICS Explorer provides a platform on which CICS tools that are integrated in the Eclipse RCP, can perform tasks and present the information in a common way.

CICS Explorer connects to either a CPSM WUI server or a single CICS region. When connected to a CPSM WUI server, CICS Explorer operations views provide a single-system image of all the CICS resources within a CICSplex. When the CICS Explorer is connected to a CICS Transaction Server for z/OS version 3 system, the resources can only be browsed, however using the operations views and CICS Transaction Server for z/OS version 4, you can perform the following actions:

- Enable and disable resources
- Open and close resources
- Acquire and release resources
- Place resources in and out of service
- Purge tasks associated with a resource
- Discard resource definitions from a CICS system where they are installed

The CICS Explorer administration views display details of the CICS and CICSplex SM resource definitions, and with CICS Transaction Server for z/OS version 4 you can update the resource definitions using the CICS Explorer resource editor. Using the administration views and the resource editor, you can perform the following actions:

- Display resource definitions
- Create and update resource definitions
- Install or remove resource definitions

The CICS Explorer resource editor displays the resource definition attributes in two or more pages in the editor. One page, the Attributes page, displays the attributes in tabular form grouped together in the same way as they appear in CEDA or the CPSM WUI to facilitate the move to CICS Explorer for experienced CICS users. The other page, or pages, display the attributes logically grouped together, but each attribute has more detail provided to simplify the update process for

less-experienced CICS users. Every attribute benefits from field-level verification, where the entry is validated in real time. Errors are identified by the **Error** icon, which identifies the field in error and the page on which the field appears.

The CICS Explorer contains context and more detailed help for the resource views and help on using the CICS Explorer and the basic Eclipse functions. Detailed help for CICS and CICSplex SM is found in the CICS Transaction Server for z/OS Information Center. Detailed help for Eclipse is found at the Eclipse homepage.

Changes to installation procedures

The installation of the CICS Explorer is performed separately to CICS. The Installation Guide is updated with procedures to plan and install the CICS Explorer.

CICS Explorer planning

The CICS Explorer is a system management tool that is delivered as part of CICS Transaction Server. It is designed to provide an easy-to-use way of managing one or more CICS systems. Based on an Eclipse platform, it provides a view of some of the CICSplex SM functions in a CICS Transaction Server and a platform for the integration of future CICS tools. You must decide how you will run the CICS Explorer in your organization before installing it.

CICS Explorer installation

You can install the CICS Explorer in a number of places depending on the requirements of your organization. After installation, you must configure the CICS Explorer connections to your CICS or CICSplex SM systems and you can also define where the CICS Explorer saves its workspace,

Chapter 20. Extensions to CICS-to-CICS intercommunications over TCP/IP

You can now use a TCP/IP network to route transactions and send START and CANCEL commands between CICS TS regions.

IP interconnectivity (IPIC) was introduced in CICS TS for z/OS, Version 3.2, providing an alternative to intersystem communications over SNA (ISC over SNA).

IPIC connections offer similar capabilities and qualities of service to those provided by ISC over SNA, including transactional syncpointing, security controls, encryption, and support for the exchange of either COMMAREAs or channels and containers. Coupled with the latest System z TCP/IP networking enhancements, IPIC enables CICS intercommunications to fully exploit modern high capacity and adaptable TCP/IP communication frameworks.

IPIC supports these intercommunication functions and releases:

- Distributed program link (DPL) calls between CICS TS 3.2, or later regions
- Distributed program link (DPL) calls between CICS TS and TXSeries Version 7.1, or later
- Asynchronous processing of **EXEC CICS START**, **START CHANNEL**, and **CANCEL** commands, between CICS TS 4.1, or later regions
- Traditional transaction routing of 3270 terminals, where the terminal-owning region (TOR) is uniquely identified by an APPLID between CICS TS 4.1, or later regions
- ECI requests from CICS Transaction Gateway Version 7.1 or later

Up to two send and two receive sockets are now available for each IPIC connection, replacing the single send and single receive socket. The increase in the number of sockets can improve your transaction throughput.

Changes to external interfaces

Some external interfaces have changed to support extended IPIC function.

Changes to resource definitions

You can use the REMOTESYSTEM attribute in the TERMINAL and TRANSACTION resources to identify an IPIC connection, by specifying the first 4 characters of the IPCONN name.

The REMOTESYSTEM attribute of the TERMINAL and TRANSACTION resources now allows you to specify information about IP connections.

Changes to the system programming interface

New options have been added to the **INQUIRE ASSOCIATION** and **INQUIRE IPCONN** commands. The REMOTESYSTEM option of **INQUIRE TERMINAL** and **INQUIRE TRANSACTION** returns information about an IP connection.

Changed command, INQUIRE ASSOCIATION

INQUIRE ASSOCIATION is updated with a new option, **CLIENTLOC**, which provides information about the **SO_CLUSTERCONNTYPE** options returned by z/OS Communications Server.

Changed command, INQUIRE IPCONN

INQUIRE IPCONN is updated with new options, **CLIENTLOC**, which provides information about the **SO_CLUSTERCONNTYPE** options returned by z/OS Communications Server, and **PARTNER**, which provides information about a partner system.

Changed commands, INQUIRE TERMINAL and INQUIRE TRANSACTION

The **REMOTESYSTEM** option of the **INQUIRE TERMINAL** and **INQUIRE TRANSACTION** commands now provides information about IP connections.

Changes to CEMT (master terminal transaction) commands

The **INQUIRE IPCONN** command has a new option, **PARTNER**, and the **REMOTESYSTEM** option of **INQUIRE TERMINAL** and **INQUIRE TRANSACTION** returns information about an IP connection.

Changed command, INQUIRE IPCONN

The **INQUIRE IPCONN** command has a new option, **PARTNER**.

Changed commands, INQUIRE TERMINAL and INQUIRE TRANSACTION

The **REMOTESYSTEM** option of the **INQUIRE TERMINAL** and **INQUIRE TRANSACTION** commands now provides information about IP connections.

Changes to supplied transactions

The routing transaction, **CRTE**, now supports transaction routing over an **IPIC** connection.

Other transactions

- Changes to CEMT commands are described in “Changes to CEMT (master terminal transaction) commands.”
- Changes to resources managed by the **CEDA** transaction are described in “Changes to resource definitions” on page 137.

Changes to CICSplex SM resource tables

The **IPCONN** and **TASKASSOC** operations base tables include new information.

Changed table IPCONN

The **IPCONN** operation base table has new attributes, **CLIENTLOC** and **PARTNER**. **IPCONN** also includes information about the number and size of transaction routing requests and interval control requests for function shipping, and the number of **START** requests sent from a remote terminal.

Changed table TASKASSC

The TASKASSC operations base table has a new CLIENTLOC attribute.

Changes to CICSplex SM views and menus

Views EYUSTARTIPCONN and EYUSTARTTASKASSC have new fields.

Changed view EYUSTARTIPCONN

EYUSTARTIPCONN has new fields, CLIENTLOC and PARTNER. EYUSTARTIPCONN also displays information about the number and size of transaction routing requests and interval control requests for function shipping, and the number of **START** requests sent from a remote terminal.

Changed view EYUSTARTTASKASSC

EYUSTARTTASKASSC has a new field, CLIENTLOC.

Changes to global user exits

A new global user exit, XISQLCL, has been added. This allows you to control how **START** requests are processed that are sent over an IPIC connection.

XISQLCL is used for EXEC CICS START NOCHECK commands that are scheduled for an IPIC connection, and allows you to decide whether to add the request to a local queue.

Changes to monitoring data

A new field, MNR_ID_ISIPICNM, in the transaction resource monitoring record, DFHMNRDS, provides information about IP interconnectivity.

Changes to statistics

The statistics sample program, DFH0STAT, and the statistics utility program, DFHSTUP, now display information about transaction routing and function-shipped interval control requests when using IPIC connections.

Changed statistics type, IPCONN

The IPCONN report displays information about transaction routing requests relating to the number of requests, the number of bytes sent, and the number of bytes received over an IPIC connection by transaction routing requests.

The IPCONN report displays information about interval control requests that have been sent using function shipping. The information relates to the number of requests, the number of bytes sent, and the number of bytes received over an IPIC connection by function-shipped interval control requests.

The IPCONN report displays information about the number of **START** requests sent from a remote terminal.

Changes to sample programs

A new sample global user exit program, DFHLEXISL, has been added.

You can use the XISQLCL sample global user exit program, DFH£XISL, to control the queueing of START NOCHECK requests that are scheduled for an IPIC connection.

Changes to problem determination

There is new information to help you diagnose problems relating to IPIC.

New messages

A new error message, DFHIS2000, and new informational messages in the range DFHIS1032 to DFHIS1040 are issued by the Inter-system domain.

A new error message, DFHRT4406, is issued by the ISC transaction routing component.

A new error message, DFHDS0007, is issued by the Dispatcher domain.

New abend codes

New abend codes, ABMT, in response to a BMS request that has been issued for a terminal type which does not support the request and ACRQ, in response to invalid transaction routing.

New trace points

New trace points AP DB10 and AP DB11, AP 068A and AP 068B, and trace points in the ranges AP 0930 to AP 093E, AP 0940 to 0945, and AP 0950 to AP 0955 are issued by the Application Manager domain.

New trace point IS 110F, new trace points IS 0688 and IS 0689, and trace points in the ranges IS 0531 to IS 0536 and IS 0631 to IS 0636 are issued by the Inter-system domain.

Changed trace points

The trace points IS 0906, IS 120B, and IS 1407 are changed in the Inter-system domain.

Chapter 21. Improvements to system links for IPIC connections

You can now define and install IPIC connections between CICS regions by using SYSLINK resources in CICSplex SM.

A SYSLINK resource defines the connection between two CICS regions. When you install a SYSLINK resource, CICSplex SM installs the defined resources for you in the correct CICS regions to create the connection.

You can also manage all of your SYSLINK resources together or filter them by connection type.

Changes to external interfaces

Some external interfaces now support the improvements to system links in CICSplex SM.

Changes to CICSplex SM resource tables

The CICSplex SM resource tables now support the creation and management of IPIC connections using SYSLINK resource objects.

Changed tables, SYSLINK, CSYSDEF, and MAS

The SYSLINK table has a new field called LINKTYPE, which describes the type of connection resource, either SNA or IPIC. The field level help for this table has also changed to reflect that this new field affects the values of other fields:

Field name	Description	Field level help
CONNDEF	Primary connection definition name	The name of the primary connection definition that describes the link. The resource type depends on the value of the LINKTYPE field.
CONNDEFVER	Primary connection definition version	The version of the primary connection definition, in the range 1 to 15.
SESSDEF	Secondary connection definition name	The name of the secondary connection definition that describes the link. The resource type depends on the value of the LINKTYPE field.
SESSDEFVER	Secondary connection definition version	The version of the secondary connection definition, in the range 1 to 15.

Field name	Description	Field level help
LINKTYPE	Connection definition type	<p>The protocol of the connection between the CICS regions. The supported values are SNA, which is the default, and IPIC:</p> <p>SNA The primary connection definition is a MRO/ISC connection definition and the secondary connection definition is a session definition</p> <p>IPIC The primary connection definition is an IPIC connection definition and the secondary connection definition is a TCP/IP service definition.</p>

The CSYSDEF and MAS tables have three new fields that are used when defining IPIC connections:

Field name	Description	Field level help
NETWORKID	Network ID	The Network ID of this CICS region. NETWORKID is used with the APPLID option to ensure unique naming for regions that use IPIC connections.
HOST	Host name	The host name of the CICS region or its dotted decimal IP address; for example, <code>www.example.com</code> or <code>208.77.188.166</code> . The host name is used for IPIC connections that are defined using SYSLINK. The host name can be up to 116 characters in length.
PORT	Port number	A number, in the range 1 through 65535, that specifies the port number to be used for outbound requests on this IPIC connection. That is, the number of the port on which the remote CICS region will be listening.

Changes to CICSplex SM views and menus

CICSplex SM views and menus now support the creation and management of IPIC connections using SYSLINK resource objects, including filtering SYSLINK definitions by connection type.

New menu, EYUSTARTADMSYSLINK

This new menu is the starting point for viewing existing SYSLINK resources and related CICSplex SM resources. To access this menu from the main menu, click **Administration views > CICS system links and related resources**. The menu is organized into three sections:

1. The first section contains a link to System Link Definitions. You can select to view all the SYSLINK resources, or look at connections of a specific type; the types that are available are SNA (ISC/MRO) or IPIC.
2. The second section contains links to related BAS resource definitions, namely CONNDEF, SESSDEF, IPCONDEF, TCPDEF, and CSYSDEF views.
3. The third section contains a set of links to the installed resources in the current context and scope, to provide an easy way to verify the installation of a

SYSLINK resource. This set of links let you view CONNECT, IPCONN, TCPIPS, and CICSRRGN operational objects.

Changed menus, EYUSTARTADMBAS and EYUSTARTADMBAS2

The EYUSTARTADMBAS and EYUSTARTADMBAS2 menus, which previously linked directly to the System Link Definition (SYSLINK) view, now link to the new menu EYUSTARTADMSYSLINK as described above.

Changed views

The view set for SYSLINK resources now use the new LINKTYPE field. To access this view set from the main menu, click **Administration views > CICS system links and related resources > All system links**. The following views have changed:

Tabular view

The links in the Tabular views take you to a view of the BAS resource definition. These links are now conditional. Depending on the value that is specified in the Connection resource type column (LINKTYPE field), the links show CONNDEF or IPCONDEF and SESSDEF or TCPDEF definitions.

Detail view

This view displays the connection type in the details of the SYSLINK definition.

Create and Update views

You can select either SNA or IPIC in the **connection definition type** field. The values that you can select in the subsequent fields change depending on the connection type that you require.

Install view

This view displays the connection type in the SYSLINK definition that you want to install.

Remove view

This view displays the connection type in the SYSLINK definition that you want to remove.

The CSYSDEF and MAS views have three additional fields that are used to define IPIC connection definitions into the CICS region. To access these views from the main menu, click **Administration views > RTA system availability monitoring > CICS system definitions** and **History views > MASs known to CICSplex**.

CSYSDEF Create and Update views

You can specify the hostname, network ID, and port number of the CICS region.

CSYSDEF Detail view

This view displays the hostname, network ID, and port number of the CICS region if specified.

MAS Detail view

This view displays the hostname, network ID, and port number of the CICS region if specified.

Chapter 22. Support for IPv6

With CICS TS 4.1, you can now operate in an Internet Protocol version 6 (IPv6) network without the need for address translation. CICS still operates in an IPv4 network as before.

IPv6 is the protocol designed by the IETF to replace the current version Internet Protocol, IP Version 4 (IPv4). The IPv6 format has a much larger address space than IPv4 (128 bits compared to 32 bits), which provides greater flexibility in allocating addresses and for routing traffic.

CICS operating in a dual-mode environment uses both IPv4 and IPv6 networks and always attempts to communicate using IPv6 before using the IPv4 network. A single-mode environment operates in an IPv4 network only. You need a minimum level of CICS TS 4.1 to communicate using IPv6. The CICS TS 4.1 region must be running in a dual-mode (IPv4 and IPv6) environment and the client or server that CICS is communicating with must also be running in a dual-mode environment.

Changes to external interfaces

Some external interfaces now support IPv6 function.

Changes to the CICS application programming interface

New and updated options on WEB and EXTRACT TCPIP application programming commands provide support for IPv6.

Changed command, EXTRACT TCPIP

New client options, CLNTADDR6NU and CLNTIPFAMILY, and server options, SRVRADDR6NU and SRVRIPFAMILY, return IPv6 address information. Existing options CADDRLENGTH, CLIENTADDR, SADDRLENGTH, and SERVERADDR are updated to return IPv6 information.

Changed commands, EXTRACT WEB, WEB EXTRACT and WEB PARSE URL

The HOST option is extended to support IPv6 addresses. A new option, HOSTTYPE, returns the format of the HOST option.

Changed command, WEB OPEN

The HOST option is extended to support IPv6 addresses.

Changes to the JCICS application programming interface

The JCICS API has been enhanced to return colon-hexadecimal addresses from a Java application.

The JCICS API provides equivalents for the following new and changed API commands:

EXTRACT TCPIP

WEB EXTRACT

The support is provided by the following existing JCICS classes:

com.ibm.cics.server.TcpipRequest

New methods, getClientHostAddress6() and getServerHostAddress6() return an IPv6 address. New methods, getClientIpFamily() and getServerIpFamily() return whether the IP address is in IPv4 or IPv6 format, or whether an incorrect host is returned.

com.ibm.cics.server.HttpSession and com.ibm.cics.server.HttpRequest

New method, getHostType() returns whether the host IP address is in IPv4 or IPv6 format, or whether an incorrect host is returned.

Changes to resource definitions

Resource definitions that use the HOST attribute CORBASERVER, IPCONN, and URIMAP, now allow IPv6 addresses. For TCPIP SERVICE, a new attribute, HOST, supports IPv6 addresses. For URIMAP, you can use a new option, PORT, to specify port number information.

Changed HOST attribute on CORBASERVER, IPCONN, and URIMAP resource definitions

The HOST attribute on CORBASERVER, IPCONN, and URIMAP resources now supports IPv6 addresses.

Changed resource definition, TCPIP SERVICE

A new attribute, HOST, specifies the host name or IP address of the local system, replacing the IPADDRESS option. IPADDRESS is supported for existing IPv4 function only. Use the HOST option for new TCPIP SERVICE definitions.

Changed resource definition, URIMAP

A new attribute, PORT, specifies the port number used for the connection. This option replaces the use of the HOST option for specifying a port number. Use the HOST option to specify the port number for existing programs only.

Changes to the system programming interface

Commands that use the HOST option now allow IPv6 addresses. For TCPIP SERVICE, a new option, HOST, supports IPv6 addresses. New options, HOSTTYPE, IPRESOLVED, and IPFAMILY return information about the host address. For URIMAP, you can use a new option, PORT, to specify port number information.

Changed command, CREATE TCPIP SERVICE

A new option, HOST, replaces IPADDRESS for new programs.

Changed command, CREATE URIMAP

A new option, PORT, specifies the port number used for the connection. This option replaces the use of the HOST option for specifying a port number. Use the HOST option to specify the port number for existing programs only.

Changed command, INQUIRE ASSOCIATION

A new option, SRVRIPFAMILY, replaces the IPFAMILY option for new programs. A new option, CLNTIPFAMILY, is also available. The ODIPFAMILY option now includes IPv6 values. The CLIENTIPADDR, ODCLNTIPADDR, and SERVERIPADDR options now return IPv6 addresses.

Changed commands, INQUIRE CORBASERVER, IPCONN, and URIMAP

The HOST option is updated to allow IPv6 addresses and a new option, HOSTTYPE, returns the format of the contents of the HOST option. A new option, IPRESOLVED, returns the IP address of the host and a new option, IPFAMILY, returns the format of the IPRESOLVED option.

For more information, see INQUIRE CORBASERVER, INQUIRE IPCONN, and INQUIRE URIMAP.

Changed command, INQUIRE TCPIP SERVICE

A new option, HOST, returns the host name, IPv4, or IPv6 address of the remote system, which replaces the IPADDRESS option for new programs. A new option, HOSTTYPE, returns the format of the contents of the HOST option, or the IPADDRESS option if HOST is not specified. A new option, IPRESOLVED, returns the IP address of the host and a new option, IPFAMILY, returns the format of the IPRESOLVED option.

Changed command, INQUIRE URIMAP

A new option, PORT, returns the port number used for the connection.

Changed command, INQUIRE WORKREQUEST

A new option, CLNTIP6ADDR, replaces the CLIENTIPADDR option for new programs. A new option, CLNTIPFAMILY, indicates the address type returned in CLNTIP6ADDR and CLIENTIPADDR.

A new option, TARGETSYS, returns the applid of the target CICS system, replacing the TSYSTEM option. A new option, TSYSTYPE, returns the format of the IP address returned in TSYSTEM or TARGETSYS.

Changes to CEMT (master terminal transaction) commands

INQUIRE CORBASERVER, INQUIRE IPCONN, INQUIRE TCPIP SERVICE, and INQUIRE URIMAP have new display items so that you can return IPv6 information. INQUIRE WORKREQUEST has new display fields.

Changed commands, INQUIRE CORBASERVER, INQUIRE IPCONN, and INQUIRE URIMAP

The option, HOST, now displays IPv6 addresses. The values, HOSTNAME, IPV4HOST, IPV6HOST, and NOTAPPLIC filter the contents of the new HOST option. A new option, IPRESOLVED, displays the IPv4 or IPv6 address of the host. The values IPV4FAMILY, IPV6FAMILY, and UNKNOWN filter the contents of the new IPRESOLVED option.

Changed command, INQUIRE TCIPSERVICE

A new option, HOST, displays the host name, or IPv4 or IPv6 address of the remote system. This option is an alternative to IPADDRESS. The values HOSTNAME, ANY, DEFAULT, IPV4HOST, IPV6HOST, and NOTAPPLIC filter the contents of the IPADDRESS option and the new HOST option. A new option, IPRESOLVED, displays the IPv4 or IPv6 address of the host. The values IPV4FAMILY, IPV6FAMILY, and UNKNOWN filter the contents of the new IPRESOLVED option.

Changed command, INQUIRE URIMAP

A new option, PORT, displays the port number of the connection.

Changed command, INQUIRE WORKREQUEST

A new option, CLNTIP6ADDR, displays the IPv6 address of the client that originated the request. The values IPV4FAMILY, IPV6FAMILY, and UNKNOWN filter the contents of the CLIENTIPADDR option and the new CLNTIP6ADDR option. A new option, TARGETSYS, displays the IPv4 or IPv6 address of the target system. This option is an alternative to TSYSTEM. The values IPV4TSYS, IPV6TSYS, APPLID, and NOTAPPLIC filter the contents of the TSYSTEM option and the new TARGETSYS option.

Changes to supplied transactions

The Web User Interface control transaction, COVC, now displays IPv6 information.

Note: This topic does not describe the following changes:

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

Changed transaction, COVC

A new field, TCP/IP Family, in the Status Details panel, displays whether the address of the connected region is an IPv4 or IPv6 address.

An existing field, ClientIp, in the User Sessions panel, now displays IPv6 addresses. The address extends over two lines, which reduces the number of users visible per page. IPv4 addresses are displayed on a single line.

Changes to CICSplex SM resource tables

Several CPSM operations base tables and resource definition base tables now include IPv6 information.

Changed tables

CSYSDEF

The CSYSDEF resource definition base table has one changed attribute, HOST.

EJCODEF, IPCONDEF, and URIMPDEF

These resource definition base tables have one changed attribute, HOST.

EJCOSE, IPCONN, and URIMAP

These operations base tables have one changed attribute, HOST, and three new attributes, HOSTTYPE, IPRESOLVED, and IPFAMILY.

HTASK and TASK

The HTASK history base table and the TASK operations base table have two changed attributes, CLIPADDR and OCLIPADR.

TASKASSC

The TASKASSC operations base table has one new attribute, CLNTIPFAMILY, and five changed attributes, IPFAMILY, ODIPFAMILY, CLIENTIPADDR, ODCLNTIPADDR, and SERVERIPADDR.

TCPDEF

The TCPDEF resource definition base table has one changed attribute, IPADDRESS, and one new attribute, HOST.

TCPIPS

The TCPIPS operations base table has one changed attribute, IPADDRESS, and four new attributes, HOST, HOSTTYPE, IPRESOLVED, and IPFAMILY.

WORKREQ

The WORKREQ operations base table has two changed attributes, CLIENTIPADDR and TSYSTEM, and two new attributes, CLNTIPFAMILY and TSYSTYPE.

Changes to CICSplex SM views and menus

The CICS operations views now display IPv6 information.

Changed views

EYUSTARTEJCOSE.DETAILED2, EYUSTARTIPCONN.DETAILED, and EYUSTARTURIMAP.DETAILED operations views (CICS operations views > Enterprise Java component operations views > CorbaServers)

Three new attributes, HOSTTYPE, IPFAMILY and IPRESOLVED are displayed.

EYUSTARTEJCOSE.TABULAR and EYUSTARTIPCONN.TABULAR operations views (CICS operations views > Enterprise Java component operations views > CorbaServers)

One new attribute, HOSTTYPE, is displayed.

EYUSTARTTASKASSC.DETAILED operations view (CICS operations views > Task operations views > Task association information)

A new attribute, CLNTIPFAMILY is displayed.

EYUSTARTTCPDEF.DETAILED and EYUSTARTTCPDEF.CREATE resource definition views (Administration views > CICS resource definitions > TCP/IP service definitions)

A new attribute, HOST is displayed.

EYUSTARTTCPIPS.DETAILED operations view (CICS operations views > TCP/IP service operations views > TCP/IP services)

Four new attributes, HOST, HOSTTYPE, IPFAMILY and IPRESOLVED are displayed.

EYUSTARTTCPIPS.TABULAR operations view (CICS operations views > TCP/IP service operations views > TCP/IP services)

A new attribute, HOSTTYPE is displayed.

EYUSTARTURIMAP.TABULAR operations view (CICS operations views > TCP/IP service operations views > URI maps)

Two new attributes, HOSTTYPE and PORT are displayed.

EYUSTARTURIMPDEF.DETAILED and EYUSTARTURIMPDEF.CREATE resource definition views (Administration views > CICS resource definitions > URI mapping definitions)

A new attribute, PORT is displayed.

EYUSTARTWORKREQ.DETAILED operations view (CICS operations views > Task operations views > Work requests)

Two new attributes, CLNTIPFAMILY and TSYSTYPE are displayed.

Changes to global user exits

Global user exits XWBAUTH, XWBOPEN, and XWBSNDO now support IPv6 addressing.

You must ensure that any programs that use these global user exits can process any IPv6 addresses that are returned by the UEPHOST parameter.

Changes to user-replaceable programs

New fields in the COMMAREAs used to pass parameters between the Analyzer, Converter, and Web error programs handle IPv6 addressing.

Analyzer program for CICS Web support

New fields, **wbra_client_ipv6_address** and **wbra_server_ipv6_address**, handle IPv6 addressing. User replaceable modules will behave as before with all IPv4 connections and you do not need to recompile existing modules unless they use the new parameters. If you introduce an IPv6 connection, the **wbra_client_ip_address** and **wbra_server_ip_address** fields are populated with zeros.

Converter program for CICS Web support

New fields, **decode_client_ipv6_address** and **decode_client_ipv6_address_string**, handle IPv6 addressing. User replaceable modules will behave as before with all IPv4 connections and you do not need to recompile existing modules unless they use the new parameters. If you introduce an IPv6 connection, the **decode_client_address** and **decode_client_address_string** fields are populated with zeros.

DFHWBEP, Web error program

New fields, **wbep_client_ipv6_address_len**, **wbep_client_ipv6_address**, **wbep_server_ipv6_address_len**, and **wbep_server_ipv6_address** handle IPv6 addressing. User replaceable modules will behave as before with all IPv4 connections and you do not need to recompile existing modules unless they use the new parameters. If you introduce an IPv6 connection, the **wbep_client_address_len**, **wbep_client_address**, **wbep_server_address_len**, and **wbep_server_address** fields are populated with zeros.

If a URIMAP resource associated with the current HTTP request is disabled, error message DFHWB0763 is issued and the Web error program is started. This message is written to the CICS log each time the disabled URIMAP resource is encountered. Use the XMEOUT global user exit to suppress or reroute your messages if you do not want them to be written to the CICS log.

Changes to monitoring data

DFHCICS and DFH SOCK fields for resolved IP addresses are replaced by new fields to accommodate larger IPv6 addresses. New fields in the transaction resource monitoring record, DFHMNRDS, support IPv6 addressing and retrieve transaction origin information.

Changed performance class group, DFHCICS

Field 368 is replaced by field 372.

Changed performance class group, DFH SOCK

Field 244 is replaced by field 318.

Changed transaction resource monitoring record, DFHMNRDS

A new field, MNR_ID_CLIPADDR, and a number of fields starting "MNR_ID_ORIGIN" support IPv6 addressing and retrieve transaction origin information.

Changes to statistics

The statistics sample program, DFH0STAT, and the statistics utility program, DFHSTUP, display new fields that support IPv6 addressing for CORBASERVER, IPCONN, URIMAP, and TCIPSERVICE.

IPFAMILY and IPRESOLVED fields are displayed in CORBASERVER, IPCONN, URIMAP, and TCIPSERVICE reports.

The PORT field is displayed in URIMAP reports.

The TCP/IP Services reports are now presented as one report called TCP/IP Services, including the HOST, IPFAMILY, and IPRESOLVED fields.

Changes to sample programs

Sample programs now allow you to test IPv6 addressing.

Changed sample autoinstall programs for terminals

The sample autoinstall programs, DFHZATDX, DFHZCTDX, DFHZPTDX, and DFHZDTDX, now support IPv6 addressing.

Changed sample analyzer programs

The sample analyzer programs, DFHWBADX, DFHWBAHX, DFHWBALX, and DFHWBAOX, now support IPv6 addressing.

Changed sample converter programs, DFH\$WBSC and DFH\$WBAU

The sample converter programs, DFH\$WBSC and DFH\$WBAU, now support IPv6 addressing.

Changed HTTP client sample exit programs, DFH\$WBEX, DFH\$WBX1, and DFH\$WBX2

The HTTP client sample exit programs, DFH\$WBEX, DFH\$WBX1, and DFH\$WBX2, now support IPv6 addressing in the UEPHOST parameter.

Changes to problem determination

New and updated messages help you to diagnose problems that relate to IPv6 addressing.

New messages

New warning messages, DFH5559, DFH5560, DFHCA5559, and DFHCA5560, provide information about the attributes you should use when you define your IPv4 and IPv6 connections.

A new error message, DFHWB0763, provides information when a URIMAP resource associated with the current HTTP request is disabled.

Changed messages

Messages DFHSO0110 and DFHSO0113 have changed to handle IPv6 addresses.

Chapter 23. Support for WebSphere MQ queue-sharing groups

You can now specify a WebSphere MQ queue-sharing group for the CICS-WebSphere MQ connection, so CICS uses any eligible queue manager in the group when it reconnects to WebSphere MQ, rather than waiting for a single queue manager. Queue-sharing groups increase reliability when you reconnect to WebSphere MQ, and help you standardize this aspect of CICS setup across CICS regions and z/OS images.

Instead of defining default settings for the CICS-WebSphere MQ connection in the DFHMQRPM operand of an INITPARM system initialization parameter, you must now use the new MQCONN resource definition. You can use the MQCONN resource definition to specify a queue-sharing group, or you can use it to specify the name of a single queue manager.

If you have specified a queue-sharing group for the connection, you can select appropriate resynchronization actions for CICS using the RESYNCMEMBER attribute of the MQCONN resource definition. Resynchronization works in the same way as it does for the group attach function for DB2. Resynchronization takes place when the connection to WebSphere MQ is lost and CICS is holding outstanding units of work for the last queue manager. You can choose whether CICS waits to reconnect to the same queue manager, or whether CICS makes one attempt to reconnect to the same queue manager, but if that attempt fails, connects to a different eligible queue manager in the group. A queue manager is eligible for connection to a CICS region if it is currently active on the same LPAR as the CICS region.

You can upgrade to use the new EXEC CICS and CEMT commands or CICSplex SM to start and stop the CICS-WebSphere MQ connection and change all the attributes of the connection. Alternatively, you can continue to use the existing methods of operating the CICS-WebSphere MQ adapter to initiate and manage connections between CICS and WebSphere MQ. You can use the CKQC transaction from the CICS-WebSphere MQ adapter control panels, or call it from the CICS command line or from a CICS application.

Changes to external interfaces

Some external interfaces now support WebSphere MQ queue-sharing groups.

Changes to system initialization parameters

The INITPARM system initialization parameter no longer applies to the CICS-WebSphere MQ connection.

You can no longer use the INITPARM system initialization parameter with a DFHMQRPM operand to specify a default WebSphere MQ queue manager name and initiation queue name for the CICS-MQ connection. Instead, set up an MQCONN resource definition for the CICS region to provide these defaults. CICS issues a warning message if the DFHMQRPM operand is present on INITPARM when you start the CICS-MQ connection, and the settings are ignored. The INITPARM system initialization parameter itself is still valid with other operands.

Changes to resource definitions

The new MQCONN resource definition defines the attributes of the connection between CICS and WebSphere MQ, and supersedes the DFHMQRPM operand of the INITPARM system initialization parameter.

For details of this resource, see “MQCONN resources” on page 300.

Changes to the system programming interface

New EXEC CICS commands act on the MQCONN and MQINI resources. You can use the SET MQCONN command to start and stop the CICS-WebSphere MQ connection, as an alternative to issuing CKQC START or STOP commands. The INQUIRE SYSTEM command now returns the name of the installed MQCONN resource definition.

Support for new resources

The following commands support the new MQCONN resource:

CREATE
DISCARD
All **CSD** commands

For information about the new resource, see “Changes to resource definitions”

New commands

INQUIRE MQCONN

Inquire on the attributes and status of the connection between CICS and WebSphere MQ.

For details of the command, see “INQUIRE MQCONN” on page 374.

INQUIRE MQINI

Inquire on the name of the default initiation queue to be used for the connection between CICS and WebSphere MQ.

For details of the command, see “INQUIRE MQINI” on page 377.

SET MQCONN

Change information about the attributes of the connection between CICS and WebSphere MQ, and start or stop the connection.

For details of the command, see “SET MQCONN” on page 390.

Changed command, INQUIRE SYSTEM

Because only one MQCONN resource definition can be installed in a CICS region, the INQUIRE MQCONN command does not require or return its name. If you need to find the name of the installed MQCONN resource definition, use the new MQCONN option on the INQUIRE SYSTEM command.

MQCONN(*data-area*)

Returns the 1- to 8-character name of the MQCONN resource definition that is currently installed for the CICS region or blanks if no MQCONN definition is currently installed. Only one MQCONN definition can be installed at a time. The MQCONN resource definition specifies the attributes of the connection between CICS and WebSphere MQ.

Changes to CEMT (master terminal transaction) commands

New CEMT commands are available to work with the MQCONN and MQINI resources and to start or stop the connection between CICS and WebSphere MQ.

Support for new resources

The **DISCARD** command supports the new MQCONN resource.

For information about the new resource, see “Changes to resource definitions” on page 154

New commands

INQUIRE MQCONN

Retrieve information about the connection between CICS and WebSphere MQ.

For details of the command, see “CEMT INQUIRE MQCONN” on page 411.

INQUIRE MQINI

Retrieve the name of the default initiation queue used for the connection between CICS and WebSphere MQ.

For details of the command, see “CEMT INQUIRE MQINI” on page 414.

SET MQCONN

Change information about the attributes of the connection between CICS and WebSphere MQ, and start or stop the connection.

For details of the command, see “CEMT SET MQCONN” on page 421.

Changed command, INQUIRE SYSTEM

If you need to find the name of the installed MQCONN resource definition, use the new MQCONN option on the INQUIRE SYSTEM command.

MQCONN(*value*)

Displays the name of the MQCONN resource definition that is currently installed for the CICS region. If no MQCONN resource definition is currently installed, the field is blank. Only one MQCONN resource definition can be installed at a time in a CICS region. The MQCONN resource definition specifies the attributes of the connection between CICS and WebSphere MQ.

Changes to CICSplex SM resource tables

The MQCONN resource definition and implicit MQINI resource definition are supported by new resource tables.

New tables

These new resource tables apply to the MQCONN and MQINI resource definitions.

MQCON

Shows the attributes of the CICS-WebSphere MQ connection, as defined by the MQCONN resource definition. The table includes:

- MQCONN resource definition name
- MQNAME and RESYNCMEMBER settings
- Connect and disconnect times for the CICS-WebSphere MQ connection

- CONNECT, DISCONNECT, FORCE, and DISCARD actions
- Statistics for the CICS-WebSphere MQ connection

MQCONDEF

BAS resource definition for the MQCONN resource definition. This table includes all the attributes for defining an MQCONN resource definition.

MQCINGRP

Information about BAS MQCONDEF resources contained in BAS resource groups.

MQINI

Operations resource table for the installed MQINI resource definition.

Changed tables

The existing MQCONN resource table, which shows statistics for the CICS-WebSphere MQ connection, now includes the name of the MQCONN resource definition. Note that the existing MQCONN resource table does not show the attributes of the new MQCONN resource definition. The new resource table that relates to the new MQCONN resource definition is called MQCON (with one N) to differentiate it from the existing resource table.

The existing MQCONN resource table also now includes statistics for the new API calls in WebSphere MQ Version 7.

The RESDESC resource table is updated to support MQCONDEF resources.

Changes to CICSplex SM views and menus

The new MQCON viewset supports actions and data for the CICS-WebSphere MQ connection, and the new MQCONDEF viewset is provided for the MQCONN resource definition. The existing MQCONN viewset is renamed and includes some additional information.

New views

WebSphere MQ Connection - MQCON

The views in this viewset shows the attributes of the CICS-WebSphere MQ connection, as defined by the MQCONN resource definition for the CICS region. The view includes the following items:

- MQCONN resource definition name
- MQNAME and RESYNCMEMBER settings
- Connect and disconnect times for the CICS-WebSphere MQ connection
- Action buttons to start and stop the CICS-WebSphere MQ connection
- Statistics for the CICS-WebSphere MQ connection, including statistics for the new API calls in WebSphere MQ Version 7

WebSphere MQ initiation queue - MQINI

The views in this viewset show the attributes of the implicit MQINI resource definition for the CICS region, which CICS installs when you install an MQCONN resource definition that includes a setting for the INITQNAME attribute.

WebSphere MQ connection definitions - MQCONDEF

The views in this viewset show the attributes of the MQCONN resource definition for the CICS region.

MQCONDEF in resource group - MQCINGRP

The views in this viewset display information about the membership of an MQCONN resource definition (MQCONDEF) in a resource group (RESGROUP).

Changed views

The existing MQCONN viewset is renamed to **WebSphere MQ Connection Statistics - MQCONN**. This viewset now displays the name of the MQCONN resource definition for the CICS region. The viewset also now displays statistics for the new API calls in WebSphere MQ Version 7.

The view "Resource descriptions - RESDESC" and the default map object EYUSTARTMAPBAS are changed to support the MQCONN resource definition.

Changes to supplied transactions

When you use the CKQC transaction from the CICS-WebSphere MQ adapter control panels or call it from the CICS command line or a CICS application, the default settings in the transaction are now taken from the MQCONN resource definition for the CICS region, rather than from an INITPARM system initialization parameter.

Changed transaction, CKQC

The default values supplied on the CICS-MQ adapter control panels for the queue manager name and initiation queue name are taken from the MQCONN resource definition and its implied MQINI resource definition.

When CICS is connected to WebSphere MQ, the field "QMgr name" in the Display Connection panel shows the name of the queue manager to which CICS is connected, or to which CICS is waiting to connect (if resynchronization is in progress). When CICS is not connected to WebSphere MQ, the field is blank. The new field "Mqname" in the Display Connection panel shows the name of the default WebSphere MQ queue-sharing group or queue manager for the connection, which you specified using the MQNAME attribute in the MQCONN resource definition. The value for the Mqname field is displayed whether or not CICS is connected to WebSphere MQ.

You can issue the CKQC START command without specifying a queue manager name, and CICS connects to the queue manager or a member of the queue-sharing group that you have specified in the MQCONN resource definition. You can also specify the name of a queue-sharing group on the CKQC START command in place of the name of a single queue manager. If you specify the name of a queue manager or queue-sharing group on the CKQC START command, the name that you specify replaces the setting for MQNAME in the installed MQCONN resource definition.

Note: This topic does not describe the following changes:

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

Other transactions

- Changes to CEMT commands are described in "Changes to CEMT (master terminal transaction) commands" on page 155.

- Changes to resources managed by the CEDA transaction are described in “Changes to resource definitions” on page 154.

Changes to global user exits

The resource management install and discard exit XRSINDI supports the new MQCONN and MQINI resources.

- For an MQCONN resource, the **UEPIDTYP** parameter returns a value of UEIDMQCN.
- For an MQINI resource, the **UEPIDTYP** parameter returns a value of UEIDMQIN.

For information about the new resource, see “Changes to resource definitions” on page 154.

Changes to statistics

The WebSphere MQ Connection statistics now have additional fields showing the name of the MQCONN resource definition, the MQNAME and RESYNCMEMBER attributes for the connection, and the connect and disconnect times. They also show the resource signature information for the MQCONN resource definition.

You can access the global statistics for the CICS-WebSphere MQ connection online using the EXEC CICS EXTRACT STATISTICS MQCONN command. They are mapped by the DFHMQGDS DSECT.

The WebSphere MQ Connection Report that is produced by the sample statistics report program DFH0STAT also shows the new fields.

Changes to problem determination

A number of new messages and trace points are issued in support of WebSphere MQ queue-sharing groups, and some existing messages are deleted.

New messages

The new messages that relate to WebSphere MQ queue-sharing groups are DFHMQ0209, DFHMQ0210, DFHMQ0218, DFHMQ0303, DFHMQ0317, DFHMQ0320, DFHMQ0324, DFHMQ0325, DFHMQ2064, DFHMQ2100, DFHMQ2101, DFHMQ2102, DFHMQ2103, DFHMQ2107, DFHMQ2108, DFHMQ2109.

CICS issues warning message DFHMQ0218W if settings for the CICS-WebSphere MQ connection in the DFHMQPRM operand of an INITPARM system initialization parameter are present when you start the CICS-WebSphere MQ connection. The settings are ignored. You must set up an MQCONN resource definition to specify settings for the CICS-WebSphere MQ connection.

Changed messages

Message DFHMQ0307I, which reports successful connection to a WebSphere MQ queue manager, now also reports the release of WebSphere MQ to which CICS is connected.

Message DFHMQ0434E no longer references the DFHMQPRM operand of the INITPARM system initialization parameter in the message explanation.

Discontinued messages

Messages DFHMQ0212E, DFHMQ0213E, DFHMQ0214E, DFHMQ0216E, and DFHMQ0217E are no longer issued. These messages related to incorrect settings specified in the DFHMQRPM operand of the INITPARM system initialization parameter.

Discontinued abend code AMQL

Abend code AMQL, which was issued if the CICS enqueue domain could not create an enqueue pool for use in CICS-MQ adapter processing, is no longer used.

New trace points

The new trace points that relate to WebSphere MQ queue-sharing groups are AP A010 - AP A012, AP A0C0 - AP A0CD, AP A0D0 - AP A0DF, AP A0E0 - AP A0EF, and AP A0F0 - AP A0F9.

Changed trace points

Trace points AP A06A and AP A06B have an additional data item.

Changes to security

You can choose to enable command security checks for MQCONN and MQINI resource commands.

The CREATE, DISCARD, INQUIRE, and SET commands for MQCONN resources and the INQUIRE command for MQINI resources are subject to command security checks. Resource security does not apply. If you want to enable command security for these commands, add the resource names MQCONN and MQINI to the CCICSCMD or VCICSCMD supplied resource classes.

The QUERY SECURITY command has new RESID values of MQCONN and MQINI for use with RESTYPE('SPCOMMAND').

Chapter 24. Improvements to workload management support

Dynamic workload management takes advantage of the Coupling Facility to store current region status information posted directly from CICS. The stored region status is then used to make more efficient routing decisions. CICSplex SM now supports percentile goals when using goal algorithms.

Dynamic workload management provides these benefits:

- Additional WUI views that can assist with problem determination in route selection
- Exploitation of the z/OS coupling facility to improve cross-sysplex routing of distributed workloads
- Dynamic tuning of CICSplex SM workload manager resource consumption in the coupling facility
- Improved recognition of CICS region status for more efficient WLM routing decisions
- Implementation of the new functions requiring minimal configuration effort

When you upgrade from a previous release of CICS TS, existing workloads can be migrated without interrupting them, or modifying their definitions.

Sysplex optimized workload routing overview

In CICS TS for z/OS, Version 4.1, workload throughput is improved through a more efficient workload management optimization function. This function is most effective for distributed workloads, for which the routing and target regions are managed by different CMASs. With new CPSM WUI views, you can monitor the distribution of dynamic workloads through your CICSplex.

Sysplex optimized workload routing is enabled at the z/OS coupling facility level by a region status (RS) server.

When a target region is running in optimized mode, the target region maintains the task count using the CICS transaction manager. The count includes all tasks in the CICS region, not just those that are dynamically routed. The load value for the CICS region, with its basic health status, is periodically broadcast to the coupling facility, where it is available for interrogation by other CICS regions and CMASs and by other CICS address spaces. If region status data is available, CICSplex SM uses the data when it makes its dynamic routing decision.

For sysplex optimized workloads, routing regions review the same status data in the coupling facility for a potential target region regardless of which CMAS manages it. As a result, the routing region is using status data that might be updated many times a second to evaluate a target region, rather than status data that might be up to 15 seconds old. The refresh interval can vary from 2 seconds down to 1 millisecond. As the scale of this value is reduced, the usage effect on the coupling facility increases. Choose a value that provides a balance between workload throughput and the effect on the coupling facility. The default refresh value is 200 milliseconds. In an environment in which all routing targets are in a similar health and connectivity state, the spread of work across the workload target scope is more evenly distributed than in nonoptimized mode.

If the coupling facility is not available, workload routing is managed by CICSplex SM Workload Manager using z/OS data spaces owned by a CMAS to share cross-region load and status data.

Benefits of sysplex optimized workload routing

A sysplex optimized workload is a workload that is best suited to workloads contained in a single sysplex. For a workload that runs in a CICSplex that spans more than one sysplex, the benefits of optimized routing are reduced because region status data stored in the z/OS coupling facility is not shared across sysplexes.

Sysplex optimized workload routing is beneficial in the following scenarios:

- When the topology of a CICSplex is such that regions in a workload can be managed by the same CMAS, and nondynamic throughput is a high proportion of the workload.
- When a workload consists of routers and targets managed by different CMASs and the bulk of the dynamic traffic flows through the DSRTPGM exit. For example, if you use MQ triggers to feed transactional data into CICSplexes, where the trigger regions tend to be managed by different CMASs to the processing regions. In these instances, the benefit of running workloads in optimized state is that no workload batching occurs, and the overall workload runs through faster. Fewer, if any, routed transactions are waiting in the queue of a CICS region already at its MAXTASKS limit.

Impact of sysplex optimized workload routing on the coupling facility

Caching mechanisms are built into the coupling facility (CF) to reduce the number of I/O operations. CICS region status data is broadcast to the CF by target regions, and the data is subsequently read back by the routing regions when a route decision is being made. If CICS status data is broadcast at every change instance, and read back on every occasion that a route decision is made, the impact to the coupling facility might be considerable.

To reduce the impact to the coupling facility, CICS provides two system management parameters:

- **READRS** controls how long region status data is cached by a routing region before requesting a refresh.
- **UPDATERS** controls how often the CF is updated with task throughput data.

Support for z/OS workload manager percentile goals

CICSplex SM extends the workload balancing function of workload management (WLM) to support z/OS percentile goals in addition to average response-time goals.

CICSplex SM workload management optimizes processor capacity in your enterprise by dynamically routing transactions and programs to the CICS region that is the most appropriate at the time, taking into account any transaction affinities that exist.

Workload balancing makes the best use of the available CICS systems, and provides opportunities for increased throughput and performance by routing transactions or programs among a group of target regions according to the

availability and activity levels of those target regions. CICSplex SM decides which target region processes the work by using a routing algorithm.

CICSplex SM uses the *goal algorithm* to select the target region that is most likely to meet the transaction response-time goals set for it using the Workload Manager component of z/OS. Before CICS TS 4.1, average response-time goals are the only response-time goals supported by CICSplex SM.

You can use goal mode to provide efficient routing decisions, where routers and targets are managed by the same CMAS, in the following scenarios:

- Dynamic routing using DTRPGM for dynamic transactions
- Dynamic routing using DTRPGM for EXEC CICS START TERMID over APPC or MRO connections
- Distributed routing using DSRTPGM for business transaction service routing

CICS TS 4.1 provides support for z/OS workload manager percentile goals.

A percentile goal sets the percentage of transactions to be completed in the target response time. For example, consider a percentage goal set to 80% and response time set to 1 second. If a period contains 200 transactions, 160 of them must complete with a response time less than or equal to 1 second to meet the percentile goal.

Before CICS TS 4.1, any percentile goals were ignored by CICSplex SM. So customers who use the CICSplex SM goal algorithm and have z/OS WLM percentile goals will see different routing behavior when they migrate to CICS TS 4.1.

Use z/OS WLM percentile response time goals in preference to average response time goals in situations where a workload can have a few unusually long transactions that distort the average response time.

Terminology

New terms are introduced to explain the improvements to workload management support.

region status server

A region status server is the bridge between CICS and a coupling facility structure that is used to record region status data. The region status server function can be shared with existing coupling facility data table (CFDT) servers. You can allocate a discrete CFDT server and pool for optimal workload throughput.

Changes to external interfaces

Some external interfaces have changed to support improvements to workload management.

Changes to CICSplex SM resource tables

The CICSplex SM resource tables now support update frequency and read interval attributes at the coupling facility.

New table, WLMATARG

This CPSM Workload Manager object describes a CICS system used as a target region in an active workload.

For more information, see the *CICSplex System Manager Resource Tables Reference*.

Changed tables, CPLEXDEF, CPLXCMAS, CMASPLEX, CICSPLEX, CSYSDEF, EPLEXCHG, and MAS

The CPLEXDEF, CPLXCMAS, CMASPLEX, CICSPLEX, CSYSDEF, EPLEXCHG, and MAS tables have five new fields that are used to update and read from the coupling facility.

Field name	Description
BOTRSUPD	RS server bottom tier
READRS	RS server read interval
RSPoolID	RS server pool name
TOPRSUPD	RS server top tier
UPDATERS	RS server update frequency

Changed table WLMWORK

The WLMWORK table has a new field that is used to show the optimized workload status.

Field name	Description
OWSTATE	Optimized workload status

Changed table WLMAWTOR

The WLMAWTOR table has a new field that is used to show the optimization status of the router.

Field name	Description
OWSTATE	Optimization status of the router

Changed table WLMAWAOR

The WLMAWAOR table has new fields that are used to show the status of the active workload targets.

Field name	Description
BOTRSUPD	RS server bottom tier
CFUPDCNT	Coupling facility update count
EVENTS	RTA events active
HLTHDUMP	Region dumping status
HLTHMAXT	Region at MAXTASK status
HLTHSOS	Short-on-storage health status

Field name	Description
HLTHSTALL	Region stall status
MAXTASKS	The local MAXTASK specification
OWSTATE	Sysplex optimized routing status
READRS	RS server read interval
ROUTEWGHT	The calculated WLM routing weight for the local region
ROUTINGLOAD	Routing Load
RSPOLID	RS server pool name
TASKINC	Task count increment value
TASKCNT	Task count used for dynamic routing
TASKLOAD	Task load percentage
TOPRSUPD	RS server top tier
UPDATERS	RS server update frequency
WLMQMODE	Task load queue mode
WLMTHRSH	Task load health threshold

Changes to CICSplex SM views and menus

CICSplex SM views and menus now display region status server-related attributes.

Changed menu

EYUSTARTWORKLOAD menu

To access this menu from the main menu, click: **Active workload view**. The main **Active Workload** review menu **EYUSTARTWORKLOAD** is changed:

- A new view is added to the **WLMWAOR** view set:
 - Active workload target distribution factors view (WLMWAOR.TABULAR2);

The existing **Routing regions** link has been moved under the **Active workloads** heading to signify that it no longer requires an explicit workload name.

New views

EYUSTARTWLMWAOR.DETAIL view

To access from the main menu, click: **Active workload views > Target regions in an active workload**. The WLMWAOR detail view describes all the existing and new attributes of the WLMWAOR base table. The CICS region hyperlink that was active on the CICS region name column of the tabular view is now in the same field name in this detail view.

To enable comparisons between the state and usage of Target regions in workloads, the WLMWAOR view set now contains a new tabular view.

The tabular2 view, “Target regions in an active workload”, displays the distribution factors for all workload and AOR records.

Both tabular views are launched with a Workload name of “*”, showing all workloads.

EYUSTARTWLMATARG view set

To access from the main menu, click: **Active workload views > Active target regions**. You review active target regions using the EYUSTARTWLMATARG view set, without deference to any workload associations. The only associating attribute between the region details and a workload is the **workload count** attribute. Using the workload count hyperlink, users can determine the workloads for which the region is a target.

The EYUSTARTWLMATARG view set now contains two new tabular views and a detailed view.

The WLMATARG view set - Tabular 2 now displays the distribution statistics for target regions regardless of their workload affiliations. Using the view set, you can perform comparisons between the state and usage of target regions in the workload.

The WLMATARG view set - Tabular 3 displays complete workload target details in list form on a CMAS to CMAS basis. Using this view you may directly compare and summarize all data relating to workload target regions as recognized by each CMAS in a workload.

All tabular views are launched with a Workload name of '*', showing all Workloads.

Changed views

EYUSTARTCPLEXDEF view set

To access from the main menu, click: **Active workload views > CICSplex definitions**. You define and modify CICSplexes using the EYUSTARTCPLEXDEF view set.

EYUSTARTCICSPLEX view set

To access from the main menu, click: **CICSplex SM operations views > CMASs managing CICSplex**. You review CICSplexes using the EYUSTARTCICSPLEX view set. The CICSPLEX detail view now displays the new region status server-related attributes.

EYUSTARTCPLXCMAS view set

To access from the main menu, click: **Admin views > CMAS configuration administration views > CMAS in CICSplex definitions**. You review associations between CICSplexes and CMASs using the EYUSTARTCPLXCMAS view set. The CPLXCMAS detail view now displays the new region status server-related attributes.

EYUSTARTCSYSDEF view set

To access from the main menu, click: **Active workload views > CICS system definitions**. You define CICS regions to CICSplex SM using the CSYSDEF base table. The CPLEXDEF detail view now supports the specification of the default region status server read interval and update frequency for optimized workloads. Also associated with the update frequency are the task load top tier and task load bottom tier values. The default value for all these attributes is INHERIT. The WLM EYUPARMS for **WMLLOADTHRESH** and **WLMCOUNT** are no longer supported. These are now defined as attributes in the CSYSDEF base table. The attributes are: **Task load health threshold** and **Task load queue mode**.

EYUSTARTMAS view set

To access from the main menu, click: **Active workload views > Active MASes in CICSplex**. You monitor the CICSplex SM component of an active CICS region using the MAS view set. The MAS detail view now displays the current settings of the region status server read interval and update frequency for optimized workloads. Associated with the update frequency are the task load top tier and task load bottom tier values, which can be modified in this view set. The new **Task load health threshold** and **Task load queue mode** are also modified from this view set.

EYUSTARTWORKLOAD menu

To access from the main menu, click: **Active workload views**. You review workload using the **EYUSTARTWORKLOAD** menu. Two new links are provided to display "Active workload target distribution factors" and "Target region distribution statistics". Both of these views are now supported in the **WLMWAOR** view set. The existing "Routing regions" link has moved under the "Active workloads" heading to signify that it no longer requires an explicit workload name.

EYUSTARTWLMAWORK view set

To access from the main menu, click: **Active workload views > Active workloads**. You review active workloads using the **EYUSTARTWLMAWORK** view set. The **WLMAWORK** list and detail views now display the overall sysplex optimized routing status. In the Workload List view, the **Share status** field, is now removed. The **Share status** field was an indicator of whether the Workload shared a CMAS with a CICS TS for z/OS version 1.3.

EYUSTARTWLMAWTOR view set

To access from the main menu, click: **Active workload views > Active routing regions**. You review active workload routers using the **EYUSTARTWLMAWORK** view set. The **WLMAWTOR** list view now displays the sysplex optimized routing status of the router.

EYUSTARTWLMAWAOR view set

To access from the main menu, click: **Active workload views > Active workload target distribution factors**. You review active workload targets (AORs) using the **EYUSTARTWLMAWAOR** view set. The **WLMAWAOR** tabular view now includes the sysplex optimization status. The hyperlink on the target region name column now links to a new **WLMAWAOR** detail view. The **TABULAR** view now accepts workload names of '*' to list all routers in all workloads. A new query parameter is included so that you can enter a transaction name. You can correctly calculate the routing weight according to the supplied transaction name using this new parameter.

Changes to problem determination

Improvements to workload management introduce new messages, abend codes, trace points, and a new trace selection parameter.

New component codes

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

RS The following component code has been added to support the Region Status domain.

Component code	Component keyword	Description
RS	REGIONSTAT	Region Status domain

You use the component code in the following ways:

- To select the level of standard and special tracing in each component:
 - In the CETR transaction.
 - In the **STNTRxx** and **SPCTRxx** system initialization parameters.
 - In the **INQUIRE TRACETYPE** and **SET TRACETYPE** system programming commands. If you have a component keyword, you can use it instead of the component code in these commands.
- To specify the areas of CICS storage that you want to be included in a formatted dump and the amount of data you want formatted.
- To specify the trace entries that you want to include in a formatted dump and in the output from the trace utility program.

On output, CICS uses the component code to identify messages and trace entries. New messages with the prefix DFHRS relate to the initialization and operation of the Region Status (RS) domain.

New messages

New messages DFHRS0001 and DFHRS0002 are issued by the Region Status Domain.

New messages EYUWM0438I and EYUWM00503I to EYUWM00508I are issued by CICSplex SM workload management.

New trace points

New trace points are provided, in the range RS 0100 - RS 0504.

Changes to security

To implement the improvements to workload management support, additional permissions must be granted to access the coupling facility data table (CFDT).

Changes to the security manager

Access to the routing CFDT pool is controlled using RACF or an equivalent external security manager. Each region status server region must be given access to its associated coupling facility list structure according to the CFRM security rules. This access requires ALTER access to a RACF resource in the FACILITY class called *IXLSTR.structure*.

If you do not modify the default RS server pool name specified in the CICSplex definition, you must give CONTROL access to a resource called DFHCF.DFHRSTAT for the server region, so that it can establish itself as a server for the given pool name. If you use your own privately specified pool name for recording region status data, you must give CONTROL access to the resource *DFHCF.user_specified_pool_name*. Each CICS region that is allowed to share objects in that pool must be given UPDATE access to the same resource.

Chapter 25. CICS management client interface

The CICS management client interface (CMCI) is a system management application programming interface, designed using Representational State Transfer (RESTful) principles, for use by HTTP client applications including IBM CICS Explorer. You can use this interface to develop HTTP client applications that manage installed and definitional CICS and CICSplex SM resources on CICS regions being managed by CICSplex SM or CICS resources on stand-alone CICS regions.

The client initiates an HTTP request to the CICS management client interface. If the interface determines that the request is valid, it constructs a CICSplex SM API command or, in the case of a stand-alone CICS region, a CICS system command. After running the command the CMCI creates an HTTP response. If the request is successful, this takes the form of an HTTP 200 (OK) response and an XML feed containing a result set, which it passes back to the client. If the request is not successful, the response consists of a non-OK HTTP response code with details of the failure.

The format for CICS management client interface HTTP requests and responses is based on the HTTP/1.1 protocol. See the *CICS Internet Guide* for more information about this protocol.

A CICS management client interface request takes the form of an HTTP header followed by a URI (Universal Resource Identifier) and, where appropriate, an XML body containing details of any changes to be made to CICS or CICSplex SM resources.

The header incorporates one of the following HTTP methods:

DELETE

Removes resources from the CICSplex SM data repository, removes resources from the CSD, or discards installed resources.

GET Retrieves information about resources on the CICSplex SM data repository, retrieves information about resources on the CSD, or retrieves information about installed resources.

POST Creates resources on the CICSplex SM data repository or resources in the CSD.

PUT Updates existing resources in the CICSplex SM data repository, updates existing resources in the CSD, or sets attributes and performs actions on installed resources. Also performs actions on CICSplex SM and CSD resources.

The URI includes the name of a CICS or CICSplex SM resource, and specifies a series of parameters that refine the scope and nature of the query to identify one or more instances of the specified resource. In a GET request, the URI also specifies whether the API retains or discards a set of results. If the API retains the results, a new request can act on the retained results without having to repeat the retrieval operation. You can also use subsequent requests to page through the retained results selecting one or more records at a time.

POST and PUT requests include an XML body. In a PUT request the body contains either details of the changes to be made to resource attributes, or the action to be

performed on the targeted resources. In a POST request, the body incorporates the attribute values you wish to give to the new resource instance.

GET and DELETE requests do not require an XML body. If additional parameters are required for a DELETE request, those parameters must be included in the URI and can optionally be added to the XML body. For example; if you are deleting a CSD resource definition you must include `PARAMETER=CSDGROUP(csdgrp)` in the URI, and you can optionally add `<parameter name="CSD"/>` to the XML body.

You can use CMCI either in a CICSplex SM environment or in a stand-alone CICS region (SMSS).

If you use CMCI with CICSplex SM, you can manage definitional, operational, and CSD resources in all of the regions managed by CICSplex SM.

If you use CMCI in a stand-alone CICS region, you can manage only the operational and CSD resources associated with that region. If you use CMCI in a stand-alone CICS region, the context is specified as the application ID of that CICS region.

Setting up the CICS management client interface

You set up the CICS management client interface (CMCI) differently depending on whether you are setting it up in a CICSplex SM environment or in a stand-alone CICS region (SMSS). In CICSplex SM, you must configure a WUI server and set the `CMCIPORT` WUI server initialization parameter. In a stand-alone CICS region, you must install `TCPIPSERVICE` and `URIMAP` definitions and alter the CICS startup JCL.

CICS management client interface DELETE requests

The CICS management client interface uses the HTTP DELETE method to remove resources from the data repository, or to discard installed resources from CICS or CICSplex SM.

The client forms a DELETE request using an HTTP header comprised of the following parts:

- The method name, in this case DELETE
- The URI that identifies the resources to be deleted
- The HTTP version
- Authorization credentials if required

DELETE requests can operate on either installed resources or on definitional resources. The interface constructs a CICS DISCARD command when operating on operational resources, and a CICSplex SM REMOVE command when operating on definitional resources.

For example, to delete all transaction definitions in CICSplex PLEX1 that have a name beginning with TR and a program beginning with P, the client constructs the following HTTP header:

```
DELETE /CICSSystemManagement/CICSDefinitionTransaction/PLEX1?CRITERIA=NAME%3DTR%2A%20
AND%20PROGRAM%3DP%2A HTTP/1.1
Host: example.com:23792
Authorization: Basic R1JFRDpQQVNTVzBSRA==
```

The initial line has three parts, separated by spaces:

- The method name
- The request URI, which is specified as an absolute path that begins `/CICSSystemManagement/` immediately followed by the external name that identifies the resource together with the parameters and filters that determine which instances of that resource are to be deleted. To make the URI suitable for processing by the CICS management client interface, the client replaces certain restricted characters such as spaces and asterisks with escape sequences.
- The HTTP version, which, when connecting to the CICS management client interface, is always HTTP/1.1

The second line identifies the host name and port number of the target system separated by a colon as specified in the URI. This line ends with a carriage return, followed by a line-feed.

The optional third line contains authorization credentials. If your system is running with the CICS system initialization parameter **SEC** as YES, you must supply a user ID and password in base-64 in a basic authentication header. In this example the user ID is *FRED* and the password is *PASSWORD*.

The HTTP header is then finalized by a final carriage return and line-feed on a separate line.

Response to a DELETE request

On the completion of a DELETE request, the client receives a response from CMCI made up of an HTTP header, and an XML feed containing a result summary and details of the deleted resources.

The header consists of an HTTP response code, the date and time, and details of the server and content type. The following example shows the header for a successful DELETE request:

```
HTTP/1.1 200 OK
Cache-Control: no-store
Date: Tue, 02 Jun 2009 14:51:37 GMT
Server: IBM_CICS_Transaction_Server/4.1.0(zOS)
Content-Type: application/xml; charset=UTF-8
Transfer-Encoding: chunked
```

The body of the response consists of a `<response>` root element containing a `<resultsummary>` element displaying summarized information about the request, for example:

```
<response xmlns="http://www.ibm.com/xmlns/prod/CICS/smw2int"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ibm.com/xmlns/prod/CICS/smw2int
http://example.com:30061/CICSSystemManagement/schema/CICSSystemManagement.xsd" version="1.0"
connect_version="0410">
<resultsummary api_response1="1024" api_response1_alt="OK"
api_response2="0" api_response2_alt="" recordcount="1" successcount="1" />
</response>
```

CICS management client interface GET requests

The CICS management client interface uses the HTTP GET method to retrieve resources from CICS or CICSplex SM.

The client forms a GET request from the following parts:

- The HTTP method, in this case GET

- The URI that identifies the resources to be retrieved and indicates whether the results are stored for later use or discarded
- The HTTP version
- Authorization credentials if required

For example, to retrieve all CICS local files in the CICSplex PLEX1, the client might construct the following request:

```
GET /CICSSystemManagement/CICSLocalFile/PLEX1/ HTTP/1.1
Host: example.com:22958
Authorization: Basic RIJFRDpQQVNTVzBSRA==
```

The initial request line of the HTTP header has three parts, each separated by spaces and ended by a carriage return, followed by a line-feed:

- The HTTP method name
- The request URI, which is specified as an absolute path that begins /CICSSystemManagement/ immediately followed by the external name that identifies the resource, and the parameters and filters that determine which instances of that resource are to be retrieved.

A GET request can either operate directly on CICS or CICSplex SM resources or on the results retained from a previous GET request. You can retain a set of results by specifying the **NODISCARD** option on the GET request URI. Each set of retained results is identified by a unique cache token.

If the request is operating directly on a resource, you specify a resource name, which can be of an operational or a definitional resource, add the context and optionally the scope, and then further refine and filter the results by including one or more query parameters.

If the request operates on a set of retained results, you identify the results by replacing the resource name with CICSResultCache and specifying a cache token. You can narrow down the request to one or more records in the retained results by adding values for the **index** and **count** options.

- The HTTP version, which, when connecting to the CICS management client interface, is always HTTP/1.1

The second line identifies the host name and port number of the target system separated by a colon as specified in the URI. This line ends with a carriage return, followed by a line-feed.

The optional third line contains authorization credentials. If your system is running with the CICS system initialization parameter **SEC** as YES, you must supply a user ID and password in base-64 in a basic authentication header. In this example the user ID is *FRED* and the password is *PASSWORD*.

The HTTP header is then finalized by a final carriage return and line-feed on a separate line.

Response to a GET request

On the completion of a GET request, the client receives a response from the CICS management client interface made up of an HTTP header, an XML feed containing a result summary, and details of the retrieved resources. However, if the URI included the **SUMMONLY** parameter, the response consists only of the HTTP header and the result summary.

The header consists of an HTTP response, the date and time, and details of the server and content type. The following example shows the header for a successful GET request:

```
HTTP/1.1 200 OK
Cache-Control: no-store
Date: Wed, 06 Aug 2008 08:32:00 GMT
Server: IBM_CICS_Transaction_Server/4.1.0(zOS)
Content-Type: application/xml; charset=UTF-8
Transfer-Encoding: chunked
```

The body of the response consists of a <response> root element containing a <resultsummary> element displaying summarized information about the request, and typically a <records> element containing details of the retrieved resources. You can suppress the <records> element of the response by specifying the **SUMMONLY** parameter in the URI. If the request completed with errors, diagnostic information is provided in an <errors> element.

CICS management client interface POST requests

The CICS management client interface uses the HTTP POST method to create resources in CICS or CICSplex SM.

The client forms a POST request from the following parts:

- The HTTP method, in this case POST
- The URI that identifies the resources to be retrieved and indicates whether the results are cached for later use or discarded
- The HTTP version
- Authorization credentials if required
- The XML body containing details of the resource to be created.

For example,

```
POST /CICSSystemManagement/CICSLocalFile/PLEX1/ HTTP/1.1
Host: example.com:22958
Authorization: Basic RIJFRDpQQVNTVzBSRA==
```

```
<request>
<create>
<parameter name="RESGROUP" value="BASIC" />
<attributes name="FILE1" defver="1" />
</create>
</request>
```

The initial line has three parts, separated by spaces:

- The HTTP method name
- The request URI, which is specified as an absolute path that begins /CICSSystemManagement/ immediately followed by the external name that identifies the resource, and the parameters and filters that identify the instance of the resource that is to be created.
- The HTTP version, which, when connecting to the CICS system management client API, is always HTTP/1.1

The second line identifies the host name and port number of the target system separated by a colon as specified in the URI. This line ends with a carriage return, followed by a line-feed.

The optional third line contains authorization credentials. If your system is running with the CICS system initialization parameter **SEC** as **YES**, you must supply a user ID and password in base-64 in a basic authentication header. In this example the user ID is *FRED* and the password is *PASSWORD*.

The HTTP header is then finalized by a final carriage return and line-feed on a separate line.

The remainder of the request is the XML body specifying the attributes of the new resource. The body consists of a the XML <request> element containing one <attributes> child element.

Response to a POST request

On the completion of a POST request, the client receives a response from the client API made up of an HTTP header, an XML feed containing a result summary, and details of the resources created. However, if the request included the **SUMMONLY** parameter, the response consists only of the HTTP header and the result summary.

The header consists of an HTTP response, the date and time, and details of the server and content type. The following example shows the header for a successful request:

```
HTTP/1.1 200 OK
Cache-Control: no-store
Date: Wed, 06 Aug 2008 08:32:00 GMT
Server: IBM_CICS_Transaction_Server/4.1.0(zOS)
Content-Type: application/xml; charset=UTF-8
Transfer-Encoding: chunked
```

The body of the response consists of a <response> root element containing a <resultsummary> element displaying summarized information about the request, and typically a <records> element containing details of the new resource. However, you can suppress the <records> element of the response by specifying the **SUMMONLY** parameter in the URI. If the request completed with errors, diagnostic information is provided in an <errors> element.

CICS management client interface PUT requests

The CICS management client interface uses the HTTP PUT method to change the attributes of CICS or CICSplex SM resources or to perform actions, including **INSTALL**, on those resources.

The client forms a PUT request from the following components:

- The HTTP method, in this case **PUT**
- The URI that identifies the resources to be changed
- The HTTP version
- Authorization credentials if required.
- The HTTP body that specifies in XML the changes to be made, or the action to be performed

For example, to update attributes of transaction definitions in CICSplex PLEX1 that have a name beginning with **TR** and a program beginning with **P**, the client constructs the following request:

```

PUT /CICSSystemManagement/CICSDefinitionTransaction/PLEX1?CRITERIA=NAME%3DTR%2A%20
AND%20PROGRAM%3DP%2A
HTTP/1.1
Host: example.com:23792
Authorization: Basic RIJFRDpQQVNTVzBSRA==

<request>
  <update>
    <attributes STATUS="ENABLED" />
  </update>
</request>

```

The initial line has three parts, separated by spaces:

- The method name
- The request URI, which is specified as an absolute path that begins `/CICSSystemManagement/` immediately followed by the external name that identifies resource together with the parameters and filters that determine which instances of that resource type are to be selected. To make the URI suitable for processing by the CICS management client interface, the client replaces certain restricted characters such as spaces and asterisks with escape sequences.
- The HTTP version, which, when connecting to the CICS management client interface, is always HTTP/1.1

The second line identifies the host name and port number of the target system separated by a colon as specified in the URI. This line ends with a carriage return, followed by a line-feed.

The optional third line contains authorization credentials. If your system is running with the CICS system initialization parameter **SEC** as YES, you must supply a user ID and password in base-64 in a basic authentication header. In this example the user ID is *FRED* and the password is *PASSWORD*.

The HTTP header is then finalized by a final carriage return and line-feed on a separate line.

The remainder of the request is the XML body specifying the change to be made to the identified resource instances. In this case the request changes the STATUS attribute of the selected transaction definitions to ENABLED. The body consists of a the XML `<request>` element containing either one `<attributes>` element or one `<action>` element.

A PUT request can operate on either installed resources or on definitional CICS or CICSplex SM resources. A single PUT request can either specify attribute changes or perform an action. You cannot combine attribute changes and an action in a single request.

The PUT method does not support all actions. The following actions are supported by the POST method:

- CREATE for definitional resources.
- ADD for SYSDUMP and TRANDUMP resource types.

The following actions are supported by the DELETE method:

- DISCARD for operational resources.
- REMOVE for definition resources.

Response to a PUT request

On completion of a PUT request, the client receives a response from the CICS management client interface made up of an HTTP header, and an XML feed containing a result summary, and if the request is successful, details of the changed resources.

The header consists of an HTTP response, the date and time, and details of the server and content type; for example:

```
HTTP/1.1 200 OK
Cache-Control: no-store
Date: Wed, 10 Aug 2008 12:56:00 GMT
Server: IBM_CICS_Transaction_Server/4.1.0(zOS)
Content-Type: application/xml; charset=UTF-8
Transfer-Encoding: chunked
```

If the PUT request fails, the server issues an HTTP non-OK response. For example if the body of the request is not valid, the header includes the following 400 response:

```
400 Bad request. The body of the request is invalid.
```

This response can occur if the client sends multiple tags on a PUT request, or the body of the request contains an unknown tag, or the tag name does not match the model name in the URI.

The body of the response consists of a <response> root element containing a <resultsummary> element displaying summarized information about the request, and typically a <records> element containing details of the selected resources. However, you can suppress the <records> element of the response by specifying the **SUMMONLY** parameter in the URI. If the request completed with errors, diagnostic information is provided in an <errors> element.

CICS management client interface XML

CICS management client interface requests and responses contain information in XML format. The unique XML elements in CMCI are defined in a schema named CICSSystemManagement. This schema is constructed by incorporating a separate schema for each resource type.

Client applications need to access the CICSSystemManagement schema to validate and format the information in the body of responses to CICS management client interface requests. Clients can access the schema that defines this output XML at the following URI: `http://hostname:portnumber/CICSSystemManagement/schema/CICSSystemManagement.xsd`.

Note: CMCI can interpret only characters that are handled by the EBCDIC 037 codepage. Using other characters in CMCI requests leads to unpredictable results.

Changes to external interfaces

Some external interfaces have changed to support the CICS management client interface.

Changes to system initialization parameters

CMCIPORT is a new CICSplex SM Web User Interface server initialization parameter that is required to set up the CICS management client interface with CICSplex SM.

You use CMCIPORT to specify the TCP/IP port number allocated to CMCI. This parameter is required to use CMCI in a CICSplex SM environment. Specify a number in the range 1 - 65535 but do not use the same port number as specified for the WUI in the TCPIPPORT parameter. When you set this parameter, CMCI is installed in your WUI server region and TCPIPSERVICE and URIMAP definitions are autoinstalled to support CMCI. The level of security for CMCI is derived from the SEC CICS system initialization parameter and the TCPIPSSL WUI server initialization parameter.

CMCIPORT is included in DFHISTAR. Running the DFHISTAR job to set up your CICS environment ensures that CMCI is installed in your WUI server.

Changes to CICSplex SM resource tables

All resource tables that support CICS management client interface requests include a new URI resource name attribute. The CICS management client interface uses these resource name attributes to specify CICS and CICSplex SM resources in URI requests.

Changes to sample programs

DFH\$WUUR and DFH\$WUTC are new sample resource definitions to help you set up the CICS management client interface.

New sample programs

- DFH\$WUUR is a sample URI map definition. The URI map uses transaction CWWU and calls program DFHWBA to analyze the CICS Web request. DFH\$WUUR includes the following attribute values:

Description	Sample System Management Interface URI map
Group	DFH\$WU
Host	*
Path	CICSSystemManagement/*
Port	No
Program	DFHWUIPG
Scheme	HTTP
Status	Enabled
TCP/IP service name	DFH\$WUTC
Transaction	CWWU
URI map	DFH\$WUUR
Usage	Server

Other values are left to default.

- DFH\$WUTC is a sample TCP/IP service definition. It includes the following attribute values:

Authentication level	No
----------------------	----

CICS transaction ID	CWXN
Description	Sample System Management Interface TCPIP service
Group	DFH\$WU
Host	Any
Port	1490
Protocol	HTTP
Queue backlog limit	10
SSL	No
Status	Open
TCP/IP service name	DFH\$WUTC
Timeout for socket close	No
User-replaceable module name	DFHWBAAX

Other values are left to default.

The new sample definitions are supplied in group DFH\$WU. You must install definitions like these before you can use CMCI.

Changes to problem determination

CICS provides new information to help you diagnose problems relating to the CICS management client interface.

New component codes

The following component code has been added to support the CICS management client interface:

Component code	Component keyword	Description
WU	WEBRESTMGR	Application domain: CMCI component

You can use the component code in the following ways:

- To select the level of standard and special tracing in each component:
 - In the CETR transaction.
 - In the **STNTRxx** and **SPCTRxx** system initialization parameters.
 - In the **INQUIRE TRACETYPE** and **SET TRACETYPE** system programming commands. If there is a component keyword, you can use it instead of the component code in these commands.
- To specify the areas of CICS storage that you want to be included in a formatted dump, and the amount of data that you want formatted
- To specify the trace entries that you want to be included in a formatted dump and in the output from the trace utility program

On output, CICS uses the component code to identify messages and trace entries.

New error messages

The CICS management client interface issues XML messages with message identifiers in the range DFHWU4001 to DFHWU5002 to indicate the status of CMCI requests. These messages are associated with HTTP response codes and are accompanied by short and full explanations to help you understand the cause of any failures

Chapter 26. New system programming commands for managing the CSD

You can now manage CICS resource definitions in the CSD using a set of new CICS system programming interface (SPI) commands. These commands give you a command-level alternative to the CEDA transaction and the DFHCSDUP batch utility for updating the CSD.

Like other SPI commands, you can use the CSD commands in programs written in any CICS-supported language. They are recognized by the command interpreter (CECI), the execution diagnostic facility (EDF), and the CICS translator. The CICS management client interface uses these commands when making requests to a CICS region that is not being managed by CICSplex SM.

Changes to the system programming interface

A set of new system commands help you manage the CSD.

New commands

CSD ADD

Add a group to a list on the CSD.

For details of the command, see “CSD ADD” on page 312.

CSD ALTER

Change the attributes of an existing resource definition on the CSD.

For details of the command, see “CSD ALTER” on page 314.

CSD APPEND

Append the groups in one list on the CSD to the end of another list.

For details of the command, see “CSD APPEND” on page 316.

CSD COPY

Copy a resource definition in a group, to a different group, or copy an entire group.

For details of the command, see “CSD COPY” on page 317.

CSD DEFINE

Create a new resource definition on the CSD.

For details of the command, see “CSD DEFINE” on page 320.

CSD DELETE

Delete a list, a group, or a single resource definition in a group, from the CSD.

For details of the command, see “CSD DELETE” on page 324.

CSD DISCONNECT

Disconnect the current task from the CSD.

For details of the command, see “CSD DISCONNECT” on page 326.

CSD ENDBRGROUP

End the current browse of the groups in the CSD, or of the groups in a list.

For details of the command, see “CSD ENDBRGROUP” on page 326.

CSD ENDBRLIST

End the current browse of the lists in the CSD.

For details of the command, see “CSD ENDBRLIST” on page 327.

CSD ENDBRRSRCE

End the current browse of the resources in a specified group.

For details of the command, see “CSD ENDBRRSRCE” on page 328.

CSD GETNEXTGROUP

Get the next group in a group browse.

For details of the command, see “CSD GETNEXTGROUP” on page 328.

CSD GETNEXTLIST

Get the next list in a list browse.

For details of the command, see “CSD GETNEXTLIST” on page 329.

CSD GETNEXTRSRCE

Get the details of next resource in a resource browse.

For details of the command, see “CSD GETNEXTRSRCE” on page 330.

CSD INQUIREGROUP

Inquire on a group in the CSD, or on a group in a specified list in the CSD.

For details of the command, see “CSD INQUIREGROUP” on page 333.

CSD INQUIRELIST

Inquire on a list in the CSD.

For details of the command, see “CSD INQUIRELIST” on page 334.

CSD INQUIRERSRCE

Inquire on the attributes of a resource in a specified group in the CSD.

For details of the command, see “CSD INQUIRERSRCE” on page 334.

CSD INSTALL

Install a list, a group, or a single resource definition in a group, in the CSD.

For details of the command, see “CSD INSTALL” on page 337.

CSD LOCK

Restrict update and delete access for a group or list to a single operator identifier.

For details of the command, see “CSD LOCK” on page 340.

CSD REMOVE

Remove a group from a list on the CSD.

For details of the command, see “CSD REMOVE” on page 341.

CSD RENAME

Rename a resource definition on the CSD.

For details of the command, see “CSD RENAME” on page 342.

CSD STARTBRGROUP

Start a browse of the groups in the CSD, or of the groups in a list.

For details of the command, see “CSD STARTBRGROUP” on page 344.

CSD STARTBRLIST

Start a browse of the lists in the CSD.

For details of the command, see “CSD STARTBRLIST” on page 346.

CSD STARTBRRSRCE

Start a browse of the resources in a specified group.

For details of the command, see “CSD STARTBRRSRCE” on page 347.

CSD UNLOCK

Remove the lock from a group or list of definitions.

For details of the command, see “CSD UNLOCK” on page 348.

CSD USERDEFINE

Create a new resource definition with user-specified default values on the CSD .

For details of the command, see “CSD USERDEFINE” on page 349.

Chapter 27. The Discovery Library Adapter for CICS

The Discovery Library Adapter (DLA) for CICS is a utility that collects runtime information about CICS and CICSplex SM resources. It generates XML files that can be used by a discovery library reader to track these resources and their interdependencies across your enterprise.

The CICS DLA uses CICSplex SM services to generate XML files that conform to the Discovery Library IdML XML schema and Common Data Model (CDM). The XML files called discovery library books can be loaded by several products including Tivoli Change and Configuration Management Database (CCMDB) and Tivoli Business Service Management (TBSM).

The CICS DLA operates against all the CICSplexes that are associated with a specified CMAS. The specified CMAS must be running at the current release level; however, the CICS regions discovered by the DLA can be at earlier release levels, as allowed by the cross-release support rules for CICSplex SM.

To use the CICS DLA you run EYUJXDD0 as a batch job on a z/OS system. Running the DLA creates IdML discovery library books in a PDS data set. The DLA also writes control files and debug information to this data set.

When you run the DLA, it can generate three types of discovery library books:

- One CTSPLEX discovery library book containing summary information about discovered CICSplexes
- One or more CICSplex discovery library books containing information about the CICS system topology in a CICSplex and also the CICSplex SM logical scopes and their associated resources.
- One or more CICSREGION discovery library books containing information about CICS resources associated with a CICS region

The DLA also generates FTP PUT and RENAME statements to facilitate the transfer of the discovery library books to the discovery library file store (DLFS) located on your local system or accessible through a network connection. You control whether the FTP transfer is carried out as part of the DLA run or is done later as an independent operation.

From the DLFS you can load the discovery library books into CCMDB or TBSM using FTP or another transfer mechanism.

You control the output of the DLA using input parameters. These parameters control the range and depth of the discovery and specify which discovery library books are created. The parameters also set general options about the operating environment, including the name of the CMAS under which the discovery takes place. You can supply these parameters in startup JCL or in an input file referred to by the parameter DD statement in the JCL.

You can run the DLA in either create mode or refresh mode:

- Create mode is the default. In this mode, the DLA produces a snapshot of your CICSplexes. The DLA produces discovery library books that might not contain information about all affected resources because some are not currently active.

- In refresh mode, the discovery library books contain everything known about the environment. The reader treats as obsolete anything that was previously loaded from a similar discovery library book and is not in this discovery library book. CCMDB and TBSM support refresh mode by deleting these obsolete resources. Refresh mode can be useful when there has been a significant configuration change but you should use it with caution because of the large amount of processing that might be involved.

DLA terminology

The discovery library adapter for CICS introduces new terms and acronyms.

discovery library book

An XML document that stores discovery information and their relationships. The discovery library book can be later consumed by other discovery library aware systems.

common data model (CDM)

The data model that defines the data classes, their attributes, and the relationships between them. CDM produces the IdML schemas.

discovery

The process of finding resources in an enterprise, including finding the new locations of monitored resources that were moved.

Discovery Library Adapter (DLA)

A program that copies data from a management software system, converts it to IdML, and stores it in books in the discovery library.

Identity Markup Language (IdML)

The XML format that is used to store data in the discovery library.

Open Process Automation Library (OPAL)

The IBM Tivoli OPAL library contains extensions, such as DLAs, for IBM Tivoli Service Management applications. All DLAs are required to place an entry in this library with a description of the DLA and a link to any other information that can be downloaded about it.

Running the CICS DLA

To use the CICS DLA, you specify the necessary DLA parameters with startup JCL file, run the DLA utility as a batch program, then transfer the discovery library books in the resulting data set to the discovery library file store for exploitation by a discovery library reader.

Before you begin

- Ensure that CICS and CICSplex SM are properly installed and configured. You must install CICSplex SM because the CICS DLA uses the CICSplex SM API.
- Ensure that you have the necessary update access to the partitioned data set used for the IdML output of the DLA.

About this task

This task assumes that you want to use FTP to transfer the DLA output to your DLFS. If you use a transfer mechanism other than FTP, you must ensure the discovery library books are saved in the DLFS using the correct naming conventions.

To run the DLA:

Procedure

1. Open the sample JCL file EYUJXDD0 in the SDFHSAMP library for editing. EYUJXDD0 contains startup JCL and sample input parameters for controlling the operation of the DLA.
2. Enter high-level qualifiers for CICS, CICSplex SM, and for the CICS DLA partitioned data set by overtyping the parameter values in CPSMHLQ=@th1q@.CPSM, CICS HLQ=@th1q@.CICS with your own values.
3. Specify the CMAS, your z/OS IP host name, and your organization name using the mandatory **SET CMAS**, **SET HOSTNAME**, and **SET ORGNAME** parameters respectively.
4. Ensure that the **SET FTP** parameter is set to YES, the default. With this parameter set to YES, if the operation runs successfully, the DLA automatically transfers the discovered data to the DLFS as part of the DLA operation. FTP PUT and RENAME statements are always generated as part of the DLA run regardless of the SET FTP parameter.
5. Specify the remaining **SET** parameters. If you want to accept the default values, you do not have to make any changes. If you want to make changes to any parameters, overtype the defaults with your own values. For example, if you want to run the DLA in refresh mode, specify SET REFRESH YES.
6. Specify the **BOOK TYPE** parameters. These three parameters control which of the three types of discovery library books are generated, and also which information the discovery library books contain.

You can include multiple instances of each of these parameters and use the optional *cplexname* or *cics_name* attributes to include or exclude the specific CICSplexes or CICS regions from the information gathering process. The default for each of these parameters is YES, which means that all discovery library books of the specified type are included in the discovery. If you want to select specific discovery library books, set the parameter to NO, to suppress the discovery of all discovery library books of that type, and then add one or more subsequent parameters specifying the discovery library books to be selected. For example, if you want to select only the CICSplex discovery library book named plex1 specify:

```
BOOK TYPE CICSplex NO
BOOK TYPE CICSplex.plex1 YES
```

If you specify BOOK TYPE CICSplex.plex1 NO, the DLA generates an discovery library book for each CICSplex associated with the CMAS except CICSplex plex1.

You can also specify partial CICSplex or CICS region names. For example, to select only CICS regions beginning with the letter a, specify:

```
BOOK TYPE CICSREGION NO
BOOK TYPE CICSREGION.a YES
```

If you omit *cplexname* or *cics_name* attributes, the parameter applies to all of the CICSplexes or CICS regions associated with the CMAS.

Using multiple instances of the same **BOOK TYPE** parameter can lead to consequences that are difficult to predict. The DLA processes these parameters in the order that they are included in the input file. It is therefore possible for one parameter to conflict with an earlier one.

7. Save and submit the EYUJXDD0 module to start the DLA operation. The DLA populates a partitioned data set with discovery library books containing the discovered data. The default name of the data set is

hlq.cmas.CICSTS.DLAnn.IDML, where *hlq* is the high-level qualifier you supply, *cmas* is the name of the CMAS specified in the **SET CMAS** parameter, and *nnn* is the CICS version number, for example 660. If the data set already exists, its contents are overwritten with the newly discovered data.

The DLA also creates FTP PUT and RENAME statements in to facilitate the transfer of the data.

If the DLA operation is successful, it generates a return code of zero, then continues and transfers the discovered data to the DLFS using the FTP statements in the data set member @FTPRUN.

If the DLA generates a nonzero return code, or if you have specified **SET FTP NO**, the DLA does not carry out the FTP part of the operation. You can run the EYUJXDD1 module at any time to carry out the FTP step independently. In this case, the DLA uses the FTP statements from the @FTPALLP or @FTPCHGP data set members to transfer the data.

Results

After transferring the DLA output, your DLFS contains the discovery library books required for validation and exploitation by your discovery library reader.

The CICS DLA package

The CICS DLA is shipped as a series of modules in the SEYUAUTH, SEYULOAD, SEYUSAMP, and SEYUPROC CICSplex SM libraries.

Table 3. DLA modules

Modules	Library	Description
EYU9XDDA	SEYULOAD	Includes the main program and any other EYU9XDDc programs it calls.
EYUJXDD0	SEYUSAMP	Batch JCL for calling procedure EYUJXDDP to run the DL and transfer the IdML members to the DLFS.
EYUJXDD1	SEYUSAMP	Batch JCL to transfer the IdML members to the DLFS as a standalone job.
EYUJXDDP	SEYUPROC	Procedural JCL for running the DLA.
EYUMCT1C	SEYUAUTH	Simplified Chinese message load modules.
EYUMCT2C		
EYUMCT1E	SEYUAUTH	English language message load modules.
EYUMCT2E		
EYUMCT1K	SEYUAUTH	Japanese Kanji message load modules.
EYUMCT2K		

DLA parameters

You control the output of the CICS DLA using parameters that you include in the SYSIN DD card of the EYUJXDD0 JCL file. The DLA SET parameters specify general options such as the organization name and debug options. DLA BOOK TYPE parameters control which discovery library books are written.

The DLA parameters are as follows:

BOOK TYPE CICSplex.cplexname {YES | NO}

Determines whether the DLA creates a CICSplex discovery library book and controls which CICSplexes to include in the discovery. Specify one of the following values:

- YES to include the selected CICSplexes in the discovery.
- NO to suppress the selected CICSplexes from the discovery.

The default is YES.

cplexname is an optional character string containing the name of a CICSplex. You can use partial names or a trailing asterisk (*) as a wildcard character. You can use *cplexname* with either the YES or NO option to restrict the discovery to specific CICSplexes. If you omit *cplexname*, all the CICSplexes associated with the CMAS are included in or excluded from the discovery.

You can use multiple BOOK TYPE CICSplex parameters. The DLA processes multiple parameters in the order in which you include them in the input file.

BOOK TYPE CICSREGION.cics_name{YES | NO}

Determines whether the DLA creates a CICSREGION discovery library book and controls which CICS regions to include in the discovery. Specify one of the following values:

- YES to include the selected CICS regions in the discovery.
- NO to suppress the selected CICS regions from the discovery.

The default is YES.

cics_name is an optional character string containing the name of a CICS region. You can use partial names or a trailing asterisk (*) as a wildcard character. You use *cics_name* with either the YES or NO options to restrict the discovery to specific CICS regions. If you omit *cics_name*, all of the CICS regions associated with the CMAS are included in or excluded from the discovery.

You can use multiple BOOK TYPE CICSREGION parameters. The DLA processes multiple parameters in the order in which you include them in the input file.

BOOK TYPE CTSPLEX {YES | NO}

Determines whether the DLA creates a CTSPLEX discovery library book containing a summary of the discovered CICSplexes. Specify one of the following values:

- YES to produce a CTSPLEX discovery library book with a summary of all CICSplexes discovered by the DLA
- NO to suppress the output discovery library book for the CTSPLEX CICSplex summary.

The default is YES. You can include no more than one instance of this parameter.

SET CMAS CMAS_name

The name of a CMAS. The CMAS must be running at the CICS TS 4.1 release level. The DLA discovers information about the CICSplexes connected directly to this CMAS, and about CICSplexes connected to other CMASs associated with this CMAS.

SET CONSOLE_MESSAGES {YES | NO}

Use this parameter only under the supervision of IBM support staff.

Specify one of the following values:

- YES to copy messages to the MVS console.

- NO to bypass copying messages to the MVS console.

The default is NO.

Console messages are always displayed in English because the console does not support the double-byte character sets necessary to display the alternative languages.

SET FEEDBACK {QUIET | VERBOSE}

FEEDBACK specifies how the DLA handles exception condition reporting if an error is returned from the CICSplex SM API. The options are as follows:

- QUIET: the DLA writes only progress messages and exception conditions. QUIET is the default FEEDBACK option.
- VERBOSE: in addition to the standard message reporting response and reason codes, the DLA writes any associated CICSplex SM feedback data.

All feedback information is sent to the SYSPRINT DD destination and the @DIALOG file.

SET FTP {YES | NO}

Specify one of the following values:

- YES to transfer discovered data to the discovery library file store (DLFS) as part of the DLA run.
- NO to suppress the transfer.

The FTP transfer takes place only if the DLA discovery operation completed successfully with a return code of zero. If you specify NO, or if the DLA run does not complete successfully, you can transfer the data to the DLFS as a separate operation by running the EYUJXDD1 module.

During operation the DLA generates FTP PUT and RENAME in the following PDSE data set members:

- @FTPALLP: contains FTP statements to PUT and RENAME all discovery library books.
- @FTPCHGP: contains FTP statements to PUT and RENAME all changed discovery library books.

The contents of either @FTPALLP or @FTPCHGP are copied to the @FTPRUN data set member depending on the settings of the SET IGNORE_CHECKSUMS and SET REFRESH parameters.

SET HOSTNAME *host_name*

Specify the z/OS IP host name.

SET IGNORE_CHECKSUMS {YES | NO}

Specify one of the following values:

- YES, to bypass checksum processing for the discovery library books.
- NO, to calculate checksums for the discovery library books.

The default is NO.

Checksum processing determines whether each discovery library book has changed since the last DLA run. The DLA calculates a checksum for each discovery library book that it creates. The @FTPCHGP member contains a list of the discovery library books that the DLA has changed. You can then transfer only the changed discovery library books to the DLFS.

The advantage of checksum processing is that it reduces the amount of data transferred to the DLFS. The disadvantage is that it increases the DLA run

time. Because the checksum processing increases DLA processing, you might want to bypass it by setting SET IGNORE_CHECKSUMS YES.

SET LIMIT {1 - 9999999}

An integer between 1 and 9999999 specifying a limit to the number of resources to be discovered. The default is 10000.

The DLA issues warning messages if the limit is exceeded. Processing continues, but a non-zero DLA return code is set so that the FTP step, to transfer the discovery library books to the DLFS, will not run.

SET MESSAGE_LANGUAGE { | CHN | ENU | JPN }

This parameter determines the language used for the messages written to the destination specified on the SYSOUT option of the SYSPRINT file. Specify CHN for simplified Chinese, ENU for English, and JPN for Japanese Kanji. The default language is English. The Chinese and Japanese options use double-byte character sets.

SET ORGNAME *organization_name*

A single word without imbedded blanks denoting the name of your organization. This parameter can contain alphanumeric and national characters, with XML escape characters being used for other characters (for example: < for < and > for >).

The z/OS SMFID and SYSPLEX names are typically unique in an enterprise. In some cases, for example in the case of a company acquisition, duplicates can exist: for example, two independent z/OS systems with SMFID called MVS1. In order to maintain separate instances with a CMDB, set different ORGNAME values.

The default is SET ORGNAME <default0rg>. This default is a special value that informs CCMDB that the CICS resources are related to the CCMDB defined default organization. Use this default setting unless your site has duplicate z/OS SMFIDs and SYSPLEXes that you want to have loaded into the same CCMDB.

SET REFRESH {YES | NO}

Specify one of the following values:

- YES to produce refresh-type discovery library books
- NO to produce create-type discovery library books.

The default is NO.

Specifying SET REFRESH YES causes the output XML files to be refresh-type discovery library books. This setting indicates the discovery library books contain everything known about the environment. The readers of the discovery library books (such as CCMDB and TBSM) delete data loaded from corresponding discovery library books provided by the CICS DLA. Specify SET REFRESH YES only after a major configuration change.

Specifying SET REFRESH NO causes the output XML files to be create-type discovery library books. This setting indicates that the generated discovery library books contain information only about resources that have changed since the last DLA run.

DLA output

Each time you run the CICS TS DLA, it creates or reuses a partitioned data set. The DLA writes the discovery library books, control files, and debug information to this data set. You can control which discovery library books the DLA generates by setting the appropriate input parameters.

The default name of the partitioned (PDS or PDSE) data set is: *hlq.cmas.CICSTS.DLA410.IDML*, where *hlq* is a high-level qualifier that you supply and *cmas* is the name of the CMAS as specified on the SET CMAS parameter.

The DLA writes, or rewrites the following data set members:

- IdML member for the CTSPLEX discovery library book. This member contains a summary of the CICSplexes discovered by the DLA.
- IdML members for each CICSplex discovery library book that you request. The member name is the CICSplex name.
- IdML members for each CICSREGION discovery library book that you request. The member name is assigned by the DLA. The @CICSNAM member translates these names to the CICS region name and CICSplex associated with the CICS region.
- @CHCKSUM containing an entry for each of the discovery library book members on the data set.
- The @DLALOG message log from the latest DLA run.
- @FTPALLP containing FTP PUT and RENAME statements for all discovery library book members from the latest DLA run.
- @FTPCHGP containing FTP PUT and RENAME statements for some or all discovery library book members depending on the DLA run.

If you set the **IGNORE_CHECKSUMS** and **SET REFRESH** parameters to NO, this member contains FTP PUT and RENAME statements for any changed discovery library book members from the previous DLA run that used the same data set. If there are no changes, this member is empty. In all other circumstances, this member contains FTP statements for all the discovery library book members from the latest DLA run, and it is therefore identical in content to @FTPALLP.

- @FTPRUN containing FTP statements or a **QUIT** statement depending on the following conditions:
 - If you have set the **SET FTP** parameter to NO, this member contains only the statement **QUIT**.
 - If you have set the **SET FTP** parameter to YES and the **SET REFRESH** parameter to YES, this member contains the FTP statements copied from @FTPALLP.
 - If you have set the **SET FTP** parameter to YES and the **SET REFRESH** parameter to NO, this member contains the FTP statements copied from @FTPCHGP.

The DLA uses the contents of @FTPRUN to make the FTP transfer if you set the **SET FTP** parameter to YES and the DLA runs successfully with a return code of zero. You can also run the FTP step independently. To do this you use the EYUJXDD1 module in which you specify either @FTPRUN, @FTPALLP or @FTPCHGP.

Security and the DLA

The CICS DLA uses CICSplex SM API services and is therefore subject to CICSplex SM security for accessing CICS TS data.

You need update access to the partitioned data set used for the IDML output when running the DLA. When using FTP to transfer the data set members to the discovery library file store, you need read access to the data set and both write access and rename ability to the discovery library file store.

DLA problem determination

The CICS DLA issues messages with identifiers in the range EYUXD1000 through EYUXD1031.

If a CICSplex API request returns an error response, messages identify the request made and the responses received.

You control the destination and level of detail returned for any errors using the `CONSOLE_MESSAGES` and `FEEDBACK` input parameters.

If a run of the CICS DLA stalls, perform the following steps before cancelling the job.

- Take an SDUMP to include the DLA address space, the CMAS to which the utility is connected, and the associated DMDS, DAT, TOP, COM, and QUE data spaces.
- Save the DLA job log, the CMAS job log, and the output partitioned data set.

Chapter 28. Improvements in CICS monitoring

New performance data metrics improves the performance reporting and analysis of Web and Web service applications. Use the new transaction resource class monitoring data for distributed program link requests to improve the work load management of DPL applications. Also, to help reduce the volume of monitoring output produced, the default now is to compress monitor records.

You can set the monitoring options that limit the number of distributed program link requests, files, and temporary storage queues for which CICS is to perform transaction resource monitoring. Changes to the DPLLIMIT, FILELIMIT, and TSQUEUELIMIT values are made dynamically to the running CICS system.

Changes to external interfaces

These external interfaces now support the improvements to CICS monitoring.

Changes to the system programming interface

The INQUIRE MONITOR and SET MONITOR commands have new and changed options to support the improvements to CICS monitoring.

Changed command, INQUIRE MONITOR

A new DPLLIMIT option performs transaction monitoring for distributed program links. Monitoring records will be compressed by default.

COMPRESSST (*cvda*)

Returns a CVDA value indicating whether data compression is active for the CICS SMF 110 monitoring records produced by the CICS monitoring facility. CVDA values are as follows:

COMPRESS

Data compression is being performed for the monitoring records. Data compression is the default.

NOCOMPRESS

Data compression is not being performed for the monitoring records.

DPLLIMIT (*data-area*)

Returns the maximum number of distributed program link requests for which CICS is to perform transaction resource monitoring.

Changed command, SET MONITOR

New transaction resource monitoring options apply to distributed program links, files, and temporary storage queues, with new RESP2 values. To help reduce the volume of output produced when monitoring is active, the monitor records will be compressed by default.

COMPRESSST (*cvda*)

Specifies whether you want data compression to be performed for the CICS SMF 110 monitoring records produced by the CICS monitoring facility. If you change the setting for the data compression option, the new setting applies to all monitoring records written from that point on, even if they are for a task being processed at the time the change is made. The new setting also applies

to any records that are in the buffer waiting to be written to SMF at the time the change is made. The change applies only until a CICS restart.

COMPRESS

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

NOCOMPRESS

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

DPLLIMIT (*data-value*)

Specifies the maximum number of distributed program link requests for which CICS is to perform transaction resource monitoring, as a halfword binary value. The value specified must be in the range 0 - 64.

FILELIMIT (*data-value*)

Specifies the maximum number of files for which CICS is to perform transaction resource monitoring, as a halfword binary value. The value specified must be in the range 0 - 64.

TSQUEUELIMIT (*data-value*)

Specifies the maximum number of temporary storage queues for which CICS is to perform transaction resource monitoring, as a halfword binary value. The value specified must be in the range 0 - 64.

- 12 The FILELIMIT value is out of range.
- 13 The DPLLIMIT value is out of range.
- 14 The TSQUEUELIMIT value is out of range.

Changes to CEMT

The **INQUIRE MONITOR** and **SET MONITOR** commands have new options. If monitoring is active, the default is for monitoring records to be compressed.

New and changed options on the INQUIRE MONITORING command

Data compression is the default when monitoring is active. New options, **COMPRESS** and **DPLLIMIT**, have been added to support transaction resource monitoring for distributed program links.

COMPRESSST

Displays whether data compression is performed for monitoring records. The values are as follows:

COMPRESS

Data compression is performed. The default is for monitoring records to be compressed.

NOCOMPRESS

Data compression is not performed.

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

DPLLIMIT (*value*)

Displays the maximum number of distributed program link requests for which transaction resource class monitoring data is being collected.

New and changed options on the SET MONITORING command

New options, COMPRESS, DPLLIMIT, FILELIMIT, and TSQUEUELIMIT, support transaction resource monitoring. Data compression is the default when monitoring is active.

COMPRESS

Data compression is to be performed for monitoring records. The default is for monitoring records to be compressed.

DPLLIMIT (value)

The maximum number of distributed program link requests for which CICS is to perform transaction resource monitoring. The value specified must be in the range 0 - 64.

FILELIMIT (value)

The maximum number of files for which CICS is to perform transaction resource monitoring. The value specified must be in the range 0 - 64.

TSQUEUELIMIT (value)

The maximum number of temporary storage queues for which CICS is to perform transaction resource monitoring. The value specified must be in the range 0 - 64.

Changes to CICSplex SM resource tables

The MONITOR resource table supports the new DPLLIMIT parameter.

Changes to CICSplex SM views and menus

The monitoring details view has new and changed fields.

Changed view, EYUSTARTMONITOR.DETAIL1

The EYUSTARTMONITOR.DETAIL1 view has a new field, DPLLIMIT, for the distributed program link limit. Also, you can now change the DPLLIMIT, FILELIMIT, and TSQLIMITNN values using the EYUSTARTMONITOR.DETAIL1 view.

For more information, see the *CICSplex System Manager Resource Tables Reference*.

Changes to supplied transactions

The CEMN transaction has new and changed options.

The CEMN transaction includes the new DPL resource limit and the CEMN transaction is split into a primary panel and a second options panel. Also, you can change the DPLLIMIT, FILELIMIT, and TSQUEUELIMIT values using the CEMN transaction.

Other transactions

Changes to CEMT commands are described in "Changes to CEMT" on page 196.

Changes to monitoring data

Some performance data fields in performance class groups DFHWEBB and DFHPROG are changed. The DFHMCT TYPE=INITIAL macro supports the new DPLLIMIT parameter and the COMPRESS parameter default is changed to YES. Also, the monitoring record has new DPL entry fields.

New entry fields in the monitor record

New DPL entry fields apply to the transaction resource data section of a monitoring record.

MNR_DPL_PROGRAM_NAME (TYPE-C, 8 BYTES)

The name of the program to which the following data fields refer.

MNR_DPL_SYSID (TYPE-C, 4 BYTES)

The name of the remote system to which this program was routed for the distributed program link.

MNR_DPL_LINK_REQS (TYPE-C, 4 BYTES)

The number of distributed program link requests issued by the user task for this program and sysid combination.

Changed performance class group, DFHWEBB

This group has nw performance data fields 390 and 392. For Web service provider applications, the new performance data fields provide the size of the SOAP request and response body lengths

New performance data fields 386, 387, and 388 assist in the analysis of SOAP faults encountered by web service applications.

Changed performance class group, DFHPROG

Existing performance data field 071 (PGMNAME) is changed. For Web service applications, the initial program name field will be updated with the target application program name.

Changed DFHMCT TYPE=INITIAL macro

The DFHMCT TYPE=INITIAL macro contains the new DPL resource limit, and the compression parameter default is COMPRESS=YES.

Changes to statistics

The DFHSTUP statistics utility program and the monitoring domain global statistics support the new DPL resource limit.

Changed statistics type, DFHSTUP

The DFHSTUP statistics utility program Interval and Summary reports support the new DPL resource limit parameter.

Changed monitoring domain global statistics

The monitoring domain global statistics supports the new DPL resource limit

DFHSTUP name	Field name	Description
--------------	------------	-------------

DPL Resource Limit	MNGDPLRL	Shows the maximum number of distributed program links for which transaction resource monitoring is being performed.
		<u>Reset characteristic:</u> not reset

Changes to sample programs

The DFH0STAT and DFH\$MOLS sample programs contains the new DPL resource limit.

Changed sample program, DFH0STAT

The DFH0STAT system status report shows the current setting of the DPL resource limit parameter.

Changed sample program, DFH\$MOLS

The DFH\$MOLS monitoring sample program formats and prints the new DPL program resource monitoring data. The **RESOURCE** control statement also supports the new DPL option.

RESOURCE

ALL, DPL, FILE, TSQUEUE

Use this option to enable selective printing of the resource type data in the transaction resource monitoring records.

Chapter 29. Improved CICSplex SM views

Over 70 CICSplex SM Web User Interface operations view sets have been redesigned to improve their usability and function. The improvements include the reordering of fields into more logical groupings and the addition of new detailed views to make common tasks easier and faster.

The redesigned views include the following improvements:

- Additional hyperlinks to speed up the navigation between often-used views
- New detailed views that display logical groupings of fields on a single screen
- Reordered fields and columns that make it easier to find associated attributes
- Additional filters to quickly display the resources and objects that you want

New techniques to make better use of screen space are introduced. For example, in the Dynamic storage areas (DSA) view (EYUSTARTCICSSTOR.DETAIL1), DSA information is now displayed in a matrix-like arrangement so that you can view more information on one screen.

These changes do not involve the addition or removal of any resource table attributes.

Changes to external interfaces

Some changed external interfaces now support the improved CICSplex SM views.

Changes to CICSplex SM views

Over 70 CICSplex SM Web User Interface view sets have been improved.

Changed view sets

The following view sets have changed:

View set	Navigation
EYUSTARTAIMODEL	CICS operations views → Terminal operations views → Auto install models
EYUSTARTCFDTPPOOL	CICS operations views → File operations views → Coupling facility data table pools
EYUSTARTCICSRGN	CICS operations views → CICS region operations views → CICS regions
EYUSTARTCICSSTOR	CICS operations views → CICS region operations views → CICS storage
EYUSTARTCMDT	CICS operations views → File operations views → Managed data tables
EYUSTARTCONNECT	CICS operations views → Connection operations views → ISC/MRO connections
EYUSTARTDB2CONN	CICS operations views → DB2, DBCTL and WebSphere MQ operations views → Connections
EYUSTARTDB2ENTRY	CICS operations views → DB2, DBCTL and WebSphere MQ operations views → Entries

View set	Navigation
EYUSTARTDB2TRAN	CICS operations views → DB2, DBCTL and WebSphere MQ operations views → Thread associated transactions
EYUSTARTDB2TRN	CICS operations views → DB2, DBCTL and WebSphere MQ operations views → Entry associated transactions
EYUSTARTDBCTLSS	CICS operations views → DB2, DBCTL and WebSphere MQ operations views → DBCTL subsystem
EYUSTARTDOCTEMP	CICS operations views → Document template operations views → Document template
EYUSTARTDOMSPOOL	CICS operations views → CICS region operations views → Domain subpool
EYUSTARTDSNAME	CICS operations views → File operations views → Physical data sets for files
EYUSTARTDSPGBL	CICS operations views → CICS region operations views → Global dispatcher information
EYUSTARTEJCOSE	CICS operations views → Enterprise Java component operations views → CorbaServers
EYUSTARTEJDJAR	CICS operations views → Enterprise Java component operations views → CICS-deployed JAR files
EYUSTARTENQMODEL	CICS operations views → Enqueue model operations views → Enqueue model
EYUSTARTENQUEUE	CICS operations views → CICS region operations views → Enqueue pools
EYUSTARTEVENT	Real Time Analysis (RTA) views → Outstanding events
EYUSTARTEVENTDTL	EYUSTARTEVENT → Detail
EYUSTARTEXITGLUE	CICS operations views → Exit operations views → Global user exits
EYUSTARTEXITTRUE	CICS operations views → Exit operations views → Task related user exits
EYUSTARTTEXTGLORD	CICS operations views → CICS region operations views → Global user exits
EYUSTARTXTRATDQ	CICS operations views → Transient data queue (TDQ) operations views → Extrapartition
EYUSTARTFEPICONN	CICS operations views → FEPI operations views → Connections
EYUSTARTFEPINODE	CICS operations views → FEPI operations views → Nodes
EYUSTARTFEPPOOL	CICS operations views → FEPI operations views → Pools
EYUSTARTFEPIPROP	CICS operations views → FEPI operations views → Property sets
EYUSTARTFEPITRGT	CICS operations views → FEPI operations views → Targets
EYUSTARTINDTDQ	CICS operations views → Transient data queue (TDQ) operations views → Indirect
EYUSTARTINTRATDQ	CICS operations views → Transient data queue (TDQ) operations views → Intrapartition
EYUSTARTJRNLMDL	CICS operations views → Journal operations views → Models
EYUSTARTJRNLNAME	CICS operations views → Journal operations views → Names
EYUSTARTJVMPOOL	CICS operations views → Enterprise Java component operations views → JVM pool

View set	Navigation
EYUSTARTLOADACT	CICS operations views → CICS region operations views → Loader by dynamic storage area
EYUSTARTLOADER	CICS operations views → CICS region operations views → Loader activity
EYUSTARTLOCFILE	CICS operations views → File operations views → Local files
EYUSTARTLOCTRAN	CICS operations views → Transaction operations views → Local or dynamic
EYUSTARTLSRPBUF	CICS operations views → File operations views → VSAM LSR pool buffers
EYUSTARTLSRPOOL	CICS operations views → File operations views → VSAM LSR pools
EYUSTARTMODENAME	CICS operations views → Connection operations views → LU 6.2 mode names
EYUSTARTMONITOR	EYUSTARTCICSRGN.DETAILED → Logging and journaling activity → Monitoring and statistics status
EYUSTARTPARTNER	CICS operations views → Connection operations views → Partners
EYUSTARTPROCTYP	CICS operations views → CICS Business Transaction Services (BTS) operations views → Process type
EYUSTARTPROGRAM	CICS operations views → Program operations views → Programs
EYUSTARTRECOVERY	EYUSTARTCICSRGN.DETAILED → Logging and journaling activity → Recovery manager
EYUSTARTREMFIL	CICS operations views → File operations views → Remote files
EYUSTARTREMTDQ	CICS operations views → Transient data queue (TDQ) operations views → Remote
EYUSTARTREMTTRAN	CICS operations views → Transaction operations views → Remote
EYUSTARTREQID	CICS operations views → Task operations views → Interval control requests
EYUSTARTRPLLIST	CICS operations views → Program operations views → Static DFHRPL data set names
EYUSTARTRQMODEL	CICS operations views → Transaction operations views → Request model
EYUSTARTSTREAMNM	CICS operations views → Journal operations views → MVS log stream names
EYUSTARTSYSDUMP	CICS operations views → CICS region operations views → System dump codes
EYUSTARTTASKESTG	CICS operations views → Task operations views → Storage element by task
EYUSTARTTCPIPGBL	CICS operations views → TCP/IP service operations views → Global TCP/IP statistics
EYUSTARTTCPIPS	CICS operations views → TCP/IP service operations views → TCP/IP service
EYUSTARTTDQGBL	CICS operations views → Transient data queue (TDQ) operations views → Global transient data queue attributes

View set	Navigation
EYUSTARTTERMNL	CICS operations views → Terminal operations views → Terminals
EYUSTARTTRANCLAS	CICS operations views → CICS region operations views → Transaction classes
EYUSTARTTRANDUMP	CICS operations views → CICS region operations views → Transaction dump codes
EYUSTARTTSKSPOLS	EYUSTARTTSKSPOLS.DETAILED → Task subpools
EYUSTARTTSKSPool	EYUSTARTCICSRGN.DETAILED → Task subpools
EYUSTARTTSMODEL	CICS operations views → Temporary storage queue (TSQ) operations views → Models
EYUSTARTTSPool	CICS operations views → Temporary storage queue (TSQ) operations views → Pools
EYUSTARTTSQGBL	CICS operations views → Temporary storage queue (TSQ) operations views → Global temporary storage statistics
EYUSTARTTSQNAME	CICS operations views → Temporary storage queue (TSQ) operations views → Temporary storage queues
EYUSTARTTSQSHR	CICS operations views → Temporary storage queue (TSQ) operations views → Shared queues
EYUSTARTUOW	CICS operations views → Unit of work (UOW) operations views → Unit of work
EYUSTARTUOWDSNF	CICS operations views → Unit of work (UOW) operations views → Unit of work shunted and holding data set locks
EYUSTARTUOWENQ	CICS operations views → Unit of work (UOW) operations views → Unit of work enqueue
EYUSTARTUOWLInK	CICS operations views → Unit of work (UOW) operations views → Unit of work link

Part 5. Architectural enhancements

CICS Transaction Server for z/OS, Version 4 Release 1 delivers a set of capabilities represented in the following themes:

- Compete
- Comply
- Control
- Architectural enhancements

The capabilities represented by the *Architectural enhancements* theme provide enhancements, optimizations, and improved performance for CICS workloads.

Chapter 30. Improvements to XML parsing in CICS

The performance of XML parsing in CICS has improved with the introduction of the IBM z/OS XML System Services (XMLSS) parser, which can be accessed directly from CICS. The XMLSS parser also allows XML parsing to be offloaded to a System z Application Assist Processor (zAAP), which can reduce the cost of transactions.

The XMLSS parser uses above-the-bar storage, leaving more below-the-bar storage available for user programs.

For more information about zAAP, see the introduction to *System z Application Assist Processor (zAAP)* on the IBM website: <http://www-03.ibm.com/systems/z/advantages/zaap/index.html>.

Changes to external interfaces

Some changed external interfaces support the improvements to XML parsing in the CICS pipeline.

Changes to monitoring data

New monitoring data provides information on how the XML System Services parser is performing when parsing XML documents.

Changed performance class group, DFHWEBB

Additional performance class data is added to the DFHWEBB group:

411 (TYPE-S, 'MLXSSCTM', 12 BYTES)

The CPU time taken to convert a document using the z/OS XML System Services parser. This field is a subset of the total CPU time as measured in the USRCPUT field (owner DFHTASK, field ID 008).

412 (TYPE-A, 'MLXSSTD', 4 BYTES)

The total length of the documents that were parsed using the z/OS XML System Services parser.

413 (TYPE-A, 'MLXMLTCT', 4 BYTES)

The number of EXEC CICS TRANSFORM commands issued by the user task.

Changes to problem determination

CICS provides new information to help you diagnose problems relating to XML parsing.

New component codes

This component code supports XML parsing:

Component code	Component keyword	Description
ML	None	Markup language domain

You can use the component code in the following ways:

- To select the level of standard and special tracing in each component:
 - In the CETR transaction.
 - In the **STNTRxx** and **SPCTRxx** system initialization parameters.
 - In the **INQUIRE TRACETYPE** and **SET TRACETYPE** system programming commands. If there is a component keyword, you can use the component keyword instead of the component code in these commands.
- To specify the areas of CICS storage that you want to include in a formatted dump and the amount of data that you want formatted.
- To specify the trace entries that you want to include in a formatted dump and in the output from the trace utility program.

On output, CICS uses the component code to identify messages and trace entries.

New messages

New error messages in the range DFHML0001 to DFHML9999 report problems with the markup language domain.

New abend codes

New abend codes AMLY and AMLZ are issued by CICS when an unrecoverable error occurs in the markup language domain.

New trace points

New trace points in the range ML 0100 to ML 0212 are issued by the markup language domain.

Chapter 31. New option for VTAM persistent sessions support

You can increase the number of CICS regions in an LPAR by removing VTAM persistent sessions support where it is not required, such as in a CICS region that is used only for development or testing. You can now disable VTAM persistent sessions support for a CICS region using the new value NOPS for the **PSTYPE** system initialization parameter.

If you have a large number of CICS regions in the same LPAR (around 500), with persistent sessions support available for all the regions, you might reach a z/OS limit on the maximum number of data spaces and be unable to add any more CICS regions. In this situation, when you attempt to start further CICS regions, you see messages IST967I and DFHSI1572, stating that the ALESERV ADD request has failed and the VTAM ACB cannot be opened. However, a region without persistent sessions support does not use a data space and so does not count towards the limit. To obtain a greater number of CICS regions in the LPAR:

1. Identify existing regions that can run without persistent sessions support.
2. Change the **PSTYPE** system initialization parameter for those regions to specify NOPS, and specify a zero value for the **PSDINT** system initialization parameter.
3. Cold start the regions to implement the change.

You can then start further CICS regions with or without persistent sessions support as appropriate, provided that you do not exceed the limit for the number of regions that do have persistent sessions support.

Changes to external interfaces

Some external interfaces have changed to support the new option to disable VTAM persistent sessions support.

Changes to system initialization parameters

The **PSTYPE** system initialization parameter has a new option, NOPS.

If you do not require persistent sessions support, specify NOPS. A CICS region that is used only for development or testing might not require this support. Removing persistent sessions support where it is not required reduces resource consumption, and can enable you to increase the number of CICS regions in an LPAR. If you specify NOPS, a zero value is required for the **PSDINT** system initialization parameter.

Changes to the system programming interface

The **INQUIRE VTAM** command can now inquire on the type of persistent sessions support for the CICS region. The **SET VTAM** command has changed to support the use of the system initialization parameter **PSTYPE=NOPS**.

The **INQUIRE VTAM** command has a new option PSTYPE.

PSTYPE(*cvda*)

Returns a CVDA value indicating the type of VTAM persistent sessions support for the CICS region. CVDA values are as follows:

SNPS Single-node persistent sessions. VTAM sessions can be recovered after a CICS failure and restart.

MNPS

Multinode persistent sessions. VTAM sessions can also be recovered after a VTAM or z/OS failure in a sysplex.

NOPS VTAM persistent sessions support is not used for this CICS region.

The **SET VTAM** command does not allow you to change PSDINTERVAL, PSDINTHRS, PSDINTMINS, or PSDINTSECS to a nonzero value when the system initialization parameter PSTYPE=NOPS is in effect.

Changes to CEMT (master terminal transaction) commands

The **INQUIRE VTAM** command can now inquire on the type of persistent sessions support for the CICS region.

The **INQUIRE VTAM** command has a new option PSTYPE.

PSTYPE

Displays the type of VTAM persistent sessions support for the CICS region. The values are as follows:

SNPS Single-node persistent sessions. VTAM sessions can be recovered after a CICS failure and restart.

MNPS

Multinode persistent sessions. VTAM sessions can also be recovered after a VTAM or z/OS failure in a sysplex.

NOPS VTAM persistent sessions support is not used for this CICS region.

Changes to CICSplex SM resource tables

The CICSRCN resource table has a new attribute PSTYPE, showing the type of persistent sessions support for the CICS region, as specified by the PSTYPE system initialization parameter.

Changes to CICSplex SM views and menus

The CICS regions view (**CICS operations views > CICS region operations views > CICS regions - CICSRCN**) now displays the type of persistent sessions support for the CICS region, shown as **CICS-VTAM Persistent sessions type**.

Changes to statistics

The DFH0STAT report **Autoinstall and VTAM Report** has new fields **Persistent Session Type** and **Persistent Session Interval**, showing the type of persistent sessions support for the CICS region and the persistent session delay interval.

Chapter 32. Wild branch diagnosis improvements

CICS TS 4.1 supports the Breaking Event Address Register (BEAR). This register captures the address of the instruction that generated the last successful branch before the error occurred in the program. This information helps when investigating wild branch problems in programs. The Breaking Event Address Register is held in the system diagnostic work area.

Wild branch diagnosis is to help identify where a program has taken a branch to a random address because of a computational error. When a program fails, you can use the address of the last successful instruction in the BEAR and the program location in the PSW (program status word) to help identify the sequence of events in the program. Using this information can speed up and improve the quality of problem diagnosis.

Changes to external interfaces

These external interfaces now support the improvements for wild branch diagnosis.

Changes to global user exits

The transaction abend control block (TACB) supports the improvement to the wild branch diagnosis. The TACB is passed to the XPCTA, XPCABND and XPCHAIR global user exits.

Global user exits, XPCTA, XPCABND, and XPCHAIR

The TACB has been extended with the breaking event address register information. The XPCTA, XPCABND, and XPCHAIR global user exits are passed a pointer to the TACB parameter. These exits only have to be reassembled if the new information is to be processed by the exit or the ABNDMSGT is not referenced by its address in field ABNDAMSG.

Changes to sample programs

The supplied program error program, DFHPEP, supports the improvements in wild branch diagnosis.

Changed sample program, DFHPEP

A new field has been added to the supplied program error program to support the improvements in wild branch diagnosis.

```
*           Breaking Event Address
*
PEP_COM_BEAR           DS      AD           Breaking Event Addr
*

```

Chapter 33. Extended z/Architecture MVS linkage support

CICS TS 4.1 supports the z/Architecture MVS linkage support for the access registers, all the floating point registers, and the 64-bit general registers. This improved linkage support provides execution context for application programs, that follow the linkage conventions, to ensure they can successfully pass control from one to the other while preserving the register contents and parameter data that they need to run successfully. System and transaction dumps display all of the registers.

However, if global user exits or task-related user exits modify the contents of the registers, the registers must be saved by the exit before they are altered and then restored to their original values before the exit passes control to CICS.

The new release sensitive XPI call, RELENSCALL is an alternative to the CALL XPI call. Using the RELENSCALL XPI call and assembling against the CICS TS 4.1 libraries, XPI calls execute successfully on all currently supported CICS releases. The release sensitive XPI call alternative applies to all XPI commands.

If a global user exit or task-related user exit is assembled using CICS libraries from a release earlier than CICS TS 4.1 and makes an XPI call on a CICS TS 4.1 system the exit will fail. An error message is issued and the transaction that called the exit might end. You must reassemble all global user exits and task-related user exits, against the CICS TS 4.1 libraries, if they contain *any* XPI calls.

Changes to external interfaces

These external interfaces now support the extended z/Architecture MVS linkage support

Changes to global user exits

New fields have been added to the system recovery program exit, XSRAB, for extended z/Architecture MVS linkage support. Similar changes have been made to the TACB which is passed to exits XPCTA, XPCABND and XPCHAIR.

Changed global user exit, XSRAB

New fields have been added to the system recovery program exit, XSRAB, to support the extended z/Architecture MVS linkage support.

SRP_ADDITIONAL_REG_INFO

An area containing additional register information.

SRP_ADDITIONAL_REGS_FLAG

1 byte containing flags:

SRP_CICS_GPR64_AVAIL

The 64-bit CICS GP registers are available.

SRP_SYSTEM_GPR64_AVAIL

The 64-bit system GP registers are available.

SRP_ADDITIONAL_FPR_AVAIL

Additional FP registers are available.

SRP_CICS_GP64_REGS

128-byte area containing the CICS 64-bit GP registers at the time of the abend.

SRP_SYSTEM_GP64_REGS

128-byte area containing the system 64-bit GP registers at the time of the abend.

SRP_FP_REGS

128-byte area containing all of the FP registers at the time of the abend.

SRP_FPC_REG

4-byte field containing the FPC register at the time of the abend.

Global user exits, XPCTA, XPCABND, and XPCHAIR

The TACB has been extended with additional GP and FP register information. The XPCTA, XPCABND, and XPCHAIR global user exits are passed a pointer to the TACB parameter. These exits only have to be reassembled if the new information is to be processed by the exit or the ABNDMSGT is not referenced by its address in field ABNDAMSG.

Changes to the exit programming interface (XPI)

Release sensitive XPI call, RELENSCALL, is an alternative to the CALL XPI parameter.

New function call, RELENSCALL

By replacing the CALL XPI parameter with the release sensitive XPI parameter, a XPI call assembled using the CICS TS 4.1 libraries will execute successfully on all currently supported CICS releases. The release sensitive XPI call alternative applies to all XPI commands.

Changes to sample programs

The supplied program error program, DFHPEP, supports the extended z/Architecture MVS linkage support.

Changed sample program, DFHPEP

New fields have been added to the supplied program error program to support the extended z/Architecture MVS linkage conventions.

```
*
*      Additional register information
*
PEP_COM_FLAG1          DS      0D      Force alignment
PEP_COM_FLAG1          DS      X       Flag byte
PEP_COM_GP64_REGS_AVAIL EQU     X'80'  64 bit register values
*                          available in
*                          PEP_COM_G64_REGISTERS
PEP_COM_ACCESS_REGS_AVAIL EQU     X'40'  64 bit register values
*                          available in
*                          PEP_COM_ACCESS_REGISTERS
PEP_COM_ORIGINAL_FPR_AVAIL EQU     X'20'  FPR 0, 2, 4 & 6 values
*                          available in
*                          PEP_COM_FP_REGISTERS
PEP_COM_ADDITIONAL_FPR_AVAIL EQU    X'10'  All FPR available in
*                          PEP_COM_FP_REGISTERS &
*                          FPCR in
*                          PEP_COM_FPC_REGISTER
```

	DS	CL7	Reserved
PEP_COM_GP64_REGISTERS	DS	CL128	64 bit GP registers
PEP_COM_FP_REGISTERS	DS	0CL132	FP registers
PEP_COM_FP_REGISTER0	DS	FD	FP register 0
PEP_COM_FP_REGISTER1	DS	FD	FP register 1
PEP_COM_FP_REGISTER2	DS	FD	FP register 2
PEP_COM_FP_REGISTER3	DS	FD	FP register 3
PEP_COM_FP_REGISTER4	DS	FD	FP register 4
PEP_COM_FP_REGISTER5	DS	FD	FP register 5
PEP_COM_FP_REGISTER6	DS	FD	FP register 6
PEP_COM_FP_REGISTER7	DS	FD	FP register 7
PEP_COM_FP_REGISTER8	DS	FD	FP register 8
PEP_COM_FP_REGISTER9	DS	FD	FP register 9
PEP_COM_FP_REGISTER10	DS	FD	FP register 10
PEP_COM_FP_REGISTER11	DS	FD	FP register 11
PEP_COM_FP_REGISTER12	DS	FD	FP register 12
PEP_COM_FP_REGISTER13	DS	FD	FP register 13
PEP_COM_FP_REGISTER14	DS	FD	FP register 14
PEP_COM_FP_REGISTER15	DS	FD	FP register 15
PEP_COM_FPC_REGISTER	DS	F	FPC register
PEP_COM_ACCESS_REGISTERS	DS	CL64	Access registers
*			

Changes to problem determination

New messages and abend codes are issued if global user exits or task-related exits which make XPI calls have not been reassembled against the CICS TS 4.1 libraries.

New messages

Three new messages:

DFHAP0702

DFHAP0703

DFHAP0708

New abend codes

Two new abend codes:

ASRJ

AKEJ

Chapter 34. New FLAGSET option on INQUIRE and SET TRACETYPE commands

The new FLAGSET option on the INQUIRE and SET TRACETYPE commands, with its CVDA, replaces the STANDARD and SPECIAL options. Existing programs that use this command will work unchanged. There is no functional change to the command because the CVDA values include STANDARD and SPECIAL.

Changed command, EXEC CICS INQUIRE TRACETYPE

A new FLAGSET option has replaced the SPECIAL and STANDARD options in the **EXEC CICS INQUIRE TRACETYPE** command and a new INVREQ condition has been added.

FLAGSET(*cvda*)

Indicates whether the standard or special flags for the specified component are to be returned. CVDA values are as follows:

SPECIAL

Indicates that CICS returns the trace levels for special tracing.

STANDARD

Indicates that CICS returns the trace levels for standard tracing.

INVREQ

RESP2 values:

- 1 An incorrect value was specified for FLAGSET.

Changed command, EXEC CICS SET TRACETYPE

A new FLAGSET option has replaced the SPECIAL and STANDARD options in the **EXEC CICS SET TRACETYPE** command and a new INVREQ condition has been added.

FLAGSET(*cvda*)

Indicates whether the standard or special flags, for the specified component, are to be set. CVDA values are as follows:

SPECIAL

Specifies that you want to set levels for special tracing, for the components listed.

STANDARD

Specifies that you want to set levels for standard tracing, for the components listed.

INVREQ

RESP2 values:

- 1 An incorrect value was specified for FLAGSET.

Chapter 35. New ENQUEUE_TYPE option on ENQUEUE and DEQUEUE XPI functions

You can use the new optional ENQUEUE_TYPE parameter, on the XPI ENQUEUE command, to enqueue on the same resource being enqueued on by EXEC CICS ENQ or vice versa. Applications can synchronize processes using EXEC CICS and EXEC XPI commands.

If the ENQUEUE_TYPE option is not specified, XPI is assumed. Existing exits will work unchanged.

Changed functions, ENQUEUE and DEQUEUE

A new ENQUEUE_TYPE option has been added to the ENQUEUE and DEQUEUE functions.

ENQUEUE_TYPE (XPI | EXECSTRN | EXECADDR)

Specifies the type of resource being enqueued on. The XPI option specifies the typical DFHNQEDX behavior. The resource pool used is exclusive to XPI and cannot be accessed by the CICS API. Use EXECSTRN or EXECADDR to indicate that ENQUEUE_NAME1 specifies the name or address respectively of an enqueue resource, located in the same namespace, as the one being used by EXEC CICS ENQ.

Chapter 36. New business application manager domain INQUIRE_ACTIVATION XPI function

You can use the new INQUIRE_ACTIVATION XPI function to obtain the activity name and process type for the business transaction activity of the current transaction.

New function, INQUIRE_ACTIVATION

The new INQUIRE_ACTIVATION function is provided on the DFHABRX macro call.

Chapter 37. The JVM server runtime environment

CICS includes a new runtime environment to perform Java processing. The runtime environment is called a *JVM server* and provides a mechanism for CICS to use the same JVM for multiple tasks concurrently.

The JVM server requires a Language Environment enclave and a pool of T8 TCBS. Each task uses a T8 TCB to run as a pthread in the JVM. The JVM server is represented in CICS by a JVMSERVER resource. The JVMSERVER resource defines the runtime options for the JVM, including the location of the JVM profile and the maximum number of threads that the JVM supports.

The JVM server is not available for use by Java applications. The JVM server is an architectural enhancement that is provided for CICS to perform system processing only, if required by future developments.

Changes to external interfaces

A number of external interfaces now support the JVM server runtime environment.

Changes to resource definitions

The new JVMSERVER resource defines the properties of the JVM and the Language Environment enclave.

For details of this resource, see JVMSERVER resource definitions.

Changes to the system programming interface

A number of new commands help you to start and manage the JVM server runtime environment.

Support for new resources

The following commands support the new JVMSERVER resource:

CREATE

DISCARD

All **CSD** commands

For information about the new resource, see “Changes to resource definitions.”

New commands

INQUIRE JVMSERVER

Retrieve information about the JVM server runtime environment.

For details of the command, see Appendix C, “New system programming commands,” on page 305.

SET JVMSERVER

Enable the JVM server runtime environment.

For details of the command, see Appendix C, “New system programming commands,” on page 305.

Changed commands

The **INQUIRE DISPATCHER** command has new options, **ACTTHRDTCBS** and **MAXTHRDTCBS** to manage the pool of T8 TCBs.

The **INQUIRE JVMPPOOL** command has a new option, **PROFILEDIR**, to return the directory on z/OS UNIX that contains the JVM profiles for CICS.

Changes to CEMT (master terminal transaction) commands

A number of new and changed CEMT commands help you to initialize and manage the JVM server runtime environment.

Support for new resources

The **DISCARD** command supports the new **JVMSERVER** resource.

For information about the new resource, see “Changes to resource definitions” on page 223

New commands

INQUIRE JVMSERVER

Retrieve the status of the JVM server.

For details of the command, see Appendix D, “New CEMT commands,” on page 395.

SET JVMSERVER

Enable the JVM server.

For details of the command, see Appendix D, “New CEMT commands,” on page 395.

Changed commands

The **INQUIRE DISPATCHER** command has two new options:

ACTTHRDTCBS(*value*)

Displays the number of T8 mode open TCBs that are currently active; that is, allocated to a user task.

MAXTHRDTCBS(*value*)

Displays the maximum number of T8 mode open TCBs that CICS can attach and maintain for all enabled JVM servers. CICS calculates the maximum number by adding one TCB for each JVM server to the value of the **THREADLIMIT** attribute from all the enabled **JVMSERVER** resources. The value can be in the range 1 - 1024.

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

The **INQUIRE JVMPPOOL** command has a new option:

PROFILEDIR(*directory*)

Displays the directory in z/OS UNIX that contains the JVM profiles for CICS. This value is taken from the **JVMPROFILEDIR** system initialization parameter.

Changes to CICSplex SM resource tables

There are new and changed resource tables to support JVM servers in CICS.

New tables

The following new resource tables support the JVMSERVER resource:

CRESJVMS

A topology table for installed JVMSERVER resources.

JVMSVDEF

A BAS resource definition table for JVMSERVER resources. This table includes all of the attributes for defining a JVMSERVER resource definition.

JVMSESRV

An operations table for installed JVMSERVER resources.

JMSINGRP

A resource table for BAS JVMSVDEF resource objects that are contained in BAS resource groups.

Changed tables

The BAS resource description table (RESDESC) has new attributes to support JVMSVDEF resource objects:

Attribute	Description
JMSDEFRG	Resource group for JVMSERVER definitions
JMSDEFTS	Target group for JVMSERVER definitions
JMSDEFRS	Related scope for JVMSERVER definitions

The TASK and HTASK resource tables have new attributes for monitoring JVMSERVER resources:

Attribute	Description
TMRT8CPU	User task T8 mode CPU time
TMRTDLY	Maximum thread TCB delay time
TMRJSTWT	JVM server thread wait time

The following tables have also changed:

- The CICSRRGN and MONITOR resource tables have new attributes to support JVMSERVER statistics reporting.
- The CICSRRGN resource table also has a new attribute to show the number of T8 TCBs in a CICS region.
- The JVMPOOL resource table has a new attribute to show the z/OS UNIX directory where the JVM profiles are located.

Changes to CICSplex SM views and menus

There are new view sets and changed views to support JVM servers.

Changed menus

The following menus now support the JVMSVDEF resource object:

- The BAS CICS resource definitions menu, EYUSTARTADMRES, includes a link to the new JVMSVDEF definitions view set.
- The Enterprise Java operations menu, EYUSTARTENTJAVA, includes a link to the new JVMSERV view set.

New views

JVMSERVER definitions - EYUSTARTJVMSVDEF

To access this view set from the main menu, click **Administration views > Basic CICS resource administration views > Resource definitions > JVM server definitions**.

The views in this view set show the attributes of JVMSERVER resource definitions.

JVMSERVER resources - EYUSTARTJVMSEV

To access this view set from the main menu of the Web User Interface, click **CICS operations > Enterprise Java component operations views > JVM servers**.

The views in this view set show the installed JVMSERVER resources in the selected scope.

The detailed view of a JVMSERVER resource shows information such as the status of the resource, the maximum and current number of threads, and the name of the JVM profile.

Changed views

The following views now support the JVMSVDEF resource object:

- The resource description, EYUSTARTRESDESC
- Resource selected by resource descriptions, EYUSTARTRDSCPROC
- Mapping of BAS relationships, EYUSTARTMAPBAS

The following views now support installed JVMSERVER resources:

- The CICS regions view now reports JVMSERVER statistics and the number of T8 TCBS. Navigate to this view by clicking **CICS operations views > CICS region operations views > CICS regions**
- The Monitor status view now reports JVMSERVER statistics. From the CICS regions view, click **EYUSTARTCICSRGN.DETAILED > Logging and journaling activity > Monitor status**.
- The Active tasks view and Completed tasks view now include monitoring data for JVMSERVER resources. Navigate to these views by clicking **CICS operations views > Task operations views**.
- The JVM pool detailed view now includes the name of the z/OS UNIX directory where the JVM profiles are located. Navigate to this view by clicking **CICS operations views > Enterprise Java component operations views > JVM pool** and selecting the CICS system.

Changes to global user exits

The standard list of parameters that are passed to all global user exit programs now support the JVM server.

The **UEPGCICS** parameter in DFHUEPAR now has TCB indicators for the new TP and T8 mode open TCBS:

Symbolic value	2-byte code	Description
UEPTTP	TP	The TP open TCB owns the Language Environment enclave and the pool of T8 TCBs for a JVM server.
UEPTT8	T8	A T8 TCB is used by CICS to attach multiple tasks to the JVM server concurrently.

Support for new resources in global user exit, XRSINDI

The resource management install and discard exit XRSINDI supports the new JVMSERVER resource. For a JVMSERVER resource, the **UEPIDTYP** parameter returns a value of UEIDJSRV.

For information about the new resource, see “Changes to resource definitions” on page 223

Changes to monitoring data

Some monitoring fields are changed and new fields are added to return information about the TP and T8 mode TCBs.

Changed performance class group, DFHTASK

The following fields have changed to include the new TCB modes:

007 (TYPE-S, 'USRDISPT', 12 BYTES)

Total elapsed time during which the user task was dispatched on each CICS TCB under which the task ran. The TCB modes managed by the CICS dispatcher are: QR, RO, CO, FO, SZ, RP, SL, SP, SO, EP, J8, J9, L8, L9, S8, TP, T8, X8, X9, JM, and D2. Be aware that, for each CICS release, new TCB modes might be added to this list, or obsolete TCB modes might be removed.

008 (TYPE-S, 'USRCPUT', 12 BYTES)

Processor time for which the user task was dispatched on each CICS TCB under which the task ran. The TCB modes managed by the CICS dispatcher are: QR, RO, CO, FO, SZ, RP, SL, SP, SO, EP, J8, J9, L8, L9, S8, TP, T8, X8, X9, JM, and D2. Be aware that, for each CICS release, new TCB modes might be added to this list, or obsolete TCB modes might be removed.

252 (TYPE-A, 'DSTCBHM', 4 BYTES)

The peak number of CICS open TCBs (in TCB modes J8, J9, L8, L9, S8, T8, X8, and X9) that have been concurrently allocated to the user task.

262 (TYPE-S, 'KY8DISPT', 12 BYTES)

The total elapsed time during which the user task was dispatched by the CICS dispatcher on a CICS Key 8 mode TCB:

- A J8 mode TCB is allocated when a transaction calls a Java program that is defined with EXECKEY=CICS, indicating that the program requires a JVM in CICS key. A J8 mode TCB can also be allocated if the Java program is defined with EXECKEY=USER, but the storage protection facility is inactive. The TCB remains associated with the task until the Java program completes.
- An L8 mode TCB is allocated when a transaction calls an OPENAPI application program defined with EXECKEY=CICS or a task-related user exit program that has been enabled with the OPENAPI option. The TCB remains associated with the task until the transaction is detached.

- An S8 mode TCB is allocated when a transaction is using the secure sockets layer (SSL) during client certificate negotiation. The S8 mode TCB remains associated with the same task for the life of the SSL request.
- A T8 mode TCB is allocated when a transaction is using a JVM server to perform multithreaded processing. When a thread is allocated a T8 mode TCB, that same TCB remains associated with the thread until the processing completes.
- An X8 mode TCB is allocated when a transaction calls a C or C++ program that was compiled with the XPLINK option and that is defined with EXECKEY=CICS. The TCB remains associated with the task until the program ends.

This field is a component of the task dispatch time field, **USRDISPT** (field ID 007 in group DFHTASK).

263 (TYPE-S, 'KY8CPUT', 12 BYTES)

The processor time during which the user task was dispatched by the CICS dispatcher on a CICS Key 8 mode TCB. The usage of the CICS Key 8 mode TCBs is shown in the description for field **KY8DISPT** (field ID 262 in group DFHTASK). This field is a component of the task CPU time field, **USRCPUT** (field ID 008 in group DFHTASK).

The following new fields are added to the DFHTASK class group:

283 (TYPE-S, 'MAXTTDLY', 12 BYTES)

The elapsed time for which the user task waited to obtain a T8 TCB, because the CICS system reached the limit of available threads. The T8 mode open TCBs are used by a JVM server to perform multithreaded processing. Each T8 TCB runs under one thread. The thread limit is 1024 for each CICS region and each JVM server in a CICS region can have up to 256 threads. This field is a component of the task suspend time field, **SUSPTIME** (group name: DFHTASK, field ID: 014).

400 (TYPE-S, 'T8CPUT', 12 BYTES)

The processor time during which the user task was dispatched by the CICS dispatcher domain on a CICS T8 mode TCB. T8 mode TCBs are used by a JVM server to perform multithreaded processing. When a thread is allocated a T8 mode TCB, that same TCB remains associated with the thread until the processing completes. This field is a component of the total task CPU time field, **USRCPUT** (field ID 008 in group DFHTASK), and the task key 8 CPU time field, **KY8CPUT** (field ID 263 in group DFHTASK).

401 (TYPE-S, 'JVMTHDWT', 12 BYTES)

The elapsed time that the user task waited to obtain a JVM server thread because the CICS system had reached the thread limit for a JVM server in the CICS region. This field is a component of the task suspend time field, **SUSPTIME** (group name: DFHTASK, field ID: 014).

Changes to statistics

CICS now collects statistics on the T8 mode open TCBs that are used by the JVM server to process tasks concurrently.

Changed statistics type, dispatcher domain

A new option of JVMSERVER on the **PERFORM STATISTICS RECORD** and **CEMT PERFORM STATISTICS** commands records statistics for JVM servers.

A new RESTYPE value of JVMSERVER for the **EXEC CICS EXTRACT STATISTICS** command retrieves statistics relating to JVM servers.

The DSECT that maps the JVM server statistics is DFHSJSDS.

The statistics utility programs DFHSTUP and DFH0STAT now support JVM servers:

- DFHSTUP includes the new statistics for JVM servers.
- The sample extract program DFH0STAT formats the new statistics. The JVM pool DFH0STAT report has a new field to return the JVM profile directory.
- The dispatcher statistics now report on the TP mode open TCB and the pool of T8 mode open TCBs.

Changes to sample programs

New samples are provided to help you start up the JVM server runtime environment.

New samples

New sample program, DFHJVMAX

DFHJVMAX is a new JVM profile file that specifies the options for initializing the JVM server. The JVMSERVER resource defines the name of the JVM profile. Its location is determined by the **JVMPROFILEDIR** system initialization parameter. The new CICS options that you can use only in this JVM profile are as follows:

- SETUP_CLASSES
- TERMINATION_CLASSES
- THREAD_TIMEOUT

All other CICS options, with the exception of the **WORK_DIR** option, are ignored. CICS passes the JVM options and JVM system properties directly to the JVM for processing. The environment variables are set on the Language Environment enclave by a runtime options program.

New sample program, DFHAXRO

DFHAXRO is a new sample program that provides default values for configuring the Language Environment enclave of a JVM server. Modify and recompile this program to change the Language Environment enclave for a JVM server.

The JVMSERVER resource defines the name of the program that controls the options for the Language Environment enclave. Each JVM server can use a different version of the runtime options if required. The program must be in the *hlq.SDFHLOAD* library.

To modify the runtime options, you must write your own version of the DFHAXRO program in assembler language and it must not be translated with the CICS translator. You can use the source of DFHAXRO as the basis for your own program. The source is in the *hlq.SDFHSAMP* library. The runtime options must not exceed 255 bytes in length. If CICS detects that

the options exceed 255 bytes, the Language Environment enclave initializes using the default values from DFHAXRO.

The program is called during the CEEPIPI preinitialization phase of each Language Environment enclave that is created for a JVM server.

Changes to problem determination

The dump and trace formatter now include the new TP and T8 mode open TCBs.

New messages

New DFHAP messages in the range 1600 - 1603 relate to the initialization and operation of a JVM server. New DFHSJ messages in the range 0910 - 0917 relate to the JVMSERVER resource.

New abend code

The new abend code ASJO is issued when the CSJR transaction fails to initialize one or more JVM servers when CICS starts up.

New trace points

New trace points, in the range AP 193A - AP 1977 and SJ 0A01 - SJ 0C0C, trace entries and exits for the JVM server.

Changes to security

Resource and command security apply to JVMSERVER resource definitions if RESSEC(YES) and CMDSEC(YES) are enabled in the CICS region.

Transaction security categories

The CJSR transaction, or CICS JVM server resolution transaction, resolves JVMSERVER resources after the group list is installed when a CICS region restarts. The CJSR transaction is a category 1 transaction that is implemented by program DFHCJSR. The transaction and the program are in the CSD group, DFHSTAND, which is included in DFHLIST.

Resource security

Resource security for JVMSERVER 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. You must apply a prefix of JVMSERVER to the name of the JVMSERVER resource definition.

Command security

Command security for JVMSERVER resources uses the JVMSERVER resource in the CCICSCMD class or the VCICSCMD grouping class.

Part 6. Discontinued functions

Some functions which were supported in CICS Transaction Server for z/OS, Version 3 Release 2 have been discontinued, or reduced in scope, in Version 4 Release 1.

Chapter 38. Removal of support for Version 1.4.2 and Version 1.5.0 of IBM SDK for z/OS, Java Technology Edition

In CICS TS for z/OS, Version 4.1, support for Version 1.4.2 and Version 1.5.0 is withdrawn, and you must upgrade to Version 6 of the SDK.

When you upgrade to Java 6 from Java 1.4.2 you can benefit from the new Java language features, and also from the improvements to runtime technology in the IBM SDK for z/OS, V6. These improvements include better garbage collection and simpler class sharing.

Changes to installation procedures

The default for the JAVADIR parameter in the DFHISTAR job has changed to java/J6.0, which is the default installation location for Version 6 of the IBM SDK for z/OS, Java Technology Edition.

In CICS TS for z/OS, Version 3.2 and earlier releases, the default for the JAVADIR parameter was java142/J1.4, the default install location for Version 1.4.2 of the SDK, which is no longer supported.

Changes to external interfaces

These external interfaces have changed because of the removal of support for Version 1.4.2 and Version 5 of the IBM SDK for z/OS, Java Technology Edition.

Changes to system initialization parameters

The behavior of the JVMCCSIZE and JVMCCSTART system initialization parameters has changed, and the JVMCCPROFILE system initialization parameter is now obsolete.

Changed system initialization parameters

The behavior of the JVMCCSIZE and JVMCCSTART parameters has changed.

JVMCCSIZE

JVMCCSIZE, which specifies the size of the shared class cache, typically applies only on an initial or cold start of CICS. If you specify it as an override, it has no effect on a warm or emergency start.

JVMCCSTART

The startup behavior specified for the shared class cache typically applies only on an initial or cold start of CICS. If you specify it as an override, it has no effect on a warm or emergency start.

Because CICS supports a single version of the IBM SDK for z/OS, the JVMCCSTART=YES system initialization parameter starts the shared class cache at CICS initialization on an initial or cold start, as it did in releases prior to CICS TS for z/OS, Version 3.2. In this situation, an initialization program (PLTPI program) is not required to start the shared class cache.

Discontinued system initialization parameter, JVMCCPROFILE

JVMCCPROFILE, which specified the JVM profile to be used for the master JVM that initialized a Version 1.4.2 shared class cache, is now obsolete. If you specify this system initialization parameter, a warning error message (MNOTE 4) is produced.

Changes to options in JVM profiles and the optional JVM properties files

A number of changes apply to the options in JVM profiles as a result of the removal of support for Version 1.4.2 and Version 5 of the IBM SDK for z/OS, Java Technology Edition. No properties files are supplied with CICS TS 4.1 as they are optional.

As with Version 5, when you use Version 6 of the IBM SDK for z/OS, Java Technology Edition for Java support, the shared class cache has no master JVM. JVMs that use the shared class cache do not inherit values for JVM options from a master JVM, and you do not need to place classes on the library path and shareable application class path in a JVM profile or JVM properties file for a master JVM. All the JVM options and classes are specified in the JVM profiles for the individual JVMs.

With the Version 6 shared class cache, the JVM options for a JVM that uses the shared class cache are the same as for a JVM that does not. Except for the CLASSCACHE option, the JVM profiles are set up in the same way, and the same class paths are used. The CICS-supplied sample JVM profiles CICS TS 4.1 reflect these changes.

Obsolete options caused by the removal of support for Java 1.4.2

Do not use these options in JVM profiles and JVM properties files, because they applied only when Version 1.4.2 of the IBM SDK for z/OS, Java Technology Edition was used for Java support.

CLASSCACHE_MSGLOG

Specified the file for messages from the master JVM. If you include this option in a JVM profile, CICS ignores it.

-Dibm.jvm.shareable.application.class.path

Specified the shareable application class path. This option was in the JVM properties file for the JVM. You must place all application classes on the standard class path, using the CLASSPATH_SUFFIX option in the JVM profile. If you include the **-Dibm.jvm.shareable.application.class.path** option in a JVM properties file, CICS accepts it and places the classes on the standard class path.

-Xinitsh

Specified the initial size for the system heap. If you include this option in a JVM profile, CICS passes the -Xinitsh option to the JVM, which ignores the option.

Options that now apply to JVMs using the Version 6 shared class cache

The options that were ignored for a JVM that used the Version 1.4.2 shared class cache (called a worker JVM) now apply for a JVM that uses the Version 6 shared class cache.

When you upgrade from Version 1.4.2 of the IBM SDK for z/OS, Java Technology Edition, if you have existing JVM profiles for worker JVMs that you want to continue to use, you must make changes to these profiles to include the options listed here. The sample JVM profile DFHJVMPC, for a JVM that uses the shared class cache, is not supplied with CICS TS for z/OS, Version 4.1.

Required options

The following options are required in the profile for a JVM that uses the Version 6 shared class cache, and for a JVM that does not use the shared class cache:

CICS_HOME

Specifies the path for the home directory for CICS files on z/OS UNIX.

JAVA_HOME

Specifies the installation location for the IBM SDK for z/OS.

For a JVM that used the Version 1.4.2 shared class cache, these options were specified in the master JVM profile. However, note that the values that were specified for these options in a JVM profile set up for Version 3.2 of CICS Transaction Server and Version 1.4.2 of the IBM SDK for z/OS, Java Technology Edition are not correct for your upgraded JVM profile. You must replace them with the paths for the CICS TS for z/OS, Version 4.1 home directory and the IBM SDK for z/OS, Java Technology Edition, Version 6 installation location. The defaults for these paths are in the CICS TS for z/OS, Version 4.1 sample JVM profiles.

Other options

The following options were ignored in the profile for a JVM that used the Version 1.4.2 shared class cache, but they can be included for a JVM that uses the Version 6 shared class cache:

LIBPATH_PREFIX and LIBPATH_SUFFIX

Specify additional items on the library path that do not already appear on the base library path. If your Version 1.4.2 master JVM profile includes these options, copy the options and their values into each of the JVM profiles for JVMs that use the shared class cache.

REUSE

Specifies whether the JVM is reusable. You can either add this option to your JVM profiles for JVMs that use the shared class cache, or allow it to default to REUSE=YES. JVMs that used the Version 1.4.2 shared class cache could not be single-use JVMs (REUSE=NO), but JVMs that use the Version 6 shared class cache can be single-use JVMs if required.

-Xdebug

Enables debugging support in the JVM. Do not add this option to the JVM profiles for JVMs that use the shared class cache unless you need to enable this support. JVMs that used the Version 1.4.2 shared class cache could not run with debugging support, but JVMs that use the Version 6 shared class cache can run with debugging support if required.

Changes to CICS-supplied sample JVM profiles

The sample profiles for the master JVM profile DFHJMCC, the worker JVM profile DFHJVMPC, and the single-use JVM profile DFHJMPS are not provided because they are obsolete.

The associated master JVM properties file dfhjmcc.props is also not supplied because it is obsolete.

Changes to the system programming interface

Changed options on the INQUIRE CLASSCACHE, PERFORM CLASSCACHE, and SET CLASSCACHE commands.

AUTOSTARTST option on the PERFORM and SET CLASSCACHE commands

This option has no effect when the shared class cache persists across warm and emergency starts. However, in circumstances where the shared class cache does not persist across a warm or emergency start (such as an IPL of z/OS), the changed setting specified by the AUTOSTARTST option is used, unless the JVMCCSTART system initialization parameter is specified as an override.

PROFILE option on the INQUIRE, SET, and PERFORM CLASSCACHE commands

This option is obsolete. It specified the JVM profile for the master JVM, which is no longer required. For the INQUIRE CLASSCACHE command, CICS returns blanks for this option. For the PERFORM CLASSCACHE command, the option is ignored.

REUSEST option on the INQUIRE CLASSCACHE command

This option refers only to the reuse status of the JVMs that use the shared class cache.

Changes to CEMT (master terminal transaction) commands

There are changes to the INQUIRE CLASSCACHE, PERFORM CLASSCACHE, and SET CLASSCACHE commands.

Autostartst option on the PERFORM CLASSCACHE and SET CLASSCACHE commands

This option has no effect when the shared class cache persists across warm and emergency starts. However, in circumstances where the shared class cache does not persist across a warm or emergency start (such as an IPL of z/OS), the changed setting specified by the **Autostartst** option is used, unless the JVMCCSTART system initialization parameter is specified as an override.

Profile option on the INQUIRE CLASSCACHE, SET CLASSCACHE, and PERFORM CLASSCACHE commands

This option is obsolete because it specified the JVM profile for the master JVM, which is no longer required.

Reusest option on the INQUIRE CLASSCACHE command

This option refers only to the reuse status of the JVMs that use the shared class cache, because the Version 6 shared class cache has no master JVM.

Changes to CICSplex SM resource tables

In the CLCACHE resource table, the PROFILE field does not apply for CICS TS for z/OS, Version 4.1 regions.

Changes to CICSplex SM views and menus

In the view **JVM class cache - CLCACHE**, the **PROFILE** field does not apply for CICS TS for z/OS, Version 4.1 regions.

Changes to statistics

The **Class Cache Profile** field, in the **DFH0STAT JVM Pool and Class Cache Report**, which reported the name of the JVM profile for the master JVM is obsolete and has been removed.

Changes to security

The Category 1 transaction **CJMJ**, which was used to start the master JVM for the Version 1.4.2 shared class cache, is no longer used because the Version 6 shared class cache does not have a master JVM. The CICS region user ID no longer requires access to this transaction.

Chapter 39. Removal of support for the DFHCSDUP MIGRATE command

Support for the DFHCSDUP MIGRATE command is withdrawn in CICS TS for z/OS, Version 4.1.

In previous versions of CICS, the DFHCSDUP MIGRATE command migrated the eligible DFHDCT, DFHRCT, DFHTCT, and DFHTST macro resource definitions to the CICS system definition data set (CSD).

If you use any of these tables, you must migrate them to the CSD before you upgrade to CICS TS for z/OS, Version 4.1. To do so, you can use the DFHCSDUP MIGRATE command on any supported release up to CICS TS for z/OS, Version 3.2.

Chapter 40. Removal of support for the CICSplex SM WLMLOADCOUNT and WLMLOADTHRSH EYUPARM values

Support for the WLMLOADCOUNT and WLMLOADTHRSH EYUPARM values is discontinued in CICS TS for z/OS, Version 4.1.

If you use the WLMLOADCOUNT or WLMLOADTHRSH EYUPARM values, you must now specify them using the **Task load health threshold** and **Task load queue mode** attributes in the CSYSDEF resource table.

Note: If you are using WLMLOADTHRSH EYUPARM, you must now specify the WLMLOADTHRSH EYUPARM as a **Task load health threshold** attribute in your target regions. This specification is a change from the discontinued EYUPARM which you previously specified in the routing regions.

Part 7. 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 41. Threadsafe API and SPI commands

All new application programming interface (API) and many new system programming interface (SPI) commands in CICS Transaction Server for z/OS, Version 4 Release 1 are threadsafe. No existing commands have been made threadsafe in this release.

New API commands that are threadsafe

- "INVOKE SERVICE" on page 262
- "SIGNAL EVENT" on page 266
- "TRANSFORM DATATOXML" on page 267
- "TRANSFORM XMLTODATA" on page 270
- "WEB ENDBROWSE QUERYPARM" on page 274
- "WEB READ QUERYPARM" on page 274
- "WEB READNEXT QUERYPARM" on page 276
- "WEB STARTBROWSE QUERYPARM" on page 278
- "WSACONTEXT BUILD" on page 279
- "WSAEPR CREATE" on page 284
- "WSACONTEXT DELETE" on page 287
- "WSACONTEXT GET" on page 287

New SPI commands that are threadsafe

- "DISCARD ATOMSERVICE" on page 352
- "DISCARD BUNDLE" on page 353
- "DISCARD EVENTBINDING" on page 354
- "DISCARD JVMSERVER" on page 355
- "DISCARD MQCONN" on page 356
- "INQUIRE ATOMSERVICE" on page 357
- "INQUIRE EVENTBINDING" on page 367
- "INQUIRE BUNDLE" on page 360
- "INQUIRE BUNDLEPART" on page 363
- "INQUIRE CAPTURESPEC" on page 365
- "INQUIRE EVENTPROCESS" on page 370
- "INQUIRE JVMSERVER" on page 371
- "INQUIRE MQCONN" on page 374
- "INQUIRE MQINI" on page 377
- "INQUIRE XMLTRANSFORM" on page 379
- "SET ATOMSERVICE" on page 383
- "SET BUNDLE" on page 384
- "SET EVENTBINDING" on page 386
- "SET EVENTPROCESS" on page 387
- "SET JVMSERVER" on page 388
- "SET MQCONN" on page 390
- "SET XMLTRANSFORM" on page 393

Existing API commands made threadsafe in this release

None.

Existing SPI commands made threadsafe in this release

None.

New API commands that are not threadsafe

"BIF DIGEST" on page 261

New SPI commands that are not threadsafe

"CREATE ATOMSERVICE" on page 305

"CREATE BUNDLE" on page 307

"CREATE JVMSERVER" on page 308

"CREATE MQCONN" on page 310

"CSD ADD" on page 312

"CSD ALTER" on page 314

"CSD APPEND" on page 316

"CSD COPY" on page 317

"CSD DEFINE" on page 320

"CSD DELETE" on page 324

"CSD DISCONNECT" on page 326

"CSD ENDBRGROUP" on page 326

"CSD ENDBRLIST" on page 327

"CSD ENDBRRSRCE" on page 328

"CSD GETNEXTGROUP" on page 328

"CSD GETNEXTLIST" on page 329

"CSD GETNEXTRSRCE" on page 330

"CSD INQUIREGROUP" on page 333

"CSD INQUIRELIST" on page 334

"CSD INQUIRERSRCE" on page 334

"CSD INSTALL" on page 337

"CSD LOCK" on page 340

"CSD REMOVE" on page 341

"CSD RENAME" on page 342

"CSD STARTBRGROUP" on page 344

"CSD STARTBRLIST" on page 346

"CSD STARTBRRSRCE" on page 347

"CSD UNLOCK" on page 348

"CSD USERDEFINE" on page 349

Chapter 42. 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 1.

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
Note: For Enterprise COBOL for z/OS V4.2, to make the compiler listings show the CICS options that are in effect, apply the PTF for APAR PK89224 to CICS.			
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 V3.5 and above	5655-H31	Supported	Supported

C and C++

Product name	PID	Translator	Run time
z/OS V1.9 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 V1.5 and above	5696-234	Supported	Supported

Java

Product name	PID	Run time
IBM 31-bit SDK for z/OS, Java Technology Edition, V6	5655-R31	Supported

Product name	PID	Run time
IBM 64-bit SDK for z/OS, Java Technology Edition, V6	5655-R32	Cannot run
IBM SDK for z/OS, Java Technology Edition, V5	5655-N98	See note
IBM SDK for z/OS, Java Technology Edition, V1.4.2	5655-I56	See note
Note: Version 1.4.2 and Version 5 of the IBM 31-bit SDK for z/OS Java 2 Technology Edition cannot be used with CICS TS 4.1. Most Java applications created using these SDK versions are expected to run on Version 6 of the 31-bit SDK (5655-R31) unchanged, provided the applications do not use deprecated APIs, and are not affected by the small number of incompatibilities introduced industry wide with Java version changes.		

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 1 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.

Language Environment 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 Language Environment.

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 1. 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.

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.

COBOL

Product name	PID	CICS translator	Integrated CICS translator	Run time
OS/VS COBOL (pre-Language Environment)	5740-CB1 5740-LM1	Cannot use	Cannot use	Cannot run under CICS

Product name	PID	CICS translator	Integrated CICS translator	Run time
VS COBOL II Release 3 or later (pre-Language Environment)	5668-023 5668-958	Cannot use	Cannot use	Supported
COBOL/370	5688-197	Supported	Cannot use	Supported
COBOL for MVS & VM V1.2	5688-197	Supported	Cannot use	Supported
COBOL for OS/390® & VM V2.1	5648-A25	Supported	Cannot use	Supported
COBOL for OS/390 & VM V2.2	5648-A25	Supported	Supported	Supported
Enterprise COBOL for z/OS and OS/390 V3.1 and V3.2	5655-G53	Supported	Supported	Supported
Enterprise COBOL for z/OS V3.3	5655-G53	Supported	Supported	Supported

PL/I

Product name	PID	CICS translator	CICS integrated translator	Run time
OS PL/I Optimizing Compiler Version 1 (pre-Language Environment)	5734-PL1	Cannot use	Cannot use	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	Cannot use	Cannot use	Supported
SAA AD/Cycle® PL/I for MVS and VM	5688-235	Supported	Cannot use	Supported
PL/I for MVS and VM V1	5688-235	Supported	Cannot use	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, and V3.4	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	Cannot use	Supported
C/370 V2 (pre-Language Environment)	5688-187 5688-188	Supported	Cannot use	Supported
SAA AD/Cycle C/370	5688-216	Cannot use	Cannot use	Supported
C/C++ for MVS/ESA	5655-121	Supported	Cannot use	Supported
C/C++ for OS/390	5647-A01 - Component of OS/390	Supported	Cannot use	Supported
C/C++ for z/OS and OS/390	5694-A01 - Component of z/OS	Supported	Cannot use	Supported
z/OS V1.4, V1.5, and V1.6 C/C++	5694-A01 - Component of z/OS	Supported	Cannot use	Supported
z/OS V1.7 and V1.8 XL C/C++	5694-A01 - Component of z/OS	Supported	Supported	Supported

Java

Product name	PID	Run time
Java for OS/390 V1.1.8	5655-A46	These products cannot be used with CICS TS 4.1. Java application bytecode created using these products should run on the supported product IBM SDK for z/OS Java 2 Technology Edition, SDK 1.6 (5655-R31), 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	

Chapter 43. 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 at the following locations:

- **Upgrading > Upgrading from Version 2 Release 3**
- **Upgrading > Upgrading from Version 3 Release 1**
- **Upgrading > Upgrading from Version 3 Release 2**

The same information sets are listed under the library section of the navigation at the following locations:

- **The CICS Transaction Server library: HTML > Upgrading from Version 2 Release 3**
- **The CICS Transaction Server library: HTML > Upgrading from Version 3 Release 1**
- **The CICS Transaction Server library: HTML > Upgrading from Version 3 Release 2**

The PDF versions of these information sets are listed on the page **The CICS Transaction Server library: PDF**.

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

Part 8. Publications

Chapter 44. 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 1, 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, GI13-0536
CICS Transaction Server for z/OS What's New, GC34-6994
CICS Transaction Server for z/OS Upgrading from CICS TS Version 2.3, GC34-6996
CICS Transaction Server for z/OS Upgrading from CICS TS Version 3.1, GC34-6997
CICS Transaction Server for z/OS Upgrading from CICS TS Version 3.2, GC34-6998
CICS Transaction Server for z/OS Installation Guide, GC34-6995

Access to CICS

CICS Internet Guide, SC34-7021
CICS Web Services Guide, SC34-7020

Administration

CICS System Definition Guide, SC34-6999
CICS Customization Guide, SC34-7001
CICS Resource Definition Guide, SC34-7000
CICS Operations and Utilities Guide, SC34-7002
CICS RACF Security Guide, SC34-7003
CICS Supplied Transactions, SC34-7004

Programming

CICS Application Programming Guide, SC34-7022
CICS Application Programming Reference, SC34-7023
CICS System Programming Reference, SC34-7024
CICS Front End Programming Interface User's Guide, SC34-7027
CICS C++ OO Class Libraries, SC34-7026
CICS Distributed Transaction Programming Guide, SC34-7028
CICS Business Transaction Services, SC34-7029
Java Applications in CICS, SC34-7025

Diagnosis

CICS Problem Determination Guide, GC34-7034
CICS Performance Guide, SC34-7033
CICS Messages and Codes, SC34-7035
CICS Diagnosis Reference, GC34-7038
CICS Recovery and Restart Guide, SC34-7012
CICS Data Areas, GC34-7014
CICS Trace Entries, SC34-7013
CICS Supplementary Data Areas, GC34-7015
CICS Debugging Tools Interfaces Reference, GC34-7039

Communication

CICS Intercommunication Guide, SC34-7018
CICS External Interfaces Guide, SC34-7019

Databases

CICS DB2 Guide, SC34-7011
CICS IMS Database Control Guide, SC34-7016
CICS Shared Data Tables Guide, SC34-7017

CICSplex SM books for CICS Transaction Server for z/OS

General

CICSplex SM Concepts and Planning, SC34-7044
CICSplex SM Web User Interface Guide, SC34-7045

Administration and Management

CICSplex SM Administration, SC34-7005
CICSplex SM Operations Views Reference, SC34-7006
CICSplex SM Monitor Views Reference, SC34-7007
CICSplex SM Managing Workloads, SC34-7008
CICSplex SM Managing Resource Usage, SC34-7009
CICSplex SM Managing Business Applications, SC34-7010

Programming

CICSplex SM Application Programming Guide, SC34-7030
CICSplex SM Application Programming Reference, SC34-7031

Diagnosis

CICSplex SM Resource Tables Reference, SC34-7032
CICSplex SM Messages and Codes, GC34-7035
CICSplex SM Problem Determination, GC34-7037

Licensed publications

None of the publications for CICS Transaction Server for z/OS, Version 4 Release 1 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 1.

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*

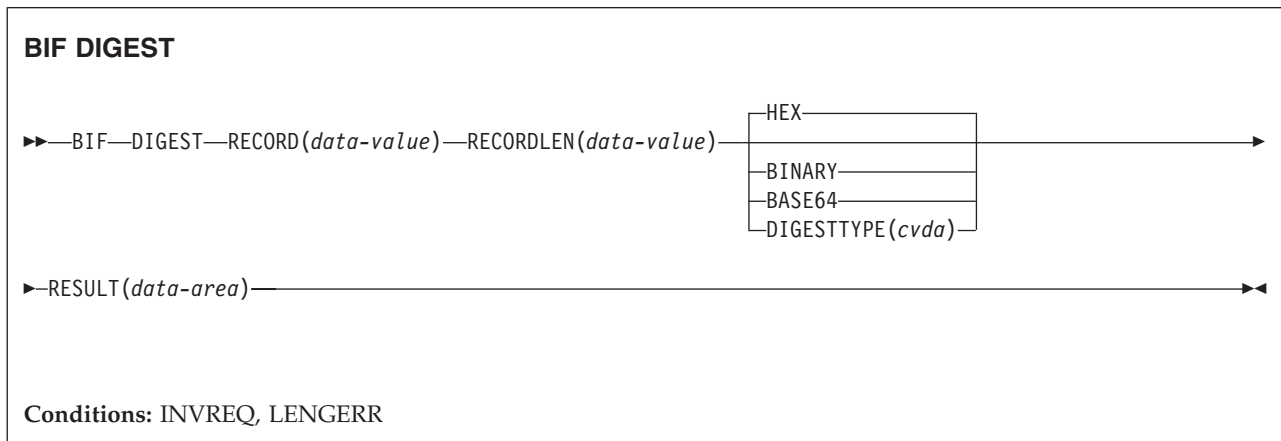
Part 9. Appendixes

Appendix A. New application programming commands

CICS Transaction Server for z/OS, Version 4 Release 1 extends the CICS application programming interface with new commands.

BIF DIGEST

Calculate the SHA-1 digest of a string of data.



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.

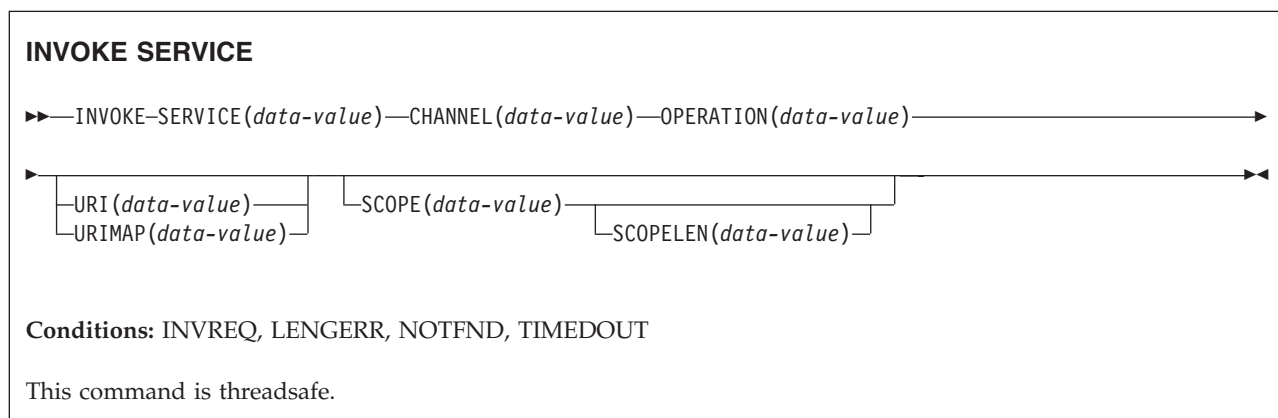
22 LENGERR

RESP2 values:

- 2 The RECORDLEN value is less than 1.

INVOKE SERVICE

This command calls 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.



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 about writing a web service application, see the *CICS Web Services 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** 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 this 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 or for SCA applications.

URIMAP(*data-value*)

Specifies the name of the URIMAP resource that CICS uses to derive the URI value. The URIMAP must be defined as an HTTP client with the attribute USAGE(CLIENT). 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 invalid character or combination of characters.
- 2 The name specified for the OPERATION option contains an invalid 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 invalid 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 can 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 DFHPIxxxx 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 data structure of the application. Some possible causes of this condition are:

- A value in the SOAP response message is larger than the corresponding field in the data structure of the application.
- 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 DFHPIxxxx 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.
- 41 The connection has been closed.
- 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 an unrecoverable 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.

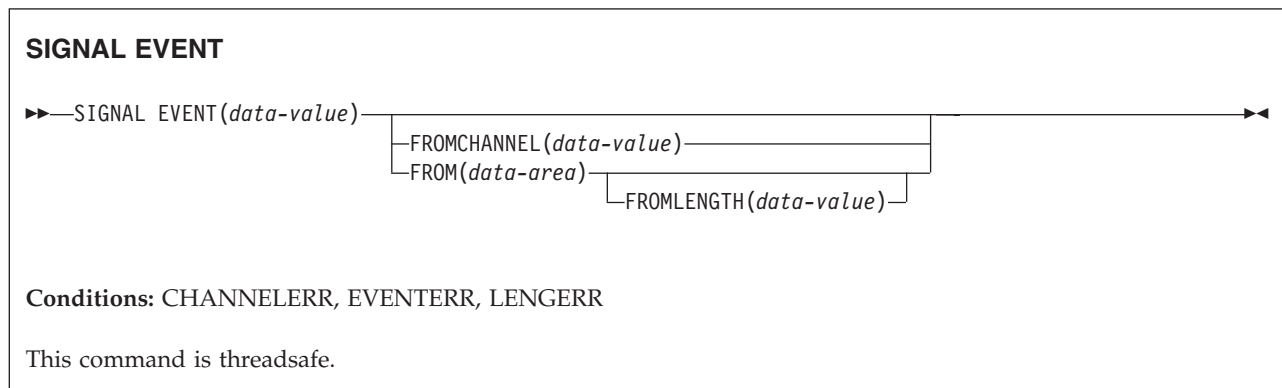
124 TIMEDOUT

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.



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 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.

TYPE(*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. 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.

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

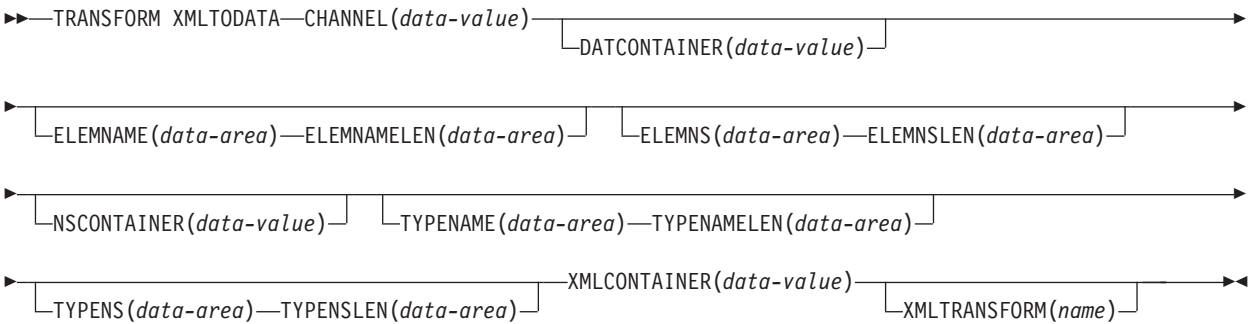
RESP2 values:

- 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.
- 7 The container is not in text mode.
- 8 The container is not 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 The container is not in text mode.
- 8 The container is not 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.

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.

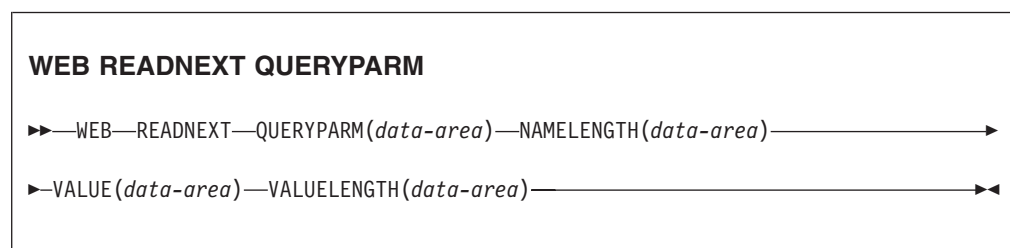
13 NOTFND

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.



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.

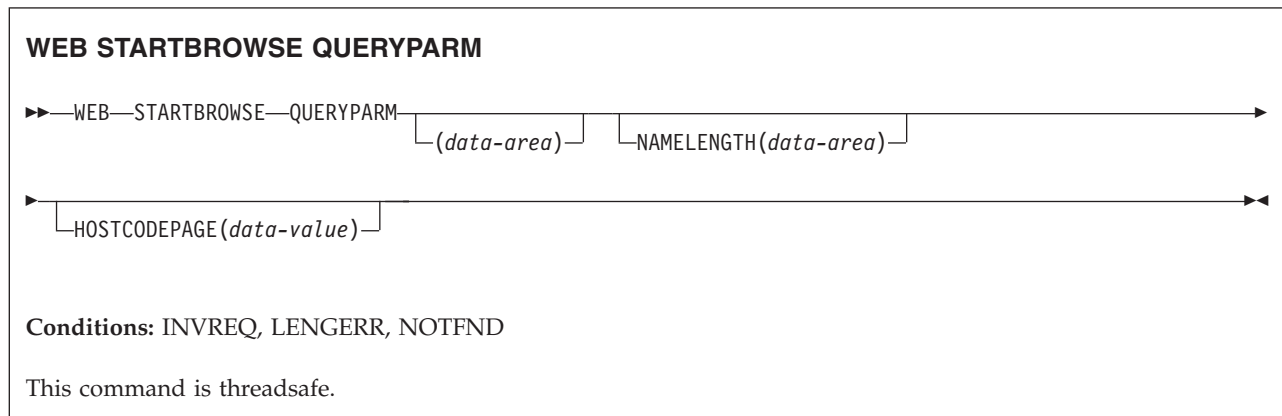
22 LENGERR

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.

WEB STARTBROWSE QUERYPARM

Start browsing query string data in a URL.



Description

WEB STARTBROWSE QUERYPARM signals the start of a browse of the keyword parameters, consisting of name and value pairs, from a query string in a URL.

For forms, you can also use the WEB STARTBROWSE FORMFIELD command. Forms are messages with the media types application/x-www-form-urlencoded or multipart/form-data.

Options

QUERYPARM(*data-area*)

is the keyword that initiates the STARTBROWSE QUERYPARM command. You can optionally specify the name of the keyword parameter at which browsing is to start, by specifying this in a data-area, followed by the NAMELENGTH option, for example,

```
WEB STARTBROWSE QUERYPARM(name) NAMELENGTH(1en)
```

The name is a string of text containing the name of the requested keyword parameter. If a name is not specified, browsing starts at the first name and value pair in the query string.

HOSTCODEPAGE(*data-value*)

specifies the 8-character name of the CICS (host) code page required by the application program, into which the query string data is to be converted. This code page is normally an EBCDIC code page.

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 now 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 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

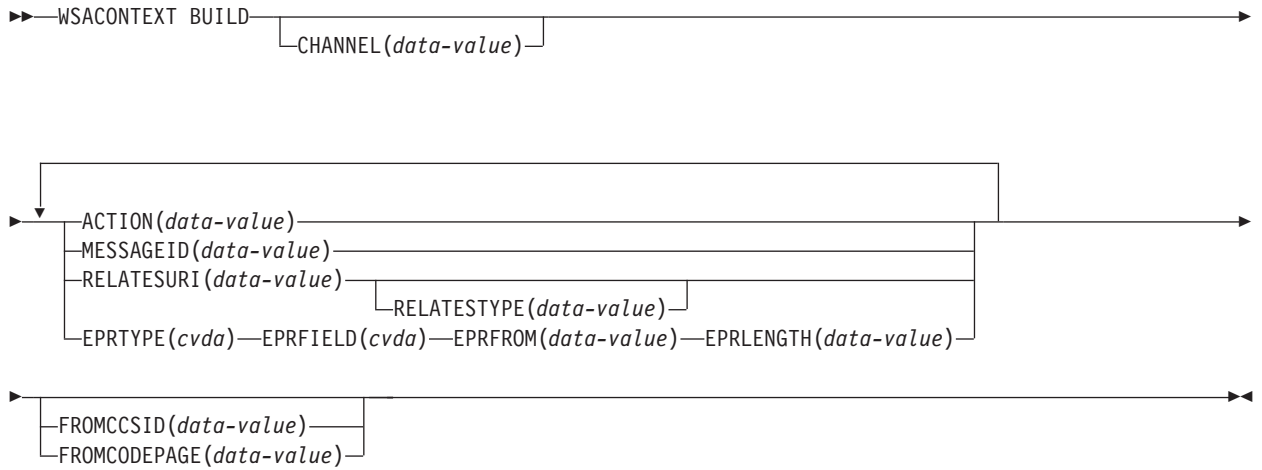
RESP2 value is:

- 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 holding 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:

" , > , < , ' , &

EPRLLENGTH(*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:

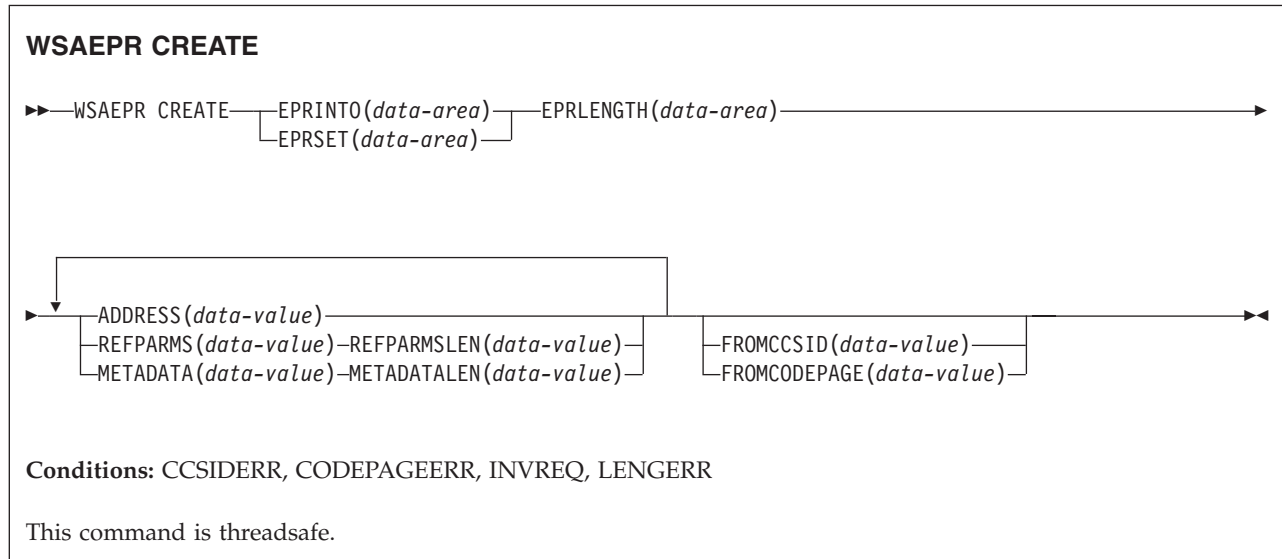
- 20 This condition occurs when the length of the stored data is greater

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.



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.

METADATALEN(*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

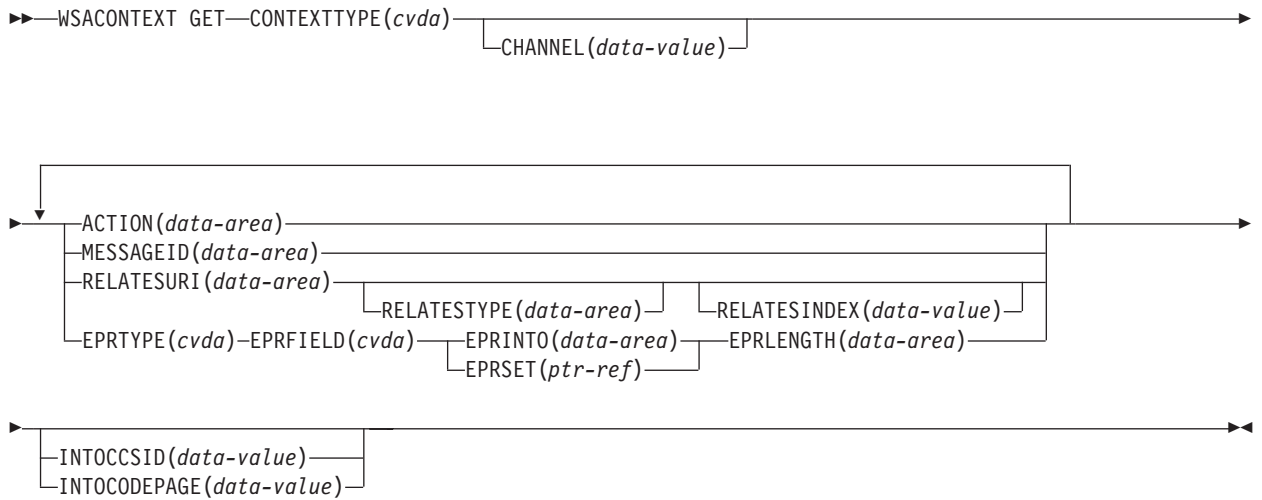
RESP2 values:

- 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.

WSACONTEXT GET



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 holding 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 recommend 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.

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.

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.

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 the *CICS Family: Communicating from CICS on zSeries*.

INTOCODEPAGE(*data-value*)

Specifies an IANA-registered alphanumeric charset name or a Coded Character Set Identifier (CCSID) for the code page into which the character data in the 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 the *CICS Family: Communicating from CICS on zSeries*.

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.

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.

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.

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 1.

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.

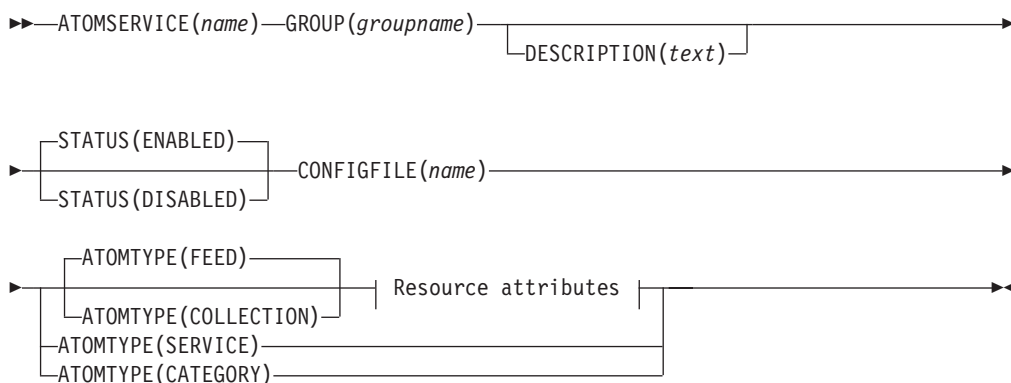
Atom feed documents and collections are documents that contain one or more Atom entries. Web clients cannot edit the Atom entries in a feed, but they can edit the Atom entries in a collection. When a Web client makes an HTTP request relating to the Atom feed or collection, CICS assembles the appropriate 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.

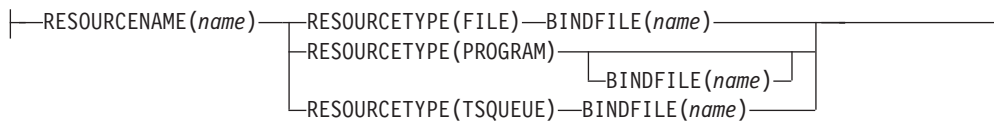
ATOMSERVICE attributes

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 DFHATOMPARM 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. The *CICS Internet Guide* explains the syntax.

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

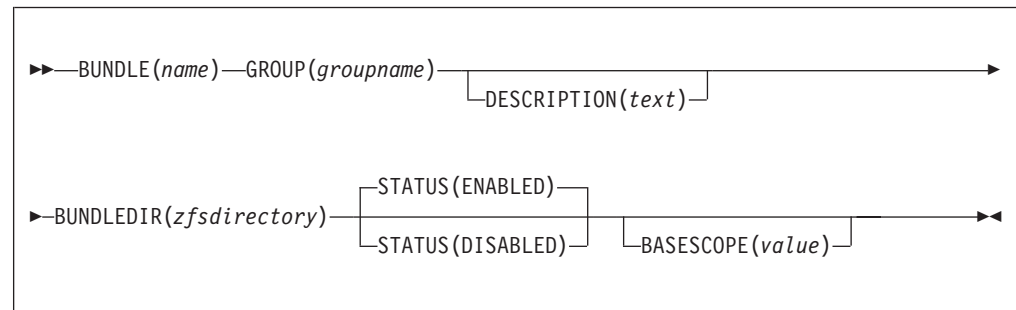
state in the CICS region. If you reenable the resource successfully, the BUNDLE resource also changes to the ENABLED state. If you try to discard a disabled BUNDLE resource when enabled resources that belong to the bundle are in the CICS region, CICS issues a message and the discard fails. You must disable 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.

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.

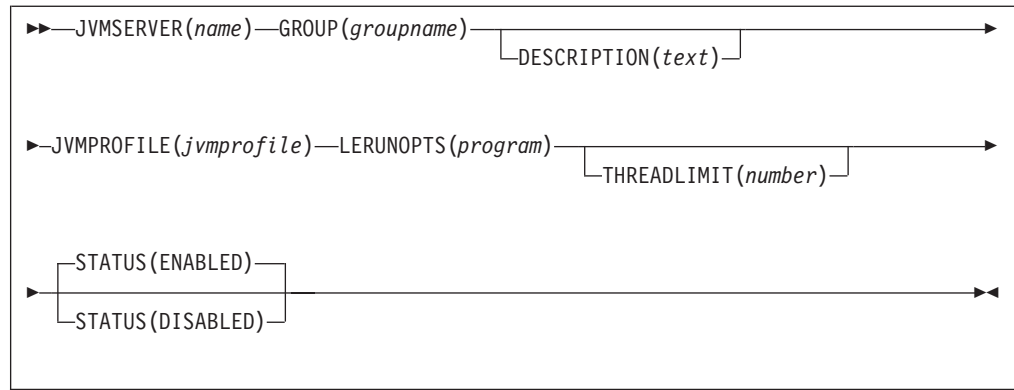
JVMSERVER resources

A JVMSERVER resource defines the runtime environment for a JVM server.

The JVMSERVER 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.

JVMSERVER attributes

Describes the syntax and attributes of the JVMSERVER 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(*jvmprofile*)

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(*program*)

Specifies the 1 - 8 character name of the program that defines the runtime

options for the Language Environment enclave. The program must be in the *hlq.SDFHLOAD* library. You must specify the name of the program in uppercase characters.

You can use the sample program, DFHAXRO, which 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.

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 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.

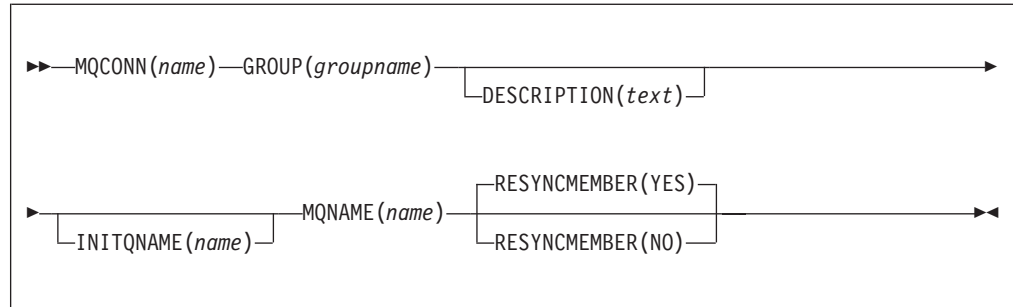
When you install an MQCONN resource that includes a setting for the INITQNAME attribute, CICS also dynamically creates and installs an MQINI resource. The MQINI resource represents the default initiation queue that is specified by the INITQNAME attribute of the MQCONN resource. The name of the MQINI 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-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-MQ connection 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-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.

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-MQ connection 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})

This attribute applies only if you have used the MQNAME attribute to specify a WebSphere MQ queue-sharing group. 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.

- YES** CICS connects to the same queue manager, waiting, if necessary, until the queue manager becomes active to resolve the indoubt units of work. This 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.

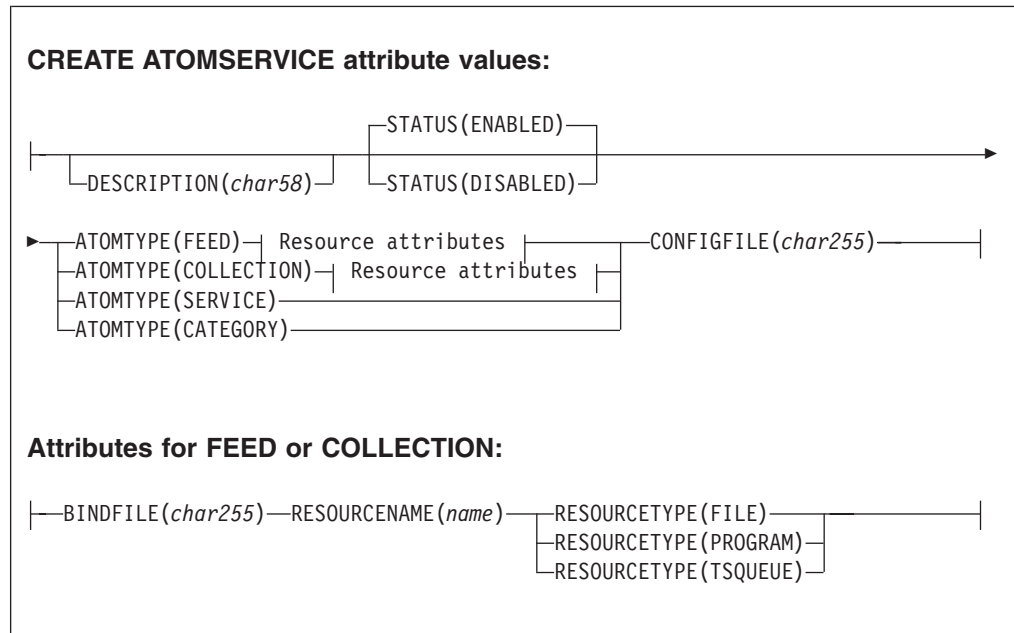
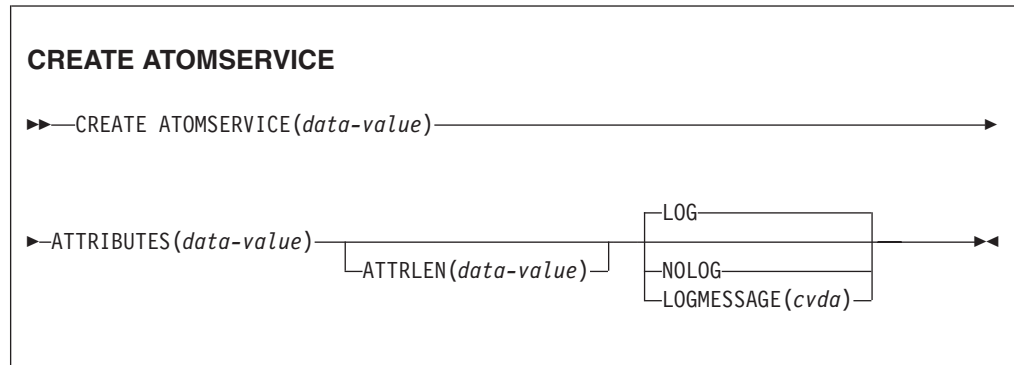
Units of work that are shunted indoubt are not included in this process, because CICS itself cannot resolve those units of work at this time. Resynchronization for those UOWs occurs when CICS has resynchronized with its remote coordinator.

Appendix C. New system programming commands

CICS Transaction Server for z/OS, Version 4 Release 1 extends the CICS system programming interface with new commands.

CREATE ATOMSERVICE

Define an ATOMSERVICE resource definition 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 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.

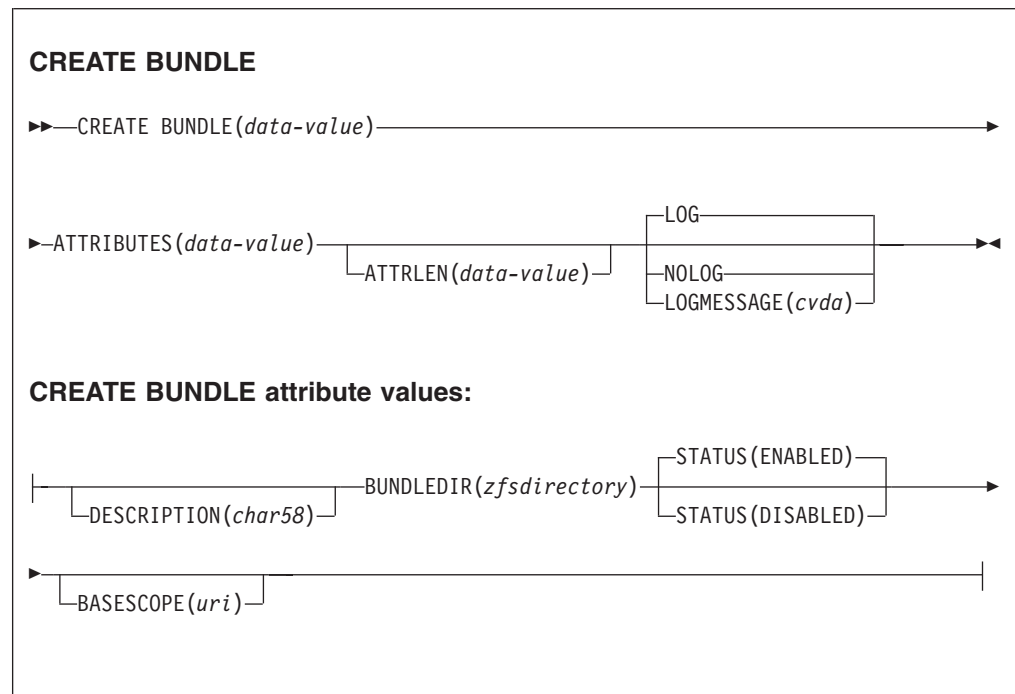
NOTAUTH

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.
- 634** Installation of the BUNDLE resource failed because one of its associated resources failed to install.

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.

NOLOG

The resource attributes are not logged.

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.

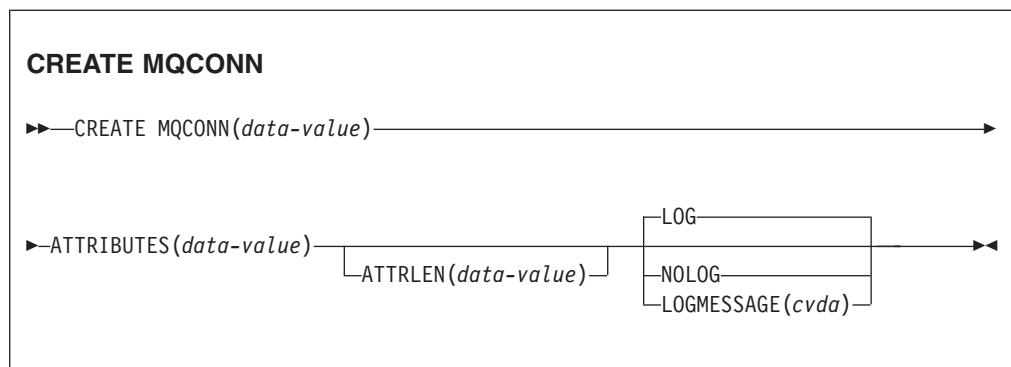
NOTAUTH

RESP2 values:

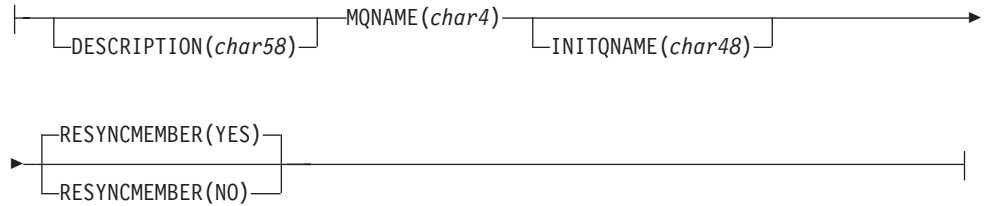
- 100** The user associated with the issuing task is not authorized to use this command.
- 101** The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

CREATE MQCONN

Define an MQCONN resource in the local CICS region.



CREATE MQCONN attribute values:



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 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 a 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 from 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 executed 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.

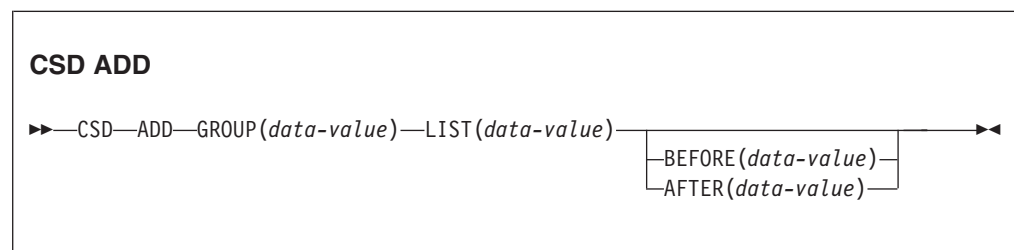
NOTAUTH

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.

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.

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:

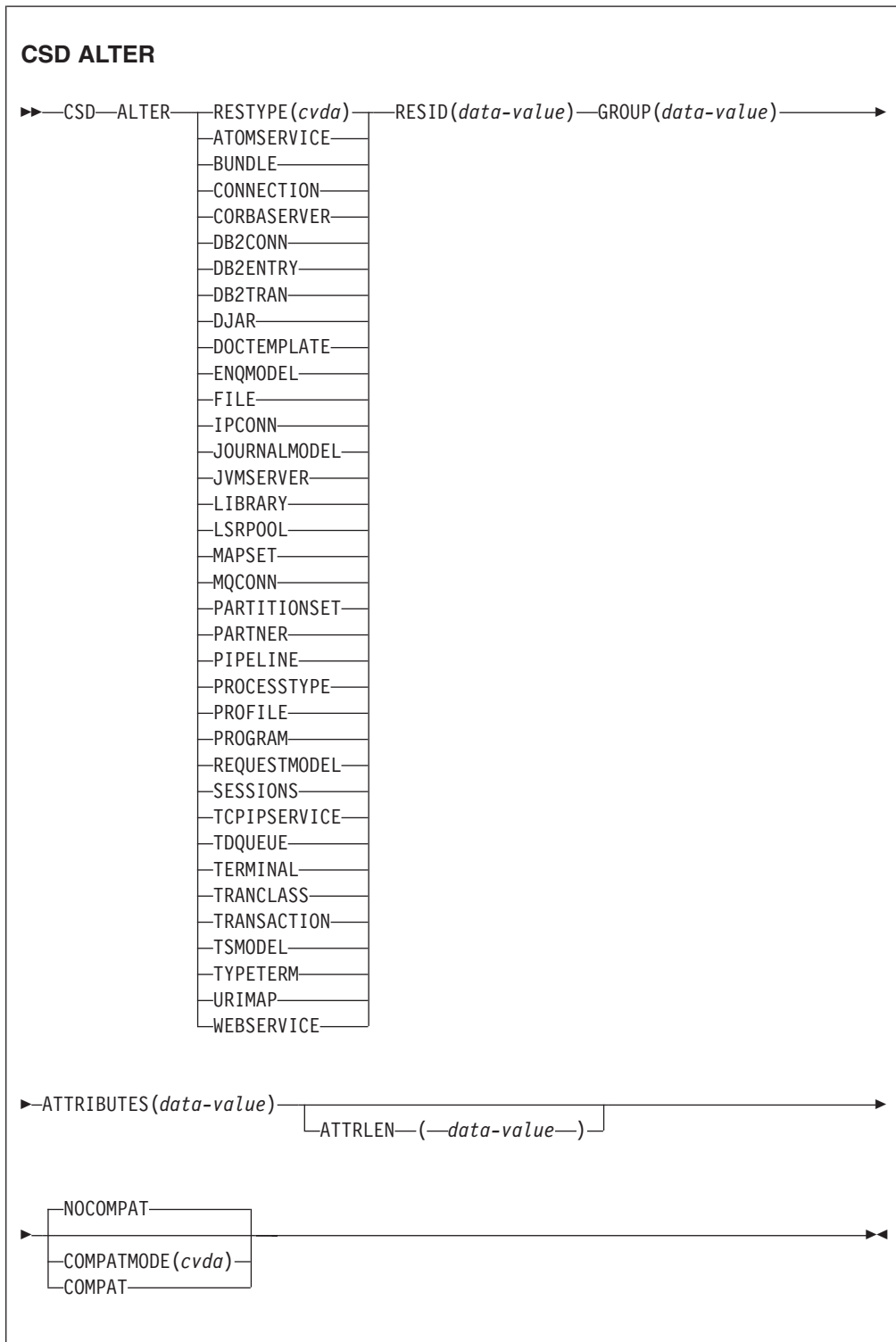
- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 value:

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.

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

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.

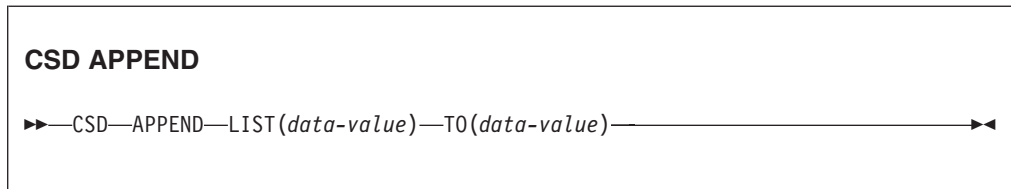
NOTFND

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.



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.

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

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.

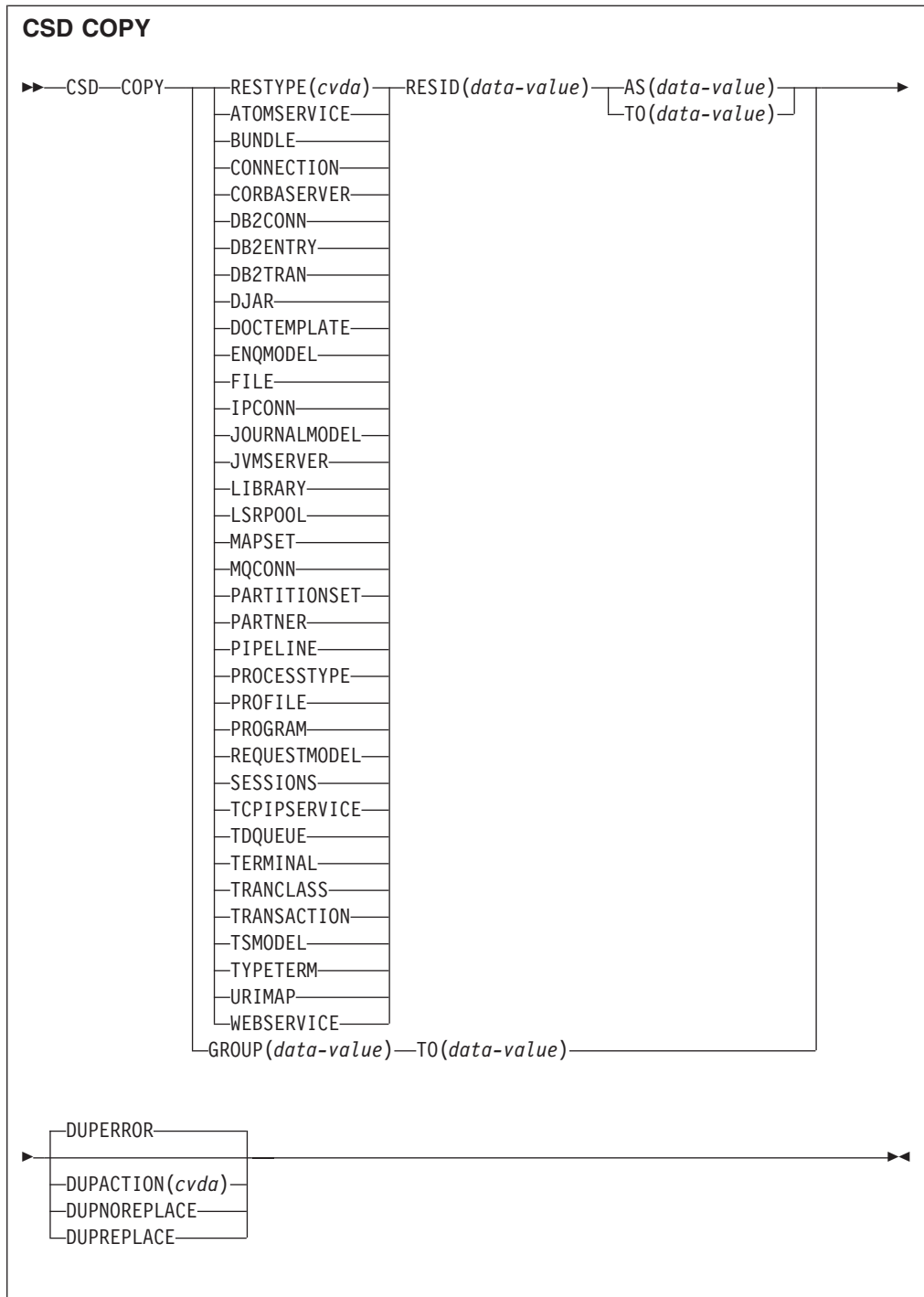
NOTFND

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.

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

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.

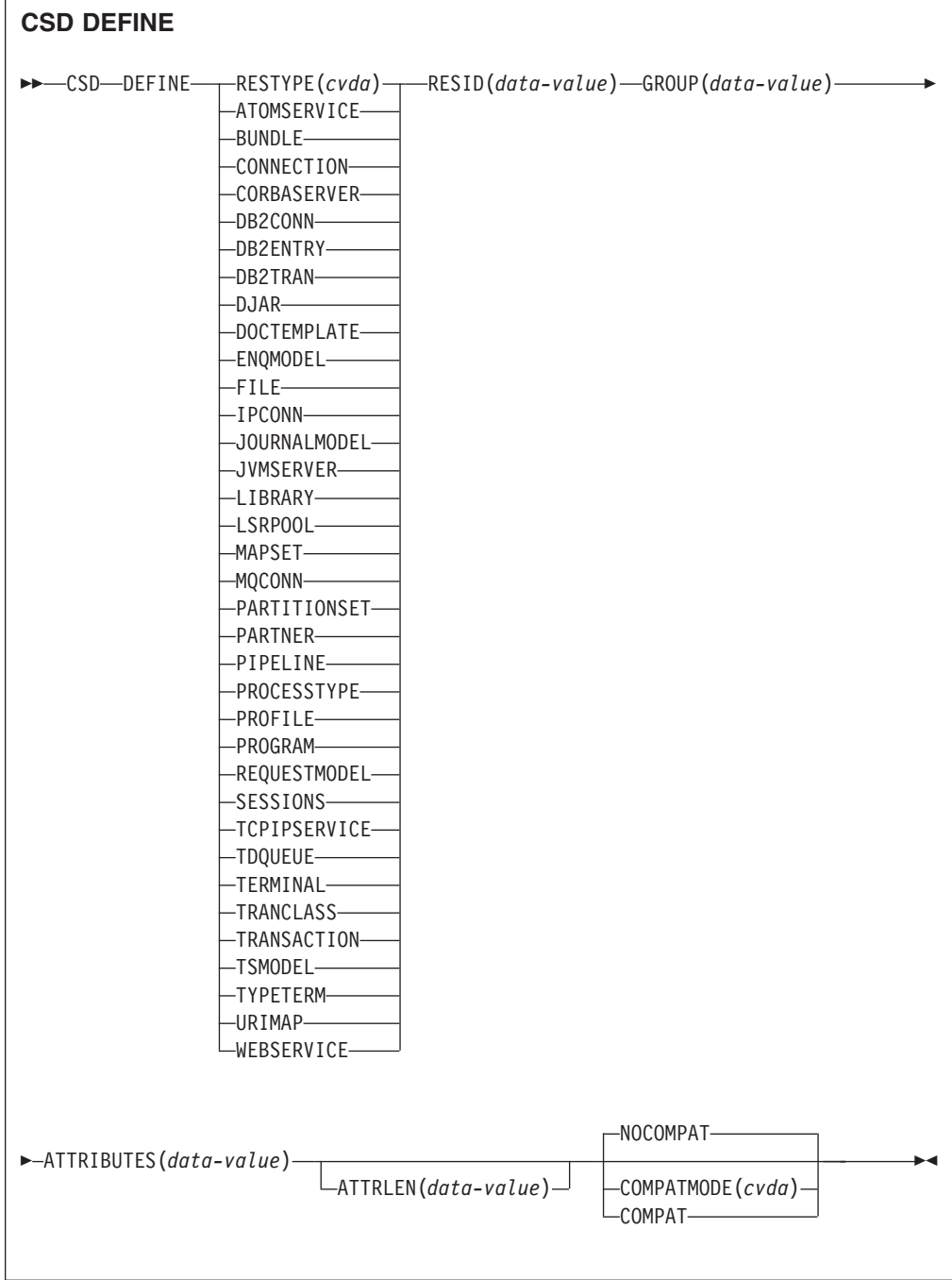
NOTFND

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.

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

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.

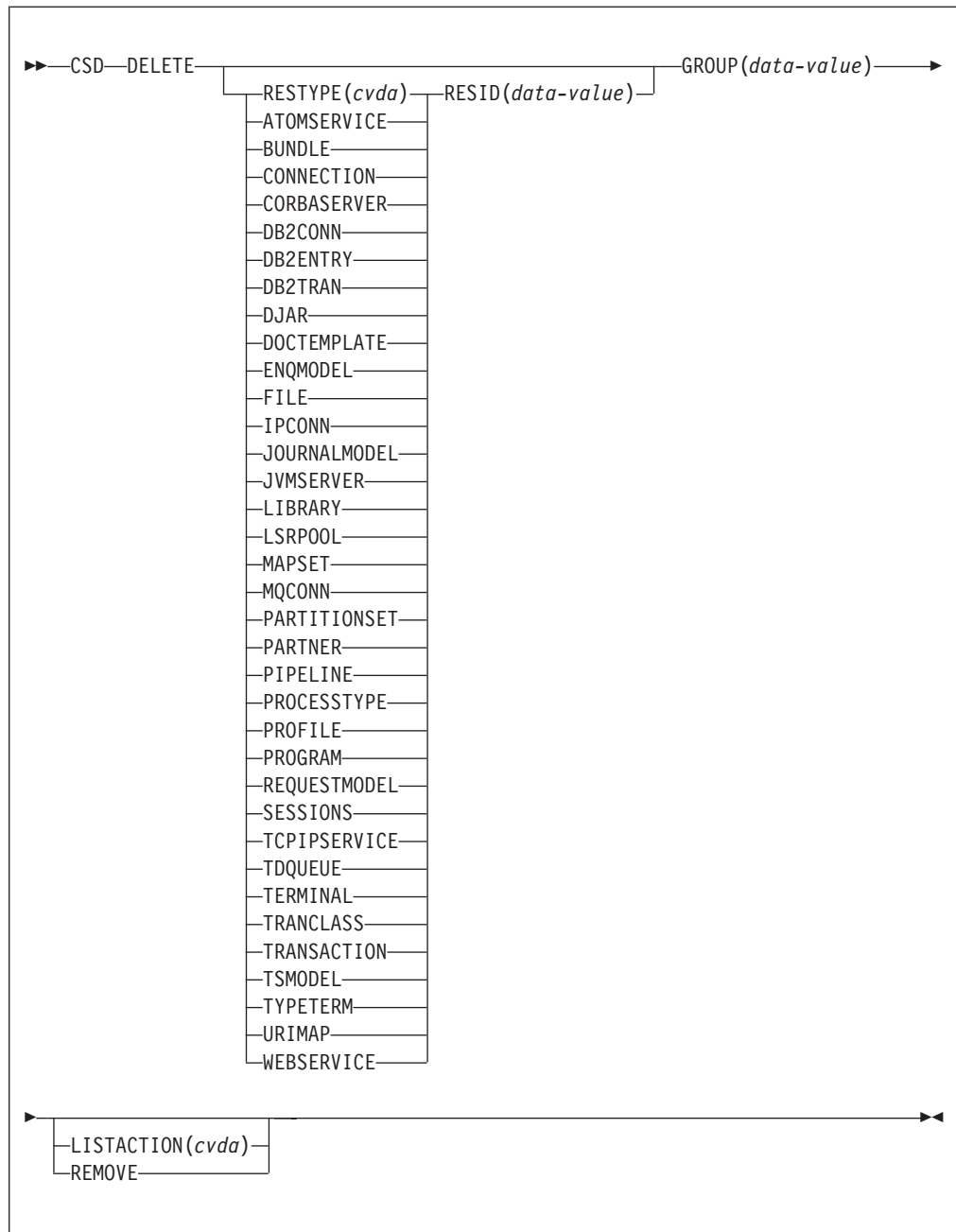
NOTAUTH

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.

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.

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:

- 100 The user associated with the issuing task is not authorized to use this command.

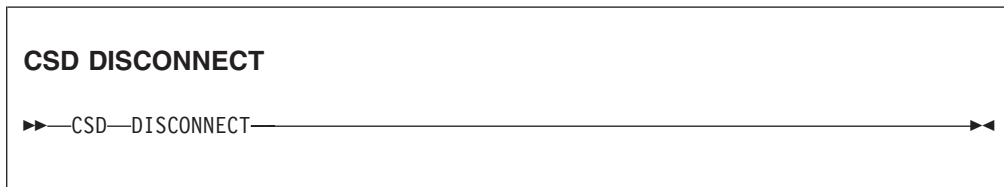
NOTFND

RESP2 values:

- 1 The specified individual resource definition cannot be found.

CSD DISCONNECT

Disconnect the current task from the CSD.



Conditions: NOTAUTH

Description

The CSD DISCONNECT command removes the current task's connection to the CSD. This connection is acquired automatically when a task issues its first EXEC CICS CSD command. CSD DISCONNECT closes the CSD if no other tasks are accessing it. This command is not normally necessary because this processing occurs automatically at task end.

Consider using CSD DISCONNECT for long-running tasks after each series of CSD commands, particularly in a non-RLS environment, because another system cannot share the CSD while a task in this system is connected.

Note: CSD DISCONNECT causes a syncpoint.

Options

None

Conditions

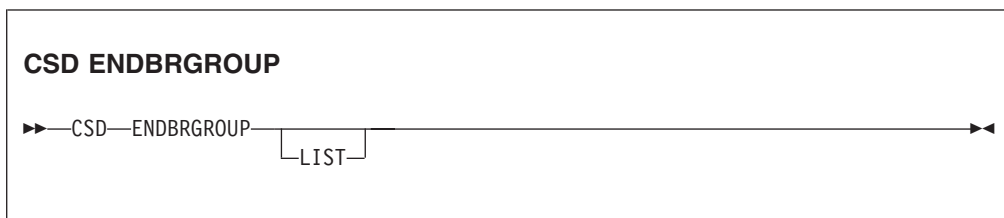
NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

CSD ENDBRGROUP

End the current browse of the groups in the CSD, or of the groups in a LIST.



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.

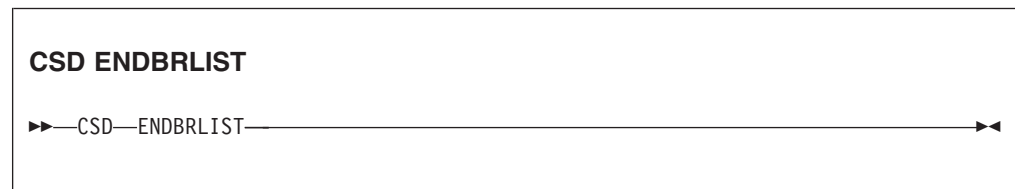
NOTAUTH

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.

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.

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.

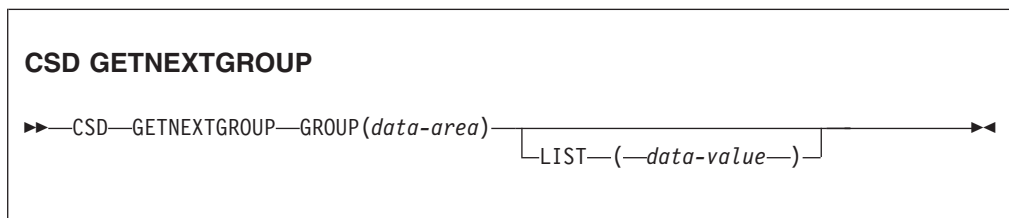
NOTAUTH

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.

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.

CSD GETNEXTLIST

▶—CSD—GETNEXTLIST—LIST(*data-area*)—◀

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.

END

RESP2 value:

- 1 The CSD contains no more lists.

ILLOGIC

RESP2 value:

- 1 A list browse is not in progress.

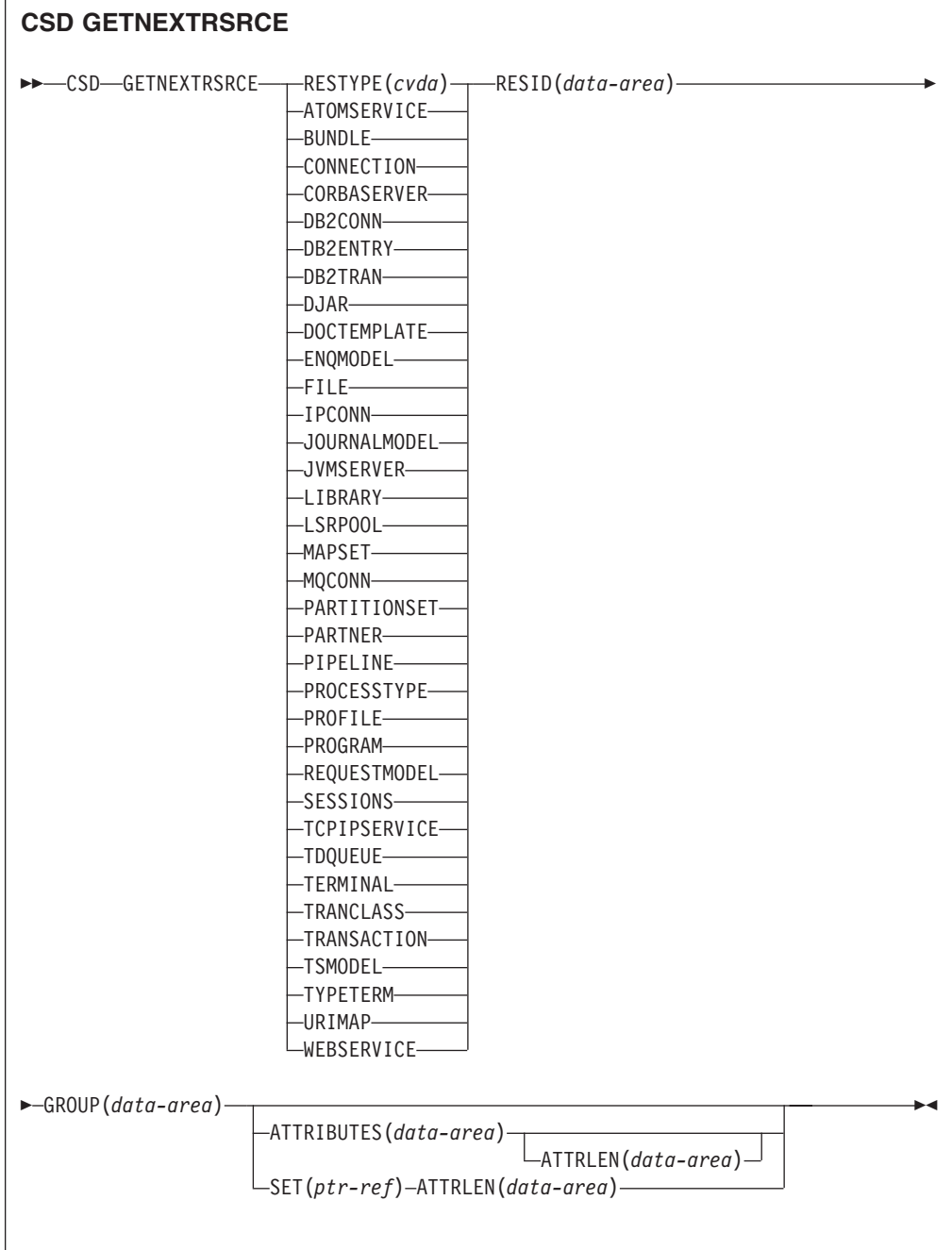
NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

CSD GETNEXTRSRCE

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.

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.

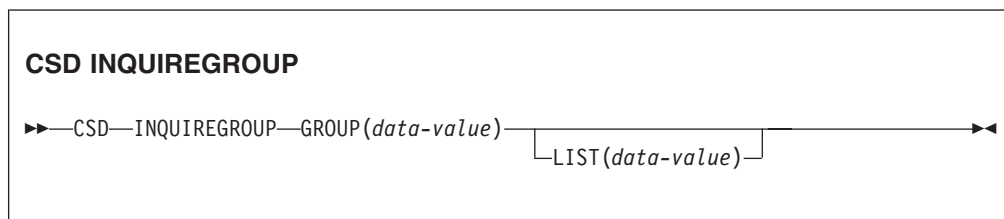
NOTAUTH

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.

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.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

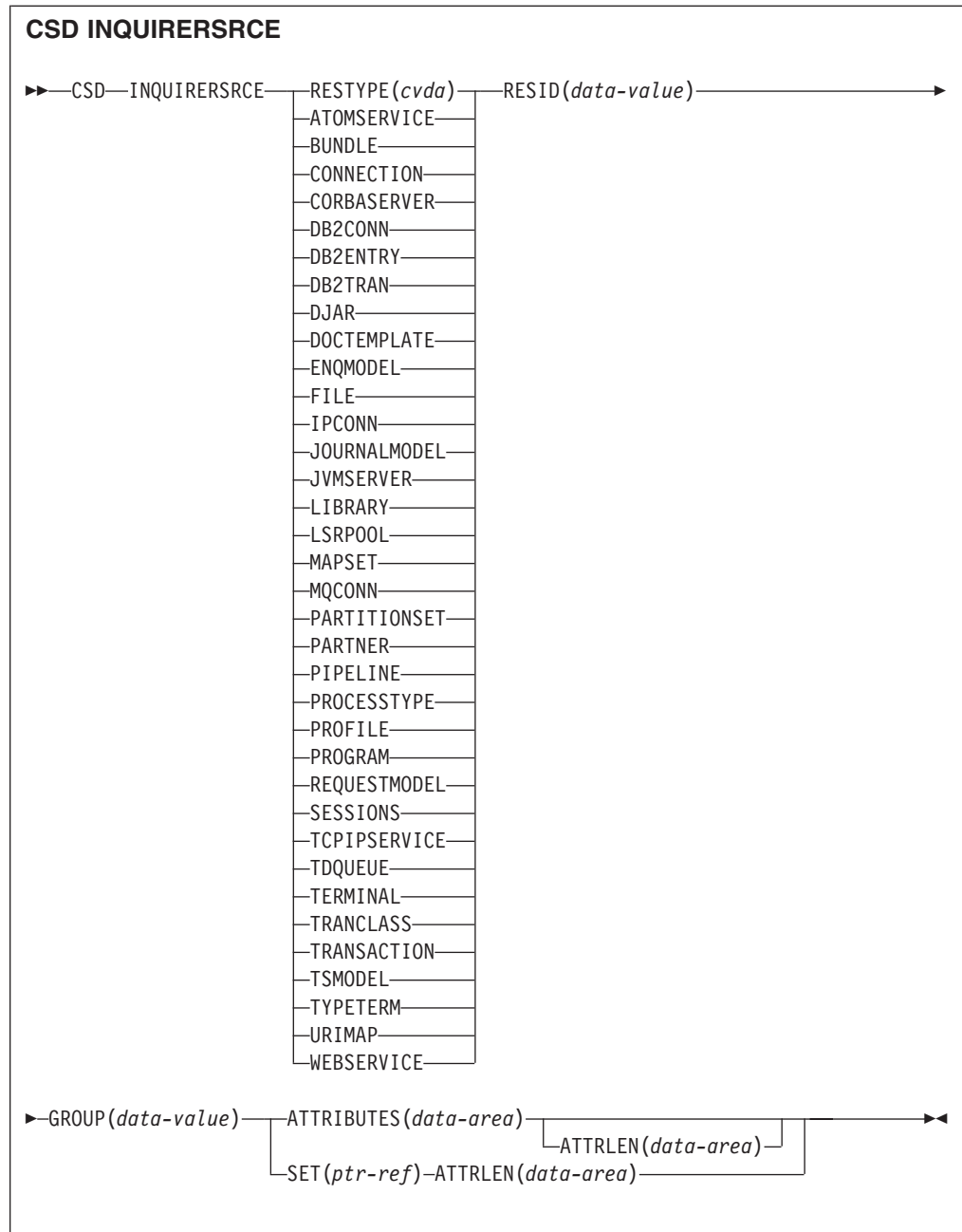
NOTFND

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.
- 4 The CSD is being used by another CICS system and is not configured for sharing.

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:

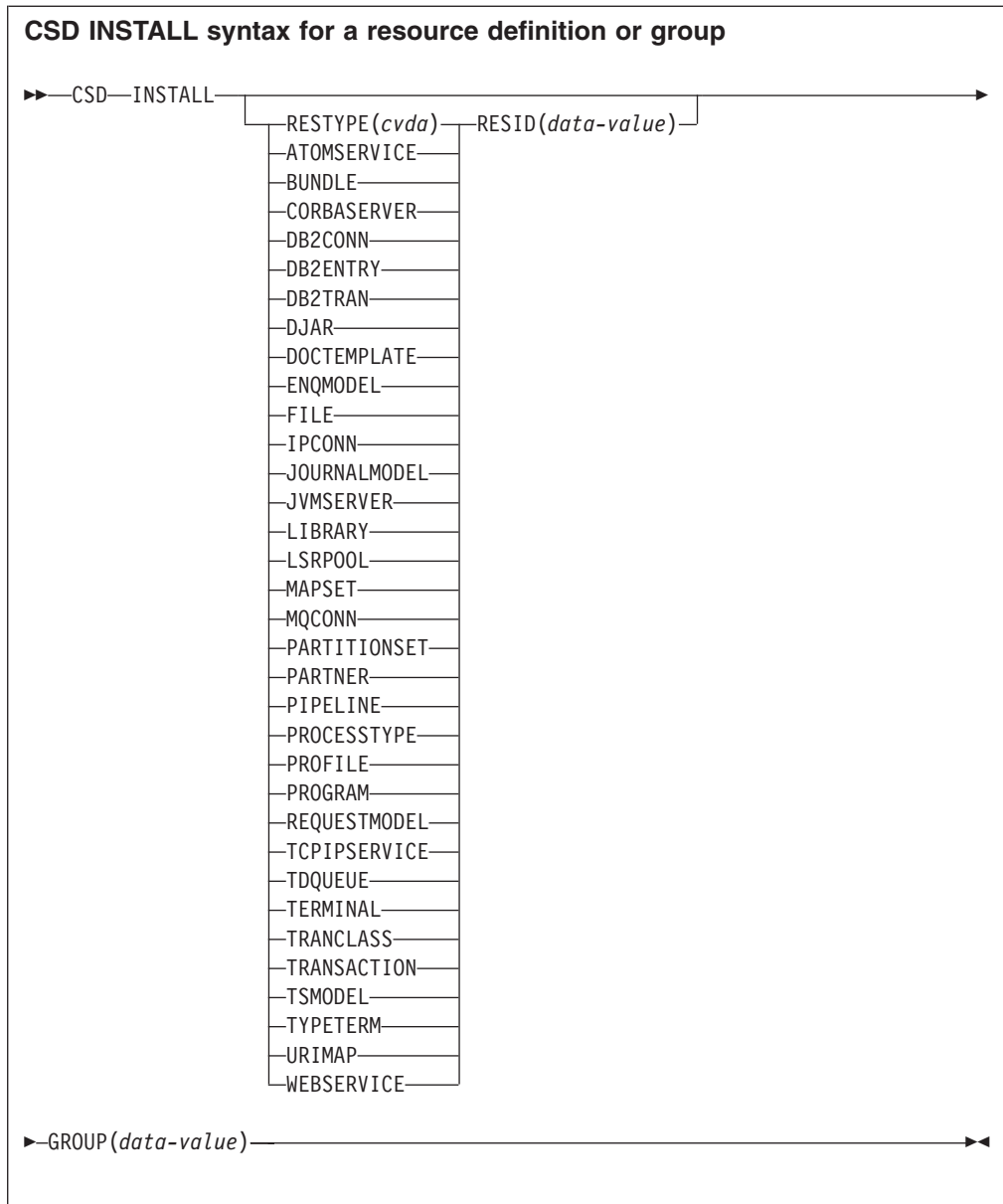
- 1 The specified resource definition cannot be found.
- 2 The specified group cannot be found.

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.

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.

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.

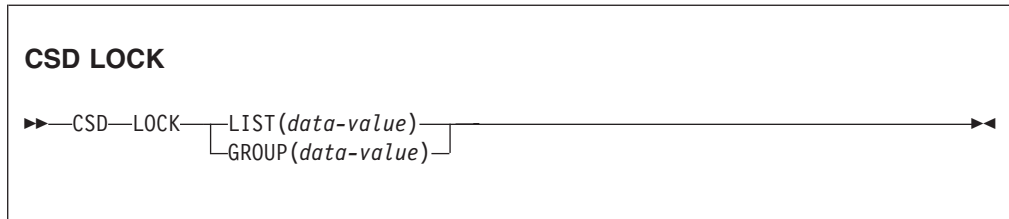
NOTFND

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.

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.

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.

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.

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.

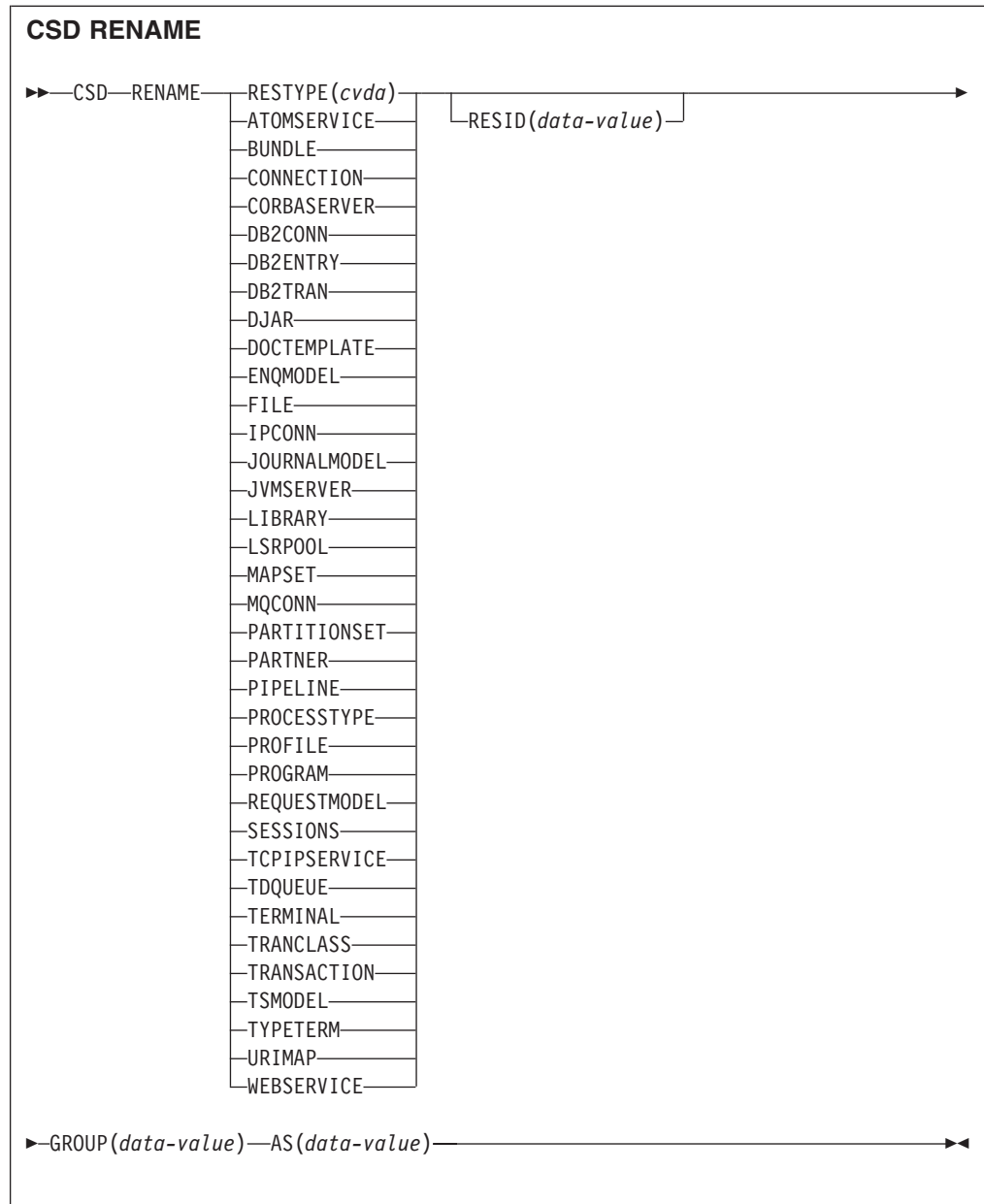
NOTFND

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.

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.

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.

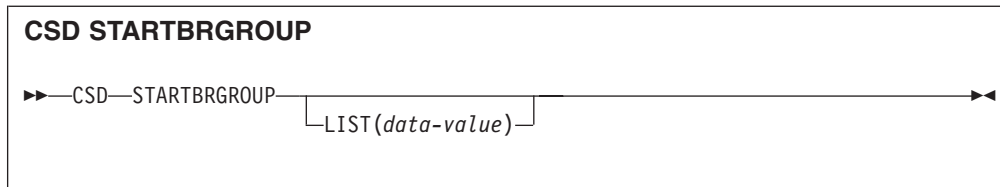
NOTFND

RESP2 values:

- 1 The specified resource definition cannot be found.
- 2 The specified group 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.

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.

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.

CSD STARTBRRSRCE

►►—CSD—STARTBRRSRCE—GROUP(*data-value*)—————◄◄

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.

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.

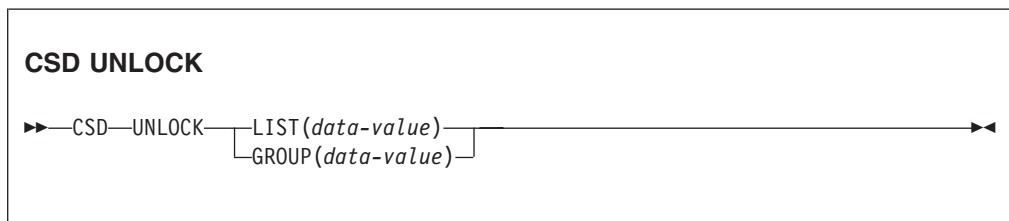
NOTFND

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.

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.

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.

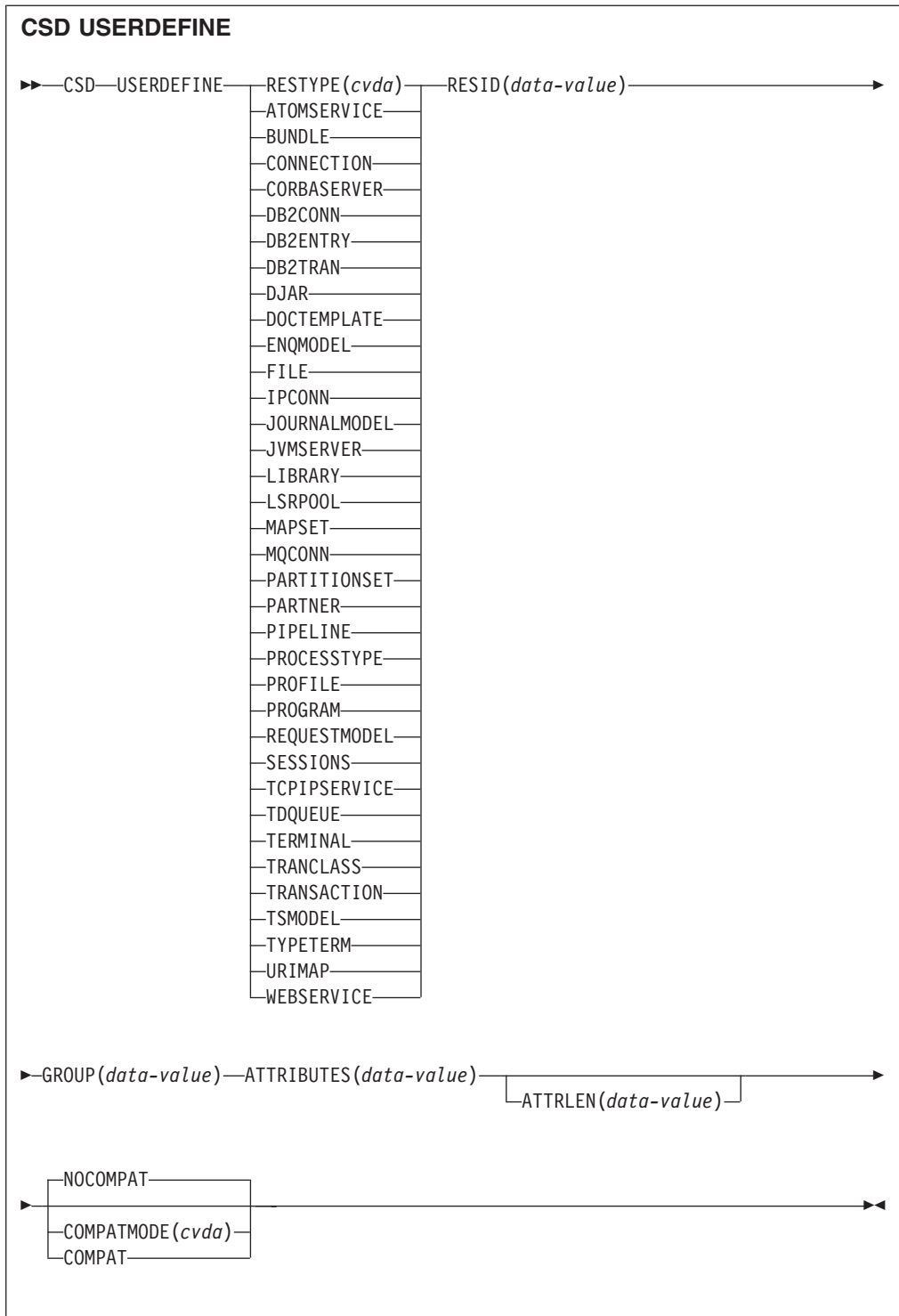
NOTAUTH

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.

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.

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.

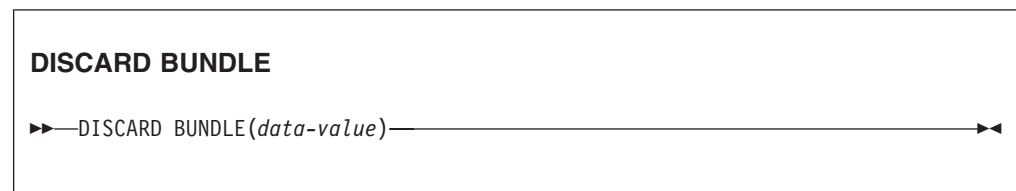
NOTFND

RESP2 values:

- 3 The ATOMSERVICE cannot be found.

DISCARD BUNDLE

Remove a BUNDLE resource definition.



Conditions: INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

Use the DISCARD BUNDLE to remove a BUNDLE resource from your CICS region. The BUNDLE and the resources it dynamically created must be disabled 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.
- 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 EVENTBINDING

Remove the definition of an event binding from the CICS system, so that the system no longer has access to the resource.

DISCARD EVENTBINDING

▶▶—DISCARD EVENTBINDING(*data-value*)—◀◀

Conditions: INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

The DISCARD EVENTBINDING command removes an event binding from the local CICS system. The event binding must be disabled before it can be discarded. Capture of events matching capture specifications within this event binding is stopped immediately. Any events already captured will be emitted.

Options

EVENTBINDING (*data-value*)

specifies the 32-character name of an event binding.

Conditions

INVREQ

RESP2 values are:

- 7 This event binding is enabled.

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 alter the event binding.

NOTFND

RESP2 values are:

- 3 The specified event binding 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.



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:

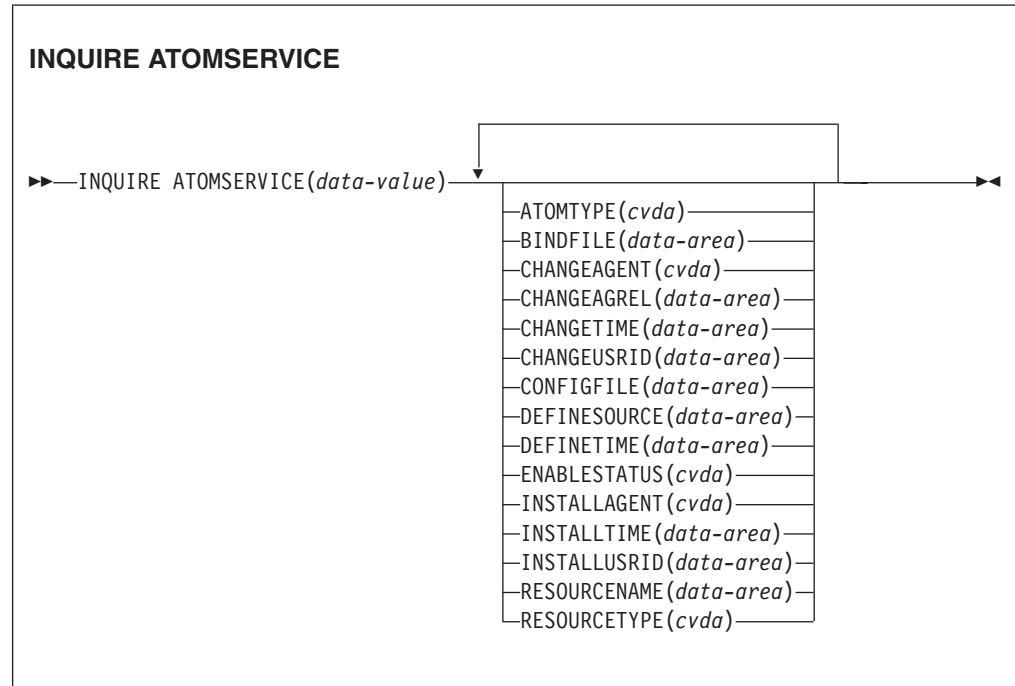
- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 values:

INQUIRE ATOMSERVICE

Retrieve information about ATOMSERVICE resources in the local system.



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 display the resource signature fields, which you can use to manage resources by capturing details of when the resource is defined, installed, and last changed. For more information, see in the *CICS Resource Definition Guide*. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See the *CICS 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.

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 identifying 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.

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 source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT. For details, see the *CICS 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 identifying 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.

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.

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.

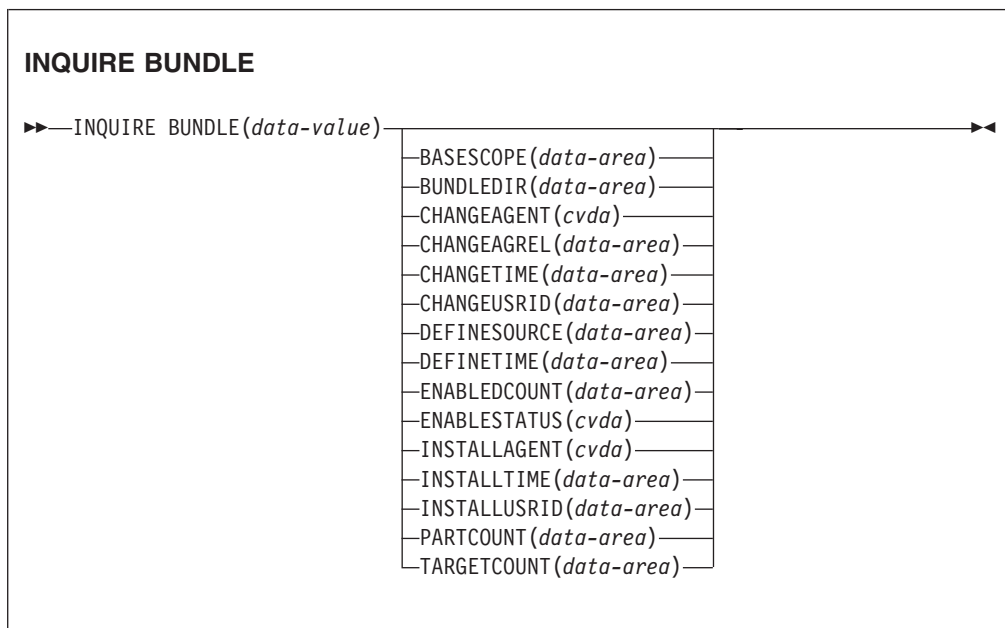
NOTFND

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 display the resource signature fields, which you can use to manage resources by capturing details of when the resource is defined, installed, and last changed. For more information, see in the *CICS Resource Definition Guide*. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See the *CICS 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 identifying 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.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

DEFINESOURCE (*data-area*)

Returns the source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT. For details, see the *CICS 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 identifying 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.

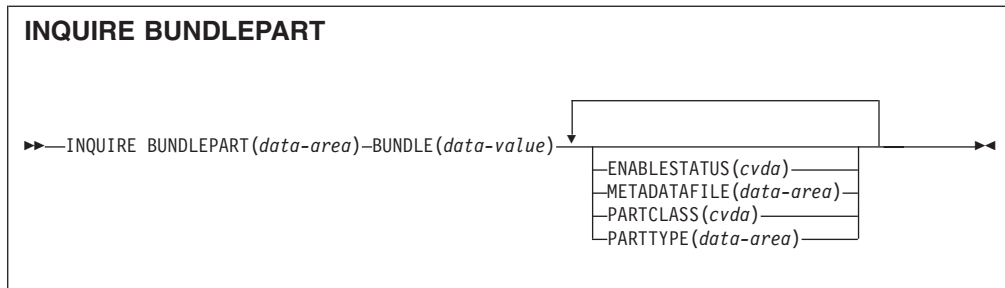
NOTFND

RESP2 values:

- 3 The BUNDLE cannot be found.

INQUIRE BUNDLEPART

Retrieve information about the resources that are contained in an installed BUNDLE resource.



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-area*)

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. Valid values are as follows:

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.

Eventname (*data-area*)

Specifies a 32-character data area to receive the associated business event name.

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.

INVREQ

- 4 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.

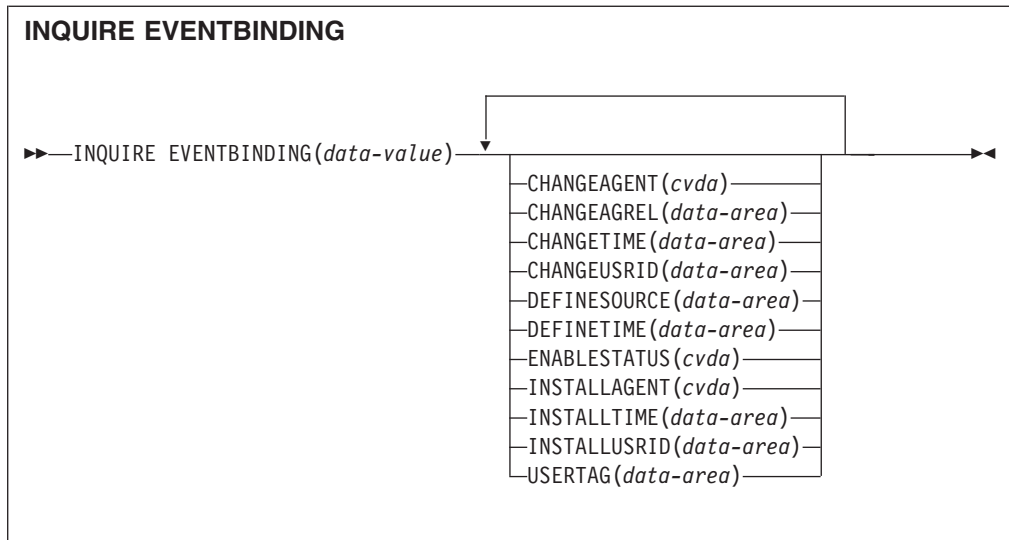
NOTFND

RESP2 values:

- 2 The specified capture specification cannot be found.
- 3 The specified event binding cannot be found.

INQUIRE EVENTBINDING

Retrieve information about a specified event binding.



Conditions: END, ILLOGIC, NOTAUTH, NOTFND

This command is threadsafe.

Description

The INQUIRE EVENTBINDING command returns information about a particular 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 display the resource signature fields, which you can use to manage resources by capturing details of when the resource is defined, installed, and last changed. For more information, see in the *CICS Resource Definition Guide*. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See the *CICS Resource Definition Guide* for detailed information about the content of the resource signature fields.

Options

CHANGEAGENT (*value*)

Displays a value identifying 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 in the *CICS 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.

EVENTBINDING (*data-value*)

Specifies the 32-character name of an event binding.

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.

INSTALLAGENT (*value*)

Displays a value identifying 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 (*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.
- 8 The specified event binding cannot be found.

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

▶▶—INQUIRE EVENTPROCESS—EPSTATUS(*cvda*)—————▶▶

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.

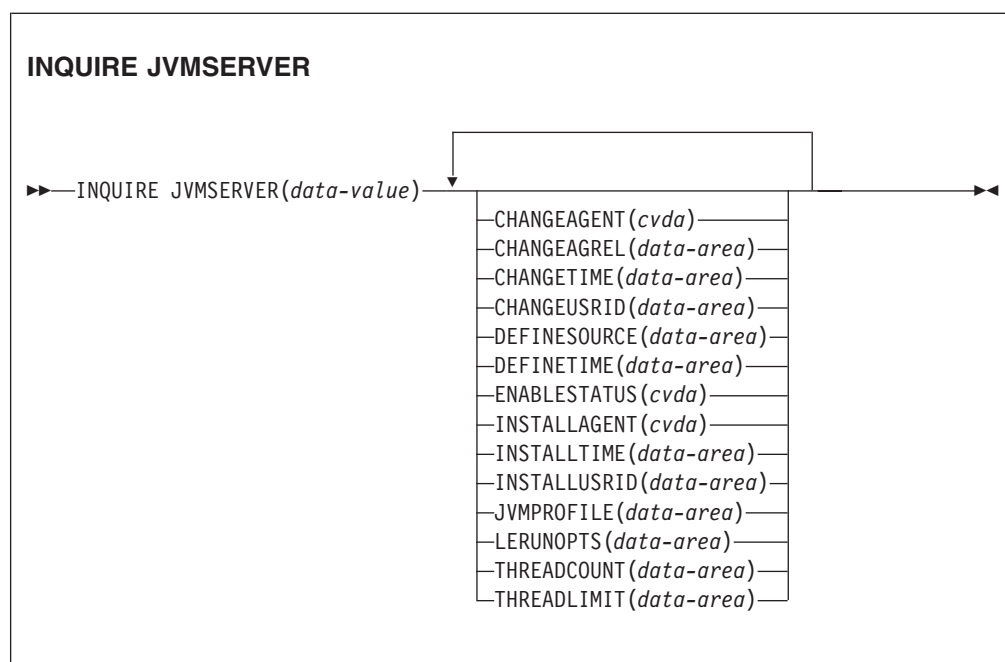
STOPPED
CICS is not processing events.

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 display the resource signature fields, which you can use to manage resources by capturing details of when the resource is defined, installed, and last changed. For more information, see in the *CICS Resource Definition Guide*. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See the *CICS Resource Definition Guide* for detailed information about the content of the resource signature fields.

Options

CHANGEAGENT(*cvda*)

Returns a CVDA value identifying 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.

CHANGEUSRID(*data-area*)

Returns the 8-character user ID that ran the change agent.

DEFINESOURCE(*data-area*)

Returns the source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT. For details, see the *CICS 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 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.

INSTALLAGENT (*cvda*)

Returns a CVDA value identifying 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.

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.

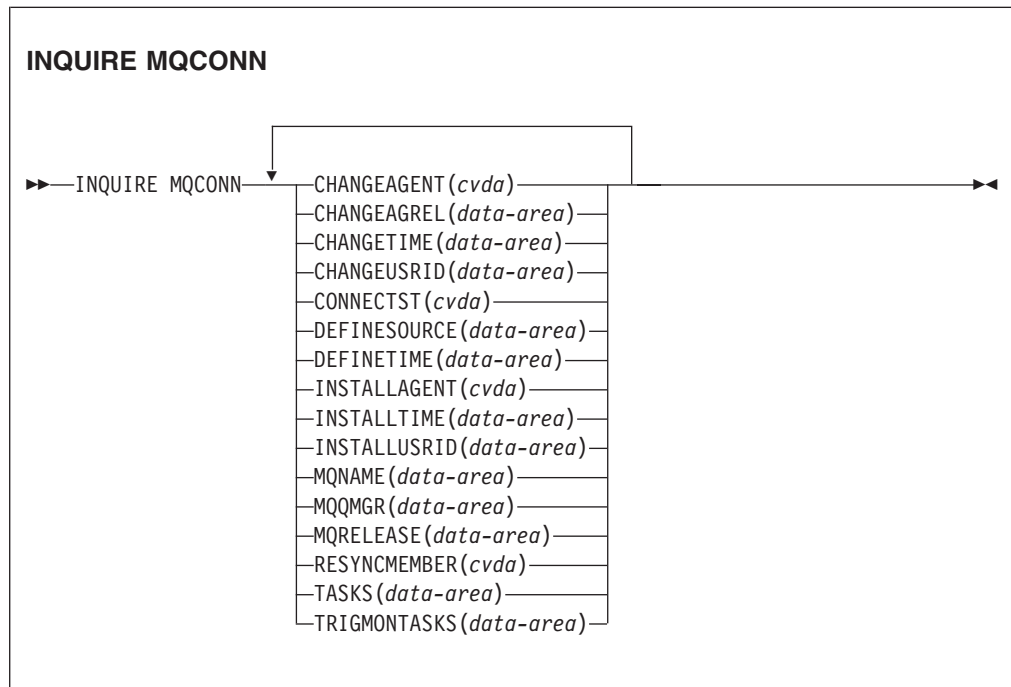
NOTFND

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.

**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 display the resource signature fields, which you can use to manage resources by capturing details of when the resource is defined, installed, and last changed. For more information, see in the *CICS Resource Definition Guide*. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See the *CICS Resource Definition Guide* for detailed information about the content of the resource signature fields.

Options

CHANGEAGENT(*cvda*)

Returns a CVDA value identifying 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.

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 source of the resource definition. The **DEFINESOURCE** value depends on the **CHANGEAGENT**. For details, see the *CICS 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 identifying 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, this option 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.

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.

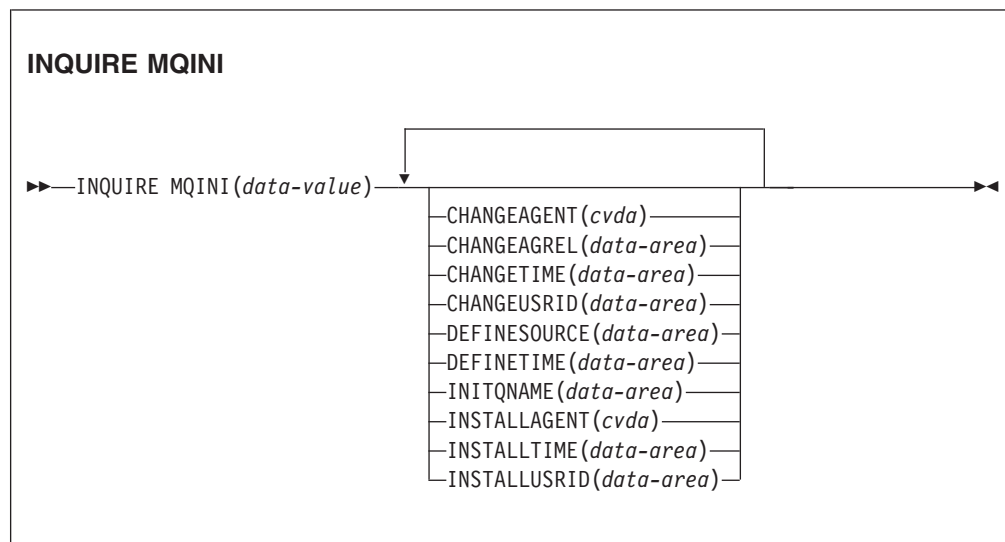
NOTAUTH

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.



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 and is dynamically created when you install an MQCONN resource 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, you must reinstall the MQCONN resource definition with an appropriate MQINI attribute.

The resource signature

You can use this command to display the resource signature fields, which you can use to manage resources by capturing details of when the resource is defined, installed, and last changed. For more information, see in the *CICS Resource Definition Guide*. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See the *CICS Resource Definition Guide* for detailed information about the content of the resource signature fields.

Options

CHANGEAGENT (*cvda*)

Returns a CVDA value identifying 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 a 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.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

DEFINESOURCE (*data-area*)

Returns the source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT. For details, see the *CICS 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 identifying 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 a 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

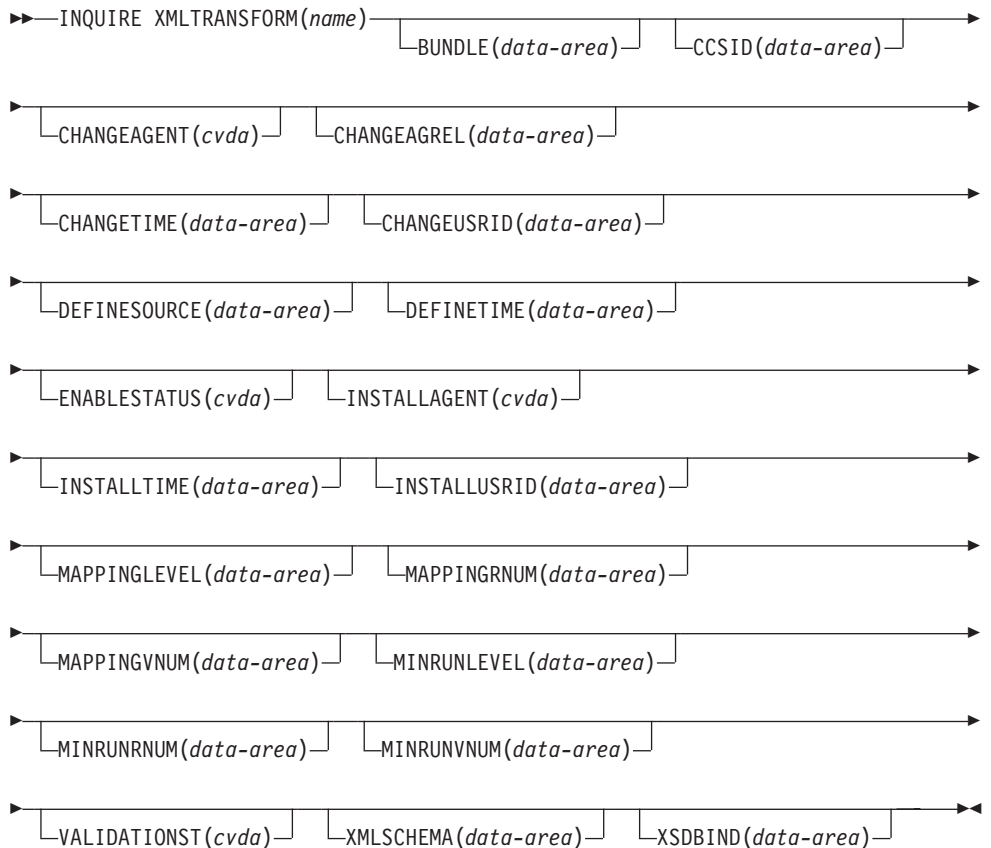
RESP2 values:

100 Command authorization failure.

INQUIRE XMLTRANSFORM

Use the **INQUIRE XMLTRANSFORM** command to retrieve information about an installed XMLTRANSFORM resource.

INQUIRE XMLTRANSFORM



Conditions: NOTAUTH, NOTFND

This command is threadsafe.

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 display the resource signature fields, which you can use to manage resources by capturing details of when the resource is defined, installed, and last changed. For more information, see in the *CICS Resource Definition Guide*. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE,

DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See the *CICS 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 identifying 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.

CHANGEUSRID(*data-area*)

Returns the 8-character user ID that ran the change agent.

DEFINESOURCE(*data-area*)

Returns the source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT. For details, see the *CICS 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 identifying 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.

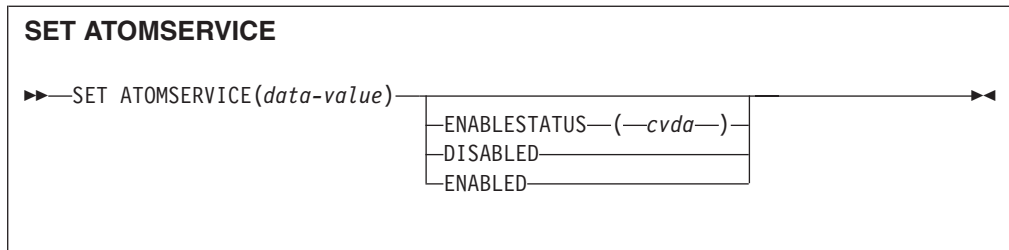
NOTFND

RESP2 values are:

- 3** The XMLTRANSFORM cannot be found.

SET ATOMSERVICE

Enables or disables an ATOMSERVICE definition.



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.

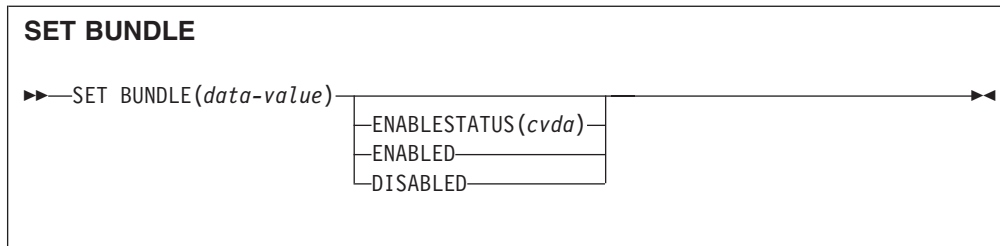
NOTFND

RESP2 values:

3 The ATOMSERVICE cannot be found.

SET BUNDLE

Change the status of an installed BUNDLE resource.



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.

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.

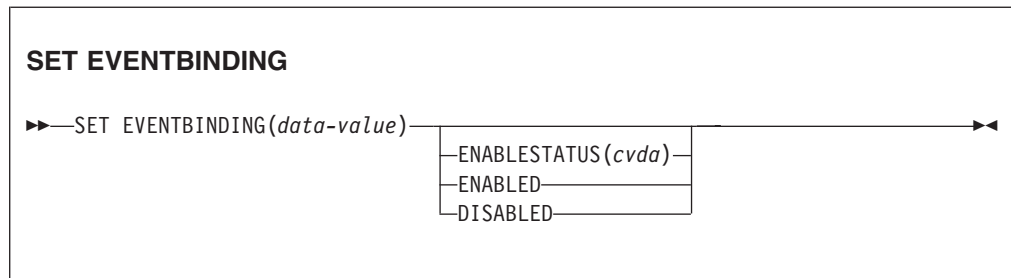
NOTFND

RESP2 values:

- 3 The BUNDLE was not found.

SET EVENTBINDING

Set the status of a specified event binding to enabled or disabled.



Conditions: INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

Use the SET EVENTBINDING command to change the status of a particular event binding.

Options

EVENTBINDING (*data-value*)

Specifies the 32-character name of a event binding.

ENABLESTATUS (*cvda*)

CVDA values are as follows:

ENABLED

The event binding is enabled. Capture of events matching capture specifications within this event binding starts immediately.

DISABLED

The event binding is disabled. Capture of events matching capture specifications within this event binding is stopped immediately. Any events already captured will be 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.

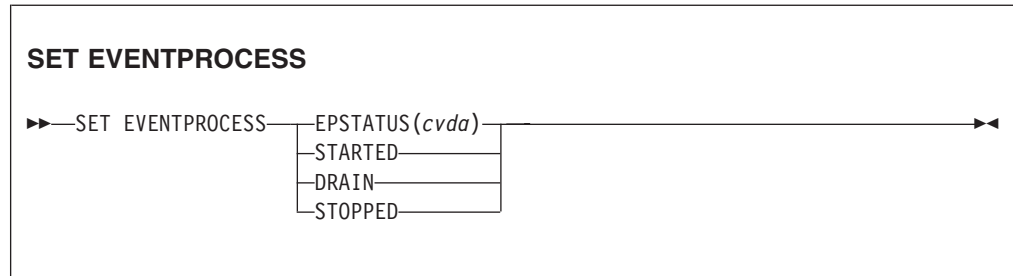
NOTFND

RESP2 values are:

3 The specified event binding cannot be found.

SET EVENTPROCESS

Set the status of event processing.



Conditions: INVREQ, NOTAUTH

This command is threadsafe.

Description

Use the SET EVENTPROCESS command to change the status of event processing.

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.

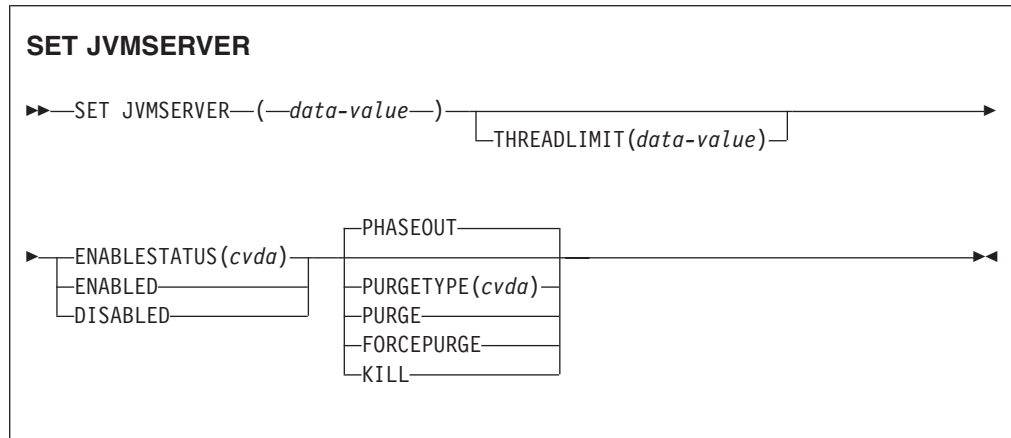
NOTAUTH

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 JVMSERVER enables and disables a Java server runtime environment that is used by CICS to perform multithreaded processing by calling the same JVM repeatedly. Use the **SET JVMSERVER** command to change the status of the JVMSERVER resource.

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 where some tasks could not be purged successfully. 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 where some tasks could not be purged successfully. 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.

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.

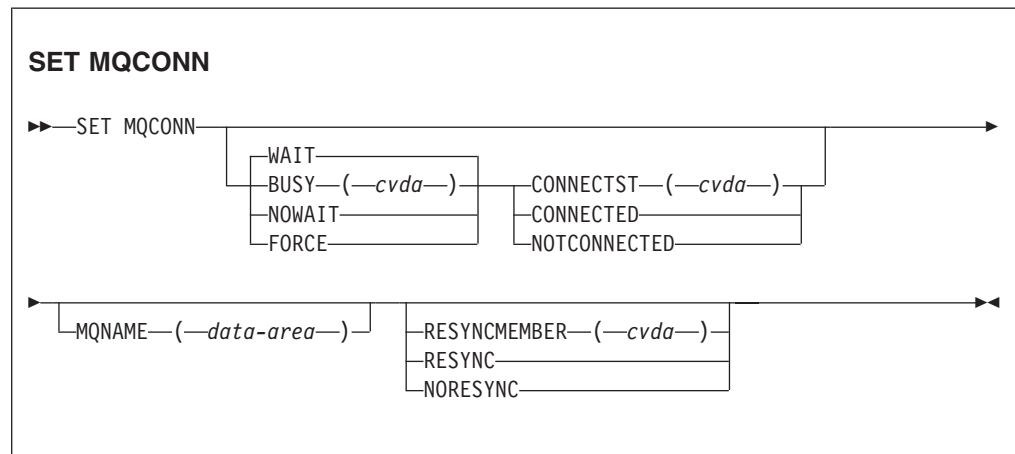
NOTFND

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.



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-MQ connection **NOTCONNECTED**. If you specify **CONNECTED**, **BUSY** is ignored. The CVDA values are as follows:

FORCE

Any CICS transactions currently 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-MQ connection. This action has the same effect as issuing a **CKQC START** command to start the CICS-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-MQ adapter is in connecting state and will connect to WebSphere MQ as soon as the requested queue manager becomes active.

NOTCONNECTED

Stops the CICS-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-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 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.

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.

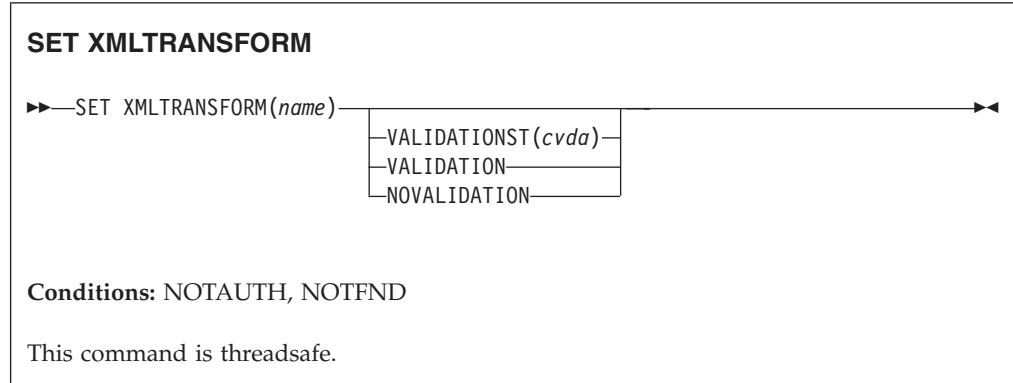
INVREQ

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.

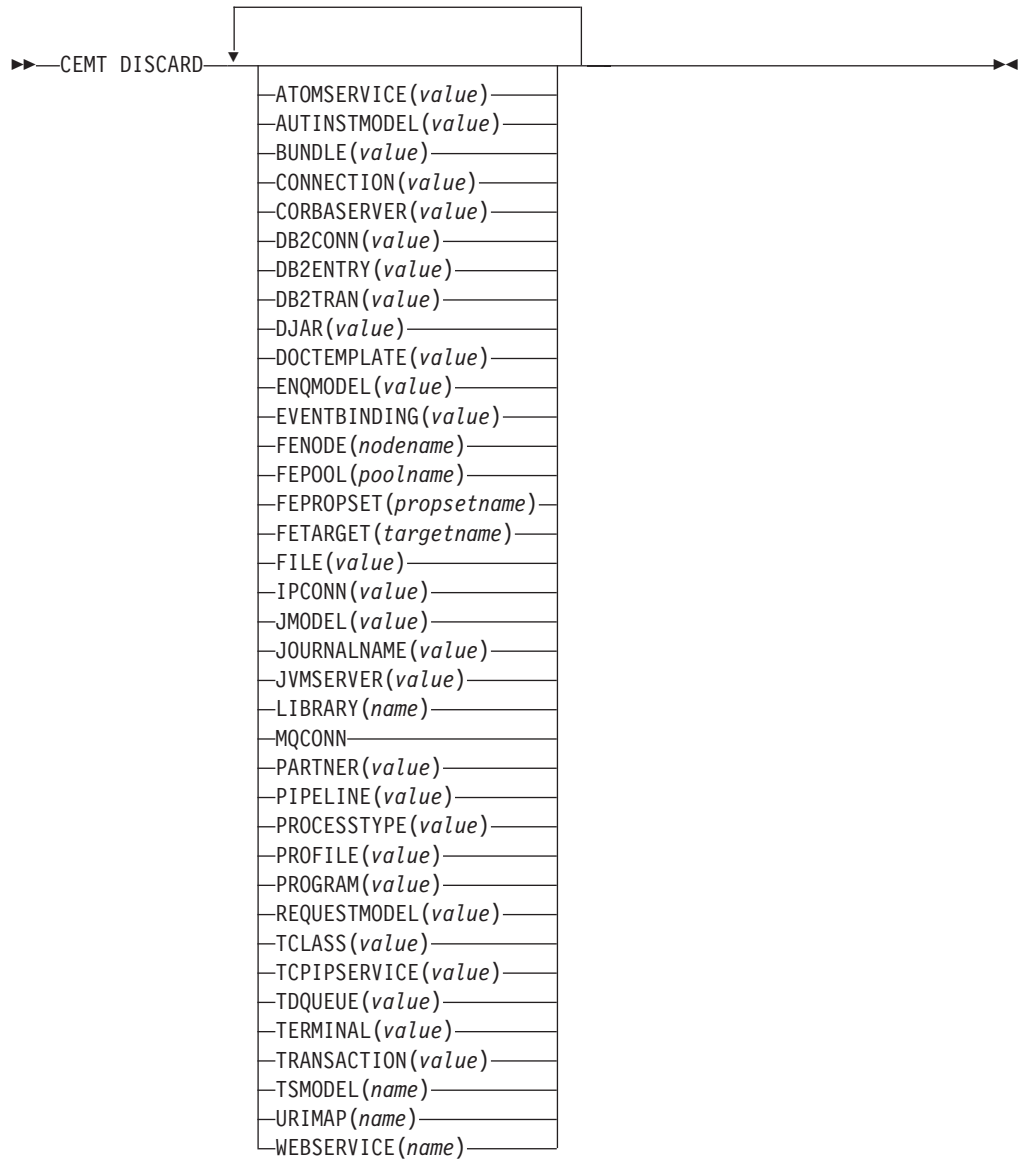
Appendix D. New CEMT commands

CICS Transaction Server for z/OS, Version 4 Release 1 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 autoinstall terminal model (AITM) table 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 VTAM 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.

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.

FEPool (*poolname*)

The name of the FEPI pool to be discarded.

FEPool (*poolname*)

The name of the FEPI pool to be discarded.

FEPropset (*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 file control table 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 installed journal model table 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 journal table 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 eight 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 side information table 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 profile table (PFT) 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 processing program table (PPT) 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 program control table (PCT) 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 DELETEDQ 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 syncpoint is taken after the DELETEDQ 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 terminal control table (TCT) and the CICS catalog.

Only terminals that use VTAM 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*)

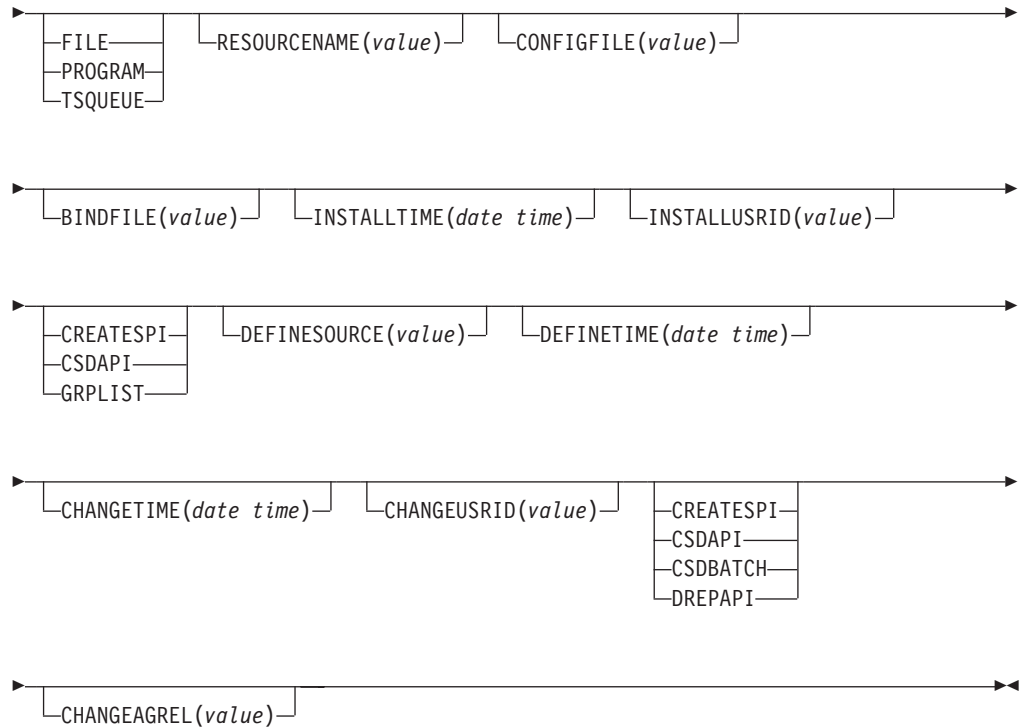
Specifies the name of the transaction that is to be removed. The name can be up to 8 characters long. The **DISCARD TRANSACTION** command removes the installed name of a transaction from the CICS program control table (PCT) and the CICS catalog.

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



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, so, if Atomtype is Service or Category, this field is blank.

CHANGEAGENT(*value*)

Displays a value identifying 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 in the *CICS 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 identifying 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.

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.

CEMT INQUIRE BUNDLE

Retrieve information about a BUNDLE resource.

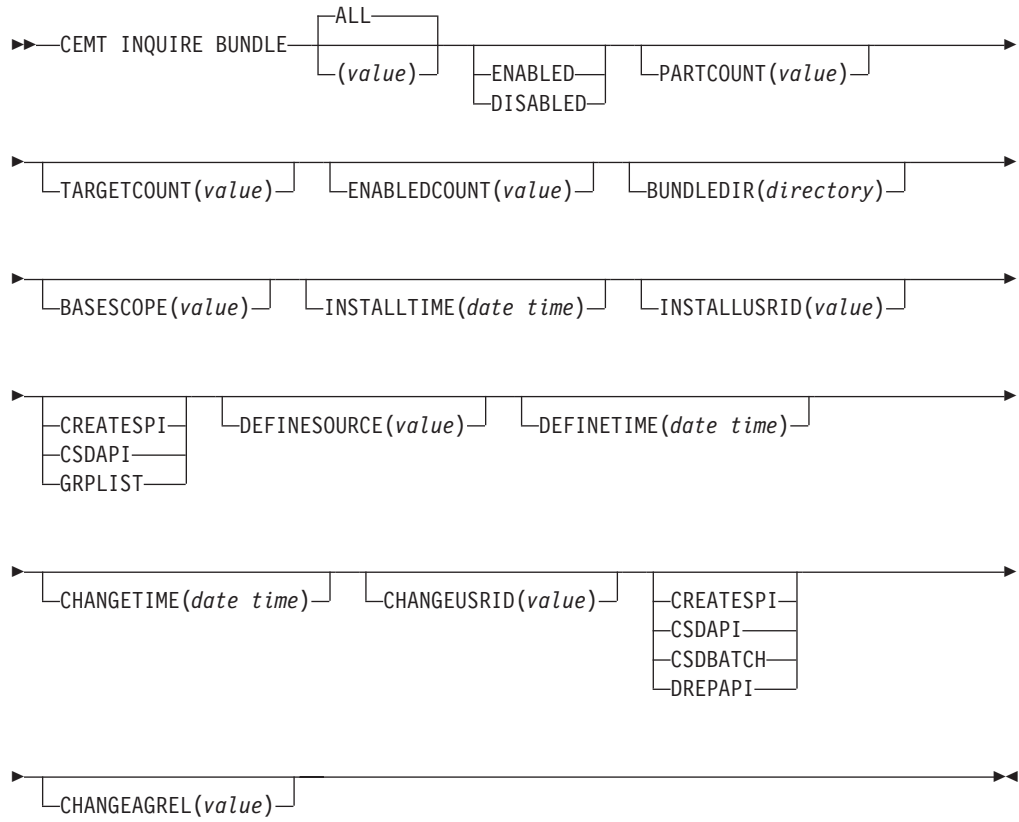
The resource signature

You can use this command to display the resource signature fields, which you can use to manage resources by capturing details of when the resource is defined, installed, and last changed. For more information, see in the *CICS Resource Definition Guide*. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See the *CICS 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 identifying 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 in the *CICS 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 identifying 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 EVENTBINDING

Retrieve information about an event binding.

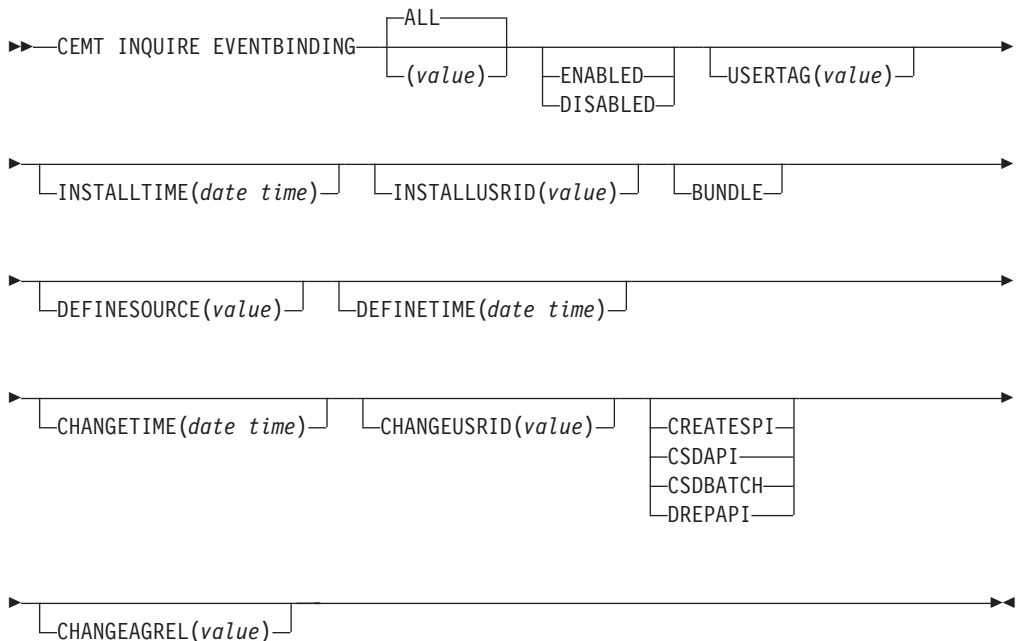
The resource signature

You can use this command to display the resource signature fields, which you can use to manage resources by capturing details of when the resource is defined, installed, and last changed. For more information, see in the *CICS Resource Definition Guide*. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See the *CICS Resource Definition Guide* for detailed information about the content of the resource signature fields.

Input

Press the Clear key to clear the screen then enter CEMT INQUIRE EVENTBINDING to obtain information about the event binding.

CEMT INQUIRE EVENTBINDING



Displayed fields

CHANGEAGENT(*value*)

Displays a value identifying 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 in the *CICS 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:

ENABLED

The event binding is enabled.

DISABLED

The event binding is disabled.

INSTALLAGENT(*value*)

Displays a value identifying 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 usertag of the event binding.

CEMT INQUIRE EVENTPROCESS

Retrieve the status of event processing.

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:

STARTED

event processing is started.

DRAINING

event processing is draining.

STOPPED

event processing is stopped.

CEMT INQUIRE JVMSERVER

Retrieve information about the status of a JVM server runtime environment.

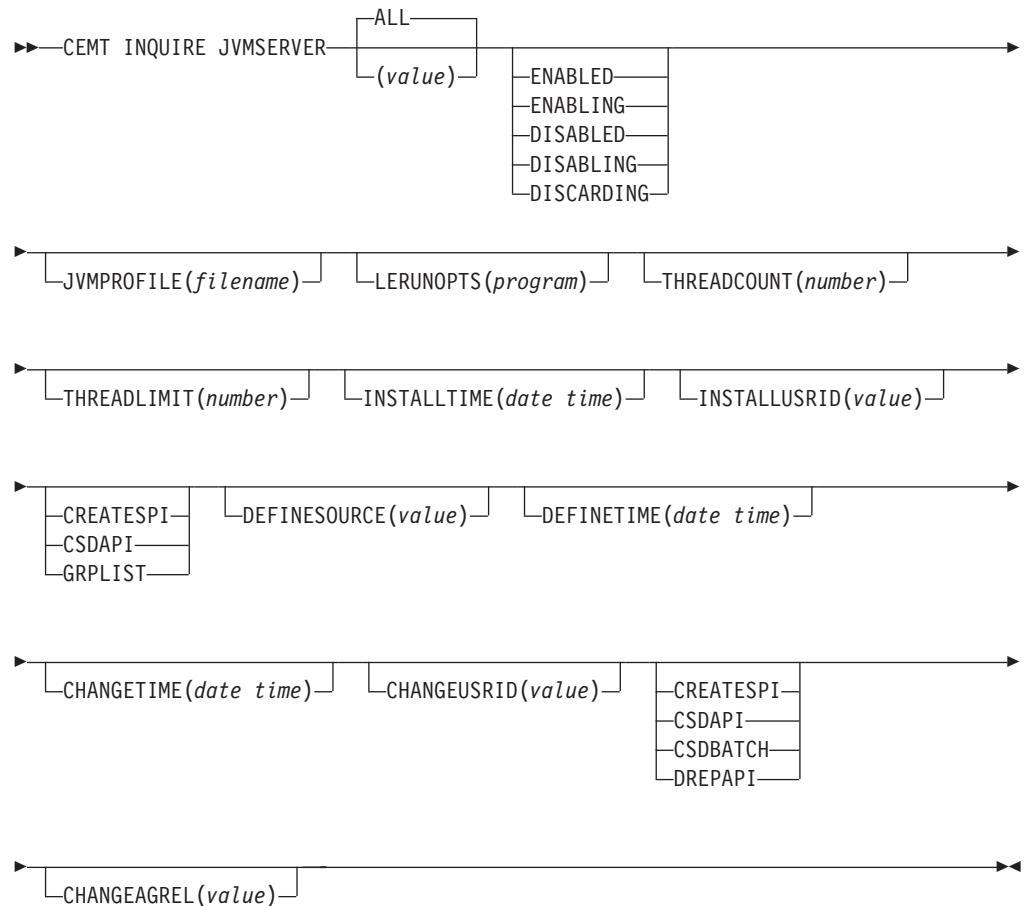
The resource signature

You can use this command to display the resource signature fields, which you can use to manage resources by capturing details of when the resource is defined, installed, and last changed. For more information, see in the *CICS Resource Definition Guide*. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See the *CICS 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 identifying 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 in the *CICS 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 identifying 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.

STATUS(*value*)

Displays the overall status of the JVM server runtime environment. The possible values are as follows:

ENABLED

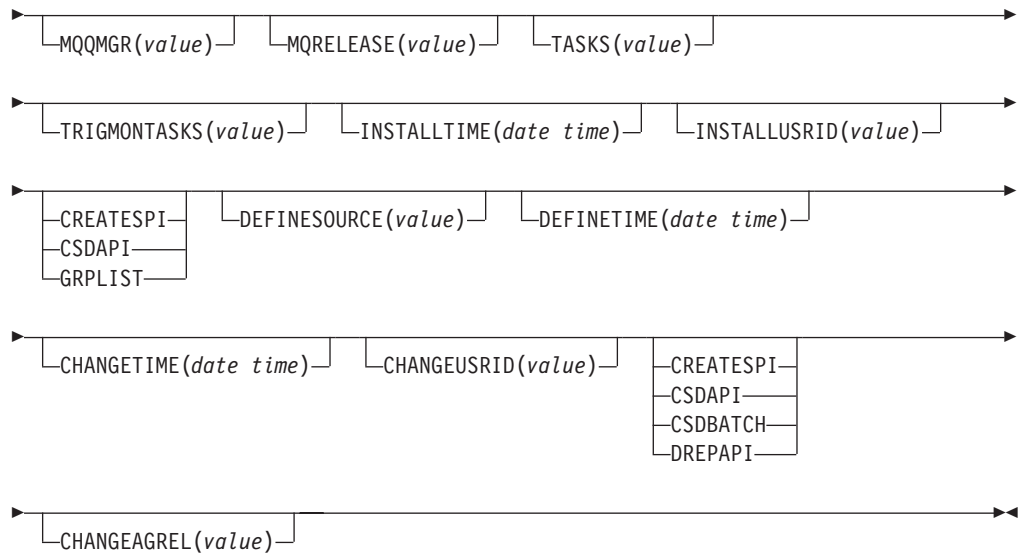
The JVM server is enabled and can perform processing.

ENABLING

The JVM server is starting.

DISABLED

The JVM server is disabled and new requests cannot be processed in the runtime environment.



Displayed fields

CHANGEAGENT(*value*)

Displays a value identifying 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 in the *CICS 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 identifying 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 - 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.

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. It 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.

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 about the outstanding units of work.

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.

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

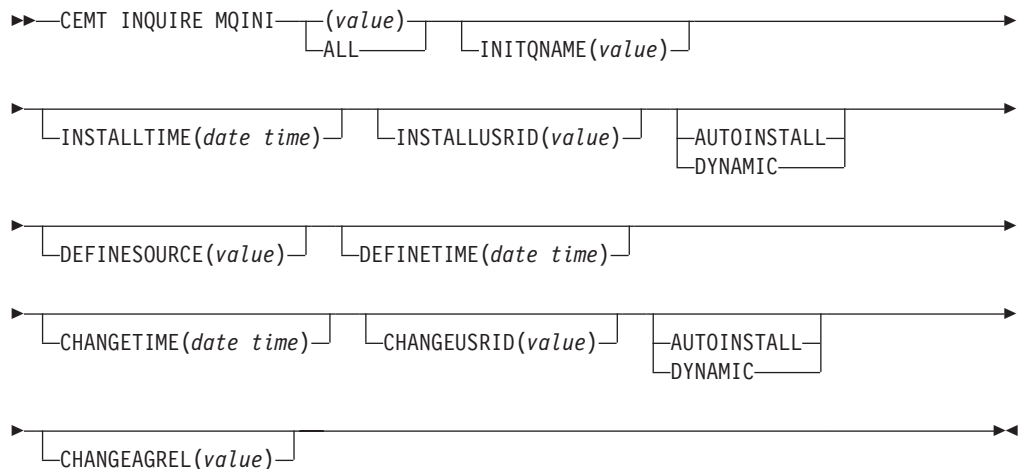
You can use this command to display the resource signature fields, which you can use to manage resources by capturing details of when the resource is defined, installed, and last changed. For more information, see in the *CICS Resource Definition Guide*. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See the

CICS 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. The minimum abbreviation is CEMT I MQI. You obtain a display that lists the current status. You can then 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 identifying 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 a 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 in the *CICS 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 identifying 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:

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 a **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.

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

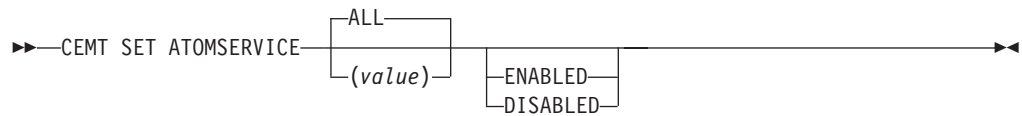
Press the Clear key to clear the screen. You can start this transaction in two ways:

- Type **CEMT SET ATOMSERVICE** (the minimum abbreviation is **CEMT S AT0**) followed by one or more **ATOMSERVICE** identifiers or **ALL**. You obtain a display that lists the current status, similar to that obtained by **CEMT INQUIRE ATOMSERVICE**. You can then tab to the highlighted or blank fields and overwrite them with the required values.
- Type **CEMT SET ATOMSERVICE (CEMT S AT0)** followed by one or more **ATOMSERVICE** identifiers or **ALL**, followed by the attribute setting that you

want to change. For example, `cent 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 - 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.

If the optional attributes are omitted, the command has no effect.

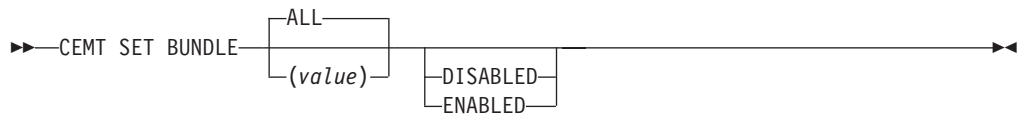
Syntax

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 EVENTBINDING

Enable or disable an event binding.

Description

CEMT SET EVENTBINDING



Options

(value)

The 32-character name of an event binding.

DISABLED

specifies that the event binding is to be disabled.

ENABLED

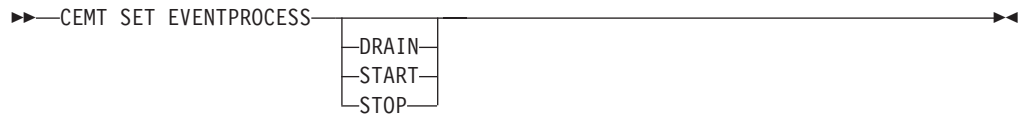
specifies that the event binding is to be enabled. You can enable an event binding that is in any state except DISABLING or DISCARDING.

CEMT SET EVENTPROCESS

Change the status of event processing.

Description

CEMT SET EVENTPROCESS



Options

DRAIN

Set the status of event processing to draining. 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 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.

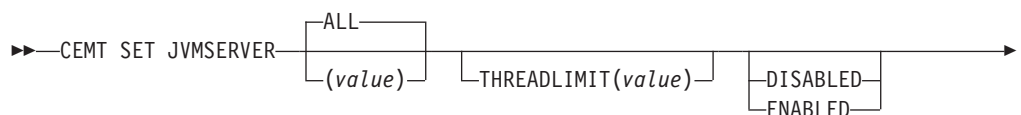
Syntax

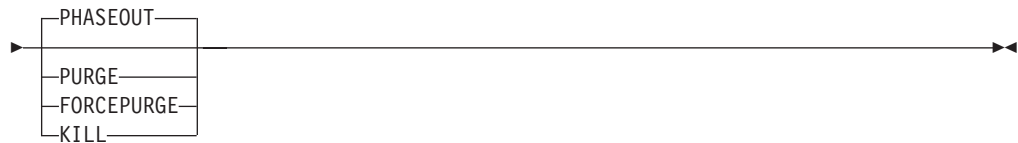
Press the Clear key to clear the screen. You can start this transaction in two ways:

- 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

value

Specifies the 1 - 8 character name of the JVMSERVER resource.

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 the JVM profile and Language Environment runtime options are used to update the JVM server. If you enable a JVMSERVER resource after it has been in the DISABLED state, the JVM server 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 JVMSERVER resources, CICS adjusts the value accordingly.

CEMT SET MQCONN

Change information about the attributes of the connection between CICS and WebSphere MQ, and start or stop the connection.

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.

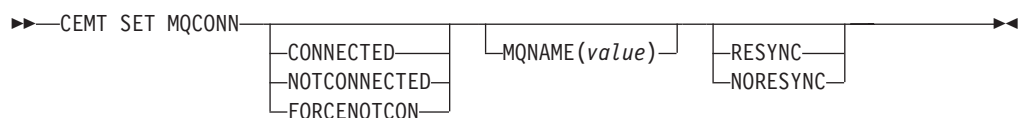
Syntax

Press the Clear key to clear the screen. You can start this transaction in two ways:

- Type CEMT SET MQCONN. The minimum abbreviation is CEMT S MQ. You obtain a display that lists the current status, similar to that obtained by CEMT INQUIRE MQCONN. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type CEMT SET MQCONN (CEMT S MQC) 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 need to 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.

Appendix E. New global user exits

CICS Transaction Server for z/OS, Version 4 Release 1 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. 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.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other product and service names might be trademarks of IBM or other companies.

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 1
What's New

Publication No. GC34-6994-03

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



GC34-6994-03

