Tivoli® Discovery Library Adapter for z/OS
Version 3 Release 1

User's Guide and Reference

IBM
Tivoli® Discovery Library Adapter for z/OS
Version 3 Release 1

User's Guide and Reference
# Contents

## Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>vii</td>
</tr>
</tbody>
</table>

## Chapter 1. Introduction

Product overview .................................. 1  
- z/OS DLA discovery output to XML books ............... 1  
- TADDM WebSphere Sensor seed file discovery .......... 2  
- TADDM stitching .................................... 3  
- Additional discovery options .......................... 3  
- zSeries resources in TADDM and TBSM ................. 4  
- z/OS DLA discovery detail ............................ 7  
- New in this release .................................. 9  
- Supported software .................................. 9  
- Supported hardware ................................... 10  
- Related documents ................................... 10  

## Chapter 2. Working with the DLA for z/OS

- Running the z/OS DLA ................................ 11 
  - IZDJRUN batch job ................................ 11 
  - IZDPDISC procedure ................................ 12 
  - IZDJFTP batch job ................................ 12 
  - IZDPFTP procedure ................................ 13 
- Output books ....................................... 13 
  - Configuring TADDM to display topology graphs of highly populated systems .... 14 
  - Specifying discovered resources by using sample configuration members .... 15 
  - ZOSBASE discovery ................................ 15 
  - ZOSTASK discovery ................................ 16 
  - DB2 subsystem discovery ............................ 17 
  - IMS subsystem discovery ............................ 19 
  - MQ subsystem discovery ............................ 21 
  - CICS region discovery .............................. 22 
  - WebSphere Application Server discovery .......... 24 
  - ZOSALL book ...................................... 27 
- FTP transfer to the Discovery Library File Store .... 27 
  - Target DLFS FTP server ............................. 28 
  - Discovery Library file naming standards .......... 28 
  - Checksums to detect if members must be transferred .... 29 
  - Automating discovery .............................. 30 
- What did you discover? - Tips and Tools .............. 30 
  - IdML XML Browser ................................ 30 
- Loading the XML files into Tivoli TADDM .............. 31 

## Chapter 3. Reference information

- Creating and managing custom servers .................. 33 
  - Identifying unknown server patterns ................ 33 
  - Adding custom servers .............................. 34 
  - Adding this custom server resource to a business application .... 34 
  - Associate an icon with your custom server ........ 35 
- z/OS DLA discovery customization ...................... 37 
  - SET directive .................................... 38 
  - DISC directive ................................... 39 
  - FILT directive ................................... 43 
  - CUSTSRVR directive ............................... 52 
  - CRITERIA directive ............................... 52 
  - FILES directive ................................... 53 
  - BOOK directive ................................... 55 

© Copyright IBM Corp. 2008, 2010
Contents v

Appendix D. Samples provided in hlq.SIZDSAMP .......................... 171

Glossary ............................................................................ 173

Notices .............................................................................. 175

Index ................................................................................. 177
## Figures

1. Discovery overview of Discovery Library Adapter for z/OS .......................... 2
2. DLA that discover zSeries resources .............. 4
3. zSeries resources discovered by TADDM ......... 5
4. Comparing changes in zSeries resources and configuration settings across similar resources over time ........................................ 6
5. zSeries topology and resource status at TBSM ...... 7
6. Executable files that run z/OS DLA and send books by using FTP .................. 12
7. Requested graph exceeded the number of allowed nodes error message ................ 14
8. ZOSBASE—zSeries hardware and z/OS details 16
9. ZOSTASK—AddressSpace details and relationships .................................. 17
10. DB2 Subsystem details and relationships—No Data Sharing ......................... 18
11. DB2 Subsystem details and relationships—Data Sharing .............................. 19
12. IMS Subsystem details and relationships .................................................. 20
13. MQ Subsystem details and relationships .................................................. 22
14. CICS region details and relationships ..................................................... 23
15. TADDM WebSphere Sensor seed file ....................................................... 24
16. WebSphere Application Server details and relationships ........................... 26
17. ZOSALL—Everything that is active in one book ...................................... 27
18. The Discovery Library IdML XML File Browser ....................................... 31
19. Custom server business application ......................................................... 35
20. Custom server dialog box ................................................................. 36
21. Custom server application topology and custom server icon ....................... 37
Chapter 1. Introduction

Use the IBM® Tivoli® Discovery Library Adapter for z/OS® (z/OS DLA) to discover IBM zSeries resources (such as zSeries hardware, z/OS systems, TCP/IP connections, and z/OS system applications). The product provides the discovered resources in output files (known as books) that conform to the Common Data Model (CDM). These books can be displayed by various product consoles that provide change and configuration management, and business process management. These consoles include IBM Tivoli Application Dependency Discovery Manager (TADDM), IBM Tivoli Change and Configuration Management (CCMDB), and IBM Tivoli Business Service Manager (TBSM).

Product overview

The IBM Tivoli Discovery Library Adapter for z/OS discovers IBM zSeries hardware and software resources. This product stores the discovered resources in books. These books consist of XML tags and values that define the zSeries resources in conformance with the IBM Common Data Model (CDM) classes, attributes, and relationships. The z/OS DLA books can be loaded into the database of several different IBM products that make use of the discovered resources (and understand CDM). These products can then display the resources in their respective consoles.

The IBM products that make use of the discovered resources include IBM Tivoli Application Dependency Discovery Manager (TADDM), IBM Tivoli Business Service Manager (TBSM), and IBM Tivoli Change and Configuration Management Database (CCMDB). The Discovery Library Adapter for z/OS product includes the ability to use FTP to copy the books to a file server, where they can be loaded into a product that uses the books.

The products that use the DLA for z/OS can load books from various discovery library adapters (DLAs) that conform to CDM. These products also provide local dynamic discovery. Dynamic discovery from TADDM is called a sensor. TADDM refers to any resources provided in DLA books, or by TADDM sensors, as component items (CIs). Most discovery of hardware and software for zSeries systems is through the use of a discovery library adapter. Most discovery of hardware and software for distributed systems is by dynamic discovery, such as TADDM sensors.
The principal mechanism that DLA for z/OS uses for discovery is to query APIs and registers, and then write the resulting output to XML books (also known as IdML books) that conform to CDM. DLA for z/OS can provide the following types of resources in books:

1. zSeries machine information, including serial number, processing capacity, LPAR, and VM guest
2. zSeries virtualization, including Sysplex, PR/SM (processor resource/systems manager) LPARs, ZcouplingFacility, and VMGuest
3. z/OS information, including PARMLIB active member contents, LNKLIST, IODF data set, system exits, APF list, and IPL information
4. Storage hardware (DASD) and volume information
5. IBM CICS information, including Regions, version number, transactions, programs, files, and System Initialization Table (SIT) details
6. IBM DB2 for z/OS information, including version number, databases and tables pages
7. IBM IMS information, including version, transactions, programs, and databases
8. IBM MQSeries for z/OS information, including version number, ports and connections, Sender and Receiver channels, Queues, NameLists, Processes, AuthInfo, buffer pools, PageSets, and MQStorage
9. IBM WebSphere Application Server for z/OS information, including version number, cell, node, and configuration files
10. Address space information, including servers, version, DD allocations, address space type, program name, and program parameters
11. TCP/IP information, including interfaces, IP addresses, fully qualified domain names, ports, bind addresses between applications, and IPv6 support
12. Ability to extend discovery to additional z/OS software (IBM or nonIBM software) by creating user-defined custom servers

Some DLA for z/OS discovery is enhanced by TADDM stitching and TADDM sensor seed file discovery.

**TADDM WebSphere Sensor seed file discovery**

z/OS DLA can discover higher-level WebSphere resources and include them in books. Lower-level WebSphere discovery can be accomplished by using the WebSphere Application Server seed utility for the z/OS DLA. This utility creates an XML seed file by scanning a z/OS DLA book. This seed file is loaded into TADDM and used to prime the TADDM WebSphere Sensor, which can discover low-level WebSphere resources on z/OS.

**TADDM stitching**

By analyzing data loaded from multiple sources, and potentially "stitching" it to other data, the TADDM server effectively discovers the target resource and its relationships. Stitching allows TADDM to discover additional resource relationships that might not have been discoverable by z/OS DLA alone.

The z/OS DLA subsystems that currently make use of TADDM stitching include MQ, CICS, DB2, and IMS.

**Additional discovery options**

Several other DLAs and many TADDM sensors all do discovery, and then create CIs that conform to CDM. Because of common data model definitions, discovered resources can be integrated at TADDM, CCMDB, or TBSM to show more than any one discovery application can interpret. See [Figure 2 on page 4](#) for an overview of discovered resources.
Determining which DLAs you need

To discover resources on z/OS, start with the z/OS DLA and add other DLAs as needed. The z/OS DLA provides the best discovery of z/OS systems, zSeries virtualization, and zSeries hardware.

The z/OS DLA provides good discovery of applications on z/OS (such as MQSeries, IMS, and CICS). If you need more, you might want to use the application-specific DLAs. Most of these DLAs are built to discover the following additional detail that the z/OS DLA did not discover:

- CICS DLA
- IBM System Automation for z/OS DLA
- IBM NetView for z/OS DLA
- IBM Tivoli Monitoring DLA
- Linux for System z discovery can now be accomplished with the TADDM Linux sensor

TADDM Sensor discovery


TADDM defines Level 1, 2, and 3 Sensor discovery levels. These levels are based on Sensor groups and Sensor authorization, which do not apply to DLAs.
Product overview

Similar to TADDM L1 Sensor discovery, unmodified z/OS DLA provides high-level resources with minimal authorization required.

Lower-level resource discovery is filtered out by default. You can extend discovery to include lower-level resources (CIs) by completing the following steps:
1. Remove some or all discovery filters in the z/OS DLA IZDCDEF customization sample.
2. Provide additional authorization if required. For example, low-authority users can typically discover DB2 subsystems (high-level), but you might need to configure specific ACF authorization to also discover DB2 databases.
3. In WebSphere on z/OS resource discovery, use the WebSphere Application Server seed utility with the TADDM Websphere Sensor.

If you removed all z/OS DLA filters, you have the equivalent of TADDM L3 discovery, that is, you discover everything that z/OS DLA can discover. But you pay a performance penalty at bulk load and screen refresh of consuming consoles if you discover and load everything.

zSeries resources in TADDM and TBSM

You can use TADDM to display zSeries resources in layout diagrams and in the detail window. See

Figure 3. zSeries resources discovered by TADDM

A powerful use of TADDM and CCMDB is to compare changes in zSeries resources and configuration settings across similar resources over time. See Figure 4 on page 6.
With TBSM, you can see zSeries topology layouts and resource status by using z/OS DLA books and Event Pump for z/OS events.
z/OS DLA discovery detail

z/OS DLA starts by discovering the zSeries hardware and the instance of the z/OS where it is installed. The z/OS DLA can discover the following zSeries virtualization:

- Sysplex
- z/VM Guest
- PR/SM partitioning of the CPC

This resulting discovery information is produced as output that is sent to the ZOSBASE book.

Next, z/OS DLA discovers all z/OS address spaces and examines them to see if it recognizes any subsystems. You can do the same at your z/OS keyboard by going to TSO SDSF Display Active (=SDSF.DA) to see if an application (WAS, DB2 or NetView) is running. z/OS DLA also examines TCP/IP connections to discover network connectivity and assess which subsystems might be interconnected. This discovery information is sent as output to the ZOSTASK book.

Every z/OS DLA discovery produces one ZOSBASE book and one ZOSTASK book. Depending on what is installed on the LPAR, you can get dozens of additional books for DB2, MQ, CICS, and other subsystems.

When z/OS DLA recognizes a subsystem (ex. A DB2 subsystem) that is up and running, it creates 1 book with the name of the master or control address space for that subsystem.

In one system (one LPAR), there can be an address space named D81DMSTR. The z/OS DLA creates a D81DMSTR book to represent that instance of DB2 (v8.1). You might also have a D91AMSTR book for that LPAR. A DB2 instance (one book) can include several lower-level DB2 tablespaces and databases.
Note: The unmodified z/OS DLA can discover zSeries, zSeries virtualization, z/OS systems, and z/OS application subsystems (such as CICS Region C88RLB5). Discovering lower-level detail requires that you change the default filters.

IMPORTANT: Do not assume that in a z/OS DLA book you will automatically see all the resources that can be discovered. You can review filter settings in the IBM Tivoli Discovery Library Adapter for z/OS User’s Guide and Reference (this book), or look at the sample IZDCDEF customization member.

The z/OS DLA filters out lower-level detail for the following reasons:

• One product can take much longer than another to load in all the discovered resources.
• A slow screen-refresh at the product can cause confusion to operators.
• A highly populated consuming product database can cause slow screen-refresh and overload operators with thousands of resources and lines interconnected on-screen.

One useful way to see zSeries resource detail without incurring the burden of loading thousands of low-level resources, is to use z/OS DLA support for CDM ZReportFiles. For example, z/OS DLA can create ZReportFiles that list all CICSTransactions, all DB2 databases and all z/OS PARMLIB definitions.

These text files contain most of the detail you might see in 2040 individual CICS Transaction objects. You can view these reports in TADDM detail, but you cannot see the individual transactions as icons in a topology layout or as rows in a detail window table. You can run TADDM or CCMDB Change Configuration against ZReportFiles to see what has changed since a prior discovery.

Stitching detail

MQ stitching can connect MQ resources even if the channels were reported from discovery of different zSeries systems in the same sysplex. TCP/IP stitching can connect resources of several subsystems that use TCP/IP Sockets, such as CICS, DB2, IMS, and MQ.

Review the following information about the stitching process:

• MQ: TADDM can use the information provided by z/OS DLA to stitch (connect) MQSenderChannels to MQReceiverChannels.
• TCP/IP: TADDM compares TCP/IP Socket Addresses (CDM BindAddress objects) provided by z/OS DLA on one z/OS system to other socket addresses. The additional sources for socket addresses can include z/OS DLA running on other z/OS systems in the same sysplex, other DLAs, or TADDM sensors.
• TCP/IP stitching can enable TADDM to show connections (lines connecting nodes in a topology layout) between software applications reported from different systems or different discovery agents. TCP/IP is the primary transport vehicle for communications between applications. TCP/IP stitching is not just for TCP/IP resources. Any application that uses TCP/IP Sockets can be enhanced by TADDM TCP/IP Stitching. For z/OS DLA, the following list of applications is included:
  - CICS – MQ
  - CICS – DB2
  - CICS – IMS
  - IMS – DB2
Product overview

TADDM stitching is explained in the "How TADDM Works" article at http://www.ibm.com/developerworks/wikis/display/tiviaddm/A+Flexible+Approach+to+Discovery#AFlexibleApproachtDiscovery-HowTADDMWorks.

New in this release

The DLA on z/OS V3.1 includes the following updates and enhancements:

- Two new classes have been added: ZCouplingFacility and ZReportFile.
- The SysplexGroup class now represents an Application Server cluster.
- Enhanced zSeries hardware and LPAR discovery through PR/SM (processor resource and system manager):
  - Processor Resource/Systems Management (PR/SM) discovery
  - All LPARs on the CPC (central processor complex)
  - Discovery of the ZIIP (z9 Integrated Information Processor), ZAAP (IBM System z Application Assist Processor), IFL (Integrated Facility for Linux), IFC, shared, and dedicated co-processors
  - Real and expanded memory on LPARs
- Enhanced MQSeries discovery that is equivalent to the distributed MQ sensor. You can now discover the following additional MQ resources and relationships:
  - Clusters and ClusterChannel relationships such as ClusterReceiver and ClusterSender
  - Channels and relationships such as Sender, Receiver, ServerConnection, and ClientConnection
  - Queues such as Local, Remote, Alias, and Model
  - AuthInfo
  - Listener
  - NameList
  - Process
  - Storage class
  - Pageset
  - Bufferpool
- Added feature to allow discovery of security packages other than RACF 4 (Resource Access Control Facility).
- Enhanced CICS support, including CICS V4.1 and additional filters:
  - Currency support for the CICS V4.1 Transaction Server
  - Filters to exclude discovery of CICS transactions and programs that have not been used
- Support for additional types of WebSphere servers.
- Enhanced sample customization file that includes all default and optional discovery filters.
- All z/OS DLA V2.3 APAR fixes are included.

Supported software

The DLA supports discovery for all releases of z/OS, CICS®, IMS™, MQ, DB2, and WebSphere® that are currently in service. See http://www.ibm.com/software/support/lifecycle/index_i.html for a list of releases that are currently in service.

This list includes the following releases:
Supported software

- z/OS v1.10 and v1.11, or later
- IMS for z/OS v8.1, or later
- CICS Transaction Server v3.1, or later
- DB2® for z/OS v8.1, v9.1, or later
- WebSphere MQ for z/OS v5.3.1, or later
- WebSphere Application Server for z/OS v5.0, or later
- Common Data Model v2.9.2

Supported hardware

- Any hardware environment that supports the supported software.

Related documents

- IBM z/OS V1R7.0 MVS™ JCL Reference, SA22-7597-10.
Chapter 2. Working with the DLA for z/OS

Discovery in the IBM Tivoli Discovery Library Adapter for z/OS product is completed by running the z/OS DLA batch job or started task on each z/OS system. The books that are produced are saved as members of a partitioned data set. The DLA also generates an FTP statement deck that is used to transfer the XML books to the Discovery Library File Store (DLFS). From the DLFS, the files can be loaded into TADDM, CCMDB, or TBSM.

Use this user’s guide to learn about the following tasks:
• Running the z/OS DLA
• Customizing discovery to include or exclude resource types
• Contents of the books that are produced
• Using FTP to transfer the XML book files to the Discovery Library File Store
• Using the Discovery Library IdML file viewer
• Loading the XML book files into the Tivoli CCMDB product
• Automating discovery of resources

Running the z/OS DLA

The z/OS DLA can be run as a batch job or started task. The following sample JCL members are included in the SIZDSAMP data set:

IZDJRUN
   Batch job to run the z/OS DLA. This batch job is a front end for the IZDPDISC procedure.
IZDPDISC
   Procedure to run z/OS DLA discovery and optionally FTP XML files.
IZDJFTP
   Procedure to FTP XML files to the Discovery Library File Store
IZDPFTP
   Batch job to FTP XML files to the Discovery Library File Store
IZDJRUN batch job

This sample batch job provides a batch front end to the IZDPDISC procedure. For detailed information on the statements in the IZDJRUN batch job, see the sample in SIZDSAMP.

IZDPDISC procedure

The IZDPDISC procedure includes a step to run the z/OS DLA and another optional step to FTP transfer the output XML files to the Discovery Library File Store.

This procedure is invoked by sample IZDJRUN. It can also be invoked by an operator command or automation product by issuing the S IZDPDISC command.

The following parameters are included for the IZDPDISC procedure:

IDMLDSN
The PDSE (partitioned data set extended) for XML output. This data set is allocated if it does not exist. Although not required, use a different PDSE for each z/OS system that runs the z/OS DLA. The default value supplied in the sample is &SYSUID.IZD.V3R1M0.IDML.

PARMMEM
The z/OS DLA configuration file. The z/OS DLA includes several sample
configuration files for convenience. See Appendix D, "Samples provided in hlq.SIZDSAMP," on page 171. The default IZDCDEF causes the DLA to discover z/OS, zSeries hardware, virtualization (Sysplex and z/VM guest), and active subsystems (such as DB2 v9.1 and IMS Subsystem I81A), lower-level, potentially high-volume detail, is by default filtered out. For example, Databases, Transactions and Programs are by default filtered out.

Alternatively, a configuration file may be specified in stream within batch jobs by using //ZOSDLA.IZDPARMS DD *.

The default value included in the sample is IZDCDEF.

IZDHLQ
The high-level qualifier of the z/OS DLA target data sets. The default value included in the sample is IZD.V3R1M0.

FTP
Set to either Y or N. The z/OS DLA generates FTP statements to transfer the output XML files to the Discovery Library File Storage. If FTP=Y, the FTP step in the IZDPDISC procedure is invoked to perform the transfer.

The default value included in the sample is Y.

FTPMEM
The FTP initial statement file. This parameter is applicable only if FTP=Y is specified. The initial FTP statement file contains the FTP statements that are required to connect to the remote system that contains the Discovery Library File Store. The default value included in the sample is IZDCFTPI.

Note that the default IDMLDSN parameter contains system symbol &SYSNAME. When this procedure is invoked as a started class, this symbol resolves to the SYSNAME of the z/OS that the DLA is running on. System symbols are not resolved in batch jobs. (See the z/OS MVS JCL Reference.)

When discovery completes, the started task also completes. That is, this started task is a transient started task. For detailed information on the statements in the IZDPDISC procedure, see the sample in SIZDSAMP.

IZDJFTP batch job
This batch job sample provides a batch front end to the IZDPFTP procedure. For detailed information on the statements in the IZDJFTP batch job, see the sample in SIZDSAMP.

IZDPFTP procedure
The IZDPFTP procedure completes an FTP transfer of the XML files to the Discovery Library File Store.

This procedure is invoked by samples IZDPDISC and IZDJFTP. The procedure can also be invoked by an operator command or automation product by issuing the S IZDPFTP command.

Output books
Books contain a snapshot discovery and do not necessarily contain everything known about the environment. For example, if a subsystem is not active at the time of discovery, it is not discovered. There is no difference in what Create and Refresh books discover or include. What differs is that Refresh books include the string 'Refresh' in their file name. Create books do not include the string 'Create'. Meeting CDM rules, any book that does not include 'Refresh' is a Create book. TADDM will process a Refresh book differently than a Create book:
Output books

- Create: The default book type generated by the z/OS DLA.
- Refresh: This mode can be enabled by using a DLA configuration option. The reader of the book, for example, CCMDB and TBSM, counts as obsolete anything that was loaded previously from a similar book (a book with the same member@hostname MSSName), and is not in this book. This mode is typically used following a major configuration change.

The DLA can be run to produce a single output file called ZOSALL. This option is not recommended for TADDM or CCMDB. Using the ZOSALL option to combine all books into one ZOSALL book will not reduce the time required to load books into a consuming product. ZOSALL might be needed if you use the option to directly load TBSM, rather than loading multiple books into TADDM, allowing TADDM to perform object reconciliation, and then extracting the objects to TBSM (or CCMDB).

Product support specialist teams that are responsible for managing discovery within their area typically determine when to perform discovery. By default, the DLA generates multiple books as members of a PDSE (partitioned data set extension) to support different teams performing discovery, including when the Refresh mode is used.

Depending on the configuration options that are specified and the subsystems that are active at the time of the discovery, the following members are generated:
- A member called ZOSBASE that includes hardware, zSeries virtualization, and z/OS details
- A member called ZOSTASK that includes subsystems and address space details
- A member for each DB2 Subsystem
- A member for each IMS Subsystem
- A member for each MQ Subsystem
- A member for each CICS region
- A member for each WebSphere Application Server

Configuring TADDM to display topology graphs of highly populated systems

The discovery process in Tivoli Application Dependency Discovery Manager (TADDM) creates a topology graph of discovered systems. By default, TADDM only creates topology graphs of systems with 500 or fewer nodes. The following error message indicates that the topology graph of a TADDM discovery contains more than 500 nodes:

![Error](image)

*Figure 7. Requested graph exceeded the number of allowed nodes error message*

On Windows systems, you can configure TADDM to draw topology graphs of systems with greater than 500 nodes by following these steps:

1. Navigate to C:\ibm\cmdb\dist\etc.
2. Open the collation.properties files with the Wordpad editor.
3. Press CTRL+F and find the "maxnodes" phrase.
4. Increase the value of maxnodes from 500 to 1500 (or higher for larger graphs)
5. Save and close the collation.properties file.
6. Shut down TADDM by issuing the following command from the command-line processor: C:\ibm\cmdb\dist\bin> control stop
7. Restart TADDM.

TADDM discovery should now display topology graphs with greater than 500 nodes. Do not set the value of maxnodes too high. Discovery of highly-populated systems can take an inconveniently large amount of time.

Specifying discovered resources by using sample configuration members

The z/OS DLA includes the following sample configuration members that you can use to specify which resources are discovered:

IZDCDEF
Default DLA configuration parameter deck in IZDPDISC JCL PROC.
This configuration parameter deck includes all configuration settings that control which resource types are included in books, and the type of books to create. As a result of default configuration settings, or any configuration overrides a customer specifies in IZDCDEF, resources are then written out to books (CICS, IMS, DB2, MQ, and WebSphere). The IZDCDEF parameter deck excludes some potentially high volume resources, for example, transactions, programs, and databases.

IZDCBASE
DLA configuration parameter deck to discover ZOSBASE only.

IZDCTASK
DLA configuration parameter deck to discover ZOSTASK only.

All books have an MSSName attribute that identifies the “author” of the book. The MSSName includes the member name and fully qualified host name. This enables a refresh to be performed per subsystem independent of whether other subsystems are active or inactive.

By having separate books, checksum comparisons can be made against previous discoveries, which avoids having to transfer books to the DLFS that have not changed since the previous discovery. For more details, see the FTP section.

ZOSBASE discovery

The ZOSBASE book contains zSeries® hardware and z/OS configuration details.

The following diagram shows the classes and relationships that are populated in the ZOSBASE book. The boxes that are clear are not populated.
See Appendix C, “z/OS DLA data model class types represented in CDM,” on page 81 for details on what attributes and relationships are discovered for these classes.

**Note:**

Notes:

1. To include the contents of the active parmlib members, the user ID running the DLA must have read access to the parmlib data sets.

**ZOSTASK discovery**

The ZOSTASK book contains address space details and relationships. By default only started class address spaces are discovered. System, batch job and TSO users can also be discovered via DLA configuration options.

Known address spaces are identified by their functionality. Their functionality is determined by a signature defined by using the program name and contents of the start parameters.

The following diagram shows the classes and relationships that are populated in the ZOSTASK book. The boxes that are clear are not populated.
ZOSTASK discovery

See Appendix C, “z/OS DLA data model class types represented in CDM,” on page 81 for details on the attributes and relationships are discovered for these classes.

Notes:
1. The z/OS, DB2 Data Sharing Group, DB2 Subsystem, IMS Subsystem, MQ Subsystem, CICS Region and WebsShere Server instances are “stubs”. The instances contain enough information to identify the instance and are populated for naming rules or relationships with the address spaces. Other books are used to provide full details about these instances.
2. A consolidated list of DASD volumes accessed by the address space is deduced from the allocations and is used to populate relationships to the StorageVolume instances.

DB2 subsystem discovery

The DB2 subsystem book contains the following:
• DB2 Subsystem details and relationships
• DB2 data Sharing Group details and relationships
• Database details and relationships
• Tablespace details and relationships
• Reports:
  – Databases
  – Tablespaces

To avoid unnecessarily populating a potentially high volume of instances and impacting CCMDB performance, by default, DB2 Database and DB2 Tablespace instances are not populated.
Whenever a DB2 has a configuration change, its catalog tables are updated. For example, a change to a DB2 Table column definition updates the SYSCOLUMNS catalog table. For each configuration change, an altered timestamp is updated in the catalog. The DLA discovers the maximum altered timestamp for the following catalog tables: SYSDATABASE, SYSTABLESPACE, SYSTABLES, SYSINDEXES and SYSCOLUMNS. These timestamps are saved as attributes. If a subsequent discovery for the subsystem discovers a different maximum altered timestamp, the CCMDB is able to highlight that the DB2 has had a configuration change. This is a highly efficient way to detect changes without the overhead of loading high volumes of low-level resources.

DB2 Subsystems can be configured to have their own data resources, or to share data resources with other DB2 Subsystems by using a DB2 Data Sharing Group. The z/OS DLA automatically discovers both types of DB2 environments.

The associated components that comprise the DB2 Subsystem are associated to the DB2 Subsystem. The address spaces for the MSTR, DIST, IRLM, DIST and SPAS components contain a federates relationship to the DB subsystem.

The following diagrams show the classes and relationships that are populated in the DB2 books. The boxes that are clear are not populated.
See the Appendix C, “z/OS DLA data model class types represented in CDM,” on page 81 section for details on what attributes and relationships are discovered for these classes.

Notes:

1. The z/OS instance is a stub. This instance contains enough information to identify the instance and is populated for DB2 subsystem naming rules. The ZOSBASE book contains full details about z/OS.

2. For the DLA to query the Maximum Altered Timestamps, the user ID running the DLA needs to have the read authority to access the DB2 catalog tables. For more details, see “DB2 customization” on page 61.

3. When multiple versions of DB2 are supported in the z/OS environment it is necessary to specify the correct SDSNLOAD load library in the STEPLIB or LINKLST for the discovery job. This means DB2 discovery must be run separately for each version of DB2.

**IMS subsystem discovery**

The IMS subsystem book contains the following:

- IMS Subsystem details and relationships
- IMS Sysplex Group details and relationships
- Program details and relationships
- Transaction details and relationships
- Database details and relationships
- IMS Connect details and relationships

Reports:
- Programs
- Transactions
- Databases
To avoid a potentially high volume of instances being populated unnecessarily and impacting CCMDB performance, by default, IMS transactions, programs and databases are not populated.

Unlike DB2, IMS does not have timestamps that can be discovered to detect configuration changes. However, the DLA does have highly efficient discovery code for transactions, programs, and databases via memory control blocks. The DLA internally performs the discovery and calculates a checksum value for each resource type, and these checksums are saved as attributes. If a subsequent discovery for the subsystem discovers a different checksum, the CCMDB is able to highlight that the IMS has had a configuration change. This is a highly efficient way to detect changes without the overhead of loading high volumes of low-level resources.

The following diagram shows the classes and relationships that are populated in the IMS books. The boxes that are clear are not populated.

Figure 12. IMS Subsystem details and relationships

See Appendix C, “z/OS DLA data model class types represented in CDM,” on page 81 for details on what attributes and relationships are discovered for these classes.
Note: The z/OS, DB2 Subsystem and MQ Subsystem instances are stubs. The instances contain enough information to identify the instance and are populated to satisfy naming rules. Other books are used to provide full details about these instances.

**MQ subsystem discovery**

The MQ subsystem book contains the following:

- MQ Subsystem details and relationships
- Cluster details and relationships
- Channel details and relationships:
  - Sender Channel
  - Receiver Channel
  - Server Channel
  - Requester Channel
  - Cluster Sender Channel
  - Cluster Receiver Channel
  - Client Connection Channel
  - Server Connection Channel
- Queue details and relationships:
  - Local Queue
  - Remote Queue
  - Alias Queue
  - Model Queue

To avoid a potentially high volume of instances being populated unnecessarily and impacting CCMDB performance, by default, MQ Queues are not populated.

The following diagram shows the classes and relationships that are populated in the MQ books. The boxes that are clear are not populated.
See Appendix C, “z/OS DLA data model class types represented in CDM,” on page 81 for details on what attributes and relationships are discovered for these classes.

Notes:
- The z/OS instance is a stub. This instance contains enough information to identify the instance and is populated for MQ subsystem naming rules. The ZOSBASE book contains full details about z/OS.
- The DLA uses an MQ API for MQ discovery. If the MQ load module datasets are not in the LINKLST, they must be added to the DLA's STEPLIB DD concatenation. For more details, refer to the Installation and Customization section.

CICS region discovery

The CICS Region book contains the following:
- CICS Region details and relationships
- Program details and relationships
- Transaction details and relationships
- Database details and relationships
- Transaction to Program details and relationships
- CICS Connection details and relationships:
  - DB2 Connections
  - CICS Connections
  - MQ Connections
- CICS Transaction Gateway details and relationships
- Reports:
  - Programs

Figure 13. MQ Subsystem details and relationships
In order to avoid a potentially high volume of instances being populated unnecessarily and impacting CCMDB performance, by default, CICS Transactions, Programs and Databases are not populated.

Unlike DB2, CICS does not have timestamps that can be discovered to detect configuration changes. However, the DLA does have highly efficient discovery code for Transactions, Programs and Databases via memory control blocks. The DLA internally performs the discovery and calculates a checksum value for each resource type, and these checksums are saved as attributes. If a subsequent discovery for the subsystem discovers a different checksum, the CCMDB is able to highlight that the CICS has had a configuration change. This is a highly efficient way to detect changes without the overhead of loading high volumes of low-level resources.

The following diagram shows the classes and relationships that are populated in the CICS books. The boxes that are clear are not populated.

See Appendix C, “z/OS DLA data model class types represented in CDM,” on page 81 for details on what attributes and relationships are discovered for these classes.
CICS region discovery

Notes:
1. The z/OS, DB2 Subsystem, IMS Subsystem and MQ Subsystem instances are "stubs". The instances contain enough information to identify the instance and are populated to satisfy naming rules. Other books are used to provide full details about these instances.
2. CICS Discovery works independently of address space filters when discovering started class and batch jobs.

WebSphere Application Server discovery

In this section, see information about TADDM WebSphere Sensor seed file discovery detail, and determining how often to run the WebSphere Application Server seed utility for the z/OS DLA.

TADDM WebSphere Sensor seed file discovery detail

z/OS DLA books can discover the following higher-level WebSphere resources and include them in books:

- WebSphereServer
- WebSphereCell
- WebSphereNode
- WebSphereNamedEndPoint
- ConfigFile
- LogicalContent
- AppDescriptor

The WebSphere Application Server seed utility for the z/OS DLA can scan information in a z/OS DLA WebSphere book to produce a WebSphere XML seed file. This seed file is used by the TADDM WebSphere Sensor to discover lower-level WebSphere resources on z/OS systems. The seed file consists of XML tags that includes version and TCP/IP address information about the WebSphere resources that you are trying to discover on the z/OS system. Here's an example:

```
<WebSphereServer>
    <WAS_Name>WebSphereServer1</WAS_Name>
    <WAS_Type>Server</WAS_Type>
    <WAS_Cell_Name>WebSphereCell1</WAS_Cell_Name>
    <WAS_Node_Name>WebSphereNode1</WAS_Node_Name>
    <Fqdn>webserver1.example.com</Fqdn>
</WebSphereServer>
```

Figure 15. TADDM WebSphere Sensor seed file

If you have WebSphere Deployment Managers on z/OS, you only need to run the WebSphere Application Server seed utility on those z/OS LPARs. The TADDM
WebSphere application server discovery

WebSphere Sensor discovers and populates TADDM with WebSphere lower-level resources for all WebSphere cells controlled by the WebSphere Deployment Managers on z/OS.

If you have not installed WebSphere Deployment Managers on z/OS, you must run the WebSphere Application Server seed utility on any z/OS LPARs where you want to discovery WebSphere. After this seed file is created and installed on the TADDM server, on the next discovery, the WebSphere IdmlSeedSensor looks for z/OS WebSphere seed files on the TADDM Server. If a seed file exists, it parses that seed file and creates a discovery seed file that is used to kick off the WebSphere sensor. The WebSphere sensor then does a deeper discovery of WebSphere on the z/OS system. It uses a TCP/IP SOAP Server connection from TADDM to the z/OS LPAR.

You can obtain the WebSphere Application Server seed utility for the z/OS DLA from the IBM Service Management (previously called OPAL) website. However, if you have installed z/OS DLA, you already have a copy of WebSphere Application Server seed utility installed on z/OS. You should use that z/OS utility. Downloading the WebSphere Application Server seed utility for the z/OS DLA from IBM Service Management OPAL website provides only the software you have already installed with z/OS DLA.

Note that you do not have to use the WebSphere Application Server seed utility or the TADDM WebSphere Sensor for WebSphere discovery on z/OS:

- If your needs are met by the seven higher-level WebSphere resource classes discovered by z/OS DLA, go no further.
- If you have all the information to create a WebSphere sensor XML seed file at your TADDM server, you don't need to run the WebSphere Application Server seed utility – it's only provided for your convenience.
- The z/OS DLA discovery is not changed by running the WebSphere Application Server seed utility.

Procedure

Run a z/OS DLA discovery first to create an output book that identifies your WebSphere environment on this z/OS LPAR. Then, in the z/OS DLA <HLQ> SIZDEXEC installation data set, run the IZDJSEED utility, which invokes the IZDRSEED REXX program that scans the output WebSphere book.

After running the WebSphere Application Server seed utility on z/OS, you must transfer (FTP) the seed file to your TADDM Server, and then copy this .xml file into the $COLLATION_HOME/var/dla/zos/was directory.

Next, you need to follow instructions for the TADDM WebSphere sensor to configure and start the sensor. The sensor discovers additional WebSphere resources on z/OS and automatically loads them into your TADDM Server. Configuration instructions for the IBM WebSphere sensor are available at the TADDM Information Center at http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/index.jsp?topic=/com.ibm.taddm.doc_7.2/welcome_page/welcome.html.

Determining how often to run the WebSphere Application Server seed utility for the z/OS DLA

Review the following information to determine how often you need to run the WebSphere Application seed utility:
WebSphere application server discovery

- Most customers can run the utility once and then have TADDM WebSphere sensor detect any changes. You do not need to run the utility every time you run a z/OS DLA discovery.
- The seed file identifies the WebSphere version number, higher-level resource names (such as the WebSphere Server name), and TCP/IP socket connections. If those configuration parameters change, you should re-run the WebSphere Application Server seed utility.
- If you are already running WebSphere sensor to discover WebSphere on z/OS resources, and WebSphere configuration changes only slightly, such as the WebSphere Server name changes from TVS5999:server1 to TVS5999:server2, you can simply edit the seed file at your TADDM server and restart sensor discovery.
- When you install a new version of WebSphere on z/OS, you should re-run the Seed utility.
- If you change TCP/IP connector ports for WebSphere on z/OS, you should re-run the Seed utility.

Using the WebSphere seed sensor on a z/OS system is described in this section of the TADDM Information Center at http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/index.jsp?topic=/com.ibm.taddm.doc_7.1.2/AdminGuide/c_cmdb_aboutwasseed.html

The following diagram shows the classes and relationships that are populated in the WebSphere Application Server books. The boxes that are clear are not populated.

See the Appendix C, “z/OS DLA data model class types represented in CDM,” on page 81 section for details on what attributes and relationships are discovered for these classes.
WebSphere application server discovery

Notes:
- To include the contents of the WebSphere Application Server configuration files, the user ID running the DLA must have read access to the files.
- WebSphere Application Server Discovery works independently of address space filters when discovering started class and batch jobs.
- The WebSphere Named Endpoint discovery includes the SOAP port the WebSphere Application Server API is configured for. The Tivoli Application Dependency Discovery Manager (TADDM) WebSphere Application Server Sensor uses this information to perform low-level WebSphere Application Server discovery.

ZOSALL book
This ZOSALL book contains everything in a single book. See the Reference section on how to enable this type of book. Do not load the ZOSALL book into TADDM. Use of ZOSALL will not reduce bulk load time, and can cause possible Refresh conflicts in TADDM.

The following diagram shows the classes and relationships that are populated in the ZOSALL books.

Figure 17. ZOSALL—Everything that is active in one book

FTP transfer to the Discovery Library File Store
The z/OS DLA generates several output XML files as members in a PDSE. These files need to be downloaded to the Discovery Library File Store (DLFS) for loading into CCMDB or TBSM.

If the server where the DLFS resides has an FTP Server service/daemon, the DLA provides sample JCL to perform the transfer. The DLA generates the FTP statement deck and this is used in a subsequent FTP transfer step.
FTP transfer to the Discovery Library File Store

If FTP transfer is not possible, an alternate technique must be used, for example, manual download via 3270 emulator. Note that the transfer must do an EBCDIC to ASCII translation. It is highly recommended that the downloaded file name complies to the Discovery Library file name standards explained below.

**Target DLFS FTP server**

The IZDCFTPI member in SIZDSAMP is referenced by JCL samples IZDJRUN and IZDPDISC.

This member contains the initial FTP statements to establish a session with the target DLFS FTP Server.

The following is an extract of this sample. Please refer to the SIZDSAMP data set for the complete version.

```
* Name: IZDCFTPI
* Function: Initial FTP statements used to PUT members to the
* Discovery Library File Store (DLFS).
* The z/OS DLA will generate the following members in the IdML PDSE:
  * @FTPCHGP - FTP statements to PUT changed IdML members
  * @FTPALLP - FTP statements to PUT all IdML members
* @FTPALLP is the default FTPMEM parameter in the IZDJFTP proc.
  * @FTPALLP is used as the FTPMEM parameter to IZDJFTP in IZDJDISC.
* The FTP statements comply with the Discovery Library standards
  * explained below. Before the PUT statements, the z/OS DLA
  * reads this member and copies the lines that do not have '*' in
  * the first column to the output FTP statements.
* This member may contain:
  * <dlfs>
  * <userid> <password>
  * cd <directory>
  * type a
  * Where:
    * <dlfs> = FTP IP name or FTP IP address of the DLFS
    * <userid> = FTP userid
    * <password> = FTP password
    * <directory> = Directory to transfer to files to
* FTP Notes:
  * - The "<dlfs>" line should be omitted if the FTP JCL has been
    * customized to specify the destination in the EXEC PARM clause.
  * - The "<userid> <password>" should be omitted if the FTP JCL
    * has been customized to specify a NETRC DD.
  * - The "cd <directory>" can be omitted if the default directory
    * for the specified userid is appropriate.
```

```
pthtsd3.perthapc.au.ibm.com
userid password
cd /u/userdirectory
type a
```

**Discovery Library file naming standards**

All files in the DLFS must comply with the Discovery Library file name standards:

```
<appcode>.<hostname>.<timestamp>.<refresh>.xml
```

where:
Discovery Library file naming standards

- **<AppCode>=** the DLA Application Code
  - ZOSDISC310ALL for ZOSALL books
  - ZOSDISC310BASE for ZOSBASE books
  - ZOSDISC310TASK for ZOSTASK books
  - ZOSDISC310DB2 for DB2 Subsystem books
  - ZOSDISC310IMS for IMS Subsystem books
  - ZOSDISC310MQ for MQ Subsystem books
  - ZOSDISC310CICS for CICS Region Subsystem books
  - ZOSDISC310WAS for WebSphere Application Server books

- **<Hostname>=** the MSS Hostname specified in the IdML MSSName, which the z/OS DLA populates in the format < member>@<fqdn>

- **<timestamp>=** the timestamp specified in the IdML create or IdML refresh clauses.

- **<.refresh>=** ".refresh" if the IdML has a refresh clause.

- File name conversions are stored in @FTPCHGG members.

Discovery Library standards also say that files being transferred to the DLFS include a suffix of ".partial". When transfer is complete, the file is renamed to remove the suffix. This file name convention is not applicable on z/OS because of a 44-character limit for the data set name and a restriction on symbols used. So the z/OS DLA generates FTP statements to transfer the IdML members and renames them to the appropriate Discovery Library format, for example:

```plaintext
PUT IMP1CTL
    ZOSDISC210IMS.IMP1CTL@pthomo1.au.ibm.com.2006-09-04T02.07.17Z.xml.partial
```

```plaintext
REN ZOSDISC210IMS.IMP1CTL@pthomo1.au.ibm.com.2006-09-04T02.07.17Z.xml.partial
    ZOSDISC210IMS.IMP1CTL@pthomo1.au.ibm.com.2006-09-04T02.07.17Z.xml
```

The mapping of z/OS members to longer file names that meet Discovery Library standards is specified in members @FTPCHGG and @FTPCHGP, which are written to the same directory as the books.

**Checksums to detect if members must be transferred**

To improve performance, the DLA calculates a checksum for each book it generates, ignoring the IdML timestamp that is within the book. When a new discovery is performed the checksum is compared and if it is identical to the previous discovery, the book is not replaced, and FTP statements are not placed into the @FTPCHGP statement deck.

The ZOSBASE and subsystem books may contain a high volume of data and in many cases rarely change. So this checksum processing works well for these books.

The ZOSTASK book is more volatile since the address spaces may be active or inactive at the time of discovery. It is up to the CCMDB to handle what has changed, but at least this book is not as large as ZOSBASE and subsystem books.

The ZOSALL book has volatile and possible high volume content. Checksum processing is not effective for this book. The use of this book is recommended only when the book is to be consumed locally on the z/OS and not transferred to the DLFS.
Automating discovery

Automating discovery

Repeated discoveries per z/OS are required for TADDM or CCMDB to detect changes. You can automate repeated discoveries by using an automation product such as IBM Tivoli Systems Automation for z/OS, or a job scheduling product, such as IBM Tivoli Workload Scheduler.

For example, after an IPL, the ZOSBASE book must be rediscovered to pick up changes in the PARMLIB and SYSRES.

Update the change control procedures to include the requirement of performing a rediscovery immediately after the change has been implemented. For example, immediately after a CICS application has been rolled out, a CICS rediscovery should be performed.

What did you discover? - Tips and Tools

Most users will want to confirm that their initial discovery runs are getting what they expected. It will save time to do this before loading books into a consuming product. You may need to ensure the resources are active at time of discovery, or change DISC and FILT statements in the customization file. See "z/OS DLA discovery customization" on page 37.

The output directory contains the XML books as well as files that summarize discovery results. You designated the output directory with the parameter IDMLDSN in IZDJRUN or IZDPDISC.

- DLALOG shows the filter parameters enabled from the customization file used.
  It also shows Subsystems discovered, any resource excluded by the SET limit, and Subsystems output to books.
- @PERFLOG shows the elapsed time for discovery and output of each Subsystem.

You might need to look in the books, to learn which lower-level resources were output. You can use any editor to open the XML books. An XML editor makes book review easier because these editors highlight results and offer improved search tools. The IdML XML Browser, a tool that can be downloaded from IBM, provides a handy count of each resource type (CDM class), provides hot links that follow relationships between objects, and formats resources (objects) into tables.

IdML XML Browser

Use the idml_browse.hta utility to browse the output IDML XML files in a web page format, with hyperlink navigation based on discovered relationships. You can download the idml_browse.hta utility for free from the IBM Integrated Service Management Library at http://www.ibm.com/software/brandcatalog/portal/opal/details?catalog.label=ITW10CC0B. Search on "IDML Browser" or "idml_browse.hta."

Complete the following steps to use the idml_browse.hta utility file:

- Download IdML files that the z/OS DLA has generated, or access the files that have been transferred to the Discovery Library File Store.
- Open the idml_browse.hta file.
- Specify the IdML XML file, or click Browse to locate an IdML XML file that you want to view. The idml_browse.hta file uses your default web browser.
- Click Submit. The XML file read in and formatted for viewing in the window.
Loading the XML files into Tivoli TADDM

After downloading the XML files to the Discovery Library File Store, use the loadidml.sh TADDM bulk loader program to load the books into TADDM, for example:

1. cd <hlq>/cmdb/dist/bin
2. ./loadidml.sh -g -f <path_to_books>

Bulk load into TADDM is explained in the IBM Tivoli Application Dependency Discovery Manager User's Guide.

Note that a TADDM LOADIDML Return Code of 0 means that the program completed. This does not mean that everything was loaded. Check the results file for that information.


If you do not see all the z/OS resources you expected in TADDM, consider the following:
Loading the XML files into TADDM

1. Did z/OS DLA discover the resources?
   a. Edit the z/OS DLA output XML books to see if the resources you need are included. If not, consider the following:
      1) Have you turned off any z/OS DLA default filters that would prevent discovery? Many detail-level resources are only included in output books if you set their FILTER to INCLUDE.
      2) For example: Active CICS Regions are always included. To add CICS Transactions, you must change IZDCDEF customization member to FILT CICS.TRANSACTION INCLUDE.
   b. Was the resource active when z/OS DLA discovery was run?

2. Did TADDM load the missing resources?
   a. Are the resources displayed at TADDM? Note that some lower-level resources will show in TADDM Details, but not in the TADDM topology layout.
   b. Examine what was loaded in the <hlq>/cmdb/dist/support/buik/results directory.

3. Can the missing resources be displayed in the portion of the TADDM UI you are using?
   a. Some detail-level resources can be displayed in the Detail view, Configuration Change dialog boxes, and database queries but not in Topology Layout (such as AddressSpace objects).
   b. Some high-volume, detail-level resources can be displayed in Configuration Change dialogs and database queries, but not in other windows (ex. CICS Transaction objects).
   c. If z/OS DLA filters excluded some resources (CIs) you need, you may be able to see a complete list of those resources in a Report tab from the Details view (ex. CICS Transactions).

You can use TADDM Change and Comparison with z Reports.
Chapter 3. Reference information

Review the following sections to customize your server resources.

Creating and managing custom servers

You can create custom servers to discover and categorize applications that are not, by default, supported by the Discovery Library Adapter for z/OS (DLA). This is an advanced technique for configuring the z/OS DLA to discover applications that it does not know about by default.

Your infrastructure might contain software applications and server types, such as custom database applications (servers), that are not automatically categorized by the z/OS DLA. Any address space with a TCP listening port can be discovered and created in output books.

You can define a custom server to create a template that sets up the membership rules for the custom server. Custom servers are displayed in the topology and you can view details about them. Although these details are not as complete as those provided for supported applications (servers), defining custom servers allows all components in your infrastructure to participate in the topology and comparisons.

Identifying unknown server patterns

Before adding a server, run a basic discovery to check for address spaces that are available as a possible custom server.

For the discovery process to identify this address space, add the following configuration directive to the IZDCDEF default parameter member:

DISC CUSTOMSERVER YES

The discovery process message IZD0070I is displayed for every available job that can be used to create custom servers.

To identify patterns in unknown servers, complete the following steps from the TADDM product console:

1. Click Physical Infrastructure ➔ System Tier.
2. Navigate to the z/OS system to view the details.
3. View the contents of the Address Space tab.

Identify a pattern in the configuration of the unknown address space, such as the JOB name, program name, program arguments, JOB user ID, and JOB type. Use this pattern to create the identifying criteria for the customer server template. See “Adding custom servers” on page 34 for more information.

For IP ports to be used in the configuration of the unknown address spaces, view the IP Ports tab from the Address Space details.
Adding custom servers

A custom server template contains descriptive criteria that are used to assign unknown server processes to the custom server. You specify these criteria when defining the template for the custom server.

The following information associated with running processes is parsed to match the process to a particular custom server:

**JOB name**

The name of the JOB name of the address space.

**Program name**

The name of the executable program.

**Argument**

The arguments passed to the program.

**Environment**

The environment variables set for the program (JOB Userid, JOB Type).

The custom server general information and criteria details include the name, the type of server, and identifying criteria for the custom server.

Adding this custom server resource to a business application

When an address space has been defined to create a custom server, this application can always be associated with a business application.

It is very easy to associate this application with a business application. You only need to update the CUSTSRVR configuration directive in the z/OS DLA IZDCDEF default configuration file with the location of the Application Descriptor file, as shown in the following example:

```
CUSTSRVR name.type.criteria appdesc_loc
```

where `appdesc_loc` is the location of the application descriptor to be used for this application.

The TADDM Application Descriptor files can be used to profile an application with a business application. See the IBM Tivoli Application Dependency Discovery Manager User’s Guide for more information.

The following example is the Application Descriptor file XML markup:

```xml
<component-app-descriptor app-instance-name="Payroll - Staging">
  <component-descriptor
    type="server"
    name="tn3270x"
    functional-group="zApp Tier"
    marker-module="false"
  />
</component-app-descriptor>
```

If this markup is saved into a member in the hlq:(TN3270XX)(PAY001), the CUSTSRVR ATM.APPSERVER.TN3270 <hlq>(TN3270XX)(PAY001) rule associates all of the address spaces that satisfy the criteria with this application descriptor, as shown in the following example:
Associate an icon with your custom server

Now that you have created custom servers based on the criteria you selected, you can associate an icon of your choice. To do this, create a custom server template in the TADDM user interface using the following steps:

1. From the tool bar, click **Database → Custom Server**.
2. Select **Add** to create a new custom server.
3. On the dialog box that is displayed, enter in the information for this custom server from the CUSTSRVR statement.

**Name:** ATM  
Use the same name that was used in the CUSTSRVR configuration directive.

**Type:** APPSERVER  
Use the same type that was used in the CUSTSRVR configuration directive.
4. Select the Enabled box.
5. In the criteria section, enter a program name of ZOSDLA.

**Note:** The ZOSDLA name is being used only to associate an icon with this custom server. The value used in the criteria should be something that will never match a resource on the distributed side.

When the custom server template is created, the TADDM program associates the icon with the resource wherever the resource is referenced.

In Figure 19 on page 35, the icon is associated in the Discovered Components section under **Application Infrastructure > Custom Servers > ATM > TN3270-J80**.

When displaying the application topology, you will see the icon you selected when creating the custom server template associated with the application.
The range and depth of z/OS discovery can be controlled by a combination of four filtering directives that can be specified in the IZDPARMS parameter file. The easiest way to customize discovery is to modify the sample filtering directive in the IZDCDEF default configuration file. This file is read by default for each discovery.

The filtering directives SET, DISC, FILT, BOOK, and DEBUG that are described in this section control which resources are discovered, which information is output, and how the information is output.

The SET directive is used to specify general options and values such as organization name, debug options, and output limits to be placed on resources that typically exist in large numbers. The SET LIMIT directive is used to help with performance which may be degraded when large numbers of resources are discovered and loaded into the CCMDDB.

The DISC directive is used to specify which resources you want to discover. Use this directive to limit discovery to CICS or IMS only, for example, or to discover IMS database resources but not discover IMS program resources.
The FILT directive is used to limit the XML output. The input to the bulk load program is a file that contains an IDML-formatted XML document.

The FILES directive is used to read user-defined files associated with a subsystem. The FILES directive can also be used to read an Application Descriptor and associate the subsystems.

The BOOK directive is used to specify how the IdML output is stored. The IdML can be storage in a single book (member) named ZOSALL or split into multiple books, each named using the jobname or STC name of the discovered resource.

The DEBUG directive is used to set debug options.

Several sample members are provided in the hlq.SIZDSAMP library. The IZDCDEF member contains the default directive recommended for a typical discovery. Other books are described in Appendix D, “Samples provided in hlq.SIZDSAMP,” on page 171.

**SET directive**

The SET directive is used to set general parameter values such as the organization name and debug settings and can also be used to set limits for high volume resources.

Syntax: SET index value

Supported indexes:

<table>
<thead>
<tr>
<th>Index</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orgname</td>
<td>This is the name of the organization. It should be a single word without any imbedded blanks. It should contain only alphanumeric and national characters, with xml escape characters being used for any other characters. The default value is &lt;default&gt;</td>
</tr>
<tr>
<td>REFRESH</td>
<td>Yes to generate a refresh configuration. No to generate an update configuration. The default is No.</td>
</tr>
<tr>
<td>FTP</td>
<td>Yes to generate an output member containing FTP put instructions. The default is Yes.</td>
</tr>
<tr>
<td>FTP_GET</td>
<td>Yes to generate an output member containing FTP get instructions. If this parameter is specified then an additional DD statement named IZDFTPIG must be specified in the discovery JCL pointing to a sequential data set or PDS member containing the initial FTP get control statements. The default is No.</td>
</tr>
<tr>
<td>HOSTNAME</td>
<td>The DLA discovers the host name. If the z/OS system is behind a firewall, it is possible that the system might have a different external host identity. In this case, you can override the discovered host name. This can also be used if the DLA is unable to obtain the host name because of IUCV errors.</td>
</tr>
</tbody>
</table>
HOSTDOMAIN
The DLA discovers the host domain. If the z/OS system is behind a firewall, it is possible that the system might have a different external host identity. In this case, you can override the discovered host domain. This can also be used if the DLA is unable to obtain the host domain because of IUCV errors.

HOSTIPADDR
The DLA discovers the IP address. If the z/OS system is behind a firewall, it is possible that the system might have a different external identity. In this case, you can override the discovered IP address. This can also be used if the DLA is unable to obtain the IP address because of IUCV errors.

DISC directive
The DISC directive is used to set discovery parameter values. This directive can be used to specify or limit the scope of discovery.

You can use this parameter, for example, to restrict discovery to only CICS resources, allowing the CICS support group to run CICS-only discovery independent of other discovery processes.

If you set DISC to yes, and debug to 1 (or ON), you can see what is discovered in the DLALOG file. Discovered resources are shown in output books (in xml files) only if you also set FILT INCLUDE for associated values, as illustrated in the following examples:

DISC DB2.IFI YES
FILT DB2.IFI INCLUDE

The default for most DISC statements is 'YES'.

Syntax: DISC index value

Supported indexes:

Index Value
DB2, IMS, CICS, MQ, WEBSPHERE, and STORAGE
The following DISC (discovery) statements not only result in discovery, but also create these high-level resources in the books. DISC DB2 YES, DISC IMS YES, DISC CICS YES, DISC MQ YES, DISC WEBSPHERE YES, DISC STORAGE YES. As explained in latest IZDCDEF customization sample, high-level Subsystems are both DISCovered and FILT Included by DISC yes (for example, DISC MQ YES). Lower-level resource types require both a DISC DB2 YES and a FILT DB2 INCLUDE statement.

The default is Yes.

CUSTOMSERVER
Yes to list and No not to list all address spaces that are available for custom server processing. Only address spaces that have a listening IP port qualify as a possible custom server resource.

The default is Yes.

DB2

DB2 The DB2 DISC directive controls the discovery of the highest-level
resources (discovery subsystems). By including the default of Yes, all DB2 resources are discovered. A setting of No excludes the discovery of all DB2 resources.

The default is Yes.

**DB2.jobprefix**

Yes to include and No to exclude DB2 discovery for a specific DB2 subsystem.

The default is Yes.

**DB2.DATABASE**

Yes to include and No to exclude DB2 database discovery.

The default is No.

**DB2.GROUP**

Yes to include and No to exclude DB2 group discovery. DB2 group discovery is required to populate the DB2 VersionString attribute.

Authority to issue GROUP statements against these DB2 resources is required to use this option. DSNREXX must be bound to each DB2 instance where GROUP discovery is required. See the “DB2 customization” on page 61 section for a list of customization requirements for all DB2 systems where discovery is run.

The default is Yes.

**DB2.IFI**

Yes to include and No to exclude DB2 IFI discovery routines.

DB2 IFI discovery is required to discover DB2 group resources and relationships. Authority to issue DB2 DISPLAY DB and DISPLAY GROUP commands is required to use this option.

The default is Yes.

**DB2.SQL**

Yes to include and No to exclude DB2 SQL discovery routines. If you set DB2.SQL to Yes, the following attributes are discovered:

SYSIBM.SYSDATABASE/SysdatabaseMaxAlterdts attribute
SYSIBM.SYSTABLESPACE/SystablespaceMaxAlterdts attribute
SYSIBM.SYSTABLES/SystablesMaxAlterdts attribute
SYSIBM.SYSINDEXES/SysindexesMaxAlterdts attribute
SYSIBM.SYSCOLUMNS/SyscolumnsMaxAlterdts attribute

Authority to issue SQL statements against these DB2 resources is required to use this option. DSNREXX must be bound to each DB2 instance where SQL discovery is required. See the “DB2 customization” on page 61 section for a list of customization requirements for all DB2 systems where discovery is run.

The default is Yes.

**DB2.TABLESPACE**

Yes to include and No to exclude DB2 tablespace discovery.

The default is Yes.

**WEBSHERE**
DISC directive

WEBSHARE

The WebSphere DISC directive controls the discovery of the highest-level resources (discovery subsystems). Set the DISC directive to Yes to include and No to exclude WebSphere subsystems if it is the first entry in a DISC or FILT statement.

The default is Yes.

WEBSHARE.jobprefix

Yes to include and No to exclude WebSphere discovery for a specific WebSphere subsystem.

The default is Yes.

IMS

IMS

The IMS DISC directive controls the discovery of the highest-level resources (discovery subsystems). Set the DISC directive to Yes to include and No to exclude IMS subsystems if it is the first entry in a DISC or FILT statement.

The default is Yes.

IMS.jobprefix

Yes to include and No to exclude IMS discovery for a specific IMS subsystem. A setting of Yes discovers all IMS subsystems.

The default is Yes.

IMS.DATABASE

Yes to include and No to exclude IMS database discovery and IMS DatabasesChecksum attribute.

The default is Yes.

IMS.DEPREG

Yes to include and No to exclude IMS dependent region discovery.

The default is Yes.

IMS.PROGRAM

Yes to include and No to exclude IMS program discovery and IMS ProgramsChecksum attribute.

The default is Yes.

IMS.TRANSACTION

Yes to include and No to exclude IMS transaction discovery and IMS TransactionsChecksum attribute.

The default is Yes.

MQ

MQ

The MQ DISC directive controls the discovery of the highest-level resources (discovery subsystems). Set the DISC directive to Yes to include and No to exclude MQ subsystems when it is the first entry in a DISC or FILT statement.

The default is Yes.

MQ.jobprefix

Yes to include and No to exclude MQ discovery for a specific MQ subsystem.

The default is No.
DISC directive

MQ.PROCESS
Yes to include and No to exclude MQ discovery for a specific MQ process.
The default is Yes.

MQ.PROCESS.jobprefix
Yes to include and No to exclude MQ discovery for a specific MQ process.
The default is No.

MQ.QUEUE
Yes to include and No to exclude MQ queues.
The default is Yes.

MQ.QUEUE.jobprefix
Yes to include and No to exclude MQ discovery for a specific MQ queue.
The default is No.

MQ.QUEUE_ALIAS
Yes to include and No to exclude MQ discovery for a specific MQ queue.
The default is No.

MQ.QUEUE_ALIAS.jobprefix
Yes to include and No to exclude MQ discovery for a specific MQ queue.
The default is No.

MQ.QUEUE_LOCAL
Yes to include and No to exclude MQ discovery for a specific MQ queue.
The default is No.

MQ.QUEUE_LOCAL.jobprefix
Yes to include and No to exclude MQ discovery for a specific MQ queue.
The default is No.

MQ.QUEUE_REMOTE
Yes to include and No to exclude MQ discovery for a specific MQ queue.
The default is No.

MQ.QUEUE_REMOTE.jobprefix
Yes to include and No to exclude MQ discovery for a specific MQ queue.
The default is No.

MQ.QUEUE_MODEL
Yes to include and No to exclude MQ discovery for a specific MQ queue.
The default is No.

MQ.QUEUE_MODEL.jobprefix
Yes to include and No to exclude MQ discovery for a specific MQ queue.
The default is No.

MQ.QUEUE_REMOTE
CICS

The CICS DISC directive controls the discovery of the highest-level resources (CICS regions). Set the DISC directive to Yes to include and No to exclude CICS regions if it is the first entry in a DISC or FILT statement.

The default is Yes.

CICS.jobprefix

Yes to include and No to exclude CICS discovery for a specific CICS subsystem.

The default is Yes.

CICS.PROGRAM

Yes to include and No to exclude CICS program discovery and CICS ProgramsChecksum attribute.

The default is Yes.

CICS.TRANSACTION

Yes to include and No to exclude CICS transaction discovery and CICS TransactionsChecksum attribute.

The default is Yes.

CICS.FILE

Yes to include and No to exclude CICS file discovery and CICS FilesChecksum attribute.

The default is Yes.

STORAGE

STORAGE

The STORAGE DISC directive controls the discovery of the highest-level resources (discovery subsystems). Set the DISC directive to Yes to include and No to exclude STORAGE subsystems if it is the first entry in a DISC or FILT statement.

The default is Yes.

STORAGE.VOLUME

Yes to include and No to exclude storage discovery routines.

The default is Yes.

FILT directive

The FILT directive is used to set the filter discovery parameter values. This directive can be used to specify or limit the type and amount of discovery output.

If you set DISC to yes, and debug to 1 (or ON), you can see what is discovered in the DLALOG file. Discovered resources are shown in output books (in xml files only) if you also set FILT INCLUDE for associated values, as illustrated in the following examples:

DISC DB2.IFI YES
FILT DB2.IFI INCLUDE

Syntax: FILT index value
FILT directive

Supported indexes:

<table>
<thead>
<tr>
<th>Index</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASTYPE.BATCH</td>
<td>EXCLUDE suppresses output for all batch jobs.</td>
</tr>
<tr>
<td></td>
<td>The default is EXCLUDE.</td>
</tr>
<tr>
<td>ASTYPE.BATCH.binaryname</td>
<td>EXCLUDE suppresses output for all batch jobs that match the binaryname or partial binaryname.</td>
</tr>
<tr>
<td></td>
<td>For example, FILT ASTYPE.BATCH.PROD EXCLUDE suppresses all output for jobs name PROD or starting with PROD.</td>
</tr>
<tr>
<td></td>
<td>The default is EXCLUDE.</td>
</tr>
<tr>
<td>ASTYPE.STARTED</td>
<td>EXCLUDE suppresses output for all started tasks.</td>
</tr>
<tr>
<td></td>
<td>The default is INCLUDE.</td>
</tr>
<tr>
<td>ASTYPE.STARTED.binaryname</td>
<td>EXCLUDE suppresses output for all started tasks that match the binaryname or partial binaryname.</td>
</tr>
<tr>
<td></td>
<td>For example, FILT ASTYPE.STARTED.JES EXCLUDE suppresses all output for jobs name JES or starting with JES.</td>
</tr>
<tr>
<td></td>
<td>The default is EXCLUDE.</td>
</tr>
<tr>
<td>ASTYPE.SYSTEM</td>
<td>EXCLUDE suppresses output for all system tasks.</td>
</tr>
<tr>
<td></td>
<td>The default is EXCLUDE.</td>
</tr>
<tr>
<td>ASTYPE.SYSTEM.binaryname</td>
<td>EXCLUDE suppresses output for all system tasks that match the binaryname or partial binaryname.</td>
</tr>
<tr>
<td></td>
<td>For example, FILT ASTYPE.STARTED.JES EXCLUDE suppresses all output for jobs name JES or starting with JES.</td>
</tr>
<tr>
<td></td>
<td>The default is INCLUDE.</td>
</tr>
<tr>
<td>ASTYPE.TSOUSER</td>
<td>EXCLUDE suppresses output for all TSO users.</td>
</tr>
<tr>
<td></td>
<td>The default is EXCLUDE (suppress output).</td>
</tr>
<tr>
<td>ASTYPE.TSOUSER.binaryid</td>
<td>EXCLUDE suppresses output for all TSO users that match the binaryid or partial binaryid.</td>
</tr>
<tr>
<td></td>
<td>For example, FILT ASTYPE.TSOUSER.FRED EXCLUDE suppresses all output for jobs name FRED or starting with FRED.</td>
</tr>
<tr>
<td></td>
<td>The default is EXCLUDE (suppress output).</td>
</tr>
<tr>
<td>JOBNAME</td>
<td>All jobs.</td>
</tr>
<tr>
<td></td>
<td>The default is INCLUDE.</td>
</tr>
<tr>
<td>JOBNAME.binaryname</td>
<td>EXCLUDE suppresses output for all started tasks, jobs, and TSO users that match the binaryname or partial binaryname.</td>
</tr>
</tbody>
</table>
FILT directive

For example, FILT JOBNAME.TEST EXCLUDE suppresses all output for STCs, jobs, and TSO users where the STCname, jobname or TSO user ID is TEST or starts with TEST.

The default is EXCLUDE.

ZOS.REPORT
The default is INCLUDE.

CICS

CICS.MQCONN
EXCLUDE suppresses the output for the CICS to MQ links.

The default is EXCLUDE.

CICS.MQCONN.jobprefix
EXCLUDE suppresses the output for the CICS to MQ links for a specific CICS region.

The default is EXCLUDE.

CICS.DB2CONN
EXCLUDE suppresses the output for the CICS to DB2 links.

The default is EXCLUDE.

CICS.DB2CONN.jobprefix
EXCLUDE suppresses the output for the CICS to DB2 links for a specific CICS region.

CICS.CICSCONN
EXCLUDE suppresses the output for the CICS to CICS links.

The default is EXCLUDE.

CICS.CICSCONN.jobprefix
EXCLUDE suppresses the output for the CICS to CICS links for a specific CICS region.

The default is EXCLUDE.

CICS.CICSCONN_NAME.applid
EXCLUDE suppresses the output for the CICS to CICS links for a specific APPLID.

The default is EXCLUDE.

CICS.PROGRAM_NAME.programprefix
EXCLUDE suppresses output for CICS program resources that match the specified program prefix for all CICS regions.

The default is EXCLUDE.

CICS.REPORT
EXCLUDE suppresses the report for CICS resources such as Transactions, Programs, and Databases, if they have been discovered.

The default is EXCLUDE.

CICS.TRANSACTION_NAME.transactionprefix
EXCLUDE suppresses output for CICS transaction resources that match the specified transaction prefix for all CICS regions.

The default is EXCLUDE.
**FILT directive**

CICS.FILE_NAME.<fileprefix>
EXCLUDE suppresses output for CICS file resources that match the specified file prefix for all CICS Regions.

The default is EXCLUDE.

CICS.PROGRAM
EXCLUDE suppresses output for CICS program resources.

The default is EXCLUDE.

CICS.PROGRAM.<jobprefix>
EXCLUDE suppresses output for CICS program resources for a specific CICS region.

The default is EXCLUDE.

CICS.PROGRAM_NAME.<programprefix>
EXCLUDE suppresses output for CICS program resources for a specific CICS region.

The default is EXCLUDE.

CICS.PROGRAM_UNUSED
EXCLUDE suppresses output for unused CICS programs.

The default is EXCLUDE.

CICS.TRANSACTION
EXCLUDE suppresses output for CICS transaction resources.

The default is EXCLUDE.

CICS.TRANSACTION.<jobprefix>
EXCLUDE suppresses output for CICS transaction resources for a specific CICS region.

The default is EXCLUDE.

CICS.TRANSACTION_NAME.<transactionnameprefix>
EXCLUDE suppresses output for CICS transaction resources for a specific CICS region.

The default is EXCLUDE.

CICS.TRANSACTIONS_UNUSED
EXCLUDE suppresses output for unused CICS transactions.

The default is EXCLUDE.

CICS.TRANPROG
EXCLUDE suppresses output for CICS program to transaction relationships.

The default is EXCLUDE.

CICS.TRANPROG.<jobprefix>
EXCLUDE suppresses output for CICS program to transaction relationships for a specific CICS region.

The default is EXCLUDE.

CICS.FILE
EXCLUDE suppresses output for CICS file resources.

The default is EXCLUDE.
**CICS.FILE.jobprefix**
EXCLUDE suppresses output for CICS file resources for a specific CICS region.
The default is EXCLUDE.

**IMS**

**IMS.DATABASE**
EXCLUDE suppresses output for IMS database resources.
The default is EXCLUDE.

**IMS.DATABASE.jobprefix**
EXCLUDE suppresses output for IMS database resources for a specific IMS control region.
The default is EXCLUDE.

**IMS.PROGRAM**
EXCLUDE suppresses output for IMS program resources.
The default is EXCLUDE.

**IMS.PROGRAM.jobprefix**
EXCLUDE suppresses output for IMS program resources for a specific IMS control region.
The default is EXCLUDE.

**IMS.REPORT**
EXCLUDE suppresses the report for IMS resources; Transactions, Programs, and Databases, if they have been discovered.
The default is EXCLUDE.

**IMS.TRANSACTION**
EXCLUDE suppresses output for IMS transaction resources.
The default is EXCLUDE.

**IMS.TRANSACTION.jobprefix**
EXCLUDE suppresses output for IMS transaction resources for a specific IMS control region.
The default is EXCLUDE.

**IMS.TRANPROG**
EXCLUDE suppresses output for IMS program to transaction relationships.
The default is EXCLUDE.

**IMS.TRANPROG.jobprefix**
EXCLUDE suppresses output for IMS program to transaction relationships for a specific IMS control region.
The default is EXCLUDE.

**IMS.DATABASE_NAME.databaseprefix**
EXCLUDE suppresses output for IMS database resources that match the specified database prefix for all IMS subsystems.
The default is EXCLUDE.

**IMS.DDB2CONN**
EXCLUDE suppresses output for IMS to DB2 links.
**FILT directive**

The default is EXCLUDE.

**IMS.DEPREG**
EXCLUDE suppresses output for IMS transaction resources that match the specified transaction programs for all IMS Subsystems.

The default is EXCLUDE.

**IMS.PROGRAM_NAME.programprefix**
EXCLUDE suppresses output for IMS program resources that match the specified program prefix for all IMS Subsystems.

The default is EXCLUDE.

**IMS.TRANSACTION_NAME.transactionprefix**
EXCLUDE suppresses output for IMS transaction resources that match the specified transaction prefix for all IMS Subsystems.

The default is EXCLUDE.

**IMS.TRANPROG**
EXCLUDE suppresses output for IMS transaction resources that match the specified transaction programs for all IMS Subsystems.

The default is EXCLUDE.

**IMS.TRANPROG.jobprefix**
EXCLUDE suppresses output for IMS transaction resources that match the specified transaction program job names for all IMS Subsystems.

The default is EXCLUDE.

**DB2**

**DB2.DATABASE**
EXCLUDE suppresses output for DB2 database resources.

The default is EXCLUDE.

**DB2.DATABASE.jobprefix**
EXCLUDE suppresses output for DB2 database resources for a specific DB2 subsystem.

The default is EXCLUDE.

**DB2.DATABASE_NAME.databaseprefix**
EXCLUDE suppresses output for DB2 database resources for a specific DB2 subsystem.

The default is EXCLUDE.

**DB2.REPORT**
EXCLUDE suppresses the report for DB2 REPORT reports.

The default is INCLUDE.

**DB2.TABLESPACE**
EXCLUDE suppresses output for DB2 tablespace resources.

The default is EXCLUDE.

**DB2.TABLESPACE.jobprefix**
EXCLUDE suppresses output for DB2 tablespace resources for a specific DB2 subsystem.

The default is EXCLUDE.
DB2.TABLESPACE_NAME.tablespaceprefix
EXCLUDE suppresses output for DB2 database resources that match the specified database prefix for all DB2 Subsystems.
The default is EXCLUDE.

Storage

STORAGE.REPORT
EXCLUDE suppresses the report for Storage Volume resources if they have been discovered.
The default is EXCLUDE.

STORAGE.VOLUME
EXCLUDE suppresses output for Storage Volume resources.
The default is EXCLUDE.

STORAGE.VOLUME.volumeprefix
EXCLUDE suppresses output for Storage volumes that match the specified volume prefix.
The default is EXCLUDE.

MQ

MQ.AUTHINFO
Filter MQ Authinfo.
The default is EXCLUDE.

MQ.AUTHINFO.jobprefix
The default is EXCLUDE.

MQ.AUTHINFO_NAME.jobprefix
The default is EXCLUDE.

MQ.BUFFERPOOL
The default is EXCLUDE.

MQ.BUFFERPOOL.jobprefix
The default is EXCLUDE.

MQ.BUFFERPOOL_NAME.jobprefix
The default is EXCLUDE.

MQ.CHANNEL
EXCLUDE suppresses output for all MQ channel resources.
The default is EXCLUDE.

MQ.CHANNEL.jobprefix
EXCLUDE suppresses output for MQ channel resources for a specific MQ subsystem.
The default is EXCLUDE.

MQ.CHANNEL_CLUSTERS
EXCLUDE suppresses output for all MQ channel cluster resources.
The default is EXCLUDE.
MQ.CHANNEL_CONNECT
EXCLUDE suppresses output for all MQ channel connect resources.
The default is EXCLUDE.

MQ.CHANNEL_NAME.channelprefix
EXCLUDE suppresses output for MQ channel resources that match the specified channel prefix for all MQ subsystems.
The default is EXCLUDE.

MQ.CHANNEL_RQSTSRVR
EXCLUDE suppresses output for all MQ channel request server resources.
The default is EXCLUDE.

MQ.CHANNEL_SNDRRCVR
EXCLUDE suppresses output for all MQ channel sender receiver resources.
The default is EXCLUDE.

MQ.NAMELIST
EXCLUDE suppresses output for all MQ namelist resources.
The default is EXCLUDE.

MQ.NAMELIST_NAME.jobprefix
EXCLUDE suppresses output for all MQ namelist resources.
The default is EXCLUDE.

MQ.NAMELIST.channelprefix
EXCLUDE suppresses output for all MQ namelist resources.
The default is EXCLUDE.

MQ.PAGESET
EXCLUDE suppresses output for all MQ pageset resources.
The default is EXCLUDE.

MQ.PAGESET.jobprefix
EXCLUDE suppresses output for all MQ pageset resources.
The default is EXCLUDE.

MQ.PROCESS
EXCLUDE suppresses output for all MQ process resources.
The default is EXCLUDE.

MQ.PROCESS.jobprefix
EXCLUDE suppresses output for all MQ process resources.
The default is EXCLUDE.

MQ.PROCESS_NAME.jobprefix
EXCLUDE suppresses output for all MQ process name resources.
The default is EXCLUDE.

MQ.QUEUE
EXCLUDE suppresses output for all MQ queues. This takes precedence over the queue type filters.
FILT directive

The default is EXCLUDE.

MQ.QUEUE.jobprefix
EXCLUDE suppresses output for MQ queue resources for a specific MQ subsystem.
The default is EXCLUDE.

MQ.QUEUE_ALIAS
EXCLUDE suppresses output for all MQ queue alias resources.
The default is EXCLUDE.

MQ.QUEUE_ALIAS.qType
EXCLUDE suppresses output for all MQ alias resources.
The default is EXCLUDE.

MQ.QUEUE_LOCAL
EXCLUDE suppresses output for all MQ local resources.
The default is EXCLUDE.

MQ.QUEUE_LOCAL.qType
EXCLUDE suppresses output for all MQ local resources.
The default is EXCLUDE.

MQ.QUEUE_MODEL
EXCLUDE suppresses output for all MQ alias resources.
The default is EXCLUDE.

MQ.QUEUE_MODEL.qType
EXCLUDE suppresses output for all MQ model resources.
The default is EXCLUDE.

MQ.QUEUE_NAME.queueprefix
EXCLUDE suppresses output for MQ queue resources that match the specified queue prefix for all MQ subsystems.
The default is EXCLUDE.

MQ.QUEUE_REMOTE
EXCLUDE suppresses output for all MQ remote queue resources.
The default is EXCLUDE.

MQ.QUEUE_REMOTE.qType
EXCLUDE suppresses output for all MQ remote queue resources.
The default is EXCLUDE.

MQ.QUEUE_REPORT
EXCLUDE suppresses output for all MQ report resources.
The default is EXCLUDE.

MQ.STGCLASS
EXCLUDE suppresses output for all MQ STG class resources.
The default is EXCLUDE.

MQ.STGCLASS.jobprefix
EXCLUDE suppresses output for all MQ STG class resources.
The default is EXCLUDE.
MQ.STGCLASS_NAME.jobprefix
   EXCLUDE suppresses output for all MQ STG class name resources.
   The default is EXCLUDE.

**CUSTSRVR directive**

The CUSTSRVR directive is used to identify the name of the custom server and the criteria that should be used in identifying the custom server.

Syntax: CUSTSRVR name.type.criteria appdesc_loc

Where:

name
   The name of the custom server

type
   One of the following types of custom server:
      • APPSERVER - AppServer
      • J2EESERVER - J2EE Server
      • WEBSERVER - Web Server
      • DBSERVER - Database Server
      • MQSERVER – MQ Queue Manager

criteria
   The name of the CRITERIA directives to use

appdesc_loc
   The location of the Application Descriptor file

During the discover process, if the file designated in the appdesc_loc fields is unable to be read, the following messages will be displayed in the log and a return code of 8 will be returned:

IRX0555E The input or output file IZDFILE is not allocated.
   It cannot be opened for I/O
IRX0670E EXECIO error while trying to GET or PUT a record.
IZD0007E EXECIO DISKR from TADDM.APPDESC.PDS failed - rc: 20

**CRITERIA directive**

The CRITERIA directive is used to define the criteria that will be used to uniquely identify this address space.

Syntax: CRITERIA criteria.crittype compare;value

Where:

criteria
   The name of the criteria signature

crittype
   One of the following types of criteria to be used:
      • JOBNAME
      • JOBTYPE
      • PROGRAM
      • ARGUMENT
      • USER

compare
   The comparison operator
CRITERIA directive

**JOBNAME**
- is, starts

**JOBTYPE**
- is
  - STARTED
  - SYSTEM
  - BATCH
  - TSOUSER

**PROGRAM**
- is, starts (starts with), ends (ends width)

**ARGUMENT**
- is, contains, starts (starts-with) , ends (ends-with)

**USER**
- is, contains, starts (starts-with) , ends (ends-with)

**value**
The value to be compared against

Example 1:
CUSTSRVR Payroll.OBSERVER.EOMDB TADDM.APPDSEC.PDS(PAY001)
CRITERIA EOMDB.PROGRAM is;AAABBBCC
CRITERIA EOMDB.ARGUMENT contains;stack=yes
CRITERIA EOMDB.USER is;MYUSER
CRITERIA EOMDB.ARGUMENT starts;c:\aa
CRITERIA EOMDB.ARGUMENT ends;test.xml

Example 2:
CUSTSRVR ATM.APPSERVER.ATMCR TADDM.APPDESC.PDS(ATMCFG)
CRITERIA ATMCR.PROGRAM is;ATMSRV

**FILES directive**
The FILES directive is used to specify the files that the z/OS DLA reads and associates with discovered resources. This directive impacts Report and Appdesc objects provided in books. z/OS DLA will create Report and Appdesc objects for most installations. However, if you have customized standard data sets, you might need to indicate that customization in FILES directives.

Syntax: FILES objtype.filetype.job dsn;report

Where:
- **objtype**
  - One of the following object types:
    - ZOS
    - TASK
    - IMS
    - CICS
    - DB2
    - MQ
    - WEBSPHERE

- **filetype**
  - One of the following file types:
    - REPORT
    - APPDESC
**FILES directive**

*jobprefix*
Optional jobname prefix criteria.

*dsn*
The name of the data set to read.

*report*
Optional report name. The default is the name of the data set.

Example:

```plaintext
*FILES ZOS.REPORT.dataset;report_name
*FILES TASK.REPORT.jobprefix dataset;report_name
*FILES MQ.REPORT.jobprefix dataset;report_name
*FILES MQ.APPDESC.jobprefix dataset
*FILES IMS.REPORT.jobprefix dataset;report_name
*FILES IMS.APPDESC.jobprefix dataset
*FILES DB2.REPORT.jobprefix dataset;report_name
*FILES DB2.APPDESC.jobprefix dataset
*FILES CICS.REPORT.jobprefix dataset;report_name
*FILES CICS.APPDESC.jobprefix dataset
*FILES WEBSHERE.REPORT.jobprefix dataset;report_name
*FILES WEBSHERE.APPDESC.jobprefix dataset
```

The REPORT files are shown in the TADDM Report tab in the z/OS and Subsystem Detail View pane.

The APPDESC files are TADDM Application Descriptor files, which can be used to profile a subsystem with a Business Application. See the IBM Tivoli Application Dependency Discovery Manager User’s Guide for more information. The following example is the Application Descriptor XML markup:

```xml
<component-app-descriptor app-instance-name="Order Management">
  <component-descriptor
    type="server"
    name="zApp1"
    functional-group="zApp Tier"
    marker-module="false"
  />
</component-app-descriptor>
```

If this markup is saved into a member in the TADDM.APPDESC.PDS, the FILES MQ.APPDESC TADDM.APPDESC.PDS(*) rule associates all discover MQ subsystems with this AppDescriptor.

**Adding modules to a Business Application**

If you want to include an individual module, the name attribute is significant only if the descriptor refers to a module and not a server. The identifier for the module is the name, and the context in which the application descriptor is being processed is that of the server that it is discovered with. If specific modules are being identified, the name is the only way to identify which modules are being included.

**Adding servers to a Business Application**

If you want to include an entire server in the application, the name is not important. You can create a single component descriptor file for a given application (for example, Order Management) and then deploy that file for whichever CICS Regions, IMS SubSystems, MQ Managers, or DB2 Subsystems you want to be part of that application.

For example, note the following XML markup:
The key point of the example is the functional group name in bold. A single XML file can be used for all components of the Order Management application that are in that functional group (for example, a group called CICS Regions). However, the Business Application must include some MQ Managers or DB2 SubSystems that are not part of that functional group. Those components require a file that differs only in the name of the functional group. For the MQ Manager, the functional group might be named MQ Managers, for example.

In summary, for server type components, you must have only as many versions of application descriptors as there are functional groups.

**Note:** Create the Application Descriptor file with SET NUMS OFF for no line numbers.

**BOOK directive**

The BOOK directive is used to set the book discovery parameter values. This directive can be used to specify which idml books are output from the discovery process.

Syntax: BOOK index value

Supported indexes:

<table>
<thead>
<tr>
<th>Index</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE DB2</td>
<td>Yes creates output books for all discovered DB2 regions.</td>
</tr>
<tr>
<td></td>
<td>The default is No.</td>
</tr>
<tr>
<td>TYPE CICS</td>
<td>Yes creates output books for all discovered CICS regions.</td>
</tr>
<tr>
<td></td>
<td>The default is No.</td>
</tr>
<tr>
<td>TYPE IMS</td>
<td>Yes creates output books for all discovered IMS regions.</td>
</tr>
<tr>
<td></td>
<td>The default is No.</td>
</tr>
<tr>
<td>TYPE MQ</td>
<td>Yes creates output books for all discovered MQ regions.</td>
</tr>
<tr>
<td></td>
<td>The default is No.</td>
</tr>
<tr>
<td>TYPE WEBSHERE</td>
<td>Yes creates output books for all discovered WEBSHERE regions.</td>
</tr>
<tr>
<td></td>
<td>The default is No.</td>
</tr>
<tr>
<td>ZOSALL</td>
<td>Yes creates one output book named ZOSALL for all discovered resources.</td>
</tr>
<tr>
<td></td>
<td>The default is No.</td>
</tr>
</tbody>
</table>
BOOK directive

ZOSBASE
Yes creates one output book named ZOSBASE for all discovered hardware and system data.
The default is Yes.

ZOSTASK
Yes creates one output book named ZOSTASK for all discovered address space data.
The default is Yes.

DEBUG directive
Debug: For IZDRDLA and Included modules, 1 is on.

SET DEBUG 1 sets ALL debug setting to on.

SET DEBUG 0 and SET DEBUG_CICS 1 sets only CICS debug on.

DEBUG
1 to turn debug on, 0 to turn debug off.
The default is 0.

DEBUG_A5
1 to turn debug on, 0 to turn debug off.
The default is 0.

DEBUG_CICS
1 to turn debug on, 0 to turn debug off.
The default is 0.

DEBUG_CS
1 to turn debug on, 0 to turn debug off.
The default is 0.

DEBUG_DB2
1 to turn debug on, 0 to turn debug off.
The default is 0.

DEBUG_DB2_IFI
1 to turn debug on, 0 to turn debug off.
The default is 0.

DEBUG_DB2_SQL
1 to turn debug on, 0 to turn debug off.
The default is 0.

DEBUG_IMS
1 to turn debug on, 0 to turn debug off.
The default is 0.

DEBUG_MQ
1 to turn debug on, 0 to turn debug off.
The default is 0.

DEBUG_OUTPUT
1 to turn debug on, 0 to turn debug off.
The default is 0.

**DEBUG_STORAGE**
1 to turn debug on, 0 to turn debug off.
The default is 0.

**DEBUG_SYSTEM**
1 to turn debug on, 0 to turn debug off.
The default is 0.

**DEBUG_TCPIP**
1 to turn debug on, 0 to turn debug off.

**DEBUG_TIMING**
1 to turn debug on, 0 to turn debug off.
The default is 0.

**DEBUG_WEBSPHERE**
The default is 0.

**DEBUG_IZDDCMDI**
1 to turn debug on, 0 to turn debug off

**DEBUG_IZDHIOSI**
The default is 0.

**DEBUG_IZDMCMDI**
1 to turn debug on, 0 to turn debug off

**DEBUG_IZDPPLI**
The default is 0.

**DEBUG_IZDPSUBI**
1 to turn debug on, 0 to turn debug off

**DEBUG_IZDPSYSI**
The default is 0.

**DEBUG_IZDSXCFI**
The default is 0.

**DEBUG_IZDUDALC**
The default is 0.

**DEBUG_IZDUASCB**
1 to turn debug on, 0 to turn debug off.
The default is 0.

**DEBUG_IZDUAUTH**
1 to turn debug on, 0 to turn debug off

**DEBUG_IZDUPDSR**
The default is 0.

**DEBUG_GROUP**
1 to turn debug on, 0 to turn debug off.
The default is 0.

**DEBUG_OUTPUTBOOK**
1 to turn debug on, 0 to turn debug off.
The default is 0.
DEBUG directive

DEBUG_USR
1 to turn debug on, 0 to turn debug off.
The default is 0.
Appendix A. Installation and customization

Overview

The DLA must be installed or accessible to each z/OS system or sysplex that is the subject of discovery. This involves:

- SMP/E installation of the z/OS DLA product libraries. Details are provided in the z/OS DLA Program Directory (publication number GI11-9449-00).
- Making the product target libraries available to all systems where discovery is required.
- Authorizing the z/OS DLA target load library. The discovery module needs APF authorization to perform a full discovery of all class instances and attributes.
- Customizing the JCL for the DLA job and procedures.
- Allocating a user ID with an OMVS segment to run the DLA.
- Binding the DSNREXX DB2 interface if required. This procedure is described in the IBM DB2 UDB for z/OS V8 Installation Guide.
- Scheduling the DLA batch job to execute.
- Providing SAF (System Authorization Facility) access as required for resources, including WebSphere for MQ resources, DB2 resources, and z/OS system resources.

See the “Installation checklist” on page 63 for a complete checklist of installation tasks.

SMP/E installation

Follow the instructions in the z/OS DLA Program Directory to allocate the required SMP/E control and target datasets and run the RECEIVE, APPLY, and ACCEPT jobs from the hlq.SIZDINST library.

The following data set is allocated as part of the installation process.

SMP/E target data sets

<table>
<thead>
<tr>
<th>Data set</th>
<th>LLQ</th>
<th>DSORG</th>
<th>LRECL</th>
<th>RECFM</th>
<th>Est. # of Members</th>
<th>Est. Tracks</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIZDLOAD</td>
<td>PO</td>
<td>0</td>
<td>U</td>
<td>50</td>
<td>120</td>
<td></td>
<td>z/OS DLA load modules.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>This data set must be APF authorized for full discovery function.</td>
</tr>
<tr>
<td>SIZDEXEC</td>
<td>PO</td>
<td>80</td>
<td>FB</td>
<td>1</td>
<td>10</td>
<td></td>
<td>z/OS DLA REXX code</td>
</tr>
<tr>
<td>SIZDSAMP</td>
<td>PO</td>
<td>80</td>
<td>FB</td>
<td>10</td>
<td>10</td>
<td></td>
<td>z/OS DLA samples.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Customer needs to customize and copy JCL PROCs and to JES PROCLIB data set.</td>
</tr>
<tr>
<td>SIZDMESG</td>
<td>PO</td>
<td>80</td>
<td>FB</td>
<td>1</td>
<td>1</td>
<td></td>
<td>z/OS DLA message templates.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Only English member in this release.</td>
</tr>
<tr>
<td>SIZDMAPS</td>
<td>PO</td>
<td>1024</td>
<td>VB</td>
<td>1</td>
<td>2</td>
<td></td>
<td>z/OS DLA mapping data</td>
</tr>
</tbody>
</table>
**APF authorization**

To permanently authorize the z/OS DLA for APF (Authorized Program Facility) authorization from an MVS TSO session, complete these steps (which require an IPL, but eliminates the requirement to re-authorize after every IPL):

1. Add an entry to the PROG01 data set member (typically found in USER.PARMLIB):
   
   ```
   SETPROG APF,ADD,DSNAME(&DLAHLQ.SIZDLOAD) SMS
   ```

2. Execute an IPL on the z/OS LPAR by using local procedures.

Follow these steps to temporarily APF-authorize the z/OS DLA from an MVS TSO session (which remain in effect until you execute another IPL):

```
=S.LOG
/
SETPROG APF,FORMAT=DYNAMIC Set APF format dynamic
and
SETPROG APF,ADD,DSN=<DLAHLQ>.SIZDLOAD,SMS

D PROG,APF,ALL Confirm which DSNs are APF-authorized
```

You must issue an APF authorization command for z/OS DLA to run.

**System Authorization Facility authorization**

The user ID that runs the z/OS DLA job or STC must have the following SAF (System Authorization Facility) authorities:

- READ access is recommended for all the data sets in the system PARMLIB concatenation used during IPL.
- SAF authorizations are required for WebSphere for MQ resources and commands in order to discover WebSphere for MQ objects. See the "WebSphere for MQ Customization" section for a description of the required SAF steps for WebSphere for MQ.
- SAF authorizations are required for DB2 resources and commands in order to discover DB2 objects. See the "DB2 Customization" section for a description of the required SAF steps for DB2.
- READ access is recommended to WebSphere configuration files.
- An OMVS (USS) segment with authority to issue the netstat, host, and home commands.

**WebSphere for MQ customization**

The following customization is required for all Websphere for MQ systems where discovery is run:
WebSphere for MQ customization

- The z/OS DLA uses a batch connection to communicate with Websphere for MQ. See the IBM WebSphere for MQ System Setup Guide for additional information about how to set up connection security profiles for a batch connection.

- The z/OS DLA opens and puts messages into the SYSTEM.COMMAND.INPUT queue by using the MQOO.OUTPUT option on the MQOPEN API and the MQPMO_DEFAULT_CONTEXT option on the MQPUT API. See the IBM WebSphere for MQ System Setup Guide for additional information about how to set up queue security profiles.

- The z/OS DLA opens the SYSTEM.COMMAND.REPLY.MODEL queue to define dynamic queues to receive response messages for commands that are issued. These dynamic queues are defined by using the MQOO_SAVE_ALL_CONTEXT and MQOO_INPUT_EXCLUSIVE options on the MQOPEN API. See the IBM WebSphere for MQ System Setup Guide for additional information about how to set up queue security profiles.

  Note: The names of the dynamic queues created by the z/OS DLA are prefixed with the ‘RXMQVC’ fixed value.

- The z/OS DLA receives messages from the dynamic queues created in the previous step. These queues are then deleted by using the MQCLOSE API with the ‘MQCO_DELETE_PURGE’ option. See the IBM WebSphere for MQ System Setup Guide for additional information about how to set up queue security profiles.

- The z/OS DLA issues the following commands to the discovered queue managers:
  - DISPLAY QMGR
  - DISPLAY CLUSQMGR
  - DISPLAY AUTHINFO
  - DISPLAY CHANNEL
  - DISPLAY NAMELIST
  - DISPLAY USAGE
  - DISPLAY PROCESS
  - DISPLAY QUEUE
  - DISPLAY STGCLASS

  See the IBM WebSphere for MQ System Setup Guide for additional information about how to set up command security profiles.

DB2 customization

The following customization is required for all DB2 systems where discovery is run:

- z/OS DLA uses the DSNREXX DB2 REXX interface to issue SQL commands from REXX. The DSNREXX interface must be bound to each DB2 subsystem where discovery is required. See the DB2 Installation for more information about how to bind the packages for DB2 REXX language support.

- The z/OS DLA user ID must have authority to issue the following DB2 DISPLAY commands:
  - DISPLAY GROUP
  - DISPLAY DDF
  - DISPLAY DB
DB2 customization

- The z/OS DLA user ID must also have the authority to read the following SYSIBM tables:
  - SYSIBM.SYSDATABASE
  - SYSIBM.SYSTABLES
  - SYSIBM.SYSTABLESPACE
  - SYSIBM.SYSINDEXES
  - SYSIBM.SYSCOLUMNS

The sample JCL IZDJGRANT is included in the SIZDSAMP library to assist in granting these authorities to the z/OS DLA user ID. For more information on granting command and table authority, see the *IBM DB2 Administration Guide*.

- If multiple versions of DB2 are installed on the operating system, a STEPLIB (or LINKLST entry) for the SDSNLOAD data set matching the DB2 version of the discovery target must be supplied. Order the SDSNLOAD library data sets in the STEPLIB/LINKLST concatenation from the latest level (first) to the earliest level (last).

JCL customization

Consider the following restrictions for JCL customization:

- The JCL member IZDJRUN in hlq.SIZDSAMP must be customized to specify the correct HLQ and data set names for the z/OS DLA target load library and the output IdML file.

- The PROC member IZDPDISC has these restrictions:
  - This member must be copied to a JES PROCLIB if discovery is run as a started task or if the JCLLIB JCL statement is omitted from the IZDJRUN JCL.
  - The allocation for the data set referenced in the IZDIDML DD statement must be changed to accommodate the number and size of the generated output BOOKS.

For best practices, the primary allocation should be one cylinder for each subsystem discovered. The following subsystems are included:
- CICS
- DB2
- IMS
- MQ
- WebSphere

For example, if 150 subsystems will be discovered, use the following space allocation:

```
SPACE=(CYL,(150,20,50))
```

Prefix this directory with "IDML" (for example, IDMLDSN=<HLQ>.ZOSDLA.IDML.SYSL)

Codepage

By default the z/OS DLA outputs XML files in EBCDIC codepage 1047. If your site does not use 1047, the IZDC* members in SIZDSAMP must be updated to define the appropriate codepage. For example, many customers in Europe and the Middle East must define SET CODEPAGE 1148 in the z/OS DLA configuration members.

Failure to set the correct codepage can lead to incorrect EBCDIC to ASCII text translation, when the XML files are download to distributed platforms for loading into CCMDB etc. For example, the z/OS DLA files contain XML comments in <!--
WebSphere customization

READ access is recommended to WebSphere configuration files. All of the .XML files from the CONFIG_ROOT/cells/CELL_NAME directory are included in the output Books as report files. The CONFIG_ROOT and CELL_NAME are determined from the BBOENV DD card in the WebSphere JCL.

Installation checklist

__ 1. SMPE install the z/OS DLA datasets. This process is detailed in the z/OS DLA Program Directory (publication number GG11-9449-00).

__ 2. Schedule a rollout of the DLA run-time environment to all z/OS images that will be the target of discovery.

__ 3. Roll out the DLA run-time environment to all target z/OS images:
   __ a. Make the target DLA datasets available on each z/OS image.
   __ b. Update the IPL process to APF-authorize the SIZDLOAD data set on each z/OS image (Optional for PoC phase if the site allows datasets to be APF authorized via the SETPROG z/OS command)
   __ c. Add IZDPCDISC JCL PROC to a system PROCLIB data set on each z/OS image (Optional for PoC phase).

__ 4. DB2 customization is required for all DB2 subsystems that are the target of discovery:
   __ a. The DSNREXX interface must be bound to each DB2 subsystem where discovery is required. See the IBM DB2 UDB for z/OS V8 Installation Guide for more information about how to bind the packages for DB2 REXX language support.
   __ b. The z/OS DLA user ID must have authority to issue the following DB2 DISPLAY commands and have read access to SYSIBM tables. Use the sample JCL IZDJGRANT in the SIZDSAMP library to assist in granting these authorities to the z/OS DLA user ID.
   __ c. If multiple versions of DB2 are installed on the operating system, a STEPLIB (or LINKLST entry) for the SDSNLOAD data set matching the DB/2 version of the discovery target must be supplied.

__ 5. Define security access as required:
   __ a. The user ID that runs the z/OS DLA job or STC must have the following SAF authorities:
      __ 1) READ access is recommended for all the data sets in the system PARMLIB concatenation used during IPL.
      __ 2) If you have WebSphere for MQ resources, ensure that all the required SAF authorities have access to run WebSphere for MQ discoveries.
      __ 3) If RACF is used to protect DB2 resources then authority is required to issue DB2 DISPLAY commands (including DISPLAY DB and DISPLAY GROUP) and to access (READ) SYSIBM resources using dynamic SQL.

__ 6. READ access is recommended to WebSphere HFS configuration files.
7. The user ID that runs the discovery job must have an OMVS segment with authority to issue the netstat, host, and home commands.
Appendix B. Messages and codes

Messages

IZD0000W  MESSAGE message_number UNDEFINED
Explanation: The indicated message number is not defined in member IZDMSGEN in SIZDMSGS.
Message Variables:
message_number
The number of the message for which text was missing.
System action: Processing stops.
System programmer response: Check that SIZDMSGS is correctly specified in the Discovery JCL.

IZD0001I  text
Explanation: This message is a general informational message.
System action: Processing continues.
Operator response: None.
System programmer response: None.

IZD0002I  z/OS DLA Started date time
Explanation: z/OS DLA processing has started. The date and time are provided for correlation.
Message Variables:
date The current date.
time The current time.
System action: Processing continues.

IZD0003I  z/OS DLA Completed date time
Explanation: z/OS DLA processing has completed. The date and time are provided for correlation.
Message Variables:
date The current date.
time The current time.
System action: Processing stops.
System programmer response: Contact IBM Software Support.

IZD0004E  condition TRAPPED ON LINE
line_number The REXX line number where the error occurred.
return_code The REXX return code.
description A description of the error.
System action: Processing stops.
System programmer response: Contact IBM Software Support.

IZD0005E  FATAL ERROR - UNABLE TO CONTINUE
Explanation: A fatal error occurred during REXX processing.
System action: Processing stops.
System programmer response: Contact IBM Software Support.

IZD0006I  z/OS DLA Processing completed successfully
Explanation: Processing has completed successfully.

IZD0007E  EXECIO DISKR FROM ddbname FAILED
- RC - return_code
Explanation: An I/O error has occurred reading from the indicated DD during REXX processing.
Message Variables:
ddbname The name of the output DD.
return_code The EXECIO return code.
System action: Processing stops.
System programmer response: Contact IBM Software Support.

IZD0008E  EXECIO DISKW FROM ddbname FAILED
- RC - return_code
Explanation: An I/O error has occurred writing to DD ddbname during REXX processing.
Message Variables:
ddbname The name of the output DD.
return_code The EXECIO return code.
System action: Processing stops.
Messages

System programmer response: Contact IBM Software Support.

IZD009W DISCOVERY ROUTINE module_name RETURNED NO DATA

Explanation: A module returned no data in the output stem.

Message Variables:
module_name
The name of the module that returned a null stem variable.

System action: Processing continues but no data was discovered. This might lead to errors in the idml output.

System programmer response: Contact IBM Software Support.

IZD0010W DISCOVERY ROUTINE module_name GAVE BAD RETURN CODE, RC return_code

Explanation: A module returned the indicated return code.

Message Variables:
module_name
The name of the module that returned a null stem variable.
return_code
The return code.

System action: Processing continues but discovery data might be incomplete.

System programmer response: If the module_name is Discover_CICS, then this can be caused by insufficient storage to continue processing. The following message might also be in the job log:
ID1000S IDI1ADD:629 Storage allocation for 18 (X'12') bytes failed - processing terminated

If this is the case, increase the value for the REGION parameter in the DLA JCL. Otherwise, contact IBM Software Support.

IZD0011W UNRECOGNIZED IMS RELEASE: control_region - version

Explanation: IMS discovery routines have detected an unsupported IMS release in the system. Only currently supported IMS versions are supported by DLA.

Message Variables:
control_region
The name of the IMS control region in error.
version
The version of the IMS control region.

System action: Processing continues but no data is returned for the named control region and any associated dependent regions.

IZD0012W ERROR DATA RETURNED BY: module_name (ssid debug command)

Explanation: DB2 discovery routines have encountered an error while attempting to issue a command using the DB2 IFI interface.

Message Variables:
module_name
The name of the module in error.
ssid
The DB2 subsystem ID.
debug
Any debug data returned.
command
The command that was issued using the API that failed.

System action: Processing continues but discovery data might be incomplete.

System programmer response: Message IZD0013W should follow this message with additional data.

IZD0013W data

Explanation: DB2 discovery routines have encountered an error while attempting to issue a command using the DB2 IFI interface.

Message Variables:
data
The data returned from the IFI interface.

System action: Processing continues but no data is returned for the named DB2 subsystem.

System programmer response: See message IZD0012W for additional data.

IZD0014W UNABLE TO ACCESS dataset_name - SYSDSN gave; sysdsn_data

Explanation: An attempt was made to read a data set or member but the REXX SYSDSN() function returned an error.

Message Variables:
dataset_name
The name of the data set returned from the IFI interface.
sydsns_data
The data returned by SYSDSN().

System action: Processing continues but no data is returned from the named data set.

System programmer response: Check that the data set exists and that the user that runs the DLA job has sufficient RACF authority to read the data set.

IZD0015W UNABLE TO FIND DDNAME ddname IN allocation_string

Explanation: An attempt to allocate a data set failed.

Message Variables:/ddname
The name of the data definition statement.
allocation_string

The allocation details.

System action: Processing continues.

System programmer response: Check the DLA JCL DD statements for the correct IdML output data set specification.

IZD0016W UNABLE TO OPEN PATH path_name -
RC return_code

Explanation: An attempt to open path path_name failed.

Message Variables:
path_name The name of the path.
return_code The return code that was received.

System action: Processing continues.

System programmer response: Check that the user running the DLA job has the correct OMVS segment and sufficient RACF authority to read the path.

IZD0017W NO DATA READ FOR file_name DD ddname

Explanation: No data was returned after the indicated file name was read.

Message Variables:
file_name The name of the file.
ddname The name of the DD.

System action: Processing continues.

System programmer response: Check that the user running the DLA job has the correct OMVS segment and sufficient RACF authority to read the file.

IZD0018W UNABLE TO FIND VALUE FOR file_name IN job_name ddname

Explanation: No data was returned after the indicated file name was read.

Message Variables:
file_name The name of the file.
job_name The name of the job.
ddname The name of the DD.

System action: Processing continues.

IZD0019W FATAL ERROR - UNABLE TO FIND target

Explanation: A fatal error occurred.

Message Variables:
target The host name or IPV4 address.

IZD0020I DISCOVERY OF discovery_data SUPPRESSED BY RULE

Explanation: Discovery for the discovery data has been suppressed by the user.

Message Variables:
discovery_data The data that will not be discovered.

System action: Processing continues.

IZD0021E APF DISCOVERY REQUESTED BUT NOT RUNNING APF AUTHORIZED

Explanation: The DISC APF flag has been set to 1 but the SIZDLOAD library is not APF-authorized.

System action: Processing stops.

System programmer response: Either authorize the SIZDLOAD library or change the DISC APF flag to 0.

IZD0022I date time: function

Explanation: Discovery for function has either started or ended.

Message Variables:
date The date that the function started or ended.
time The time that the function started or ended.
function The name of the discovery function.

System action: Processing continues.

IZD0023I CHECKSUM DATA UPDATED

Explanation: A change was detected between the current discovery and the previous discovery.

System action: Processing is continued. The checksum and the associated output book are updated.

IZD0024E CHECKSUM CLEARED; BAD RUN

Explanation: Discovery has failed and checksum data for all output members has been cleared.

System action: Processing stops.

System programmer response: Determine why discovery failed and rerun the discovery. By clearing the checksum data, the system ensures new data is output on the next correct run.

IZD0025I OUTPUT OF BOOK book_name SKIPPED: CHECKSUM MATCHES

Explanation: Discovery has detected no change between the checksum for book book_name and the newly calculated checksum for the same book. No output book is written because the book data has not changed.
Messages

Message Variables:

book_name
The name of the IDML book for which no change was detected.

System action: Processing continues but no output book is written.

IZD0026I  UNEXPECTED RESPONSE TO COMMAND: command

Explanation: Discovery has received an unexpected response to an MQ command issued using the MQ API command interface. Message IZD0027I follows this message.

Message Variables:

command
The name of the MQ command.

System action: Processing continues.

System programmer response: Check for additional messages to determine why the MQ command might be incorrectly processed. Turn on debug for MQ and IZDMCMDI.

IZD0027I  command_data

Explanation: Discovery has received an unexpected response to an MQ command issued using the MQ API command interface. Message IZD0026I precedes this message.

Message Variables:

command_data
The data returned from the MQ command.

System action: Processing continues.

System programmer response: Check for additional messages to determine why the MQ command might be incorrectly processed. Turn on debug for MQ and IZDMCMDI.

IZD0028I  POSSIBLE CAUSE: DSNREXX NOT BOUND TO DB2

Explanation: This message is preceded by message IZD0010W and indicates a possible reason for the DB2 request failure.

System action: Processing continues.

System programmer response: DSNREXX must be bound to DB2 before discovery can issue DB2 SQL requests. If DSNREXX is bound to DB2, then check for additional messages to determine why the DB2 request might have failed. Turn on debug for DB2 and IZDDCMDI.

IZD0029I  POSSIBLE CAUSE: NOT AUTHORIZED

Explanation: This message is preceded by message IZD0010W and indicates a possible reason for the DB2 request failure.

System action: Processing continues.

System programmer response: The discovery user must have DB2 or RACF authority to issue SQL requests to query SYSIBM resources and issue DB2 DISPLAY commands. If the correct authorities are granted, then check for additional messages to determine why the DB2 request might have failed. Turn on debug for DB2 and IZDDCMDI.

IZD0030I  discovery_type DISCOVERY SUPPRESSED

Explanation: Discovery has been suppressed for discovery_type because of a DISC directive.

System action: Processing continues.

System programmer response: The discovery user must have DB2 or RACF authority to issue SQL requests to query SYSIBM resources and issue DB2 DISPLAY commands. If the correct authorities are granted, then check for additional messages to determine why the DB2 request might have failed. Turn on debug for DB2 and IZDDCMDI.

IZD0031I  DATA IN IDML FILE TRUNCATED: INCREASE OUTPUT LRECL

Explanation: Discovery has generated IdML output records greater than the output LRECL for the IZDIDML output file.

System action: Processing continues but IdML output truncation has occurred.

System programmer response: Increase the LRECL on the IZDIDML output DD.

IZD0032I  FTP CHANGED ftp_type DATA UPDATED

Explanation: The FTP control statements for type ftp_type have been updated.

System action: Processing continues.

System programmer response: New data is ready to be transferred using FTP.

IZD0033I  FTP CHANGED ftp_type DATA CLEARED: BAD RUN

Explanation: The FTP control statements for type ftp_type have been cleared because of a bad discovery run.

Message Variables:
ftp_type  The type of FTP control statements that have been cleared. This value can be GET or PUT.

**System action:**  Processing stops.

**System programmer response:**  The FTP control statements are cleared to ensure no FTP data transfer occurs.

IZD0034I  FTP ALL ftp_type DATA UPDATED

**Explanation:**  The FTP control statements for type ftp_type have been updated.

**Message Variables:**  ftp_type

- The type of FTP control statements that have been cleared. This value can be GET or PUT.

**System action:**  Processing continues.

**System programmer response:**  New data is ready to be transferred using FTP.

IZD0035I  INCORRECT VALUE FOR directive specified: value

**Explanation:**  An incorrect directive value has been specified as input to discovery.

**Message Variables:**

- directive
  - The discovery directive.
- value
  - The value specified.

**System action:**  Processing continues.

**System programmer response:**  Correct the input parameter value and rerun the discovery.

IZD0036I  UNRECOGNIZED INPUT PARAMETER FOUND: parameter

**Explanation:**  An incorrect input parameter has been specified as input to discovery.

**Message Variables:**

- parameter
  - The parameter that was specified.

**System action:**  Processing continues.

**System programmer response:**  Correct the input parameter and rerun the discovery.

IZD0037I  OUTPUT LIMIT: limit_specified OF resources_found POTENTIAL resource_type

**Explanation:**  An output limit has been specified using a LIMIT directive and more than the limited number of resources had been discovered.

**Message Variables:**

- limit_specified
  - The output limit specified with the LIMIT directive.
- resources_found
  - The number of resources or resource type discovered.
- resource_type
  - The resource type that was discovered.

**System action:**  Processing continues.

**System programmer response:**  Optionally increase the LIMIT specified for this resource.

IZD0038I  POSSIBLE CAUSE: WRONG DB2 SDSNLOAD IN STEPLIB/LINKLST

**Explanation:**  This message is preceded by message IZD0010W and indicates a possible reason for the DB2 request failure.

**System action:**  Processing continues.

**System programmer response:**  Ensure the DB2 SDSNLOAD library in the STEPLIB or LINKLST matches the DB2 version of the DB2 subsystem being discovered. If the correct SDSNLOAD is accessible to discovery, then check for additional messages to determine why the DB2 request might have failed. Turn on debug for DB2 and IZDDCMDI.

IZD0039I  POSSIBLE CAUSE: UNSUPPORTED DB2 LEVEL

**Explanation:**  This message is preceded by message IZD0010W and indicates a possible reason for the DB2 request failure.

**System action:**  Processing continues.

**System programmer response:**  Discovery supports DB2 version 8 and later. If a supported version of DB2 is being discovered, then check for additional messages to determine why the DB2 request might have failed. Turn on debug for DB2 and IZDDCMDI.

IZD0040I  INVALID CODEPAGE number SPECIFIED. DEFAULT CODEPAGE 37 ASSUMED

**Explanation:**  The codepage specified was not found on the system.

**Message Variables:**

- number
  - The codepage that was specified by the user.

**Operator response:**  Correct the codepage specification.

IZD0041I  BOOK FOR disc_type disc_jobname SKIPPED: DISCOVERY FORBIDDEN

**Explanation:**  An output limit has been specified using a LIMIT directive and more than the limited number of resources had been discovered.

**Message Variables:**

- disc_type
  - The type of discovery being processed.
**Messages**

*disc_jobname*

The job name or STC name of the *disc_type*
that is being processed.

**System action:** Processing continues but the indicated
book is not output.

**System programmer response:** There is a
contradiction between the DISC directive and the
BOOK directive for the indicated discovery type in the
input parameter definitions. Correct the definitions and
rerun discovery.

---

**IZD0042I**  
*date time: Discovery skipped for subsystem name*

**Explanation:** The indicated job name has been skipped
for discovery. Message IZD0043I should follow this
message.

**Message Variables:**

name The name of the job that was not discovered.

**System action:** The subsystem is skipped and
discovery continues.

---

**IZD0043I**  
*Cause: SUBSYSTEM NOT CURRENTLY ACTIVE*

**Explanation:** A verification of the subsystem failed.
The subsystem might not be currently active in the
system.

**System action:** The subsystem is skipped and
discovery continues.

---

**IZD0044I**  
*Possible Cause: NOT AUTHORIZED FOR DSNREXX PLAN*

**Explanation:** The access to DB2 failed. The DLA job
might not have the authority to access the DSNREXX
plan.

**System action:** Resources are not accessed.

**System programmer response:** Grant the user the
authority to access the DSNREXX plan. See “DB2
customization” on page 61.

---

**IZD0045I**  
*count resource_type discovered for subsystem*

**Explanation:** This message displays the number of this
type of resources that have been discovered for this
subsystem.

**Message Variables:**

*count* The number of resources discovered for this
*subsystem*.

*resource_type* The type of resources that have been
discovered for this subsystem.

---

**IZD0046I**  
*count resource_type skipped discovery for subsystem*

**Explanation:** The number of this resource type that
have been skipped during the discovery process. The
resource type DISC directive was set to NO.

**Message Variables:**

*count* The number of resources skipped for this
*subsystem*.

*resource_type* The type of resources that has been skipped
for this subsystem.

*subsystem* The name of the subsystem that is being
discovered.

**System action:** The skipped resources will not be
available in a Report File for this subsystem.

---

**IZD0047I**  
*count resource_type suppressed by filter for subsystem*

**Explanation:** The number of this resource type that
have been ignored during the discovery process based
on filtering. The resource type FILT directive was
ENABLED.

**Message Variables:**

*count* The number of resources that were filtered out
for this subsystem.

*resource_type* The type of resources that has been filtered for
this subsystem.

*subsystem* The name of the subsystem that is being
discovered.

**System action:** The filtered resources will not be
available in the Report File or as individual CIs in the
generated books.

---

**IZD0050I**  
*DB2 Command Interface (IFI) not authorized for command command on Subsystem subsystem*

**Explanation:** The access to DB2 resources failed. The
DLA job might not have the authority to perform the
command.

**Message Variables:**

*command* The command being attempted.

*subsystem* The subsystem that was being discovered.
### Messages

**IZD0051I** DB2 Discovery encountered condition will retry

**Explanation:** During a request to the DB2 subsystem, it encountered a condition which caused the request to fail.

**Message Variables:**
- condition
  - The type of condition which caused the request to fail. This value is either timeout or deadlock.

**System action:** The request is attempted again.

**IZD0052I** DB2 Discovery encountered condition retries exhausted

**Explanation:** The retries of the request have been exhausted.

**Message Variables:**
- condition
  - The type of condition causing the issue. This value is either timeout or deadlock.

**System action:** The request failed and the information requested was unable to be discovered. Discovery continues.

**IZD0060W** Dataset DSN=dataset_name VOL=volume failed READ-ONLY check RC=return_code

**Explanation:** During the process of creating the TASKLIB for the IMS or DB2 subsystems, the check for READ-ONLY access failed.

**Message Variables:**
- dataset_name
  - The name of the data set.
- volume
  - The volser of the data set.
- return_code
  - The return code received.

**System programmer response:** If this data set is required to obtain the information that is being requested, the READ-ONLY access must be given to the discovery job.

**IZD0070I** Address Space jobname available for Custom Server processing, listening on Port port_number

**Explanation:** The indicated job is available for custom server processing.

<table>
<thead>
<tr>
<th>Message Variables:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>jobname</strong></td>
</tr>
<tr>
<td><strong>port_number</strong></td>
</tr>
</tbody>
</table>

**IZD0080W** SOCKET request failed rc=return_code errno_name function more_data

**Explanation:** The indicated IP request failed during processing.

**Message Variables:**
- request
  - One of the following IP socket requests issued:
    - INIT
    - GETHOSTID
    - GETHOSTNAME
    - GETDOMAINNAME
- return_code
  - The request return code received
- errno_name
  - The TCP/IP errno name for the return code
- function
  - The name of the function that failed
- more_data
  - Additional text related to the error

**System action:** Processing continues but the information request is not available.

**System programmer response:** The return codes are standard TCP/IP return codes. If you received return code 1004 (EIBMUCVERR), request failed due to IUCV error, this can be caused when the required VMCF and TNF services are not started. If these services cannot be started or if the TCP/IP information cannot be obtained, you might need to set the following TCP/IP variables:

- **SET HOSTNAME**
  - host1
- **SET HOSTDOMAIN**
  - domain1.com
- **SET HOSTIPADDR**
  - 1.2.3.4

**IZD0081W** SOCKET calls skipped

**Explanation:** An unrecoverable error occurred during SOCKET INIT processing. See the IZD0080W message for more information.

**System programmer response:** All of the SOCKET calls have been skipped. The DLA uses the values defined in the SET HOSTNAME, SET HOSTDOMAIN, and SET HOSTIPADDR directives. If these directives are not set, the DLA ends.
Messages

IZD0100W  Duplicate id resource_id
Explanation: During the process of creating the BOOK, a duplicate resource ID was detected.
Message Variables:
resource_id
The ID of the duplicate resource.
System programmer response: Determine why a duplicate resource ID was detected. If necessary, contact IBM Software Support.

IZD0110I  Output of book book skipped; checksum matches
Explanation: The book specified was not written because the checksum matches the previous execution.
Message Variables:
book
The name of the book that was not written.

IZD0111I  Output of book book skipped; subsystem not active
Explanation: The book specified was not written because the subsystem was not active during discovery.
Message Variables:
book
The name of the book that was not written.
System action: No information was gathered.

IZD1000E  modname SYSEVENT MACRO FAILED
RC = return_code
Explanation: The SYSEVENT(QVS) macro has failed.
Message Variables:
modname
The name of the module that issued the SYSEVENT macro.
return_code
The SYSEVENT return code received.
System action: Processing stops.
System programmer response: Return codes for the SYSEVENT (QVS) macro are documented in z/OS MVS Authorized Assembler Services Reference. Check that the SYSEVENT macro is available to the DLA job using the STEPLIB or LNKLIST.

IZD1003E  modname CAF LOAD OF DSNALI FAILED RC = return_code RS = reason_code
Explanation: The DB2 module DSNALI could not be loaded.
Message Variables:
modname
The name of the module that issued the LOAD for DSNALI.
return_code
The LOAD return code received.
reason_code
The LOAD reason code received.
System action: Processing stops.
System programmer response: Return codes for the LOAD macro are documented in z/OS MVS Authorized Assembler Services Reference. Check that the DSNALI module is available to the DLA job using the STEPLIB or LNKLIST.

IZD1004W  modname DB2 CONNECT FAILED FOR ssid
COMMAND command
RC = return_code RS = reason_code BM = bytesMoved BNM = bytesNotMoved R0 = registerZero R15 = registerFifteen
Explanation: The DB2 connect to subsystem ssid failed.
Message Variables:
modname
The name of the module that issued the DB2 connect macro.
ssid
The ID of the DB2 subsystem that the connect macro was issued against.
return_code
The connect return code received.
reason_code
The connect reason code received.
BM
The number of bytes moved.
BNM
The number of bytes not moved.
R0
The value of register zero.
R15
The value of register fifteen.
System action: Processing stops.
System programmer response: Return codes for the DB2 connect macro are documented in DB2 for z/OS Codes.
Messages

Explanation: A command issued using the DB2 IFI command interface failed.

Message Variables:

- **modname**
  - The name of the module that issued the DB2 command.
- **ssid**
  - The ID of the DB2 subsystem that the command was issued against.
- **command**
  - The DB2 command that was issued.
- **return_code**
  - The IFI return code received.
- **reason_code**
  - The IFI reason code received.
- **bytes_moved**
  - The number of data bytes returned by the command.
- **bytes_not_moved**
  - The number of bytes that could not be moved because of buffer overflow.
- **register_zero**
  - The contents of R0 on return from the IFI call.
- **register_fifteen**
  - The contents of R15 on return from the IFI call.

System action: Processing stops.

System programmer response: Return codes for the IFI interface are documented in DB2 for z/OS Codes.

IZD1007W

**modname** DB2 DISCONNECT FAILED FOR **ssid** RC = **return_code** RS = **reason_code**

Explanation: The DB2 disconnect from subsystem **ssid** failed.

Message Variables:

- **modname**
  - The name of the module that issued the DB2 disconnect macro.
- **ssid**
  - The ID of the DB2 subsystem that the disconnect macro was issued against.
- **return_code**
  - The disconnect return code received.
- **reason_code**
  - The disconnect reason code received.

System action: Processing continues.

System programmer response: Return codes for the disconnect macro are documented in DB2 for z/OS Codes.

IZD1010E

**modname** IOSINFO MACRO FAILED RC = **return_code** RS = **reason_code**

Explanation: The IOSINFO macro has failed.

Message Variables:

- **modname**
  - The name of the module that issued the IOSINFO macro.
- **return_code**
  - The IOSINFO return code received.
- **reason_code**
  - The IOSINFO reason code received.

System action: Processing stops.

System programmer response: Return codes for the IOSINFO macro are documented in z/OS MVS Authorized Assembler Services Reference.

IZD1011E

**modname** UCBSCAN MACRO FAILED RC = **return_code**

Explanation: The UCBSCAN macro has failed.

Message Variables:

- **modname**
  - The name of the module that issued the UCBSCAN macro.
- **return_code**
  - The UCBSCAN return code received.

System action: Processing stops.

System programmer response: Return codes for the UCBSCAN macro are documented in z/OS MVS Authorized Assembler Services Reference.
Messages

IZD1012E  modname UCBINFO MACRO FAILED
         RC = return_code
Explaination: The UCBINFO macro has failed.
Message Variables:
modname
   The name of the module that issued the UCBINFO macro.
return_code
   The UCBINFO return code received.
System action: Processing stops.
System programmer response: Return codes for the UCBINFO macro are documented in z/OS MVS Authorized Assembler Services Reference.

IZD1013E  modname IOSCDR MACRO FAILED RC = return_code
Explaination: The IOSCDR macro has failed.
Message Variables:
modname
   The name of the module that issued the IOSCDR macro.
return_code
   The IOSCDR return code received.
System action: Processing stops.
System programmer response: Return codes for the IOSCDR macro are documented in z/OS MVS Authorized Assembler Services Reference.

IZD1020W  modname SQLCODES = sql_codes
Explaination: The IOSCDR macro has failed.
Message Variables:
modname
   The name of the module that issued the SQL call.
sql_codes
   The SQL codes received.
System action: Processing stops.
System programmer response: Return codes for SQL calls area documented in DB2 for z/OS Codes.

IZD1030E  modname STORAGE MACRO FAILED
         FOR type RC = return_code
Explaination: The STORAGE macro has failed.
Message Variables:
modname
   The name of the module that issued the STORAGE macro.
type
   The type of STORAGE request that failed.
return_code
   The STORAGE return code received.
System action: Processing stops.
System programmer response: Return codes for the STORAGE macro are documented in z/OS MVS Authorized Assembler Services Reference.

IZD1031E  modname IX CQUERY MACRO FAILED
         FOR type RC = return_code RS = reason_code
Explaination: The DB2 module DSNALI could not be loaded.
Message Variables:
modname
   The name of the module that issued the IXCQUERY macro.
type
   The IXCQUERY type, which is GROUP or MEMBER.
return_code
   The IXCQUERY return code received.
reason_code
   The IXCQUERY reason code received.
System action: Processing stops.
System programmer response: Return codes for the IXCQUERY macro are documented in z/OS MVS Authorized Assembler Services Reference. Check that the DSNALI module is available to the DLA job using the STEPLIB or LNKLIST.

IZD1040E  modname COULD NOT LOAD
         MODULE loadmod ABEND = return_code
         RS = reason_code
Explaination: The named module could not be loaded.
Message Variables:
modname
   The name of the module that issued the LOAD for DSNALI.
loadmod
   The name of the module that was being loaded.
return_code
   The LOAD return code received.
reason_code
   The LOAD reason code received.
System action: Processing stops.
System programmer response: Return codes for the LOAD macro are documented in z/OS MVS Authorized Assembler Services Reference. Check that the module available to the DLA job using the STEPLIB or LNKLIST.

IZD1041E  modname COULD NOT ATTACH
         MODULE attachmod RC = return_code
Explaination: The named module could not be attached.
Message Variables:
modname
The name of the module that issued the ATTACH macro.

attachmod
The name of the module that was being attached.

return_code
The ATTACH return code received.

System action: Processing stops.
System programmer response: Contact IBM Software Support.

---
IZD1050E modname ENCOUNTERED UNEXPECTED RETURN CODE FORM DELETE RC = return_code
Explanation: A DELETE macro call has returned an unexpected return code.
Message Variables:
modname
  The name of the module that issued the DELETE macro.
return_code
  The DELETE return code received.

System action: Processing continues.
System programmer response: Return codes for the DELETE macro are documented in z/OS MVS Authorized Assembler Services Reference.

---
IZD1051E modname ENCOUNTERED UNEXPECTED RETURN CODE FROM IRXEXCOM RC = return_code
Explanation: The REXX interface routine IRZEXCOM has failed.
Message Variables:
modname
  The name of the module that issued the call to the IRXEXCOM routine.
return_code
  The IRXEXCOM return code received.

System action: Processing continues.
System programmer response: Return codes for the IRXEXCOM routine are documented in z/OS MVS Authorized Assembler Services Reference.

---
IZD1052E modname VARIABLE variable NOT CODED IN LOOKUP TABLE
Explanation: The REXX interface routine IRZEXCOM has failed.
Message Variables:
modname
  The name of the module that issued the call to the IRXEXCOM routine.
variable
  The name of the missing variable.

System action: Processing continues.
System programmer response: Return codes for the READ macro are documented in DF/SMS Macro Services for Datasets.
Messages

IZD1072E  modname MEMBER member_name TOO LARGE
Explanation:  A PARMLIB member could not be processed because it is too large.
Message Variables:
  modname
    The name of the module that attempted to read the PARMLIB member.
  member_name
    The name of the member being read.
System action:  Processing continues.

IZD1073I  modname MEMBER member_name EMPTY
Explanation:  A PARMLIB member contains no data.
Message Variables:
  modname
    The name of the module that attempted to read the PARMLIB member.
  member_name
    The name of the empty member.
System action:  Processing continues.

IZD1074I  modname MEMBER member_name FOUND VOLUME = volume DSN = dataset_name
Explanation:  A PARMLIB member has been found in the PARMLIB concatenation and read successfully.
Message Variables:
  modname
    The name of the module that read the PARMLIB member.
  member_name
    The name of the member.
  volume
    The volume serial number of the volume where the PARMLIB data set resides.
  dataset_name
    The name of the system PARMLIB data set.
System action:  Processing continues.

IZD1075W  modname NO READ ACCESS TO dataset_name ON VOLUME volume
Explanation:  A system PARMLIB data set is read protected.
Message Variables:
  modname
    The name of the module that attempted to read the PARMLIB member.
  dataset_name
    The name of the system PARMLIB data set.
  volume
    The volume serial number of the volume where the PARMLIB data set resides.
System action:  Processing continues but the PARMLIB member data is not discovered.

IZD1080E  modname DIRECTORY OPEN FAILED FOR DD = ddname RC = return_code
Explanation:  An attempt to open a PDS/PDSE directory has failed.
Message Variables:
  modname
    The name of the module that attempted to open the PDS/PDSE directory.
  ddname
    The name of the DD that is allocated to the input PDS/PDSE directory.
  return_code
    The return code received from the OPEN macro.
System programmer response:  Return codes for the OPEN macro are documented in DF/SMS Macro Services for Datasets.

IZD1081E  modname DIRECTORY READ FAILED FOR DD = ddname RC = return_code
Explanation:  An attempt to read a PDS/PDSE directory has failed.
Message Variables:
  modname
    The name of the module that attempted to read the PDS/PDSE directory.
  ddname
    The name of the DD that is allocated to the input PDS/PDSE directory.
  return_code
    The return code received from the READ macro.
System programmer response:  Return codes for the READ macro are documented in DF/SMS Macro Services for Datasets.

IZD1090I  modname CODEPAGE TRANSLATION ROUTINE CUNLCNV RETURNED RC = return_code RS = reason_code
Explanation:  The z/OS system codepage translation routine CUNLCNV has returned the indicated return and reason code.
Message Variables:
  modname
    The name of the module that issued the call to the CUNLCNV routine.
  return_code
    The return code received from the CUNLCNV routine.
  reason_code
    The reason code received from the CUNLCNV routine.
### System action
Processing continues but codepage translation might be in error.

### System programmer response
Check the SET CODEPAGE directive for the correct codepage number.

#### IZD1100E
**module** module LOAD of EZBNMIFR program failed

**Explanation:** The module attempted to load module EZBNMIFR and it failed.

**Message Variables:**
- **module** The module that attempted the load.

#### IZD1101E
**module** module EZBNMIFR service invocation failed **function** type

**ReturnCode return code and ReasonCode reason code**

**Explanation:** The module attempted to use a function that failed with the indicated return code and reason code.

**Message Variables:**
- **module** The module that issued the function.
- **function** The function that was attempted.
- **return code** The return code returned from the failed function.
- **reason code** The reason code returned from the failed function

#### IZD2000I
**modname debug data**

**Explanation:** This message is used for debugging purposes.

**Message Variables:**
- **modname** The name of the module that attempted to read the PDS/PDSE directory.
- **debug data** The string of data to be used for debugging purposes.

### System action
Processing continues.

### System programmer response
This message is issued when the DEBUG options are set on.

#### IZD3000E
**module ALESERV MACRO FAILED**

**RC=return_code TOKEN=token**

**Explanation:** The module issued the ALESERV macro and failed.

**Message Variables:**
- **module** The module that issued the ALESERV macro.
- **return code** The return code returned from the ALESERV macro.
- **token** The token returned from the ALESERV macro.

#### IZD8010I
CICS Discovery **InputModule=InputModule**

**ASCBPtr=ascb_ptr**

**Explanation:** The discovery process displays the input pointer and ASCB pointer.

**Message Variables:**
- **InputModule** The address of the input parameters passed to the discovery routine.
- **ascb_ptr** The address of the ASCB passed to the discovery routine.

#### IZD8011I
CICS Discovery exit. Execution time **seconds seconds**

**Explanation:** This message displays the number of seconds used by the discovery process.

**Message Variables:**
- **seconds** The number of seconds.

#### IZD8012E
Unable to access ASCB at address **address**

**Explanation:** The discovery process was unable to access the ASCB that was passed.

**Message Variables:**
- **address** The ASCB address.

#### IZD8013I
Unable to determine CICS Version

**Explanation:** The discovery process was unable to determine the version of the CICS Region.

#### IZD9000E
**modname WAS NOT PASSED CORRECT PARAMETERS**

**Explanation:** An incorrect call was made to the named module.

**Message Variables:**
- **modname** The name of the module that was called with incorrect parameters.

### System action
Processing stops.

### System programmer response
If the module name is IZDUIRX, then check the JCL parameters. Otherwise, contact IBM Software Support.

#### IZD9001I
**modname CALLING: rexxname**

**Explanation:** The IZDUIRX module has invoked the indicated REXX routine.

**Message Variables:**
- **modname** The name of the IZDUIRX REXX front end routine.
Messages

rexname
The name of the REXX routine.
System action: Processing continues.

IZD9002I  modname PARMS: <parameters>
Explanation: The IZDUIRX module was called with the indicated parameters.
Message Variables:
modname
   The name of the IZDUIRX REXX front end routine.
parameters
   The input parameters.
System action: Processing continues.

IZD9003I  modname DEBUG IS ON
Explanation: The IZDUIRX module was called in debug mode.
Message Variables:
modname
   The name of the IZDUIRX REXX front end routine.
System action: Processing continues.

IZD9004I  modname COMPLETED WITH RETURN CODE return_code
Explanation: The IZDUIRX module was called in debug mode.
Message Variables:
modname
   The name of the IZDUIRX REXX front end routine.
return_code
   The return code received from the IZDUIRX module.
System action: Processing is complete.
System programmer response: Contact IBM Software Support.

Return codes

Return codes from IZDRDLA

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Normal completion</td>
</tr>
<tr>
<td>1</td>
<td>Normal completion - FTP=N was specified. A non-zero return code is set to force a bypass of the FTP step.</td>
</tr>
<tr>
<td>4</td>
<td>z/OS DLA is not APF authorized or is not authorized to access requested data.</td>
</tr>
<tr>
<td>12</td>
<td>No data was returned from 1 or more discovery routines. EXECIO failed. Allocation failed. A call to an external interface routine has failed Bad parameters were passed to an external interface routine.</td>
</tr>
</tbody>
</table>
Return codes from IZDRDLA

16 Fatal REXX error has occurred.

Other return codes

Other return codes from system macros and system service routines may be issued with an associated message. In this case, see the relevant operating system or product manual for an explanation of return and reason codes.
Appendix C. z/OS DLA data model class types represented in CDM

This section describes how the various source application data objects map to classes in the Common Data Model (CDM). For more details about the CDM, see the CDM CD that is included with CCMDB and TADDM.

The following "All books" section is grouped by output book produced by the z/OS DLA (for example, ZOSBASE).

All books
ManagementSoftwareSystem

ManagementSoftwareSystem
The ManagementSoftwareSystem (MSS) class identifies what discovered the resources in the XML book. Resources can be discovered by multiple MSS classes. If an MSS is deleted, all resources discovered by that MSS, per unique MSSName, and which do not have another MSS, are deleted too. The MSS is also important for discovery REFRESH processing.

Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hostname</td>
<td>If IP stack - Bookname@Fqdn&gt; Else Bookname@SNA_HOST.&lt;NetidSSCP&gt;</td>
<td><a href="mailto:ZOSABC@wlag.ibm.com">ZOSABC@wlag.ibm.com</a></td>
</tr>
<tr>
<td>ManufacturerName</td>
<td>IBM</td>
<td>IBM</td>
</tr>
<tr>
<td>ProductName</td>
<td>IBM Tivoli Discovery Library Adapter for z/OS</td>
<td>IBM Tivoli Discovery Library Adapter for z/OS</td>
</tr>
<tr>
<td>ProductVersion</td>
<td>3.1.0</td>
<td>3.1.0</td>
</tr>
<tr>
<td>MSSName</td>
<td>ibm-cdm:///CDMMSS/Hostname=&lt;hostname&gt;+ManufacturerName=IBM+ProductName=IBM z/OS Discovery Library Adapter</td>
<td>ibm-cdm:///CDMMSS/Hostname=<a href="mailto:ZOSABC@wlag.ibm.com">ZOSABC@wlag.ibm.com</a></td>
</tr>
</tbody>
</table>

Notes:
1. MSS is populated within the <idml:source> envelope and does not contain an ID attribute. All other classes listed below are populated within the <cdm:CDM-ER-Specification> envelope and include an ID attribute.
2. Do not populate the Label.

Requirements for discovery
- None. One instance is populated in all z/OS DLA books.

CDM specific information

<table>
<thead>
<tr>
<th>CDM naming rule</th>
<th>process.ManagementSoftwareSystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM attribute</td>
<td>Value</td>
</tr>
<tr>
<td>CDMSchemaVersion</td>
<td>2.92</td>
</tr>
</tbody>
</table>

Naming rules
GlobalMSSName

Relationships
None
The process.Organization class provides a GlobalName that can be used to distinguish between copies of the same z/OS, and to preserve uniqueness in the CCMDB. For example, if the site has a production z/OS system with an SMFID called MVS1 and another z/OS system in a Disaster Recovery environment with the same SMFID called MVS1, to distinguish between the z/OS images, the Organization must be set to different values. The value for GlobalName is provided by the ORGNAME parameter at discovery time or by the defaultOrg value when the IdML book file is loaded into the CCMDB.

The z/OS DLA provides a configuration option in member IZDCDEF that enables you to set the Global Name. The default is <defaultOrg>, which means that the CCMDB-configured default is used.

Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Attribute</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>GlobalName</td>
<td>Customer-assigned name or the literal &lt;defaultOrg&gt;</td>
<td>TESTORG</td>
</tr>
</tbody>
</table>

Requirements for discovery

- None. One instance is populated in all z/OS DLA books.

CDM specific information

<table>
<thead>
<tr>
<th>CDM naming rule</th>
<th>process.Organization</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>id (IdML)</td>
<td>&lt;GlobalName&gt;- Organization</td>
<td>&lt;defaultOrg&gt;-Organization</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;GlobalName&gt;- Organization</td>
<td>&lt;defaultOrg&gt;-Organization</td>
</tr>
</tbody>
</table>

Naming rules

GlobalOrg

Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owns</td>
<td>Organization</td>
<td>sys.zOS.Sysplex</td>
<td></td>
</tr>
<tr>
<td>Owns</td>
<td>Organization</td>
<td>sys.zOS.zOS</td>
<td></td>
</tr>
<tr>
<td>Owns</td>
<td>Organization</td>
<td>sys.zOS.ZVM</td>
<td></td>
</tr>
</tbody>
</table>
IPv4Address

**net.IPAddress**

The net.IPAddress class represents an IP address on z/OS systems. There can be multiple IP addresses and multiple IP stacks but initially the DLA discovers only the primary IP address.

**Attributes**

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>StringNotation</td>
<td>An IP address in string form. The expected form for IPv6 address instances is x:x:x:x:x:x:x.</td>
<td>192.168.123.54 2001:DB8:0:0:800:200C:417A</td>
</tr>
</tbody>
</table>

**Requirements for discovery**

- Filtering enabled (which is the default).

**CDM specific information**

<table>
<thead>
<tr>
<th>CDM naming policy</th>
<th>net.IPAddress</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>&lt; StringNotation &gt;</td>
<td>192.168.123.54</td>
</tr>
<tr>
<td>ID (IdML)</td>
<td>&lt; StringNotation -&gt;IpAddress</td>
<td>192.168.123.54-IPAddress</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt; StringNotation -&gt;IpAddress</td>
<td>192.168.123.54-IPAddress</td>
</tr>
</tbody>
</table>

**Naming rules**

StringNotation

**Relationships**

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>bindsTo</td>
<td>net.IpInterface</td>
<td>net.IPAddress</td>
<td></td>
</tr>
<tr>
<td>assignedTo</td>
<td>net.Fqdn</td>
<td>net.IPAddress</td>
<td></td>
</tr>
</tbody>
</table>
**net.IpInterface**

This class allows the relationship between an IP interface and a z/OS system to be represented.

**Attributes**

None

**Requirements for discovery**

- Filtering enabled (which is the default).

**CDM specific information**

<table>
<thead>
<tr>
<th>CDM naming policy</th>
<th>net.IpInterface</th>
<th>ID (IdML)</th>
<th>&lt;StringNotation&gt;-IpInterface 192.168.123.54-IpInterface</th>
</tr>
</thead>
<tbody>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;StringNotation&gt;-IpInterface</td>
<td>sourceToken (IdML)</td>
<td>192.168.123.54-IpInterface</td>
</tr>
</tbody>
</table>

**Naming rules**

AddressOnCS

**Relationships**

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>contains</td>
<td>sys.zOS.zOS</td>
<td>net.IpInterface</td>
<td></td>
</tr>
<tr>
<td>bindsTo</td>
<td>net.IpInterface</td>
<td>IPAddress</td>
<td></td>
</tr>
<tr>
<td>bindsTo</td>
<td>net.TcpPort</td>
<td>net.IpInterface</td>
<td></td>
</tr>
<tr>
<td>bindsTo</td>
<td>net.UdpPort</td>
<td>net.IpInterface</td>
<td></td>
</tr>
<tr>
<td>contains</td>
<td>sys.ZSeriesComputerSystem</td>
<td>net.IpInterface</td>
<td></td>
</tr>
<tr>
<td>contains</td>
<td>sys.LPAR</td>
<td>net.IpInterface</td>
<td></td>
</tr>
</tbody>
</table>
net.Fqdn

The fully qualified IP name of the z/OS system.

Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fqdn</td>
<td>The fully qualified IP host name.</td>
<td>wlag.ibm.com</td>
</tr>
</tbody>
</table>

Requirements for discovery

- Filtering enabled (which is the default).

CDM specific information

<table>
<thead>
<tr>
<th>CDM naming policy</th>
<th>net.Fqdn</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM attribute</td>
<td>Value</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;Fqdn&gt;-Fqdn</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;Fqdn&gt;-Fqdn</td>
</tr>
</tbody>
</table>

Naming rules

GlobalHostName

Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>assignedTo</td>
<td>net.Fqdn</td>
<td>IPAddress</td>
<td></td>
</tr>
</tbody>
</table>
sys.ZOS.Sysplex

A sysplex is a set of z/OS systems that communicate and cooperate with each other through specific multisystem hardware components and software services to process customer workloads.

Not all members of the sysplex need to be operating all the time. Systems can join, leave, and rejoin the sysplex at any time. A z/OS system cannot concurrently join multiple sysplexes, but it can leave one sysplex and join another. It is also possible for a sysplex to only ever have the one z/OS system as a member. This situation is referred to as a monoplex system, where data can still be shared between subsystems that execute on the same system.

It is also possible for a z/OS system to run stand-alone in XCF-local mode (non-sysplex), which differs from monoplex in that data sharing between subsystems does not use a coupling facility.

Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name commonly used by support personnel to identify the group of z/OS images that participate in the sysplex.</td>
<td>SYSPLEXO</td>
</tr>
</tbody>
</table>

Requirements for discovery

- None. One instance is populated in all z/OS DLA books.

CDM specific information

<table>
<thead>
<tr>
<th>CDM naming policy</th>
<th>sys.ZOS.Sysplex</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>If Organization GlobalName = &quot;&lt;defaultOrg&gt;&quot; Then Label = &lt;Name&gt; Else Label = &lt;Name&gt;-&lt;GlobalName&gt;</td>
<td>SYSPLEXO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYSPLEXO-TESTORG</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;Name&gt;-Sysplex</td>
<td>SYSPLEXO-SYSPLEX</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;Name&gt;-Sysplex</td>
<td>SYSPLEXO-SYSPLEX</td>
</tr>
<tr>
<td>memberOf</td>
<td>sys.ZOS.zOS</td>
<td>sys.ZOS.Sysplex</td>
</tr>
<tr>
<td>supports</td>
<td>sys.ZOS.zCouplingFacility</td>
<td>sys.ZOS.Sysplex</td>
</tr>
</tbody>
</table>

Naming rules

QualifiedSysplex

Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owns</td>
<td>process.Organization</td>
<td>sys.ZOS.Sysplex</td>
<td></td>
</tr>
<tr>
<td>hostedCollection</td>
<td>sys.ZOS.Sysplex</td>
<td>core.SystemSpecificCollection</td>
<td></td>
</tr>
<tr>
<td>Contains</td>
<td>sys.ZOS.Sysplex</td>
<td>sys.ZOS.IMSSysplexGroup</td>
<td></td>
</tr>
</tbody>
</table>
### sys.ZOS.Sysplex

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains</td>
<td>sys.ZOS.Sysplex</td>
<td>sys.ZOS.DB2DataSharingGroup</td>
<td></td>
</tr>
</tbody>
</table>
So that Author products can use the discovery library information, you must comply with the CIM. This class is required in order to be standards-compliant.

### Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>SystemSpecificName</td>
<td>Sysplex Node Set</td>
<td>Sysplex Node Set</td>
</tr>
</tbody>
</table>

### Requirements for discovery

- None. One instance is populated in all z/OS DLA books.

### CDM specific information

<table>
<thead>
<tr>
<th>CDM naming policy</th>
<th>core.SystemSpecificCollection</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>id (IdML)</td>
<td>&lt;Name (Sysplex*)&gt;</td>
<td>SYSPLEXO-SystemSpecificCollection</td>
<td></td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;Name (Sysplex*)&gt;</td>
<td>SYSPLEXO-SystemSpecificCollection</td>
<td></td>
</tr>
</tbody>
</table>

### Naming rules

SSCollectionIdentity

### Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>hostedCollection</td>
<td>sys.zOS.Sysplex</td>
<td>core.SystemSpecificCollection</td>
<td></td>
</tr>
<tr>
<td>memberOf</td>
<td>sys.zOS.ZOS</td>
<td>core.SystemSpecificCollection</td>
<td></td>
</tr>
</tbody>
</table>
core.SystemSpecificCollection

ZOSBASE
sys.zOS.ZSeriesComputerSystem

**sys.zOS.ZSeriesComputerSystem** <virtual=false>

The `zSeriesComputerSystem` (<virtual=false>) represents the instance of physical hardware, or the box (CPC) that a `z/OS` image executes on. A `zSeriesComputerSystem` can host multiple LPARs with different instances of and `OperatingSystems` assigned to each LPAR.

**Attributes**

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>MemorySize</td>
<td>Total Central Storage allocated to all LPARs</td>
<td>32235643</td>
</tr>
<tr>
<td>ExpandedMemorySize</td>
<td></td>
<td>10.5</td>
</tr>
<tr>
<td>Manufacturer</td>
<td>Manufacturer ID</td>
<td>IBM</td>
</tr>
<tr>
<td>Model</td>
<td>Machine type 2084</td>
<td></td>
</tr>
<tr>
<td>ModelID</td>
<td>Model ID 327</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Customer-assigned hardware name of the CEC.</td>
<td>PTHES5</td>
</tr>
<tr>
<td>NumCPUs</td>
<td>Total number of CPUs</td>
<td>2</td>
</tr>
<tr>
<td>ProcessCapacityUnits</td>
<td>Millions of service units per hour</td>
<td>MSU</td>
</tr>
<tr>
<td>ProcessingCapacity</td>
<td>An indication of the relative processing capacity of the machine (CEC). Useful in the context of licensing, accounting, charge-back, and planning.</td>
<td>108</td>
</tr>
<tr>
<td>ProcessWeight</td>
<td>The total weight of all CPs, minus dedicated CPs</td>
<td>107</td>
</tr>
<tr>
<td>SerialNumber</td>
<td>Serial number</td>
<td>00000000000123D7</td>
</tr>
<tr>
<td>WLMManaged</td>
<td>Indicates at least 1 CPU is WLM Managed</td>
<td>true</td>
</tr>
<tr>
<td>CPUSpeed</td>
<td>Processor Speed in Mips</td>
<td>886810000</td>
</tr>
</tbody>
</table>

**Requirements for discovery**

- Filtering enabled (which is the default).

**CDM specific information**

<table>
<thead>
<tr>
<th>CDM Naming rule</th>
<th>sys.zOS.ZSeriesComputerSystem</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM attribute</td>
<td>Value</td>
<td>Example</td>
</tr>
<tr>
<td>Type</td>
<td>ComputerSystem</td>
<td>ComputerSystem</td>
</tr>
<tr>
<td>Virtual</td>
<td>This is not a virtual computer system.</td>
<td>false</td>
</tr>
<tr>
<td>label</td>
<td>&lt;Name&gt;</td>
<td>PTHES5</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;Name&gt;-zSeriesComputerSystem</td>
<td>PTHES5-zSeriesComputerSystem</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;Name&gt;-zSeriesComputerSystem</td>
<td>PTHES5-zSeriesComputerSystem</td>
</tr>
</tbody>
</table>

**Naming rules**

CSProduct
### sys.zOS.ZSeriesComputerSystem

**Relationships**

<table>
<thead>
<tr>
<th>Link Type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>installedOn</td>
<td>sys.zOS.ZOS</td>
<td>sys.zOS.ZSeriesComputerSystem</td>
<td>Populated only if z/OS is running in stand-alone mode, that is no LPAR.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>installedOn</td>
<td>sys.zOS.ZVM</td>
<td>sys.zOS.ZSeriesComputerSystem</td>
<td>Populated only if z/OS is running under a ZVM that is running in stand-alone mode, that is, no LPAR.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>runsOn</td>
<td>sys.zOS.ZOS</td>
<td>sys.zOS.ZSeriesComputerSystem</td>
<td>Populated only if z/OS is running in stand-alone mode, that is, no LPAR.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>runsOn</td>
<td>sys.zOS.ZVM</td>
<td>sys.zOS.ZSeriesComputerSystem</td>
<td>Populated only if z/OS is running under a ZVM that is running in stand-alone mode, that is, no LPAR.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>virtualizes</td>
<td>sys.zOS.LPAR</td>
<td>sys.zOS.ZSeriesComputerSystem (false)</td>
<td>Populated only if z/OS is directly under an LPAR, or z/OS is under VM under an LPAR.</td>
</tr>
<tr>
<td></td>
<td>sys.zOS.ZSeries \ ComputerSystem (true)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>virtualizes</td>
<td>sys.zOS.ZVMGuest</td>
<td>sys.sOS.ZSeriesComputerSystem</td>
<td>Populated only if z/OS is running under a ZVM that is running in stand-alone mode, that is, no LPAR.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**sys.zOS.LPAR**

An LPAR is a logical partition of the physical hardware where an operating system instance can run, for example, z/OS, z/VM®, and Linux®. The computing platform can operate in either basic mode (non-partitioned) or lpar mode. The LPAR class instance is discovered only if the hardware is operating in LPAR mode. Only an LPAR that runs z/OS or z/OS hosted by z/VM are discovered.

**Note:** The z/OS DLA supports the CDM V2.9.2 change to use CDM class ZseriesComputerSystem to represent an LPAR. All LPARs are provided as instances of a ZseriesComputerSystem object, where Virtual = true. For migration, every LPAR is also provided as an object of a CDM-class LPAR. When TADDM loads z/OS DLA books, it merges matching LPAR and ZseriesComputerSystem (where virtual = true) objects.

**Attributes**

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Logical Partition name.</td>
<td>ES5C</td>
</tr>
<tr>
<td>VMID</td>
<td>Logical Partition ID. - Naming rule virtualizes ID. Note: Prior versions of z/OS DLA provided attribute LPARID; the LPAR ID is now shown in the VMID to match a data model change in CDM V2.9.2.</td>
<td>12</td>
</tr>
<tr>
<td>MemorySize</td>
<td>30932992</td>
<td></td>
</tr>
<tr>
<td>ExpandedMemorySize</td>
<td>30208</td>
<td></td>
</tr>
<tr>
<td>MIFId</td>
<td>The Multiple Image Facility image ID assigned to this partition (also known as partition number). If you are providing the MIFId, provide the LPAR name, too, which helps when object instances are reconciled.</td>
<td>15</td>
</tr>
<tr>
<td>ProcessingCapacity</td>
<td>An indication of the relative processing capacity of the LPAR. Useful in the context of licensing, accounting, charge-back, and planning.</td>
<td>81</td>
</tr>
<tr>
<td>ProcessCapacityUnits</td>
<td>Millions of service units per hour</td>
<td>MSU</td>
</tr>
<tr>
<td>LPARName</td>
<td>Logical partition name. Example = ES5C.</td>
<td></td>
</tr>
</tbody>
</table>

**Requirements for discovery**

- Filtering enabled (which is the default). This class is also sensitive to the ZSeriesComputerSystem filter.
- Discovered only if applicable, which is when the z/OS is running in an LPAR.

**CDM specific information**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.LPAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM attribute</td>
<td>Value</td>
</tr>
<tr>
<td>Type</td>
<td>ComputerSystem</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;Name&gt;_&lt;HardwareName*&gt;</td>
</tr>
<tr>
<td>* Name from 1.1.11.</td>
<td></td>
</tr>
</tbody>
</table>
## sys.zOS.LPAR

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>id (IdML)</td>
<td>&lt;Name&gt;-&lt;HardwareName*&gt;-LPAR</td>
<td>ESSC-PTHES5-LPAR</td>
</tr>
<tr>
<td></td>
<td>* Name from 1.1.11.</td>
<td></td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;Name&gt;-&lt;HardwareName*&gt;-LPAR</td>
<td>ESSC-PTHES5-LPAR</td>
</tr>
<tr>
<td></td>
<td>* Name from 1.1.11.</td>
<td></td>
</tr>
</tbody>
</table>

### Naming rules

- **LPARName**
- **VMIDInHost**

### Relationships

<table>
<thead>
<tr>
<th>Link Type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>virtualizes</td>
<td>sys.zOS.LPAR</td>
<td>sys.zOS.ZSeriesComputerSystem</td>
<td>Populated only if z/OS is directly under an LPAR, or z/OS is under VM under an LPAR.</td>
</tr>
<tr>
<td>runsOn</td>
<td>sys.zOS.ZOS</td>
<td>sys.zOS.LPAR</td>
<td>Populated only if z/OS is directly under an LPAR (no VM and not in stand-alone mode).</td>
</tr>
<tr>
<td>installedOn</td>
<td>z/OS</td>
<td>LPAR</td>
<td>Populated only if z/OS is directly under an LPAR (no VM and not in stand-alone mode).</td>
</tr>
<tr>
<td>runsOn</td>
<td>sys.zOS.ZVM</td>
<td>sys.zOS.LPAR</td>
<td>Populated only if z/OS is under VM.</td>
</tr>
<tr>
<td>installedOn</td>
<td>sys.zOS.ZVM</td>
<td>sys.zOS.LPAR</td>
<td>Populated only if z/OS is under VM.</td>
</tr>
<tr>
<td>virtualizes</td>
<td>sys.zOS.ZVMGuest</td>
<td>sys.zOS.LPAR</td>
<td>Populated only if z/OS is under VM.</td>
</tr>
</tbody>
</table>
sys.zOS.ZVM

Discovered if the z/OS runs as a guest of a z/VM host.

Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSName</td>
<td>ZVM name</td>
<td>PTHOMO1</td>
</tr>
</tbody>
</table>

Requirements for discovery

- Filtering enabled (which is the default). This class is also sensitive to the sys.zOS.ZSeriesComputerSystem and sys.zOS.LPAR filters.
- Discovered only if applicable, which is when z/OS is running in a ZVMGuest.
- APF mode.

CDM specific information

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zVM.ZVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>Label</td>
<td>ZVM name</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;Name&gt;-ZVM</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;Name&gt;-ZVM</td>
</tr>
</tbody>
</table>

Naming rules

OSName and owns relationship from process.Organization.

Relationships

<table>
<thead>
<tr>
<th>Link Type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>owns</td>
<td>process.Organization</td>
<td>sys.zOS.ZVM</td>
<td>Naming rule for ZVM.</td>
</tr>
<tr>
<td>runsOn</td>
<td>sys.zOS.ZVMGuest</td>
<td>sys.zOS.ZVM</td>
<td>Naming rule for sys.zOS.ZVMGuest.</td>
</tr>
<tr>
<td>runsOn</td>
<td>sys.zOS.ZVM</td>
<td>sys.zOS.LPAR</td>
<td>Only populated if ZVM is running in an LPAR.</td>
</tr>
<tr>
<td>installedOn</td>
<td>sys.zOS.ZVM</td>
<td>sys.zOS.LPAR</td>
<td>Populated only if ZVM is running in an LPAR.</td>
</tr>
<tr>
<td>runsOn</td>
<td>sys.zOS.ZVM</td>
<td>sys.zOS.ZSeriesComputerSystem</td>
<td>Populated only if ZVM is running in stand-alone mode</td>
</tr>
<tr>
<td>installedOn</td>
<td>sys.zOS.ZVM</td>
<td>sys.zOS.ZSeriesComputerSystem</td>
<td>Populated only if ZVM is running in stand-alone mode.</td>
</tr>
<tr>
<td>owns</td>
<td>process.Organization</td>
<td>sys.zOS.ZVM</td>
<td>Naming rule for ZVM.</td>
</tr>
<tr>
<td>runsOn</td>
<td>sys.zOS.ZVMGuest</td>
<td>sys.zOS.ZVM</td>
<td>Naming rule for sys.zOS.ZCMGuest.</td>
</tr>
<tr>
<td>runsOn</td>
<td>sys.zOS.ZVM</td>
<td>sys.zOS.LPAR</td>
<td>Populated only if ZVM is running in an LPAR.</td>
</tr>
</tbody>
</table>

Appendix C. z/OS DLA data model class types represented in CDM 95
### sys.zOS.ZVM

<table>
<thead>
<tr>
<th>Link Type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>installedOn</td>
<td>sys.zOS.ZVM</td>
<td>sys.zOS.LPAR</td>
<td>Populated only if ZVM is running in an LPAR.</td>
</tr>
<tr>
<td>runsOn</td>
<td>sys.zOS.ZVM</td>
<td>sys.zOS.ZSeriesComputerSystem</td>
<td>Populated only if ZVM is running in stand-alone mode.</td>
</tr>
<tr>
<td>installedOn</td>
<td>sys.zOS.ZVM</td>
<td>sys.zOS.ZSeriesComputerSystem</td>
<td>Populated only if ZVM is running in stand-alone mode.</td>
</tr>
</tbody>
</table>
**sys.zOS.ZVMGuest**

Discovered if the z/OS runs as a guest of a z/VM host.

**Attributes**

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>VM Guest name as known by the z/VM host.</td>
<td>PTHOMO1</td>
</tr>
<tr>
<td>VMID</td>
<td>&lt;Guest name&gt;-&lt;VM name&gt;</td>
<td>PTHOMO1-PTHVM8</td>
</tr>
<tr>
<td>ProcessingCapacity</td>
<td>An indication of the relative processing capacity of the VM Guest. Useful in the context of licensing, accounting, charge-back, and planning.</td>
<td>27</td>
</tr>
<tr>
<td>ProcessCapacityUnits</td>
<td>Millions of service units per hour</td>
<td>MSU</td>
</tr>
</tbody>
</table>

**Requirements for discovery**

- Filtering enabled (which is the default). This class is also sensitive to the `sys.zOS.ZSeriesComputerSystem` and LPAR filters.
- Discovered only if applicable, which is when z/OS is running in a ZVM Guest.
- APF mode is required to discover the parent `sys.zOS.ZVM` class, which is required for the `sys.zOS.ZVMGuest` naming rule.

**CDM specific information**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zVM.ZVMGuest</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM attribute</td>
<td>Value</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;Name&gt;-&lt;VM&gt;</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;Name&gt;-ZVMGuest</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;Name&gt;-ZVMGuest</td>
</tr>
</tbody>
</table>

**Naming rules**

VMID and virtualizes relationship to `sys.zOS.ZSeriesComputerSystem` or `sys.zOS.LPAR`.

**Relationships**

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>virtualizes</td>
<td>sys.zOS.ZVMGuest</td>
<td>sys.zOS.ZSeriesComputerSystem</td>
<td>Populated only if z/OS is running under a ZVM that is running in stand-alone mode, that is, no LPAR.</td>
</tr>
<tr>
<td>virtualizes</td>
<td>sys.zOS.ZVMGuest</td>
<td>sys.zOS.LPAR</td>
<td>Populated only if z/OS is under VM.</td>
</tr>
<tr>
<td>runsOn</td>
<td>sys.zOS.ZVMGuest</td>
<td>sys.zOS.ZVM</td>
<td></td>
</tr>
<tr>
<td>runsOn</td>
<td>sys.zOS.ZOS</td>
<td>sys.zOS.ZVMGuest</td>
<td></td>
</tr>
<tr>
<td>installedOn</td>
<td>sys.zOS.ZOS</td>
<td>sys.zOS.ZVMGuest</td>
<td></td>
</tr>
</tbody>
</table>
The sys.zOS.zOS class type represents the z/OS operating system image, for example, the z/OS system. One LPAR can support one instance of zOS.

### Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSName</td>
<td>The SYSNAME value commonly used by support personnel to identify the z/OS instance. OSName must be unique within a sysplex.</td>
<td>OMO1</td>
</tr>
<tr>
<td>SMFID</td>
<td>The SMFID value commonly used by support personnel to identify the z/OS instance. Note that Sysname and SMFID are often the same. Tivoli OMEGAMON® product uses the SMFID in its z/OS related Managed System Names.</td>
<td>OMO1</td>
</tr>
<tr>
<td>NetID</td>
<td>SNA Network-ID</td>
<td>AUIBMQXP</td>
</tr>
<tr>
<td>SSCP</td>
<td>VTAM® SSCP Name</td>
<td>SA53CDRM</td>
</tr>
<tr>
<td>NetidSSCP</td>
<td>&lt;netid&gt;.&lt;sscp&gt;</td>
<td>AUIBMQXP.SA53CDRM</td>
</tr>
<tr>
<td>Name</td>
<td>The primary IP name. It is possible that a z/OS does not participate in an IP network and therefore has no IP address.</td>
<td>pthomo1.au.ibm.com</td>
</tr>
<tr>
<td>PrimaryJES</td>
<td>The job or subsystem name of the primary Job Entry Subsystem, used by support personnel.</td>
<td>JES2</td>
</tr>
<tr>
<td>JESNode</td>
<td>The node name of the primary Job Entry Subsystem, used by support personnel particularly in relation to communications between JES subsystems on different z/OS images.</td>
<td>PTHOMO1</td>
</tr>
<tr>
<td>VersionString</td>
<td>The Version, Release, and Modification level of the z/OS system.</td>
<td>1.4</td>
</tr>
<tr>
<td>SysResVolume</td>
<td>The name of the disk volume containing the system resident nucleus that z/OS was IPL’ed with, which is important for support personnel to determine maintenance levels.</td>
<td>$$SR4S</td>
</tr>
<tr>
<td>ProcessingCapacity</td>
<td>An indication of the relative processing capacity of the z/OS. Useful in the context of licensing, accounting, charge-back, and planning.</td>
<td>27</td>
</tr>
<tr>
<td>ProcessCapacityUnits</td>
<td>Millions of service units per hour</td>
<td>MSU</td>
</tr>
<tr>
<td>IPLParmDataset</td>
<td>The data set that contains the IPL parameters.</td>
<td>SYS8.IPLPARM</td>
</tr>
<tr>
<td>IPLParmMember</td>
<td>The member that contains the IPL parameters.</td>
<td>LOAD01</td>
</tr>
<tr>
<td>IPLParmDevice</td>
<td>IPL device number</td>
<td>0500</td>
</tr>
<tr>
<td>IPLParmVolume</td>
<td>IPL volume</td>
<td>$$SR4S</td>
</tr>
<tr>
<td>IPLTime</td>
<td>The date and time that z/OS was IPL’d.</td>
<td>2006-03-24T04:32:00+08:00</td>
</tr>
<tr>
<td>FQDN</td>
<td>Fully qualified host name if the z/OS has an IP.</td>
<td>pthomo1.au.ibm.com</td>
</tr>
</tbody>
</table>
### Requirements for discovery

- Applies to all members, but only ZOSBASE has all attributes. ZOSTASK and subsystem members contain the naming rule-related attributes only.
- None. One instance is populated in all z/OS DLA books.
- Attributes and relationships marked with an asterisk (*) are discovered only when the DLA runs in APF-authorized mode.
- The Parmlib* attributes require the user ID running the DLA to have read access to the PARMLIB concatenation of data sets. The DLA automatically determines the applicable PARMLIB data set names.

### CDM specific information

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.ZOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM attribute</td>
<td>Value</td>
</tr>
</tbody>
</table>
| Label             | If Organization GlobalName = "<defaultOrg>"
                   | The Label = <SMFID>-<SysplexName>
                   | Else Label = <SMFID>-<SysplexName>-<GlobalName> | OMO1-SYSPLEXO
                   |                   | OMO1-SYSPLEXO-TESTOR |
| id (IdML)         | <SMFID>-ZOS | OMO1-ZOS               |
| sourceToken (IdML)| <SMFID>-ZOS | OMO1-ZOS               |

### Naming rules

The z/OS DLA populates several Naming rules:

- NetidSSCP
- OwnerSMFID
- OSName
- OSProduct

### Relationships

<table>
<thead>
<tr>
<th>Link Type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>owns</td>
<td>process.Organization</td>
<td>sys.zOS.ZOS</td>
<td></td>
</tr>
<tr>
<td>memberOf</td>
<td>sys.zOS.ZOS</td>
<td>core.SystemSpecificCollection</td>
<td></td>
</tr>
<tr>
<td>runsOn</td>
<td>sys.zOS.ZOS</td>
<td>sys.zOS.ZSeriesComputerSystem</td>
<td>If the z/OS system is running in stand-alone mode.</td>
</tr>
<tr>
<td>Link Type</td>
<td>Source</td>
<td>Target</td>
<td>Comment</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------</td>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>installedOn</td>
<td>sys.zOS.ZOS</td>
<td>sys.zOS.ZSeriesComputerSystem</td>
<td>If the z/OS system is running in stand-alone mode.</td>
</tr>
<tr>
<td>runsOn</td>
<td>sys.zOS.ZOS</td>
<td>sys.zOS.LPAR</td>
<td>If zOS is in running direct under LPAR.</td>
</tr>
<tr>
<td>installedOn</td>
<td>sys.zOS.ZOS</td>
<td>sys.zOS.LPAR</td>
<td>If zOS is in running direct under LPAR.</td>
</tr>
<tr>
<td>runsOn</td>
<td>sys.zOS.ZOS</td>
<td>sys.zOS.ZVMGuest</td>
<td>If zOS is in running under ZVMGuest.</td>
</tr>
<tr>
<td>installedOn</td>
<td>sys.zOS.ZOS</td>
<td>sys.zOS.ZVMGuest</td>
<td>If zOS is in running under ZVMGuest.</td>
</tr>
<tr>
<td>runsOn</td>
<td>sys.zOS.ZOS</td>
<td>sys.ComputerSystem</td>
<td>CS used for ipInterface.</td>
</tr>
<tr>
<td>runsOn</td>
<td>sys.zOS.AddressSpace</td>
<td>sys.zOS.ZOS</td>
<td></td>
</tr>
<tr>
<td>runsOn</td>
<td>sys.zOS.CICSRegion</td>
<td>sys.zOS.ZOS</td>
<td></td>
</tr>
<tr>
<td>hostedDependency</td>
<td>sys.zOS.IMSSubsystem</td>
<td>sys.zOS.ZOS</td>
<td></td>
</tr>
<tr>
<td>hostedDependency</td>
<td>sys.zOS.DB2Subsystem</td>
<td>sys.zOS.ZOS</td>
<td></td>
</tr>
<tr>
<td>hostedDependency</td>
<td>sys.zOS.MQSubsystem</td>
<td>sys.zOS.ZOS</td>
<td></td>
</tr>
<tr>
<td>accesses</td>
<td>sys.zOS.ZOS</td>
<td>dev.StorageVolume</td>
<td></td>
</tr>
</tbody>
</table>
The `sys.ZReportFile` class type represents a collection of data about a zSeries operating system instance, or a zSeries application. It is typically used to show current configuration parameters. Contents may consist of PARMLIBS, a file (a member on z/OS) or the results of a display command. How this data is stored is product-specific, although generally, this is done using a Character Large Object (CLOB).

### Predefined ZReportFiles for ZOS

<table>
<thead>
<tr>
<th>Parmlibs</th>
<th>A list of the parmlib data set used during IPL. The list is in a table format with two columns: DatasetName and Volume.</th>
<th>SYS1.PARMLIB SYSRES</th>
</tr>
</thead>
<tbody>
<tr>
<td>ParmlibMembers</td>
<td>A list of the parmlib members used during IPL. The list is in a table format with three columns: MemberName, Dataset, and Volume.</td>
<td>ALLOC01 SYS1.PARMLOIB SYSRES</td>
</tr>
<tr>
<td>ParmlibALLOC</td>
<td>A copy of the ALLOCxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies system allocation defaults.</td>
<td></td>
</tr>
<tr>
<td>ParmlibBPXPRM</td>
<td>A copy of BPXPRMxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the default z/OS UNIX® System Services and file system parameters.</td>
<td></td>
</tr>
<tr>
<td>ParmlibCEEPRM</td>
<td>A copy of CEEPRMxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the default Language Environment® (LE) runtime environment file system parameters.</td>
<td></td>
</tr>
<tr>
<td>ParmlibCLOCK</td>
<td>A copy of CLOCKxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the default date and time system parameters.</td>
<td></td>
</tr>
<tr>
<td>ParmlibCOMMD</td>
<td>A copy of COMMANDxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies a list of commands that are executed during IPL.</td>
<td></td>
</tr>
<tr>
<td>ParmlibCONSOL</td>
<td>A copy of CONSOLxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the system console configuration.</td>
<td></td>
</tr>
<tr>
<td>ParmlibCOUPLE</td>
<td>A copy of COUPLExx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the cross system coupling facility default parameters.</td>
<td></td>
</tr>
</tbody>
</table>
### Predefined ZReportFiles for ZOS

<table>
<thead>
<tr>
<th>ParmlibCSVRTL</th>
<th>A copy of CSVRTLxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the RLTS configuration.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ParmlibCUNUNI</td>
<td>A copy of CUNUNIxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the Unicode conversion environment.</td>
</tr>
<tr>
<td>ParmlibDEVSUP</td>
<td>A copy of DEVSUPxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the system device support options.</td>
</tr>
<tr>
<td>ParmlibDIAG</td>
<td>A copy of DIAGxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the common storage trace and GFS trace parameters.</td>
</tr>
<tr>
<td>ParmlibEXIT</td>
<td>A copy of EXITxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the default system exit points and their associated load modules.</td>
</tr>
<tr>
<td>ParmlibGRSCNF</td>
<td>A copy of GRSCNFxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the Global Resource Serialization (GRS) configuration.</td>
</tr>
<tr>
<td>ParmlibGRSRNL</td>
<td>A copy of GRSRNLxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the Global Resource Serialization (GRS) resource list.</td>
</tr>
<tr>
<td>ParmlibIEAAPF</td>
<td>A copy of IEAAPFx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the default static Authorized Program Facility (APF) list.</td>
</tr>
<tr>
<td>ParmlibIEAFIX</td>
<td>A copy of IEAFIXxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the fixed Link Pack Area (FLPA) list.</td>
</tr>
<tr>
<td>ParmlibIEALPA</td>
<td>A copy of IEALPAXxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the static Modified Link Pack Area (MLPA) list.</td>
</tr>
<tr>
<td>Predefined ZReportFiles for ZOS</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>ParmlibIEAOPT</td>
<td>A copy of IEAOPTxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies system performance criteria for customers not use WLM for workload management.</td>
</tr>
<tr>
<td>ParmlibIEAPAK</td>
<td>A copy of IEAPAKxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the LPA pack list.</td>
</tr>
<tr>
<td>ParmlibIEASVC</td>
<td>A copy of IEASVCxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies installation-defined SVCs.</td>
</tr>
<tr>
<td>ParmlibIEASYM</td>
<td>A copy of IEAPRMxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the system symbol definitions.</td>
</tr>
<tr>
<td>ParmlibIEASYS</td>
<td>A copy of IEASYSxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the system parameter list.</td>
</tr>
<tr>
<td>ParmlibIECIOS</td>
<td>A copy of IECIOSxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the MIH, HOTIO, IOTIMING, IOS CTRACE, TERMINAL, FICON®, STORAGE, CAPTUCB, HYPERPAV, and MIDAW parameters.</td>
</tr>
<tr>
<td>ParmlibIEFSSN</td>
<td>A copy of IEFSSNxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the subsystem definitions.</td>
</tr>
<tr>
<td>ParmlibIFAPRD</td>
<td>A copy of IFAPRDxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the Product Enablement Policy.</td>
</tr>
<tr>
<td>ParmlibIGDSMS</td>
<td>A copy of IGDSMSxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the Storage Management Subsystem (SMS) definitions.</td>
</tr>
<tr>
<td>ParmlibIKJTSO</td>
<td>A copy of IKJTSOxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the TSO command and program definitions.</td>
</tr>
<tr>
<td>ParmlibLNKLST</td>
<td>A copy of LNKLSTxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the static link list.</td>
</tr>
</tbody>
</table>
### Predefined ZReportFiles for ZOS

<table>
<thead>
<tr>
<th>ZReportFile</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ParmlibLOAD</strong></td>
<td>A copy of LOADxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the system configuration information.</td>
</tr>
<tr>
<td><strong>ParmlibLPALST</strong></td>
<td>A copy of LPALSTxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the static LPA list.</td>
</tr>
<tr>
<td><strong>ParmlibMSTJCL</strong></td>
<td>A copy of MSTJCLxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the Master Scheduler JCL.</td>
</tr>
<tr>
<td><strong>ParmlibPROG</strong></td>
<td>A copy of PROGxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the dynamic Authorized Program Facility (APF) list, dynamic system exits, dynamic link list (LNKLST), and dynamic LPA.</td>
</tr>
<tr>
<td><strong>ParmlibSCHED</strong></td>
<td>A copy of SCHEDxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the PPT, master trace table, and abend codes for automatic restart.</td>
</tr>
<tr>
<td><strong>ParmlibSMFPRM</strong></td>
<td>A copy of SMFPRMxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the System Management Facilities (SMF) parameters.</td>
</tr>
<tr>
<td><strong>ParmlibVATLST</strong></td>
<td>A copy of VATLSTxx members from the PARMLIB data set concatenation that was used for the current IPL. This member specifies the Volume Attribute (VAT) list.</td>
</tr>
<tr>
<td><strong>DynamicLNKLSTName</strong></td>
<td>Name of the currently active dynamic lnklst.</td>
</tr>
<tr>
<td><strong>DynamicLNKLST</strong></td>
<td>A copy of the currently active Dynamic LNKLST. The LNKLST can be modified at any time after IPL by an authorized operator command. The list is in a table format with two columns: DatasetName and Volume.</td>
</tr>
<tr>
<td><strong>DynamicAPF</strong></td>
<td>A copy of the currently active Authorized Program Facility list. The APF list can be modified at any time after IPL by an authorized operator command. The list is in a table format with two columns: Volume and DatasetName.</td>
</tr>
</tbody>
</table>
**Predefined ZReportFiles for ZOS**

| DynamicEXIT | A copy of the currently active system exits. System exits be modified at any time after IPL by an authorized operator command. The list is in a table format with two columns: ExitName and ModuleName. | SUBSTC.IEFUJI OPCUJI SYS.IEFUSI IEEUSI CAZ00990 |

**Requirements for discovery**

- None. One instance of each ZOS report is populated in the ZOSBASE book. Other books contain additional ZReportFile objects (for example, DB Subsystem books contain DB2 ZReportFiles).
- The Parmlib* attributes require the user ID running the DLA to have read access to the PARMLIB concatenation of data sets. The DLA automatically determines the applicable PARMLIB data set names.

**CDM specific information**

<table>
<thead>
<tr>
<th>CDM naming policy</th>
<th>ZReportFile</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM attribute</td>
<td>Value</td>
</tr>
<tr>
<td>Checksum</td>
<td>7742261</td>
</tr>
<tr>
<td>Content</td>
<td>373 bytes</td>
</tr>
<tr>
<td>Data set</td>
<td>Volume</td>
</tr>
<tr>
<td>SYS2.IPLPARM</td>
<td>G1B01B</td>
</tr>
<tr>
<td>SYS1.IPLPARM</td>
<td>CPL001</td>
</tr>
<tr>
<td>SYS1.PARMLIB</td>
<td>CSYSG1</td>
</tr>
<tr>
<td>SYS1.UNICODE.PARMLIB</td>
<td>PPSMP1</td>
</tr>
<tr>
<td>SYS1.PARMLIB.INSATLL</td>
<td>G1B01B</td>
</tr>
<tr>
<td>FixedPath</td>
<td>Parmlib</td>
</tr>
<tr>
<td>Label</td>
<td>SP23-ZOS-Parmlib</td>
</tr>
<tr>
<td>Name</td>
<td>Parmlib</td>
</tr>
</tbody>
</table>

**Naming rules**

- OSZReport

**Relationships**

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>appliesTo</td>
<td>ZReportFile</td>
<td>sys.zOS.ZOS</td>
<td></td>
</tr>
</tbody>
</table>
ZReportFile

ZOSTASK
Address spaces are the instance of executing programs. All system and user programs execute within an address space.

### Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>JobName</td>
<td>The name of the address space. Address spaces are also referred to as jobs. Job names are not guaranteed to be unique within a z/OS system but they usually are because it is difficult to manage them otherwise. Long-running jobs very likely have unique job names. Transient jobs can sometimes have the same job name and even run concurrently but they are not really of interest to this DLA. If the discovery encounters multiple instances of the jobname, only the first instance is represented in the IdML book.</td>
<td>IMPT1IRM</td>
</tr>
<tr>
<td>JobUserid</td>
<td>The user ID associated with the job.</td>
<td>STCIMSU</td>
</tr>
</tbody>
</table>
| JobType       | The type determines the nature of the job and how support personnel typically interact with it. *Started class* includes jobs that are initiated by the START command. They differ from the system type in that they have a JES Jobid with the “S” prefix. Interaction with the job is typically through the MODIFY command. Started tasks are often long-running and are analogous to a daemon under *nix. *Batch jobs* include jobs that are started by JES. They have a JES Jobid with the “J” prefix. Jobs are often short-running. *System* is used for jobs that are considered system jobs. They are typically started early in the IPL and often have have no JES Jobid value. SYS jobs usually run for the life of the IPL and are analogous to a device driver process. *TSO User* is used for jobs that are TSO Users. They have a JES Jobid with the “T” prefix. | Started class
Batch job
System
TSO User                      |
| JobFunction   | A descriptive string that can assist support personnel to identify the job.                                                                                                                               | SYSTEM_VTAM |
| ProcStep      | Identifies the JCL procedure name of the job. This is really only useful if the job is long-running and has only one JCL procedure with one job step.                                                           | IMSIRLM     |
| StepName      | Identifies the Jobstep name of the job. This is really only useful if the job is long-running and has only one job step.                                                                                | IMPT1IRM    |
sys.zOS.AddressSpace

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pgm</td>
<td>The name of the program load module.</td>
<td>DXRRLM00</td>
</tr>
<tr>
<td>Allocations</td>
<td>A list of DDNAMEs and associated DSNs. The list is in a table format</td>
<td>STEPLIB SYS1.LINKLIB SYSRES</td>
</tr>
<tr>
<td></td>
<td>with three columns: DDName, DatasetName, and Volume.</td>
<td></td>
</tr>
<tr>
<td>PgmParm</td>
<td>The optional parameter string that was passed to the program when</td>
<td>PRLM,1,NODISCON,5,1,6,NO,,IMP0IRM</td>
</tr>
<tr>
<td></td>
<td>it was started.</td>
<td></td>
</tr>
</tbody>
</table>

Requirements for discovery

- Filtering enabled (which is the default for started class).

**Note:** Unlike other classes that have Class, Attribute, and Relationship level filters, the address space filtering also provide Jobname wildcarded discover criteria (default *) and JobType level filtering (default StatedTask).

- Attributes marked with an asterisk (*) are only discovered when the DLA runs in authorized mode.

CDM specific information

<table>
<thead>
<tr>
<th>CDM naming policy</th>
<th>sys.zOS.AddressSpace</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>&lt;JobName&gt;&lt;SMFID &gt;</td>
<td>IMPT1IRM-OMO1</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;JobName&gt;&lt;SMFID &gt;&lt;AddressSpace&gt;</td>
<td>IMPT1IRM-OMO1-AddressSpace</td>
</tr>
<tr>
<td>sourceToken</td>
<td>&lt;JobName&gt;&lt;SMFID &gt;&lt;AddressSpace&gt;</td>
<td>IMPT1IRM-OMO1-AddressSpace</td>
</tr>
</tbody>
</table>

Naming rules

JobName and runsOn relationship to zOS.

Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>runsOn</td>
<td>sys.zOS.AddressSpace</td>
<td>sys.zOS.ZOS</td>
<td>Part of a primary naming rule along with the JobName attribute.</td>
</tr>
<tr>
<td>federates</td>
<td>sys.zOS.IMSSubsystem</td>
<td>sys.zOS.AddressSpace</td>
<td></td>
</tr>
<tr>
<td>federates</td>
<td>sys.zOS.DB2Subsystem</td>
<td>sys.zOS.AddressSpace</td>
<td></td>
</tr>
<tr>
<td>federates</td>
<td>sys.zOS.MQSubsystem</td>
<td>sys.zOS.AddressSpace</td>
<td></td>
</tr>
<tr>
<td>uses</td>
<td>sys.zOS.AddressSpace</td>
<td>sys.zOS.IMSSubsystem</td>
<td></td>
</tr>
<tr>
<td>uses</td>
<td>sys.zOS.AddressSpace</td>
<td>sys.zOS.DB2Subsystem</td>
<td></td>
</tr>
<tr>
<td>uses</td>
<td>sys.zOS.AddressSpace</td>
<td>sys.zOS.MQSubsystem</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>sys.zOS.AddressSpace</td>
<td>net.TcpPort</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>sys.zOS.AddressSpace</td>
<td>net.UdpPort</td>
<td></td>
</tr>
<tr>
<td>realizes</td>
<td>sys.zOS.AddressSpace</td>
<td>app.j2ee.websphere.WebSphereServer</td>
<td></td>
</tr>
<tr>
<td>uses</td>
<td>sys.zOS.AddressSpace</td>
<td>dev.StorageVolume</td>
<td>For all DASD volumes used in the allocations.</td>
</tr>
</tbody>
</table>
**net.TcpPort**

Each net.TcpPort is a TCP/IP port that is mapped to an active address space.

**Attributes**

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>The port number</td>
<td>80</td>
<td></td>
</tr>
</tbody>
</table>

**Requirements for discovery**

- Filtering enabled (which is the default).

**CDM specific information**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>net.TcpPort</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM attribute</td>
<td>Value</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;Port&gt;-&lt;IPv4Address&gt;</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;Port&gt;-&lt;IPv4Address&gt;-TcpPort</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;Port&gt;-&lt;IPv4Address&gt;-TcpPort</td>
</tr>
</tbody>
</table>

**Naming rules**

None

**Relationships**

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>bindsTo</td>
<td>net.TcpPort</td>
<td>net.IpInterface</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>sys.zOS.AddressSpace</td>
<td>net.TcpPort</td>
<td></td>
</tr>
</tbody>
</table>
net.UdpPort

Each net.UdpPort is a UDP IP port that is mapped to an active address space.

**Attributes**

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>The port number</td>
<td>1305</td>
</tr>
</tbody>
</table>

**Requirements for discovery**

- Filtering enabled (which is the default).

**CDM specific information**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>net.UdpPort</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM attribute</td>
<td>Value</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;Port&gt;-&lt;IPv4Address&gt;</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;Port&gt;-&lt;IPv4Address&gt;-UdpPort</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;Port&gt;-&lt;IPv4Address&gt;-UdpPort</td>
</tr>
</tbody>
</table>

**Naming rules**

None

**Relationships**

<table>
<thead>
<tr>
<th>Link Type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>bindsTo</td>
<td>net.UdpPort</td>
<td>net.IpInterface</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>sys.zOS.AddressSpace</td>
<td>net.UdpPort</td>
<td></td>
</tr>
</tbody>
</table>
**net.BindAddress**

This class is a combination of IP address and Port. It is used as part of naming rules for some other classes, such as WebSphereServer.

**Attributes**

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>PortNumber</td>
<td>The port number</td>
<td>5655</td>
</tr>
<tr>
<td>Path</td>
<td>(none)</td>
<td>(none)</td>
</tr>
</tbody>
</table>

**Requirements for discovery**

- ZOSTASK and subsystem books
- Filtering enabled (which is the default).

**CDM specific information**

<table>
<thead>
<tr>
<th>CDM naming policy</th>
<th>net.BindAddress</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;PortNumber&gt;-&lt;SMFID&gt;-BindAddress</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;PortNumber&gt;-&lt;SMFID&gt;-BindAddress</td>
</tr>
</tbody>
</table>

**Naming rules**

None

**Relationships**

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>bindsAsPrimary</td>
<td>net.BindAddress</td>
<td>net.IPv4Address</td>
<td></td>
</tr>
<tr>
<td>bindsTo</td>
<td>net.BindAddress</td>
<td>net.IPv4Address</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>app.j2ee.websphere.WebSphereServer</td>
<td>net.BindAddress</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>sys.zOS.IMSSubsystem</td>
<td>net.BindAddress</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>sys.zOS.DB2Subsystem</td>
<td>net.BindAddress</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>sys.zOS.MQSubsystem</td>
<td>net.BindAddress</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>sys.zOS.CICSRegion</td>
<td>net.BindAddress</td>
<td></td>
</tr>
</tbody>
</table>
storage.StorageSubSystem

storage.StorageSubSystem
storage.Storage subsystem

Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manufacturer</td>
<td>Manufacturer</td>
<td>IBM</td>
</tr>
<tr>
<td>SerialNumber</td>
<td>Sequence Number</td>
<td>000000022562</td>
</tr>
<tr>
<td>Type</td>
<td>Literal &quot;StorageDevice&quot;</td>
<td>StorageDevice</td>
</tr>
<tr>
<td>Model</td>
<td>Model number and Model Type number concatenated, for example, concatenate 2105 and F20 (no hyphen.)</td>
<td>2105F20</td>
</tr>
</tbody>
</table>

Requirements for discovery
- Filtering enabled (which is the default).

CDM specific information

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>storage.StorageSubSystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;Manufacturer&gt;–&lt;SerialNumber&gt;–&lt;Model&gt;–</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;Manufacturer&gt;–&lt;SerialNumber&gt;–&lt;Model&gt;–&lt;StorageSubsystem&gt;</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;Manufacturer&gt;–&lt;SerialNumber&gt;–&lt;Model&gt;–&lt;StorageSubsystem&gt;</td>
</tr>
</tbody>
</table>

Naming rules
Manufacturer, SequenceNumber, and Model.

Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains</td>
<td>storage.StorageSubSystem</td>
<td>dev.StorageVolume</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
dev.StorageVolume

DASD Storage volume

Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeviceID</td>
<td>Device Number</td>
<td>1000</td>
</tr>
<tr>
<td>Name</td>
<td>DASD Volume Serial Number</td>
<td>VOL001</td>
</tr>
<tr>
<td>NumOfCylinders</td>
<td>Number of Cylinders this volume has</td>
<td>10017</td>
</tr>
</tbody>
</table>

Requirements for discovery

- Filtering enabled (which is the default).
- The DLA must be running in APF-authorized mode.

CDM specific information

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>dev.StorageVolume</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Name Policy</td>
<td></td>
</tr>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;Volume&gt;-&lt;DeviceNumber&gt;-&lt;SMFID&gt;</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;Volume&gt;-&lt;DeviceNumber&gt;-&lt;SMFID&gt;-StorageVolume</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;Volume&gt;-&lt;DeviceNumber&gt;-&lt;SMFID&gt;-StorageVolume</td>
</tr>
</tbody>
</table>

Naming rules

Name and contains relationship from storage.StorageSubSystem.

Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>contains</td>
<td>storage.StorageSubSystem</td>
<td>dev.StorageVolume</td>
<td>Naming rule for StorageVolume</td>
</tr>
<tr>
<td>memberOf</td>
<td>dev.StorageVolume</td>
<td>storage.StorageSubSystem</td>
<td></td>
</tr>
<tr>
<td>accesses</td>
<td>sys.zOS.ZOS</td>
<td>dev.StorageVolume</td>
<td>All online DASD</td>
</tr>
<tr>
<td>uses</td>
<td>sys.zOS.AddressSpace</td>
<td>dev.StorageVolume</td>
<td>Based on allocations</td>
</tr>
</tbody>
</table>
dev.StorageVolume

DB2 Subsystem
DB2 subsystems consist of several specialized address spaces components running on a z/OS system. Each provide different DB2 services for the subsystem. Each subsystem has one address space known as the Master which regulates the DB2 subsystem as a whole. Attributes marked with an asterisk (*) are discovered only when the DLA runs in authorized mode.

### Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Subsystem) SubsystemName</td>
<td>The 1-4 character name of the DB2 subsystem.</td>
<td>DB7A</td>
</tr>
<tr>
<td>(Subsystem) CommandPrefixName</td>
<td>A prefix used by support personnel to direct commands to the DB2 subsystem.</td>
<td>DB7A</td>
</tr>
<tr>
<td>(Subsystem) ControllingAddressSpace</td>
<td>The Jobname of the Master address space. This is needed to identify the address space to support personnel.</td>
<td>DB7AMSTR</td>
</tr>
<tr>
<td>(Subsystem) VersionString</td>
<td>DB2 Version Release and Modification level.</td>
<td>8.1.0</td>
</tr>
<tr>
<td>SysDatabaseMaxAlteredTs</td>
<td>Maximum ALTEREDTS value for the SYSIBM.SYSDATABASE table. This attribute enables changes to the SYSDATABASE catalog to be tracked.</td>
<td>datestamp</td>
</tr>
<tr>
<td>SysTableSpaceMaxAlteredTs</td>
<td>Maximum ALTEREDTS value for the SYSIBM.SYSTABLESPACE table. This attribute enables changes to the SYSTABLESPACE catalog to be tracked.</td>
<td>datestamp</td>
</tr>
<tr>
<td>SysTablesMaxAlteredTs</td>
<td>Maximum ALTEREDTS value for the SYSIBM.SYSTABLES table. This attribute enables changes to the SYSTABLES catalog to be tracked.</td>
<td>datestamp</td>
</tr>
<tr>
<td>SysIndexesMaxAlteredTs</td>
<td>Maximum ALTEREDTS value for the SYSIBM.SYSINDEXES table. This attribute enables changes to the SYSINDEXES catalog to be tracked.</td>
<td>datestamp</td>
</tr>
<tr>
<td>SysColumnsMaxAlteredTs</td>
<td>Maximum ALTEREDTS value for the SYSIBM.SYSCOLUMNS table. This attribute enables changes to the SYSCOLUMNS catalog to be tracked.</td>
<td>datestamp</td>
</tr>
</tbody>
</table>

### Requirements for discovery

- DB2 Subsystem book has all attributes and all relationships except AddressSpaces.
- ZOSTASK book has naming rules and relationships to AddressSpaces.
- Filtering enabled (which is the default).
- DB2 Subsystem and related address spaces must be active and fully initialized.
- Attributes and relationships are marked with an asterisk (*) are discovered only when the DLA runs in authorized mode.
- The user ID running the DLA must have authority to issue the DB2 DISPLAY DB(*) and DISPLAY GROUP(*) commands.
The user ID running the DLA must have read access to the following DB2 Tables:

- SYSIBM.SYSDATABASE for SysdatabaseMaxAlterdts attribute
- SYSIBM.SYSTABLESPACE for SystablespaceMaxAlterdts attribute
- SYSIBM.SYSTABLES for SystablesMaxAlterdts attribute
- SYSIBM.SYSINDEXES for SysindexesMaxAlterdts attribute
- SYSIBM.SYSCOLUMNS for SyscolumnsMaxAlterdts attribute

Rexx bind for DSNREXX interface must be run for Sys* attributes.

CDM specific information

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.DB2Subsystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>&lt;SubsystemName&gt;-&lt;SMFID&gt;</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;SubsystemName&gt;-&lt;SMFID&gt;-DB2Subsystem</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;SubsystemName&gt;-&lt;SMFID&gt;-DB2Subsystem</td>
</tr>
</tbody>
</table>

Example

- DB7A-OMO1
- DB7A-OMO1-DB2Subsystem
- DB7A-OMO1-DB2Subsystem

Naming rules

SubsystemName and hostedDependency relationship to zOS.

Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>hostedDependency</td>
<td>sys.zOS.DB2Subsystem</td>
<td>sys.zOS.ZOS</td>
<td>Part of a primary naming rule along with SubsystemName attribute.</td>
</tr>
</tbody>
</table>
sys.DB2DataSharingGroup

A DB2 Data Sharing Group can be used by one or more DB2 subsystems in the same sysplex to share the same data.

Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>(SysplexGroup) Name</td>
<td>Data Sharing Group Name, 1-8 chars</td>
<td></td>
</tr>
<tr>
<td>GroupAttachName</td>
<td>A 1-4 character name that can also be used to identify a data sharing group. CICS, IMS and TSO can use this name to access DB2 rather than having to use subsystem name.</td>
<td>DBAT</td>
</tr>
<tr>
<td>(SysplexGroup) GroupFunction</td>
<td>Literal “DB2DataSharing”</td>
<td>sys.zOS.DB2DataSharing</td>
</tr>
<tr>
<td>VersionString</td>
<td>Group level</td>
<td>8.1.0</td>
</tr>
<tr>
<td>SysDatabaseMaxAlteredTs</td>
<td>Maximum ALTEREDTS value for the SYSIBM.SYSDATABASE table. This attribute enables changes to the SYSDATABASE catalog to be tracked.</td>
<td>datestamp</td>
</tr>
<tr>
<td>SysTableSpaceMaxAlteredTs</td>
<td>Maximum ALTEREDTS value for the SYSIBM.SYSTABLESPACE table. This attribute enables changes to the SYSTABLESPACE catalog to be tracked.</td>
<td>datestamp</td>
</tr>
<tr>
<td>SysTablesMaxAlteredTs</td>
<td>Maximum ALTEREDTS value for the SYSIBM.SYSTABLES table. This attribute enables changes to the SYSTABLES catalog to be tracked.</td>
<td>datestamp</td>
</tr>
<tr>
<td>SysIndexesMaxAlteredTs</td>
<td>Maximum ALTEREDTS value for the SYSIBM.SYSINDEXES table. This attribute enables changes to the SYSINDEXES catalog to be tracked.</td>
<td>datestamp</td>
</tr>
<tr>
<td>SysColumnsMaxAlteredTs</td>
<td>Maximum ALTEREDTS value for the SYSIBM.SYSCOLUMNS table. This attribute enables changes to the SYSCOLUMNS catalog to be tracked.</td>
<td>datestamp</td>
</tr>
</tbody>
</table>

Requirements for discovery

- DB2 Subsystem book has all attributes and all relationships except AddressSpaces.
- Filtering enabled (which is the default).
- DB2 subsystem and related address spaces must be active and fully initialized.
- Attributes/Relationships marked with an asterisk (*) are only discovered when the DLA runs in authorized mode.
- The user ID running the DLA must have authority to issue the DISPLAY GROUP(*) commands.

CDM specific information

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.DB2DataSharingGroup</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;name&gt;-&lt;sysplex&gt;</td>
</tr>
</tbody>
</table>
sys.DB2DataSharingGroup

<table>
<thead>
<tr>
<th>CDM Attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>id (IdML)</td>
<td>&lt;name&gt;-&lt;sysplex&gt;-DB2DataSharingGroup</td>
<td></td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;name&gt;-&lt;sysplex&gt;-DB2DataSharingGroup</td>
<td></td>
</tr>
</tbody>
</table>

Naming rules
Name and contains relationship from Sysplex.

Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>contains</td>
<td>sys.zOS.Sysplex</td>
<td>sys.zOS.DB2DataSharingGroup</td>
<td></td>
</tr>
<tr>
<td>federates</td>
<td>sys.zOS.DB2DataSharingGroup</td>
<td>sys.zOS.DB2Subsystem</td>
<td></td>
</tr>
<tr>
<td>contains</td>
<td>sys.zOS.DB2DataSharingGroup</td>
<td>app.db.db2.Db2Database</td>
<td></td>
</tr>
</tbody>
</table>
app.db.db2.Db2Database

DB2 databases.

**Attributes**

<table>
<thead>
<tr>
<th>CDM Attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the database</td>
<td></td>
</tr>
</tbody>
</table>

**Requirements for discovery**

- Filtering enabled. By default, this is SUPPRESSED because there can be many instances.
- DB2 subsystem and related address spaces must be active and fully initialized.
- The user ID running the DLA must have authority to issue the DB2 DISPLAY DB(*) command.

**CDM specific information**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.db.Db2Database</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CDM Attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>For DataSharing: &lt;name&gt;-&lt;data sharing group&gt;-&lt;sysplex&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For not shared: &lt;name&gt;-&lt;subsystem&gt;</td>
<td></td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;label&gt;-Db2Database</td>
<td></td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;label&gt;-Db2Database</td>
<td></td>
</tr>
</tbody>
</table>

**Naming rules**

Name and contains relationship from sys.zOS.DB2DataSharingGroup or sys.zOS.DB2Subsystem (if no sys.zOS.DB2DataSharingGroup).

**Relationships**

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>contains</td>
<td>sys.zOS.DB2Subsystem</td>
<td>app.db.Db2Database</td>
<td></td>
</tr>
<tr>
<td>contains</td>
<td>sys.zOS.DB2DataSharingGroup</td>
<td>app.db.Db2Database</td>
<td></td>
</tr>
<tr>
<td>contains</td>
<td>app.db.Db2Database</td>
<td>app.db.Db2Tablespace</td>
<td></td>
</tr>
</tbody>
</table>
**app.db.db2.Db2TableSpace**

**app.db.db2.Db2TableSpace**

DB2 tablespaces. By default, this is SUPPRESSED because there can be many instances.

**Attributes**

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the tablespace</td>
<td></td>
</tr>
</tbody>
</table>

**Requirements for discovery**

- Filtering enabled.
- DB2 subsystem and related address spaces must be active and fully initialized.
- The user ID running the DLA must have authority to issue the DB2 DISPLAY DB(*) command.

**CDM specific information**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.db.db2.Db2TableSpace</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>Label</td>
<td>For DataSharing:</td>
</tr>
<tr>
<td></td>
<td>&lt;name&gt;-&lt;database&gt;-&lt;data sharing group&gt;-&lt;sysplex&gt;.</td>
</tr>
<tr>
<td></td>
<td>For not shared:</td>
</tr>
<tr>
<td></td>
<td>&lt;name&gt;-&lt;database&gt;-</td>
</tr>
<tr>
<td></td>
<td>&lt;subsystem&gt;.</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;label&gt;- Db2TableSpace</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;label&gt;- Db2TableSpace</td>
</tr>
</tbody>
</table>

**Naming rules**

Name and contains relationship from app.db.db2.Db2Database.

**Relationships**

<table>
<thead>
<tr>
<th>Link Type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>contains</td>
<td>app.db.db2.Db2Database</td>
<td>app.db.db2.Db2TableSpace</td>
<td></td>
</tr>
</tbody>
</table>
IMS subsystems consist of several specialized address spaces running on a z/OS system. Each provide different IMS services for the subsystem. Each subsystem has one address space known as the Control region which regulates the IMS subsystem as a whole. The subsystem name is also known to all address spaces in the subsystem as the IMSID.

### Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Subsystem) SubsystemName</td>
<td>The 1-4 character name of the IMS subsystem.</td>
<td>IMN3</td>
</tr>
<tr>
<td>(Subsystem) VersionString</td>
<td>The version and release of the IMS control region</td>
<td>8.1</td>
</tr>
<tr>
<td>(Subsystem) CommandPrefixName</td>
<td>A prefix used by support personnel to direct commands to the IMS subsystem. It is often the same as the IMSID.</td>
<td>IMN3</td>
</tr>
<tr>
<td>(Subsystem) ControllingAddressSpace</td>
<td>The Jobname of the Control region. This is needed to identify the address space to support personnel.</td>
<td>IMN1CTL</td>
</tr>
<tr>
<td>IMSSubsysType</td>
<td>There are three types of IMS subsystem and this is reflected by the type of Control region, they are:</td>
<td>DCCTL</td>
</tr>
<tr>
<td></td>
<td>• DCCTL has only the Transaction Manager component and provides access to the IMS message queues for IMS applications.</td>
<td>DBCTL</td>
</tr>
<tr>
<td></td>
<td>• DBCTL has only the Database Manager component and provides IMS database functions.</td>
<td>DB/DC</td>
</tr>
<tr>
<td></td>
<td>• DB/DC provides Transaction Manager and Database Manager functions for large application systems.</td>
<td></td>
</tr>
<tr>
<td>IRLMGroupName</td>
<td>The XCF group name used by all CQS address spaces that share the same set of XCF structures.</td>
<td>IMN0IRM</td>
</tr>
<tr>
<td>CQSGROUPName</td>
<td>The XCF group name used by all CQS address spaces that share the same set of XCF structures. All CQS group names have the prefix CQS.</td>
<td>CQSIMN3</td>
</tr>
</tbody>
</table>
## Requirements for discovery

- IMS Subsystem book has all attributes and all relationships except to AddressSpaces.
- ZOSTASK book has naming rules and relationships to AddressSpaces.
- Filtering enabled (which is the default). Note that following Attribute level filters are disabled by default because they can be large at some sites. The customer needs to consciously choose to discover these: Transactions, Programs.
- IMS subsystem and related address spaces must be active and fully initialized.
- Attributes and relationships that are marked with an asterisk (*) are discovered only when the DLA runs in authorized mode.
- No configuration datasets are read for this discovery, that is, no special data set authority required.

## CDM specific information

### CDM Naming Policy

<table>
<thead>
<tr>
<th>CDM Attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td><code>&lt;SubsystemName&gt;</code>&lt;SMFID &gt;</td>
<td>IMN3-OMO1</td>
</tr>
<tr>
<td>id (IdML)</td>
<td><code>&lt;SubsystemName&gt;</code>&lt;SMFID &gt;-IMSSubsystem</td>
<td>IMN3-OMO1-IMSSubsystem</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td><code>&lt;SubsystemName&gt;</code>&lt;SMFID &gt;-IMSSubsystem</td>
<td>IMN3-OMO1-IMSSubsystem</td>
</tr>
</tbody>
</table>

### Naming rules

SubsystemName and hostedDependency relationship to sys.zOS.zOS.

ControllingAddressSpace and hostedDependency relationship to sys.zOS.zOS.
### Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>hostedDependency</td>
<td>sys.zOS.IMSSubsystem</td>
<td>sys.zOS.zOS</td>
<td>Part of primary naming rule a long with SubsystemName attribute.</td>
</tr>
<tr>
<td>federates</td>
<td>sys.zOS.IMSSubsystem</td>
<td>sys.zOS.AddressSpace</td>
<td>For all related Control region, DLI, DBRC, IRLM, CQS, FDBR, CONNECT, OM, SCI, RM,</td>
</tr>
<tr>
<td>uses</td>
<td>sys.zOS.AddressSpace</td>
<td>sys.zOS.IMSSubsystem</td>
<td>For all related Dependant Regions, for example, MPR, JMP, IFP, BMP, JBP, DBT</td>
</tr>
<tr>
<td>uses</td>
<td>sys.zOS.IMSSubsystem</td>
<td>sys.zOS.DB2Subsystem</td>
<td>There may be 0 or 1 DB2 related to the IMS.</td>
</tr>
<tr>
<td>uses</td>
<td>sys.zOS.IMSSubsystem</td>
<td>sys.zOS.MQSubsystem</td>
<td>There may be 0 or 1 MQ related to the IMS.</td>
</tr>
<tr>
<td>uses</td>
<td>sys.zOS.IMSSubsystem</td>
<td>sys.zOS.IMSSysplexGroup</td>
<td>For related IRLMGROUP, CQSGROUP and IMSPLEX.</td>
</tr>
<tr>
<td>accessedVia</td>
<td>sys.zOS.IMSSubsystem</td>
<td>net.BindAddress</td>
<td>For all related federated address spaces that are listening on a TCP and UDP Port. This relationship is used by the DLA sensor to stitch relationships with distributed resources.</td>
</tr>
<tr>
<td>uses</td>
<td>sys.zOS.CICSRegion</td>
<td>sys.zOS.IMSSubsystem</td>
<td></td>
</tr>
<tr>
<td>Contains</td>
<td>sys.zOS.IMSSubsystem</td>
<td>sys.zOS.IMSTransaction</td>
<td></td>
</tr>
<tr>
<td>Contains</td>
<td>sys.zOS.IMSSubsystem</td>
<td>sys.zOS.IMSProgram</td>
<td></td>
</tr>
<tr>
<td>Contains</td>
<td>sys.zOS.IMSSubsystem</td>
<td>sys.zOS.IMSDatabase</td>
<td></td>
</tr>
</tbody>
</table>
sys.zOS.IMSSysplexGroup

IMS Sysplex Groups used by related CQS, IRLM, and IMSPlex address spaces.

Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The XCF group name that represents a group of address spaces that share data. Group members can execute on more than one z/OS system (except when Monoplex) and there can be multiple members per z/OS system.</td>
<td>CSLIMN0</td>
</tr>
<tr>
<td>GroupFunction</td>
<td>Indicates the type of data that is shared between the group members. The IMSPlex function shares IMS data objects between IMS subsystems. The IMS CQS function shares IMS message queues objects between IMS subsystems. The IMS CQS group can be used by multiple IMS subsystems but an IMS subsystem can only use one IMSCQS group. The IMS IRLM function shares IMS database lock data for database integrity. The IMSIRLM group can be used by multiple IMS subsystems and an IMS subsystem can use multiple IMSIRLM groups. IRLM sysplex groups exist for other subsystem types such as DB2 but it is because of the many to many relationship between IMS and IRLM that we discover IMSIRLM.</td>
<td>IMSPlex, IMS CQS, IMS IRLM</td>
</tr>
</tbody>
</table>

Requirements for discovery

- ZOSTASK and subsystem books where applicable.
- Filtering enabled (which is the default).
- DLA runs in APF authorized mode.

CDM specific information

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.IMSSysplexGroup</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;Name&gt;-&lt;SysplexName&gt;</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;Name&gt;-&lt;SysplexName&gt;-IMSSysplexGroup</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;Name&gt;-&lt;SysplexName&gt;-IMSSysplexGroup</td>
</tr>
</tbody>
</table>

Naming rules

Name and contains relationship from Sysplex.
### Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>contains</td>
<td>sys.zOS.Sysplex</td>
<td>sys.zOS.IMSSysplexGroup</td>
<td></td>
</tr>
<tr>
<td>memberOf</td>
<td>sys.zOS.AddressSpace</td>
<td>sys.zOS.IMSSysplexGroup</td>
<td></td>
</tr>
<tr>
<td>federates</td>
<td>sys.zOS.IMSSysplexGroup</td>
<td>sys.zOS.IMSSubsystem</td>
<td></td>
</tr>
</tbody>
</table>
**sys.zOS.IMSTransaction**

IMS Transactions.

**Attributes**

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the transaction</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Requirements for discovery**

- Filtering enabled. By default this is SUPPRESSED since there can be many instances. Note that the `sys.zOS.IMSSubsystem` attribute `TransactionsChecksum` enables CMDB to detect Transaction changes without needing to populate many `sys.zOS.IMSTransaction` instances.
- IMS subsystem and related address spaces must be active and fully initialized.
- The DLA must be running in APF authorized mode.

**CDM specific information**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.IMSTransaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM AttributeValue</td>
<td>Value</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;name&gt;-&lt;ims subsystem&gt;</td>
</tr>
<tr>
<td>id (ldML)</td>
<td>&lt;label&gt;- IMSTransaction</td>
</tr>
<tr>
<td>sourceToken (ldML)</td>
<td>&lt;label&gt;- IMSTransaction</td>
</tr>
</tbody>
</table>

**Naming rules**

Name and contains relationship from `sys.zOS.IMSSubsystem`.

**Relationships**

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>contains</td>
<td>sys.zOS.IMSSubsystem</td>
<td>sys.zOS.IMSTransaction</td>
<td></td>
</tr>
</tbody>
</table>
sys.zOS.IMSProgram

sys.zOS.IMSProgram
IMS programs.

Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the program</td>
<td></td>
</tr>
</tbody>
</table>

Requirements for discovery

- Filtering enabled. By default this is SUPPRESSED since there may be many instances. Note that the sys.zOS.IMSSubsystem attribute ProgramsChecksum enables CMDB to detect Program changes without needing to populate many sys.zOS.IMSProgram instances.
- IMS subsystem and related address spaces must be active and fully initialized.
- The DLA must be running in APF authorized mode.

CDM specific information

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.IMSProgram</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;name&gt;-&lt;ims subsystem&gt;</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;label&gt;- IMSProgram</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;label&gt;- IMSProgram</td>
</tr>
</tbody>
</table>

Naming rules
Name and contains relationship from the sys.zOS.IMSSubsystem.

Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>contains</td>
<td>sys.zOS.IMSSubsystem</td>
<td>sys.zOS.IMSProgram</td>
<td></td>
</tr>
</tbody>
</table>
sys.zOS.IMSDatabase

IMS databases.

Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the database</td>
<td></td>
</tr>
<tr>
<td>DatabaseType</td>
<td>Enumeration: Database, Area, Partition</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Database: A database provides for the storing and control of business data, independent from (but not separate from the processing requirements of) one or more applications. The database records details (attributes) of particular items (entities) and the relationships between the different types of entities. IMS uses a hierarchical model as the basic method of storing data.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Area: Data entry databases (DEDBs) provide efficient storage for and access to large volumes of data. DEDBs also provide a high level of availability of that data. A DEDB can be organized into one or more data sets called areas. Each area contains the entire data structure.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Partition: HALDB allows the grouping of DL/I database records into sets of partitions that are treated as a single database while permitting functions to be performed independently for each partition. Each HALDB partition has the same capacity limit as a DL/I non-HALDB database.</td>
<td></td>
</tr>
</tbody>
</table>

Requirements for discovery

- Filtering enabled. By default this is SUPPRESSED since there may be many instances. Note that the sys.zOS.IMSSubsystem attribute DatabasesChecksum enables CMDB to detect Database changes without needing to populate many sys.zOS.IMSDatabase instances.
- IMS subsystem and related address spaces must be active and fully initialized.
- The DLA must be running in APF authorized mode.
sys.zOS.IMSDatabase

**CDM specific information**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.IMSDatabase</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CDM Attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>&lt;name&gt;-&lt;ims subsystem&gt;</td>
<td></td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;label&gt;- IMSDatabase</td>
<td></td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;label&gt;- IMSDatabase</td>
<td></td>
</tr>
</tbody>
</table>

**Naming rules**
Name and contains relationship from sys.zOS.IMSSubsystem.

**Relationships**

<table>
<thead>
<tr>
<th>Link Type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>contains</td>
<td>IMSSubsystem</td>
<td>IMSDatabase</td>
<td></td>
</tr>
</tbody>
</table>
sys.zOS.IMSDatabase

sys.zOS.MQ Subsystem
MQ subsystems consist of a Queue Manager address space and a MQ Channel Initiator address space running on a z/OS system.

### Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Subsystem) Subsystemname</td>
<td>The 1-4 character name of the MQ subsystem.</td>
<td>MQ01</td>
</tr>
<tr>
<td>(Subsystem) CommandPrefixName</td>
<td>A prefix used by support personnel to direct commands to the MQ subsystem. It is often the same as the subsystem name.</td>
<td>MQ01</td>
</tr>
<tr>
<td>(Subsystem) ControllingAddressSpace</td>
<td>The Jobname of the Queue Manager address space. This identifies the address space to support personnel.</td>
<td>CSQ1MSTR</td>
</tr>
<tr>
<td>(Subsystem) VersionString</td>
<td>MQ Version Release and modification level.</td>
<td>5.3.1</td>
</tr>
</tbody>
</table>

### Requirements for discovery

- MQ Subsystem book has all attributes and all relationships except to StorageVolume and AddressSpaces.
- ZOSTASK book has naming rules and relationships to StorageVolume and AddressSpaces.
- The user ID running the DLA must have read access to the hlq.DISPLAY.CHANNEL profile in the MQCMDS class.
- The highest level SCSQAUTH data set must be accessible to the DLA discover job via the LINKLST or STEPLIB.

### CDM specific information

<table>
<thead>
<tr>
<th>CDM naming policy</th>
<th>sys.zOS.MQSubsystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM attribute</td>
<td>Value</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;SubsystemName&gt;-&lt;SMFID&gt;</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;SubsystemName&gt;-&lt;SMFID&gt;-MQSubsystem</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;SubsystemName&gt;-&lt;SMFID&gt;-MQSubsystem</td>
</tr>
</tbody>
</table>

### Naming rules

- SubsystemName and hostedDependency relationship to zOS.
- MQSubsystem accessedVia through BindAddress relationship.

### Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>hostedDependency</td>
<td>sys.zOS.MQSubsystem</td>
<td>sys.zOS.zOS</td>
<td>Part of primary naming rule a long with SubsystemName attribute.</td>
</tr>
<tr>
<td>federates</td>
<td>sys.zOS.MQSubsystem</td>
<td>sys.zOS.AddressSpace</td>
<td>For all related master address space and channel initiator address space.</td>
</tr>
<tr>
<td>uses</td>
<td>sys.zOS.IMSSubsystem</td>
<td>sys.zOS.MQSubsystem</td>
<td>Discovered via IMS</td>
</tr>
</tbody>
</table>
### sys.zOS.MQSubsystem

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>uses</td>
<td>sys.zOS.CICSRegion</td>
<td>sys.zOS.MQSubsystem</td>
<td>Discovered via CICS</td>
</tr>
<tr>
<td>accessedVia</td>
<td>sys.zOS.MQSubsystem</td>
<td>net.BindAddress</td>
<td>For all related federated address spaces that are listening on a TCP and UDP port. This relationship is used by the DLA sensor to stitch relationships with distributed resources.</td>
</tr>
<tr>
<td>federates</td>
<td>sys.zOS.MQSubsystem</td>
<td>app.messaging.mq.MQSenderChannel</td>
<td></td>
</tr>
<tr>
<td>federates</td>
<td>sys.zOS.MQSubsystem</td>
<td>app.messaging.mq.MQReceiverChannel</td>
<td></td>
</tr>
</tbody>
</table>
### Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>BatchHeartbeatInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BatchInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BatchSize</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ConnectionName</td>
<td>Channel connection name</td>
<td>PTHOMO2.PERTHAPC.AU.IBM.COM(1414)</td>
</tr>
<tr>
<td>DataConversion</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DisconnectInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HeaderCompression</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HeartbeatInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KeepAliveInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Label</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LocalCommunicationAddress</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LongRetryCount</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LongRetryTimer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MaxMessageLength</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCAName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCAType</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCAUserId</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageExitName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageExitUserData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageRetryCount</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageRetryExitName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageRetryExitUserData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageRetryInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageRetryTimer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Channel name</td>
<td>OMO1.TO.OMO4</td>
</tr>
<tr>
<td>NonPersistentMessageSpeed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PutAuthority</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QueueSharingGroupDisposition</td>
<td>Queue Disposition</td>
<td>QMGR</td>
</tr>
<tr>
<td>ReceiveExitName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ReceiveExitUserData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SecurityExitName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SecurityExitUserData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SendExitName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SendExitUserData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ShortRetryCount</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Requirements for discovery
• None

CDM specific information

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.messaging.mq.MQSenderChannel</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;name&gt;-&lt;MQ Queue Manager name&gt;-&lt;hostname &gt;</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID&gt;-MQSenderChannel</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID&gt;-MQSenderChannel</td>
</tr>
</tbody>
</table>

Naming rules
Name and federates relationship to sys.zOS.MQSubsystem (MQQueueManager).

Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>federates</td>
<td>sys.zOS.MQSubsystem</td>
<td>app.messaging.mq.MQSenderChannel</td>
<td></td>
</tr>
</tbody>
</table>
### Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>BatchHeartbeatInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BatchInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BatchSize</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ConnectionName</td>
<td>Channel connection name</td>
<td>PTHOMO2.PERTHAPC.AU.IBM.COM(1414)</td>
</tr>
<tr>
<td>DataConversion</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DisconnectInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HeaderCompression</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HeartbeatInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KeepAliveInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LocalCommunicationAddress</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LongRetryCount</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LongRetryTimer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MaxMessageLength</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCAName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCAType</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCAUserId</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageExitName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageExitUserData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageRetryCount</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageRetryExitName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageRetryExitUserData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageRetryInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageRetryTimer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Channel name</td>
<td>OMO4.TO.OMO1</td>
</tr>
<tr>
<td>NonPersistentMessageSpeed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PutAuthority</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QueueSharingGroupDisposition</td>
<td>Queue Disposition</td>
<td>QMGR</td>
</tr>
<tr>
<td>ReceiveExitName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ReceiveExitUserData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SecurityExitName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SecurityExitUserData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SendExitName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SendExitUserData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ShortRetryCount</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ShortRetryTimer</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Requirements for discovery
- None

### CDM specific information

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.messaging.mq.MQReceiverChannel</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;name&gt;-&lt;MQ Queue Manager name&gt;-&lt;hostname&gt;</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID&gt;-MQReceiverChannel</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID&gt;-MQReceiverChannel</td>
</tr>
</tbody>
</table>

### Naming rules
Name and federates relationship to sys.zOS.MQSubsystem (MQQueueManager).

### Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>federates</td>
<td>sys.zOS.MQSubsystem</td>
<td>app.messaging.mq.MQReceiverChannel</td>
<td></td>
</tr>
</tbody>
</table>
### Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>app.messaging.mq.MQClusterReceiverChannelExample</th>
</tr>
</thead>
<tbody>
<tr>
<td>BatchHeartbeatInterval</td>
<td></td>
</tr>
<tr>
<td>BatchInterval</td>
<td></td>
</tr>
<tr>
<td>ConnectionName</td>
<td>Channel connection name</td>
</tr>
<tr>
<td>DataConversion</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>DisconnectInterval</td>
<td></td>
</tr>
<tr>
<td>HeaderCompression</td>
<td></td>
</tr>
<tr>
<td>HeartbeatInterval</td>
<td></td>
</tr>
<tr>
<td>KeepAliveInterval</td>
<td></td>
</tr>
<tr>
<td>LocalCommicationAddress</td>
<td></td>
</tr>
<tr>
<td>LongRetryCount</td>
<td></td>
</tr>
<tr>
<td>LongRetryTimer</td>
<td></td>
</tr>
<tr>
<td>MaxMessageLength</td>
<td></td>
</tr>
<tr>
<td>MCAName</td>
<td></td>
</tr>
<tr>
<td>MCAType</td>
<td></td>
</tr>
<tr>
<td>MCUserId</td>
<td></td>
</tr>
<tr>
<td>MessageExitName</td>
<td></td>
</tr>
<tr>
<td>MessageExitUserData</td>
<td></td>
</tr>
<tr>
<td>MessageRetryCount</td>
<td></td>
</tr>
<tr>
<td>MessageRetryExitName</td>
<td></td>
</tr>
<tr>
<td>MessageRetryUserData</td>
<td></td>
</tr>
<tr>
<td>MessageRetryInterval</td>
<td></td>
</tr>
<tr>
<td>MessageRetryTimer</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Channel name</td>
</tr>
<tr>
<td>NonPersistentMessageSpeed</td>
<td></td>
</tr>
<tr>
<td>PutAuthority</td>
<td></td>
</tr>
<tr>
<td>QueueSharingGroupDisposition</td>
<td>Queue Disposition</td>
</tr>
<tr>
<td>ReceiveExitName</td>
<td></td>
</tr>
<tr>
<td>ReceiveExitUserData</td>
<td></td>
</tr>
<tr>
<td>SecurityExitName</td>
<td></td>
</tr>
<tr>
<td>SecurityExitUserData</td>
<td></td>
</tr>
</tbody>
</table>
app.messaging.mq.MQReceiverChannel

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>app.messaging.mq.MQClusterReceiverChannelExample</th>
</tr>
</thead>
<tbody>
<tr>
<td>SendExitName</td>
<td></td>
</tr>
<tr>
<td>SendExitUserData</td>
<td></td>
</tr>
<tr>
<td>ShortRetryCount</td>
<td></td>
</tr>
<tr>
<td>ShortRetryTimer</td>
<td></td>
</tr>
<tr>
<td>SSLCipherSpecification</td>
<td></td>
</tr>
<tr>
<td>SSLClientAuthentication</td>
<td></td>
</tr>
<tr>
<td>SSLPeer</td>
<td></td>
</tr>
<tr>
<td>TransmissionQueue</td>
<td></td>
</tr>
<tr>
<td>UserId</td>
<td></td>
</tr>
</tbody>
</table>

**Requirements for discovery**
- None

**CDM specific information**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.messaging.mq.MQReceiverChannel</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>Label</td>
<td><code>&lt;name&gt;~-&lt;MQ Queue Manager name&gt;-&lt;hostname&gt;</code></td>
</tr>
<tr>
<td>id (IdML)</td>
<td><code>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID&gt;-MQReceiverChannel</code></td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td><code>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID&gt;-MQReceiverChannel</code></td>
</tr>
</tbody>
</table>

**Naming rules**
Name and federates relationship to sys.zOS.MQSubsystem (MQQueueManager).

**Relationships**

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>federates</td>
<td>sys.zOS.MQSubsystem</td>
<td>app.messaging.mq.MQClusterReceiverChannel</td>
<td></td>
</tr>
<tr>
<td>connects</td>
<td>app.messaging.mq.MQCluster</td>
<td>app.messaging.mq.MQClusterReceiverChannel</td>
<td></td>
</tr>
</tbody>
</table>
### Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>BatchHeartbeatInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BatchInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BatchSize</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ClusterNameList</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLWLChannelPriority</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLWLChannelRank</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLWLChannelWeight</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ConnectionName</td>
<td>Channel connection name</td>
<td>PTHOMO2.PERTHAPC.AU.IBM.COM(1414)</td>
</tr>
<tr>
<td>DataConversion</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DisconnectInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HeaderCompression</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HeartbeatInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KeepAliveInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LocalCommicationAddress</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LongRetryCount</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LongRetryTimer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MaxMessageLength</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCAName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCAType</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCAUserId</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageExitName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageExitUserData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageRetryCount</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageRetryExitName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageRetryExitUserData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageRetryInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageRetryTimer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Channel name</td>
<td>OMO4.TO.OMO1</td>
</tr>
<tr>
<td>NonPersistentMessageSpeed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PutAuthority</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QueueSharingGroupDisposition</td>
<td>Queue Disposition</td>
<td>QMGR</td>
</tr>
<tr>
<td>ReceiveExitName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ReceiveExitUserData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SecurityExitName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SecurityExitUserData</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**app.messaging.mq.MQSenderChannel**

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>SendExitName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SendExitUserData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ShortRetryCount</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ShortRetryTimer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSLCipherSpecification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSLClientAuthentication</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSLPeer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TransmissionQueue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UserId</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Requirements for discovery**
- None

**CDM specific information**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.messaging.mq.MQReceiverChannel</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;name&gt;-&lt;MQ Queue Manager name&gt;-&lt;hostname&gt;</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID&gt;-MQReceiverChannel</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID&gt;-MQReceiverChannel</td>
</tr>
</tbody>
</table>

**Naming rules**
Name and federates relationship to sys.zOS.MQSubsystem (MQQueueManager).

**Relationships**

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>federates</td>
<td>sys.zOS.MQSubsystem</td>
<td>app.messaging.mq.MQClusterSenderChannel</td>
<td></td>
</tr>
<tr>
<td>connects</td>
<td>app.messaging.mq.MQCluster</td>
<td>app.messaging.mq.MQClusterSenderChannel</td>
<td></td>
</tr>
</tbody>
</table>
### Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>BatchHeartbeatInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BatchInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BatchSize</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ConnectionName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DisconnectInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HeaderCompression</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HeartbeatInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KeepAliveInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Label</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LocalCommunicationAddress</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MaxMessageLength</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCAName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCAType</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCAUserId</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageExitName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageExitUserData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageRetryTimer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ModeName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NonPersistentMessageSpeed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QueueSharingGroupDisposition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ReceiveExitName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ReceiveExitUserData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SecurityExitName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SecurityExitUserData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SendExitName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SendExitUserData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSLCipherSpecification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSLClientAuthentication</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSLPeer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TransmissionQueue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UserId</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Requirements for discovery
- None
CDM specific information

<table>
<thead>
<tr>
<th>CDM Attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>&lt;name&gt;-&lt;MQ Queue Manager name&gt;-&lt;hostname&gt;</td>
<td>OMO4.TO.OM01-MQ01-phtomo1.au.ibm.com</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID&gt;- MQReceiverChannel</td>
<td>OMO4.TO.OM01-MQ01-OM01-MQReceiverChannel</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID&gt;- MQReceiverChannel</td>
<td>OMO4.TO.OM01-MQ01-OM01-MQReceiverChannel</td>
</tr>
</tbody>
</table>

Naming rules
Name and federates relationship to sys.zOS.MQSubsystem (MQQueueManager).

Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>federates</td>
<td>sys.zOS.MQSubsystem</td>
<td>app.messaging.mq.MQClientConnectionChannel</td>
<td></td>
</tr>
</tbody>
</table>
### Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>BatchHeartbeatInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BatchInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BatchSize</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ConnectionName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DisconnectInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HeaderCompression</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HeartbeatInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KeepAliveInterval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Label</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MaxMessageLength</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCAName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCASType</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCAUserId</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageExitName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageExitUserData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageRetryTimer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NonPersistentMessageSpeed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QueueSharingGroupDisposition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ReceiveExitName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ReceiveExitUserData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SecurityExitName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SecurityExitUserData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SendExitName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SendExitUserData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSLCipherSpecification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSLClientAuthentication</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSLPeer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TransmissionQueue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UserId</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Requirements for discovery

- None
**CDM specific information**

<table>
<thead>
<tr>
<th>CDM Attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>&lt;name&gt;-&lt;MQ Queue Manager name&gt;-&lt;hostname&gt;</td>
<td>OMO4.TO.OMO1-MQ01-pthomo1.au.ibm.com</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID&gt;-MQReceiverChannel</td>
<td>OMO4.TO.OMO1-MQ01-OMO1-MQReceiverChannel</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID&gt;-MQReceiverChannel</td>
<td>OMO4.TO.OMO1-MQ01-OMO1-MQReceiverChannel</td>
</tr>
</tbody>
</table>

**Naming rules**
Name and federates relationship to sys.zOS.MQSubsystem (MQQueueManager).

**Relationships**

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>federates</td>
<td>sys.zOS.MQSubsystem</td>
<td>app.messaging.mq.MQServerConnectionChannel</td>
<td></td>
</tr>
</tbody>
</table>
Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFSTRUCT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ConnectionName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DefaultPersistence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DefinitionType</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Get</td>
<td></td>
<td></td>
</tr>
<tr>
<td>InitiationQueue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Label</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MaxMessageLength</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MaxMessageLength</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MaxQueueDepth</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Put</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QSDISP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerControl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerDepth</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerType</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Requirements for discovery
- None

CDM specific information

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.messaging.mq.MQReceiverChannel</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;name&gt;-&lt;MQ Queue Manager name&gt;-&lt;hostname&gt;</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID&gt;-MQReceiverChannel</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID&gt;-MQReceiverChannel</td>
</tr>
</tbody>
</table>

Naming rules
Name and federates relationship to sys.zOS.MQSubsystem (MQQueueManager).

Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>manages</td>
<td>sys.zOS.MQSubsystem</td>
<td>app.messaging.mq.MQLocalQueue</td>
<td></td>
</tr>
</tbody>
</table>
app.messaging.mq.MQRemoteQueue

MQ Local Queue

Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectionName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Label</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Put</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QSDISP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RemoteName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RemoteQueueMgrName</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Requirements for discovery

- None

CDM specific information

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.messaging.mq.MQReceiverChannel</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;name&gt;-&lt;MQ Queue Manager name&gt;-&lt;hostname &gt;</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID - MQReceiverChannel</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID - MQReceiverChannel</td>
</tr>
</tbody>
</table>

Naming rules

Name and federates relationship to sys.zOS.MQSubsystem (MQQueueManager).

Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>manages</td>
<td>sys.zOS.MQSubsystem</td>
<td>app.messaging.mq.MQRemoteQueue</td>
<td></td>
</tr>
</tbody>
</table>
app.messaging.mq.MQAliasQueue

MQ Alias Queue

Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectionName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Label</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Put</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QSDISP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TargetQueue</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Requirements for discovery

- None

CDM specific information

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.messaging.mq.MQReceiverChannel</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;name&gt;<del>&lt;MQ Queue Manager name&gt;</del>&lt;hostname &gt;</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;name&gt;<del>&lt;MQ Queue Mgr&gt;</del>&lt;SMFID &gt;~MQReceiverChannel</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;name&gt;<del>&lt;MQ Queue Mgr&gt;</del>&lt;SMFID &gt;~MQReceiverChannel</td>
</tr>
</tbody>
</table>

Naming rules

Name and federates relationship to sys.zOS.MQSubsystem (MQQueueManager).

Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>manages</td>
<td>sys.zOS.MQSubsystem</td>
<td>app.messaging.mq.MQAliasQueue</td>
<td></td>
</tr>
</tbody>
</table>
app.messaging.mq.MQModelQueue

MQ Model Queue

Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFSTRUCT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ConnectionName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DefaultPersistence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DefinitionType</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Get</td>
<td></td>
<td></td>
</tr>
<tr>
<td>InitiationQueue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Label</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Put</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QSDISP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerControl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerDepth</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerType</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Requirements for discovery

- None

CDM specific information

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.messaging.mq.MQReceiverChannel</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;name&gt;-&lt;MQ Queue Manager name&gt;-&lt;hostname&gt;</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID&gt;-MQReceiverChannel</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID&gt;-MQReceiverChannel</td>
</tr>
</tbody>
</table>

Naming rules

Name and federates relationship to sys.zOS.MQSubsystem (MQQueueManager).

Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>manages</td>
<td>sys.zOS.MQSubsystem</td>
<td>app.messaging.mq.MQModelQueue</td>
<td></td>
</tr>
</tbody>
</table>
app.messaging.mq.MQNameList

MQ Name List

Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Label</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Values</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Requirements for discovery

- None

CDM specific information

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.messaging.mq.MQReceiverChannel</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;name&gt;-&lt;MQ Queue Manager name&gt;-&lt;hostname &gt;</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID &gt;- MQReceiverChannel</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID &gt;- MQReceiverChannel</td>
</tr>
</tbody>
</table>

Naming rules

Name and federates relationship to sys.zOS.MQSubsystem (MQQueueManager).

Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>manages</td>
<td>sys.zOS.MQSubsystem</td>
<td>app.messaging.mq.MQNameList</td>
<td></td>
</tr>
</tbody>
</table>
app.messaging.mq.MQProcess

MQ Process

Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Label</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ApplicationID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ApplicationType</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EnvironmentData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UserData</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Requirements for discovery

- None

CDM specific information

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.messaging.mq.MQReceiverChannel</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td></td>
<td>Example</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;name&gt;-&lt;MQ Queue Manager name&gt;-&lt;hostname&gt;</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID &gt;- MQReceiverChannel</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID &gt;- MQReceiverChannel</td>
</tr>
</tbody>
</table>

Naming rules

Name and federates relationship to sys.zOS.MQSubsystem (MQQueueManager).

Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>manages</td>
<td>sys.zOS.MQSubsystem</td>
<td>app.messaging.mq.MQProcess</td>
<td></td>
</tr>
</tbody>
</table>
app.messaging.mq.MQAuthInfo

app.messaging.mq.MQAuthInfo
MQ AuthInfo

Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Label</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Requirements for discovery
- None

CDM specific information

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.messaging.mq.MQReceiverChannel</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;name&gt;-&lt;MQ Queue Manager name&gt;-&lt;hostname&gt;</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID&gt;-MQReceiverChannel</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID&gt;-MQReceiverChannel</td>
</tr>
</tbody>
</table>

Naming rules
Name and federates relationship to sys.zOS.MQSubsystem (MQQueueManager).

Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>manages</td>
<td>sys.zOS.MQSubsystem</td>
<td>app.messaging.mq.MQAuthInfo</td>
<td></td>
</tr>
</tbody>
</table>
app.messaging.mq.MQBufferPool

MQ Buffer Pool

Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Requirements for discovery

- None

CDM specific information

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.messaging.mq.MQReceiverChannel</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;name&gt;-&lt;MQ Queue Manager name&gt;-&lt;hostname&gt;</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID&gt;-MQReceiverChannel</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID&gt;-MQReceiverChannel</td>
</tr>
</tbody>
</table>

Naming rules

Name and federates relationship to sys.zOS.MQSubsystem (MQQueueManager).

Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>manages</td>
<td>sys.zOS.MQSubsystem</td>
<td>app.messaging.mq.MQBufferPool</td>
<td></td>
</tr>
</tbody>
</table>
**app.messaging.mq.MQPageSet**

**app.messaging.mq.MQPageSet**  
MQ PageSet

**Attributes**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSID</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Requirements for discovery**
- None

**CDM specific information**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.messaging.mq.MQReceiverChannel</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CDM Attribute</strong></td>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>Label</td>
<td>&lt;name&gt;-&lt;MQ Queue Manager name&gt;-&lt;hostname &gt;</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID&gt;-MQReceiverChannel</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID&gt;-MQReceiverChannel</td>
</tr>
</tbody>
</table>

**Naming rules**
Name and federates relationship to sys.zOS.MQSubsystem (MQQueueManager).

**Relationships**

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>manages</td>
<td>sys.zOS.MQSubsystem</td>
<td>app.messaging.mq.MQPageSet</td>
<td></td>
</tr>
</tbody>
</table>
app.messaging.mq.MQStorageClass

MQ StorageClass

Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Requirements for discovery
- None

CDM specific information

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.messaging.mq.MQReceiverChannel</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td></td>
<td>Example</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;name&gt;-&lt;MQ Queue Manager name&gt;-&lt;hostname &gt;</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID&gt;-MQReceiverChannel</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;name&gt;-&lt;MQ Queue Mgr&gt;-&lt;SMFID&gt;-MQReceiverChannel</td>
</tr>
</tbody>
</table>

Naming rules
Name and federates relationship to sys.zOS.MQSubsystem (MQQueueManager).

Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>accesses</td>
<td>sys.zOS.MQSubsystem</td>
<td>app.messaging.mq.MQStorageClass</td>
<td></td>
</tr>
<tr>
<td>advertises</td>
<td>app.messaging.mq.MQPageSet</td>
<td>app.messaging.mq.MQStorageClass</td>
<td></td>
</tr>
</tbody>
</table>
app.messaging.mq.MQStorageClass

CICS Region
sys.zOS.CICSRegion

A CICS region is an address space that runs the CICS Transaction Server program.

Attributes

<table>
<thead>
<tr>
<th>CDM Attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>JobName (sys.zOS.AddressSpace)</td>
<td>See the sys.zOS.AddressSpace class description.</td>
<td>CILB2</td>
</tr>
<tr>
<td>JobUserId (sys.zOS.AddressSpace)</td>
<td>See the sys.zOS.AddressSpace class description.</td>
<td>STCICICS</td>
</tr>
<tr>
<td>JobType (sys.zOS.AddressSpace)</td>
<td>See the sys.zOS.AddressSpace class description.</td>
<td>STC</td>
</tr>
<tr>
<td>JobFunction (sys.zOS.AddressSpace)</td>
<td>See the sys.zOS.AddressSpace class description.</td>
<td>CICS</td>
</tr>
<tr>
<td>ProcStep (sys.zOS.AddressSpace)</td>
<td>See the sys.zOS.AddressSpace class description.</td>
<td>CILB2</td>
</tr>
<tr>
<td>StepName (sys.zOS.AddressSpace)</td>
<td>See the sys.zOS.AddressSpace class description.</td>
<td>CICSTS23</td>
</tr>
<tr>
<td>Pgm (sys.zOS.AddressSpace)</td>
<td>See the sys.zOS.AddressSpace class description.</td>
<td>DFHSIP</td>
</tr>
<tr>
<td>PgmParm (sys.zOS.AddressSpace)</td>
<td>See the sys.zOS.AddressSpace class description.</td>
<td>START=INITIAL,SYSIN</td>
</tr>
<tr>
<td>NetID</td>
<td>SNA netid</td>
<td></td>
</tr>
<tr>
<td>ApplID</td>
<td>The VTAM applid used as an endpoint for SNA communications with CICS. The applid is unique within the SNA network and similar to IPName + IPPort.</td>
<td>CICSTL23</td>
</tr>
<tr>
<td>VersionString</td>
<td>The version and release of the CICS program.</td>
<td>2.5</td>
</tr>
<tr>
<td>SYSIDNT</td>
<td>CICS System Identification</td>
<td>CIC1</td>
</tr>
<tr>
<td>GRName</td>
<td>CICS Generic Resource Name</td>
<td>CICSH###</td>
</tr>
<tr>
<td>SIT</td>
<td>CICS System Initialization Table parameters.</td>
<td>SITParameter Value</td>
</tr>
<tr>
<td></td>
<td>The list is in a table format with two columns: ParameterName and ParameterValue.</td>
<td></td>
</tr>
<tr>
<td>SITOverrides</td>
<td>CICS System Initialization Table Override parameters.</td>
<td>SITParameter Value</td>
</tr>
<tr>
<td></td>
<td>The list is in a table format with two columns: ParameterName and ParameterValue.</td>
<td></td>
</tr>
<tr>
<td>TransactionsChecksum</td>
<td>Checksum of all transaction defined to the CICSRegion.</td>
<td></td>
</tr>
<tr>
<td>ProgramsChecksum</td>
<td>Checksum of all programs defined to the CICSRegion</td>
<td></td>
</tr>
<tr>
<td>FilesChecksum</td>
<td>Checksum of all programs defined to the CICSRegion</td>
<td></td>
</tr>
</tbody>
</table>
sys.zOS.CICSRegion

**Requirements for discovery**
- CICS Region book has all attributes and all relationships except AddressSpaces.
- ZOSTASK book has naming rules and relationships to AddressSpaces.
- Filtering enabled (which is the default). Note that following Attribute level filters are disabled by default since they can be large at some sites. The customer needs to consciously choose to discover these: Transactions, Programs.
- The sys.zOS.CICSRegion class is only discovered when the DLA runs in APF authorized mode.

**CDM specific information**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.CICSRegion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>&lt;JobName&gt;-&lt;SMFID&gt;</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;JobName&gt;-&lt;SMFID&gt;-CICSRegion</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;JobName&gt;-&lt;SMFID&gt;-CICSRegion</td>
</tr>
</tbody>
</table>

**Naming rules**
JobName and runsOn relationship to zOS.

**Relationships**

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>runsOn</td>
<td>sys.zOS.CICSRegion</td>
<td>sys.zOS.z/OS</td>
<td>Part of primary naming rule along with JobName attribute.</td>
</tr>
<tr>
<td>Uses</td>
<td>sys.zOS.CICSRegion</td>
<td>sys.zOS.CICSRegion</td>
<td>For cases where the remote CICS is not active on the same z/OS, a simple CICS Region is populated using the netID + ApplID name rule.</td>
</tr>
<tr>
<td>Uses</td>
<td>sys.zOS.CICSRegion</td>
<td>sys.zOS.IMSSubsystem</td>
<td></td>
</tr>
<tr>
<td>Uses</td>
<td>sys.zOS.CICSRegion</td>
<td>sys.zOS.DB2Subsystem</td>
<td></td>
</tr>
<tr>
<td>Uses</td>
<td>sys.zOS.CICSRegion</td>
<td>sys.zOS.MQSubsystem</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>sys.zOS.CICSRegion</td>
<td>net.TcpPort</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>sys.zOS.CICSRegion</td>
<td>net.UdpPort</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>sys.zOS.CICSRegion</td>
<td>net.BindAddress</td>
<td>This relationships is used by the DLA sensor to stitch relationships with distributed resources.</td>
</tr>
<tr>
<td>Contains</td>
<td>sys.zOS.CICSRegion</td>
<td>sys.zOS.CICSTransaction</td>
<td></td>
</tr>
<tr>
<td>Contains</td>
<td>sys.zOS.CICSRegion</td>
<td>sys.zOS.CICSProgram</td>
<td></td>
</tr>
<tr>
<td>Contains</td>
<td>sys.zOS.CICSRegion</td>
<td>sys.zOS.CICSFile</td>
<td></td>
</tr>
</tbody>
</table>
sys.zOS.CICSTransaction

CICS Transactions.

Attributes

<table>
<thead>
<tr>
<th>CDM Attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the transaction</td>
<td></td>
</tr>
</tbody>
</table>

Requirements for discovery

- Filtering enabled. By default this is SUPPRESSED since there may be many instances. Note that the sys.zOS.CICSRegion attribute TransactionsChecksum enables CMDB to detect Transaction changes without needing to populate many sys.zOS.CICSTransaction instances.
- IMS subsystem and related address spaces must be active and fully initialized.
- The DLA must be running in APF authorized mode.

CDM specific information

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.CICSTransaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;name&gt;&lt;cics region name&gt;</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;label&gt; CICSTransaction</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;label&gt; CICSTransaction</td>
</tr>
</tbody>
</table>

Naming rules

Name and contains relationship from CICSRegion.

Relationships

<table>
<thead>
<tr>
<th>Link Type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>contains</td>
<td>sys.zOS.CICSRegion</td>
<td>sys.zOS.CICSTransaction</td>
<td></td>
</tr>
</tbody>
</table>
sys.zOS.CICSProgram

sys.zOS.CICSProgram

CICS Programs.

Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the program</td>
<td></td>
</tr>
</tbody>
</table>

Requirements for discovery

- Filtering enabled. By default this is SUPPRESSED since there may be many instances. Note that the sys.zOS.CICSRegion attribute ProgramsChecksum enables CMDB to detect Program changes without needing to populate many sys.zOS.CICSProgram instances.
- IMS subsystem and related address spaces must be active and fully initialized.
- The DLA must be running in APF authorized mode.

CDM specific information

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.CICSProgram</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;name&gt;-&lt;cics region name&gt;</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;label&gt;- CICSProgram</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;label&gt;- CICSProgram</td>
</tr>
</tbody>
</table>

Naming rules

Name and contains relationship from sys.zOS.CICSRegion.

Relationships

<table>
<thead>
<tr>
<th>Link Type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
</table>
sys.zOS.CICSFile

CICS Files.

**Attributes**

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDName</td>
<td>DD name of the file</td>
<td></td>
</tr>
<tr>
<td>Datasets</td>
<td>List of datasets for the DD, space delimited.</td>
<td></td>
</tr>
</tbody>
</table>

**Requirements for discovery**

- Filtering enabled. By default this is SUPPRESSED since there may be many instances. Note that the sys.zOS.CICSRegion attribute FilesChecksum enables CMDB to detect File changes without needing to populate many sys.zOS.CICSFile instances.
- IMS subsystem and related address spaces must be active and fully initialized.
- The DLA must be running in APF authorized mode.

**CDM specific information**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.CICSFile</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>DDName</td>
<td>DD name of the file</td>
</tr>
<tr>
<td>Datasets</td>
<td>List of datasets for the DD, space delimited.</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;ddname&gt;-&lt;cics region name&gt;</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;label&gt;-- CICSFile</td>
</tr>
<tr>
<td>source-token (IdML)</td>
<td>&lt;label&gt;-- CICSFile</td>
</tr>
</tbody>
</table>

**Naming rules**

Name and contains relationship from sys.zOS.CICSRegion.

**Relationships**

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>contains</td>
<td>sys.zOS.CICSRegion</td>
<td>sys.zOS.CICSFile</td>
<td></td>
</tr>
</tbody>
</table>
sys.zOS.CICSFile

WAS
A WebSphere Application Server (WAS) is an address space that runs the WAS Control Process. The app.j2ee.websphere.WebSphereServer class is discovered only when the DLA runs in authorized mode.

### Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The Jobname from the corresponding AddressSpace class.</td>
<td>BBO5ACR</td>
</tr>
<tr>
<td>ProductVersion</td>
<td>Version</td>
<td>6.0.2.7</td>
</tr>
<tr>
<td>KeyName</td>
<td>AppServer</td>
<td>AppServer</td>
</tr>
</tbody>
</table>

### Requirements for discovery

- WAS book has all attributes and all relationships except AddressSpaces.
- ZOSTASK book has naming rules and relationships to AddressSpaces.
- Filtering enabled (which is the default).

### CDM specific information

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.j2ee.websphere.WebSphereServer</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;JobName&gt;-&lt;SMFID&gt;-WebSphereServer</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;JobName&gt;-&lt;SMFID&gt;-WebSphereServer</td>
</tr>
</tbody>
</table>

### Naming rules

Name, KeyName and accessedVia BindAddress

### Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>realizes</td>
<td>sys.zOS.AddressSpace</td>
<td>app.j2ee.websphere.WebSphereServer</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>app.j2ee.websphere.WebSphereServer</td>
<td>net.BindAddress</td>
<td>This relationships is used by the DLA sensor to stitch relationships with distributed resources.</td>
</tr>
<tr>
<td>configuredUsing</td>
<td>app.j2ee.websphere.WebSphereServer</td>
<td>app.ConfigFile</td>
<td>SCONFIG_DIR/cells/cell_name/nodes/node_name/servers/server_name/server.xml</td>
</tr>
<tr>
<td>configuredUsing</td>
<td>app.j2ee.websphere.WebSphereServer</td>
<td>core.LogicalContent</td>
<td>SCONFIG_DIR/cells/cell_name/nodes/node_name/servers/server_name/*.xml</td>
</tr>
</tbody>
</table>
## app.j2ee.websphere.WebSphereServer

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains</td>
<td>app.j2ee.websphere.WebSphereServer</td>
<td>app.AppDescriptors</td>
<td>For a stand-alone server $WAS_ROOT_DIR/appdescriptors/*.xml</td>
</tr>
<tr>
<td>accessedVia</td>
<td>app.j2ee.websphere.WebSphereServer</td>
<td>Wapp.j2ee.websphere.ebSphere\NamedEndPoint</td>
<td>$CONFIG_DIR/\cells/cell_name/\nodes/node_name/\serverindex.xml For example, SOAP_CONNECTOR_\ ADDRESS, \BOOTSTRAP_\ ADDRESS</td>
</tr>
</tbody>
</table>
**app.j2ee.websphere.WebSphereCell**

A WebSphere Application Server Cell.

**Attributes**

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Cell name</td>
</tr>
<tr>
<td>RootDirectory</td>
<td>Root config directory</td>
</tr>
</tbody>
</table>

**Requirements for discovery**

- Filtering enabled (which is the default).

**CDM specific information**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.j2ee.websphere.WebSphereCell</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM attribute</td>
<td>Value</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;cell name&gt;-&lt;WAS name&gt;-SMFID</td>
</tr>
<tr>
<td></td>
<td>-WebSphereCell</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;cell name&gt;-&lt;WAS name&gt;-SMFID</td>
</tr>
<tr>
<td></td>
<td>-WebSphereCell</td>
</tr>
</tbody>
</table>

**Naming rules**

Name and accessedVia BindAddress

**Relationships**

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>memberOf</td>
<td>app.j2ee.websphere.WebSphereCell</td>
<td>app.j2ee.websphere.WebSphereCell</td>
<td></td>
</tr>
<tr>
<td>memberOf</td>
<td>app.j2ee.websphere.WebSphereNode</td>
<td>app.j2ee.websphere.WebSphereCell</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>app.j2ee.websphere.WebSphereCell</td>
<td>net.BindAddress</td>
<td></td>
</tr>
<tr>
<td>configuredUsing</td>
<td>app.j2ee.websphere.WebSphereCell</td>
<td>app.ConfigFile</td>
<td>$CONFIG_DIR/cells/cell_name/cell.xml</td>
</tr>
<tr>
<td>configuredUsing</td>
<td>app.j2ee.websphere.WebSphereCell</td>
<td>core.LogicalContent</td>
<td>$CONFIG_DIR/cells/cell_name/* xml</td>
</tr>
<tr>
<td>manages</td>
<td>sys.ComputerSystem</td>
<td>app.j2ee.websphere.WebSphereCell</td>
<td></td>
</tr>
</tbody>
</table>
app.j2ee.websphere.WebSphereNode

A WebSphere Application Server Node.

Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Node name</td>
<td></td>
</tr>
<tr>
<td>RootDirectory</td>
<td>Root config directory</td>
<td></td>
</tr>
</tbody>
</table>

Requirements for discovery

- Filtering enabled (which is the default).

CDM specific information

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.j2ee.websphere.WebSphereNode</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;node name&gt;-&lt;cell name&gt;-&lt;WAS name&gt;-&lt;SMFID&gt;-WebSphereNode</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;node name&gt;-&lt;cell name&gt;-&lt;WAS name&gt;-&lt;SMFID&gt;-WebSphereNode</td>
</tr>
</tbody>
</table>

Naming rules

Name and memberOf app.j2ee.websphere.WebSphereCell

Relationships

<table>
<thead>
<tr>
<th>Link Type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>memberOf</td>
<td>app.j2ee.websphere.\WebSphereNode</td>
<td>app.j2ee.websphere.\WebSphereCell</td>
<td></td>
</tr>
<tr>
<td>configuredUsing</td>
<td>app.j2ee.websphere.\WebSphereNode</td>
<td>app.ConfigFile</td>
<td>$CONFIG_DIR/cells/cell_name/nodes/ node_name/ node.xml</td>
</tr>
<tr>
<td>configuredUsing</td>
<td>app.j2ee.websphere.\WebSphereNode</td>
<td>core.LogicalContent</td>
<td>$CONFIG_DIR/cells/cell_name/nodes/ node_name/*.*xml</td>
</tr>
<tr>
<td>manages</td>
<td>sys.ComputerSystem</td>
<td>app.j2ee.websphere.\WebSphereCell</td>
<td></td>
</tr>
</tbody>
</table>
app.j2ee.websphere.WebSphereNamedEndPoint

A WebSphere Application Server Named Endpoint.

Attributes

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>EndPoint name</td>
<td>SOAP_CONNECTOR_ADDRESS</td>
</tr>
<tr>
<td>Hostname</td>
<td>&lt;node&gt;:&lt;hostname&gt;</td>
<td></td>
</tr>
</tbody>
</table>

Requirements for discovery

- Filtering enabled (which is the default).

CDM specific information

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.j2ee.websphere.WebSphereNamedEndPoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;name&gt;-&lt;port&gt;-WebSphereNamedEndpoint</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;name&gt;-&lt;port&gt;-WebSphereNamedEndpoint</td>
</tr>
</tbody>
</table>

Naming rules

Name, hostname and accessedVia BindAddress

Relationships

<table>
<thead>
<tr>
<th>Link Type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>accessedVia</td>
<td>app.j2ee.websphere.WebSphereNamedEndPoint</td>
<td>net.BindAddress</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>app.j2ee.websphere.WebSphereServer</td>
<td>app.j2ee.websphere.WebSphereNamedEndPoint</td>
<td></td>
</tr>
</tbody>
</table>
app.ConfigFile

**app.ConfigFile**

File

**CDM specific information**

<table>
<thead>
<tr>
<th>CDM attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content</td>
<td>Contents of config file</td>
<td>&lt;![CDATA[ ... ]]&gt;</td>
</tr>
<tr>
<td>URI</td>
<td>content://hostname//filepath</td>
<td></td>
</tr>
</tbody>
</table>

**Requirements for discovery**

- Filtering enabled (which is the default).

**CDM specific information**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.ConfigFile</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;uri&gt;-ConfigFile</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;uri&gt;-ConfigFile</td>
</tr>
</tbody>
</table>

**Naming rules**

URI

**Relationships**

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
</table>
| configuredUsing   | app.j2ee.websphere.
WebSphereCell | app.ConfigFile | $CONFIG_DIR/cells/
cell_name/cell.xml |
| configuredUsing   | app.j2ee.websphere.
WebSphereNode    | app.ConfigFile | $CONFIG_DIR/cells/
cell_name/nodes/
ode_name node.xml |
| configuredUsing   | app.j2ee.websphere.
WebSphereServer  | app.ConfigFile | $CONFIG_DIR/cells/
cell_name/nodes/
node_name/servers/
server_name/server.xml |
core.LogicalContent

Configuration File

CDM specific information

<table>
<thead>
<tr>
<th>CDM Attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content</td>
<td>Contents of config file</td>
<td>&lt;![CDATA[ ...... ]]&gt;</td>
</tr>
<tr>
<td>URI</td>
<td>collation://hostname//filepath</td>
<td></td>
</tr>
</tbody>
</table>

Requirements for discovery

- Filtering enabled (which is the default).

CDM specific information

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>core.ConfigFile</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CDM Attribute</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>id (IdML)</td>
<td>&lt;uri&gt;-LogicalContent</td>
<td></td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;uri&gt;-LogicalContent</td>
<td></td>
</tr>
</tbody>
</table>

Naming rules

URI

Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>configuredUsing</td>
<td>app.j2ee.websphere.WebSphereCell</td>
<td>core.LogicalContent</td>
<td>$CONFIG_DIR/cells/cell_name/*.xml</td>
</tr>
<tr>
<td>configuredUsing</td>
<td>app.j2ee.websphere.WebSphereNode</td>
<td>core.LogicalContent</td>
<td>$CONFIG_DIR/cells/cell_name/nodes/node_name/*.xml</td>
</tr>
<tr>
<td>configuredUsing</td>
<td>app.j2ee.websphere.WebSphereServer</td>
<td>core.LogicalContent</td>
<td>$CONFIG_DIR/cells/cell_name/nodes/servers/server_name/*.xml</td>
</tr>
<tr>
<td>contains</td>
<td>app.AppDescriptor</td>
<td>core.LogicalContent</td>
<td>WebSphereServer AppDescriptor $WAS_ROOT_DIR/appdescriptors/*.xml</td>
</tr>
</tbody>
</table>
app.AppDescriptor

Configuration File

Attributes
None

Requirements for discovery
- Filtering enabled (which is the default).

CDM specific information

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.AppDescriptor</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Value</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;uri&gt;-AppDescriptor</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;uri&gt;-AppDescriptor</td>
</tr>
</tbody>
</table>

Naming rules
Contains relationship to LogicalContent

 Relationships

<table>
<thead>
<tr>
<th>Link type</th>
<th>Source</th>
<th>Target</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>contains</td>
<td>app.j2ee.websphere.WebSphereServer</td>
<td>app.AppDescriptor</td>
<td>$WAS_ROOT_DIR/appdescriptors/*.xml</td>
</tr>
<tr>
<td>contains</td>
<td>app.AppDescriptor</td>
<td>core.LogicalContent</td>
<td>WebSphereServer AppDescriptor $WAS_ROOT_DIR/appdescriptors/*.xml</td>
</tr>
</tbody>
</table>
Appendix D. Samples provided in hlq.SIZDSAMP

The following samples are provided in hlq.SIZDSAMP:

**Member in SIZDSAMP**

**Description**

**IZDJRUN**

Batch job JCL to run the DLA.
- This invokes the IZDPDISC proc with an override on the output data set name since batch jobs don't allow system symbols.
- This enables users to perform discovery without issuing a console command.

**IZDPDISC**

JCL PROC to run the DLA.
- The PROC may be invoked as a started task or a batch job.
- This procedure may be copied to a JES PROCLIB or invoked via a JCLLIB control statement in the JCL.
- The default output datasets use the &SYSNAME system symbol to ensure a unique name within a Sysplex.
- Having a started task enables discovery to be initiated via a console command, for example, include S IZDPDISC, as part of the normal post IPL start up process to pick up any system configuration changes.

**IZDCDEF**

Default DLA configuration parameter deck in IZDPDISC.
This configuration parameter deck is coded to show and cause discovery of the default resources. This includes ZOS, ZSeriesComputerSystem and Sysplex, as well as application Subsystems (such as DB2SubSystems and MQSubSystems). By default, discovery is limited to 100 resources per CICS, IMS or DB2 plus WebSphere and MQ discovery. The output from default discovery is placed in several books including ZOSBASE (basic z/OS and hardware information), ZOSTASK (z/OS address space related data), and individual books (one for each of the CICS, IMS, DB2, WebSphere and MQ subsystem that are discovered) that are named using the jobname or STC name of the discovered subsystem.

**IZDJFTP**

Batch job JCL to run the FTP only.
- This invokes the IZDPFTP proc with an override on the output data set name since batch jobs don't allow system symbols.
- This enables users to perform FTP without running discovery.

**IZDPFTP**

JCL PROC to FTP the IdML members to the Discovery Library File Store.
- The PROC may be invoked as a started task or a batch job.
- This procedure may be copied to a JES PROCLIB or invoked via a JCLLIB control statement in the JCL.
- IZDJDISC invokes this proc.
IZDCFTP1  
Initial FTP statements used to PUT members to the Discovery Library File Store. See later section for details.

IZDCBASE  
DLA configuration parameter deck to discover ZOSBASE only.

IZDCTASK  
DLA configuration parameter deck to discover ZOSTASK only.

IZDCIMS  
IZDCIMS IMS default parameter member  
The default member for IMS discovery IZDCIMS is coded to cause IMS only discovery. Additional non-IMS information is only discovered where it relates to IMS or is necessary for naming rules or IMS relationships.  
The output for IMS discovery is placed in several books each book named using the jobname or STC name of the discovered IMS control region.

IZDCDB2  
DLA configuration parameter deck to discover DB2 subsystems, including all attributes. Additional non-DB2 information is only discovered where it related to DB2 or is necessary for naming rules or DB2 relationships.  
The output for DB2 discovery is placed in several books each book named using the jobname or STC name of the discovered DB2 master region.

IZDCICS  
DLA configuration parameter deck to discover CICS regions, including all attributes. Additional non-CICS information is only discovered where it relates to CICS or is necessary for naming rules or CICS relationships.  
The output from CICS discovery is placed in several books each book named using the jobname or STC name of the discovered CICS region.

IZDCMQ  
DLA configuration parameter deck to discover MQ subsystems, including all attributes. Additional non-MQ information is only discovered where it relates to MQ or is necessary for naming rules or MQ relationships.  
The output for MQ discovery is placed in several books each book named using the jobname or STC name of the discovered MQ master address space.

IZDCWAS  
DLA configuration parameter deck to discover WebSphere Application Servers, including all attributes. Additional non-WebSphere information is only discovered where it relates to WebSphere or is necessary for naming rules or WebSphere relationships.  
The output for WebSphere discovery is placed in several books each book named using the jobname or STC name of the discovered WebSphere region.
The IBM Terminology website consolidates the terminology from IBM product libraries in one convenient location. You can access the Terminology Web site at http://www.ibm.com/software/globalization/terminology.

The abbreviations that are used in this book are defined in the following list:

**CCMDB.** IBM Tivoli Configuration and Change Management Database

**CDM.** Common Data Model

**CEC.** Central Electronics Complex

**CIM.** Common Information Model

**Coupling Facility.** A special logical partition that provides high-speed caching, list processing, and locking functions in a sysplex.

**CPC.** Central Processing Complex.

**IDTF.** Interactive Data Transmission Facility, a TSO/E utility for transmitting and receiving files.

**IPL.** Initial Program Load (boot-up)

**LPAR.** Logical Partition

**MSU.** Millions of Service Units

**RACF.** Resource Access Control Facility.

**SYSPLEX.** A set of z/OS systems communicating and cooperating with each other through certain multisystem hardware components and software services to process customer workloads.

**SMP/E.** System Modification Program/Extended. A tool for installing and maintaining software for z/OS.

**TADDM.** IBM Tivoli Application Dependency Discovery Manager

**TBSM.** IBM Tivoli Business Service Manager.

**TSO/E.** Time Sharing Option/Extensions. The primary user interface to z/OS.

**XCF.** Cross-System Coupling Facility
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:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to 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 might not apply to you.

This information 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.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.
IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM’s future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

**Trademarks**

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (© or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at [http://www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.
# Index

## A
- adding
  - custom server resource to business application 34
- custom servers 34
- APF authorization 60
- app.AppDescriptor 170
- app.ConfigFile 168
- app.db.db2.Db2Database 119
- app.db.db2.Db2TableSpace 120
- app.j2ee.websphere.WebSphereCell 165
- app.j2ee.websphere.WebSphereNamedEndPoint 167
- app.j2ee.websphere.WebSphereNode 166
- app.messaging.mq.MQAliasQueue 148
- app.messaging.mq.MQAuthInfo 152
- app.messaging.mq.MQBufferPool 153
- app.messaging.mq.MQClientConnectionChannel 142
- app.messaging.mq.MQClusterReceiverChannel 138, 140
- app.messaging.mq.MQLocalQueue 146
- app.messaging.mq.MQModelQueue 149
- app.messaging.mq.MQNameList 150
- app.messaging.mq.MQPageSet 154
- app.messaging.mq.MQProcess 151
- app.messaging.mq.MQRemoteQueue 136
- app.messaging.mq.MQSenderChannel 134
- app.messaging.mq.MQServerConnectionChannel 134
- app.messaging.mq.MQStorageClass 155

## B
- BOOK directive 55
- books
  - output 13
- business applications
  - adding custom server resources 34

## C
- CDM
  - z/OS DLA data model class types represented in 81
- checksums
  - to detect members needing transfer 29
- CICS region
  - discovery 22
- CICS software
  - supported by DLA 9
  - CICSRegion 157
  - CICSTransaction 159
- codepage 62
- codes 78
- core.LogicalContent 169
- creating custom servers 33
- CRITERIA directive 52
- custom servers
  - adding 34

## D
- database
  - DB2
  - customization 61
  - DB2 software
  - supported by DLA 9
  - DB2 subsystem
discovery 17
- DB2Subsystem 115
- DEBUG directive 56
- dev.StorageVolume 113
- DISC directive 39
discovery
  - automating 30
discovery customization 37
- Discovery Library
  - file naming standards 28
  - Discovery Library File Store
  - ftp transfer to DLA running 11
  - DLFS FTP server 28
documents related 10

## F
- file naming standards
  - Discovery Library 28
- FILES directive 53
- FILT directive 43
  - filtering directives
    - BOOK 55
    - DEBUG 56
    - DISC 39
    - FILES 53
    - FILT 43
    - SET 38
  - Fqdn 86

## G
- glossary 173

## H
- hardware
  - supported by DLA 10

## I
- icons
  - custom server 35
- IdML XML browser 30
- IMS software
  - supported by DLA 9
- IMS subsystem
  - discovery 19
  - OVERVIEW 59
- installation
  - checklist 63
  - installatuiB
  - SMP/E 59
- introduction 1
  - IpInterface 85
  - IPv4Address 84
- IZDJFTP JCL job 13
- IZDJRUN batch job 12
- IZDPDISC procedure 12
- IZDPFTP procedure 13
- IZDRDLA
  - return codes 78

## J
- JCL
  - customization 62

## M
- ManagementSoftwareSystem 82
  - managing custom servers 33
  - members
    - checksummed to detect need for transfer 29
    - messages 65
    - MQ software
      - supported by DLA 9
    - MQ subsystem
      - discovery 21
  - MQSubsystem 132

## N
- net.BindAddress 111
- net.TcpPort 109
- net.UdpPort 110

## O
- Organization 83
  - output books 13
  - overview
    - of installation and customization 59

© Copyright IBM Corp. 2008, 2010

hlq.SIZDSAMP
  - samples provided in 171

icons
  - custom server 35
IdML XML browser 30
IMS software
  - supported by DLA 9
IMS subsystem
  - discovery 19
IMSProgram 128
installation 59
overview 59
installation checklist 63
installatuiB
  - SMP/E 59
introduction 1
IpInterface 85
IPv4Address 84
IZDJFTP JCL job 13
IZDJRUN batch job 12
IZDPDISC procedure 12
IZDPFTP procedure 13
IZDRDLA
  - return codes 78

JCL
  - customization 62

ManagementSoftwareSystem 82
  - managing custom servers 33
  - members
    - checksummed to detect need for transfer 29
    - messages 65
    - MQ software
      - supported by DLA 9
    - MQ subsystem
      - discovery 21
  - MQSubsystem 132

Organization 83
  - output books 13
  - overview
    - of installation and customization 59
overview (continued)

product 1

R
reference 33
reference information 33
related documents 10
return codes
from IZDRDLA 78
other 79

S
SAF
authorization 60
samples
provided in hlq.SIZDSAMP 171
servers
adding 34
adding resource to business application 34
associating an icon 35
creating 33
custom 33
managing 33
unknown patterns 33
SET directive 38
SMP/E
installation 59
target datasets 59
software
supported by DLA 9
storage.StorageSubSystem 112
supported hardware 10
supported software 9
sys.DR2DataSharingGroup 117
sys.zOS.AddressSpace 107
sys.zOS.CICSFile 161
sys.zOS.CICSProgram 160
sys.zOS.IMSDatabase 129
sys.zOS.IMSSubsystem 122
sys.zOS.IMSSysplexGroup 125
sys.zOS.IMSTransaction 127
sys.zOS.LPAR 93
sys.zOS.ZVMGuest 97
SystemSpecificCollection 89
sysZReportFile 101

T
TADDM
loading with XML files 31
target DLFS FTP server 28
tips and tools 30

W
WebSphere
customization 63
WebSphere Application Server
discovery 24

WebSphere for MQ
customization 60
WebSphere software
supported by DLA 9
WebSphereServer 163
Working with the DLA for z/OS 11

X
XML files
loading into TADDM 31

Z
z/OS 98
z/OS DLA data model class types represented in CDM 81
z/OS DLA Discovery
customization 37
z/OS software
supported by DLA 9
ZOSALL book 27
ZOSBASE
discovery 15
ZOSTASK
discovery 16
ZSeriesComputerSystem 91
Readers’ Comments — We'd Like to Hear from You

Discovery Library Adapter for z/OS
User's Guide and Reference
Version 3 Release 1

Publication No. SC27-2893-00

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.
Submit your comments using one of these channels:
• Send your comments to the address on the reverse side of this form.
• Send a fax to the following number: 800-426-7773 (from within the U.S.) or 408-463-2629 (from outside the U.S.)

If you would like a response from IBM, please fill in the following information:

Name

Address

Company or Organization

Phone No.

E-mail address