Note!
Before using this information and the product it supports, be sure to read the general information under "Notices" on page 119.
Contents

Figures ............................................. v

Chapter 1. Introduction ........................ 1
Product overview .................................. 1
Supported software ............................... 1
Supported hardware ............................... 2
Related documents ............................... 2

Chapter 2. Installation and customization ......... 3
Overview .......................................... 3
SMP/E install ...................................... 3
APF Authorization .................................. 4
RACF authorization ................................ 4
DB/2 customization ............................... 4
JCL customization .................................. 5
Codepage .......................................... 5

Chapter 3. User guide .............................. 7
Running the z/OS DLA ............................. 7
IZDJRUN JCL batch job ............................ 7
IZDPDISC JCL proc ................................ 8
IZDPFTP JCL proc .................................. 9
IZDJKFTP JCL batch job ............................ 9
Output books ...................................... 9
ZOSBASE discovery ............................... 11
ZOSTASK discovery ............................... 12
DB2 subsystem discovery ........................ 12
IMS subsystem discovery ........................ 14
MQ subsystem discovery ........................ 15
CICS region discovery ............................ 16
WebSphere application server discovery 17
ZOSALL book ..................................... 18
FTP transfer to the Discovery Library File Store .. 19
Target DLFS FTP server ........................... 19
Discovery Library file naming standards ........ 20
Checks to determine if members need to be transferred . 21
Automating discovery ............................ 21
IdML XML Viewer .................................. 21
z/OS DLA FTP utility .............................. 22
Loading the XML files into Tivoli CCMDB ...... 23

Chapter 4. User reference ......................... 25
z/OS DLA filtering ................................ 25
SET directive ...................................... 25
DISC directive .................................... 31
FILT directive ..................................... 33
BOOK directive ................................... 35

Appendix A. Messages and codes ................ 37
Messages .......................................... 37
Codes ............................................. 48

Appendix B. z/OS DLA data model class types represented in CDM ..................... 49
ManagementSoftwareSystem .................... 50
Organization ...................................... 51
ZSeriesComputerSystem ......................... 52
LPAR ............................................... 54
ZVM .............................................. 56
ZVMGuest ........................................ 57
ComputerSystem .................................. 58
IpInterface ....................................... 59
IPv4Address ...................................... 60
Fqdn .............................................. 61
z/OS .............................................. 62
Sysplex ........................................... 69
SystemSpecificCollection ....................... 70
AddressSpace ..................................... 71
TcpPort .......................................... 74
UdpPort .......................................... 75
BindAddress ...................................... 76
StorageSubsystem ............................... 77
StorageVolume .................................... 78
DB2Subsystem .................................... 79
DB2DataSharingGroup ............................ 81
Db2Database ..................................... 83
Db2TableSpace ................................... 84
IMSSubsystem .................................... 85
IMSSysplexGroup ................................ 88
IMSTransaction .................................. 90
IMSprogram ...................................... 91
IMSDatabase ..................................... 92
MQSubsystem .................................... 94
MQSenderChannel ................................ 96
MQReceiverChannel ............................... 97
CICSRegion ....................................... 98
CICSTransaction ................................ 101
CICSPGM ......................................... 102
CICSFile ......................................... 103
WebSphereServer ................................ 104
WebSphereCell ................................... 106
WebSphereNode .................................. 107
WebSphereNamedEndPoint ....................... 108
ConfigFile ....................................... 109
LogicalContent .................................. 110
AppDescriptor .................................... 111

Appendix C. Samples provided in hlq.SIZDSAMP .................. 113

Appendix D. Installation checklist ................ 115

Glossary ........................................... 117
### Figures

1. ZOSBASE—zSeries hardware and z/OS details 11  
2. ZOSTASK—AddressSpace details and relationships 12  
3. DB2 Subsystem details and relationships—No Data Sharing 13  
4. DB2 Subsystem details and relationships—Data Sharing 14  
5. IMS Subsystem details and relationships 15  
6. MQ Subsystem details and relationships 16  
7. CICS region details and relationships 17  
8. WebSphere Application Server details and relationships 18  
9. ZOSALL—Everything that is active in one book 19  
10. The Discovery Library IdML XML File Browser 22  
11. The z/OS DLA FTP utility 23
Chapter 1. Introduction

The IBM Tivoli Discovery Library Adapter for z/OS® (DLA) discovers z/OS resources and generates output XML files. The files, often referred to as Books, conform to the Discovery Library IdML XML schema and Common Data Model (CDM). The Books can be loaded by various products, including Tivoli® Change and Configuration Management Database (CCMDB) and Tivoli Business Service Management (TBSM) products.

Product overview

The z/OS DLA is implemented as a combination of z/OS load modules and REXX routines that can be executed as a batch job or started task on the z/OS system to perform the discovery. The modules typically use system service macros, various memory control blocks, APIs including DB/2 IFI, DSNREXX SQL, and product utilities including netstat and the MQ command interface (API) to identify those resources and relationships that are active at the time discovery is executed.

Discovery coverage includes:

- z/OS information, for example, parmlib active member contents, LNKLST, IODF dataset, system exits, APF list and IPL information.
- zSeries® machine information, for example, serial number, processing capacity, LPAR, VM guest.
- IMS™ information, for example, transactions, programs, and data bases.
- CICS® information, for example, transactions, programs, files, and System Initialization Table (SIT) details.
- DB/2 for z/OS information, for example, databases and tables spaces.
- MQSeries® for z/OS information, for example, ports and connections.
- WebSphere® Application Server for z/OS information, for example, Cell, Node, and configuration files.
- Address space information, for example, DD allocations, address space type, program name, and program parameters.
- DASD volume information.

Supported software

The DLA supports discovery for all releases of z/OS, CICS, IMS, MQ, DB/2 and WebSphere that are currently in service. Refer to http://www.ibm.com/software/support/lifecycle/index_a_z.html for a list of releases that are currently in service.

This includes:

- z/OS v1.6 and above.
- IMS for z/OS v8.1 and above.
- CICS Transaction Server v2.2 and above.
- DB2® for z/OS v8.1 and above.
- WebSphere MQ for z/OS v5.3.1 and above.
- WebSphere Application Server for z/OS v5.0 and above.
Supported hardware

Supported hardware

- Any hardware environment that supports the “supported software”.

Related documents

- z/OS V1R7.0 MVS JCL Reference, SA22-7597-10.
Chapter 2. 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 install of the z/OS DLA product libraries. Details are provided in the z/OS DLA Program Directory.
- 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 userid with an OMVS segment to run the DLA.
- Providing READ access to RACF® protected resources as required (including system PARMLIB data sets, MQCMDs class resources, WebSphere Configuration files).
- Binding the DSNREXX DB/2 interface if required. This procedure is described in the DB2 UDB for z/OS Installation Guide.
- Granting READ access to DB/2 controlled resources as required including DISPLAY commands (DISPLAY GROUP and DISPLAY DB), and SYSIBM resources (SYSIBM.SYSDATABASE, SYSIBM.SYSTABLES, SYSIBM.SYSTABLESPACE, SYSIBM.SYSINDEXES and SYSIBM.SYSCOLUMNNS).
- Granting READ access to MQ controlled resources in the MQCMDs class for DISPLAY commands (QMGR, DQM (V5.3.1) or CHINIT (V6.0), and CHANNEL).
- Scheduling the DLA batch job to execute.

Please refer to Appendix D, “Installation checklist,” on page 115 for a checklist of installation tasks.

SMP/E install

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 datasets

<table>
<thead>
<tr>
<th>Dataset</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>z/OS DLA load modules. This dataset 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>z/OS DLA REXX code</td>
</tr>
</tbody>
</table>
SMP/E target datasets

<table>
<thead>
<tr>
<th>Dataset 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>SIZDSAMP</td>
<td>PO</td>
<td>80</td>
<td>FB</td>
<td>10</td>
<td>10</td>
<td>z/OS DLA samples. Customer needs to customize and copy JCL PROCs and to JES PROCLIB dataset.</td>
</tr>
<tr>
<td>SIZDMESG</td>
<td>PO</td>
<td>80</td>
<td>FB</td>
<td>1</td>
<td>1</td>
<td>z/OS DLA message templates. 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>z/OS DLA mapping data</td>
</tr>
<tr>
<td>SIZDINST</td>
<td>PO</td>
<td>80</td>
<td>FB</td>
<td>1</td>
<td>1</td>
<td>z/OS DLA installation jobs</td>
</tr>
</tbody>
</table>

The samples provided in hlq.SIZDSAMP are listed in Appendix C, “Samples provided in hlq.SIZDSAMP,” on page 113.

APF Authorization

The z/OS DLA load module dataset must be APF authorized. Add hlq.SIZDLOAD to the APF list, PROGxx APF member or issue the SETPROG APF, ADD,DSN=hlq.SIZDLOAD,SMS command.

Note that the SETPROG command is good for initial discovery, but it will be regressed in the next IPL.

RACF authorization

The userid that runs the z/OS DLA job or STC must have the following RACF authorities:

- READ access is recommended for all the data sets in the system PARMLIB concatenation used during IPL.
- READ access is required to profiles in the MQCMDS class (if active) to allow the z/OS DLA to issue MQ DISPLAY commands via the MQ command interface.
- If RACF is used to protect DB/2 resources then authority is required to issue DB/2 DISPLAY commands and to access (READ) SYSIBM resources using dynamic SQL.
- READ access is recommended to WebSphere configuration files.
- An OMVS segment with authority to issue the netstat, host and home commands.

DB/2 customization

The follow customization is required for all DB/2 systems where discovery is to be run:

- z/OS DLA uses the DSNREXX DB/2 REXX interface to issue SQL commands from REXX. The DSNREXX interface must be bound to each DB/2 subsystem where discovery is required. Refer to DB/2 Installation, Step 18: Bind the packages for DB2 REXX Language Support: DSNTIJRX for details. (Optional—only required if DB/2 SQL discovery is required.)
- The z/OS DLA userid must have authority to issue DB/2 DISPLAY commands including DISPLAY DB and DISPLAY GROUP.
The z/OS DLA userid must have authority to READ SYSIBM.tables.  
(Optional—only required if DB/2 SQL discovery is required.)

If multiple versions of DB/2 are installed on the operating system then a  
STEPLIB (or LINKLST entry) for SDSNLOAD matching the DB/2 version of the  
discovery target must be supplied.

**JCL customization**

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.

PROC member IZDPDISC must be copied to a JES PROCLIB if discovery is to be  
rut as a started task or if the JCLLIB JCL statement is omitted from the IZDJRUN  
JCL.

**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 EMEA customers need to 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 <!--  
.... --> format and some attributes contain XML CDATA in <![CDATA[ .... ]]>  
format. The !, ] and ] characters are different in codepage 1047 and 1148. In turn  
the downloaded files would not be valid XML if the !, ] and ] were not translated  
correctly.
Chapter 3. User guide

The IBM Tivoli Discovery Library Adapter for z/OS (DLA) discovers z/OS resources and generates output XML files. The files, often referred to as Books, conform to the Discovery Library IdML XML schema and Common Data Model (CDM). The Books can be loaded by various products, including Tivoli Change and Configuration Management Database (CCMDB) and Tivoli Business Service Management (TBSM) products.

Discovery is performed by running the z/OS DLA batch job or started task on each system. The output books are saved as members of a partitioned dataset. The DLA also generates an FTP statement deck used to transfer the XML books to the Discovery Library File Store (DLFS). From the DLFS, the files can be loaded into CCMDB or TBSM.

This section explains:
- How to run the z/OS DLA
- What is in the output Books
- FTPing the output XML files to the Discovery Library File Store
- Discovery Library IdML file viewer
- Loading the XML files into Tivoli CCMDB
- Automating discovery

Running the z/OS DLA

The z/OS DLA can be run as a batch job or a short running started task. The following sample JCL members are provided in the SIZDSAMP dataset:

IZDJRUN
JCL batch job to run the z/OS DLA

IZDPDISC
JCL PROC to run z/OS DLA discovery and optionally FTP XML files

IZDPFTP
JCL PROC to FTP XML files to the Discovery Library File Store

IZDJFTP
JCL batch job to FTP XML files to the Discovery Library File Store

IZDJRUN JCL batch job

IZDJRUN is a sample JCL batch job that invokes JCL PROC IZDPDISC to run a discovery and optionally transfer XML files to the Discovery Library File Store. Refer to the IZDPDISC section for more details.

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

//RUN EXEC  PROC=IZDPDISC,
//  IDMLDSN=&SYSUID..ZOSDLA.IDML,
//  PARMMEM=IZDCDEF,
//  IZDHLQ=IZD.V2R1M0,
//  FTPMEM=IZDCFTPI,
//  FTP=Y
IZDJRUN JCL batch job

- IDMLDSN = the PDSE dataset for output XML.
  This dataset is allocated if it doesn’t already exist. It is recommended that a
different output PDSE is used per z/OS that you perform discovery on, in order
to leverage the FTP checksum processing. See FTP section for more details.
- PARMMEM = the z/OS DLA configuration deck.
  The z/OS DLA ships several sample configuration decks for convenience. Refer to
Appendix C, “Samples provided in hlq.SIZDSAMP,” on page 113. The default
IZDCDEF causes the DLA to discover all z/OS and active subsystem details,
filtering out potential high volume instances.
Alternatively, a configuration deck may be specified in stream within batch jobs
by using //ZOSDLA.IZDPARMS DD *.
- IZDHLQ = the dataset high level qualifier where the z/OS DLA run time
datasets have been installed.
  This parameter is not needed if the IZDPDISC proc has been customized to
specify the correct default parameter.
- FTP = Y or N. FTP the XML files to the Discovery Library File Store.
  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 IZDPDISC JCL PROC
is invoked to perform the transfer. Refer to the FTP section for more details.
- FTPMEM = the FTP initial statement deck.
  This parameter is only applicable if FTP=Y is specified. Refer to the FTP section
for more details.

IZDPDISC JCL proc

The IZDPDISC JCL PROC has 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 PROC is invoked by sample IZDJRUN. It can also be invoked via an operator
command or an automation product by issuing the S IZDPDISC command.

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

When this PROC is run as a started task, when discovery has completed, the
started task also completes. In other words, this is a transient started task.

The following is an extract of the IZDPDISC sample procedure. Please refer to the
SIZDSAMP dataset for the complete version.
//IZDPDISC PROC FTP=Y, Y|N FTP OUTPUT TO DLFS
// IDMLDSN=IZD.IOML.&SYSNAME, OUTPUT IDML PDSE DATASET
// PARMMEM=IZDCDEF, INPUT PARAMETER MEMBER
// FTPMEM=IZDCFTPI, FTP INPUT PARAMETERS
// IZDHLQ=IZD.V2RIMO Z/OS DLA DATASET HLQ
//*
//******************************
//* ALLOCATE OUTPUT IDML PDSE
//******************************
//ALLOC EXEC PGM=JEFBR14
//IZDIDLML DD DISP=(MOD,CATLG),DSN=&IDMLDSN,
// DSNTYPE=LIBRARY,
// DCB=(LRECL=5000,BLKSIZE=10000,DSORG=PO,RECFM=VB),
// SPACE=(CYL,(10,10,50)),UNIT=SYSDA
//******************************
IZDPDISC JCL proc

IZDPFTP JCL proc

IZDPFTP JCL PROC performs an FTP transfer of the XML files to the Discovery Library File Store.

This PROC is invoked by sample IZDPDISC and IZDJFTP. It can also be invoked by an operator command or an automation product by issuing the S IZDPFTP command.

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


IZDJFTP JCL batch job

IZDJFTP is a JCL batch job that invokes IZDPFTP to transfer XML files to the Discovery Library File Store.

By default IZDJRUN and IZDPDISC has FTP=Y specified, which causes all output XML files that have changed since a previous discovery to be transferred to the Discovery Library File Store. This is done using the @FTPCHGP member parameter that the DLA generates. The IZDJFTP batch job can also be used to FTP all XML files, using the @FTPALLP member that the DLA generates.

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

//FTP EXEC PROC=IZDPFTP,IDLDSN=&IDMLDSN,FTPMEM=FTPCHGP,
// COND=(0,NE)

Output books

The z/OS DLA supports two types of Discovery Library books:
Output books

- Create. This is the default book type generated by the z/OS DLA. These books contain a snapshot discovery and do not necessarily contain everything known about the environment. For example, if a subsystem was not active at the time of discovery it would not be discovered.

- Refresh. This mode can be enabled via a DLA configuration option and indicates the book contains everything known about the environment. The reader of the book, for example, CC MDB and TBSM, should consider anything that was loaded previously from a similar book (same MSSName—member@hostname), and is not in this book, to be obsolete. This data should be removed. This mode is typically used after there has been a major configuration change.

The DLA can be run to produce a single output file called ZOSALL. This option is not recommended for products such as CC MDB and TBSM that support Refresh books, since subsystems that are inactive at the time of the discovery are not included in the ZOSALL book and in turn are considered obsolete and deleted from CC MDB and TBSM.

Most customers have different product support specialist teams and it is envisaged that these teams will be responsible for managing discovery within their area, since they are the people who know when it is appropriate to perform discovery. In order to support different teams performing discovery, including Refresh mode, by default the DLA generates multiple books as members of a PDSE.

Depending on what configuration options are specified and what subsystems are active at the time of the discovery, the following members are generated:

- A member called “ZOSBASE” containing hardware and z/OS details
- A member called “ZOSTASK” containing address space details
- A member per DB2 Subsystem
- A member per IMS Subsystem
- A member per MQ Subsystem
- A member per CICS Region
- A member per WebSphere Application Server

The z/OS DLA ships several sample configuration members that can be used to specify what is discovered:

IZDCDEF
Default DLA configuration parameter deck in IZDPDISC JCL PROC.

This configuration parameter deck discovers ZOSBASE, ZOSTASK and subsystem books (CICS, IMS, DB/2, MQ and WebSphere), suppressing some potentially high volume attributes, for example, transactions, programs, and databases.

IZDCBASE
DLA configuration parameter deck to discover ZOSBASE only.

IZDCTASK
DLA configuration parameter deck to discover ZOSTASK only.

IZDCIMS
DLA configuration parameter deck to discover IMS subsystems only.

IZDCDB2
DLA configuration parameter deck to discover DB2 subsystems only.
IZDCCICS
DLA configuration parameter deck to discover CICS regions only.

IZDCWAS
DLA configuration parameter deck to discover WebSphere Application Servers only.

IZDCMQ
DLA configuration parameter deck to discover MQ subsystems 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.

Having separate books enables checksum comparison with previous discoveries, in order to avoid transfers of 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).

![Diagram](image)

**Attributes include:**
- Parmlib active member contents
- IODF dataset, time, editt
- LNKLST, APF datasets

**Address Space**

**Refer to** Appendix B, “z/OS DLA data model class types represented in CDM,” on page 49 for details on what attributes and relationships are discovered for these classes.

**Notes:**
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.

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

![ZOSTASK diagram]

Refer to Appendix B, “z/OS DLA data model class types represented in CDM,” on page 49 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 Websphere 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 books contain DB2 subsystem details and relationships. In order to avoid a potentially high volume of instances being populated...
unnecessarily 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 via a DB2 Data Sharing Group. The z/OS DLA automatically discovers both types of DB2 environments.

The following diagrams show the classes and relationships that are populated in the DB2 books. (The boxes that are clear are not populated).

Figure 3. DB2 Subsystem details and relationships—No Data Sharing
DB2 subsystem discovery

Refer to Appendix B, “z/OS DLA data model class types represented in CDM,” on page 49 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. In order for the DLA to query the Maximum Altered Timestamps, the userid running the DLA needs to have the read authority to access the DB2 catalog tables. For more details, refer to the DB2 Customization section.
3. When multiple versions of DB/2 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 DB/2 discovery needs to be run separately for each version of DB/2.

IMS subsystem discovery

The IMS subsystem books contain IMS subsystem details and relationships. In order 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

![Diagram of DB2 and IMS subsystems with attributes and relationships]
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).

Refer to Appendix B, “z/OS DLA data model class types represented in CDM,” on page 49 for details on what attributes and relationships are discovered for these classes.

Notes:
• 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 books contain MQ subsystem details and relationships.

The following diagram shows the classes and relationships that are populated in the MQ books. (The boxes that are clear are not populated).
MQ subsystem discovery

Refer to Appendix B, “z/OS DLA data model class types represented in CDM,” on page 49 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 books contain CICS Region 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).

![Diagram showing classes and relationships in CICS books]

**Attributes include:**
- Version
- TransactionsChecksum
- ProgramsChecksum
- DatabasesChecksum
- NetID, Applid
- SYSIDNT, GRNAME
- SIT, SITOverides

Refer to Appendix B, “z/OS DLA data model class types represented in CDM,” on page 49 for details on what attributes and relationships are discovered for these classes.

**Notes:**
- 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.
- CICS Discovery works independently of address space filters when discovering started class and batch jobs.

**WebSphere application server discovery**

The WebSphere Application Server (WAS) book contains WAS details and relationships.

The following diagram shows the classes and relationships that are populated in the WAS books. (The boxes that are clear are not populated).
WebSphere application server discovery

Figure 8. WebSphere Application Server details and relationships

Refer to Appendix B, “z/OS DLA data model class types represented in CDM,” on page 49 for details on what attributes and relationships are discovered for these classes.

Notes:
• In order to include the contents of the WAS configuration files, the userid running the DLA needs to have read access to the files.
• WAS Discovery works independently of address space filters when discovering started class and batch jobs.
• The WebSphere Named Endpoint discovery includes the SOAP port the WAS API is configured for. TADDM WAS Sensor uses this information to perform lower level WAS discovery.

ZOSALL book

This ZOSALL book contains everything in a single books. Refer to the Reference section on how to enable this type of book. The ZOSALL book is not recommended to be loaded into CCMDB, to avoid possible Refresh conflicts as explained in the Output Books section.

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

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 IZDCFTP1 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 dataset for the complete version.
Discovery Library file naming standards

* 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.
* @FTPCHGP 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:
  * <difs>
  * <userid> <password>
  * cd <directory>
  * type a
* Where:
  * <difs> = 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 "<difs>" 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.

***********************************************************************
phtsds3.perthapc.au.ibm.com
userid password
cd /u/userdirectory
type a

Discovery Library file naming standards

All files in the DLFS should comply with the Discovery Library file name standards: <appcode>.<hostname>.<timestamp>.<refresh>.xml

Where:
  * <AppCode>= the DLA Application Code
    – ZOSDISC210ALL for ZOSALL books
    – ZOSDISC210BASE for ZOSBASE books
    – ZOSDISC210TASK for ZOSTASK books
    – ZOSDISC210DB2 for DB2 Subsystem books
    – ZOSDISC210IMS for IMS Subsystem books
    – ZOSDISC210MQ for MQ Subsystem books
    – ZOSDISC210CICS for CICS Region Subsystem books
    – ZOSDISC210WAS 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>
Discovery Library file naming standards

- `<timestamp>`= the timestamp specified in the IdML create or IdML refresh clauses.
- `<.refresh>`= `.refresh` if the IdML has a refresh clause.

Discovery Library standards also say that files being transferred to the DLFS should have a suffix of `.partial`. When transfer is complete the file is renamed to remove the suffix. This file name convention is not usable on z/OS since there is a limit of 44 chars for a dataset name and a restriction on symbols used. So the z/OS DLA generates FTP statements to transfer the IdML members and rename them to the appropriate Discovery Library format, for example:

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

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

Checksums to detect if members need to 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

Repeated discoveries per z/OS are required for CCMDB to detect changes. It is recommend that the repeated discoveries are automated via 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 should be rediscovered to pick up changes in the PARMLIB and SYSRES.

It is also recommended that customer change control procedures are updated 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.

IdML XML Viewer

The idml_browse.hta utility can be used to browse the output IdML XML files in a web page format, with hyperlink navigation based on discovered relationships.
IdML XML Viewer

idml_browse.hta can be downloaded for free from the IBM Tivoli Open Process Automation Library (OPAL) website
http://catalog.lotus.com/wps/portal/topal

Search for “IDML Viewer” or “idml_browse.hta”.

To use idml_browse.hta:
- 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 idml_browse.hta and specify the IdML XML file, or click the Browse button to locate an IdML XML file that you want to view.
- Click the Submit button to have the XML file read in and formatted in the window.

![IdML XML File Browser](image)

Figure 10. The Discovery Library IdML XML File Browser

z/OS DLA FTP utility

The zosdla_ftp.hta utility is a FTP front end that can be used from Windows® to run the z/OS DLA, then download and view the output IdML files with the idml_browse.hta utility.
zosdla_ftp.hta can be downloaded for free from the IBM Tivoli Open Process Automation Library (OPAL) website
http://catalog.lotus.com/wps/portal/topal

Search for “z/OS DLA FTP” or “zosdla_ftp.hta”.

Figure 11. The z/OS DLA FTP utility

Loading the XML files into Tivoli CCMDB

Once the XML files have been downloaded to the Discovery Library File Store, the CCMDB bulk loader program loadidml.sh can be used to load the books into, for example, CCMDB.

- su - taddm
- cd /opt/IBM/cmdb/dist/bin
- ./loadidml.sh -f <path_to_idml_file>
Loading the XML files into Tivoli CCMDB
Chapter 4. User reference

z/OS DLA filtering

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 filtering directives SET, DISC, FILT and BOOK (detailed below) 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 CMDB.

The DISC directive is used to specify which resources should be discovery. This directive can be used to limit discovery to say CICS or IMS only, or to only discover IMS database resources but no IMS program resources.

The FILT directive is used to limit IdML output.

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.

A number of sample members are provided in the hlq.SIZDSAMP library. IZDCDEF contains the default directive recommended for a typical discovery. Other books are described in Appendix C, “Samples provided in hlq.SIZDSAMP,” on page 113.

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:

Index  Value
Orgname
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 <default>

REFRESH
Yes to generate a refresh configuration.
No to generate an update configuration.

The default is No.
SET directive

FTP  Yes to generate an output member containing FTP put instructions.
The default is Yes.

FTP_GET  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 dataset or PDS member containing the initial FTP get control statements.
The default is No.

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

DEBUG_AS  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_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_IZDPIPLI**
The default is 0.

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

**DEBUG_IZDPSYSI**
The default is 0.

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

**DEBUG_IZDUPDSR**
The default is 0.

**LIMIT**

n - An integer specifying a limit to the number of resources to be discovered for each resource type (IMS, CICS and DB2). This value is used where no more specific limit is specified.

The default is 100.

**LIMIT.IMS**

n - An integer specifying a limit to the number of resources to be discovered for each IMS resource type. The limit applies to each IMS subsystem.

The default is 100.

**LIMIT.IMS.jobname**

n - An integer specifying a limit to the number of resources to be discovered for the specified IMS subsystem. The jobname must match the IMS Control Region jobname or STC name.

The default is 100.

**LIMIT.IMS.PROGRAM**

n - An integer specifying a limit to the number IMS program resources to be discovered.

This parameter sets the maximum number of IMSProgram instances populated per IMS Region.

If the IdML book is to be loaded into CCMDB, performance degradation can occur if this limit is increased and it is strongly recommended not to set a limit higher than 2000.

Note that the ProgramsChecksum attribute is calculated based on all programs defined to the IMS Region, regardless of this limit.
SET directive

The default is 100.

LIMIT.IMS.PROGRAM.jobname
n - An integer specifying a limit to the number of IMS program resources to be discovered for the specified IMS subsystem. The jobname must match the IMS Control Region jobname or STC name.

The default is 100.

LIMIT.IMS.TRANSACTION
n - An integer specifying a limit to the number of IMS transaction resources to be discovered.

This parameter sets the maximum number of IMSTransaction instances populated per IMS Region.

If the IdML book is to be loaded into CCMDB, performance degradation can occur if this limit is increased and it is strongly recommended not to set a limit higher than 2000.

Note that the TransactionsChecksum attribute is calculated based on all transactions defined to the IMS Region, regardless of this limit.

LIMIT.IMS.TRANSACTION.jobname
n - An integer specifying a limit to the number of IMS transaction resources to be discovered for the specified IMS subsystem. The jobname must match the IMS Control Region jobname or STC name.

The default is 100.

LIMIT.IMS.DATABASE
n - An integer specifying a limit to the number of IMS database resources to be discovered.

This parameter sets the maximum number of IMSDatabase instances populated per IMS Region.

If the IdML book is to be loaded into CCMDB, performance degradation can occur if this limit is increased and it is strongly recommended not to set a limit higher than 2000.

Note that the DatabasesChecksum attribute is calculated based on all Transactions defined to the IMS Region, regardless of this limit.

LIMIT.IMS.DATABASE.jobname
n - An integer specifying a limit to the number of IMS database resources to be discovered for the specified IMS subsystem. The jobname must match the IMS Control Region jobname or STC name.

The default is 100.

LIMIT.IMS.TRANPROG
n - An integer specifying a limit to the number of IMS transaction to program relationships to be discovered.

The default is 100.

LIMIT.IMS.TRANPROG.jobname
n - An integer specifying a limit to the number of IMS transaction to program relationships to be discovered for the specified IMS subsystem. The jobname must match the IMS Control Region jobname or STC name.

The default is 100.
LIMIT.CICS
n - An integer specifying a limit to the number of resources to be discovery for each CICS resource type. The limit applies to each CICS region.

The default is 100.

LIMIT.CICS.jobname
n - An integer specifying a limit to the number of resources of each type to be discovered for the specified CICS region. The jobname must match the CICS jobname or STC name.

The default is 100.

LIMIT.CICS.TRANSACTION
n - An integer specifying a limit to the number of CICS transaction resources to be discovered.

This parameter sets the maximum number of CICSTransaction instances populated per CICS Region.

If the IdML book is to be loaded into CCMDB, performance degradation can occur if this limit is increased and it is strongly recommended not to set a limit higher than 2000.

Note that the TransactionsChecksum attribute is calculated based on all Transactions defined to the CICS Region, regardless of this limit.

LIMIT.CICS.TRANSACTION.jobname
n - An integer specifying a limit to the number of CICS transaction resources to be discovered for the specified CICS region. The jobname must match the CICS jobname or STC name.

The default is 100.

LIMIT.CICS.PROGRAM
n - An integer specifying a limit to the number of CICS program resources to be discovered.

This parameter sets the maximum number of CICSPogram instances populated per CICS Region.

If the IdML book is to be loaded into CCMDB, performance degradation can occur if this limit is increased and it is strongly recommended not to set a limit higher than 2000.

Note that the ProgramsChecksum attribute is calculated based on all Transactions defined to the CICS Region, regardless of this limit.

LIMIT.CICS.PROGRAM.jobname
n - An integer specifying a limit to the number of CICS program resources to be discovered for the specified CICS region. The jobname must match the CICS jobname or STC name.

The default is 100.

LIMIT.CICS.TRANPROG
n - An integer specifying a limit to the number of CICS transaction to program relationships to be discovered.

The default is 100.

LIMIT.CICS.TRANPROG.jobname
n - An integer specifying a limit to the number of CICS transaction to program relationships to be discovered for the specified CICS region. The jobname must match the CICS region jobname or STC name.

The default is 100.
SET directive

LIMIT.CICS.FILE

n - An integer specifying a limit to the number of CICS file resources to be discovered.

This parameter sets the maximum number of CICSFile instances populated per CICS Region.

If the IdML book is to be loaded into CCMDB, performance degradation can occur if this limit is increased and it is strongly recommended not to set a limit higher than 2000.

Note that the FilesChecksum attribute is calculated based on all Transactions defined to the CICS Region, regardless of this limit.

LIMIT.CICS.FILE.jobname

n - An integer specifying a limit to the number of CICS file resources to be discovered for the specified CICS region. The jobname must match the CICS jobname or STC name.

The default is 100.

LIMIT.DB2

n - An integer specifying a limit to the number of resources to be discovery for each DB2 resource type. The limit applies to each DB2 subsystem.

The default is 100.

LIMIT.DB2.jobname

n - An integer specifying a limit to the number of resources of each type to be discovered for the specified DB2 subsystem. The jobname must match the DB2 master jobname or STC name.

The default is 100.

LIMIT.DB2.DATABASE

n - An integer specifying a limit to the number DB2 database resources to be discovered.

This parameter sets the maximum number of DB2Database instances populated per DB2 Region.

If the IdML book is to be loaded into CCMDB, performance degradation can occur if this limit is increased and it is strongly recommended not to set a limit higher than 2000.

LIMIT.DB2.DATABASE.jobname

n - An integer specifying a limit to the number of DB2 database resources to be discovered for the specified DB2 subsystem. The jobname must match the DB2 master jobname or STC name.

The default is 100.

LIMIT.DB2.TABLESPACE

n - An integer specifying a limit to the number of DB2 tablespace resources to be discovered. This parameter sets the maximum number of DB2Transaction instances populated per DB2 Region.

If the IdML book is to be loaded into CCMDB, performance degradation can occur if this limit is increased and it is strongly recommended not to set a limit higher than 2000.

LIMIT.DB2.TABLESPACE.jobname

n - An integer specifying a limit to the number of DB2 tablespace resources
to be discovered for the specified DB2 subsystem. The jobname must match the DB2 master jobname or STC name.

The default is 100.

**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.

This parameter could be used for example to restrict discovery to only CICS resources, allowing the CICS support group to run CICS only discovery independent of other discovery processes.

Syntax: DISC index value

**Supported indexes:**

<table>
<thead>
<tr>
<th>Index</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2</td>
<td>Yes to include No to exclude DB/2 discovery. The default is YES.</td>
</tr>
<tr>
<td>DB2.IFI</td>
<td>Yes to include and No to exclude DB/2 IFI discovery routines. This enables DB/2 high level discovery. 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.</td>
</tr>
<tr>
<td>DB2.SQL</td>
<td>Yes to include and No to exclude DB/2 SQL discovery routines. This enables discovery of: SYSIBM.SYSDATABASE/SysdatabaseMaxAlterdts attribute SYSIBM.SYSTABLESPACE/SystablespaceMaxAlterdts attribute SYSIBM.SYSTABLES/SystablesMaxAlterdts attribute SYSIBM.SYSINDEXES/SysindexesMaxAlterdts attribute SYSIBM.SYSCOLUMNS/SyscolumnsMaxAlterdts attribute Attributes. Authority to issue SQL statements against these DB2 resources is required to use this option. DSNREXX must be bound to each DB/2 where SQL discovery is required. The default is Yes.</td>
</tr>
<tr>
<td>DB2.DATABASE</td>
<td>Yes to include and No to exclude DB/2 database discovery. The default is No.</td>
</tr>
<tr>
<td>DB2.GROUP</td>
<td>Yes to include and No to exclude DB/2 group discovery. DB/2 group discovery is required to populate the DB/2 VersionString attribute. The default is Yes.</td>
</tr>
</tbody>
</table>
DISC directive

**DB2.TABLESPACE**
Yes to include and No to exclude DB/2 tablespace discovery.
The default is No.

**WEBSPHERE**
Yes to include and No to exclude WebSphere discovery routines.
The default is Yes.

**IMS**
Yes to include and No to exclude IMS discovery routines.
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.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.

**IMS.DEPREG**
Yes to include and No to exclude IMS dependent region discovery.
The default is Yes.

**MQ**
Yes to include and No to exclude MQ discovery routines.
The default is Yes.

**CICS**
Yes to include and No to exclude CICS discovery routines.
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**
Yes to include and No to exclude DASD storage volume discovery.
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.

Syntax: FILT index value

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

CDATA
EXCLUDE suppresses output for all attributes that contain CDATA, which includes: CICS SIT, AddressSpace Allocations and z/OS Parmlib attributes.

The default is INCLUDE.

JOBNAME.jobname
EXCLUDE suppresses output for all started tasks, jobs and TSO users that match the jobname or partial jobname.

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

The default is INCLUDE.

IMS.PROGRAM
EXCLUDE suppresses output for IMS program resources.

The default is EXCLUDE.

IMS.PROGRAM.jobname
EXCLUDE suppresses output for IMS program resources for a specific IMS control region.

The default is EXCLUDE.

IMS.TRANSACTION
EXCLUDE suppresses output for IMS transaction resources.

The default is EXCLUDE.

IMS.TRANSACTION.jobname
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.jobname
EXCLUDE suppresses output for IMS program to transaction relationships for a specific IMS control region.

The default is EXCLUDE.

CICS.PROGRAM
EXCLUDE suppresses output for CICS program resources.

The default is EXCLUDE.

CICS.PROGRAM.jobname
EXCLUDE suppresses output for CICS program resources for a specific CICS region.

The default is EXCLUDE.

CICS.TRANSACTION
EXCLUDE suppresses output for CICS transaction resources.

The default is EXCLUDE.
CICS.TRANSACTION.jobname
EXCLUDE suppresses output for CICS transaction resources for a specific CICS region.
The default is EXCLUDE.

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

CICS.TRANPROG.jobname
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.jobname
EXCLUDE suppresses output for CICS file resources for a specific CICS region.
The default is EXCLUDE.

DB2DATABASE
EXCLUDE suppresses output for DB2 database resources.
The default is EXCLUDE.

DB2.DATABASE.jobname
EXCLUDE suppresses output for DB2 database resources for a specific DB2 subsystem.
The default is EXCLUDE.

DB2.TABLESPACE
EXCLUDE suppresses output for DB2 tablespace resources.
The default is EXCLUDE.

DB2.TABLESPACE.jobname
EXCLUDE suppresses output for DB2 tablespace resources for a specific DB2 subsystem.
The default is EXCLUDE.

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>
</tbody>
</table>
BOOK directive

The default is No.

TYPE CICS
Yes creates output books for all discovered CICS regions.
The default is No.

TYPE IMS
Yes creates output books for all discovered IMS regions.
The default is No.

TYPE CICS
Yes creates output books for all discovered MQ regions.
The default is No.

TYPE WEBSPHERE
Yes creates output books for all discovered WEBSPHERE regions.
The default is No.

ZOSALL
Yes creates one output book named ZOSALL for all discovered resources.
The default is No.

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.
Appendix A. Messages and codes

Messages

IZD0000W  MESSAGE message_number UNDEFINED
Explanation: Message message_number is not defined in member IZDMSCGEN in SIIDZMSGS.
The variable message_number shows the number of the message for which text was missing.
System action: Processing is halted.
Operator response: None.
System programmer response: Check that SIZDMMSG is correctly specified in the Discovery JCL.

IZD0001I  text
Explanation: Message message_number is not defined in member IZDMSCGEN in SIIDZMSGS.
The variable message_number shows the number of the message for which text was missing.
System action: Processing is halted.
Operator response: None.
System programmer response: Check that SIZDMMSG is correctly specified in the Discovery JCL.

IZD0002I  z/OS DLA Started date time
Explanation: z/OS DLA processing has started. The date and time are provided for correlation.
The variable date shows the current date.
The variable time shows the current time.
System action: Processing continues.
Operator response: None.
System programmer response: None.

IZD0003I  z/OS DLA Completed date time
Explanation: z/OS DLA processing has completed. The date and time are provided for correlation.
The variable date shows the current date.
The variable time shows the current time.
System action: None.
Operator response: None.
System programmer response: None.

IZD0004E  condition TRAPPED ON LINE line number, RC return code description
Explanation: An error occurred during REXX processing.
The variable condition shows the REXX condition that was detected.
The variable line number shows the REXX line number where the error occurred.
The variable return code shows the REXX return code.
The variable description shows a description of the error.
System action: Processing is halted.
Operator response: None.
System programmer response: Contact IBM for support.

IZD0005E  FATAL ERROR - UNABLE TO CONTINUE
Explanation: A fatal error occurred during REXX processing.
System action: Processing is halted.
Operator response: None.
System programmer response: Contact IBM for support.

IZD0006I  z/OS DLA Processing completed successfully
Explanation: Processing has completed successfully.
System action: None.
Operator response: None.
System programmer response: None.

IZD0007E  EXECIO DISKR FROM ddname FAILED - RC return code
Explanation: An I/O error has occurred reading from DD ddname during REXX processing.
The variable ddname shows the name of the output DD.
The variable return code shows the EXECIO return code.
System action: Processing is halted.
Operator response: None.
Messages

System programmer response: Contact IBM for support.

IZD0008E  EXECIO DISKW FROM ddname FAILED - RC - return code
Explanation: An I/O error has occurred writing to DD ddname during REXX processing.
The variable ddname shows the name of the output DD.
The variable return code shows the EXECIO return code.
System action: Processing is halted.
Operator response: None.
System programmer response: Contact IBM for support.

IZD0009W  DISCOVERY ROUTINE modname RETURNED NO DATA
Explanation: Module modname returned no data in the output stem.
The variable modname shows the name of the module that returned a null stem variable.
System action: Processing continues but no data was discovered by modname. This may lead to errors in the idml output.
Operator response: None.
System programmer response: Contact IBM for support.

IZD0010W  DISCOVERY ROUTINE modname GAVE BAD RETURN CODE, RC return code
Explanation: Module modname gave bad return code return code.
The variable modname shows the name of the module that returned a null stem variable.
The variable return code shows the return code.
System action: Processing continues but discovery data may be incomplete.
Operator response: None.
System programmer response: Contact IBM for 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.
The variable control region shows the name of the IMS control region in error.
The variable version shows the IMS control region’s version.
System action: Processing continues but no data is returned for the named control region and any associated dependent regions.
Operator response: None.
System programmer response: None.

IZD0012W  ERROR DATA RETURNED BY: modname ssid debug command)
Explanation: DB/2 discovery routines have encountered an error while attempting to issue a command using the DB/2 IFI interface.
The variable modname shows the name of the module in error.
The variable ssid shows the DB/2 subsystem id.
The variable debug shows any debug data.
The variable command shows the command that was issued using the API that failed.
System action: Processing continues but no data is returned for the named DB/2 subsystem.
Operator response: None.
System programmer response: Message IZD0012A should follow with additional data.

IZD0013W  data
Explanation: DB/2 discovery routines have encountered an error while attempting to issue a command using the DB/2 IFI interface.
The variable data shows the data returned from the IFI interface.
System action: Processing continues but no data is returned for the named DB/2 subsystem.
Operator response: None.
System programmer response: Message IZD0012A should follow proceed with additional data.

IZD0014W  UNABLE TO ACCESS dataset name - SYSDSN gave: sysdsn data
Explanation: An attempt was made to read a dataset or member but the REXX SYSDSN() function returned an error.
The variable dataset name shows the name of the data set data returned from the IFI interface.
The variable sysdsn data shows the data returned by SYSDSN().
System action: Processing continues but no data is returned from the named dataset.
### Messages

**Operator response:** None.

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

---

**IZD0015W** UNABLE TO FIND DDNAME `ddname` IN allocation string

**Explanation:** An attempt to allocate a dataset failed.

The variable `ddname` shows the name of the DD.

The variable `allocation string` shows the allocation details.

**System action:** Processing.

**Operator response:** None.

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

---

**IZD0016W** UNABLE TO OPEN PATH `pathname` - ...<br>**Explanation:** An attempt to open path `pathname` failed.

The variable `pathname` shows the name of the path.

The variable `return code` shows the return code.

**System action:** Processing continues.

**Operator response:** None.

**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 `filename` DD `ddname`

**Explanation:** No data was returned after a read of `filename`.

The variable `filename` shows the name of the file.

The variable `ddname` shows the DDNAME.

**System action:** Processing continues.

**Operator response:** None.

**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 `xxx` IN `yyy` `zzz`

**Explanation:** No data was returned after a read of `filename`.

The variable `xxx` shows the name of the file.

The variable `yyy` shows the jobname.

---

The variable `zzz` shows the `ddname`.

**System action:** Processing continues.

**Operator response:** None.

**System programmer response:** None.

---

**IZD0019W** FATAL ERROR - UNABLE TO FIND `xxx`

**Explanation:** A fatal error occurred.

The variable `xxx` shows Hostname or IPV4 address.

**System action:** Processing is halted.

**Operator response:** None.

**System programmer response:** None.

---

**IZD0020I** DISCOVERY OF `discovery data` SUPPRESSED BY RULE

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

The variable `discovery data` shows the data that will not be discovered.

**System action:** Processing continues.

**Operator response:** None.

**System programmer response:** None.

---

**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 is halted.

**Operator response:** None.

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

---

**IZD0022I** `xxx` `yyy` : `zzz`

**Explanation:** Discovery for function `zzz` has started or ended.

The variable `xxx` shows the date.

The variable `yyy` shows the time.

The variable `zzz` shows the discovery function.

**System action:** Processing continues.

**Operator response:** None.

**System programmer response:** None.

---

Appendix A. Messages and codes
Messages

IZD0023I  CHECKSUM DATA UPDATED
Explanation: A change was detected between the current discovery and the previous discover.
System action: Processing is continued. The checksum and the associated output book are updated.
Operator response: None.
System programmer response: None.

IZD0024E  CHECKSUM CLEARED; BAD RUN
Explanation: Discovery has failed and checksum data for all output members has been cleared.
System action: Processing is terminated.
Operator response: None.
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. As the book data has not changed no output book is written. The variable book_name shows the name of the IDML book for which no change was detected.
System action: Processing continues, no output book is written.
Operator response: None.
System programmer response: None.

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 IZD0026I follows this message. The variable command shows the name of the MQ command.
System action: Processing continues.
Operator response: None.
System programmer response: Check for additional messages to determine why the MQ command may be incorrectly processed. Turn on debug for MQ and IZDMCMDI.

IZD0027I  command_data
Explanation: Discovery has received the following unexpected response to an MQ command issued using the MQ API command interface. Message IZD0026I precedes this message. The variable command_data shows the data returned from the MQ command.
System action: Processing continues.
Operator response: None.
System programmer response: Check for additional messages to determine why the MQ command may be incorrectly processed. Turn on debug for MQ and IZDMCMDI.

IZD0028I  POSSIBLE CAUSE: DSNREXX NOT BOUND TO DB2
Explanation: This message is preceded by IZD0010W and indicates a possible reason for the DB/2 request failure.
System action: Processing continues.
Operator response: None.
System programmer response: DSNREXX must be bound to DB/2 before discovery can issue DB/2 SQL requests. If DSNREXX is bound to DB/2 then check for additional messages to determine why the DB/2 request may have failed. Turn on debug for DB/2 and IZDDCMDI.

IZD0029I  POSSIBLE CAUSE: NOT AUTHORIZED
Explanation: This message is preceded by IZD0010W and indicates a possible reason for the DB/2 request failure.
System action: Processing continues.
Operator response: None.
System programmer response: The discovery user must have DB/2 or RACF authority to issue SQL requests to query SYSIBM resources and issue DB/2 DISPLAY commands. If the correct authorities are granted then check for additional messages to determine why the DB/2 request may have failed. Turn on debug for DB/2 and IZDDCMDI.

IZD0030I  discovery_type DISCOVERY SUPPRESSED
Explanation: Discovery has been suppressed for discovery_type due to a DISC directive. The variable discovery_type shows the type of discovery that was suppressed.
System action: Processing continues.
Operator response: None.
System programmer response: None.

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.

Operator response: None.

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. The variable ftp_type shows the type of FTP control statements that have been updated. This can be GET or PUT.

System action: Processing continues.

Operator response: None.

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

IZD0033I  FTP CHANGED ftp_type DATA CLEARED: BAD RUN

Explanation: The FTP control statements for type ftp_type have been cleared due to bad discovery run. The variable ftp_type shows the type of FTP control statements that have been cleared. This can be GET or PUT.

System action: Processing is terminated.

Operator response: None.

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. The variable ftp_type shows the type of FTP control statements that have been updated. This can be GET or PUT.

System action: Processing continues.

Operator response: None.

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

IZD0035I  INCORRECT VALUE FOR directive specified: value

Explanation: An incorrect directive value has been specified as input to discovery. The variable directive shows the discovery directive. The variable value shows the value specified.

System action: Processing continues.

Operator response: None.

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. The variable parameter shows the parameter specified.

System action: Processing continues.

Operator response: None.

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 via a LIMIT directive and more than the limited number of resources had been discovered. The variable limit_specified shows output limit specified via the LIMIT directive. The variable resources_found shows the number of resources or resource type resource_type that were discovered. The variable resource_type shows the resource type that was discovered.

System action: Processing continues.

Operator response: None.

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

IZD0038I  POSSIBLE CAUSE: WRONG DB/2 SDSNLOAD IN STEPLIB/LINKLST

Explanation: This message is preceded by IZD0010W and indicates a possible reason for the DB/2 request failure.

System action: Processing continues.

Operator response: None.

System programmer response: Ensure the DB/2 SDSNLOAD library in the STEPLIB or LINKLST matches the DB/2 version of the DB/2 subsystem being discovered. If the correct SDSNLOAD is accessible to discovery then check for additional messages to determine why the DB/2 request may
Messages

have failed. Turn on debug for DB/2 and IZDDCMDI.

IZD0039I   POSSIBLE CAUSE: UNSUPPORTED DB/2 LEVEL
Explanation: This message is preceded by IZD0010W and indicates a possible reason for the DB/2 request failure.
System action: Processing continues.
Operator response: None.
System programmer response: Discovery support DB/2 version 8 and above. If a supported version of DB/2 is being discovered then check for additional messages to determine why the DB/2 request may have failed. Turn on debug for DB/2 and IZDDCMDI.

IZD0041I   BOOK FOR disc_type disc_jobname SKIPPED: DISCOVERY FORBIDDEN
Explanation: An output limit has been specified via a LIMIT directive and more than the limited number of resources had been discovered. The variable disc_type shows the type of discovery being processed. The variable disc_jobname shows 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.
Operator response: None.
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.

IZD1000E   modname SYSEVENT MACRO FAILED 
RC = return code
Explanation: The SYSEVENT(QVS) macro has failed.
The variable modname shows the name of the module that issued the SYSEVENT macro.
The variable return code shows the SYSEVENT return code.
System action: Processing is terminated.
Operator response: None.
System programmer response: Return codes for the SYSEVENT (QVS) macro are documented in z/OS MVS Authorized Assembler Services Reference.

IZD1002E   modname CAF LOAD OF DSNALI FAILED RC = return code RS = reason code
Explanation: DB/2 module DSNALI could not be loaded.
The variable modname shows the name of the module that issued the LOAD for DSNALI.
The variable return code shows the LOAD return code.
The variable reason code shows the LOAD reason code.

IZD1003E   modname CAF LOAD OF DSNWL12 FAILED RC = return code RS = reason code
Explanation: DB/2 module DSNWL12 could not be loaded.
The variable modname shows the name of the module that issued the LOAD for DSNWL12.
The variable return code shows the LOAD return code.
The variable reason code shows the LOAD reason code.
System action: Processing is terminated.
Operator response: None.
System programmer response: Return codes for the LOAD macro are documented in z/OS MVS Assembler Services Reference. Check that the DSNALI module is available to the DLA job via the STEPLIB or LNKLST.

IZD1004W   modname DB/2 CONNECT FAILED FOR ssid RC = return code RS = reason code
Explanation: DB/2 connect to subsystem ssid failed.
The variable modname shows the name of the module that issued the DB/2 connect.
The variable ssid shows the DB/2 subsystem that the connect was issued against.
The variable return code shows the connect return code.
The variable reason code shows the connect reason code.
System action: Processing is terminated.
Operator response: None.
System programmer response: Return codes for the connect macro are documented in DB/2 Codes.

IZD1005W   modname DB/2 COMMAND FAILED FOR ssid COMMAND command RC = return code RS = reason code BM = bytes moved BNM = bytes not moved R0 = register zero R15 = register fifteen
Explanation: A command issued via the DB/2 IFI command interface failed.
The variable *modname* shows the name of the module that issued the DB/2 command.

The variable *ssid* shows the DB/2 subsystem that command was issued against.

The variable *command* shows the DB/2 command that was issued.

The variable *return code* shows the IFI return code.

The variable *reason code* shows the IFI reason code.

The variable *bytes moved* shows the number of data bytes returned by the command.

The variable *bytes not moved* shows the number of bytes that could not be moved because of buffer overflow.

The variable *register zero* shows the contents of R0 on return from the IFI call.

The variable *register fifteen* shows the contents of R15 on return from the IFI call.

**System action:** Processing is terminated.

**Operator response:** None.

**System programmer response:** Return codes for the IFI interface are documented in DB/2 Codes.

---

**Explanation:** An attempt to extend the DB/2 IFI command buffer failed. The variable *modname* shows the name of the module that issued the DB/2 command. The variable *ssid* shows the DB/2 subsystem that command was issued against. The variable *command* shows the DB/2 command that was issued. The variable *return code* shows the STORAGE(OBTAIN) return code. The variable *bytes moved* shows the number of data bytes returned by the command. The variable *bytes not moved* shows the number of bytes that could not be moved because of buffer overflow.

**System action:** Processing is terminated.

**Operator response:** None.

**System programmer response:** Return codes for the IFI interface are documented in DB/2 for z/OS Codes.

---

**Explanation:** DB/2 disconnect from subsystem *ssid* failed.

The variable *modname* shows the name of the module that issued the DB/2 disconnect.

The variable *ssid* shows the DB/2 subsystem that the disconnect was issued against.

The variable *return code* shows the disconnect return code.

The variable *reason code* shows the disconnect reason code.

**System action:** Processing continues.

**Operator response:** None.

**System programmer response:** Return codes for the disconnect are documented in DB/2 Codes.
Messages

UCBINFO macro are documented in z/OS MVS Authorized Assembler Services Reference.

IZD1013E  modname IOSCDR MACRO FAILED RC = return code
Explanation:  The IOSCDR MACRO has failed.
The variable modname shows the name of the module that issued the IOSCDR macro.
The variable return code shows the IOSCDR return code.
System action:  Processing is terminated.
Operator response:  None.
System programmer response:  Return codes for the IOSCDR macro are documented in z/OS MVS Authorized Assembler Services Reference.

IZD1020W  modname SQLCODES = sql codes
Explanation:  The IOSCDR MACRO has failed.
The variable modname shows the name of the module that issued the SQL call.
The variable sql codes shows the SQL codes.
System action:  Processing is terminated.
Operator response:  None.
System programmer response:  Return codes for SQL calls area documented in DB/2 Codes.

IZD1030E  modname STORAGE MACRO FAILED FOR type RC = return code
Explanation:  The STORAGE MACRO has failed.
The variable modname shows the name of the module that issued the STORAGE macro.
The variable type shows the type of STORAGE request that failed.
The variable return code shows the STORAGE return code.
System action:  Processing is terminated.
Operator response:  None.
System programmer response:  Return codes for the STORAGE macro are documented in z/OS MVS Authorized Assembler Services Reference.

IZD1031E  modname IXCQUERY MACRO FAILED FOR type RC = return code RS = reason code
Explanation:  DB/2 module DSNALI could not be loaded.
The variable modname shows the name of the module that issued the IXCQUERY.
The variable type shows the IXCQUERY type (either GROUP or MEMBER).
The variable return code shows the IXCQUERY return code.
The variable reason code shows the IXCQUERY reason code.
System action:  Processing is terminated.
Operator response:  None.
System programmer response:  Return codes for the LOAD macro are documented in z/OS MVS Assembler Services Reference. Check that the DSNALI module is available to the DLA job via the STEPLIB or LNKLST.

IZD1040E  modname COULD NOT LOAD MODULE loadmod ABEND = return code RS = reason code
Explanation:  The named module could not be loaded.
The variable modname shows the name of the module that issued the LOAD for DSNALI.
The variable loadmod shows the name of the module that was being loaded.
The variable return code shows the LOAD return code.
The variable reason code shows the LOAD reason code.
System action:  Processing is terminated.
Operator response:  None.
System programmer response:  Return codes for the LOAD macro are documented in z/OS MVS Assembler Services Reference. Check that the module available to the DLA job via the STEPLIB or LNKLST.

IZD1041E  modname COULD NOT ATTACH MODULE attachmod RC = return code
Explanation:  The named module could not be attached.
The variable modname shows the name of the module that issued the LOAD for DSNALI.
The variable attachmod shows the name of the module that was being loaded.
The variable return code shows the LOAD return code.
The variable reason code shows the LOAD reason code.
System action:  Processing is terminated.
Operator response:  None.
System programmer response:  Return codes for the ATTACH macro are documented in z/OS MVS Assembler Services Reference.
IZD1050E  modname ENcountered unexpected return code form delete rc = return code
Explanation: A delete macro call has returned an unexpected return code.
The variable modname shows the name of the module that issued the DELETE.
The variable return code shows the LOAD return code.
System action: Processing continues.
Operator response: None.
System programmer response: Return codes for the DELETE macro are documented in z/OS MVS Assembler Services Reference.

IZD1051E  modname ENcountered unexpected return code form irxexcom rc = return code
Explanation: A call the REXX interface routine IRXEXCOM has failed.
The variable modname shows the name of the module that issued the call to IRXEXCOM.
The variable return code shows the IRXEXCOM return code.
System action: Processing is terminated.
Operator response: None.
System programmer response: Contact IBM for support.

IZD1052E  modname VARIABLE variable not coded in lookup table
Explanation: A call the REXX interface routine IRXEXCOM has failed.
The variable modname shows the name of the module that issued the call to IRXEXCOM.
The variable variable shows the name of the missing variable.
System action: Processing is terminated.
Operator response: None.
System programmer response: Contact IBM for support.

IZD1060I  modname ABEND abend code at offset offset1, retrying at offset offset2
Explanation: A recoverable ABEND has occurred.
The variable modname shows the name of the module that issued the call to IRXEXCOM.
The variable offset1 shows the ABEND location.
The variable offset2 shows the retry location.
System action: Processing continues.
Operator response: None.
System programmer response: None.

IZD1070E  modname MEMBER member name not found
Explanation: A PARMLIB member was not found in the system PARMLIB concatenation.
The variable modname shows the name of the module that attempts to read PARMLIB.
The variable member name shows the name of the missing member.
System action: Processing continues.
Operator response: None.
System programmer response: None.

IZD1071E  modname PARMLIB service failed when reading member name rc = return code rs = reason code
Explanation: A PARMLIB member could not be processed.
The variable modname shows the name of the module that attempts to read PARMLIB.
The variable member name shows the name of the member being read.
The variable return code shows the READ return code.
The variable reason code shows the READ reason code.
System action: Processing continues.
Operator response: None.
System programmer response: READ macro return codes are documented in DF/SMS Macro Services for Datasets.

IZD1072E  modname MEMBER member name too large
Explanation: A PARMLIB member could not be processed due to its size.
The variable modname shows the name of the module that attempts to read PARMLIB.
The variable member name shows the name of the member being read.
System action: Processing continues.
Operator response: None.
System programmer response: None.
The variable `modname` shows the name of the module that attempt to open the PDS/PDSE.

The variable `ddname` shows the name of DD that is allocated to the input PDS/PDSE.

The variable `return code` shows the return code from the OPEN macro.

**System action:** Processing continues.

**Operator response:** None.

**System programmer response:** OPEN macro return codes are documented in z/OS DFSMS Macro Instructions for Datasets.

---

**Explanation:** An attempt to read a PDS/PDSE directory has failed.

The variable `modname` shows the name of the module that attempt to read the PDS/PDSE.

The variable `ddname` shows the name of DD that is allocated to the input PDS/PDSE.

The variable `return code` shows the return code from the READ macro.

**System action:** Processing continues.

**Operator response:** None.

**System programmer response:** READ macro return codes are documented in z/OS DFSMS Macro Instructions for Datasets.

---

**Explanation:** The z/OS system codepage translation routine CUNLCNV has returned the indicated return and reason code. The variable `modname` shows the name of the module that issued the call to CUNLCNV. The variable `return code` shows the return code from CUNLCNV. The variable `reason code` shows the reason code from CUNLCNV.

**System action:** Processing continues but codepage translation may be in error.

**Operator response:** None.

**System programmer response:** Check the SET CODEPAGE directive for the correct codepage number.
**System action:** Processing continues.

**Operator response:** None.

**System programmer response:** This message should appear when the DEBUG options are set on.

IZD9000E  modname WAS NOT PASSED CORRECT PARAMETERS

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

The variable modname shows the name of the module that was called with incorrect parameters.

**System action:** Processing is terminated.

**Operator response:** None.

**System programmer response:** If the module name is IZDUIRX then check the JCL parameters otherwise contact IBM for support.

IZD9001I  modname CALLING: rexxname

**Explanation:** Module IZDUIRX has invoked REXX routine rexxname.

The variable modname shows the name of the IZDUIRX REXX front-end routine.

The variable rexxname shows the name of the invoked REXX routine.

**System action:** Processing continues.

**Operator response:** None.

**System programmer response:** None.

IZD9002I  modname PARMS: <parameters>

**Explanation:** Module IZDUIRX was called with the indicated parameters.

The variable modname shows the name of the IZDUIRX REXX front-end routine.

The variable parameters lists the input parameters.

**System action:** Processing continues.

**Operator response:** None.

**System programmer response:** None.

IZD9003I  modname DEBIG IS ON

**Explanation:** Module IZDUIRX was called in debug mode.

The variable modname shows the name of the IZDUIRX REXX front-end routine.

**System action:** Processing continues.

**Operator response:** None.

**System programmer response:** None.

IZD9004I  modname COMPLETED WITH RETURN CODE return code

**Explanation:** Module IZDUIRX was called in debug mode.

The variable modname shows the name of the IZDUIRX REXX front-end routine.

The variable return code shows the return code from IZDUIRX.

**System action:** Processing is complete.

**Operator response:** None.

**System programmer response:** None.

IZD9005E  modname ENCONTRUED AN UNEXPECTED RETURN CODE FROM rexxname RC = return code

**Explanation:** Module IZDUIRX received an unexpected return code from a called REXX routine.

The variable modname shows the name of the IZDUIRX REXX front-end routine.

The variable rexxname shows the name of the called REXX routine.

The variable return code shows the return code from the called REXX routine.

**System action:** Processing is complete.

**Operator response:** None.

**System programmer response:** Contact IBM for support.

IZD9006I  modname COMPLETED WITH RETURN CODE return code

**Explanation:** Module IZDUIRX has completed with the indicated return code.

The variable modname shows the name of the IZDUIRX REXX front-end routine.

The variable return code shows the return code from the called REXX routine.

**System action:** Processing is complete.

**Operator response:** None.

**System programmer response:** Contact IBM for support.
Codes

Return codes from IZDRDLA

0  Normal completion
1  Normal completion - FTP=N was specified.
   A non-zero return code is set to force a bypass of the FTP step.
4  z/OS DLA is not APF authorized or is not authorized to access requested data.
12  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.
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 refer to the relevant operating system or product manual for an explanation of return and reason codes.
Appendix B. 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, refer to the SDK, which is available on CCMDB / TADDM CD1. Unzip the CMDBWebsite.zip file from the sdk\doc\model directory and open “WebsiteFiles\misc\CDMSections.htm”.
ManagementSoftwareSystem

ManagementSoftwareSystem

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

Book members
All members.

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

Attributes

<table>
<thead>
<tr>
<th>CDM Naming rule</th>
<th>process.ManagementSoftwareSystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>Hostname</td>
<td>If IP stack - &lt;member&gt;@&lt;fqdn&gt;</td>
</tr>
<tr>
<td></td>
<td>Else &lt;member&gt;@&lt;SMFID&gt;</td>
</tr>
<tr>
<td>ManufacturerName</td>
<td>IBM</td>
</tr>
<tr>
<td>ProductName</td>
<td>IBM Tivoli Discovery Library Adapter for z/OS</td>
</tr>
<tr>
<td>ProductVersion</td>
<td>2.1.0</td>
</tr>
<tr>
<td>MSSName</td>
<td>Ibm-cdm:///CDMMSS/Hostname=</td>
</tr>
<tr>
<td></td>
<td>&lt;hostname&gt;+ManufacturerName=</td>
</tr>
<tr>
<td></td>
<td>IBM+ProductName=IBM z/OS</td>
</tr>
<tr>
<td></td>
<td>Discovery Library Adapter</td>
</tr>
</tbody>
</table>

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

Naming rules
MSSName

Relationships
n/a
Organization

The Organization class provides a GlobalName that can be used to distinguish between copies of the same z/OS and preserve uniqueness in the CMDB. For example, if the site has a production z/OS with SMFID called MVS1 and another z/OS in a Disaster Recovery environment with the same SMFID called MVS1, in order to distinguish between the z/OS images, the Organization must be set to different values. The value for GlobalName is provided via the ORGNAME parameter at discovery time or via the defaultOrg value when the IdML book file is loaded into the CMDB.

The z/OS DLA provides a configuration option that enables the customer to set the Global Name. The default is <defaultOrg> which means that CMDB configured default is used.

Book members
All members.

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

Attributes

<table>
<thead>
<tr>
<th>CDM Naming rule</th>
<th>process.Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>GlobalName</td>
<td>Customer assigned name or the literal &lt;defaultOrg&gt;</td>
</tr>
<tr>
<td>Id (IdML)</td>
<td>&lt;GlobalName&gt;-Organization</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;GlobalName&gt;-Organization</td>
</tr>
</tbody>
</table>

Naming rules
GlobalName

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>Sysplex</td>
<td>Refer to Sysplex class.</td>
</tr>
<tr>
<td>Owns</td>
<td>Organization</td>
<td>z/OS</td>
<td>Refer to z/OS class.</td>
</tr>
<tr>
<td>Owns</td>
<td>Organization</td>
<td>ZVM</td>
<td>Refer to ZVM class.</td>
</tr>
</tbody>
</table>
ZSeriesComputerSystem

ZSeriesComputerSystem
The ComputerSystem represents the instance of physical hardware, the box (CEC) that a z/OS image executes on.

Book members
ZOSBASE.

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

Attributes

<table>
<thead>
<tr>
<th>CDM Naming rule</th>
<th>sys.zOS.ZSeriesComputerSystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>Name</td>
<td>Customer assigned hardware name of the CEC.</td>
</tr>
<tr>
<td>Manufacturer</td>
<td>Manufacturer Id</td>
</tr>
<tr>
<td>Model</td>
<td>Machine type and model numbers concatenation, for example, concatenate 2066 and 004 (no hyphen).</td>
</tr>
<tr>
<td>SerialNumber</td>
<td>Serial Number</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>
</tr>
<tr>
<td>ProcessCapacityUnits</td>
<td>Millions of service units/hour</td>
</tr>
<tr>
<td>Type</td>
<td>ComputerSystem</td>
</tr>
<tr>
<td>Virtual</td>
<td>This is not a virtual computer system.</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;Name&gt;</td>
</tr>
<tr>
<td>Id (IdML)</td>
<td>&lt;Name&gt;-ZSeriesComputerSystem</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;Name&gt;-ZSeriesComputerSystem</td>
</tr>
</tbody>
</table>

Naming rules
Manufacturer, Model and SerialNumber.

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>LPAR</td>
<td>ZSeriesComputerSystem</td>
<td>Only populated if z/OS is direct under LPAR, or z/OS is under VM under LPAR.</td>
</tr>
<tr>
<td>runsOn</td>
<td>z/OS</td>
<td>ZSeriesComputerSystem</td>
<td>Only populated if z/OS is running Stand Alone, that is, no LPAR.</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>z/OS</td>
<td>ZSeriesComputerSystem</td>
<td>Only populated if z/OS is running Stand Alone, that is, no LPAR.</td>
</tr>
<tr>
<td>runsOn</td>
<td>ZVM</td>
<td>ZSeriesComputerSystem</td>
<td>Only populated if z/OS is running under a ZVM that is running Stand Alone, that is, no LPAR.</td>
</tr>
<tr>
<td>installedOn</td>
<td>ZVM</td>
<td>ZSeriesComputerSystem</td>
<td>Only populated if z/OS is running under a ZVM that is running Stand Alone, that is, no LPAR.</td>
</tr>
<tr>
<td>virtualizes</td>
<td>ZVMGuest</td>
<td>ZSeriesComputerSystem</td>
<td>Only populated if z/OS is running under a ZVM that is running Stand Alone, that is, no LPAR.</td>
</tr>
</tbody>
</table>
An LPAR is a Logical PARtition of the physical hardware where an operating system instance can run, for example, z/OS, z/VM®, z/Linux. It is possible for the computing platform to operate in either basic-mode (non-partitioned) or lpar-mode. The LPAR class instance is only discovered if the hardware is operating in LPAR mode. Only an LPAR that runs z/OS or z/OS hosted by z/VM are discovered.

**Book members**

ZOSBASE.

**Requirements for discovery**

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

**Attributes**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.LPAR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CDM Attribute</strong></td>
<td><strong>Attribute</strong></td>
</tr>
<tr>
<td>Name</td>
<td>Logical Partition name.</td>
</tr>
<tr>
<td>LPARID</td>
<td>Logical Partition ID.</td>
</tr>
<tr>
<td>VMID</td>
<td>Logical Partition ID. - Naming rule virtualizes id.</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>
</tr>
<tr>
<td>ProcessCapacityUnits</td>
<td>millions of service units/hour</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></td>
<td>* Name from 1.1.11.</td>
</tr>
<tr>
<td>Id (IdML)</td>
<td>&lt;Name&gt;-&lt;HardwareName*&gt;-LPAR</td>
</tr>
<tr>
<td></td>
<td>* Name from 1.1.11.</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;Name&gt;-&lt;HardwareName*&gt;-LPAR</td>
</tr>
<tr>
<td></td>
<td>* Name from 1.1.11.</td>
</tr>
</tbody>
</table>

**Naming rules**

VMID and virtualizes relationship to ZSeriesComputerSystem.
### 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>LPAR</td>
<td>ZSeriesComputerSystem</td>
<td>Only populated if z/OS is direct under LPAR, or z/OS is under VM under LPAR.</td>
</tr>
<tr>
<td>runsOn</td>
<td>z/OS</td>
<td>LPAR</td>
<td>Only populated if z/OS is direct under LPAR (no VM and not stand alone).</td>
</tr>
<tr>
<td>installedOn</td>
<td>ZVM</td>
<td>LPAR</td>
<td>Only populated if z/OS is under VM.</td>
</tr>
<tr>
<td>installedOn</td>
<td>ZVM</td>
<td>LPAR</td>
<td>Only populated if z/OS is under VM.</td>
</tr>
<tr>
<td>virtualizes</td>
<td>ZVMGuest</td>
<td>LPAR</td>
<td>Only populated if z/OS is under VM.</td>
</tr>
</tbody>
</table>
ZVM

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

Book members
ZOSBASE

Requirements for discovery
- Filtering enabled (which is the default). This class is also sensitive to the ZSeriesComputerSystem and LPAR filters.
- Only discovered if applicable, which is when z/OS is running in a ZVMGuest
- APF mode.

Attributes

<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>OSName</td>
<td>ZVM name</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 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>Organization</td>
<td>ZVM</td>
<td>Naming rule for ZVM.</td>
</tr>
<tr>
<td>runsOn</td>
<td>ZVMGuest</td>
<td>ZVM</td>
<td>Naming rule for ZVMGuest.</td>
</tr>
<tr>
<td>runsOn</td>
<td>ZVM</td>
<td>LPAR</td>
<td>Only populated if ZVM is running in LPAR.</td>
</tr>
<tr>
<td>installedOn</td>
<td>ZVM</td>
<td>LPAR</td>
<td>Only populated if ZVM is running in LPAR.</td>
</tr>
<tr>
<td>runsOn</td>
<td>ZVM</td>
<td>ZSeriesComputerSystem</td>
<td>Only populated if ZVM is running in Stand Alone mode</td>
</tr>
<tr>
<td>installedOn</td>
<td>ZVM</td>
<td>ZSeriesComputerSystem</td>
<td>Only populated if ZVM is running in Stand Alone mode</td>
</tr>
</tbody>
</table>
ZVMGuest

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

Book members
ZOSBASE

Requirements for discovery
- Filtering enabled (which is the default). This class is also sensitive to the ZSeriesComputerSystem and LPAR filters.
- Only discovered if applicable, which is when z/OS is running in a ZVMGuest
- APF mode is needed to discover the parent ZVM class, which is required for the ZVMGuest naming rule.

Attributes

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zVM.ZVMGuest</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>Name</td>
<td>VM Guest name as known by the z/VM host.</td>
</tr>
<tr>
<td>VMID</td>
<td>&lt;Guest name&gt;-&lt;VM name&gt;</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>
</tr>
<tr>
<td>ProcessCapacityUnits</td>
<td>millions of service units/hour</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 ZSeriesComputerSystem or 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>ZVMGuest</td>
<td>ZSeriesComputerSystem</td>
<td>Only populated if z/OS is running under a ZVM that is running Stand Alone, that is, no LPAR.</td>
</tr>
<tr>
<td>virtualizes</td>
<td>ZVMGuest</td>
<td>LPAR</td>
<td>Only populated if z/OS is under VM.</td>
</tr>
<tr>
<td>runsOn</td>
<td>ZVMGuest</td>
<td>ZVM</td>
<td></td>
</tr>
<tr>
<td>runsOn</td>
<td>z/OS</td>
<td>ZVMGuest</td>
<td></td>
</tr>
<tr>
<td>installedOn</td>
<td>z/OS</td>
<td>ZVMGuest</td>
<td></td>
</tr>
</tbody>
</table>
**ComputerSystem**

This class instance for backward compatibility with sensors that related IP Interface to a ComputerSystem instead of an OperationSystem.

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

**Attributes**

<table>
<thead>
<tr>
<th>CDM Attribute</th>
<th>Attribute</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>SMF ID</td>
<td>OMO1</td>
</tr>
<tr>
<td>Virtual</td>
<td>This is a virtual computer system.</td>
<td>true</td>
</tr>
<tr>
<td>Signature</td>
<td>The primary IP address, or if there is no IP stack then SNA_HOST.&lt;NetidSSCP&gt;.&lt;SMFID&gt;. This is the primary naming rule attribute.</td>
<td>192.168.123.53</td>
</tr>
<tr>
<td>Fqdn</td>
<td>Fully qualified host name if the z/OS has IP.</td>
<td>pthomo1.au.ibm.com</td>
</tr>
<tr>
<td>Type</td>
<td>ComputerSystem</td>
<td>ComputerSystem</td>
</tr>
<tr>
<td>Label</td>
<td>&lt;SMFID&gt;-z/OS</td>
<td>OMO1-z/OS</td>
</tr>
<tr>
<td>Id (IdML)</td>
<td>&lt;SMFID&gt;-ZOS-CS</td>
<td>OMO1-ZOS-CS</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;SMFID&gt;-ZOS-CS</td>
<td>OMO1-ZOS-CS</td>
</tr>
</tbody>
</table>

**Naming rules**

Signature.

**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>z/OS</td>
<td>ComputerSystem</td>
<td></td>
</tr>
<tr>
<td>contains</td>
<td>ComputerSystem</td>
<td>IpInterface</td>
<td></td>
</tr>
<tr>
<td>manages</td>
<td>ComputerSystem</td>
<td>WebSphereCell</td>
<td></td>
</tr>
<tr>
<td>manages</td>
<td>ComputerSystem</td>
<td>WebSphereNodel</td>
<td></td>
</tr>
</tbody>
</table>
IpInterface

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

Book members
All books.

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

Attributes

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>net.IpInterface</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;IPv4Address&gt;-IpInterface</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;IPv4Address&gt;-IpInterface</td>
</tr>
</tbody>
</table>

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>z/OS</td>
<td>IpInterface</td>
<td></td>
</tr>
<tr>
<td>contains</td>
<td>ComputerSystem</td>
<td>IpInterface</td>
<td></td>
</tr>
<tr>
<td>bindsTo</td>
<td>IpInterface</td>
<td>IPv4Address</td>
<td></td>
</tr>
<tr>
<td>bindsTo</td>
<td>TCPPort</td>
<td>IpInterface</td>
<td></td>
</tr>
<tr>
<td>bindsTo</td>
<td>UDPPort</td>
<td>IpInterface</td>
<td></td>
</tr>
</tbody>
</table>
**IPv4Address**

The IPv4Address class represents an IP address on z/OS. There can be multiple IP addresses and multiple IP stacks but initially the DLA only discovers the primary IP address.

**Book members**

All books.

**Requirements for discovery**

- Filtering enabled (which is the default).

**Attributes**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>net.IPv4Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>DotNotation</td>
<td>The dotted decimal IP address of the primary (home) address of the default IP stack on the z/OS host.</td>
</tr>
<tr>
<td>Label</td>
<td>&lt; DotNotation &gt;</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt; DotNotation &gt;-IPv4Address</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt; DotNotation &gt;-IPv4Address</td>
</tr>
</tbody>
</table>

**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>IpInterface</td>
<td>IPv4Address</td>
<td></td>
</tr>
<tr>
<td>assignedTo</td>
<td>Fqdn</td>
<td>IPv4Address</td>
<td></td>
</tr>
</tbody>
</table>
Fqdn

The fully qualified IP name of z/OS.

Book members
All books.

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

Attributes

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>Attribute</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>net.Fqdn</td>
<td>Fqdn</td>
<td>pthomo1.au.ibm.com</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;Fqdn&gt;-Fqdn</td>
<td>pthomo1.au.ibm.com-Fqdn</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;Fqdn&gt;-Fqdn</td>
<td>pthomo1.au.ibm.com-Fqdn</td>
</tr>
</tbody>
</table>

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>Fqdn</td>
<td>IPv4Address</td>
<td></td>
</tr>
</tbody>
</table>
z/OS

The z/OS class type represents the z/OS operating system image ie: the z/OS system.

**Book members**

All members, but only ZOSBASE has all attributes. ZOSTASK and subsystem members contain the naming rule related attributes only.

**Requirements for discovery**

- None. One instance is populated in all z/OS DLA books.
- Attributes/Relationships marked with an * are only discovered when the DLA runs in APF authorized mode.
- The Parmlib* attributes require the userid running the DLA to have read access to the PARMLIB concatenation of datasets. The DLA automatically determines the applicable PARMLIB dataset names.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Naming Policy</td>
<td>sys.zOS.ZOS</td>
</tr>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<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>
</tr>
<tr>
<td>SMFID</td>
<td>The SMFID value commonly used by support personnel to identify the z/OS instance. Note that often Sysname and SMF id are the same. OMEGAMON® uses the SMF id in its z/OS related Managed System Names.</td>
</tr>
<tr>
<td>NetID</td>
<td>SNA Network-ID</td>
</tr>
<tr>
<td>SSCP</td>
<td>VTAM® SSCP Name</td>
</tr>
<tr>
<td>NetidSSCP</td>
<td>&lt;netid&gt;.&lt;sscp&gt;</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>
</tr>
<tr>
<td>PrimaryJES</td>
<td>The job/subsystem name of the primary Job Entry Subsystem, used by support personnel.</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>
</tr>
<tr>
<td>VersionString</td>
<td>The Version, Release and Mod level of the z/OS.</td>
</tr>
<tr>
<td>SysResVolume</td>
<td>The name of the disk volume containing the system resident nucleus that z/OS was IPL'd with. This is important for support personnel to determine maintenance levels.</td>
</tr>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------</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>
</tr>
<tr>
<td>ProcessCapacityUnits</td>
<td>millions of service units/hour</td>
</tr>
<tr>
<td>IPLParmDataset</td>
<td>The dataset that contains the IPL parameters.</td>
</tr>
<tr>
<td>IPLParmMember</td>
<td>The member that contains the IPL parameters.</td>
</tr>
<tr>
<td>IPLParmDevice</td>
<td>IPL device number</td>
</tr>
<tr>
<td>IPLParmVolume</td>
<td>IPL volume</td>
</tr>
<tr>
<td>IPLTime</td>
<td>The date and time that z/OS was IPL'd.</td>
</tr>
<tr>
<td>FQDN</td>
<td>Fully qualified host name if the z/OS has IP.</td>
</tr>
<tr>
<td>Parmlibs</td>
<td>A list of the parmlib dataset used during IPL.</td>
</tr>
<tr>
<td></td>
<td>The list is in a table format with two columns: DatasetName and Volume.</td>
</tr>
<tr>
<td>ParmlibMembers</td>
<td>A list of the parmlib members used during IPL.</td>
</tr>
<tr>
<td></td>
<td>The list is in a table format with three columns: MemberName, Dataset, and Volume.</td>
</tr>
<tr>
<td>ParmlibALLOC</td>
<td>A copy of the ALLOCxx member(s) from the PARMLIB dataset concatenation that was used for the current IPL. This member specifies system allocation defaults.</td>
</tr>
<tr>
<td>ParmlibBPXPRM</td>
<td>A copy of BPXPRMxx member(s) from the PARMLIB dataset concatenation that was used for the current IPL. This member specifies the default z/OS UNIX System Services and file system parameters.</td>
</tr>
<tr>
<td>ParmlibCEEPRM</td>
<td>A copy of CEEPROMxx member(s) from the PARMLIB dataset concatenation that was used for the current IPL. This member specifies the default Language Environment® (LE) runtime environment file system parameters.</td>
</tr>
<tr>
<td>ParmlibCLOCK</td>
<td>A copy of CLOCKxx member(s) from the PARMLIB dataset concatenation that was used for the current IPL. This member specifies the default date and time system parameters.</td>
</tr>
<tr>
<td>ParmlibCOMMND</td>
<td>A copy of COMMANDxx member(s) from the PARMLIB dataset concatenation that was used for the current IPL. This member specifies a list of commands that are executed during IPL.</td>
</tr>
<tr>
<td>CDM Naming Policy</td>
<td>sys.zOS.ZOS</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>ParmlibCONSOL</td>
<td>A copy of CONSOLxx member(s) from the PARMLIB dataset concatenation that was used for the current IPL. This member specifies the system console configuration.</td>
</tr>
<tr>
<td>ParmlibCOUPLE</td>
<td>A copy of COUPLEExx member(s) from the PARMLIB dataset concatenation that was used for the current IPL. This member specifies the cross system coupling facility default parameters.</td>
</tr>
<tr>
<td>ParmlibCSVRTL</td>
<td>A copy of CSVRTLxx member(s) from the PARMLIB dataset concatenation that was used for the current IPL. This member specifies the RLTS configuration.</td>
</tr>
<tr>
<td>ParmlibCUNUNI</td>
<td>A copy of CUNUNIxx member(s) from the PARMLIB dataset 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 member(s) from the PARMLIB dataset 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 member(s) from the PARMLIB dataset 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 member(s) from the PARMLIB dataset 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 member(s) from the PARMLIB dataset 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 member(s) from the PARMLIB dataset 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 IEAAPFxx member(s) from the PARMLIB dataset concatenation that was used for the current IPL. This member specifies the default static Authorized Program Facility (APF) list.</td>
</tr>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------</td>
</tr>
<tr>
<td>ParmlibIEAFIX</td>
<td>A copy of IEAFIXxx member(s) from the PARMLIB dataset 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 IEALPAxx member(s) from the PARMLIB dataset concatenation that was used for the current IPL. This member specifies the static Modified Link Pack Area (MLPA) list.</td>
</tr>
<tr>
<td>ParmlibIEAOPT</td>
<td>A copy of IEAOPTxx member(s) from the PARMLIB dataset 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 member(s) from the PARMLIB dataset 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 member(s) from the PARMLIB dataset 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 member(s) from the PARMLIB dataset 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 member(s) from the PARMLIB dataset 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 member(s) from the PARMLIB dataset 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 member(s) from the PARMLIB dataset 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 member(s) from the PARMLIB dataset 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 member(s) from the PARMLIB dataset concatenation that was used for the current IPL. This member specifies the Storage Management Subsystem (SMS) definitions.</td>
</tr>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------</td>
</tr>
<tr>
<td>ParmlibIKJTSO</td>
<td>A copy of IKJTSOxx member(s) from the PARMLIB dataset 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 member(s) from the PARMLIB dataset concatenation that was used for the current IPL. This member specifies the static link list.</td>
</tr>
<tr>
<td>ParmlibLOAD</td>
<td>A copy of LOADxx member(s) from the PARMLIB dataset concatenation that was used for the current IPL. This member specifies the system configuration information.</td>
</tr>
<tr>
<td>ParmlibLPALST</td>
<td>A copy of LPALSTxx member(s) from the PARMLIB dataset concatenation that was used for the current IPL. This member specifies the static LPA list.</td>
</tr>
<tr>
<td>ParmlibMSTJCL</td>
<td>A copy of MSTJCLxx member(s) from the PARMLIB dataset concatenation that was used for the current IPL. This member specifies the Master Scheduler JCL.</td>
</tr>
<tr>
<td>ParmlibPROG</td>
<td>A copy of PROGxx member(s) from the PARMLIB dataset 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>ParmlibSCHED</td>
<td>A copy of SCHEDxx member(s) from the PARMLIB dataset 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>ParmlibSMFPRM</td>
<td>A copy of SMFPRMxx member(s) from the PARMLIB dataset concatenation that was used for the current IPL. This member specifies the System Management Facilities (SMF) parameters.</td>
</tr>
<tr>
<td>ParmlibVATLST</td>
<td>A copy of VATLSTxx member(s) from the PARMLIB dataset concatenation that was used for the current IPL. This member specifies the Volume Attribute (VAT) list.</td>
</tr>
<tr>
<td>DynamicLNKLSTName</td>
<td>Name of the currently active dynamic lnklst.</td>
</tr>
<tr>
<td>DynamicLNKLST</td>
<td>A copy of currently active Dynamic LNKLST. The LNKLST can be modified at anytime after IPL by an authorized operator command. The list is in a table format with two columns: DatasetName and Volume.</td>
</tr>
<tr>
<td>CDM Naming Policy</td>
<td>sys.zOS.ZOS</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>DynamicAPF</td>
<td>A copy of currently active Authorized Program Facility list. The APF list can be modified at anytime after IPL by an authorized operator command.</td>
</tr>
<tr>
<td></td>
<td>The list is in a table format with two columns: Volume and DatasetName.</td>
</tr>
<tr>
<td>DynamicEXIT</td>
<td>A copy of currently active system exits. System exits be modified at anytime after IPL by an authorized operator command.</td>
</tr>
<tr>
<td></td>
<td>The list is in a table format with two columns: ExitName and ModuleName.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>FmId</td>
<td>The SMP/E FMID for the z/OS system.</td>
</tr>
<tr>
<td>IODFDataset</td>
<td>The IODF dataset that was used during IPL.</td>
</tr>
<tr>
<td>IODFVolume</td>
<td>IODF volume that the IODF dataset resides on.</td>
</tr>
<tr>
<td>IODFTime</td>
<td>IODF configuration build date and time.</td>
</tr>
<tr>
<td>MasterCatalogDataset</td>
<td>The name of the master catalog dataset.</td>
</tr>
<tr>
<td>MasterCatalogVolume</td>
<td>The volume that the master catalog resides on.</td>
</tr>
<tr>
<td>SecurityPackage</td>
<td>The security package installed for this z/OS system. RACF or other.</td>
</tr>
<tr>
<td></td>
<td>Enumeration:</td>
</tr>
<tr>
<td></td>
<td>• Unknown</td>
</tr>
<tr>
<td></td>
<td>• Other</td>
</tr>
<tr>
<td></td>
<td>• RACF</td>
</tr>
<tr>
<td>Label</td>
<td>If Organization GlobalName = &quot;&lt;defaultOrg&gt;&quot;</td>
</tr>
<tr>
<td></td>
<td>The Label = &lt;SMFID&gt;-&lt;SysplexName&gt;</td>
</tr>
<tr>
<td></td>
<td>Else Label =</td>
</tr>
<tr>
<td></td>
<td>&lt;SMFID&gt;-&lt;SysplexName&gt;-&lt;GlobalName&gt;</td>
</tr>
<tr>
<td>Id (IdML)</td>
<td>&lt;SMFID&gt;-ZOS</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;SMFID&gt;-ZOS</td>
</tr>
</tbody>
</table>

**Naming rules**

The z/OS DLA populates several Naming rules:

- NetidSSCP
- SMFID and owns relationship to Organization
- FQDN
## 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>z/OS</td>
<td></td>
</tr>
<tr>
<td>memberOf</td>
<td>z/OS</td>
<td>SystemSpecificCollection</td>
<td></td>
</tr>
<tr>
<td>runsOn</td>
<td>z/OS</td>
<td>ZSeriesComputerSystem</td>
<td>If z/OS is in running stand alone.</td>
</tr>
<tr>
<td>installedOn</td>
<td>z/OS</td>
<td>ZSeriesComputerSystem</td>
<td>If z/OS is in running stand alone.</td>
</tr>
<tr>
<td>runsOn</td>
<td>z/OS</td>
<td>LPAR</td>
<td>If z/OS is in running direct under LPAR.</td>
</tr>
<tr>
<td>installedOn</td>
<td>z/OS</td>
<td>LPAR</td>
<td>If z/OS is in running direct under LPAR.</td>
</tr>
<tr>
<td>runsOn</td>
<td>z/OS</td>
<td>ZVMGuest</td>
<td>If z/OS is in running under ZVMGuest.</td>
</tr>
<tr>
<td>installedOn</td>
<td>z/OS</td>
<td>ZVMGuest</td>
<td>If z/OS is in running under ZVMGuest.</td>
</tr>
<tr>
<td>runsOn</td>
<td>z/OS</td>
<td>ComputerSystem</td>
<td>CS used for ipInterface.</td>
</tr>
<tr>
<td>runsOn</td>
<td>AddressSpace</td>
<td>z/OS</td>
<td></td>
</tr>
<tr>
<td>runsOn</td>
<td>CICSRegion</td>
<td>z/OS</td>
<td></td>
</tr>
<tr>
<td>hostedDependency</td>
<td>IMSSubsystem</td>
<td>z/OS</td>
<td></td>
</tr>
<tr>
<td>hostedDependency</td>
<td>DB2Subsystem</td>
<td>z/OS</td>
<td></td>
</tr>
<tr>
<td>hostedDependency</td>
<td>MQSubsystem</td>
<td>z/OS</td>
<td></td>
</tr>
<tr>
<td>accesses</td>
<td>z/OS</td>
<td>StorageVolume</td>
<td></td>
</tr>
</tbody>
</table>
**Sysplex**

A sysplex is a set of z/OS systems communicating and cooperating with each other through certain multisystem hardware components and software services to process customer workloads. Not all members of the sysplex need 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 may 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 and 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 doesn’t use a coupling facility.

**Book members**

All members.

**Requirements for discovery**

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

**Attributes**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.Sysplex</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<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>
</tr>
<tr>
<td>Label</td>
<td>If Organization GlobalName = &quot;&lt;defaultOrg&gt;&quot;</td>
</tr>
<tr>
<td></td>
<td>The Label = &lt;Name&gt;</td>
</tr>
<tr>
<td></td>
<td>Else Label = &lt;Name&gt;-&lt;GlobalName&gt;</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;Name&gt;-SYSPLEX</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;Name&gt;-SYSPLEX</td>
</tr>
</tbody>
</table>

**Naming rules**

Name and owns relationship to 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>Organization</td>
<td>Sysplex</td>
<td></td>
</tr>
<tr>
<td>hostedCollection</td>
<td>Sysplex</td>
<td>SystemSpecificCollection</td>
<td></td>
</tr>
<tr>
<td>Contains</td>
<td>Sysplex</td>
<td>IMSSysplexGroup</td>
<td></td>
</tr>
<tr>
<td>Contains</td>
<td>Sysplex</td>
<td>DB2DataSharingGroup</td>
<td></td>
</tr>
</tbody>
</table>
SystemSpecificCollection

So that Author products can use the discovery library information we must comply with the CIM. This class is necessary in order to be standards compliant.

**Book members**

All members.

**Requirements for discovery**

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

**Attributes**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>core.SystemSpecificCollection</th>
<th>Attribute</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Sysplex Node Set</td>
<td>Sysplex Node Set</td>
<td></td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;SysplexName *&gt;-SystemSpecificCollection</td>
<td>* Name from 1.1.22.</td>
<td>SYSPLEXO-SystemSpecificCollection</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;SysplexName *&gt;-SystemSpecificCollection</td>
<td>* Name from 1.1.22.</td>
<td>SYSPLEXO-SystemSpecificCollection</td>
</tr>
</tbody>
</table>

**Naming rules**

hostedCollection 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>hostedCollection</td>
<td>Sysplex</td>
<td>SystemSpecificCollection</td>
<td></td>
</tr>
<tr>
<td>memberOf</td>
<td>z/OS</td>
<td>SystemSpecificCollection</td>
<td></td>
</tr>
</tbody>
</table>
Address Space

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

Book members

ZOSTASK

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 * are only discovered when the DLA runs in authorized mode.

Attributes

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.AddressSpace</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>JobName</td>
<td>The name of the address space. Address spaces are also referred to as jobs. Jobnames are not guaranteed to be unique within a z/OS system but they usually are since it is difficult to manage them otherwise. Long-running jobs very likely have unique jobnames. Transient jobs may sometimes have the same jobname and may even run concurrently but these are not really of interest to this DLA. If the discovery encounters multiple instances of the jobname then only first instance is represented in the IdML book.</td>
</tr>
<tr>
<td>JobUserid</td>
<td>The userid associated with the job.</td>
</tr>
</tbody>
</table>
### CDMA Attribute
<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.AddressSpace</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CDM Attribute</strong></td>
<td><strong>Attribute</strong></td>
</tr>
<tr>
<td>JobType</td>
<td>The type determines the nature of the job and how support personnel typically interact with it.</td>
</tr>
<tr>
<td></td>
<td>“Started class” includes jobs that are initiated via 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 via the MODIFY command. Started tasks are often long running and are analogous to a daemon under *nix.</td>
</tr>
<tr>
<td></td>
<td>“Batch job” includes jobs that are started by JES. They have a JES Jobid with the “J” prefix. Jobs are often short running.</td>
</tr>
<tr>
<td></td>
<td>“System” is for jobs that are considered system jobs. They are typically started early in the IPL and have often have no JES Jobid value. SYS jobs usually run for the life of the IPL are analogous to a device driver process.</td>
</tr>
<tr>
<td></td>
<td>“TSO User” is for jobs that are TSO Users. They have a JES Jobid with the “T” prefix.</td>
</tr>
<tr>
<td>JobFunction</td>
<td>A descriptive string that can assist support personnel to identify the job.</td>
</tr>
<tr>
<td></td>
<td>SYSTEM_VTAM</td>
</tr>
<tr>
<td>ProcStep</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>IMSIRLM</td>
</tr>
<tr>
<td>StepName</td>
<td>Identifies the Jobstep name of the job. This is really only useful if the job is long running and has only one job step.</td>
</tr>
<tr>
<td></td>
<td>IMPT1IRM</td>
</tr>
<tr>
<td>Pgm</td>
<td>The name of the program load module.</td>
</tr>
<tr>
<td></td>
<td>DXRRLM00</td>
</tr>
<tr>
<td>Allocations</td>
<td>A list of DDNAMEs and associated DSNs.</td>
</tr>
<tr>
<td></td>
<td>STEPLIB SYS1.LINKLIB SYSRES</td>
</tr>
<tr>
<td>PgmParm</td>
<td>The optional parameter string that was passed to the program when it was started.</td>
</tr>
<tr>
<td></td>
<td>PRLM,1,NODISCON,5,1,6,NO,,IMP0IRM</td>
</tr>
<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;-AddressSpace</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;JobName&gt;-&lt;SMFID &gt;-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>AddressSpace</td>
<td>z/OS</td>
<td>Part of primary naming rule along with JobName attribute.</td>
</tr>
<tr>
<td>federates</td>
<td>IMSSubsystem</td>
<td>AddressSpace</td>
<td></td>
</tr>
<tr>
<td>federates</td>
<td>DB2Subsystem</td>
<td>AddressSpace</td>
<td></td>
</tr>
<tr>
<td>federates</td>
<td>MQSubsystem</td>
<td>AddressSpace</td>
<td></td>
</tr>
<tr>
<td>uses</td>
<td>AddressSpace</td>
<td>IMSSubsystem</td>
<td></td>
</tr>
<tr>
<td>uses</td>
<td>AddressSpace</td>
<td>DB2Subsystem</td>
<td></td>
</tr>
<tr>
<td>uses</td>
<td>AddressSpace</td>
<td>MQSubsystem</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>AddressSpace</td>
<td>TCPPort</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>AddressSpace</td>
<td>UDPPort</td>
<td></td>
</tr>
<tr>
<td>realizes</td>
<td>AddressSpace</td>
<td>WebSphereServer</td>
<td></td>
</tr>
<tr>
<td>uses</td>
<td>AddressSpace</td>
<td>StorageVolume</td>
<td>For all DASD volumes used in the allocations.</td>
</tr>
</tbody>
</table>
Each TcpPort is a TCP IP port that is mapped to an active address space.

**Book members**

ZOSTASK

**Requirements for discovery**

- Filtering enabled (which is the default).

**Attributes**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>net.TcpPort</th>
<th>Attribute</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>net.TcpPort</td>
<td>The port number</td>
<td>80</td>
</tr>
<tr>
<td>Label</td>
<td></td>
<td>&lt;Port&gt;-&lt;IPv4Address&gt;</td>
<td>80-192.168.123.54</td>
</tr>
<tr>
<td>id (IdML)</td>
<td></td>
<td>&lt;Port&gt;-&lt;IPv4Address&gt;-TcpPort</td>
<td>80-192.168.123.54-TcpPort</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td></td>
<td>&lt;Port&gt;-&lt;IPv4Address&gt;-TcpPort</td>
<td>80-192.168.123.54-TcpPort</td>
</tr>
</tbody>
</table>

**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>TCPPort</td>
<td>IpInterface</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>AddressSpace</td>
<td>TCPPort</td>
<td></td>
</tr>
</tbody>
</table>
UdpPort

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

Book members

ZOSTASK

Requirements for discovery

- Filtering enabled (which is the default).

Attributes

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>net.UdpPort</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>Port</td>
<td>The port number</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>

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>UDPPort</td>
<td>IpInterface</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>AddressSpace</td>
<td>UDPPort</td>
<td></td>
</tr>
</tbody>
</table>
**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.

**Book members**

ZOSTASK and subsystem books.

**Requirements for discovery**

- Filtering enabled (which is the default).

**Attributes**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>net.BindAddress</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>PortNumber</td>
<td>The port number</td>
</tr>
<tr>
<td>Path</td>
<td>(none)</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>

**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>BindAddress</td>
<td>IPv4Address</td>
<td></td>
</tr>
<tr>
<td>bindsTo</td>
<td>BindAddress</td>
<td>IPv4Address</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>WebSphereServer</td>
<td>BindAddress</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>IMSSubsystem</td>
<td>BindAddress</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>DB2Subsystem</td>
<td>BindAddress</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>MQSubsystem</td>
<td>BindAddress</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>CICSRegion</td>
<td>BindAddress</td>
<td></td>
</tr>
</tbody>
</table>
StorageSubSystem

Storage subsystem

Book members

ZOSTASK

Requirements for discovery

- Filtering enabled (which is the default).

Attributes

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>storage.StorageSubSystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manufacturer</td>
<td>Manufacturer</td>
</tr>
<tr>
<td>SerialNumber</td>
<td>Sequence Number</td>
</tr>
<tr>
<td>Type</td>
<td>Literal &quot;StorageDevice&quot;</td>
</tr>
<tr>
<td>Model</td>
<td>Model number &amp; Model Type number concatenated, for example, concatenate 2105 and F20 (no hyphen).</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;-StorageSubsystem</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;Manufacturer&gt;-&lt;SerialNumber&gt;-&lt;Model&gt;-StorageSubsystem</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>
</tr>
</thead>
<tbody>
<tr>
<td>Contains</td>
<td>StorageSubSystem</td>
<td>StorageVolume</td>
<td></td>
</tr>
</tbody>
</table>
StorageVolume

DASD Storage volume

Book members
ZOSTASK

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

Attributes

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>dev.StorageVolume</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>DeviceID</td>
<td>Device Number</td>
</tr>
<tr>
<td>Name</td>
<td>DASD Volume Serial Number</td>
</tr>
<tr>
<td>NumOfCylinders</td>
<td>Number of Cylinders this volume has</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 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>StorageSubSystem</td>
<td>StorageVolume</td>
<td>Naming rule for StorageVolume</td>
</tr>
<tr>
<td>memberOf</td>
<td>StorageVolume</td>
<td>StorageSubSystem</td>
<td></td>
</tr>
<tr>
<td>accesses</td>
<td>z/OS</td>
<td>StorageVolume</td>
<td>All online DASD</td>
</tr>
<tr>
<td>uses</td>
<td>AddressSpace</td>
<td>StorageVolume</td>
<td>Based on allocations</td>
</tr>
</tbody>
</table>
DB2Subsystems 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 * are only discovered when the DLA runs in authorized mode.

**Book members**

DB2 Subsystem book has all attributes and all relationships except AddressSpaces.

ZOSTASK book has naming rules and relationships to AddressSpaces.

**Requirements for discovery**

- Filtering enabled (which is the default).
- DB2 Subsystem and related address spaces must be active and fully initialized.
- Attributes/Relationships marked with an * are only discovered when the DLA runs in authorized mode.
- The userid running the DLA must have authority to issue the DB/2 DISPLAY DB(*) and DISPLAY GROUP(*) commands.
- The userid 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.

### Attributes

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.DB2Subsystem</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>DB/2 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>
</tbody>
</table>
## DB2Subsystem

### CDM Naming Policy

<table>
<thead>
<tr>
<th>CDM Attribute</th>
<th>Attribute</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<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>

### Label

<table>
<thead>
<tr>
<th></th>
<th>&lt;SubsetName&gt;-&lt;SMFID&gt;</th>
<th>DB7A-OM01</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>id (IdML)</th>
<th>&lt;SubsetName&gt;-&lt; SMFID &gt;-DB2Subsystem</th>
<th>DB7A-OM01-DB2Subsystem</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>sourceToken (IdML)</th>
<th>&lt;SubsetName&gt;-&lt; SMFID &gt;-DB2Subsystem</th>
<th>DB7A-OM01-DB2Subsystem</th>
</tr>
</thead>
</table>

### 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>DB2Subsystem</td>
<td>zOS</td>
<td>Part of primary naming rule a long with SubsystemName attribute.</td>
</tr>
<tr>
<td>federates</td>
<td>DB2DataSharingGroup</td>
<td>DB2Subsystem</td>
<td>For all related master address space, IRLM, DBM1, DIST, SPAS, WLM_SPAS</td>
</tr>
<tr>
<td>federates</td>
<td>DB2Subsystem</td>
<td>AddressSpace</td>
<td>For related DB2DataSharingGroup.</td>
</tr>
<tr>
<td>Uses</td>
<td>IMSSubsystem</td>
<td>DB2Subsystem</td>
<td>Discovered via IMS</td>
</tr>
<tr>
<td>Uses</td>
<td>CICSRegion</td>
<td>DB2Subsystem</td>
<td>Discovered via CICS</td>
</tr>
<tr>
<td>federates</td>
<td>DB2DataSharingGroup</td>
<td>DB2Subsystem</td>
<td>For related DB2DataSharingGroup.</td>
</tr>
<tr>
<td>accessedVia</td>
<td>DB2Subsystem</td>
<td>BindAddress</td>
<td>For all related federated address spaces that are listening on a TCP and UDP Port.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>This relationship is used by the DLA sensor to stitch relationships with distributed resources.</td>
</tr>
<tr>
<td>contains</td>
<td>DB2Subsystem</td>
<td>Db2Database</td>
<td></td>
</tr>
</tbody>
</table>
**DB2DataSharingGroup**

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

**Book members**

DB2 Subsystem book has all attributes and all relationships except AddressSpaces.

**Requirements for discovery**

- Filtering enabled (which is the default).
- DB2 subsystem and related address spaces must be active and fully initialized.
- Attributes/Relationships marked with an * are only discovered when the DLA runs in authorized mode.
- The userid running the DLA must have authority to issue the DISPLAY GROUP(*) commands.

**Attributes**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.DB2DataSharingGroup</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>(SysplexGroup) Name</td>
<td>Data Sharing Group Name, 1-8 chars</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>
</tr>
<tr>
<td>(SysplexGroup) GroupFunction</td>
<td>Literal “DB2DataSharing”</td>
</tr>
<tr>
<td>VersionString</td>
<td>Group level</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>
</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>
</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>
</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>
</tr>
</tbody>
</table>
### CDM Naming Policy

<table>
<thead>
<tr>
<th>CDM Attribute</th>
<th>Attribute</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<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>
<tr>
<td>Label</td>
<td>&lt;name&gt;-&lt;sysplex &gt;</td>
<td></td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;name&gt;-&lt;sysplex&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-DB2DataSharingGroup</td>
<td></td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;name&gt;-&lt;sysplex&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-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>Sysplex</td>
<td>DB2DataSharingGroup</td>
<td></td>
</tr>
<tr>
<td>federates</td>
<td>DB2DataSharingGroup</td>
<td>DB2Subsystem</td>
<td></td>
</tr>
<tr>
<td>contains</td>
<td>DB2DataSharingGroup</td>
<td>Db2Database</td>
<td></td>
</tr>
</tbody>
</table>
Db2Database

DB2 databases.

Book members

DB2 Subsystem book

Requirements for discovery

- Filtering enabled. By default this is SUPPRESSED since there may be many instances.
- DB2 subsystem and related address spaces must be active and fully initialized.
- The userid running the DLA must have authority to issue the DB/2 DISPLAY DB(*) command.

Attributes

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.db.Db2Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the database</td>
</tr>
<tr>
<td>Label</td>
<td>For DataSharing: &lt;name&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;subsystem&gt;.</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;label&gt;- Db2Database</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;label&gt;- Db2Database</td>
</tr>
</tbody>
</table>

Naming rules

Name and contains relationship from DB2DataSharingGroup or DB2Subsystem (if no 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>DB2Subsystem</td>
<td>Db2Database</td>
<td></td>
</tr>
<tr>
<td>contains</td>
<td>DB2DataSharingGroup</td>
<td>Db2Database</td>
<td></td>
</tr>
<tr>
<td>contains</td>
<td>Db2Database</td>
<td>Db2Tablespace</td>
<td></td>
</tr>
</tbody>
</table>
Db2TableSpace

Db2TableSpace

DB2 tablespaces. By default this is SUPPRESSED since there may be many instances.

Book members

DB2 Subsystem book

Requirements for discovery

- Filtering enabled.
- DB2 subsystem and related address spaces must be active and fully initialized.
- The userid running the DLA must have authority to issue the DB/2 DISPLAY DB(*) command.

Attributes

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.db.Db2TableSpace</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the tablespace</td>
</tr>
<tr>
<td>Label</td>
<td>For DataSharing: &lt;name&gt;-&lt;database&gt;-&lt;data sharing group&gt;-&lt;sysplex&gt;. For not shared: &lt;name&gt;-&lt;database&gt;-&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 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>Db2Database</td>
<td>Db2TableSpace</td>
<td></td>
</tr>
</tbody>
</table>
**IMSSubsystem**

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.

**Book members**

IMS Subsystem book has all attributes and all relationships except to AddressSpaces.

ZOSTASK book has naming rules and relationships to AddressSpaces.

**Requirements for discovery**

- 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.
- IMS subsystem and related address spaces must be active and fully initialized.
- Attributes/Relationships marked with an * are only discovered when the DLA runs in authorized mode.
- No configurations datasets are read for this discovery, that is, no special dataset authority required.

**Attributes**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.IMSSubsystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>(Subsystem) SubsystemName</td>
<td>The 1-4 character name of the IMS subsystem.</td>
</tr>
<tr>
<td>(Subsystem) VersionString</td>
<td>The version and release of the IMS control region</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>
</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>
</tr>
<tr>
<td>CDM Naming Policy</td>
<td>sys.zOS.IMSSubsystem</td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>CDM Attribute</strong></td>
<td><strong>Attribute</strong></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>
</tr>
<tr>
<td></td>
<td>• DCCTL has only the Transaction Manager component and provides access to the IMS message queues for IMS applications.</td>
</tr>
<tr>
<td></td>
<td>• DBCTL has only the Database Manager component and provides IMS database functions.</td>
</tr>
<tr>
<td></td>
<td>• DB/DC provides Transaction Manager and Database Manager functions for large application systems.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></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>
</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>
</tr>
<tr>
<td>IMS PlexGroupName</td>
<td>An IMSPLEX group name represents group of IMS components (subsystems) that share either data and/or message queues. All IMSPLEX members (for example, OM, RM, IMS, CQS) that are in the same IMSPLEX sharing group that share either databases or message queues must specify the same identifier.</td>
</tr>
<tr>
<td>TransactionsChecksum</td>
<td>Checksum for all transactions defined to the IMS.</td>
</tr>
<tr>
<td>ProgramsChecksum</td>
<td>Checksum for all programs defined to the IMS.</td>
</tr>
<tr>
<td>DatabasesChecksum</td>
<td>Checksum for all databases defined to theIMS.</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;-IMSSubsystem</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;SubsystemName&gt;-&lt;SMFID&gt;-IMSSubsystem</td>
</tr>
</tbody>
</table>
**Naming rules**

SubsystemName and hostedDependency relationship to zOS.

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

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

#### Book members

ZOSTASK and subsystem books where applicable.

#### Requirements for discovery

- Filtering enabled (which is the default).
- DLA runs in APF authorized mode.

#### Attributes

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.IMSSysplexGroup</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CDM Attribute</strong></td>
<td><strong>Attribute</strong></td>
<td></td>
</tr>
<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.</td>
<td>IMSplex</td>
</tr>
<tr>
<td></td>
<td>The IMSPlex function shares IMS data objects between IMS subsystems.</td>
<td>IMS CQS</td>
</tr>
<tr>
<td></td>
<td>The IMS CQS function shares IMS message queues objects between IMS subsystems. The IMSCQS group can be used by multiple IMS subsystems but an IMS subsystem can only use one IMSCQS group.</td>
<td>IMS IRLM</td>
</tr>
<tr>
<td></td>
<td>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></td>
</tr>
<tr>
<td>Label</td>
<td><code>&lt;Name&gt;-&lt;SysplexName&gt;</code></td>
<td>CSLIMN0-SYSPLEXO</td>
</tr>
<tr>
<td>id (IdML)</td>
<td><code>&lt;Name&gt;-&lt;SysplexName&gt;-IMSSysplexGroup</code></td>
<td>CSLIMN0-SYSPLEXO-SYSPLEXGROUP</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>Sysplex</td>
<td>IMSSysplexGroup</td>
<td></td>
</tr>
<tr>
<td>memberOf</td>
<td>AddressSpace</td>
<td>IMSSysplexGroup</td>
<td></td>
</tr>
<tr>
<td>federates</td>
<td>IMSSysplexGroup</td>
<td>IMSSubsystem</td>
<td></td>
</tr>
</tbody>
</table>

CDM Naming Policy | sys.zOS.IMSSysplexGroup

<table>
<thead>
<tr>
<th>CDM Attribute</th>
<th>Attribute</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>sourceToken</td>
<td>&lt;Name&gt;-&lt;SysplexName&gt;-IMSSysplexGroup</td>
<td>CSLIMN0-SYSPLEX0-SYSPLEXGROUP</td>
</tr>
</tbody>
</table>
IMSTransaction

IMSTransaction

IMS Transactions.

Book members

IMS Subsystem book

Requirements for discovery

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

Attributes

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.IMSTransaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the transaction</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;-IMSTransaction</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;label&gt;-IMSTransaction</td>
</tr>
</tbody>
</table>

Naming rules

Name and contains relationship from 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>IMSTransaction</td>
<td></td>
</tr>
</tbody>
</table>
**IMSProgram**

IMS Programs.

**Book members**

IMS Subsystem book

**Requirements for discovery**

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

**Attributes**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.IMSProgram</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the program</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 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>IMSProgram</td>
<td></td>
</tr>
</tbody>
</table>
**IMSDatabase**

**Book members**

IMS Subsystem book

**Requirements for discovery**

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

**Attributes**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.IMSDatabase</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the database</td>
</tr>
</tbody>
</table>
## CDM Naming Policy

<table>
<thead>
<tr>
<th>CDM Attribute</th>
<th>Attribute</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<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>

## Naming rules

Name and contains relationship from 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>
MQSubsystem

MQSubsystems consist of a Queue Manager address space and a MQ Channel Initiator address space running on a z/OS system.

Book members
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.

Requirements for discovery
- The userid 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.

Attributes

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.MQSubsystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>(Subsystem) Subsystemname</td>
<td>The 1-4 character name of the MQ subsystem.</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>
</tr>
<tr>
<td>(Subsystem) ControllingAddressSpace</td>
<td>The Jobname of the Queue Manager address space. This identifies the address space to support personnel.</td>
</tr>
<tr>
<td>(Subsystem) VersionString</td>
<td>MQ Version Release and modification level.</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 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>MQSubsystem</td>
<td>zOS</td>
<td>Part of primary naming rule a long with SubsystemName attribute.</td>
</tr>
<tr>
<td>federates</td>
<td>MQSubsystem</td>
<td>AddressSpace</td>
<td>For all related master address space and channel initiator address space.</td>
</tr>
<tr>
<td>uses</td>
<td>IMSSubsystem</td>
<td>MQSubsystem</td>
<td>Discovered via IMS</td>
</tr>
<tr>
<td>uses</td>
<td>CICSRegion</td>
<td>MQSubsystem</td>
<td>Discovered via CICS</td>
</tr>
</tbody>
</table>
| accessedVia          | MQSubsystem    | BindAddress      | For all related federated address spaces that are listening on a TCP and UDP port.
|                      |                |                  | This relationships is used by the DLA sensor to stitch relationships with distributed resources. |
| federates            | MQSubsystem    | MQSenderChannel  |                                                                         |
| federates            | MQSubsystem    | MQRocketChannel  |                                                                         |
MQSenderChannel

MQ Sender Channel

Attributes

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.messaging.mq.MQSenderChannel</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>Name</td>
<td>Channel name</td>
</tr>
<tr>
<td>ConnectionName</td>
<td>Channel connection name</td>
</tr>
<tr>
<td>QueueSharingGroupDisposition</td>
<td>Queue Disposition</td>
</tr>
<tr>
<td>Description</td>
<td>Description</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 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>MQSubsystem</td>
<td>MQSenderChannel</td>
<td></td>
</tr>
</tbody>
</table>
MQReceiverChannel
MQ Receiver Channel

Attributes

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.messaging.mq.MQReceiverChannel</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>Name</td>
<td>Channel name</td>
</tr>
<tr>
<td>ConnectionName</td>
<td>Channel connection name</td>
</tr>
<tr>
<td>QueueSharingGroupDisposition</td>
<td>Queue Disposition</td>
</tr>
<tr>
<td>Description</td>
<td>Description</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 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>MQSubsystem</td>
<td>MQReceiverChannel</td>
<td></td>
</tr>
</tbody>
</table>

Appendix B. z/OS DLA data model class types represented in CDM
CICSRegion

A CICS region is an address space that runs the CICS Transaction Server program.

**Book members**

CICS Region book has all attributes and all relationships except AddressSpaces.

ZOSTASK book has naming rules and relationships to AddressSpaces.

**Requirements for discovery**

- 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 CICSRegion class is only discovered when the DLA runs in APF authorized mode.

**Attributes**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>Attribute</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>JobName (AddressSpace)</td>
<td>Refer to the AddressSpace class description.</td>
</tr>
<tr>
<td></td>
<td>JobUserId (AddressSpace)</td>
<td>Refer to the AddressSpace class description.</td>
</tr>
<tr>
<td></td>
<td>JobType (AddressSpace)</td>
<td>Refer to the AddressSpace class description.</td>
</tr>
<tr>
<td></td>
<td>JobFunction (AddressSpace)</td>
<td>Refer to the AddressSpace class description.</td>
</tr>
<tr>
<td></td>
<td>ProcStep (AddressSpace)</td>
<td>Refer to the AddressSpace class description.</td>
</tr>
<tr>
<td></td>
<td>StepName (AddressSpace)</td>
<td>Refer to the AddressSpace class description.</td>
</tr>
<tr>
<td></td>
<td>Pgm (AddressSpace)</td>
<td>Refer to the AddressSpace class description.</td>
</tr>
<tr>
<td></td>
<td>PgmParm (AddressSpace)</td>
<td>Refer to the AddressSpace class description.</td>
</tr>
<tr>
<td></td>
<td>NetID</td>
<td>SNA netid</td>
</tr>
<tr>
<td></td>
<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>
</tr>
<tr>
<td></td>
<td>VersionString</td>
<td>The version and release of the CICS program.</td>
</tr>
<tr>
<td></td>
<td>SYSIDNT</td>
<td>CICS System Identification</td>
</tr>
<tr>
<td></td>
<td>GRName</td>
<td>CICS Generic Resource Name</td>
</tr>
</tbody>
</table>
### Naming Policy

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.CICSRegion</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>SIT</td>
<td>CICS System Initialization Table parameters. The list is in a table format with two columns: ParameterName and ParameterValue.</td>
</tr>
<tr>
<td>SITOOverrides</td>
<td>CICS System Initialization Table Override parameters. The list is in a table format with two columns: ParameterName and ParameterValue.</td>
</tr>
<tr>
<td>TransactionsChecksum</td>
<td>Checksum of all transaction defined to the CICSRegion.</td>
</tr>
<tr>
<td>ProgramsChecksum</td>
<td>Checksum of all programs defined to the CICSRegion</td>
</tr>
<tr>
<td>FilesChecksum</td>
<td>Checksum of all programs defined to the CICSRegion.</td>
</tr>
<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>CICSRegion</td>
<td>z/OS</td>
<td>Part of primary naming rule along with JobName attribute.</td>
</tr>
<tr>
<td>Uses</td>
<td>CICSRegion</td>
<td>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>CICSRegion</td>
<td>IMSSubsystem</td>
<td></td>
</tr>
<tr>
<td>Uses</td>
<td>CICSRegion</td>
<td>DB2Subsystem</td>
<td></td>
</tr>
<tr>
<td>Uses</td>
<td>CICSRegion</td>
<td>MQSubsystem</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>CICSRegion</td>
<td>TCPPort</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>CICSRegion</td>
<td>UDPPort</td>
<td></td>
</tr>
</tbody>
</table>
This relationships is used by the DLA sensor to stitch relationships with distributed resources.

<table>
<thead>
<tr>
<th>accessedVia</th>
<th>CICSRegion</th>
<th>BindAddress</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains</td>
<td>CICSRegion</td>
<td>CICSTransaction</td>
</tr>
<tr>
<td>Contains</td>
<td>CICSRegion</td>
<td>CICSProgram</td>
</tr>
<tr>
<td>Contains</td>
<td>CICSRegion</td>
<td>CICSFile</td>
</tr>
</tbody>
</table>
CICSTransaction

CICS Transactions.

Book members

CICS Region book

Requirements for discovery

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

Attributes

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.CICSTransaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the transaction</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>CICSRegion</td>
<td>CICSTransaction</td>
<td></td>
</tr>
</tbody>
</table>
CICSProgram

CICS Programs.

**Book members**

CICS Region book

**Requirements for discovery**

- Filtering enabled. By default this is SUPPRESSED since there may be many instances. Note that the CICSRegion attribute ProgramsChecksum enables CMDB to detect Program changes without needing to populate many CICSProgram instances.
- IMS subsystem and related address spaces must be active and fully initialized.
- The DLA must be running in APF authorized mode.

**Attributes**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.CICSProgram</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the program</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 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>CICSRegion</td>
<td>CICSProgram</td>
<td></td>
</tr>
</tbody>
</table>
CICSFile

CICS Files.

Book members
CICS Region book

Requirements for discovery
- Filtering enabled. By default this is SUPPRESSED since there may be many instances. Note that the CICSRegion attribute FilesChecksum enables CMDB to detect File changes without needing to populate many CICSFile instances.
- IMS subsystem and related address spaces must be active and fully initialized.
- The DLA must be running in APF authorized mode.

Attributes

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>sys.zOS.CICSFile</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</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>sourceToken (IdML)</td>
<td>&lt;label&gt;- CICSFile</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>CICSRegion</td>
<td>CICSFile</td>
<td></td>
</tr>
</tbody>
</table>
**WebSphereServer**

A WebSphere Application Server (WAS) is an address space that runs the WAS Control Process. The WebSphereServer class is only discovered when the DLA runs in authorized mode.

**Book members**

WAS book has all attributes and all relationships except AddressSpaces.

ZOSTASK book has naming rules and relationships to AddressSpaces.

**Requirements for discovery**

- Filtering enabled (which is the default).

**Attributes**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.j2ee.websphere.WebSphereServer</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>Name</td>
<td>The Jobname from the corresponding AddressSpace class.</td>
</tr>
<tr>
<td>ProductVersion</td>
<td>Version</td>
</tr>
<tr>
<td>KeyName</td>
<td>AppServer</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>AddressSpace</td>
<td>WebSphereServer</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>WebSphereServer</td>
<td>BindAddress</td>
<td>This relationships is used by the DLA sensor to stitch relationships with distributed resources.</td>
</tr>
<tr>
<td>configuredUsing</td>
<td>WebSphereServer</td>
<td>ConfigFile</td>
<td>$CONFIG_DIR/cells/cell_name/nodes/node_name/servers/server_name/server.xml</td>
</tr>
<tr>
<td>configuredUsing</td>
<td>WebSphereServer</td>
<td>LogicalContent</td>
<td>$CONFIG_DIR/cells/cell_name/nodes/node_name/servers/server_name/*.xml</td>
</tr>
<tr>
<td>Contains</td>
<td>WebSphereServer</td>
<td>AppDescriptors</td>
<td>For Stand alone server $WAS_ROOT_DIR/appdescriptors/*.xml</td>
</tr>
<tr>
<td>Link Type</td>
<td>Source</td>
<td>Target</td>
<td>Comment</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------</td>
<td>---------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>accessedVia</td>
<td>WebSphereServer</td>
<td>WebSphereNamedEndPoint</td>
<td><code>$CONFIG_DIR/cells/cell_name/nodes/node_name/serverindex.xml</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For example, <code>SOAP_CONNECTOR_ADDRESS</code>, <code>BOOTSTRAP_ADDRESS</code></td>
</tr>
</tbody>
</table>
WebSphereCell

A WebSphere Application Server Cell.

Book members

WAS book

Requirements for discovery

- Filtering enabled (which is the default).

Attributes

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.j2ee.websphere.WebSphereCell</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>Name</td>
<td>Cell name</td>
</tr>
<tr>
<td>RootDirectory</td>
<td>Root config directory</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;cell name&gt;-&lt;WAS name&gt;-SMFID</td>
</tr>
<tr>
<td></td>
<td>&gt;-WebSphereCell</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;cell name&gt;-&lt;WAS name&gt;-SMFID</td>
</tr>
<tr>
<td></td>
<td>&gt;-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>WebSphereServer</td>
<td>WebSphereCell</td>
<td></td>
</tr>
<tr>
<td>memberOf</td>
<td>WebSphereNode</td>
<td>WebSphereCell</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>WebSphereCell</td>
<td>BindAddress</td>
<td></td>
</tr>
<tr>
<td>configuredUsing</td>
<td>WebSphereCell</td>
<td>ConfigFile</td>
<td>$CONFIG_DIR/cells/cell_name/cell.xml</td>
</tr>
<tr>
<td>configuredUsing</td>
<td>WebSphereCell</td>
<td>LogicalContent</td>
<td>$CONFIG_DIR/cells/cell_name/*xml</td>
</tr>
<tr>
<td>manages</td>
<td>ComputerSystem</td>
<td>WebSphereCell</td>
<td></td>
</tr>
</tbody>
</table>
WebSphereNode

A WebSphere Application Server Node.

Book members

WAS book

Requirements for discovery

- Filtering enabled (which is the default).

Attributes

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.j2ee.websphere.WebSphereNode</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>Name</td>
<td>Node name</td>
</tr>
<tr>
<td>RootDirectory</td>
<td>Root config directory</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 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>WebSphereNode</td>
<td>WebSphereCell</td>
<td></td>
</tr>
<tr>
<td>configuredUsing</td>
<td>WebSphereNode</td>
<td>ConfigFile</td>
<td>$CONFIG_DIR/cells/cell_name/nodes/node_name node.xml</td>
</tr>
<tr>
<td>configuredUsing</td>
<td>WebSphereNode</td>
<td>LogicalContent</td>
<td>$CONFIG_DIR/cells/cell_name/nodes/node_name/*.xml</td>
</tr>
<tr>
<td>manages</td>
<td>ComputerSystem</td>
<td>WebSphereCell</td>
<td></td>
</tr>
</tbody>
</table>
WebSphereNamedEndPoint

WebSphereNamedEndPoint
A WebSphere Application Server Named EndPoint.

Book members
WAS book

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

Attributes

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.j2ee.websphere.WebSphereNamedEndpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>Name</td>
<td>EndPoint name</td>
</tr>
<tr>
<td>Hostname</td>
<td>&lt;node&gt;:&lt;hostname&gt;</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>WebSphereNamedEndPoint</td>
<td>BindAddress</td>
<td></td>
</tr>
<tr>
<td>accessedVia</td>
<td>WebSphereServer</td>
<td>WebSphereNamedEndPoint</td>
<td></td>
</tr>
</tbody>
</table>
Book members
WAS book

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

Attributes

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.ConfigFile</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>Content</td>
<td>Contents of config file</td>
</tr>
<tr>
<td>URI</td>
<td>content://hostname//filepath</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>
<tbody>
<tr>
<td>configuredUsing</td>
<td>WebSphereCell</td>
<td>ConfigFile</td>
<td>$CONFIG_DIR/cells/cell_name/cell.xml</td>
</tr>
<tr>
<td>configuredUsing</td>
<td>WebSphereNode</td>
<td>ConfigFile</td>
<td>$CONFIG_DIR/cells/cell_name/nodes/node_name node.xml</td>
</tr>
<tr>
<td>configuredUsing</td>
<td>WebSphereServer</td>
<td>ConfigFile</td>
<td>$CONFIG_DIR/cells/cell_name/nodes/servers/server_name/server.xml</td>
</tr>
</tbody>
</table>
LogicalContent

Configuration File

Book members

WAS book

Requirements for discovery

- Filtering enabled (which is the default).

Attributes

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>core.ConfigFile</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>Content</td>
<td>Contents of config file</td>
</tr>
<tr>
<td>URI</td>
<td>collation://hostname// (filepath)</td>
</tr>
<tr>
<td>id (IdML)</td>
<td>&lt;uri&gt;-LogicalContent</td>
</tr>
<tr>
<td>sourceToken (IdML)</td>
<td>&lt;uri&gt;-LogicalContent</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>WebSphereCell</td>
<td>LogicalContent</td>
<td>$CONFIG_DIR/cells/cell_name/*.xml</td>
</tr>
<tr>
<td>configuredUsing</td>
<td>WebSphereNode</td>
<td>LogicalContent</td>
<td>$CONFIG_DIR/cells/cell_name/nodes/node_name/*.xml</td>
</tr>
<tr>
<td>configuredUsing</td>
<td>WebSphereServer</td>
<td>LogicalContent</td>
<td>$CONFIG_DIR/cells/cell_name/nodes/servers/server_name/*.xml</td>
</tr>
<tr>
<td>contains</td>
<td>AppDescriptor</td>
<td>LogicalContent</td>
<td>WebSphereServer AppDescriptor $WAS_ROOT_DIR/appdescriptors/*.xml</td>
</tr>
</tbody>
</table>
**AppDescriptor**

Configuration File

**Book members**

WAS book

**Requirements for discovery**

- Filtering enabled (which is the default).

**Attributes**

<table>
<thead>
<tr>
<th>CDM Naming Policy</th>
<th>app.AppDescriptor</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDM Attribute</td>
<td>Attribute</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>WebSphereServer</td>
<td>AppDescriptor</td>
<td>$WAS_ROOT_DIR/appdescriptors/*.xml</td>
</tr>
<tr>
<td>contains</td>
<td>AppDescriptor</td>
<td>LogicalContent</td>
<td>WebSphereServer AppDescriptor $WAS_ROOT_DIR/appdescriptors/*.xml</td>
</tr>
</tbody>
</table>
Appendix C. Samples provided in hlq.SIZDSAMP

The following samples are provided in hlq.SIZDSAMP.

Member in SIZDSAMP
Description

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.

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 $IZDPDISC, as part of the normal post IPL start up process to pick up any system configuration changes.

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.

IZDJRUN
Batch job JCL to run the DLA.
• This invokes the IZDPDISC proc with an override on the output dataset name since batch jobs don’t allow system symbols.
• This enables users to perform discovery without issuing a console command.

IZDJFTP
Batch job JCL to run the FTP only.
• This invokes the IZDPFTP proc with an override on the output dataset name since batch jobs don’t allow system symbols.

This enables users to perform FTP without running discovery.

IZDCFTPI
Initial FTP statements used to PUT members to the Discovery Library File Store. See later section for details

IZDCDEF
Default DLA configuration parameter deck in IZDJDISC.
This configuration parameter deck is coded to cause discovery of hardware and z/OS resources including IPL, PARMLIB and DASD volume attributes, CICS, IMS and DB/2 discovery with resources such as transactions, programs and databases limited to 100 resources per CICS, IMS or DB/2 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, DB/2, WebSphere and MQ subsystem that are discovered) that are named using the jobname or STC name of the discovered subsystem.

**IZDCBASE**
DLA configuration parameter deck to discover ZOSBASE only.

**IZDCTASK**
DLA configuration parameter deck to discover ZOSTASK only.

**IZDCIMS**
DLA configuration parameter deck to discover IMS subsystems, including all attributes.

**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.

**IZDCCCICS**
DLA configuration parameter deck to discover CICS regions, including all attributes. Additional non-CICS information is only discovery 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.

**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.

**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.
Appendix D. Installation checklist

__ 1. SMPE install the z/OS DLA datasets. This process is detailed in the z/OS DLA Program Directory.
__ 2. Schedule a roll out 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 IPL process to APF authorize the SIZDLOAD dataset 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 SIZDPDISC JCL PROC to a system PROCLIB dataset on each z/OS image (Optional for PoC phase).
__ 4. DB2 customization as required for all DB2 subsystems that are the target of discovery:
   __ a. z/OS DLA uses the DSNREXX DB/2 REXX interface to issue SQL commands from REXX. The DSNREXX interface must be bound to each DB/2 subsystem where discovery is required. Refer to DB/2 Installation, Step 18: Bind the packages for DB2 REXX Language Support: DSNITJRX for details.
   __ b. The z/OS DLA userid must have authority to issue DB/2 DISPLAY commands (including DISPLAY DB and DISPLAY GROUP) via the DB2 Instrumentation Facility Interface (IFI).
   __ c. The z/OS DLA userid must have authority to READ SYSIBM tables. Authorization may be via RACF or DB/2 GRANT depending on DB/2 installation setup. (Optional—only required if DB/2 SQL discovery is required.)
   __ d. If multiple versions of DB/2 are installed on the operating system then a STEPLIB (or LINKLST entry) matching the DB/2 version of the discovery target must be supplied for the discovery job.
__ 5. Define security access as required:
   __ a. The userid that runs the z/OS DLA job or STC must have the following RACF authorities:
      __ 1) READ access is recommended for all the data sets in the system PARMLIB concatenation used during IPL.
      __ 2) READ access is required to profiles in the MQCMDs class (if active) to allow the z/OS DLA to issue MQ DISPLAY commands via the MQ command interface.
      __ 3) If RACF is used to protect DB/2 resources then authority is required to issue DB/2 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 userid that runs the discovery job must have an OMVS segment with authority to issue the netstat, host and home commands.
Installation checklist
Glossary

CEC. Central electronics complex

CIM. Common Information Model

CMDB. Configuration Management Database

**Coupling Facility.** A special logical partition that provides high-speed caching, list processing, and locking functions in a sysplex.

MSU. Millions of Service Units

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.

**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.

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
Mail Station P300
522 South Road
Poughkeepsie New York 12601-5400
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply 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 may 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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

**Trademarks**

The following are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

- CICS
- IBM
- Lotus
- OMEGAMON
- VTAM
- z/VM
- DB2
- IMS
- MQSeries
- RACF
- WebSphere
- zSeries
- FICON
- Language Environment
- MVS
- Tivoli
- z/OS

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Windows is a trademark of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.
Index

A
AddressSpace 71
APF authorization 4
AppDescriptor 111

B
BindAddress 76
BOOK directive 35
books
  output 9

C
CCMDB
  loading with XML files 23
CDM
  z/OS DLA data model class types represented in 49
checksums
to detect members needing transfer 21
CICS
  supported by DLA 1
CICS region
discovery 16
CICSFile 103
CICSProgram 102
CICSRegion 98
CICSTransaction 101
codepage 5
codes 48
ComputerSystem 58
ConfigFile 109
customization 3
  overview 3

D
DB/2
  customization 4
  supported by DLA 1
DB2 subsystem
discovery 12
DB2Database 83
DB2DataSharingGroup 81
DB2Subsystem 79
Db2TableSpace 84
DISC directive 31
discovery
  automating 21
Discovery Library
  file naming standards 20
Discovery Library File Store
  ftp transfer to 19
DLA
  running 7
DLA FTP utility 22
DLFS FTP server 19
documents
  related 2

F
file naming standards
  Discovery Library 20
FILT directive 33
filtering directives
  BOOK 35
  DISC 31
  FILT 33
  SET 25
Fqdn 61
FTP utility 22

G
glossary 117

H
hardware
  supported by DLA 2
hlq.SIZDSAMP
  samples provided in 113

I
IdML XML viewer 21
IMS
  supported by DLA 1
  IMS subsystem
discovery 14
IMSDatabase 92
IMSProgram 91
IMSSubsystem 85
IMSSysplexGroup 88
IMSTransaction 90
install
  SMP/E 3
  installation 3
  overview 3
  installation checklist 115
introduction 1
IpInterface 59
IPv4Address 60
IZDJFTP JCL batch job 9
IZDJRUN JCL batch job 7
IZDPDISC JCL proc 8
IZDPFTP JCL proc 9
IZDRDLA
  return codes 48

J
JCL
  customization 5

L
license inquiry 119
LogicalContent 110
LPAR 54

M
ManagementSoftwareSystem 50
members
  checksummed to detect need for transfer 21
messages 37
MQ
  supported by DLA 1
MQ subsystem
discovery 15
MQReceiverChannel 97
MQSenderChannel 96
MQSubsystem 94

O
Organization 51
output books 9
overview
  of installation and customization 3
  of product 1

R
RACF
  authorization 4
  reference 25
related documents 2
return codes
  from IZDRDLA 48
  other 48

S
samples
  provided in hlq.SIZDSAMP 113
SET directive 25
SMP/E
  install 3
  target datasets 3
software
  supported by DLA 1
StorageSubSystem 77
StorageVolume 78
supported hardware 2
supported software 1
Sysplex 69
SystemSpecificCollection 70
target DLFS FTP server 19
TcpPort 74
Tivoli CCMDB
  loading with XML files 23
trademarks 120

UdpPort 75
user guide 7
user reference 25

WebSphere
  supported by DLA 1
WebSphere application server
    discovery 17
WebSphereCell 106
WebSphereNamedEndPoint 108
WebSphereNode 107
WebSphereServer 104

XML files
  loading into Tivoli CCMDB 23

z/OS 62
 supported by DLA 1
z/OS DLA
  filtering 25
z/OS DLA data model class types
  represented in CDM 49
z/OS DLA FTP utility 22
ZOSALL book 18
ZOSBASE
  discovery 11
ZOSTASK
  discovery 12
ZSeriesComputerSystem 52
ZVM 56
ZVMGuest 57
Readers’ Comments — We’d Like to Hear from You

IBM Tivoli Discovery Library Adapter for z/OS
User’s Guide and Reference
Version 2 Release 1

Publication No. SC23-6521-00

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