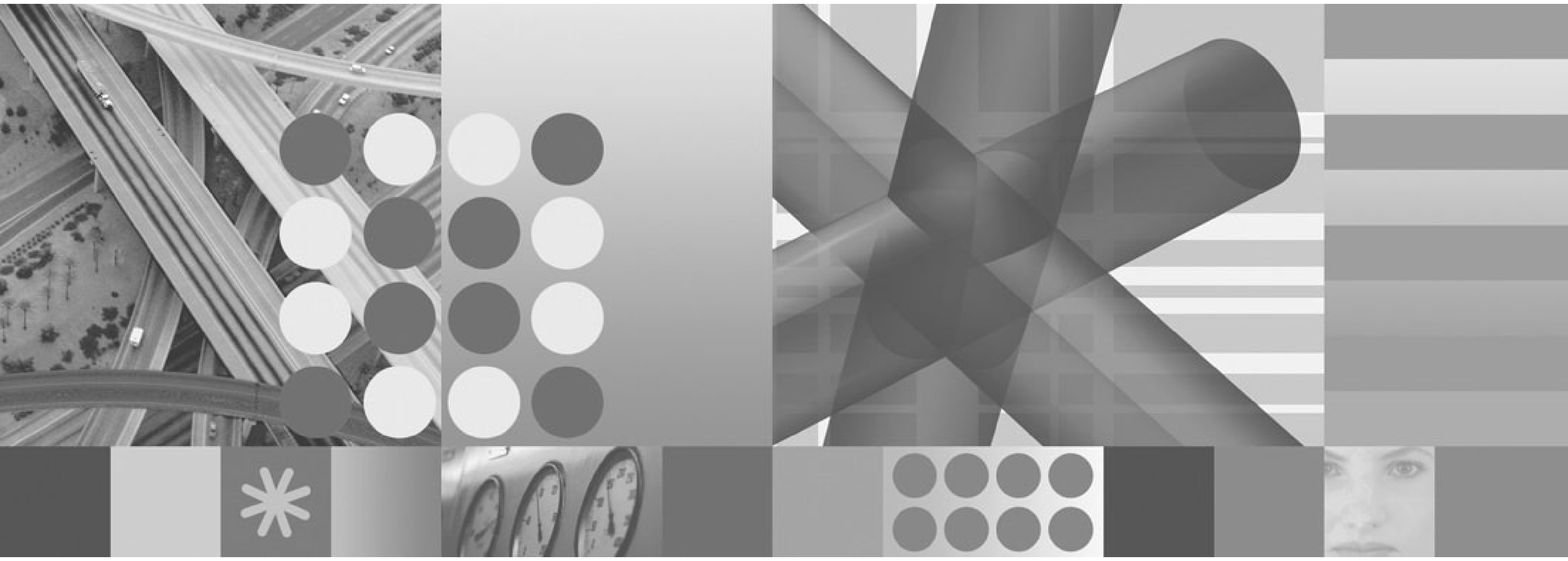
Tivoli. IBM Tivoli Change and Configuration Management Database

Version 1.1 1

Report Authoring Tool User's Guide

Tivoli® IBM Tivoli Change and Configuration Management Database

Version 1.1 1



Report Authoring Tool User's Guide

Note

Before using this information and the product it supports, read the information in Appendix B, “Notices,” on page 49.

First Edition (September 2006)

This edition applies to version 1, release 1.1 of IBM Tivoli Change and Configuration Management Database and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 1. About this guide

This guide provides information and step-by-step instructions for creating, managing, and customizing IBM Tivoli Change and Configuration Management Database reports.

For information about IBM Tivoli Change and Configuration Management Database, see http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/index.jsp.

Audience

The target audience for this document includes the following users:

Report authors and software developers
This includes users who want to create and customize IBM Tivoli Change and Configuration Management Database reports.

System administrators
This includes people who need to install and configure the Report Authoring Tool, manage connections to the reporting Web service, and manage data source connections and instances.

Database administrators and database analysts
This includes people familiar with the database schema for the underlying business data.

This guide addresses the following needs:

- Introduces important concepts related to the Report Authoring Tool
- Describes how to install and use the Report Authoring Tool
- Explains how to connect to the reporting Web service and manage data source connections and instances
- Describes how to manage report definitions, including how to create, edit, import, and export report definitions
- Describes how to manage relationship maps

Prerequisites

To use the Report Authoring Tool, users should have working knowledge of the following tools and technologies:

- Eclipse 3.0, Eclipse 3.1, or the Rational® Software Architect 6.0.2.5 or higher
- Extensible Markup Language (XML) (to manually edit report definitions)
- Database concepts, including schemas, tables, columns, and views

Chapter 2. Getting started with the Report Authoring Tool

You can use the Report Authoring Tool to create, manage, and customize IBM Tivoli Change and Configuration Management Database reports. The Eclipse-based tool enables you to do the following:

- Manage dynamic relationship maps that describe physical data sources in independent logical database terms
- Manage report definitions that describe the data to query, the organization of the data in the report, and the output format of the report
- Preview reports and view the XML source for report definitions

This chapter describes the following topics:

- "System requirements"
- "Installing the Report Authoring Tool"
- "Understanding Reporting Authoring Tool components and concepts" on page 5
- "Using the Report Authoring Tool" on page 6
- "Understanding Domain Discovery Server and Enterprise Discovery Server Synchronization" on page 8

System requirements

The Report Authoring Tool has the following system requirements:

- Eclipse 3.0, Eclipse 3.1, or Rational Software Architect 6.0.2.5 or higher
- IBM® Java™ Runtime Environment (JRE) 1.4.2 or later
- IBM DB2® V8.2 Client
- IBM DB2 JDBC driver JAR files

Important: The Report Authoring Tool does not support Eclipse 3.2.

Installing the Report Authoring Tool

You can use the Eclipse or Rational Software Architect (RSA) Install/Update wizard to install the Report Authoring Tool.

To install the Report Authoring Tool, complete the following steps:

1. Copy the ReportAuthoringTool.jar file from the distribution media to a temporary directory on your machine.
2. Perform a standalone install of WebSphere® Application Client v6.0.2.3 or later to the C:\IBM\WebSphere\AppClient directory.

   The remainder of this procedure refers to this directory as %APP_CLIENT_HOME%.
3. Launch your Eclipse-based environment.
4. Choose **Window** → **Preferences**.
5. In the Installed JREs tab, verify that IBM JRE 1.4.2 or later is selected.
6. Choose **Help** → **Software Updates** → **Find and Install**.
7. Using the Install/Update wizard, complete the following steps:
   a. Choose **Search for new features to install** and click **Next**.

b. Click **New Archived Site**, navigate to the location of the ReportAuthoringTool.jar file, and click **OK**.

c. In the Edit Local Site dialog, click **OK**.

d. In the Sites to include in search list, select the site named ReportAuthoringTool.jar and click **Finish**. A Search Results dialog appears.

e. Select all components that appear under the ReportAuthoringTool.jar feature, and click **Next**.

f. Select **I accept** and click **Next**.

g. Click **Finish**. A feature verification warning window appears.

h. Click **Install All**.

i. After the installation is complete, shut down Eclipse or RSA.

8. Copy the JAR files to the Eclipse plugins directory by completing the following steps:

   a. Create the %ECLIPSE_HOME%\plugins\com.ibm.websphere.appclient.v6_1.0.0.1\lib directory.

   b. Copy the following JAR files from %APP_CLIENT_HOME%\lib to %ECLIPSE_HOME%\plugins\com.ibm.websphere.appclient.v6_1.0.0.1\lib:

Table 1. JAR files to copy

Files				
j2ee.jar	webservices.jar	ffdc.jar	bootstrap.jar	ras.jar
wsexception.jar	wsdl4j.jar	lmproxy.jar	urlprotocols.jar	admin.jar
emf.jar	commons-logging-api.jar	commons-discovery.jar	channelfw.jar	channel.http.jar
channel.tcp.jar	utils.jar	runtime.jar	j2cClient.jar	sas.jar
was-wssecurity.jar	webservices-security.jar	workspace.jar	wsdeployenv.jar	wssec.jar
xmlsecurity.jar	ecutils.jar	security.jar	sib.common.jar	soap.jar
soap-sec.jar	wsatlib.jar	iwsorb.jar	activation-impl.jar	mail-impl.jar

9. Restart Eclipse or RSA.

Upgrading the Report Authoring Tool

You can upgrade the Report Authoring Tool from version 1.1 to version 1.1.1.

To upgrade the Report Authoring Tool, complete the following steps:

1. Launch your Eclipse-based environment.
2. Choose **Window** → **Close Perspective** to close the CCMDB Reporting perspective, if it is open.

   Note that you need to re-create the Web service connections after an update.
3. Choose **Help** → **Software** → **Updates** → **Find and Install**.
4. Using the Install/Update wizard, complete the following steps:

   a. Choose **Search for new features to install** and click **Next**.

   b. Click **New Archived Site**, navigate to the location of version 1.1.1. of the ReportAuthoringTool.jar file, and click **OK**.

   c. In the Edit Local Site dialog, click **OK**.

d. In the Sites to include in search list, select the site named ReportAuthoringTool.jar and click **Finish**. A Search Results dialog appears.

e. Select all components that appear under the ReportAuthoringTool.jar feature, and click **Next**.

f. Select **I accept** and click **Next**.

g. Click **Finish**. A feature verification warning window appears.

h. Click **Install All**.

5. Restart Eclipse or RSA.

Understanding Reporting Authoring Tool components and concepts

The Report Authoring Tool consists of client and server-based components that provide facilities for creating, customizing, and managing IBM Tivoli Change and Configuration Management Database reports.

Figure 1 shows the principal components of the Report Authoring Tool.

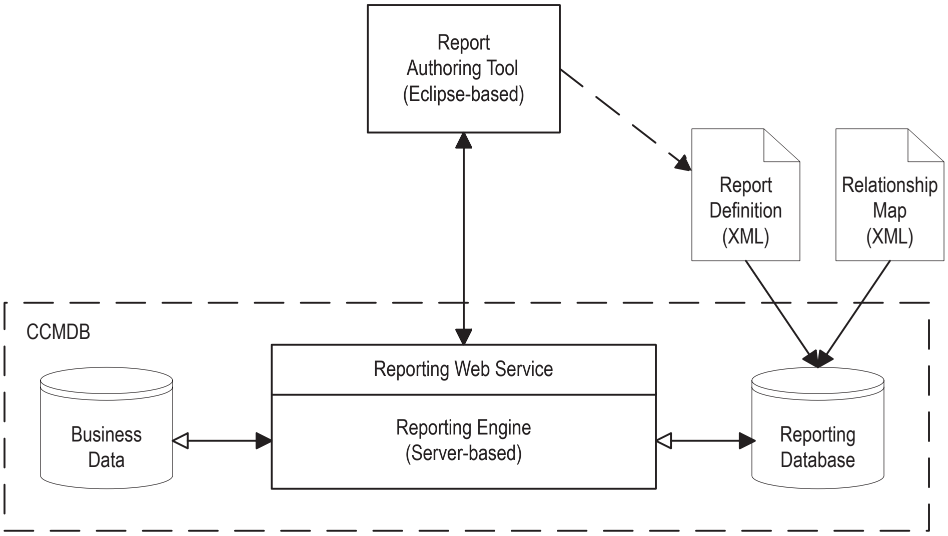


Figure 1. Overview of the Report Authoring Tool

The following describes the components shown in Figure 1.

Report Authoring Tool
The Report Authoring Tool is an Eclipse-based graphical user interface application installed on a client workstation that provides facilities for creating and managing report definitions, managing relationship maps, defining data source connections and instances, and establishing connections to the Report Authoring Tool Web service.

Reporting Web Service
The Reporting Web service is a server-based interface that enables the Report Authoring Tool client application to access the reporting engine. The Reporting Web service is installed as part of the IBM Process Manager Integration Platform (PMIP).

IBM Tivoli Change and Configuration Management Database (CCMDB)
The IBM Tivoli Change and Configuration Management Database is an enterprise-ready configuration management database and platform that serves as the foundation of IBM Tivoli® Process Managers.

Report Definition

Report definitions provide information to the reporting engine describing how to generate specific reports based on information stored in the IBM Tivoli Change and Configuration Management Database and the IBM Process Database.

Report definitions specify the information that appears in the report, the data to query, how the data should be constrained and aggregated, and details about the output format of the report. Report definitions are stored as XML-formatted files.

Relationship Map

Relationship maps contain relationships that describe the logical entities and attributes for a data source and how these logical objects are related to the physical counterparts in the physical data source (RDBMS). Relationship maps are stored as XML-formatted files.

Reporting Database

The reporting database, part of the IBM Process Database, is the repository in which report definitions, relationship maps, and related information used by the reporting engine is stored.

Using the Report Authoring Tool

The Report Authoring Tool enables you to create and customize IBM Tivoli Change and Configuration Management Database reports. You can use the Report Authoring Tool with accounts assigned the following roles:

Report Administrator

Report administrators can perform all functions available in the Report Authoring Tool

Report Author

Report authors can edit definitions, and view all settings and definitions

To use the Report Authoring Tool, complete the following steps:

1. Launch your Eclipse-based environment.
2. Open the CCMDB Reporting perspective by completing the following steps:.
   a. Choose **Window** → **Open Perspective** → **Other** from the main menu. The Select Perspective Dialog appears.
   b. Select CCMDB Reporting and click **OK**.

   The system displays the following views associated with the CCMDB Reporting perspective.

   Session Resource Browser View

   Displays the reporting artifacts on the reporting server

   Data Source Connections View

   Used to create, delete, or modify connection information for sources of data such as an RDBMS

   Data Source Instances View

   Used to create, delete, or modify instances that represent sources of data

   JDBC Driver Management View

   Used to specify the JDBC driver library jar files and the JDBC driver class names that the Report Authoring Tool should use.

Report Definition Management View
Used to create, delete, modify, import, and export report definitions

Relationship Map Management View
Used to manage schema information that describes the entities in a data source

Web Service Connections View
Used to manage instances of Web service sessions to a reporting web service.

Understanding the workflow

The typical workflow for using the Report Authoring Tool includes connecting to the reporting Web service, defining data source connections and instances, and defining report definitions and relationship maps.

Figure 2 shows an overview of the workflow when using the Report Authoring Tool.

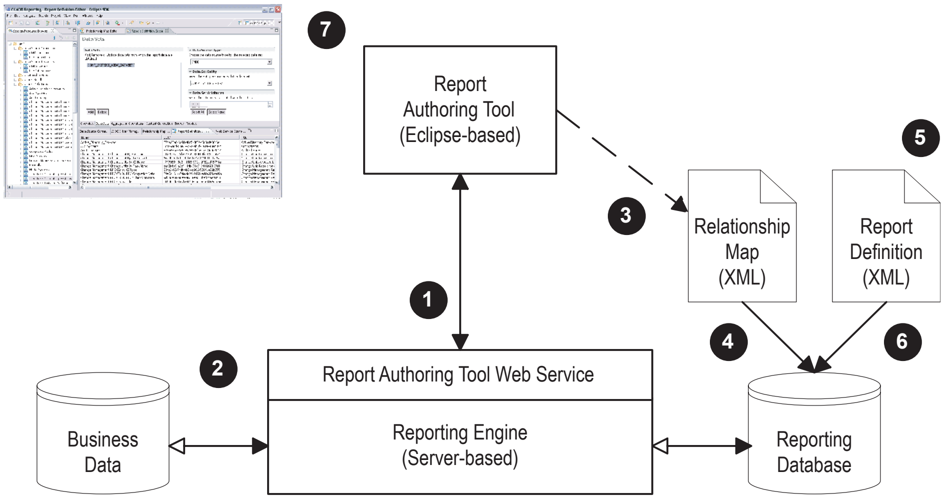

7
Report Authoring Tool (Eclipse-based)
5
3
Relationship Map (XML)
Report Definition (XML)
1
4
6
2
Report Authoring Tool Web Service
Reporting Engine (Server-based)
Business Data
Reporting Database

Figure 2. Report Authoring Tool workflow

The following describes the workflow shown in Figure 2.

1. Connect to the Report Authoring Tool service to enable access to the reporting engine and reporting database, which serves as a repository for report definitions and relationship maps.

   See “Connecting to the reporting service” on page 9 for more information.
2. Define a connection to a data source, such as a relational database management system, and create a data source instance to represent a specific physical data source.

   See “Managing data source connections” on page 11 and “Managing data source instances” on page 11 for more information.
3. Create a relationship map on the local file system.

   See “Creating a new relationship map” on page 15 for more information.
4. Import the relationship map to the reporting server.

   See “Importing a relationship map” on page 18 for more information.
5. Create a report definition on the local file system.

See “Creating a new report definition” on page 22 for more information.

6. Import the report definition to the reporting server.

   See “Importing a report definition” on page 26 for more information.

7. Preview the report, and modify the report definition and relationship maps as required.

   See “Managing report definitions” on page 21 and “Managing relationship maps” on page 14 for more information.

Understanding Domain Discovery Server and Enterprise Discovery Server Synchronization

The Enterprise Discovery Server periodically synchronizes with the Domain Discovery Server domains to which it is connected. During synchronization, a subset of the attributes are copied from the Domain Discovery Server domains to the Enterprise Discovery Server. Specifically, key attributes are copied as well as any attributes required to support the reporting views. That way, reports can always be generated from the data contained in the Enterprise Discovery Server.

The Enterprise Discovery Server maintains a text file that contains the tables and attributes to copy during synchronization. If you have added additional attributes to a report, and those attributes are currently not being synchronized from the Domain Discovery Server domains to the Enterprise Discovery Server, you need to add the new attributes to the text file of attributes to copy. The new attributes are then copied from the Domain Discovery Servers to the Enterprise Discovery Server during the next synchronization, and are available for reports.

Chapter 3. Managing connections and data sources

The Report Authoring Tool is a client application that enables access to the reporting Web service. The reporting service maintains data such as report definitions, relationship maps, data source definitions, and data source instances. Before you can create report definitions and define relationship maps, you need to connect to the reporting Web service and define the data source connections and instances that the reports and relationship maps use.

Figure 3. illustrates the typical workflow for managing connections and data sources.

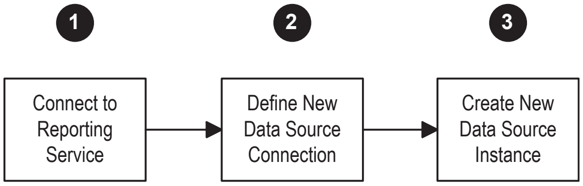

1
2
3
Connect to Reporting Service
Define New Data Source Connection
Create New Data Source Instance

Figure 3. Managing connections and data sources workflow

The following describes the sequence shown in Figure 3.

1. Connect to the reporting server to enable access to report entities such as report definitions and relationship maps.
2. Define a connection to a data source, such as a relational database management system.
3. Create a data source instance to represent a specific physical data source.

This chapter describes how to manage connections and data sources, and contains the following sections:

- "Connecting to the reporting service"
- "Managing data source connections" on page 11
- "Managing data source instances" on page 11

Connecting to the reporting service

The Report Authoring Tool serves as a client that can access the reporting Web service. To begin working with reporting entities on the reporting server, such as report definitions and relationship maps, you must first establish a connection to the reporting Web service.

To connect to the reporting service, complete the following steps:

1. Click the **Web Service Connections** tab.
2. To define a new Web service connection, complete the following steps:
   a. Right-click in the Web Service Connections table and choose **New** from the context-sensitive menu. The Web Service Connection Configuration dialog appears.
   b. Specify values for the following fields:

      Name The user-specified name of the Web service connection

Description

A user-supplied description of the Web service connection

System Information Port

The system information port provides functions that enable viewing diagnostic information about the reporting server; for example, http://ccmdbserver.company.com:9080/ReportingService/services/SystemInformation. You can also establish an https-based secure connection using the following URL: https://ccmdbserver.company.com:9443/ReportingService/services/SystemInformation.

Reporting Port

The reporting port provides functions that enable running reports and listing available report definitions and report instances. The reporting port also provides functions to determine the available entities for a data source and the attributes available for an entity; for example, http://ccmdbserver.company.com:9080/ReportingService/services/Reporting. You can also establish an https-based secure connection using the following URL: https://ccmdbserver.company.com:9443/ReportingService/services/Reporting.

Reporting Administration Port

The reporting administration port provides functions for installing and deleting report definitions and relationship maps, scheduling reports to run, and performing data source connection and data source instance operations; for example, http://ccmdbserver.company.com:9080/ReportingService/services/ReportingAdministration. You can also establish an https-based secure connection using the following URL: https://ccmdbserver.company.com:9443/ReportingService/services/ReportingAdministration.

c. Click **OK**. The entry appears in the Web Service Connections pane.

3. Right-click an entry in the Web Service Connections table and choose Connect from the context-sensitive menu.

4. Type a user name and password for a user with the ReportAdministrator role, such as wasadmin. The system prompts you for a user name and password for the system information port, reporting port, and reporting administration port.

   The Status column changes to CONNECTED and the Session Resource Browser View on the left displays a tree listing the entities installed on the reporting server.

   Important: You should be prompted for three sets of user names and passwords corresponding to the system information port, reporting port, and reporting administration port. If you are only prompted for a single user name and password, verify the URL setting for the Web service port configuration is correct. If the Web service port settings are correct, restart Eclipse and try connecting to the reporting service again.

Managing data source connections

You can use the Data Source Connections view to display and manage data connections to sources such as an RDBMS.

To manage data source connections, complete the following:

1. Click the Data Source Connections tab.
2. To define a new data source connection, complete the following steps:
   a. Right-click in the Data Source Connections table and choose **New** from the context-sensitive menu. The Data Source Connection Properties dialog appears.
   b. Type values for the following fields:

      Name The name of the data source connection.

      Type The type of data source connection, currently jdbc.

      Description
      A description of the connection.

      jdbcDriverClassName
      The JDBC driver class to use, for example, com.ibm.db2.jcc.DB2Driver.

      jdbcConnectionUrl
      The server and database information for the connection, for example, jdbc:db2://db2server:db2port/DBNAME.

      jdbcUsername
      The database user name.

      jdbcPassword
      The password associated with the database user.

   c. Click **OK**.
3. To modify an existing data source connection, right-click an item in the Data Source Connections table and choose **Modify** from the context-sensitive menu.
4. To delete a data source connection, right-click an item in the Data Source Connections table and choose **Delete** from the context-sensitive menu.

Managing data source instances

A data source instance represents a physical data source such as an RDBMS, and contains information about how the physical schema relates to the logical entities and attributes used within the reporting engine. You can use the Data Source Instances view to manage the data source instances defined on the reporting server.

To manage data source instances, complete the following steps:

1. Click the Data Source Instances tab.
2. To create a new data source instance, complete the following steps:
   a. Right-click in the Data Source Instances table and choose **New** from the context-sensitive menu. The Data Source Instance Properties dialog appears.
   b. Type values for the following fields:

      Name The name of the data source instance.

Type The name of the data source instance implementation class, from among the following values: `com.ibm.tivoli.prism.ccmdb.reporting.impl.alphablox.sql.pdb.DynamicAlphabloxSQLViewPDBDataSource` for PDB or `com.ibm.tivoli.prism.ccmdb.reporting.impl.alphablox.sql.cmdb.DynamicAlphabloxSQLViewCMDBDataSource` for CMDB.

Description
A description of the instance.

Connection Name
The name of an existing data source connection.

c. Click **OK**.

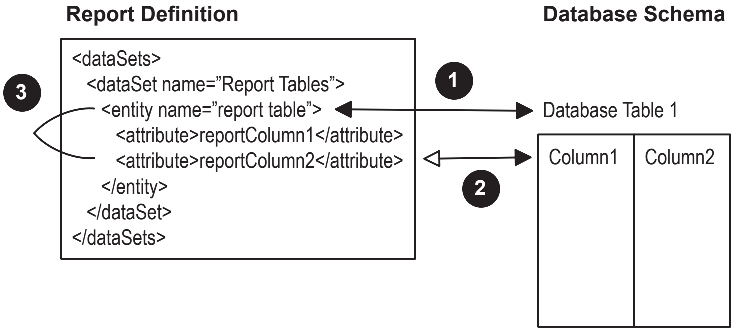
3. To modify an existing data source instance, right-click an item in the Data Source Instances table and choose **Modify** from the context-sensitive menu.
4. To delete a data source instance, right-click an item in the Data Source Instances table and choose **Delete** from the context-sensitive menu.

Chapter 4. Using relationship maps

Relationship maps contain relationships that describe the logical entities and attributes for a data source and how these logical objects are related to the physical counterparts in the physical data source (RDBMS).

The maps are XML-based files that are installed on the reporting server, and the entities and attributes defined in those maps can then be accessed through a data source instance. Since relationship maps describe the data source schema, you need to create and install them before working with report definitions.

Figure 4 illustrates the principal relationships that you can define using the Report Authoring Tool.



```
<dataSets>
  <dataSet name="Report Tables">
    <entity name="report table">
      <attribute>reportColumn1</attribute>
      <attribute>reportColumn2</attribute>
    </entity>
  </dataSet>
</dataSets>
```

Figure 4. Report definition and database relationships

The following describes the relationships shown in Figure 4.

1. Table to entity map
Maps an SQL Table (view) to an entity. This defines a logical entity that serves as a proxy to a physical table or view in an SQL-based database The entity is an abstract table object used to describe the report table.

2. Column to attribute map
Maps a column in the database table to an attribute. This defines an attribute that serves as a proxy to a physical column in a SQL based database. The attribute is a meta-element against which operations are applied.

3. Attribute to entity map
Maps an attribute to an entity. This is a compositional relationship that associates a logical attribute with a logical entity.

This chapter describes how to manage relationship maps using the Report Authoring Tool, and contains the following topics:

- "Managing relationship maps" on page 14
- "Creating a new relationship map" on page 15
- "Viewing and editing relationship maps" on page 15
- "Importing a relationship map" on page 18
- "Exporting a relationship map" on page 18

Loading the JDBC drivers

The relationship map editor uses JDBC calls to connect to a database when generating the relationship maps. You must therefore load the JDBC drivers using the Report Authoring Tool before using the relationship map editor.

To load the JDBC drivers, complete the following steps:

1. Click the JDBC Driver Management tab.
2. Click **Add**.
3. Specify the location of the db2jcc.jar and db2jcc_license_cu.jar files from your DB2 client installation.
4. In the JDBC driver class name field, type com.ibm.db2.jcc.DB2Driver.
5. Click **Load**. A dialog appears indicating that the drivers were successfully loaded.

Managing relationship maps

You can use the Relationship Map Management view to display the installed relationship maps on the reporting server, create a new relationship map, and edit existing relationship maps.

You can also use the view to import or export relationship map files between the local file system and the reporting server and delete a relationship map stored on the reporting server.

To manage relationship maps, complete the following steps:

1. Click the **Relationship Map Management** tab.
2. To create a new relationship map, right-click in the Relationship Map Management table and choose **New** from the context-sensitive menu.

   See "Creating a new relationship map" on page 15 for more information.
3. To import a relationship map, right-click in the Relationship Map Management table and choose **Import** from the context-sensitive menu.

   See "Importing a relationship map" on page 18 for more information.
4. To export a relationship map, right-click a relationship map in the Relationship Map Management table and choose **Export** from the context-sensitive menu.

   See "Exporting a relationship map" on page 18 for more information.
5. To view or edit a relationship map, right-click a relationship map in the Relationship Map Management table and choose **Edit** from the context-sensitive menu.

   See "Viewing and editing relationship maps" on page 15 for more information.
6. To view or edit a relationship map on the local file system, right-click in the Relationship Map Management table and choose **Open From Local File** from the context-sensitive menu.

   This enables you to view and edit a relationship map using the XML-formatted source. Click **File** → **Save As** to save your changes. Note that when viewing or editing a relationship map on the local file system, you cannot use the Map/DB editor since the editor requires information about the data source connections and instances from the reporting server.

7. To delete a relationship map from the reporting server, right-click a relationship map in the Relationship Map Management table and choose **Delete** from the context-sensitive menu.

Creating a new relationship map

You can create a new relationship map that describes how the logical entities and attributes for a data source are related to the physical counterparts in the physical data source (RDBMS)

Important: You need to load the JDBC drivers using the Report Authoring Tool before creating a new relationship map. See "Loading the JDBC drivers" on page 14 for more information.

To create a new relationship map, complete the following steps:

1. Click the **Relationship Map Management** tab.
2. Right-click in the Relationship Map Management table and choose **New** from the context-sensitive menu. A dialog appears.
3. Specify an existing directory name for the container.
4. Specify a filename with a .xml suffix.

   This is the filename on the local file system used to store the relationship map.
5. Click **Finish**.
6. Specify the name of the relationship map.

   This is the name that appears in the Report Authoring Tool user interface. This name is stored in the relationship map XML file using the `name` attribute.
7. Save the relationship map.
8. Install the relationship map on the reporting server using the import wizard.

   See "Importing a relationship map" on page 18 for more information.

Viewing and editing relationship maps

You can use the relationship map editor to display and edit the contents of relationship maps installed on the reporting server.

The relationship map editor consists of the Map/DB editor and the source editor. The Map/DB editor uses the following panels to display the contents of a relationship map:

Relationship map panel (left)
The relationship map panel shows a tree-based representation of the relationship map, and includes the relationship map name, the data source descriptor, the data elements defined in the map, and the relationships that connect the data elements to the physical database table and column names.

Database view panel (right)
The database view panel shows the database schema, including the physical data source details that are associated to the relationship map items.

The source editor displays the relationship map file in XML format.

Important: You must load the JDBC drivers using the Report Authoring Tool before using the relationship map editor. See "Loading the JDBC drivers" on page 14 for more information.

To view and edit a relationship map, complete the following steps:

1. Click the **Relationship Map Management** tab.
2. Right-click a relationship map and choose **Edit** from the context-sensitive menu. The Map/DB editor panel appears displaying the contents of the relationship map.
3. To edit the name of the relationship map, click the root element of the tree in the relationship map panel (left) and type a new name.

   You can reference each installed relationship map using either a name or using a system generated UUID. It is recommended that you assign a string value for the name that accurately describes the contents of the relationship map. A single relationship map can contain all relationships for an entire schema or can contain the relationships that describe a single table or view.
4. To create an SQL-table-to-entity relationship, which defines a logical entity that serves as a proxy to a physical table or view in an SQL-based database, complete the following steps:
   a. In the relationship map panel, right-click the **Relationships** node and choose **New SQLTable::Entity Relationship** from the context-sensitive menu
   b. Specify the following relationship information in the dialog that appears:

      Relationship Name
      The name of the relationship.

      Relationship Type
      The type of relationship: SQLTableEntityRelationship. You cannot modify this field.

      Entity Name for Relationship
      The name of the entity. An entity is a logical representation or proxy in the reporting namespace for a physical database table or view in the RDBMS namespace.

      Data Source Connection
      An object that contains the details instructing the reporting server how to connect to a specific physical database.

      SQL Table to Link as Entity
      The physical table or the view. The convention is to use the schema name as a prefix, for example, PDB.ACCOUNT for the ACCOUNT table in the PDB schema.
   c. Click **OK**.

   Alternatively, you can create an SQL-table-to-entity relationship using the database view panel by completing the following steps:
   a. Right-click a table name in the database view panel and choose **Map TABLE as ENTITY (Shallow)** from the context-sensitive menu. A new relationship appears in the relationship map panel with a system-generated name.
   b. Right-click the new relationship and choose **Modify** from the context-sensitive menu.
   c. Edit the details for the new relationship using the dialog that appears and click **OK**.

5. To create an SQL-column-to-attribute relationship, which defines an attribute that serves as a proxy to a physical column in a SQL based database, complete the following steps:
   a. In the relationship map panel, right-click the **Relationships** node and choose **New SQLColumn::Attribute Relationship** from the context-sensitive menu
   b. Specify the following relationship information in the dialog that appears:

      Relationship Name
      The name of the relationship.

      Relationship Type
      The type of relationship: SQLColumnAttributeRelationship. You cannot modify this field.

      Attribute Name for Relationship
      The name of the attribute. An attribute is a logical representation or proxy in the reporting namespace for a physical database column in the RDBMS namespace.

      Entity to Bind With
      The name of the logical entity to which this attribute belongs.

      Data Source Connection
      An object that contains the details instructing the reporting server how to connect to a specific physical database. See "Managing data source connections" on page 11 for more information.

      SQL Table Name
      The name of the table that contains the column to which this attribute corresponds. This should be the same table that is associated with the logical entity to which the attribute belongs.

      SQL Column Name
      The physical database column name to which the attribute corresponds.

      SQL Column Type

      The SQL Column Type can be **STRING**, **INTEGER**, or **TIMESTAMP**.
   c. Click **OK**.

   Alternatively, you can create an SQL-column-to-attribute relationship using the database view panel by completing the following steps:
   a. In the database view panel, right-click a column name and choose **Map COLUMN as ATTRIBUTE** from the context-sensitive menu. A new relationship appears in the relationship map panel with a system-generated name.
   b. Right-click the new relationship and choose **Modify** from the context-sensitive menu.
   c. Edit the details for the new relationship using the dialog that appears and click **OK**.

   You can also create a new relationship containing all columns automatically by completing the following steps:
   a. In the database view panel, right-click a table name and choose **Map TABLE as ENTITY/ATTRIBUTES (Deep)** from the context-sensitive menu. A new relationship appears in the relationship map panel with a system-generated name.

A new relationship of the appropriate type for the SQL table/view with all columns is created and appears in the relationship map panel.

   b. Right-click the new relationship and choose **Modify** from the context-sensitive menu.

   c. Edit the details for the new relationship using the dialog that appears and click **OK**.

6. To create an attribute-to-entity relationship, which is a compositional relationship that associates a logical attribute with a logical entity, complete the following steps:

   a. In the relationship map panel, right-click the **Relationships** node and choose **New Attribute::Entity Relationship** from the context-sensitive menu

   b. Specify the following relationship information in the dialog that appears:

      Relationship Name
      The name of the relationship.

      Relationship Type
      The type of relationship: SQLAttributeEntityRelationship. You cannot modify this field.

      Entity Name to Bind With
      The name of the logical entity to use as the parent or owner of the attribute.

      Attribute Name to Bind With
      The name of the logical attribute to associate with the logical entity.

   c. Click **OK**.

   Note: An attribute-to-entity relationship is analogous to the relationship an SQL column has with the table or view that contains that column.

7. Click the **Source** tab to display the relationship map file in XML format. You can edit the relationship map manually, if required.

Importing a relationship map

You can import a relationship map on a local file system to the reporting server.

To import a relationship map, complete the following steps:

1. In the Relationship Map Management view, right-click and choose **Import** from the context-sensitive menu. A dialog appears.
2. Select an existing data source onto which the relationships contained in the relationship map should be installed.
3. Type the full path name of the relationship map file.
4. Click **Finish**. The Relationship Map Management view refreshes and the new relationship map is displayed.

Exporting a relationship map

You can export a relationship map installed on the reporting server to an XML format file on the local file system.

To export a relationship map, complete the following steps:

1. In the Relationship Map Management view, right-click a relationship map and choose **Export** from the context-sensitive menu. A dialog appears.

2. Choose an existing relationship map.
3. Type the full path name of the relationship map file.
4. Click **Finish**. The relationship map is written to the local file system

Chapter 5. Using report definitions

Report definitions provide information to the reporting engine about how to generate specific reports. The definitions specify the information that appears in the report, the data to query, how the data should be constrained or manipulated, and details about the output format of the report.

This section describes the following topics:

- "Managing report definitions"
- "Creating a new report definition" on page 22
- "Editing a report definition" on page 22
- "Specifying the report definition overview" on page 23
- "Specifying the report definition data sets" on page 24
- "Specifying aggregation operations" on page 24
- "Importing a report definition" on page 26
- "Exporting a report definition" on page 26

Managing report definitions

You can use the Report Definition Management tab to display the installed report definitions on the reporting server, create a new report definition, and edit existing report definitions. You can also use the view to import or export report definition files between the local file system and the reporting server and delete a report definition stored on the reporting server.

To manage report definitions, complete the following steps:

1. Click the **Report Definition Management** tab.
2. To create a new report definition, right-click in the Report Definition Management table and choose **New** from the context-sensitive menu.

   See "Creating a new report definition" on page 22 for more information.
3. To view or edit a report definition, right-click a report definition in the Report Definition Management table and choose **Edit** from the context-sensitive menu.

   See "Editing a report definition" on page 22 for more information.
4. To view or edit a report definition on the local file system, right-click in the Report Definition Management table and choose **Open From Local File** from the context-sensitive menu.

   This enables you to edit aspects of a report definition, such as the title, header, and footer, without connecting to the reporting server. You can also edit the report definition using the XML-formatted source. Note that the **Data Source Connections**, the **Data Source Instances**, and the **Preview** tab are not available in this mode because a connection to the reporting server is required to use these features.
5. To import a report definition from a local file, right-click in the Report Definition Management table and choose **Import** from the context-sensitive menu.

   See "Importing a report definition" on page 26 for more information.

6. To export a report definition to a local file, right-click a report definition in the Report Definition Management table and choose **Export** from the context-sensitive menu.

   See “Exporting a report definition” on page 26 for more information.
7. To delete a report definition from the reporting server, right-click a report definition in the Report Definition Management table and choose **Delete** from the context-sensitive menu.

Creating a new report definition

You can create a new report definition that specifies how to generate a report.

To create a new report definition, complete the following steps:

1. Click the **Report Definition Management** tab.
2. Right-click in the Report Definition Management table and choose **New** from the context-sensitive menu. A dialog appears.
3. Specify an existing directory name for the container.
4. Specify a file name with a .xml suffix.
5. Click **Finish**. An overview page appears.

   Note that the report definition editor does not display the **Data Set** or **Preview** tabs. These tabs become available after you install the report definition on the reporting server.
6. Specify the required information for the overview page.

   See “Specifying the report definition overview” on page 23 for more information.
7. Save the report definition.
8. Install the report definition on the reporting server using the import wizard.

   See “Importing a report definition” on page 26 for more information.

Editing a report definition

You can specify general overview information, data sets, aggregation operations, and the format of the report when editing a report definition. You can also view the XML formatted source of the report definition, and display an HTML-based preview of the report.

Note that the report definition editor does not display the **Data Sets** or **Preview** tabs when you are editing a report definition on the local file system. It is recommended that you import the report definition after creating it, and then use the procedure described in this section to perform edits. For more information, see “Importing a report definition” on page 26.

To edit a report definition, complete the following steps:

1. Click the **Report Definition Management** tab.
2. Right-click a report definition in the Report Definition Management table and choose **Edit** from the context-sensitive menu.
3. Click the **Overview** tab to specify the general information for the definition.

   See “Specifying the report definition overview” on page 23 for more information.
4. Click the **Data Sets** tab to define the report definition data sets.

See “Specifying the report definition data sets” on page 24 for more information.

5. Click the **Aggregation Operations** tab to define the aggregation operations for the report definition.

   See “Specifying aggregation operations” on page 24 for more information.
6. Click the **Content Generation** tab to select the output format of the report.

   You can generate report output in HTML, PDF and CSV formats.
7. Click the **Source** tab to display the report definition in XML format.

   You can use the source view to manually edit the report definition, if required.
8. Click the **Preview** tab and click **Run Preview** to display the HTML version of the report definition.

   You need to choose **HTML** in the **Content Generation** tab to view an HTML preview of the report definition.

Specifying the report definition overview

You can specify general information about the report definition, including the report name and description. You can also specify information that should appear with the generated data such as the report title, subtitle, category, header, and footer text.

To specify the report definition overview, complete the following steps:

1. Click the **Overview** tab in the report definition editor.

   You can use the Overview tab to modify general attributes of the report definition.
2. Specify the following information in the Overview section:

   Report Definition Name
   The name of the report definition.

   Description
   A description of the report. This description is not displayed in the final report.
3. Specify the following information in the Descriptor section:

   Title The title to display at the top of the report.

   Subtitle
   The subtitle to display in a smaller font directly below the title.

   Category
   The category of the report, displayed in the category column of the portal user interface.

   Header
   Text to display in a smaller font directly below the subtitle.

   Footer Text to display at the bottom of the report.
4. Optionally, add traits to the report definition

   Traits are user-defined name-value pairs that describe aspects of the report definition.

Specifying the report definition data sets

Relationship maps define relationships between attributes and SQL columns and entities and SQL tables and views. Relationship maps also define compositional relationships between an entity and an attribute, where an entity is composed of multiple attributes just as a table is composed of multiple columns.

Report definitions use these entities and attributes described in the relationships. A report definition contains an entity and one or more attributes of that entity.

Data sets describe the logical entities and attributes that are queried to obtain the data for the report. Each data set contains one logical entity and multiple attributes. The entity names are provided by an installed data source instance of a specific type.

To specify the report definition data sets, complete the following steps:

1. Click the **Data Sets** tab in the report definition editor.
2. Click **Add** and type a data set name.
3. Select the data set in the list.
4. Choose a data source type using the **Data Source Type** drop-down list. The entities associated with the data source instance appear in the **Data Set Entity** section.
5. Choose an entity from the **Data Set Entity** drop-down list. The attributes associated with the entity appear in the **Data Set Attributes** list.
6. Select the attributes that are to appear in the final report.

   The reporting engine supports a single data set in the report definition.

Specifying aggregation operations

Aggregation operations specify the operations that the reporting engine is to perform on the data sets defined in the report definition. Using aggregation operations, you can query, sort, and group data in your reports.

The operations work sequentially with the output of one operation serving as the input to the next operation. You should specify the first operation to be a query, which you can optionally follow with sort and group operations.

To specify aggregation operations, complete the following steps:

1. Click the **Aggregation Operations** tab in the report definition editor.
2. Click **Add** under the Aggregation Operations list. A dialog appears.
3. Type the name of the operation in the dialog, and click **OK**. The new operation appears in the Aggregation Operations list.
4. Select the operation in the list.
5. Choose an operation type in the **Aggregation Operation Type** section.
6. Select the data sets that the aggregation operations are to process.

   You can define data sets using the **Data Sets** tab, as described in “Specifying the report definition data sets.”
7. If the operation type is **query**, the operation can simply query all data from the data set. However, if the data should be constrained in some way, you can add a data set constraint as a query parameter by completing the following steps:
   a. In the Query Parameters section, click **Add**.

b. In the DataSet Constraint Detail dialog, name the constraint, choose an entity, and define a constraint expression.

   The constraint expression limits the possible values that should be included in the result set.

c. Specify a default value

   The default value is assigned to the constraint expression variable when the report is run. You can override the default value by specifying a parameter, such as myConstraint=5, when running the report.

d. Select **Optional Constraint** to have the system consider this constraint optional.

   If a report definition contains a non-optional constraint, you need to enter a value for the constraint when running the report using the portal. When running the report using the RPClientCLI, the constraint is ignored if it is non-optional and no parameter value is specified.

e. Select **Include constrained data in the result** to control whether the constrained data should be included in the final result set, or whether it should be filtered out from the result set.

8. If the operation type is **group**, you can specify that group-based aggregation operations be performed on the queried data by completing the following steps:

   a. In the Group Parameters section, click **Add**.

   b. In the Group Parameter Detail dialog, choose the parameter type.

      If this is the first grouping parameter, the parameter type is locked to **Group**.

   c. Select the attributes to be used for grouping. You can adjust the order of the selected group member attributes using the **Move Up** and **Move Down** buttons. The order is important because the data is grouped in the order listed.

      After a grouping parameter is created, a **GroupMemberList** item appears in the Group Parameters listbox.

   d. Press **Add** to add other group parameters such as Sum, Max, Min, Average, and Count.

      These grouping aggregation operations are applied to the smallest subgroup after the data has been grouped. For example, assuming that the data has already been grouped by type and then by manufacturer, adding a Sum operation for the CPU count attribute displays a sum of all the processors for all computers that are grouped under the same type and manufacturer. Each subgroup displays its own sum value.

9. If the operation type is **sort**, you can specify the queried data to be sorted by one or more attributes by completing the following steps:

   a. In the Sort Parameters section, select one or more attributes from the Available Sort Attributes list. The selected attributes appear in the Selected Sort Attributes list.

   b. Use the Move Up and Move Down buttons to control which attribute should be used to sort the data in order.

   c. For each item in the Selected Sort Attributes list, select a sort order of ASCENDING or DESCENDING.

      The default is ASCENDING. When using the sort and group operations together in the same report definition, apply a sort to the query data first and then apply the group operation.

Importing a report definition

You can use the import wizard to install a report definition on the local file system to the reporting server.

To import a report definition, complete the following steps:

1. In the Report Definition Management view, right-click and choose **Import** from the context-sensitive menu. A dialog appears.
2. Choose a report definition type (text/xml) using the **Report Def. Type** drop-down list.
3. Specify the file name of an existing report definition file.
4. Click **Finish**. The imported definition appears in the Report Definition Management view.

   The report definition can now be run by the reporting engine

Exporting a report definition

You can export a report definition that is installed on the reporting server to an XML format file on the local file system.

To export a report definition, complete the following steps:

1. In the Report Definition Management view, right-click a definition and choose **Export** in the context-sensitive menu. The Export Report Definition dialog appears.
2. Choose the definition name using the **Report Definition** drop-down list.
3. Choose the export file type (text/xml) using the **Report Def. Type** drop-down list.
4. Specify the file to which the report definition is to be exported.
5. Click **Finish**. The report definition is saved to the specified file.

Chapter 6. Troubleshooting the Report Authoring Tool

You can troubleshoot problems with the Report Authoring Tool by configuring the logging properties for the application and by checking the Eclipse error log.

To troubleshoot the Report Authoring Tool, complete the following steps:

1. Configure the logging.properties file to enable tracing.

   See "Enabling logging" for more information.
2. Choose **Window** → **Show View** → **Other** → **PDE Runtime** → **Error Log** to display the Eclipse Error Log View.

Enabling logging

You can enable logging for the Reporting Authoring Tool to help you determine the cause of potential error conditions.

To enable logging, complete the following steps:

1. Open the logging.properties file using a text editor.

   You can find the logging.properties file in the lib directory of the JRE that is currently active. Note that modifying this file will change the logging defaults for all Java applications that use this JRE. Alternatively, you can create a private copy of the logging.properties file for use only with the Report Authoring Tool.
2. Enable file logging by removing the comment symbol (#) from the following line:

   ```
   #handlers=java.util.logging.FileHandler,java.util.logging.ConsoleHandler
   ```
3. Configure the logging level by modifying the following line:

   ```
   .level = INFO
   ```

   You can set the logging level to one of the following: `OFF`, `SEVERE`, `WARNING`, `INFO`, `CONFIG`, `FINE`, `FINER`, `FINEST`, `ALL`. While the level `ALL` offers the most granular level of logging, a setting of `FINE` is typically sufficient.
4. Optionally, configure the location and name of the log file by modifying the following line:

   ```
   java.util.logging.FileHandler.pattern = %h/java%u.log
   ```

   Change %h/java%u.log to the required directory and file name The default location of the log file is the home directory of your user account.
5. Optionally, configure the system to produce plain text output for the log file by modifying the following line:

   ```
   java.util.logging.FileHandler.formatter = java.util.logging.XMLFormatter
   ```

   Replace `XMLFormatter` with `SimpleFormatter`.
6. Save the logging.properties file

If using a private copy of the logging.properties file, start Eclipse or the Rational Software Architect using the following command:

```
C:\<path>\eclipse\eclipse -VM c:\<path>\jre1.4.2\jre\bin\javaw -vmargs
   -Xms256M -Djava.util.logging.config.file=c:\<path>\logging.properties
```

Chapter 7. Uninstalling the Report Authoring Tool

You can uninstall the Report Authoring Tool, if necessary, using Eclipse or the Rational Software Architect.

To uninstall the Report Authoring Tool, complete the following steps:

1. Launch your Eclipse-based environment.
2. Choose **Window** → **Close Perspective** to close the CCMDB Reporting perspective, if it is open.
3. Choose **Help** → **Software Updates** → **Manage Configuration**.
4. Select CCMDB Report Authoring DevKit 1.1.1 and click **Disable** to disable the Report Authoring Tool.
5. Select com.ibm.websphere.appclient.v6 1.0.0.1 and click **Disable** to disable the WebSphere Application Client.
6. Restart Eclipse or the Rational Software Architect.
7. Choose **Help** → **Software Updates** → **Manage Configuration**.
8. Verify that **Show Disabled Features** is active, select CCMDB Report Authoring DevKit 1.1.1, and uninstall the feature.

   If you updated the CCMDB Report Authoring DevKit from version 1.1.0 to 1.1.1, select CCMDB Report Authoring DevKit 1.1.0 and uninstall the feature.
9. Select com.ibm.websphere.appclient.v6 1.0.0.1 and uninstall the feature.
10. Restart Eclipse or the Rational Software Architect.

Appendix A. Tables used for reporting

Tables used for reporting

This appendix describes the tables and columns in the Enterprise Discovery Server database that are the basis for the views used by the report definitions delivered with the IBM Tivoli Change and Configuration Management Database.

Database and report administrators can use these tables and columns to create custom views, and then use the Report Authoring Tool to create custom relationship maps and reports definitions for use with the IBM Tivoli Change and Configuration Management Database.

ADMININFO table

This section describes the columns in the ADMININFO table.

Table 2. ADMININFO table

Column	Description
ADMINCONTACT_X	The administrative contact for the configuration item.
OBJGUID_X	The Globally Unique Identifier (GUID) of the object to which this object refers.

APPSRVR table

This section describes the columns in the APPSRVR table.

Table 3. APPSRVR table

Column	Description
DELETED_X	Indicates whether the object is deleted, from among the following values: **1** The object is deleted. **0** The object is not deleted.
GUID	The Globally Unique Identifier (GUID) of the object.
JDOCLASSX	The internal JDO class name of the object.
NAME_X	The name of the application server in a vendor-specific context. For example, for WebSphere, it is the name of the server assigned in the administration console.
PK__CONFIGFILE_X	The configuration file used to configure the application server.
PK__DATABASE_X	The foreign key to the DATABASE table.
PK__EXECUTABLE_X	The executable file used to start the application server process.
PK__HOST_X	The host on which the application server resides.
PK__NODE_X	The foreign key to the WASNODE table.

Table 3. APPSRVR table (continued)

Column	Description
PK__PARENTJ2EESERVER_X	The foreign key to the J2EEDOMN table.
PK__PRIMARYSAP_X	The primary listening port through which the application server can be accessed. This is the GUID of the BINDADDR.
PRODUCTNAME_X	The vendor-specific product name.
PRODUCTVERSION_X	The vendor-specific product version.
SID_X	The SID if this is an Oracle Application Server.
STATUS_X	The status of the application server, from among the following: • Running • Stopped
VENDORNAME_X	The name of application server vendor.

BINDADDR table

This section describes the columns in the BINDADDR table.

Table 4. BINDADDR table

Column	Description
DELETED_X	Indicates whether the object is deleted, from among the following values: **1** The object is deleted. **0** The object is not deleted.
GUID	The Globally Unique Identifier (GUID) of the object.
PK__IPADDRESS_X	The IP address to which the port is bound (for IPADDR_ANY, 0.0.0.0). This is the foreign key into the IPADDR table.
PK__PRIMARYADDRESS_X	The primary IP address of the host on which the port resides. This is the foreign key into the IPADDR table.
PK_X	The primary key.
PORTNUMBER_X	The port number to which the port is bound.
TYPE_X	The type of transport, such as TCP or UDP, among others.

CAPABILITY table

This section describes the columns in the CAPABILITY table.

Table 5. CAPABILITY table

Column	Description
CAPABILITY_NAME_X	The name of the capability.
CMDBSOURCE_X	The CMDB domain that serves as the source.

Table 5. CAPABILITY table (continued)

Column	Description
DELETED_X	Indicates whether the object is deleted, from among the following values: **1** The object is deleted. **0** The object is not deleted.
DESCRIPTION_X	The description of the capability.
DISPLAYNAME_X	The machine-generated, user-friendly display name of the object.
GUID	The Globally Unique Identifier (GUID) of the object.
JDOCLASS_X	The internal JDO class name of the object.
LABEL_X	The user-defined label associated with the object.
LASTMODIFIEDTIME_X	The date and time the object was last modified.
OBJECTTYPE_X	The extended type of the object. (Not used)
PK_X	The primary key.
VERSION_X	The baseline version number of the object.

CHANGEHISTORY table

This section describes the columns in the CHANGEHISTORY table.

Table 6. CHANGEHISTORY table

Column	Description
CLASS_NAME	The class name of the object.
OBJECT_ID	The GUID of the object whose change this represents.
WHEN_CHANGED	When the change history was recorded.

COMPSYS table

This section describes the columns in the COMPSYS table.

Table 7. COMPSYS table

Column	Description
ANSIT10ID_X	A storage identifier for the following objects: • DiskDrive • TapeDrive • StoragePool • TapeLibary • StorageSubSystem (computer system)
ARCHITECTURE_X	The kernel architecture of the host. This has implications for applications that depend on the kernel version.
CDPDEVICEID_X	The Global Device Identifier, used with the Cisco Discovery Protocol.

Table 7. COMPSYS table (continued)

Column	Description
CONTEXTIP_X	Used by the CMDB discovery engine to determine the IP address that was used to discover the specific instance.
CPUSPEED_X	The processor speed in megahertz (MHz).
CPUTYPE_X	The host ISA or processor type.
DELETED_X	Indicates whether the object is deleted, from among the following values: **1** The object is deleted. **0** The object is not deleted.
DESCRIPTION_X	The description of the system.
DISPLAYNAME_X	The machine-generated, user-friendly display name of the object.
FQDN_X	The Fully Qualified Domain Name to which the IP address of this host resolves. This is usually the host name followed by the domain name for the host.
GUID_X	The Globally Unique Identifier (GUID) of the object.
JDOCLASSX	The internal JDO class name of the object.
LABEL_X	The user-defined label associated with the object.
LASTMODIFIEDTIME_X	The date and time the object was last modified.
MANUFACTURER_X	The manufacturer of the host.
MEMORYSIZE_X	The memory size in megabytes (MB).
MODEL_X	The hardware name for the host.
NAME_X	The host name of the computer system.
NUMCPUS_X	The number of effective processors.
PK__OSRUNNING_X	The operating system currently running on the host.
PK__PRIMARYOWNER_X	A relationship that corresponds to, and represents implicitly, the *owns* relationship.
PK_X	The primary key.
RUNID_X	The unique identifier assigned to the discovery run.
SERIALNUMBER_X	The serial number of the host.
SIGNATURE_X	The unique identifier for the host. The signature is typically a combination of the lowest physical interface IP address and the MAC address associated with the interface.

Table 7. COMPSYS table (continued)

Column	Description
SOURCETOKEN_X	A token that is used locally by a management system to identify a particular instance in the CMDB. Each management system can have a different source token that it uses to correlate its data with information about that instance in the CMDB. When this attribute is queried, the CMDB determines the identity of the entity making the request and returns the source token that corresponds to that entity.
SYSTEMBOARDUUID_X	The unique identifier of the system board.
TYPE_X	The type of computer system, which usually reflects the primary function of the device. For example, values can include Bridge, Router, LoadBalancer, ComputerSystem, and IPDevice, among others.
VERSION_X	The baseline version number of the object.
VOLUMEGROUPCAPACITY_X	The storage capacity of the volume group.
VOLUMEGROUPFREESPACE_X	The storage free space of the volume group.
WORLDWIDENAME_X	The Storage Area Network (SAN) World Wide Name (WWN)

COMPUTERSYSTACES_76E84D2AX table

This section describes the columns in the COMPUTERSYSTACES_76E84D2AX table.

Table 8. COMPUTERSYSTACES_76E84D2AX table

Column	Description
PK__JDOIDX	The lookup identifier.
PK_IPINTERFACES_X	The foreign key to the IPINTRFC table

COMPUTERSYSTIONS_6C816818X table

This section describes the columns in the COMPUTERSYSTIONS_6C816818X table.

Table 9. COMPUTERSYSTIONS_6C816818X table

Column	Description
PK__FUNCTIONS_X	The foreign key to the FUNCTION table.
PK__JDOIDX	The lookup identifier.

COMPUTERSYSTLLED_B0C5E01DX table

This section describes the columns in the COMPUTERSYSTLLED_B0C5E01DX table.

Table 10. COMPUTERSYSTLLED_B0C5E01DX table

Column	Description
PK__JDOIDX	The lookup identifier.
PK_OSINSTALLED_X	The foreign key to the OPSYS table

COMPUTERSYSTTEMS_88841D4BX table

This section describes the columns in the COMPUTERSYSTTEMS_88841D4BX table.

Table 11. COMPUTERSYSTTEMS_88841D4BX table

Column	Description
PK__FILESYSTEMS_X	The foreign key to the FILESYS table
PK__JDOIDX	The lookup identifier.

COMPUTERSYSTTEMS_A267EEF0X table

This section describes the columns in the COMPUTERSYSTTEMS_A267EEF0X table.

Table 12. COMPUTERSYSTTEMS_A267EEF0X table

Column	Description
PK__COMPUTERSYSTEMS_X	The foreign key to the COMPSYS table.
PK__JDOIDX	The lookup identifier.

DBTDBASE table

This section describes the columns in the DBTDBASE table.

Table 13. DBTDBASE table

Column	Description
DELETED_X	Indicates whether the object is deleted, from among the following values: **1** The object is deleted. **0** The object is not deleted.
DISPLAYNAME_X	The machine-generated, user-friendly display name of the object.
PK__PARENTDB2DATABASE_X	A relationship that corresponds to, and represents implicitly, the *contains* relationship.

DISCRUN table

This section describes the columns in the DISCRUN table.

Table 14. DISCRUN table

Column	Description
CAUSE_X	The cause of the failure in the case of a failed run.

Table 14. DISCRUN table (continued)

Column	Description
DISCOVERRUNID_X	The unique identifier assigned to the discovery run.
DISCOVERYPROCESSLOG_X	All progress events that occurred during the run.
ENDTIME_X	The time when the run terminated.
RUNNAME_X	The user-supplied name of the discovery.
RUNTYPE_X	The type of the discovery run, from among the following: • New • Refresh Note: All discoveries are now New. The refresh run type is no longer being used, making this field deprecated.
STARTTIME_X	The time when the run started.
STATUS_X	The status of the run, from among the following: • DISCOVERY_STATUS_RUNNING • DISCOVERY_STATUS_NOT_RUNNING
TERMINATIONTYPE_X	The termination of the discovery run, from among the following: • NORMAL_TERMINATION • ABORTED_TERMINATION • FAILURE_TERMINATION

FILESYS table

This section describes the columns in the FILESYS table.

Table 15. FILESYS table

Column	Description
AVAILABLESPACE_X	The available space on the file system (in bytes).
CAPACITY_X	The capacity of the file system (in bytes).
DELETED_X	Indicates whether the object is deleted, from among the following values: **1** The object is deleted. **0** The object is not deleted.
MOUNTPOINT_X	The mount point in the file system.
PK_X	The primary key.
TYPE_X	The type of file system.

FSEXPORT table

This section describes the columns in the FSEXPORT table.

Table 16. FSEXPORT table

Column	Description
DISPLAYNAME_X	The machine-generated, user-friendly display name of the object.
PK__PARENTFILESYSTEMEXPORT_X	The foreign key to the FILESYS table.

FUNCTION table

This section describes the columns in the FUNCTION table.

Table 17. FUNCTION table

Column	Description
DELETED_X	Indicates whether the object is deleted, from among the following values: **1** The object is deleted. **0** The object is not deleted.
DISPLAYNAME_X	The machine-generated, user-friendly display name of the object.
JDOCLASSX	The internal JDO class name of the object.
PK_X	The primary key.
VLANNAME_X	The name of the Virtual LAN (VLAN), if one exists.

INCIDRECPROXY table

This section describes the columns in the INCIDRECPROXY table.

Table 18. INCIDRECPROXY table

Column	Description
DELETED_X	Indicates whether the object is deleted, from among the following values: **1** The object is deleted. **0** The object is not deleted.
DESCRIPTION_X	The description of the object.
DISPLAYNAME_X	The machine-generated, user-friendly display name of the object.
FOREIGNINCIDENTHANDLE_X	The incident handle used in an external (non ITSM) service desk application such as Remedy, Peregrine, or MRO, among others.
FOREIGNINCIDENTID_X	The incident identifier used in an external (non ITSM) service desk application such as Remedy, Peregrine, or MRO, among others.
GUID_X	The Globally Unique Identifier (GUID) of the object.
ITSMINCIDENTHANDLE_X	The internal incident handle in the external owning application. This optional handle (if known) should not be used as a display label.

Table 18. INCIDRECPROXY table (continued)

Column	Description
ITSMINCIDENTID_X	The internal incident identifier in the external owning application. The string can be a number or a combination of letters and numbers used for display purposes.
LABEL_X	The user-defined label associated with the object.
LASTMODIFIEDTIME_X	The date and time the object was last modified.
PK_X	The primary key.
TITLE_X	The incident title or abstract.

IPADDR table

This section describes the columns in the IPADDR table.

Table 19. IPADDR table

Column	Description
ADDRESSSPACE_X	The address space for the IP address (in case of NAT).
DELETED_X	Indicates whether the object is deleted, from among the following values: **1** The object is deleted. **0** The object is not deleted.
DOTNOTATION_X	The IP address in dot notation format (xxx.xxx.xxx.xxx).
PK__DESTINATION_X	The route destination.
PK__NEXTHOP_X	The next hop in the route.
PK_X	The primary key.

IPINTRFC table

This section describes the columns in the IPINTRFC table.

Table 20. IPINTRFC table

Column	Description
DELETED_X	Indicates whether the object is deleted, from among the following values: **1** The object is deleted. **0** The object is not deleted.
PK__IPADDRESS_X	The IP address of the interface, and the foreign key to the IPADDR table.
PK__IPNETWORK_X	The IP network of the interface, and the foreign key to the IPNETWORK table.
PK_X	The primary key.

IPNETWORK table

This section describes the columns in the IPNETWORK table.

Table 21. IPNETWORK table

Column	Description
DELETED_X	Indicates whether the object is deleted, from among the following values: **1** The object is deleted. **0** The object is not deleted.
NAME_X	The assigned name of the network.
NETMASK_X	The subnet mask for the network.
PK_X	The primary key.
RUNID_X	The unique identifier assigned to the discovery run.
SUBNETADDRESS_X	The subnet IP address for the network.
TYPE_X	The type of network.

IPROUTE table

This section describes the columns in the IPROUTE table.

Table 22. IPROUTE table

Column	Description
DELETED_X	Indicates whether the object is deleted, from among the following values: **1** The object is deleted. **0** The object is not deleted.
PK__DESTINATION_X	The destination of the route.
PK__PARENTIPROUTE_X	A relationship that corresponds to, and represents implicitly, the *manages* relationship.

J2EEDOMN table

This section describes the columns in the J2EEDOMN table.

Table 23. J2EEDOMN table

Column	Description
DELETED_X	Indicates whether the object is deleted, from among the following values: **1** The object is deleted. **0** The object is not deleted.
DISPLAYNAME_X	The machine-generated, user-friendly display name of the object.
JDOCLASSX	The internal JDO class name of the object.
NAME_X	The domain name.
PK__ADMINHOST_X	The computer system hosting the domain manager.

Table 23. J2EEDOMN table (continued)

Column	Description
PK__HOST_X	The foreign key to the COMPSYS table.
PK__PRIMARYSAP_X	The bind address (host name/IP address, port) on which the domain manager is listening. This is an alternate key.
PK_X	The primary key.

MSS table

This section describes the columns in the MSS table.

Table 24. MSS table

Column	Description
ACCESSSTRING_X	The access string for the MSS.
ACCESSTYPE_X	The access type for the MSS.
CREATEDBY_X	The creator of the object.
DELETED_X	Indicates whether the object is deleted, from among the following values: **1** The object is deleted. **0** The object is not deleted.
DESCRIPTION_X	The description of the MSS.
DISPLAYNAME_X	The machine-generated, user-friendly display name of the object.
HOSTNAME_X	The host name of the MSS.
LABEL_X	The user-defined label associated with the object.
LASTMODIFIEDTIME_X	The date and time the object was last modified.
MANUFACTURERNAME_X	The name of the manufacturer of the MSS.
MSSNAME_X	The name of the MSS.
PRODUCTNAME_X	The vendor-specific product name.
PRODUCTVERSION_X	The vendor-specific product version.

OPERATINGSYSENTS_FD67DE48X table

This section describes the columns in the OPERATINGSYSENTS_FD67DE48X table.

Table 25. OPERATINGSYSENTS_FD67DE48X table

Column	Description
PK__JDOIDX	The lookup identifier.
PK__SOFTWARECOMPONENTS_X	The foreign key to the SFTCMP table.

OPERATINGSYSWARE_2D8AA18X table

This section describes the columns in the OPERATINGSYSWARE_2D8AA18X table.

Table 26. OPERATINGSYSWARE_2D8AA18X table

Column	Description
PK_INSTALLEDSOFTWARE_X	The foreign key to the SOFTINSTALL table.
PK_JDOIDX	The lookup identifier.

OPSYS table

This section describes the columns in the OPSYS table.

Table 27. OPSYS table

Column	Description
DELETED_X	Indicates whether the object is deleted, from among the following values: **1** The object is deleted. **0** The object is not deleted.
DESCRIPTION_X	The description of the operating system.
DISPLAYNAME_X	The machine-generated, user-friendly display name of the object.
GUID_X	The Globally Unique Identifier (GUID) of the object.
JDOCLASSX	The internal JDO class name of the object.
KERNELARCHITECTURE_X	The kernel architecture of the operating system.
LABEL_X	The user-defined label associated with the object.
LASTMODIFIEDTIME_X	The date and time the object was last modified.
PK__KERNELMODULES RAWDATA_X	The raw list of kernel modules loaded.
KERNELVERSION_X	The version of the kernel release.
NAME_X	The name of the operating system. This is the uname for UNIX-based systems.
OSMODE_X	The instruction set of the architecture.
OSNAME_X	The name of the operating system. This is the uname for UNIX-based systems. This is the same value as the NAME_X column.
OSVERSION_X	Version of the operating system.
PK__PACKAGESINSTALLEDRAWDATA_X	Packages installed on the operating system. The parent ComputerSystem object on which this operating system is installed.
PK__PATCHESINSTALLED RAWDATA_X	The patches applied to the operating system.
PK_X	The primary key.

ORCLDBASE table

This section describes the columns in the ORCLDBASE table.

Table 28. ORCLDBASE table

Column	Description
DBNAME_X	The product name.
DBVERSION_X	The product version.
DELETED_X	Indicates whether the object is deleted, from among the following values: **1** The object is deleted. **0** The object is not deleted.
PK__PARENTORACLE DATABSE_X	The containing instance; the instance is almost always one-to-one with the database.
PK__SCHEMACONTENT_X	All database objects as specified in the Oracle Sensor configuration file. This includes table or view definitions, indexes, users, among others.
PK__SCHEMASRAW DATA_X	A small set of table names only.
PK_X	The primary key.

ORGENTITY table

This section describes the columns in the ORGENTITY table.

Table 29. ORGENTITY table

Column	Description
DELETED_X	Indicates whether the object is deleted, from among the following values: **1** The object is deleted. **0** The object is not deleted.
GLOBALNAME_X	The globally unique name of the organization.
PK_X	The primary key.

PERSOBJ table

This section describes the columns in the PERSOBJ table.

Table 30. PERSOBJ table

Column	Description
CLASSNAME_X	The internal class name for the object.
GUID_X	The Globally Unique Identifier (GUID) of the object.

RELATION table

This section describes the columns in the RELATION table.

Table 31. RELATION table

Column	Description
CREATEDBY_X	The creator of the object.

Table 31. RELATION table (continued)

Column	Description
DELETED_X	Indicates whether the object is deleted, from among the following values: **1** The object is deleted. **0** The object is not deleted.
SOURCE_X	The Globally Unique Identifier (GUID) of the source.
TARGET_X	The Globally Unique Identifier (GUID) of the target.
TYPE_X	The type of the relationship.

RFCPROXY table

This section describes the columns in the RFCPROXY table.

Table 32. RFCPROXY table

Column	Description
DESCRIPTION_X	The description of the object.
DISPLAYNAME_X	The machine-generated, user-friendly display name of the object.
FOREIGNRFCHANDLE_X	The internal request for change (RFC) handle used in an external (non ITSM) service desk application such as Remedy, Peregrine, or MRO, among others.
FOREIGNRFCID_X	The request for change identifier used in an external (non ITSM) service desk application such as Remedy, Peregrine, or MRO, among others.
GUID_X	The Globally Unique Identifier (GUID) of the object.
ITSMRFCHANDLE_X	The internal ITSM request for change handle in the external owning application. This string (if known) should not be used as a display label.
ITSMRFCID_X	The internal ITSM request for change identifier in the external owning application. The string can be a number or a combination of letters and numbers used for display purposes.
LABEL_X	The user-defined label associated with the object.
LASTMODIFIEDTIME_X	The date and time the object was last modified.
OPENDATE_X	The date when the request for change was opened.

SAP table

This section describes the columns in the SAP table.

Table 33. SAP table

Column	Description
DELETED_X	Indicates whether the object is deleted, from among the following values: **1** The object is deleted. **0** The object is not deleted.
PK__BINDADDRESS_X	The bind address (the foreign key to the BINDADDR table).
PK_X	The primary key.

SERVICE table

This section describes the columns in the SERVICE table.

Table 34. SERVICE table

Column	Description
DELETED_X	Indicates whether the object is deleted, from among the following values: **1** The object is deleted. **0** The object is not deleted.
DISPLAYNAME_X	The machine-generated, user-friendly display name of the object.
JDOCLASSX	The internal JDO class name of the object.
LDAPVERSION_X	The version of the LDAP server, if this is an LDAP service.
PK__HOST_X	The foreign key to the COMPSYS table.
PK__PRIMARYSAP_X	The bind address (the foreign key to the BINDADDR table).
PRODUCTVERSION_X	The vendor-specific product version.
ROOTDOMAIN_X	The root domain name, if this is a DNS.
SCHEMAVERSION_X	The version of the schema, if this is an LDAP service.
VENDORNAME_X	The vendor-specific name of the service software.

SFTCMP table

This section describes the columns in the SFTCMP table.

Table 35. SFTCMP table

Column	Description
DELETED_X	Indicates whether the object is deleted, from among the following values: **1** The object is deleted. **0** The object is not deleted.
INSTALLEDDATE_X	The date that the component was installed.
NAME_X	The name of the component.

Table 35. SFTCMP table (continued)

Column	Description
PK_X	The primary key.
PUBLISHER_X	The publisher of the component.
SOFTWAREVERSION_X	The version of the software for the component.

SFTPATCH table

This section describes the columns in the SFTPATCH table.

Table 36. SFTPATCH table

Column	Description
DELETED_X	Indicates whether the object is deleted, from among the following values: **1** The object is deleted. **0** The object is not deleted.
NAME_X	The name of the component.
PK_X	The primary key.
VERSION_X	The version of the software for the component.

SNMPSYS table

This section describes the columns in the SNMPSYS table.

Table 37. SNMPSYS table

Column	Description
PK__PARENTSNMPSYSTEMGROUP_X	The SNMP system group.
SYSCONTACT_X	The system contact from the SNMP MIB.

SOFTINSTALL table

This section describes the columns in the SOFTINSTALL table.

Table 38. SOFTINSTALL table

Column	Description
DELETED_X	Indicates whether the object is deleted, from among the following values: **1** The object is deleted. **0** The object is not deleted.
DESCRIPTION_X	The description of the system.
DISPLAYNAME_X	The machine-generated, user-friendly display name of the object.
GUID_X	The Globally Unique Identifier (GUID) of the object.
INSTALLEDLOCATION_X	The fully qualified path of the software installation.

Table 38. SOFTINSTALL table (continued)

Column	Description
LABEL_X	The user-defined label associated with the object.
LASTMODIFIEDTIME_X	The date and time the object was last modified.
MANUFACTURERNAME_X	The name of the software vendor.
PK__PARENTSOFTWATION_8B227479X	The operating system on which the software installation resides.
PK_X	The primary key.
PRODUCTID_X	The vendor-specific product identifier.
PRODUCTNAME_X	The vendor-specific product name.
VERSIONSTRING_X	The version of the software for the component.

SOFTWAREPATCENTS_F064A317X table

This section describes the columns in the SOFTWAREPATCENTS_F064A317X table.

Table 39. SOFTWAREPATCENTS_F064A317X table

Column	Description
PK__JDOIDX	The lookup identifier.
PK__SOFTWARECOMPONENTS_X	The foreign key to the SFTCMP table.

SYBSDBASE table

This section describes the columns in the SYBSDBASE table.

Table 40. SYBSDBASE table

Column	Description
DELETED_X	Indicates whether the object is deleted, from among the following values: **1** The object is deleted. **0** The object is not deleted.
DISPLAYNAME_X	The machine-generated, user-friendly display name of the object.
OWNER_X	The database owner.
PK__PARENTSYBASE DATABASE_X	The containing Sybase server.

WASNODE table

This section describes the columns in the WASNODE table.

Table 41. WASNODE table

Column	Description
DELETED_X	Indicates whether the object is deleted, from among the following values: **1** The object is deleted. **0** The object is not deleted.
DISCOVERYPROTOCOL_X	The communications protocol used by the node agent and servers on the node for process discovery activity. The data type is TCPIPProtocolType (UDP, TCP, MULTICAST).
NAME_X	The node name which, by convention, is the same as the host name on which it is running.
PK__ADMINHOST_X	The administration host on which the node agent is running.
PK__CELL_X	The cell which manages this node.
PK__CONFIGFILE_X	All configuration files for the node.
PK__HOST_X	The administration host on which the node agent is running. This is the same as PK__ADMINHOST_X.
PK__NODEAGENT_X	The nodeagent for the node. A nodeagent is a customized WAS server.
PK__PRINARYSAP_X	The Primary Service Access Point, which is the unique bind address to which the nodeagent is listening.
PK_X	The primary key.
ROOTDIRECTORY_X	The root directory where WebSphere is installed.
TYPE_X	The type of the node, from among the following: • STANDALONE • DISTRIBUTED

Appendix B. 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 grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

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

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, 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.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
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 information 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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

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

Trademarks

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

AIX
AIX 5L

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

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

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

Electronic emission notices

Federal Communications Commission (FCC) statement

Index

Program Number:

Printed in USA