OPTIMIZE

MERCURY BUSINESS AVAILABILITY CENTER™

CI Type Manager Administration

MERCURY™

BUSINESS TECHNOLOGY OPTIMIZATION

Mercury Business Availability Center

CI Type Manager Administration

Version 6.2

Document Release Date: June 20, 2006

MERCURY™

Mercury Business Availability Center, Version 6.2
CI Type Manager Administration

Mercury provides links to external third-party Web sites to help you find supplemental information. Site content and availability may change without notice. Mercury makes no representations or warranties whatsoever as to site content or availability.

Mercury Interactive Corporation
379 North Whisman Road
Mountain View, CA 94043
Tel: (650) 603-5200
Toll Free: (800) TEST-911
Customer Support: (877) TEST-HLP
Fax: (650) 603-5300

If you have any comments or suggestions regarding this document, please send them by e-mail to documentation@mercury.com.

AMLIB_CITAD6.2/01

Table of Contents

PART I: INTRODUCTION

PART II: MANAGING CI TYPES

PART III: APPENDIXES

Welcome to CI Type Manager Administration

This guide explains how to work with the CI Type Manager.

How This Guide Is Organized

The guide contains the following chapters:

Part I Introduction

Introduces the CI Type Manager that enables you to view the structure of your managed world.

Part II Managing CI Types

Explains how to create and add CITs to the CI Type Model. Describes how to create a predefined list whose values define an attribute type.

Part III Appendixes

Provides a table of all possible CITs in the CMDB. The CIT’s parent and any key attributes are also listed. A second table lists all permitted relationships between CITs.

Who Should Read This Guide

This guide is intended for the following users of Mercury Business Availability Center:

- Mercury Business Availability Center administrators
- Mercury Business Availability Center platform administrators
- Mercury Business Availability Center application administrators
- Mercury Business Availability Center data collector administrators

Readers of this guide should be knowledgeable about enterprise system administration, have familiarity with ITIL concepts, and be knowledgeable about Mercury Business Availability Center in general and Mercury Application Mapping technology specifically.

Getting More Information

For information on using and updating the Mercury Business Availability Center Documentation Library, reference information on additional documentation resources, typographical conventions used in the Documentation Library, and quick reference information on deploying, administering, and using Mercury Business Availability Center, refer to *Getting Started with Mercury Business Availability Center*.

Part I

Introduction

1

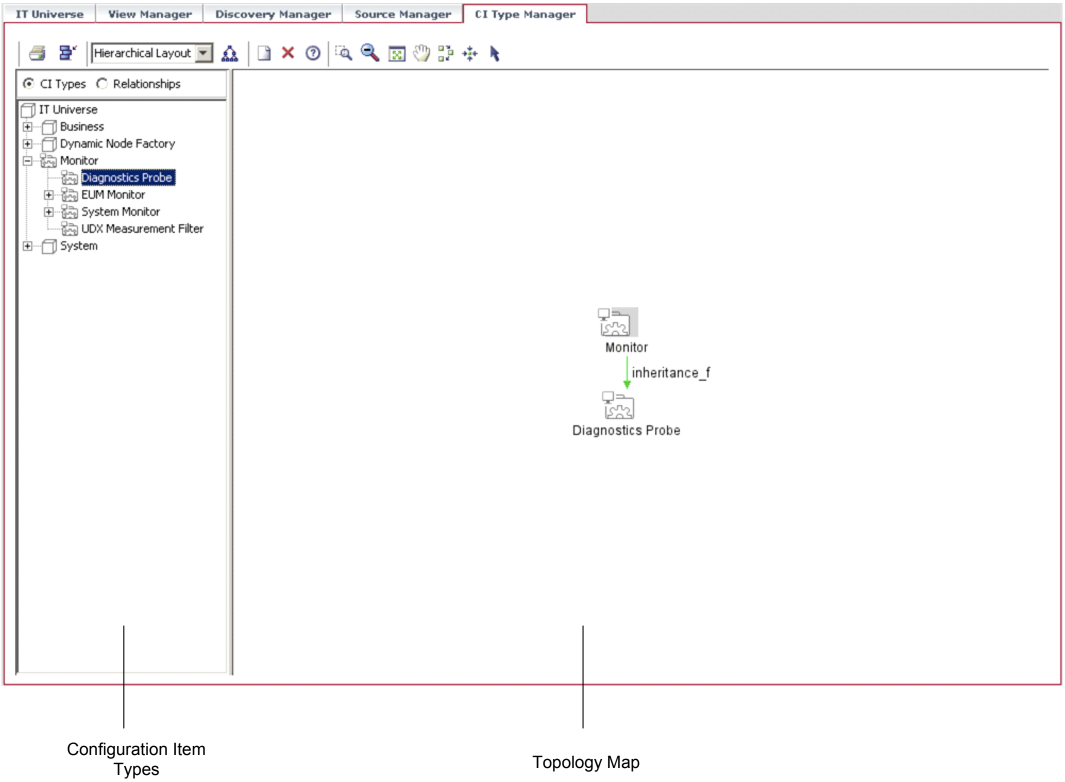
Introduction to the CI Type Manager

This chapter introduces the CI Type Manager. The CI Type Manager enables you to view the structure of your managed world.

About the CI Type Manager

The CI Type Manager is displayed by selecting **Admin** > **CMDB** and then clicking the **CI Type Manager** tab. The CI Type Manager enables you to view the information in the CI Type model, which contains the definitions of all the configuration item types (CITs) defined in the system and the relationships that define the connections between them. Each CIT has its own attributes, as well as the attributes inherited from its parent CIT. The CI Type Manager represents the structure of Mercury Business Availability Center's managed world.

The CI Type Manager visually reflects the data contained in the CMDB, including TQL queries and managed views.



The CI Type Manager tab is divided as follows:

- **Configuration Item Types –** Displays a hierarchical tree structure of the CI Type model containing the inheritance relationships among CITs. All CITs included in the CI Type Model are classified as either a CIT or a relationship. By selecting a CIT in the View Explorer, you can drill down and view the relationships and neighbors of the selected CIT in the topology map. The View Explorer displays the number of instances there are of each CIT in the CMDB.

Note: Each CIT definition type is represented by a unique icon.

- **Topology Map –** Displays various layouts of the CIT Model, including the CITs and the relationships connecting them. Each CIT has a unique icon. Its neighbors, as indicated by its connection to the CIT in the View Explorer, all share the same shape surrounding the icon. Shapes are predefined according to logical CIT groupings.

Working With the CI Type Manager

Use the CI Type Manager to perform the following functions:

- Create new CITs and edit existing ones.
- Expand or collapse the View Explorer to display the required CITs and relationships in the topology map.
- View the selected CIT's neighbors, which are connected using inheritance, container and other types of relationships.
- Zoom in and out of the topology map to view the selected CIT and its neighbors at different zoom levels.

Browsing the CIT Model

The CI Type Manager enables you to browse the CIT Model to better understand the relationships between specific CITs.

This section includes the following topics:

CI Type Manager Tooltip

When you point to a CIT in the CI Type Manager, the tooltip displays the following information:

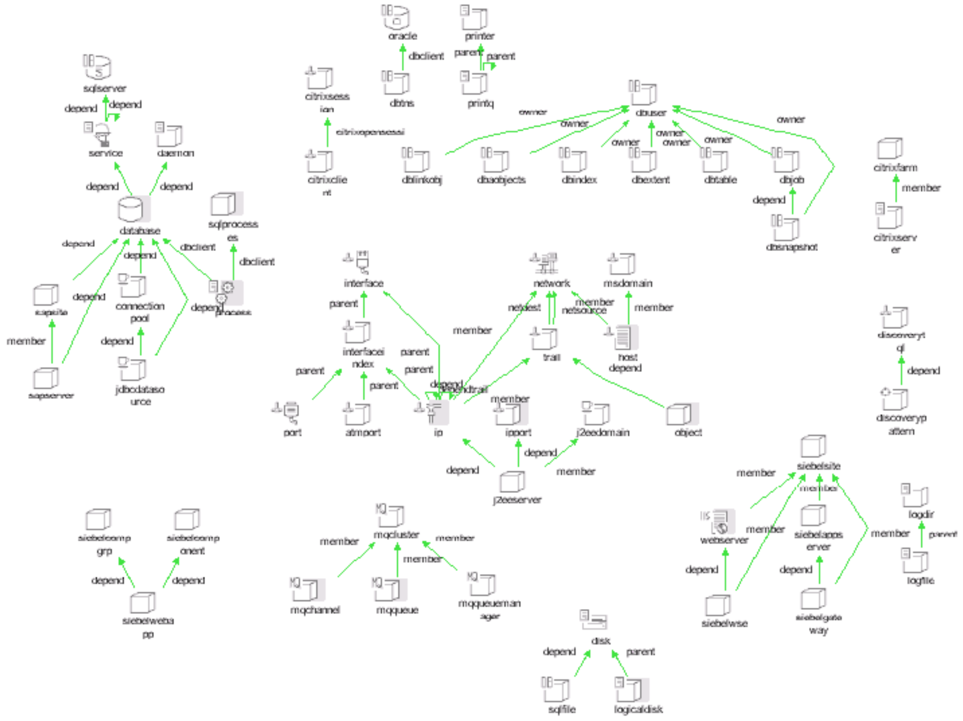
- **Label –** The name of the CIT to which the item belongs.
- **Instances –** The number of instances that currently exist in the CMDB from the selected CIT or its children.

Browsing the CI Type Model

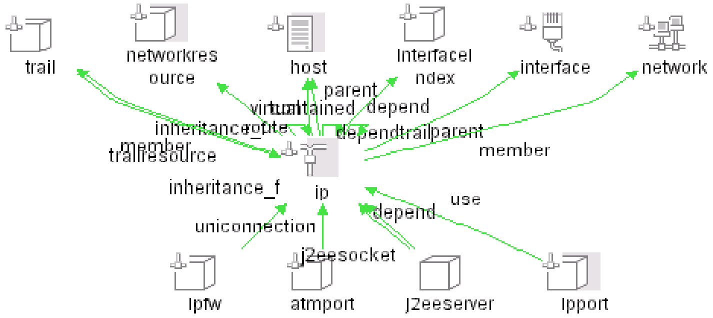
In the View Explorer, you can view inheritance relationships between CITs (such as the **Server Port**, **Client Port**, and **IP Unknown** CITs, which inherit attributes from the **TCP/IP Port** CIT). In the topology map, you can view relationships of all types that connect various CITs. If required, you can print the contents of the topology map.

To browse the CIT Model:

1 From the View Explorer, select the CIT that you want to view in the topology map. The CIT is automatically selected in the topology map.



2 Use either the **Zoom** or **Interactive zoom** button on the toolbar to zoom in on the view area of the selected layer.



3 To return to the default view, click the **Fit To Window** button on the toolbar.

Displaying CIT Instances

You can display all of the instances found in the CMDB for each CIT in a table.

To display all of the instances found for each CIT in a table:

1 In the Topology Map or the View Explorer, right-click the required CIT and select **Show CIT instances** to open the Show all instances dialog box.

The list of CITs are divided into pages. The number at the bottom of the screen indicates which page is currently being displayed. For example, 2/4 means that it is the second out of four pages.

2 To view other pages, use the left and right arrows.

3 To determine the number of node instances that appear on a page, do the following:

➤ Click the **Set bulk size** button to open the Set bulk size dialog box.

➤ Use the up and down arrows or type the number of node instances you want to appear on a page and click **OK**.

4 To see the table updated, click the **Refresh** button.

5 Click **OK** to save the settings you have defined.

Using the Toolbar Options

For a description of each toolbar option in the View Manager, see “Toolbar Options” in *Working with the CMDB*.

Printing the Contents of the Topology Map

For a description of how to print the contents of the topology map, see “Printing the Topology Map” on page 38 in *Working with the CMDB*.

Defining a View's Layout

For a description of how to customize the layout of a specific layer in a view, see "Defining a View's Layout" in *Working with the CMDB*.

Understanding Layout Options

For a description of how you can display the contents of the topology map using different layout options, see "Understanding Layout Options" in *Working with the CMDB*.

Part II

Managing CI Types

2

Managing CITs

This chapter explains how to create and add CITs to the CI Type Model.

Note: The CI Type Manager is not available to Mercury Managed Services customers.

About Managing CITs

If you have IT resources that are not included in the system's built-in CITs, you can create and add CITs to the Configuration Item Type Model. In addition, you can edit existing CITs to adjust them to the structure of your IT infrastructure.

For a list of the existing CITs and their descriptions, see "CI Type Descriptions" on page 29.

For a list of the existing relationships and their definitions, see "Relationship Definitions" on page 73.

CI Type Workflow

You create CITs according to the following workflow:

- Create CITs. For details, see "Creating CITs" on page 14.
- Assign an icon to a CIT. For details, see "Assigning an Icon to a CIT" on page 19.
- Define CIT qualifiers. For details, see "Defining Qualifiers" on page 21.
- Define an attribute for the CIT label. For details, see "Defining an Attribute for a CIT Label" on page 22.

Creating CITs

You can create and add CITs to the CI Type Model.

To create a CIT:

1. Select **Admin** > **CMDB** and click the **CI Type Manager** tab.
2. Click the **New** button on the toolbar. The Create Configuration Item Type dialog box is displayed.

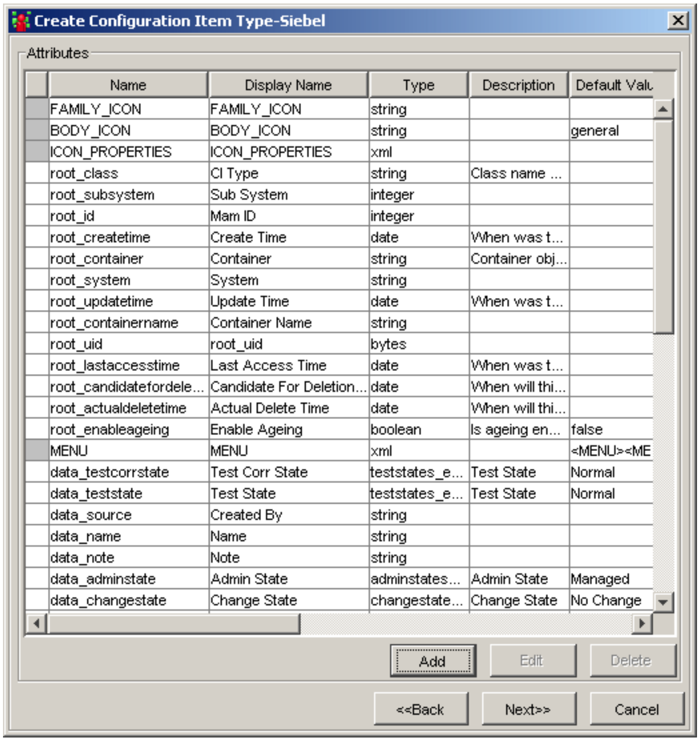
3 In the **Configuration Item Type Name** box, enter a unique name for the new CIT. Do not use a blank space or an underscore as part of the CIT name. You can use lower and upper case, but you cannot use the same name with different cases for two CITs.

4 In the **Display Name** box, type the name of the CIT as you would like it to appear on the Mercury Business Availability Center interface.

5 In the **Base Configuration Item Type** box, select a base CIT for the CIT you are creating. The new CIT inherits the base CIT's attributes.

6 (Optional) In the **Configuration Item Type Description** box, enter a description of the new CIT.

7 Click **Next**. The Attributes page displays all the attributes that the new CIT inherits from its base CIT.



Name	Display Name	Type	Description	Default Valu
FAMILY_ICON	FAMILY_ICON	string		
BODY_ICON	BODY_ICON	string		general
ICON_PROPERTIES	ICON_PROPERTIES	xml		
root_class	CI Type	string	Class name ...	
root_subsystem	Sub System	integer		
root_id	Mam ID	integer		
root_createtime	Create Time	date	When was t...	
root_container	Container	string	Container obj...	
root_system	System	string		
root_updatetime	Update Time	date	When was t...	
root_containername	Container Name	string		
root_uid	root_uid	bytes		
root_lastaccesstime	Last Access Time	date	When was t...	
root_candidatefordele...	Candidate For Deletion...	date	When will thi...	
root_actualdeletetime	Actual Delete Time	date	When will thi...	
root_enableageing	Enable Ageing	boolean	Is ageing en...	false
MENU	MENU	xml		<MENU><ME
data_testcorrstate	Test Corr State	teststates_e...	Test State	Normal
data_teststate	Test State	teststates_e...	Test State	Normal
data_source	Created By	string		
data_name	Name	string		
data_note	Note	string		
data_adminstate	Admin State	adminstates...	Admin State	Managed
data_changestate	Change State	changestate...	Change State	No Change

8 To add attributes, click **Add** to open the Add Attribute page.

9 Enter the following information about the new attribute:

- **Attribute Name** – Enter a unique name for the new attribute to identify it in the database.

  Note: Do not use a space or an underscore as part of the attribute name.

- **Display Name** – Enter a name for the new attribute to identify it on the Mercury Business Availability Center interface.
- **Description** (Optional) – Enter a description for the new attribute.
- In the **Attribute Type** area, select one of the following:
  - **Primitive –** Choose from one of the following field types: **boolean**, **bytes**, **date**, **double**, **integer**, **float, long**, **string, xml**.
  - **Enumeration/List –** Contains a list of Enumerations/Lists defined in the Enumeration Manager dialog box (for details, see “Enumerations and Lists” in the *Mercury Application Mapping Administration Guide*). This option enables you to define an attribute whose value has been predefined.

    For example, a location attribute might be defined by a **location** list containing the following values:

    New York

    Boston

    Baltimore

    The values that appear in the list in the **Attribute Type** section vary depending on the **Attribute Type** you selected.
- **Value Size** – Enter a value for the maximum physical size of the new attribute, if required. This field applies to the attribute type **string** only.
- **Default Value** – Enter a default value for the attribute, if required. This value appears when the new CIT is created and there is no runtime value for the attribute. Choose from the following fields: **Managed**, **Unmanaged**, **Unknown**, **Testing**, **Restricted**, **Disabled**.

➤ In the **Advanced** area:

- (Optional) An index provides rapid access to the attribute values. Select **Index** to accelerate the attribute retrieval performance. This option is recommended for attributes that are used frequently in search conditions. For example, IP address is usually an index attribute of a host.
- (Optional) Select **Set** to define the attribute as a list if the attribute can have multiple values simultaneously. For example, there can be multiple IP addresses for one host.
- (Optional) Select **Required** to define this attribute as a required one, if its value is required for the creation of the CIT.
- (Optional) Select **Visible** to display this attribute in the Attributes page in the topology map.
- (Optional) Select **Editable** to enable future editing of the attribute. Only attributes that are marked as **Editable** (or ones that have values) are displayed in the Attributes page.

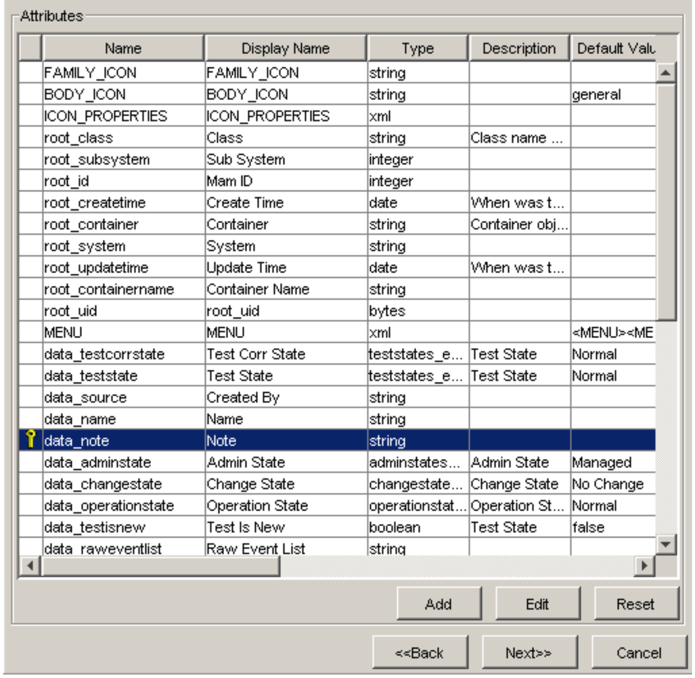
Note: The **Configuration Management only** area is not relevant for Mercury Business Availability Center.

10 Click **OK** to save the changes you have made. The new CIT attribute you created appears in the Attributes page.

Note: If you modify an attribute belonging to its parent, it turns light blue. If you modify an attribute belonging to the CIT itself, it turns dark blue.

11 If you want to add more attributes, repeat steps 8 to 10.

12 To define an attribute as a key attribute, click in the left column beside the attribute name. A key icon appears as shown in the following figure:



	Name	Display Name	Type	Description	Default Valu
	FAMILY_ICON	FAMILY_ICON	string		
	BODY_ICON	BODY_ICON	string		general
	ICON_PROPERTIES	ICON_PROPERTIES	xml		
	root_class	Class	string	Class name ...	
	root_subsystem	Sub System	integer		
	root_id	Mam ID	integer		
	root_createtime	Create Time	date	When was t...	
	root_container	Container	string	Container obj...	
	root_system	System	string		
	root_updatetime	Update Time	date	When was t...	
	root_containername	Container Name	string		
	root_uid	root_uid	bytes		
	MENU	MENU	xml		<MENU><ME
	data_testcorrstate	Test Corr State	teststates_e...	Test State	Normal
	data_teststate	Test State	teststates_e...	Test State	Normal
	data_source	Created By	string		
	data_name	Name	string		
🔑	data_note	Note	string		
	data_adminstate	Admin State	adminstates...	Admin State	Managed
	data_changestate	Change State	changestate...	Change State	No Change
	data_operationstate	Operation State	operationstat...	Operation St...	Normal
	data_testisnew	Test Is New	boolean	Test State	false
	data_raweventlist	Raw Event List	string		

13 To remove the key attribute definition, click the key icon.

14 Click **Next** to attach an icon to a new CIT (for details, see “Assigning an Icon to a CIT” below).

Assigning an Icon to a CIT

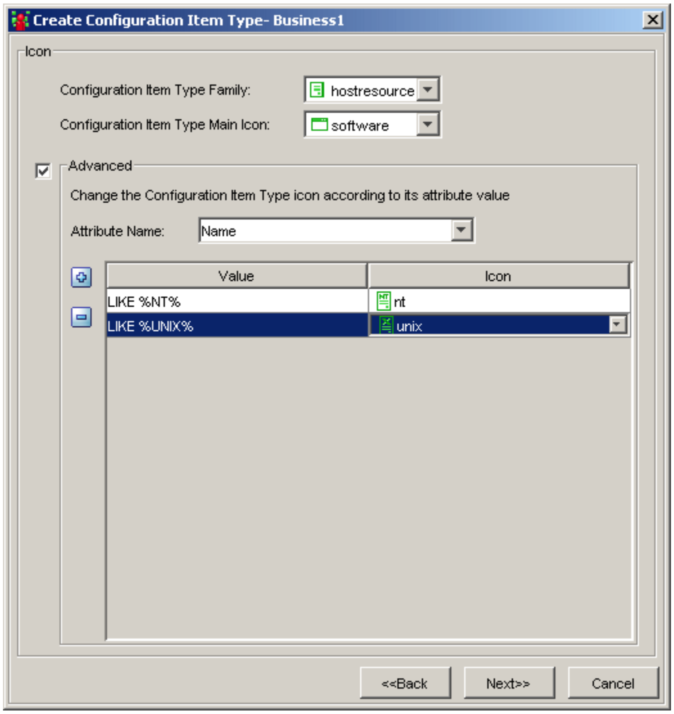
Mercury Business Availability Center allows you to attach an icon to a new CIT. You can also attach different icons to the same CIT when certain conditions apply. For example, you can associate different icons with the same CIT when one of its attribute values changes.

Note: You can only assign different icons to CITs that are classified as CI Types, and not relationships.

To assign one or more icons to a CIT:

1. In the Attributes page (see step 7 in “Creating CITs” on page 14), click **Next** to open the Icon page.
2. From the **Configuration Item Type Family** list, select the icon you want to attach to the specified CIT (for details, see “Understanding CI Icon Composition” in *IT Universe Manager Administration*).
3. From the **Configuration Item Type Main Icon** list, select the group to which the CIT belongs.
4. Select **Advanced** to assign different icons when the selected CIT attribute gets a different value.
5. From the **Attribute Name** list, select the attribute to which you want to attach different icons.
6. Click the **Expand** button to create a new row.
7. Double-click inside the **Value** column and either select a value from the list that appears, or type the required value.
8. Double-click inside the **Icon** column and select the icon you want to associate with that value.
9. To add another value, repeat steps 6 to 8.

In the following example, the computer icon is to be replaced by the NT icon when the attribute’s value is **LIKE %NT%,** and by the UNIX icon when the attribute’s value is **LIKE %UNIX%**.



Create Configuration Item Type- Business1

Icon

Configuration Item Type Family: hostresource

Configuration Item Type Main Icon: software

☑ Advanced

Change the Configuration Item Type icon according to its attribute value

Attribute Name: Name

Value	Icon
LIKE %NT%	nt
LIKE %UNIX%	unix

10 To delete a row, select the row you want to delete and click the **Delete** button.

Note: If you change the icon of a CIT that appears in an existing view, the CIT's icon is not updated in the view. For details on how to update the view with the new icon, see “Updating a View” on page 25.

---

11 Click **Next** to define qualifier conditions for the CIT.

Defining Qualifiers

The CI Type Manager allows you to assign qualifiers to a CIT definition. Qualifiers enable you to define added attribute definitions to the CIT. For example, you can use a qualifier to define a CIT as abstract, meaning you cannot create instances from it.

To define qualifier:

1. In the Icon page (for details on the Icon page, see “Assigning an Icon to a CIT” on page 19),click **Next** to display the Qualifiers page.
2. In the **Qualifiers** list, select the required qualifiers using the Add buttons to move your selection(s) to the **Configuration Item Type Qualifiers** list. You can make multiple selections by holding down the CTRL key.

   - ➤ Use this button to move the selected qualifier(s) to the **Configuration Item Type Qualifiers** list.

   - ➤ Use this button to move all the qualifiers to the **Configuration Item Type Qualifiers** list.

   Select the required qualifier(s) according to the following:

   - ➤ **NOTIFY_CHANGES** – For internal use only.
   - ➤ **BLE_LINK_CLASS** – For internal use only.
   - ➤ **ABSTRACT_CLASS** – You cannot create instances of this CIT.
   - ➤ **ITU_HIDDEN_CLASS** – For internal use only.
   - ➤ **CONTAINER** – Relevant for relationships only. Represents containment between two CIs.
   - ➤ **HIDDEN_CLASS –** Does not appear anywhere in the application.
   - ➤ **READ_ONLY_CLASS** – For internal use only.

3. To remove a qualifier, select the qualifier from the **Configuration Item Type Qualifiers** list and click the **Delete** button.
4. Click **Next** to define attributes for a CIT label. For details, see “Defining an Attribute for a CIT Label” on page 22.

Defining an Attribute for a CIT Label

This section describes how to define an attribute for a CIT label. A label is the title that appears under a CI. Label definition can be customized to include different attribute values. For example, if in host the function label is composed of hostname and network, the displayed label is: server1 10.0.65.0.

To define an attribute for the CIT label:

1. In the Qualifiers page (for details, see "Defining Qualifiers" on page 21), click **Next** to display the Default Label page.
2. Select the attribute(s) to appear in the label of the selected item (for details on how to format the text, see "Formatting Text" on page 23 below).
3. Click **Finish**. Mercury Business Availability Center creates the new CIT using the information you provided, and displays it on the View Explorer and in the topology map in the Configuration Item Type Model.

Formatting Text

...

Select the attribute(s) to appear in the formatted text by clicking the button to the left of each attribute to display the selected attribute in the **Format** box. If required, you can use the function buttons to display multiple attributes in the label as follows:

Option	Description
()	Adds parentheses to the formatted text (used in conjunction with the other functions).
&	Places an **AND** operator between two attributes in the formatted text. For example, **network_netaddr&network_domain** displays both the network address and the domain of the node.
|	Places an **OR** operator between two attributes in the formatted text.
RegExp	Adds a regular expression using regular expression syntax to the label definition. For examples of how to use regular expression syntax, see "Using Regular Expressions" on page 24.

Note: To delete an attribute from the **Format** box, highlight it and press the ENTER or DELETE key.

You can define conditions using the following combinations:

- AND
- OR
- AND NOT
- OR NOT

Using Regular Expressions

This section gives an example of how to use regular expression syntax.

To enter a regular expression:

1 In the first field, enter the regular expression pattern. This is the structure of the selected attribute.

2 In the second field, enter the group number. This is the part of the regular expression pattern to focus on when creating the label.

For example, enter a regular expression to define the IP address (aa.yy.zz.mm), as follows:

To:	In the first field enter:	In the second field enter:
Create label by zz	(.*[.].*[.])(.*)([.].*)	2
Create label by yy	(.*[.])(.*)([.].*[.].*)	2
Create label by aa	(.*)([.].*[.].*[.].*)	1
Create label by mm	(.*[.].*[.].*[.])(.*)	2

You can also enter a regular expression to create the label by the first or last letter(s) of the selected attribute as the following example shows:

To:	In the first field enter:	In the second field enter:
Create label by the first letter	(.)(.*)	1
Create label by the last letter	(.*)(.)	2
Create label by the first two letters	(..)(.*)	1
Create label by the last two letters	(.*)(..)	2

Updating a View

If you change the icon of a CIT that appears in an existing view, the CIT's icon is not updated in the view. This section explains how to update the view with a new icon.

To update a view with a new icon:

1 Launch your Web browser and navigate to: **http://<server_name>:8080/jmx-console**, where <**server_name**> is the name of the machine on which the Processing Server is installed.

2 Under **MAM**, click **service=View System** to open the JMX MBEAN View page.

3 Enter the following to the right of the **rebuildViewAsynchronic** box:

- In the **customerId** box, enter 1.
- In the **viewName** box, enter the name of the view you want to rebuild.

4 Click **Invoke**.

Editing Existing CITs

You can edit an existing CIT.

To edit an existing CIT:

1 In the View Explorer, right-click the CIT you want to edit and click **Edit CIT** to open the Edit Configuration Item Type dialog box.

The Edit Configuration Item Type dialog box tabs are similar to those in the Create Configuration Item Type dialog box (as described in “Creating CITs” on page 14). However, only editable fields are enabled in the Edit Configuration Item Type dialog box.

2 Make your changes in the different tabs as follows:

- **Information** (see “Creating CITs” on page 14)
- **Attributes** (see “Creating CITs” on page 14)
- **Icon** (see “Assigning an Icon to a CIT” on page 19)
- **Qualifiers** (see “Defining Qualifiers” on page 21)

- **Default Label** (see “Defining an Attribute for a CIT Label” on page 22)

Adding Relationships Between CITs

You can add relationships between CITs, either built-in or new ones, which define their physical or logical connections.

To add a relationship between two CITs:

1. From the View Explorer or the Map pane, select the two CITs you want to link by holding down CTRL and clicking the CIT names.
2. Right-click one of the CITs and select **Add/Remove Relationship** to open the Add/Remove Relationship dialog box.
3. Select the check box(es) that specifies the type of relationship with which you want to link the CITs.
4. Click **OK** to save the changes you have made.

Removing Relationships Between CITs

You can remove relationships between two CITs.

To remove a relationship between two CITs:

1. From the View Explorer or the Map pane, select the two CITs whose relationship you want to delete by holding down CTRL and clicking the CIT names.
2. Right-click one of the CITs and select **Add/Remove Relationship** to open the Add/Remove Relationship dialog box.
3. Clear the check box(es) of the relationships that link the selected CITs.
4. Click **OK** to save the changes you have made.

Part III

Appendixes

A

CI Type Descriptions

This section provides additional information about CI Types, permitted relationships between CIs, and key attributes of CITs. .

Diagram of CI Type Host and Permitted Relationships

A CI can represent hardware, software, services, business processes, and so forth. CIs with similar properties are grouped into a CI type (CIT). Each CIT provides a template for creating a new CI and its associated properties.

A relationship defines the link between two CIs. Relationships represent the dependencies and connections between the entities in your IT environment. As an example, in the diagram below all CITs that can connect to CIT host are listed with the permitted relationships between them.

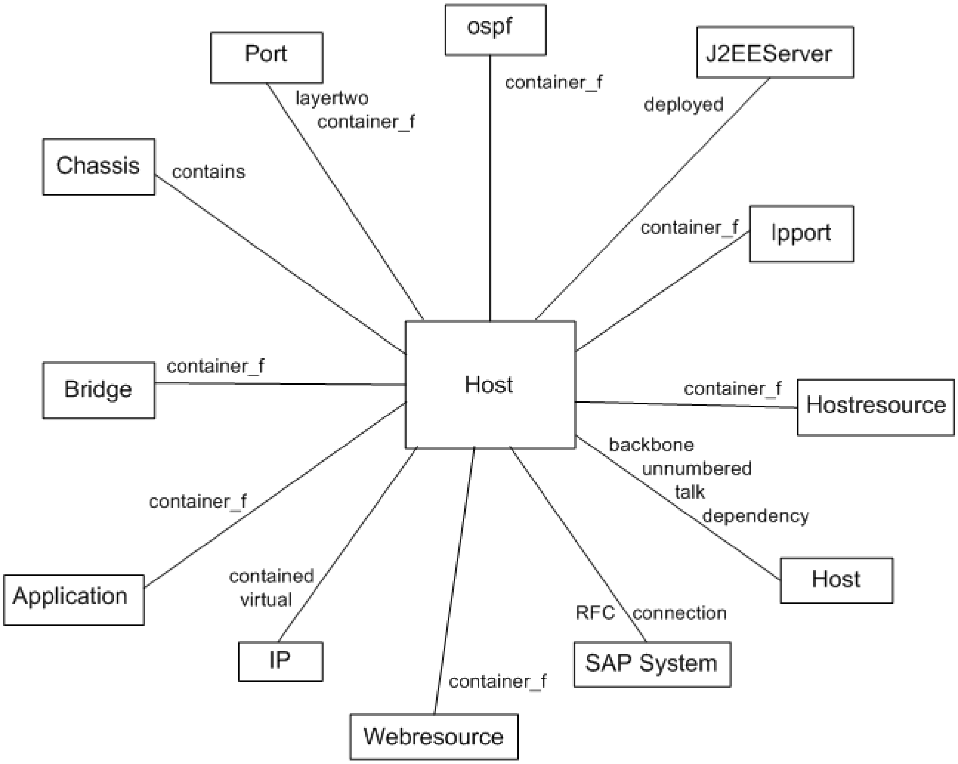
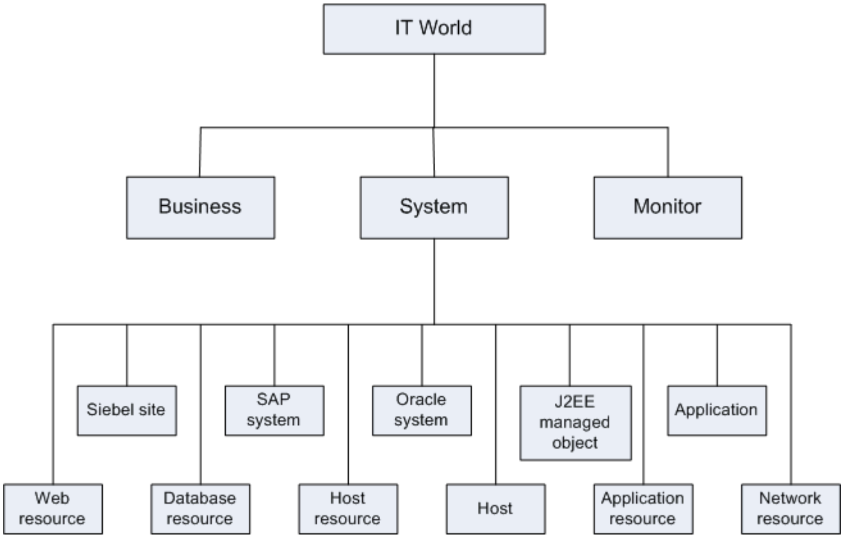


Diagram of CITs Whose Parent is System

The following diagram depicts all CITs whose immediate parent is system. To view the entire list of CITs under system, see "List of Parent, Description, and Key Attributes of all CI Types," below.



List of Parent, Description, and Key Attributes of all CI Types

The following table lists all CITs defined in the CIT Manager, the CIT's parent, a brief description, and key attributes if they exist.

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
agent	application	Virtual class to all agents.	
apache	webserver	Apache Web Server.	webserver_configf ile

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
apachevirtualhost	applicationresource	Apache Virtual Host.	
oracleapplication	oracleappsresource	Oracle E-Business Suite Application.	
applicationresourc e	system	Virtual class to all application resources.	root_container, data_name
atmport	networkresource	A port on an Asyschronous Transmit Mode switch.	atmport_pnniport number, atmswitching_pn ninodeaddress
atmswitch	host	Asynchronous Transfer Mode switch: a network technology that supports realtime voice, video, and data. The topology uses switches that establish a logical end-to-end circuit.	
atmswitching	networkresource	Asynchronous Transfer Mode switching.	ress
bridge	networkresource	A device that connects one local area network (LAN) to another.	bridge_basemacad dr
business	it_world	Business classes.	

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
chassis	networkresource	A chassis that holds Asynchronous Transfer Mode (ATM) switches.	chassis_uniqueid
ipclient	ipport	The client port to which the server port is connected.	ipclient_remoteip, ipclient_remoteport
concentrator	host	A switch or a hub with no SNMP agent.	host_key, data_name
configfile	document	Holds data from a configuration file of an application.	
connectionpool	jdbcresource	A cache of database connections maintained in the database's memory so that the connections can be reused when the database receives future requests for data. Connection pools are used to enhance the performance of executing commands on a database.	

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
cpu	hostresource	Central Processing Unit.	cpu_cid
csqlprocesses	sqlprocesses	An CI representing a Complete SQL process.	
database	application	A collection of data that is organized so that its contents can be easily accessed, managed, and updated.	
databaseresource	system	A virtual class to all database resources.	root_container, data_name
dbarchivefile	file	An archive file for a Microsoft SQL Server 2000 Analysis Services database. It contains the contents of the directory whose name is the same as the database name.	
dbcontrolfile	file	A database file containing information needed to maintain and verify database integrity.	

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
dbdatafile	databaseresource	The database file that contains the database's data.	dbdatafile_fileid
dbextent	databaseresource	A additional unit of space allocated to a SQL Server CI, such as a table or index, whenever the CI needs more space.	dbextent_segment type, dbextent_segment name, dbextent_extentid , dbextent_owner
dbindex	databaseresource	A mechanism to locate and access data within a database. An index may refer to one or more columns and be a means of enforcing uniqueness in column values.	dbindex_owner
dbjob	databaseresource	A specified series of operations performed sequentially by the database manager.	dbjob_jobid

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
dblinkobj	databaseresource	The link between databases that stores information about how to connect to the remote database.	
dbredofile	file	A transaction log that records data modifications made in the database.	
dbsegment	databaseresource	Contains the component database and any utilities provided by the developers for the DBA's use in installing and filling that particular database.	dbsegment_owner, dbsegment_segmenttype
dbsnapshot	databaseresource	A backup that can be created from an entire database or individual files.	
dbtable	databaseresource	Organizes the information about a single topic into rows and columns.	dbtable_owner

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
dbtablespace	databaseresource	Represents the amount of space required to store the data in a table.	
dbuser	databaseresource	A defined user in the database.	
db2	database	An IBM database management system that serves a number of different operating system platforms.	
dbaobjects	databaseresource	An CI that represents a database procedure, function, package, and package-body.	dbaobjects_type
daemon	hostresource	A program that forwards requests to other programs or processes.	
dir	file	The name of a group of files in the same location.	file_path
disk	hostresource	A logical division of a hard disk.	

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
document	applicationresource	A physical file.	
drive	logicaldisk	A hard disk drive.	
ejb	j2eemanagedobject	EJB component in Sun's J2EE platform that provides a Java environment for developing and running distributed applications.	
ejbmodule	j2eemodule	A module containing Enterprise JavaBeans.	
elan	networkresource	Emulated Local Area Network:can be created using ATM LAN emulation technology to join together existing LANs running protocols such as IP, Novell IPX, AppleTalk, and DECnet.	

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
enterasyse7blade	host	The atmswitch class represents a network element which its oid represents an enterasyse7blade.	
entitybean	ejb	An Enterprise JavaBean entity.	
eum_monitor	monitor	Monitor system classes.	
eventlog	hostresource	Microsoft event log data.	eventlog_type, eventlog_application
file	hostresource	An entity of data.	
filesystem	logicaldisk	UNIX file system.	
firewall	host	A gateway that limits access between networks in accordance with local security policy.	
host	system	A network element that has a unique IP address.	host_key
hostresource	system	Virtual class to all host resources.	root_container,data_name

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
httpcontext	applicationresource	An IBM HTTP server plug-in.	root_container, data_name
ibmhttpserver	webserver	An IBM HTTP server.	webserver_configfile
mqaliasq	mqqueuelocal	The Message Queue alias queue to a local queue or to a remote queue definition.	root_container, data_name
mqchannel	applicationresource	Message Queue channel data.	root_container, data_name
mqchclntconn	mqmqichannel	Message Queue client connection channel for Message Queue Interface (MQI) calls.	root_container, data_name
mqcluster	applicationresource	Message Queue cluster data.	
mqchclusrcvr	mqmsgreceiverchannel	Message Queue cluster receiver channel.	root_container, data_name
mqchclussdr	mqmsgsenderchannel	Message Queue cluster sender channel.	root_container, data_name
mqmsgchannel	mqchannel	Message Queue channel data.	root_container, data_name
mqmsgreceiverchannel	mqmsgchannel	Message Queue channel data.	root_container, data_name
mqmsgsenderchannel	mqmsgchannel	Message Queue channel data.	root_container, data_name

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
mqmqichannel	mqchannel	Message Queue channel data.	root_container, data_name
mqqueue	applicationresource	Message Queue definition data.	
mqqueuelocal	mqqueue	A local Message Queue queue.	root_container, data_name
mqqueuemanager	applicationresource	Message Queue manager data.	
mqqueueremote	mqqueue	A remote Message Queue queue definition.	root_container, data_name
mqchrcvr	mqmsgreceiverchannel	Message Queue receiver channel.	root_container, data_name
mqchrqstr	mqmsgreceiverchannel	Message Queue requester channel.	root_container, data_name
mqchsdr	mqmsgsenderchannel	Message Queue sender channel.	root_container, data_name
mqchsvr	mqmsgsenderchannel	Message Queue server channel.	root_container, data_name
mqchsvrconn	mqmqichannel	Message Queue server connection channel for Message Queue Interface (MQI) calls.	root_container, data_name
mqxmitq	mqqueuelocal	An Message Queue transmission queue.	root_container, data_name

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
webspheremq	application	Message Queue for WebSphere data.	
iis	webserver	Internet Information Server.	
interface	networkresource	A physical and logical way to support an attachment of any device to a connector or to another device.	interface_macaddr
interfaceindex	networkresource	Represents the unique number of an interface.	interfaceindex_index
ip	networkresource	Internet Protocol: the address which identifies the sender or the receiver of information sent across the internet.	ip_address, ip_domain
ipfw	ip	The IP address of a firewall.	ipfw_hopnumber, ipfw_trailid
ipunknown	ipport	Unknown TCP/UDP port.	
j2eeapplication	j2eedeployedobject	An application inside the application server	

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
j2eecluster	j2eeresource	A group of application servers that run a J2EE application as if they were a single server.	data_name
j2eedeployedobject	j2eemanagedobject	J2EE deployed CI.	root_container, data_name
j2eedomain	j2eemanagedobject	An application server domain.	
executequeue	j2eeresource	executequeue	data_name
j2eemanagedobject	system	A virtual class to all J2EE resources.	root_container, data_name
j2eemodule	j2eedeployedobject	J2EE deployed CI.	
j2eeresource	j2eemanagedobject	A virtual class to all J2EE resources.	
j2eeserver	application	A server within a J2EE application server.	
jdbcdatasource	j2eemanagedobject	Java Database Connectivity data source.	
jdbcprovider	jdbcresource	A provider for Java Database Connectivity.	

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
jdbcresource	j2eeresource	A virtual class to all Java Database Connectivity resources.	
jmsdatastore	jmsresource	Java Messaging Service data store.	
jmsdestination	jmsresource	Java Messaging Service destination.	
jmsprovider	jmsresource	A provider for Java Messaging Service.	
jmsresource	j2eeresource	A virtual class to all Java Messaging Service resources.	
jmsserver	jmsresource	Java Messaging Service server.	
JVM	j2eemanagedobject	Java Virtual Machine.	
lb	host	Load Balancer:distributes processing and communication activity evenly across a network so that no single device is overwhelmed.	

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
logdir	dir	Log directory definition.	file_path
logfile	file	Requested information which is kept in a file.	
logicaldisk	hostresource	A logical disk.	
lpar	host	Logical partition: the division of a computer's processors, memory, and storage into multiple sets of resources so that each set of resources can be operated independently with its own operating system instance and applications.	
mainframe	host	Main computer that can serve many users.	
marconiatmswitch	atmswitch	AMarconi ATM switch.	
memory	hostresource	Random access memory (RAM).	

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
messagedrivenbean	ejb	Enterprise JavaBeans in Sun's J2EE platform. Message driven beans are generated to process Java Messaging Service (JMS) messages.	
mibtree	hostresource	Management Information Base tree: a database of CIs that can be monitored by a management system. SNMP is a management protocol of MIB.	mibtree_oid
monitor	it_world	Monitoring classes.	
msdomain	networkresource	Represents Microsoft domains and workgroups in a Microsoft network.	
netdevice	host	A network device.	
netprinter	netdevice	A printer device that serves the network's users.	

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
network	networkresource	A series of elements or nodes connected to each other by communication paths.	network_domain, network_netmask, network_netaddr
networkresource	system	A virtual class to all network resources.	root_container, data_name
nt	host	Host with Microsoft operation system NT.	
service	hostresource	Software that can be controlled by the Microsoft Management Console (MMC).	
ntcmd	shell	xcmd or pstools. NT command line tools: allows you to execute applications on remote systems without installing any client software. Examples of such tools are xCmd and PsTools.	

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
oracle	database	Oracle database.	
oraclesystem	system	Oracle E-Business Suite.	oraclesystem_dbaddress
oracleappsresource	applicationresource	Oracle E-Business Suite Resource.	
oracleias	j2eeserver	Oracle Internet Application Server.	
osuser	hostresource	Operating system user.	
ospf	networkresource	Open Shortest Path First (OSPF) protocol: an interior gateway protocol used to distribute routing information within a single autonomous system.	ospf_areaid
port	networkresource	A place which being physically connected to another device.	port_number, root_container
printq	hostresource	A printer queue.	
printer	hostresource	A printer device.	
process	hostresource	An instance of a program.	

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
program	process	Name of an installed application or software.	data_name
ras	host	RAS (row address strobe) is a signal which being sent to a dynamic random access memory (DRAM Row Address Strobe: a clock signal in a dynamic random access memory (DRAM) chip used to pinpoint the row of a particular bit in a row-column matrix Remote Access Server	
router	host	A device or software connected to at least two networks. It decides the next network point to which to send the information packet.	

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
sap_application_component	sap_resource	A group of SAP transactions with common application components.	type
sap_app_server	application	SAP Application Server.	
sap_business_process	sap_resource	SAP Business Process..	
sap_business_scenario	sap_resource	SAP Business Scenario.	
sap_client	sap_resource	SAP Client.	
sap_gateway	application	SAP Gateway.	
sap_its_agate	application	SAP Internet Transaction Server AGate: an application gateway that establishes a link to the R/3 system and performs the processing tasks required to move data between an R/3 application and the Internet.	

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
sap_its_wgate	sap_resource	SAP Internet Transaction Server WGate: a web gatewaythat establishes a connection between Internet Transaction Server (ITS) and the web server. It forwards user requests to the AGate.	
sap_j2ee_app_server	sap_app_server	Represents an instance of the J2EE Application server.	
sap_j2ee_dispatcher	sap_resource	Receives a client request and forwards it to the server process with the lowest capacity usage.	
sap_j2ee_server_process	sap_resource	A single J2EE server process instance.	
sap_process_step	sap_resource	SAP Process Step.	
sap_bp_project	sap_resource	SAP Business Blueprint Project.	

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
sap_r3_server	sap_app_server	SAP's integrated software solution for client/server and distributed open systems.	
sap_resource	applicationresource	SAP Resource.	
sap_system	system	A logical unit that groups SAP-related entities and possibly other entities into one homogenous SAP deployment.	data_name
sap_transaction	sap_resource	A business process defined in the SAP System.	root_container, data_name
sap_transport	sap_resource	SAP Transport.	
sap_transport_change	sap_resource	SAP Transport Change.	
sap_work_process	sap_resource	A logical single-instance representation of a certain type of work process.	
ipserver	ipport	A known server port listening to or connected to a client port.	

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
oracleappservice	oracleappsresource	Oracle E-Business Suite Application Service.	
oracleappservicemanager	oracleappsresource	Oracle E-Business Suite Service Manager.	
servlet	j2eemanagedobject	A server-side Java program that provides additional features to the server.	
sessionbean	ejb	An Enterprise JavaBean session.	
shell	application	xcmd or pstools	
siebel_app_server	application	A server application running the Siebel business logic tier.	root_container, data_name
siebel_component	applicationresource	A process on the Siebel Application Server incorporating some Siebel application functionality.	alias

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
siebel_comp_grp	applicationresource	An collection of Siebel components running on the Siebel application server.	alias
siebel_gateway	application	A coordinating server that routes requests to the correct component.	
siebel_site	system	A logical unit describing a group of servers functioning together to build a complete Siebel toolset experience.	gateway_address, data_name
siebel_web_app	applicationresource	A Siebel application's location on the web server.	
siebel_wse	applicationresource	Web Server Extension: a supported add-on to a web server.	

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
snmp	agent	Simple Network Management Protocol: protocol used by network hosts to exchange information used in network management.	snmp_port
software	hostresource	The general name of a program.	data_name
application	system	A virtual class to all applications.	root_container, data_name
sqlalert	databaseresource	SQL Server Agent compares events to user-defined alerts. When the Agent finds a match, it fires an alert.	sqlalert_alertid
sqlbackup	databaseresource	A backup file generated by the system or by user requests.	
sqldatabase	databaseresource	A SQL database from Microsoft.	
sqlfile	databaseresource	A SQL file.	

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
sqljob	databaseresource	A specified series of operations performed sequentially by a SQL Server Agent. A job can perform a wide range of activities, including running Transact-SQL script, command line applications, and Microsoft ActiveX scripts.	sqljob_jobid
sqljobstep	databaseresource	An action that the job performs on a database or a server. Every job must have at least one job step.	sqljobstep_stepid
sqlperformancemonitor	databaseresource	Performance monitor constants that describe Microsoft Windows NT Performance Monitor polling behavior.	

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
sqlprocesses	applicationresource	SQL processes.	sqlprocesses_hostname, sqlprocesses_databasename, sqlprocesses_program
sqlserver	database	SQL server.	
ssh	shell	Secure Shell: an agent that provides secure encrypted communication between two untrusted hosts over an insecure network.	
statefulsessionbean	sessionbean	EJB stateful: an Enterprise JavaBean that is associated with one client at a time. It is assigned to a new client only after a previous client releases it or it times out.	
statelesssessionbean	sessionbean	EJB stateless: an Enterprise JavaBean that is context independent. It does not maintain a state between session calls.	

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
sunoneserver	webserver	Sun ONE Web Server.	
switch	host	A device that sends incoming data from any of multiple input ports to a specific output port in order to move the data toward its intended destination.	
switchrouter	switch	A switch that also acts as a router.	
sybase	database	Sybase.	
sybasedb	databaseresource	Sybase database.	
sysplex	networkresource	SYStem comPLEX: the multiprocessing capability of IBM MVS/ESA and OS/390 mainframes.	
system	it_world	System classes.	
system_monitor	monitor	Monitor system classes.	
ipport	networkresource	A logical connection between ports using TCP/IP Internet Protocol.	ipport_number

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
telnet	shell	A telnet connection for TCP/IP networks that allows you to execute programs on a remote computer.	
terminalserver	host	A device that provides terminals with a common connection using RS232 protocol.	
tomcat	webserver	A web server from the Apache Software Foundation that executes Java servlets and displays web pages that have embedded Java code.	
trail	networkresource	The Trail element describes the path from one network to another.	trail_destnetmask, trail_destnetaddr, trail_srcnetdomain, trail_srcnetaddr, trail_srcnetmask, trail_destnetdomain
unix	host	UNIX operating system.	

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
url	webresource	Uniform Resource Locator: the address of a resource on the Internet.	url_connectstring
vax	host	VAX: Virtual Address eXtension machine.	
vlan	networkresource	Virtual LAN: a network of computers that behave as if they are connected to the same wire even though they may be physically located on different segments of a LAN.	vlan_number
webapplication	j2eedeployedobject	Web application.	
oraclewebcomponent	oracleappsresource	Oracle E-Business Suite Web Component.	
webmodule	j2eemodule	J2EE deployed CI.	
webresource	system	A virtual class to all web resources.	root_container, data_name

CI TYPE NAME	PARENT	DESCRIPTION	KEY ATTRIBUTES
webserver	application	A web server.	
websphere	application	A WebSphere application server based on a Java 2 platform.	root_container, data_name
wmi	agent	Windows Management Instrumentation : a connection allowing scripts to monitor and control managed resources throughout the network.	
xterminal	host	A device using X11 emulation to connect to a UNIX machine.	
db2user	databaseresource	A user in a DB2 database.	

Permitted Relationships between CI Types

The following table lists all CI types defined under CIT system and their permitted relationship with other CI types. For more information about CI types, see “List of Parent, Description, and Key Attributes of all CI Types” on page 31. The definition of each relationship can be found in Relationship Definitions.

CI Type 1	CI Type 2	Relationship
application	applicationresource	container_f
atmport	atmport	pnniconnection
	interfaceindex	parent
	ip	uniconnection
atmswitch	atmswitching	container_f
atmswitching	atmport	container_f
bridge	port	container_f
	port	contains
chassis	elan	chassiselanmap
concentrator	port	layertwo
connectionpool	database	depend

CI Type 1	CI Type 2	Relationship
database	daemon	depend
	dbaobjects	container_f
	dbarchivefile	container_f
	dbcontrolfile	container_f
	dbdatafile	container_f
	dbjob	container_f
	dblinkobj	container_f
	dbredofile	container_f
	dbsegment	container_f
	dbsnapshot	container_f
	dbtablespace	container_f
	dbuser	container_f
	process	dbclient
	service	depend
dbdatafile	dbextent	container_f
dbindex	dbextent	resource
dblinkobj	database	dblink
dbsegment	dbextent	resource
dbsnapshot	dbjob	depend
	dblinkobj	resource
dbtable	dbextent	resource
dbtablespace	dbdatafile	resource
	dbindex	container_f
	dbsegment	resource
	dbtable	container_f

CI Type 1	CI Type 2	Relationship
dbuser	dbaobjects	owner
	dbextent	owner
	dbindex	owner
	dbjob	owner
	dblinkobj	owner
	dbsnapshot	owner
	dbtable	owner
dir	file	container_f
drive	drive	share
ejbmodule	ejb	container_f
elan	chassis	chassisvlanmap
	switch	bcastdomain
filesystem	filesystem	nfs

CI Type 1	CI Type 2	Relationship
host	application	container_f
	bridge	container_f
	chassis	contains
	host	backbone
	host	unnumbered
	host	talk
	host	dependency
	hostresource	container_f
	interface	container_f
	interfaceindex	container_f
	ip	virtual
	ip	contained
	ipport	container_f
	j2eeserver	deployed
	ospf	container_f
	port	layertwo
	port	container_f
	webresource	container_f
httpcontext	j2eeapplication	use
interface	interface	usb
	port	layertwo
interfaceindex	interface	parent

CI Type 1	CI Type 2	Relationship
ip	interface	parent
	interfaceindex	parent
	ip	depend
	ip	route
	ip	traffic
	ipserver	clientserver
ipport	ip	use
	ipport	tcp
	ipserver	clientserver
j2eeapplication	j2eemodule	container_f
	jdbcdatasource	use
j2eecluster	database	member
	j2eemanagedobject	container_f
	j2eeserver	member
	sap_j2ee_app_server	member
j2eedomain	j2eemanagedobject	container_f
	j2eeserver	member
j2eemanagedobject	jvm	use

CI Type 1	CI Type 2	Relationship
j2eeserver	executequeue	use
	ip	j2eesocket
	ip	depend
	ipport	depend
	j2eemanagedobject	deployed
	j2eemanagedobject	container_f
	j2eeserver	container_f
	jvm	use
jdbcdatasource	connectionpool	depend
	database	depend
jdbcprovider	jdbcdatasource	container_f
jmsserver	jmsdatastore	container_f
	jmsdestination	container_f
logdir	logfile	container_f
	logfile	parent
logicaldisk	dir	container_f
	disk	parent
	file	container_f
mainframe	lpar	member
	sysplex	member
mqchannel	mqchannel	mqmsglink
mqchclusrcvr	mqchclussdr	mqmsglink
mqchrcvr	mqchsdr	mqmsglink
	mqchsvr	mqmsglink

CI Type 1	CI Type 2	Relationship
mqchrqstr	mqchsdr	mqmsglink
	mqchsvr	mqmsglink
mqchsvrconn	mqchclntconn	mqmqilink
mqcluster	mqchannel	member
	mqqueue	member
	mqqueuemanager	mqrepository
	mqqueuemanager	member
mqmsgreceiverchannel	mqmsgsenderchannel	mqmsglink
mqqueue	mqaliasq	mqalias
	mqqueue	mqresolve
mqqueuelocal	mqqueueremote	mqresolve
mqqueuemanager	mqchannel	container_f
	mqqueue	container_f
mqqueueremote	mqxmitq	use
mqxmitq	mqchannel	mqchannelof
msdomain	host	member
network	host	member
	ip	member
nt	drive	container_f
	service	container_f
oracle	dbtns	container_f
oracleapplication	dbtablespace	use
	oracleapplication	depend
oracleappservice	process	resource
oracleappservicemanager	oracleappservice	member

CI Type 1	CI Type 2	Relationship
oracleias	oracleappsresource	deployed
	oracleappsresource	container_f
oraclesystem	oracle	member
	oracleappsresource	container_f
	oracleias	member
osuser	dir	container_f
port	bridge	contained
	bridge	bridgebackbone
	interfaceindex	parent
	port	backbone
	vlan	vlanmembership
printer	printq	parent
printq	printq	parent
process	daemon	brother
	file	use
	ipport	use
	printq	use
	service	brother
sap_application_component	sap_application_component	container_f
	sap_transaction	contains
sap_bp_project	sap_business_scenario	container_f
sap_business_process	sap_process_step	container_f
sap_business_scenario	sap_business_process	container_f
sap_its_agate	sap_its_wgate	depend

CI Type 1	CI Type 2	Relationship
sap_j2ee_server_process	j2eemanagedobject	deployed
sap_process_step	sap_transaction	execute
sap_r3_server	configfile	use
	sap_its_agate	depend
	sap_work_process	container_f
sap_system	database	depend
	host	sap_rfc_connect ion
	sap_gateway	member
	sap_r3_server	member
	sap_resource	container_f
sap_transport	sap_transaction	use
	sap_transport_change	container_f
sap_transport_change	sap_transaction	use
service	ipport	use
	service	depend
	sqlserver	depend
siebel_app_server	database	depend
	siebel_comp_grp	container_f
	siebel_gateway	depend
siebel_comp_grp	siebel_component	container_f
siebel_component	siebel_web_app	depend
siebel_gateway	siebel_wse	depend
siebel_site	host	contains
	siebel_app_server	member
	siebel_gateway	member

CI Type 1	CI Type 2	Relationship
siebel_wse	siebel_web_app	container_f
snmp	mibtree	container_f
sqldatabase	sqlalert	deployed
	sqlbackup	container_f
	sqlfile	container_f
	sqlperformancemonitor	container_f
	sqlprocesses	deployed
sqlfile	disk	depend
sqljob	sqljobstep	container_f
sqlprocesses	process	dbclient
sqlserver	sqlalert	container_f
	sqldatabase	container_f
	sqljob	container_f
	sqlperformancemonitor	container_f
	sqlprocesses	container_f
switch	vlan	container_f
sybase	sybasedb	container_f
sybasedb	process	dbclient
sysplex	lpar	member
unix	daemon	container_f
	filesystem	container_f
vlan	bridge	vlantobridge
	elan	elanvlanmap
webapplication	servlet	container_f

CI Type 1	CI Type 2	Relationship
webmodule	servlet	container_f
webserver	sap_its_wgate	container_f
	siebel_wse	container_f
websphere	j2eecluster	container_f
	j2eeserver	container_f
webspheremq	mqcluster	container_f
	mqqueuemanager	container_f

B

Relationship Definitions

This document defines relationships used throughout Mercury Business Availability Center documentation.

backbone

Represents a physical connection between two switches. The relationship is discovered by the Discover layer 2 service.

bcastdomain

The relationship between an elan (emulated LAN) and a switch.

bridgebackbone

Represents a physical connection between two switches connecting a switch port to a switch bridge. The relationship is discovered by the Discover base service.

brother

The relationship among elements who share the same parent.

chassiselanmap

The relationship between a chassis and an elan (emulated LAN).

chassisvlanmap

The relationship between a chassis and a vlan (virtual LAN).

clientserver

Represents a row of data from the tcpConnLocalAddress table in the Management Information Base (MIB) tree. This data has information about the TCP connection between two hosts' ports when you can differentiate between the server port and the client port. The tcpConnLocalAddress table is in the MIB address 1.3.6.1.2.1.6.13.1.2. The clientserver relationship is discovered by the Discover TCP connection collector.

contained

The relationship between two CIs whereby a second CI is included in the first CI. This relationship is found only between IP and host.

container_f

The functional relationship between a parent and a child. The child does not inherit any properties.

contains

The relationship between two CIs whereby a second CI is included in the first CI.

dbclient

The relationship between a process and a database.

dblink

The relationship between a database and a database link object.

depend

The relationship wherein one CI needs a functionality of another CI.

dependency

The relationship wherein one CI needs a functionality of another CI.

deployed

The relationship wherein one CI is put into action by another CI.

elanvlanmap

The relationship between elan (emulated LAN) and vlan (virtual LAN) network components.

execute

The relationship between an agent and the job it deploys.

j2eesocket

The relationship between a server and a remote client.

layertwo

Represents the physical connection between a switch and a host. It is discovered by the Discover layer 2 service.

member

The relationship between two CIs whereby one CI is included in another CI.

mqalias

The relationship between an alias queue and a local queue.

mqchannelof

The relationship between a channel and its transmission queue.

mqmqilink

The relationship between client and server channels for Message Queue Interface (MQI) calls.

mqmsglink

The relationship between two channels for message transfer.

mqrepository

The relationship between a message queue cluster and a message queue queue manager.

mqresolve

The relationship between a remote queue and the local queue to which it is mapped.

nfs

Network file server.

owner

The user of a resource.

parent

The relationship between elements where one element is parent of the other. For example, **ip** is the parent of **interface**.

pnniconnection

The relationship between two ATM ports.

resource

The relationship between elements where one element is the resource of the other. For example, **dbsnapshot** is the resource of **dblinkobject**.

route

Represents a row of data from the routing table in the Management Information Base (MIB) containing the data of the next_hop ip address and the destination network address. The routing table is in the MIB address 1.3.6.1.2.1.4.21.1.7. The relationship is discovered by the Discover route collector.

sap_rfc_connection

The relationship between a SAP system and a host. The host may be another SAP system or a non-SAP system.

share

The relationship between two drives.

talk

The relationship between two hosts.

tcp

Represents a row of data from the tcpConnLocalAddress table in the Management Information Base (MIB). This data has information about the TCP connection between two hosts' ports when you cannot differentiate between the server port and the client port. The tcpConnLocalAddress table is in the MIB address 1.3.6.1.2.1.6.13.1.2. The relationship is discovered by the Discover TCP connection collector.

traffic

Represents all network flow, regardless of protocol, between two IPs.

uniconnection

The relationship between an ATM port and an IP.

unnumbered

Represents a row of data from the routing table in the Management Information Base (MIB). This data has information about the next_hop IP address and the destination network address. The routing table is in the MIB address 1.3.6.1.2.1.4.21.1.7. Unnumbered relationships are discovered by the base collector.

usb

The relationship between two interfaces.

use

The relationship between elements whereby one element uses the other. For example, **process** uses **file**.

virtual

The relationship between a router and its virtual IP.

vlanmembership

The relationship between a vlan (virtual LAN) and a physical port.

vlantobridge

The relationship between a vlan (virtual LAN) and a bridge.

Index