# OPTIMIZE

## MERCURY BUSINESS AVAILABILITY CENTER™

### View Manager Administration

MERCURY™

BUSINESS TECHNOLOGY OPTIMIZATION

# Mercury Business Availability Center

## View Manager Administration

Version 6.2

Document Release Date: June 20, 2006

MERCURY™

Mercury Business Availability Center, Version 6.2
View Manager Administration

This manual, and the accompanying software and other documentation, is protected by U.S. and international copyright laws, and may be used only in accordance with the accompanying license agreement. Features of the software, and of other products and services of Mercury Interactive Corporation, may be covered by one or more of the following patents: United States: 5,511,185; 5,657,438; 5,701,139; 5,870,559; 5,958,008; 5,974,572; 6,137,782; 6,138,157; 6,144,962; 6,205,122; 6,237,006; 6,341,310; 6,360,332, 6,449,739; 6,470,383; 6,477,483; 6,549,944; 6,560,564; 6,564,342; 6,587,969; 6,631,408; 6,631,411; 6,633,912; 6,694,288; 6,738,813; 6,738,933; 6,754,701; 6,792,460 and 6,810,494. Australia: 763468 and 762554. Other patents pending. All rights reserved.

Mercury, Mercury Interactive, the Mercury logo, the Mercury Interactive logo, LoadRunner, WinRunner, SiteScope and TestDirector are trademarks of Mercury Interactive Corporation and may be registered in certain jurisdictions. The absence of a trademark from this list does not constitute a waiver of Mercury's intellectual property rights concerning that trademark.

All other company, brand and product names may be trademarks or registered trademarks of their respective holders. Mercury disclaims any responsibility for specifying which marks are owned by which companies or which organizations.

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

© 2005-2006 Mercury Interactive Corporation, All rights reserved

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

AMLIB_VMAD6.2/01

# Table of Contents

**Welcome to View Manager Administration** ..... **v**
How This Guide Is Organized ..... v
Who Should Read This Guide ..... vi
Getting More Information ..... vi

**PART I: INTRODUCTION**

**Chapter 1: Introduction to View Manager** ..... **3**
Understanding View Manager ..... 4
What Is a Service View? ..... 4
Instance Views and Pattern Views ..... 5

**Chapter 2: Working with View Manager** ..... **7**
The View Manager Tab ..... 8
Predefined Folders and Views ..... 12
Managing Folders ..... 16
Managing Views ..... 18
Creating and Editing Views ..... 19
Presentation Options for Views ..... 26
Activating or Deactivating Views ..... 39
Setting Permissions for Views ..... 40

**PART II: INSTANCE VIEWS**

**Chapter 3: Working with Instance Views** ..... **45**
What Is an Instance View? ..... 45
Creating a New View ..... 46
Editing an Instance View ..... 47

## PART III: PATTERN VIEWS

**Chapter 4: Working with Pattern Views ..... 51**
About Pattern Views ..... 52
Mercury Business Availability Center Topology Query System ..... 53
The Roles of TQL ..... 53
About Defining Pattern Views ..... 54
Pattern View Workflow ..... 54
Creating a Pattern View ..... 55
Understanding the Create View Page ..... 58
View Manager Tooltips ..... 60
Adding Nodes and Relationships to a TQL Query ..... 60
Setting TQL Node and Relationship Definitions ..... 69
Using the TQL Node Wizard ..... 81
Grouping CIs ..... 87
Straightening a Relationship With an Angle ..... 93
Adding Organization (Folding) Rules to Relationships ..... 94
Defining Multiple Relationship Rules ..... 95
Creating a Dependency Graph ..... 97
Showing TQL Node Instances ..... 99
Editing a Pattern View ..... 100
Copying and Pasting TQL Nodes and Relationships ..... 100
Saving the Pattern View ..... 101
Using the Toolbar Options ..... 101
Printing the Contents of the Topology Map ..... 101
Defining a View's Layout ..... 101
Understanding Layout Options ..... 101

**Index ..... 103**

# Welcome to View Manager Administration

This guide describes how to work with CMDB Administration View Manager.

## How This Guide Is Organized

The guide contains the following chapters:

### Part I Introduction

Introduces View Manager, service views, and pattern and instance view types and describes how to administer your views in View Manager. It also describes the default views and folders that come with the system and how they are used.

### Part II Instance Views

Describes how to create and edit instance views.

### Part III Pattern Views

Describes how to create and edit pattern views.

## 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 View Manager

This chapter introduces the View Manager, a part of CMDB Administration. View Manager enables you to create and administer service views in Mercury Business Availability Center.


| This chapter describes: | On page: |
|---|---|
| Understanding View Manager | 4 |
| What Is a Service View? | 4 |
| Instance Views and Pattern Views | 5 |

# Understanding View Manager

View Manager is a tool for creating and managing Mercury Business Availability Center service views. You can create your own views, or work with the out-of-the-box (factory) views provided with your Mercury Business Availability Center installation.

# What Is a Service View?

The IT universe model in the CMDB can be very large, holding thousands of configuration items (CIs). Each service view provides a subset of the overall IT universe model, containing CIs relating to a specific area of interest. You can define your own views to display only the information that is relevant to your organization's business needs.

The views are used to display performance status in Dashboard, and for setting up SLAs in Service Level Management. Mercury Business Availability Center supports two service view formats, pattern and instance views, which are each populated through different processes. For details, see "Instance Views and Pattern Views" on page 5.

Most of the out-of-the-box views included in View Manager are designed to work in coordination with the discovery packages in Discovery Manager, and the source adapters in Sources Manager. For details, see *Discovery Manager Administration* and *Source Manager Administration*.

For information on the out-of-the-box views, see "Predefined Folders and Views" on page 12.

# Instance Views and Pattern Views

Mercury Business Availability Center supports two service view formats, pattern and instance views, by which to populate your view:

- **Pattern views** are built from a TQL (Topology Query Language) query that defines the structure of the view. The view will only display those CIs and relationships that satisfy the query definition. When the view is opened up or refreshed, the View Manager queries the CMDB for all elements that fit the query, and automatically updates the view with those elements.

  All out-of-the-box views are pattern views, with the exception of the **First Service View**. For more information on pattern views, see Chapter 4, "Working with Pattern Views."

- **Instance views** are built by manually defining individual CIs and relationships directly in the view and/or by adding CIs that exist in the CMDB in the view. View Manager includes one out-of-the-box, empty instance view, the **First Service View**.

  For more information on instance views, see Chapter 3, "Working with Instance Views"

  You can also create new views (both instance and pattern views) by cloning an existing view, and editing the view contents (for details, see "Views Pane" on page 9).

# 2

# Working with View Manager

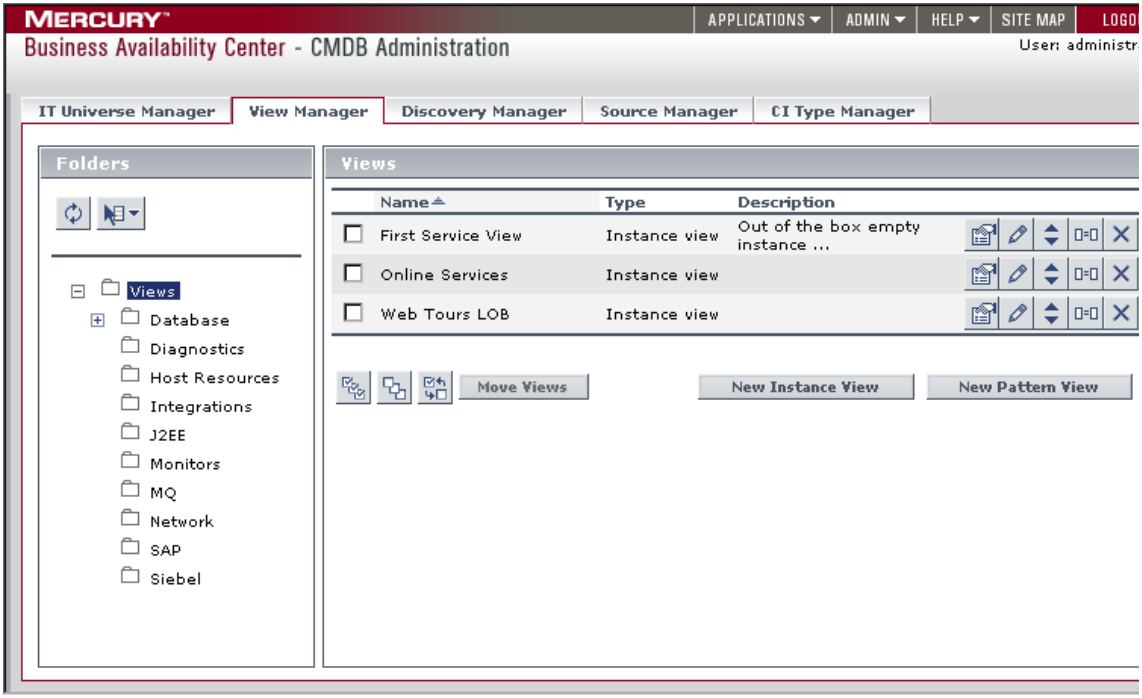
This chapter explains how to use View Manager to manage and organize the views and view folders in Mercury Business Availability Center.


| This chapter describes: | On page: |
|---|---|
| The View Manager Tab | 8 |
| Predefined Folders and Views | 12 |
| Managing Folders | 16 |
| Managing Views | 18 |
| Creating and Editing Views | 19 |
| Presentation Options for Views | 26 |
| Activating or Deactivating Views | 39 |
| Setting Permissions for Views | 40 |

# The View Manager Tab

The View Manager tab allows you to manage and organize the views in your system. Your Mercury Business Availability Center installation provides a predefined hierarchical folder structure to help you organize the views.

The View Manager tab (**Admin** > **CMDB** > **View Manager**) contains two panes: the **Folders** pane on the left, and the **Views** pane on the right.




The View Manager panes provide the following functionality:

### Folder Pane

The Folders pane displays a hierarchical structure of predefined folders containing views. The views in the selected folder are displayed in the Views pane.

The **Views** folder is the root folder. The **Views** folder has subfolders containing the out-of-the-box (factory) views that come with the Mercury Business Availability Center installation. The factory views work with Discovery Manager and Monitor Administration/Sources Manager. For more information, see "Setting Permissions for Views" on page 40.

In the Folders pane, you can create, rename, or delete a folder. For details, see "Managing Folders" on page 16.

### Views Pane

The Views pane contains the views table, displaying a list of the views contained in the selected folder in the Folders pane. The following information is provided for each view:

- **Name** – the view name
- **Type** – the view type (instance or pattern view).
- **Description** – the description of the view.

If required, you can edit the name and description of the view. For details, see "Creating and Editing Views" on page 19.

The following table describes the buttons that appear in the views table.

| Button Name | Button | Description |
|---|---|---|
| Edit view properties |  | Edit the view's properties.<br>For information on how to edit a view's properties, see "Creating and Editing Views" on page 19. |
| Edit view |  | Edit the view:<br>➤ If you are editing an instance view, the Edit View page for instance views opens. For information on editing an instance view, see Chapter 3, "Working with Instance Views".<br>➤ If you are editing a pattern view, the the Edit View page for pattern views opens. For information on editing a pattern view, see Chapter 4, "Working with Pattern Views". |
| Manually change order in view |  | Manually change the order of the CIs in a a view. For details, see "Changing the Order of CIs in a View" on page 18. |
| Clone view |  | Clone an existing view. A copy of the view appears in the Views pane. It is named: **Copy of <view>**.<br>You can change the name and description of the view in the Edit view properties dialog box and then modify the view as required. |

| Button Name | Button | Description |
|---|---|---|
| Delete view |  | Delete the view. You can delete all instance views, but there are some pattern views basic to the system that Mercury Business Availability Center does not allow you to delete. For example, **Integrations**, **Monitors**, and **Network** views. |
| Display view |  | Display the view in a new dialog box. Opens the Display View dialog box, which provides an example how the view appears in View Explorer. The view cannot be edited from the Display View dialog box; however, all standard View Explorer browse and search functionality is provided (as described in "Using View Explorer" in *Working with the CMDB*). |

In addition, the Views pane contains the following buttons:

- **Move Views –** enables you to move multiple views to a new folder. For details, see "Moving a View to a Different Folder" on page 18.
- **New Instance View/New Pattern View** – enables you to create a new view. For details, see "Creating and Editing Views" on page 19.

# Predefined Folders and Views

Your Mercury Business Availability Center installation includes folders and views that are ready to work with. The following topics describe the folders and views, and how they are used.


- About Predefined Folders – for details, see page 12
- About Factory Views – for details, see page 12
- Folder and View List – for details, see page 14


## About Predefined Folders

View Manager contains a hierarchical file folder system to help you organize your views, displayed in the Folders pane. Predefined folders are provided for the out-of-the-box views; you can modify these folders, or add additional folders according to your organization's requirements (for details, see "Managing Folders" on page 16).

## About Factory Views

Your Mercury Business Availability Center installation includes out-of-the-box pattern views, predefined to work with Discovery Manager and Monitor Administration/Source Manager. These views provide a quick start for working with the IT universe model and for setting up SLAs, and for viewing system performance information in Dashboard.

The out-of-the-box views are initially empty. As soon as CIs are added to the CMDB by one of the CI generators, the TQL query built into each pattern view is able to identify the CIs that meet the pattern requirements, and add them to the view.

There are two main types of out-of-the-box pattern views:

- Pattern views for CIs created during a discovery process; these views correspond to the relevant discovery packages. For example, Mercury Business Availability Center provides various out-of-the-box SAP views for SAP CIs; these CIs are created when Discovery Manager runs the discovery processes for a SAP package. For details on running discovery processes, see *Discovery Manager Administration*.
- Pattern views for CIs created by source adapters; these views correspond to the relevant source adapter types. Mercury Business Availability Center provides various out-of-the-box Monitors views for CIs from all the default sources (SiteScope, Business Process Monitor, Real User Monitor); and out-of-the-box Integrations views that each receive CIs from the relevant custom source. For example, the CA Unicenter view corresponds to the CA Unicenter source in Source Manager. When a CA Unicenter integration monitor is defined in Monitor Administration, and the CA Unicenter source adapter is set up in Source Manager, the CA Unicenter view is populated with the created CIs.

  For details on source adapters, see *Source Manager Administration*.

  There are also out-of-the-box views for CIs created for internal Mercury Business Availability Center data, for example, the BAC Self-Alert View and the UDX Measurement Filters View. Complex view details are provided in *Using Dashboard*.

  There is one out-of-the-box instance view included with your Mercury Business Availability Center package, the **First Service View**. This is an empty view, supplied as a quick start for creating your own instance view.

  The First Service View and many out-of-the-box Monitors views are set by default to display in all relevant Mercury Business Availability Center applications; other out-of-the-box views are hidden by default. For details on displaying or hiding the views, see "Creating and Editing Views" on page 19.

  Some of the out-of-the-box pattern views are considered basic to the system and cannot be deleted from the folders. The following section specifies which views cannot be deleted.

### Folder and View List

The out-of-the-box views are contained in the following predefined folders:

- **Views –** Top level folder, containing all other view folders and the First Service View.
- **Application** – contains the Applications Deployment view, and the following subfolders:
  - **MQ** – contains various views for WebSphere MQ; these views display results for a discovery task that looks for all MQ applications in the system.
  - **Oracle E-Business Suite** – contains the OracleApps_Topology view, which displays the results of a discovery task that looks for the E-Business Suite application in the system.
  - **SAP** – contains various views for SAP; these views display results for a discovery task that looks for all SAP applications in the system.
  - **Siebel** – contains views for Siebel topologies; these views display results for a discovery task that looks for all Siebel applications in the system.
- **Database** – contains the following subfolders:
  - **DB2** – contains the DB2 view, which displays the results of a discovery task that discovers DB2 databases in the system.
  - **Oracle –** contains the Oracle view, which displays the results of a discovery task that looks for all Oracle databases in the system.
  - **SQL Server –** contains the SQL Server view, which displays the results of a discovery task that looks for all SQL servers in the system.
- **Diagnostics** – contains the Diagnostics view, which displays a hierarchy for Mercury Diagnostics information generated by a custom source adapter.
- **Host Resources** – contains the Host Resources view, which displays the results of a discovery task that discovers data for hosts in the system.
- **Integrations** – contains various views for integration monitors; these views display hierarchies generated by the custom source adapters. The Integrations views cannot be deleted.

- **J2EE** – contains the following subfolders:
  - **Apache –** contains the Apache_Topology view, which displays the results of a discovery task that discovers Apache servers in the system.
  - **Weblogic** – contains the Weblogic_Topology view, which displays the results of a discovery task that discovers J2EE WebLogic servers and components in the system.
  - **Websphere** – contains views for J2EE WebSphere; these views display results for discovery tasks that discover J2EE WebSphere servers and components in the system.
- **Mainframe** – contains the Mainframe view, which displays the results of a discovery task that discovers information on mainframe systems.
- **Monitors** – contains various views for the Mercury Business Availability Center monitors; these views display hierarchies generated by the default source adapters for Business Process Monitoring, SiteScope, and Real User Monitor. Also contains the BAC Self-Alert View, for alerts generated within Mercury Business Availability Center, and the Deep Transaction Tracing View if the Deep Transaction Tracing package (for Bristol integration) is installed.

  Some of the views in the Monitors folder cannot be deleted.

  Contains the following subfolders:
  - **Real User Monitoring Views** – contains views for Real User Monitor information, generated by the Real User Monitor source adapter.
  - **UDX Measurement Filters** – contains the UDX Measurement Filters View, which displays UDX monitors created in Mercury Business Availability Center.
- **Network** – contains various views for network packages; these views display results for discovery tasks that build a network map of the system.

  Contains the following subfolder:
  - **Layer2** – contains the Layer2 view, which displays results for a discovery task that discovers information on catalyst Layer 2 switches in the system.
- **Service Catalog** – contains views used in the Service Level Management application.

- **SOA** – contains the Web Services view, which displays a hierarchy for your SOA environment, including UDDI business services, Web services, and the underlying infrastructure.
- **Webservices** – contains the UDDI view, which displays the results of a discovery task that discovers UDDI Registry components in the system.

# Managing Folders

View Manager allows you to create, rename, and delete folders in the Folders pane. This section contains the following topics:


- "Creating a New Folder" on page 16
- "Renaming a Folder" on page 17
- "Deleting a Folder" on page 17
- "Refreshing the Display of Folders" on page 17


## Creating a New Folder

This section describes how to create new folders in the Folders pane. You can create additional folders in the View Manager to organize the new view that you create for your organization. For example, you might add a folder requested by a customer or for a line of business views.

**To create a new folder:**

1. Select the folder to which you want to add a new folder.
2.  Right-click the folder or click the **Menu options** button and select **New Folder** to open the Create Folder dialog box.
3. Type the folder name.
4. Click **OK**.

### Renaming a Folder

This section describes how to rename a folder.

**To rename a folder:**

1. Select the folder that you want to rename.
2.  Right-click the folder or click the **Menu options** button and select **Rename Folder** to open the Rename Folder dialog box.
3. Rename the folder as required.
4. Click **OK**.

### Deleting a Folder

This section describes how to delete a folder.

**To delete a folder:**

1. Select the folder that you want to delete.
2.  Right-click the folder or click the **Menu options** button and select **Delete Folder**. A message appears asking you whether you are sure you want to delete the selected folder.
3. Click **OK**.

---

**Note:** You cannot delete a folder that contains a subfolder or a view.

---

### Refreshing the Display of Folders

This section describes how to update the display of folders in the Folders pane. This may be required if, for example, a folder was created by another user, and the new folder is not yet displayed.

**To update the display of folders:**

 Click the **Refresh** button.

# Managing Views

View Manager allows you to organize and manage your views. This section has the following topics:


- "Moving a View to a Different Folder" on page 18
- "Changing the Order of CIs in a View" on page 18


## Moving a View to a Different Folder

You can move one or more views to a different folder.

**To move a view to a different folder:**

1 In the Views pane, select the check box for each view you want to move.

---



**Tip:** To make your selections, you can also use the buttons at the bottom of the pane for, **Select All**, **Clear All**, and **Invert Selection**.

---




2 Click the **Move Views** button to open the Move Views to Selected Folder dialog box.

3 Select the folder to which you want to move the view(s).

4 Click **OK** to move the view to the selected folder.

## Changing the Order of CIs in a View

You can manually change the order of CIs in a view. After changing the position of a CI, you can reset the order to the way it was before making your changes.

**To manually move a CI to a new position within a view:**



1 In the Views pane, click the **manually change order in view** button to open the Manual Ordering dialog box.

2 Select the CI you want to move.

3 Use the **Up** and **Down** arrows to move the CI to a new position.

- If you select a CI containing one or more child CIs, the CI is moved together with all its children CIs to the new location.
- If you select the child of a CI, the child CI is moved to the new location under the parent CI.

4 To reset the order of the CIs to the way it was before manually changing the order, click the **Reset** button. The Reset Ordering dialog box opens.




- Select <**CI**>**'s direct child CI** to reset the order only of the child CIs that were rearranged directly under the CI currently selected.
- Select **Entire View** to reset the order of all the CIs that were rearranged in the view.

5 Click **Reset**.

6 Click **OK** in the Manual Ordering dialog box to save the changes you have made.

## Creating and Editing Views

There are two stages to the process of creating a new view: First you define the properties to create the view framework; then you build the view by adding CIs (for instance views), or by defining a TQL query (for pattern views).

This section deals with the first stage, which involves defining the name and display properties for the view. You create and edit view properties from the View Manager tab. For instance views only, you can also create a new view from the IT Universe Manager tab. (For more information, see "Creating a New View" in *IT Universe Manager Administration*.)

For details on the second stage, building the view, see "Adding CIs to the Created View" on page 46 or "Working with Pattern Views" on page 51.

This section includes the following topics:

- "Defining View Properties" on page 20
- "KPI Selection" on page 25

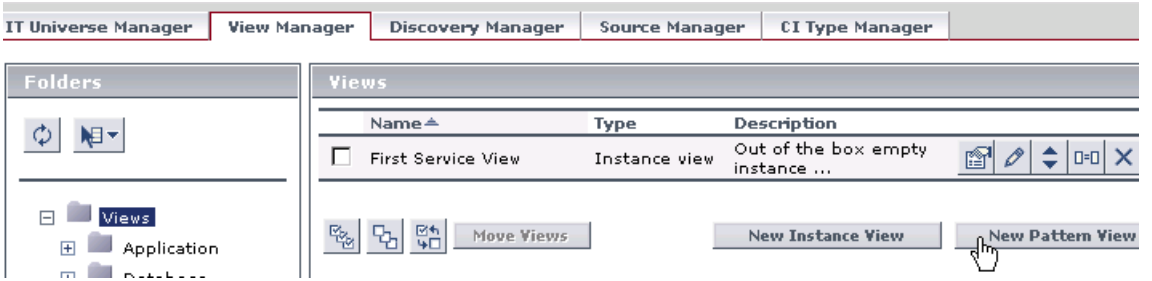
### Defining View Properties

When defining or editing view properties, the following presentation information is defined for the view:

- You can select to hide the view in the IT Universe Manager, Dashboard, and Service Level Management applications (if, for example, you do not want the view to be seen by the application users until you have finished building it). When you use this option, the full view is still built in the CMDB, but it is hidden in the appropriate application.
- You can limit the KPIs displayed for the view in Dashboard. For more information, see "KPI Selection" on page 25.
- You can define how folding should be applied for the view. For more information on the folding options, see "Presentation Options for Views" on page 26.
- You can select to deactivate a view if you do not currently require it in Mercury Business Availability Center. (This option is enabled only while editing a view). For more information, see "Activating or Deactivating Views" on page 39.
- For pattern views, you can select to merge multiple instances of CIs in the view. For more information, see "Merging Identical Instances" on page 56.

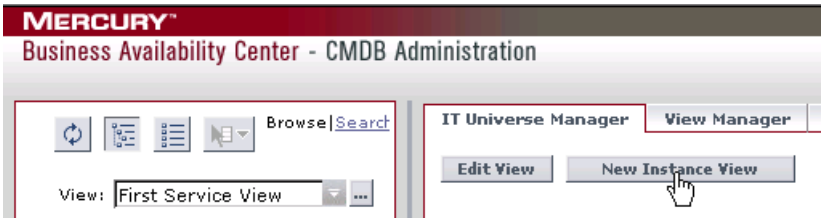
**To define/edit view properties:**

**1** If you want to create a new view: In the **View Manager** tab, select the folder in the **Folders** pane in which you want to create the view, then click either **New Instance View** or **New Pattern View**, according to the view type you want to create.

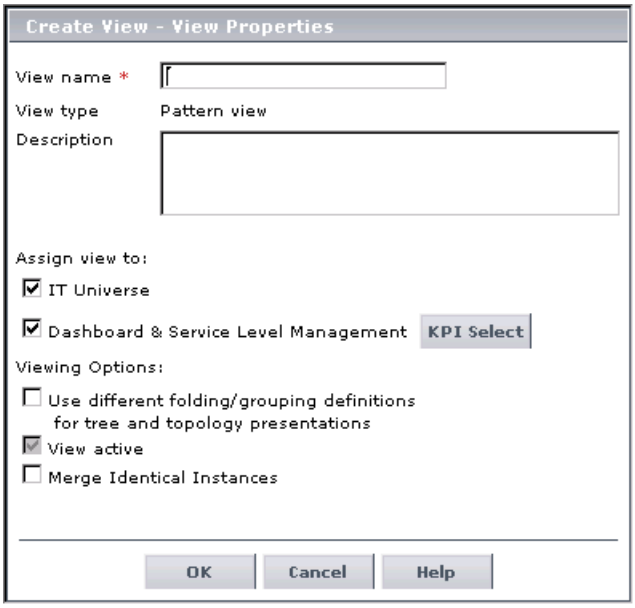



Or,

In the **IT Universe Manager** tab, click the **New Instance View** button at the top of the tab.



The Create View - View Properties dialog box opens.




Skip to step 3.

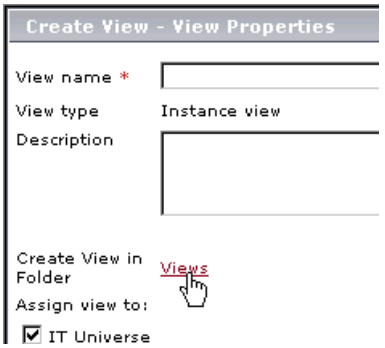
**2** If you want to edit the properties for an existing view: In the **View Manager** tab, click the **Edit view properties** button for the relevant view. The Edit View Properties dialog box opens.

**3** Type a unique name for the view in the **View name** box. (This is a mandatory field.)

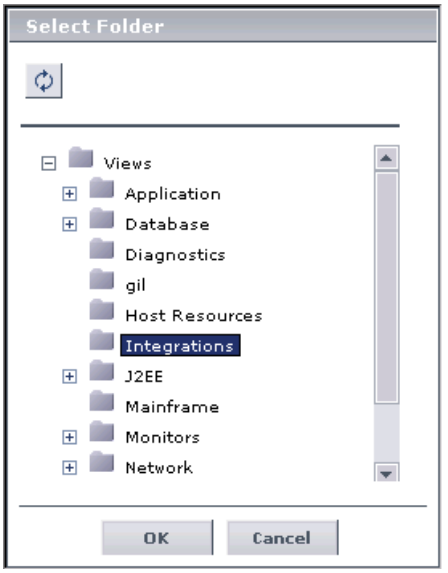
**4** If required, enter a description of the view in the **Description** box.

**5** (Only relevant when creating a new instance view from the IT Universe Manager tab.) The **Create View in Folder** parameter displays the name of the views folder in which the new view will be placed. This is the folder that contains the view currently selected in IT Universe Manager.

If you want to select a different folder for the view, click the folder name.




In the displayed Select Folder dialog box, select the required folder and click **OK**.




The selected folder is displayed in the Create View - View Properties dialog box.

---

**Note:** After creating a view, you can move it to a different folder at any time. For details, see "Moving a View to a Different Folder" on page 18.

---

**6** Under **Assign view to**, select the check boxes for the applications in which you want the view you are creating to be visible:

- **IT Universe**
- **Dashboard & Service Level Management**

**7** If you want to exclude certain KPIs from the view in Dashboard, click the **KPI Select** button (to the right of the **Dashboard & Service Level Management** option). In the displayed KPI Selection dialog box, deselect the check boxes of the KPIs that you do not require.

For more information, see "KPI Selection" on page 25.

**8** Under **Viewing Options**, select the check box for **Use different folding/grouping definitions for tree and topology presentations** if you want and folding and grouping definitions to be separate for the tree and topology a presentations of the view.

For more information, see "Using the Different Folding/Grouping Definitions Option" on page 27.

**9** The check box for **View active** is selected by default, and disabled when creating a new view. When editing a view, you can deselect the check box if the view is not currently required for use in Mercury Business Availability Center.

For more information, see "Activating or Deactivating Views" on page 39.

**10** (For pattern views only) Select the **Merge identical instances** check box if you want to merge multiple instances of CIs so they only appear once in the view. For details, see "Merging Identical Instances" on page 56.

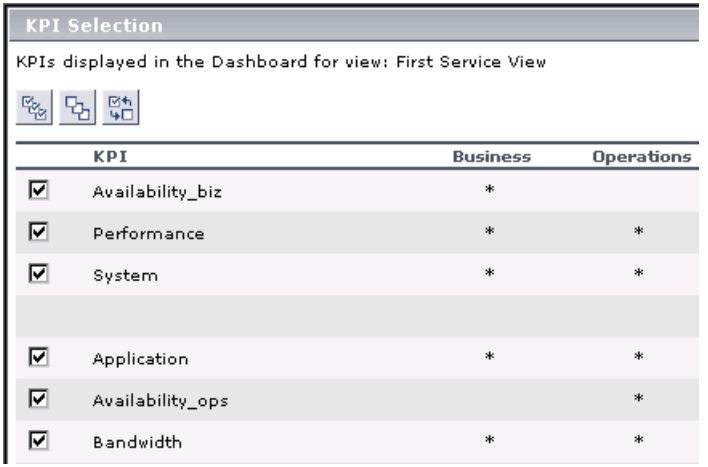
**11** Click **OK**.

If you are creating a new view, you continue directly to the building view stage, by defining the CIs or TQL for the view. For details on building instance views, see "Adding CIs to the Created View" on page 46; for details on building TQLs, see "Working with Pattern Views" on page 51.

## KPI Selection

The KPI Selection dialog box is accessed from the View Properties dialog box from the view (described in "Creating and Editing Views" on page 19). The KPI Selection dialog box enables you to limit the KPIs that are displayed for a view in Dashboard. The dialog box contains all KPIs that are defined in the Dashboard Repositories (**Admin** > **Dashboard** > **Repositories** > **KPIs**) regardless of whether these KPIs are used by any CIs in the view. By default, all KPIs are selected to appear in the view.

The dialog box also shows for each KPI if it is assigned to business or operations user mode, or if it is used for both modes.




If you select to display in Dashboard a KPI that is assigned to only one of the user modes (as is the case for **Availability_biz** and **Availability_ops** in the picture above), that KPI will only be seen in the view by users with the appropriate user mode setting. For more information on user modes, see "KPIs for User Modes" in Dashboard Administration.

Status in Dashboard is calculated based on visible KPIs only. This means that any KPI that is excluded from a view in Dashboard will have no influence on the overall status calculations made for CIs or locations in the Top View, Geographical Map, Custom Map, or Topology Map tabs.

**To limit the KPIs that appear in the view:**

1 Open the KPI Selection dialog box by clicking **KPI Select** in the View Properties dialog box for the view.

2 Select or deselect the KPI check boxes, according to what you want to be displayed in the view.

---



**Tip:** To make your selections, you can also use the buttons at the top of the dialog box for, **Select All**, **Clear All**, and **Invert Selection**.

---

3 Click **OK**.

## Presentation Options for Views

Mercury Business Availability Center enables you to define grouping and folding options to apply to the presentation layer of your view. These options enable you to organize the way the view is presented throughout Mercury Business Availability Center.

If required, you can define different presentation options for when the view is presented in tree format or topology format.

This section includes the following topics:

➤ "Using the Different Folding/Grouping Definitions Option" on page 27

➤ "Grouping Options (Pattern Views Only)" on page 29

➤ "Folding Options" on page 32

### Using the Different Folding/Grouping Definitions Option

When defining or editing view properties (as described in "Creating and Editing Views" on page 19), you can define whether different folding/grouping definitions should be used for the topology and tree presentations of the view.

- The topology presentation refers to the way the view is shown in the IT Universe Manager and Dashboard Topology Maps.
- The tree presentation refers to the way the view is shown in the tree formats, as used, for example, in View Explorer and in the Dashboard Console.

When the **Use different folding/grouping definitions** option is not selected, the organization and layout of the view are the same for both presentation types.

When the option is selected, the organization and layout of the view can be different in the topology presentation and in the tree presentation. The application of this depends on whether the view is an instance view or a pattern view:

- For instance views, the difference is only in the folding used for the topology presentation, as described in "Topology Presentation" on page 33.
- For pattern views, different folding/grouping enables you to define different group-by options (see "Grouping Options (Pattern Views Only)" on page 29), and different folding rules (see "Using Folding Rules (Pattern Views Only)" on page 35) for each presentation type.

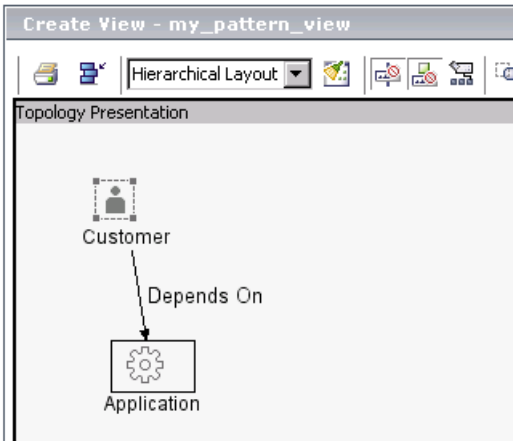
  Different folding definitions for pattern views is discussed in the following section.

Applying different folding definitions for a view means that two separate versions of the view (topology and tree) are stored in memory, each one containing the appropriate definitions. Maintaining two versions of each view puts pressure on the system, so you should apply this option carefully.

### Defining Pattern View Properties with Different Folding Definitions

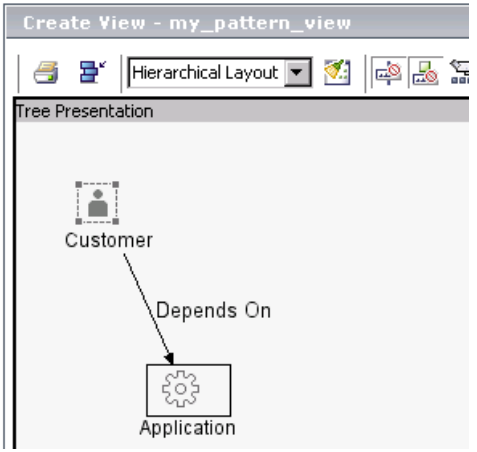
Selecting the **Use different folding/grouping definitions** option changes the editing pane in the Create View/Edit View pages for pattern views (described in "Understanding the Create View Page" on page 58). When the option is selected, there are two versions of the editing pane, one for the topology presentation, and one for the tree presentation. (When the option is not selected, there is one editing pane for the overall view.)

When you access the Create View/Edit View page, you are in the **Topology Presentation** editing pane.





Use the **Switch between topology and tree presentations** button to toggle between the two presentation types.



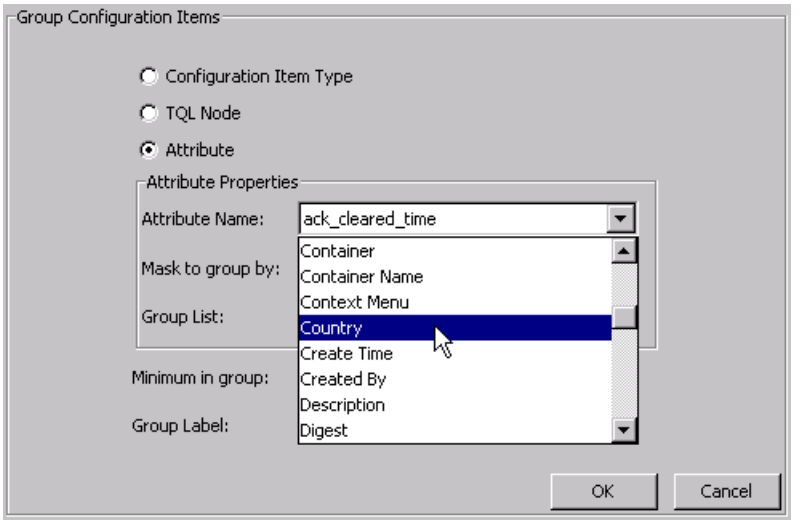

Each editing pane contains all editing options, so that you can set different group-by and folding rule definitions for each presentation. For information on defining group-by options, see "Grouping Options (Pattern Views Only)" on page 29. For information on defining folding rules, see "Using Folding Rules (Pattern Views Only)" on page 35.

## Grouping Options (Pattern Views Only)

Grouping options organize the CIs in a pattern view according to certain criteria, for example, by CIT or by a particular attribute. You define grouping options for a pattern view when creating or editing the TQL for the view, as described in "Grouping CIs" on page 87.

You define the grouping options per node, so that all CIs that relate to that node are organized according to those grouping options. After defining grouping options for a node in the TQL, the view organization changes to display the relevant CIs under the group containers.

For example: A grouping option is defined for TQL node **Application**, so that it will be grouped by attribute **Country**:

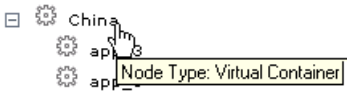



As a result of this definition, all Application CIs included in the view (because they correspond to the Application TQL node) are organized into groups, according to the country defined for each Application CI in its properties.



Each country name (Albania, China, USA) in the above example is a **Virtual Container** node type. Virtual Container nodes are not CIs; they are not part of the IT universe model in the CMDB, and they have no properties of their own. They function simply as an organizational container in the view for the defined groupings.

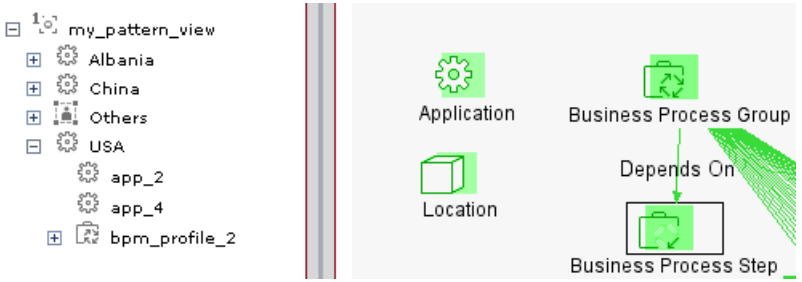
The icon displayed for a Virtual Container node is the icon used for the CIs organized by the Virtual Container. So for the countries grouping, the Virtual Container nodes display the Application CI icon.




### Different Grouping Definitions for Topology and Tree Presentations

Grouping options can apply for the overall view, or different grouping options can be defined to apply for the topology presentations and tree presentations of the view. To define different grouping options for topology and tree presentations, you select the **Use different folding/grouping definitions** option in the view properties. For details, see “Using the Different Folding/Grouping Definitions Option” on page 27.

In the following example, all nodes are grouped by attribute **Country** in the tree presentation for the view, and by **CIT** in the topology presentation for the view.




When different grouping is used for the tree and topology presentations, there may not be correlation between the presentations, so that an item in the tree presentation does not have a corresponding item in the topology presentation, or vice versa.

## Folding Options

Folding refers to how related CIs are displayed in the Mercury Business Availability Center views. When folding is used, the CIs are displayed on different levels; when folding is not used, the CIs are shown on the same level. For example, for a parent-child relationship between CIs with folding, the child CI is shown on a hierarchical level below the parent.

The default folding settings works differently for the tree and topology presentations of the view, as described in the following sections. In addition, folding rules can be defined that change the organizational structure for the presentation layer of the view, as described in "Using Folding Rules (Pattern Views Only)" on page 35.

This section includes the following topics:

➤ "Tree Presentation" on page 32

➤ "Topology Presentation" on page 33

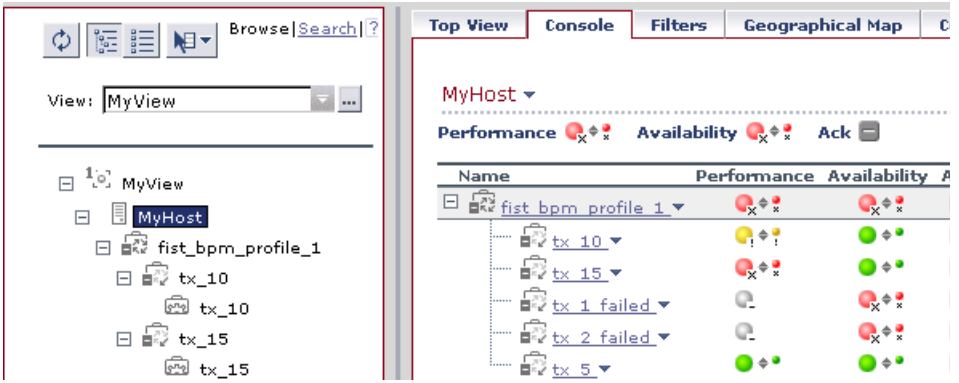
➤ "Using Folding Rules (Pattern Views Only)" on page 35

### Tree Presentation

The tree presentation for a view (as seen, for example, in View Explorer and in the Dashboard Console) always shows the CIs with folding, for both pattern and instance views, regardless of what folding options are set for the view.

The folding is determined by the hierarchical structure defined for the CIs in the view. You move down through the levels to see the CIs on each lower branch.

In the following picture, both panes show the view in tree presentation, using folding.

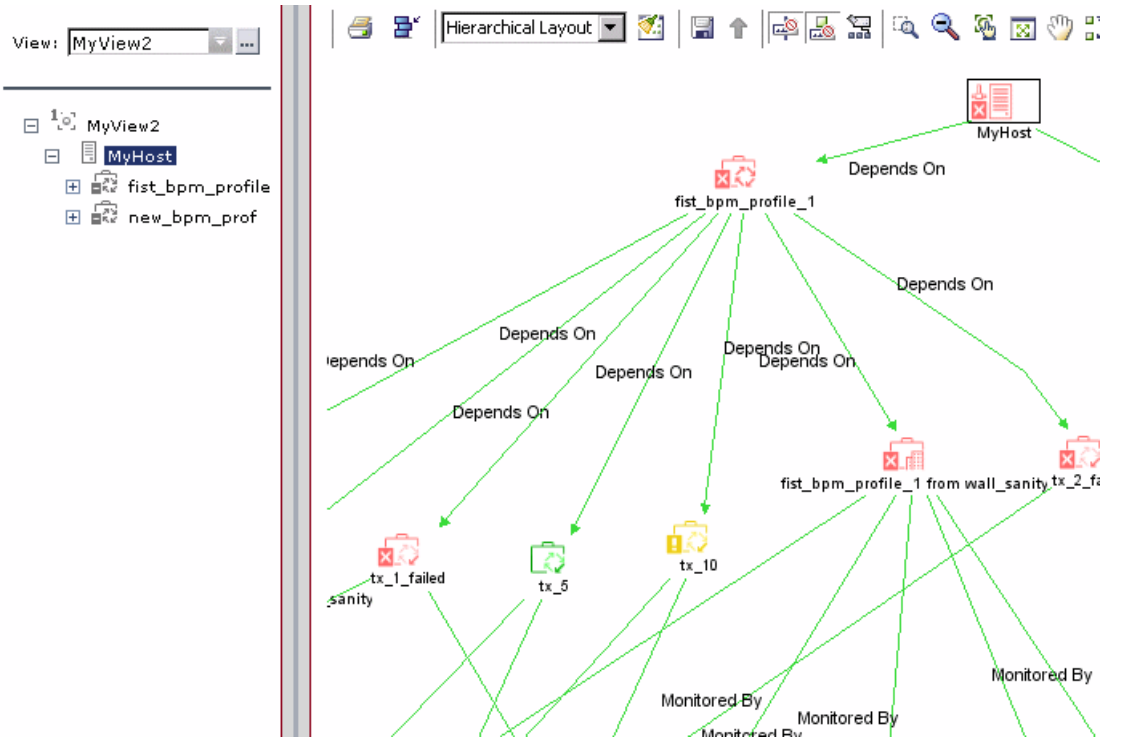



### Topology Presentation

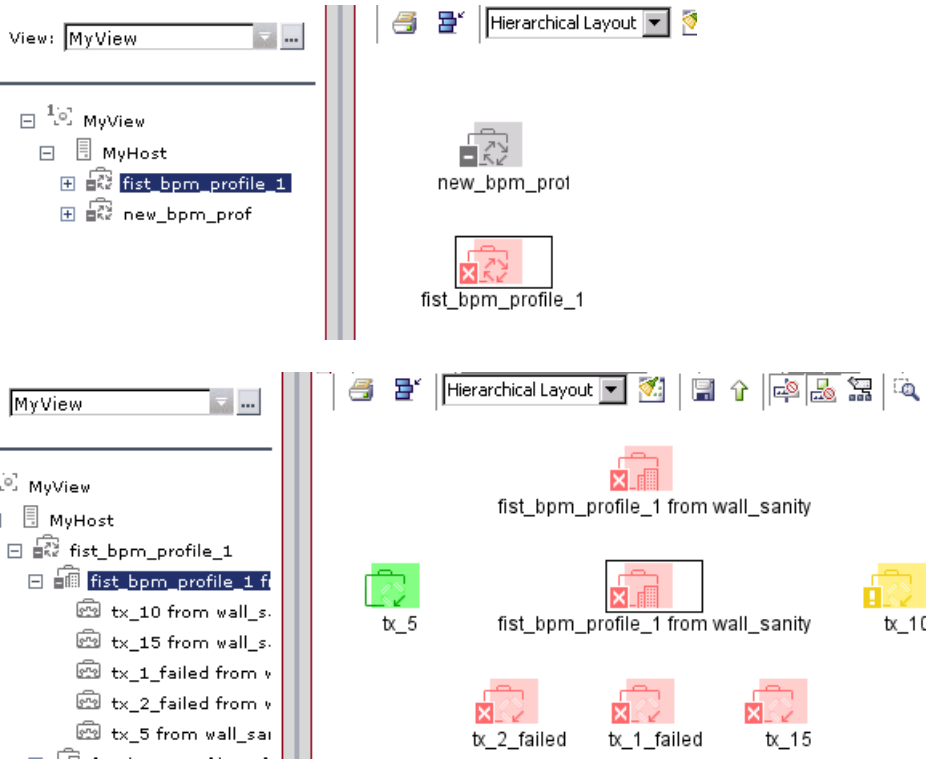
The use of folding in the topology presentation for a view (as seen in the IT Universe Manager Topology Map and Dashboard Topology Map) depends on whether the **Use different folding/grouping definitions** option is selected for the view in the view properties. (For information on this option, see “Using the Different Folding/Grouping Definitions Option” on page 27.)

The **Use different folding/grouping definitions** option influences topology presentation for both pattern and instance views as follows:

➤ By default, the topology presentation for a view does not use folding. Therefore, if the **Use different folding/grouping definitions** option is selected (meaning that the tree and topology presentations each use their own settings), the topology map uses its default setting of no folding, and the CIs are shown on one level. The CIs are connected by extended arrows, representing the relationships between them.



➤ If the **Use different folding/grouping definitions** option is not selected for the view, the topology presentation uses the same folding definitions as the tree presentation, so folding is used. If an icon is displayed with a shadow in the topology map, it means that there is an additional CI layer beneath it, and you drill down to see the underlying layer of CIs.




### Using Folding Rules (Pattern Views Only)

You can define folding rules for a pattern view when creating or editing the TQL for the view, as described in “Adding Organization (Folding) Rules to Relationships” on page 94. The folding rules are defined per relationship in the view.

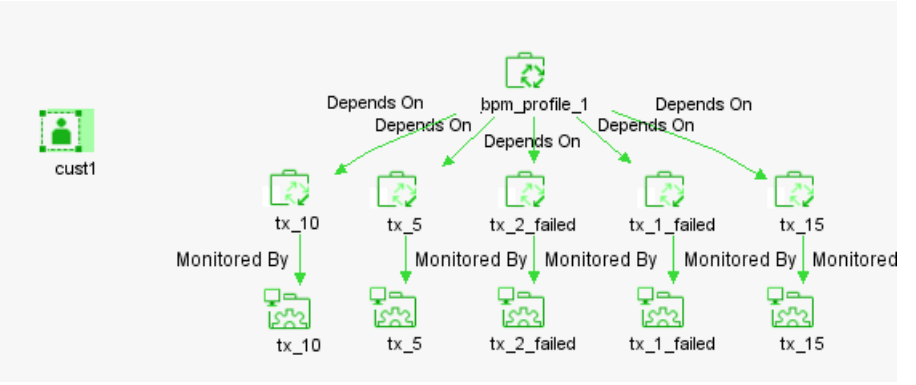
If the view is set to use different folding definitions for the topology and tree presentations, then you can define different folding rules for each presentation. For more information on using this option, see “Using the Different Folding/Grouping Definitions Option” on page 27.

The **Right Sibling** and **Left Sibling** folding rules have no effect on the tree or topology presentations of the view.

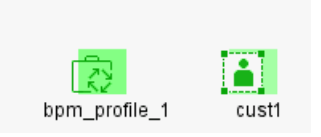
The **Parent** and **Child** folding rules override the general folding definitions for the relationship and, in some cases, may change the organizational structure for the topology and/or tree presentation in IT Universe Manager and Dashboard.

The Parent/Child folding rules influence the view as follows:

- If you define a **Parent** or **Child** rule on a relationship for the topology presentation (when using different folding definitions, so the CIs are shown by default on one level)—then the display of the CIs for that relationship changes, so that they are shown with folding. In the following example, a Parent rule is defined for the relationship between cust1 and its child CIs.




- If you define a **Parent** rule on a relationship, when the CIs are shown with folding, the rule has no effect on the display of the CIs in the topology or tree presentations.



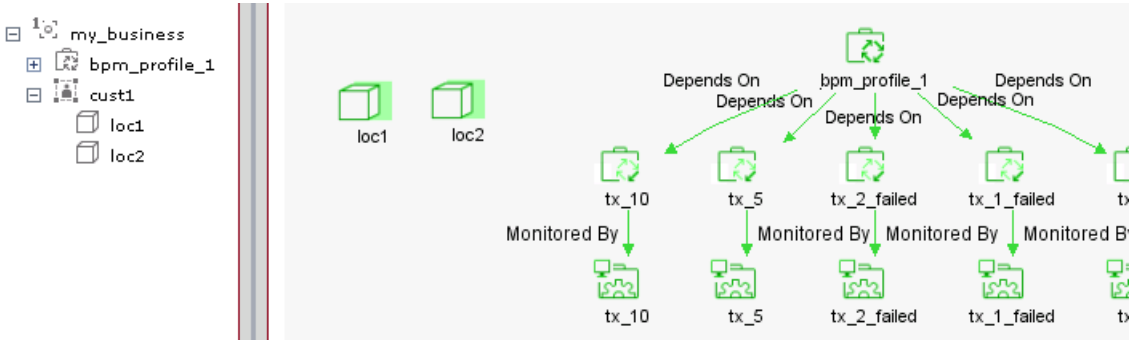
➤ If you define a **Child** rule on a relationship, the hierarchical order is changed, so that the child CIs are displayed above the parent. This can be applied to both the topology and tree presentations, or applied to only one of them if you are using different folding definitions.

---

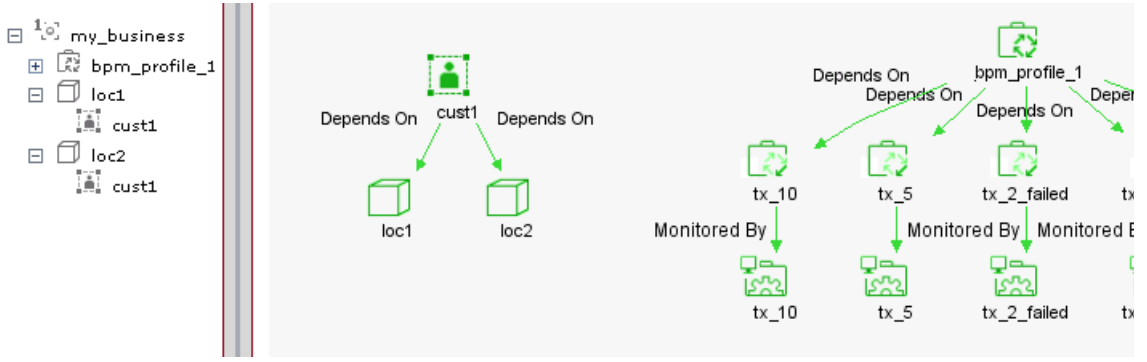
**Note:** The Child rule only affects presentation, not the dependency relationship between the parent and child CIs—the parent CI still takes status from the child CIs.

---

In the following example, a Child rule is defined in the topology presentation for the view, on the relationship between Customer and Location CIs. In the topology map, the hierarchy is reversed, so that child CIs loc1 and loc2 are shown with drill down to the parent CI, cust1. The tree for the view (with no Child rule) is unchanged.



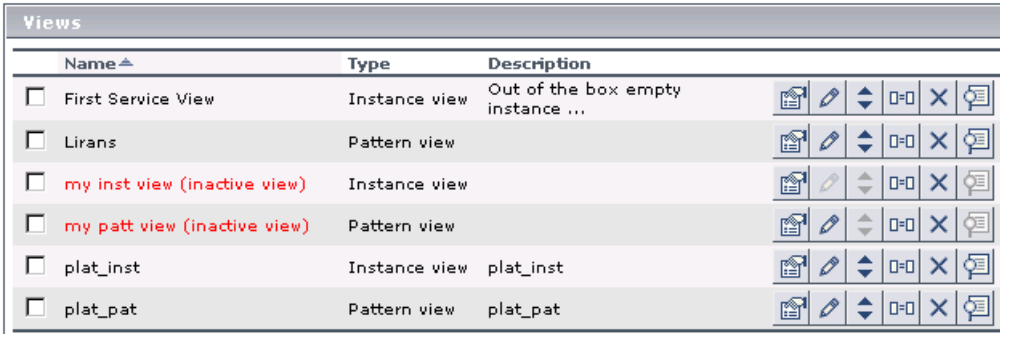
In the following example, a Child rule is defined in the tree presentation for the view, on the relationship between Customer and Location CIs. In the tree, the hierarchy is reversed, so that child CIs loc1 and loc2 are shown above the parent CI, cust1. The topology map (with no Child rule) shows the cust1 CI as the parent, and without folding.



# Activating or Deactivating Views

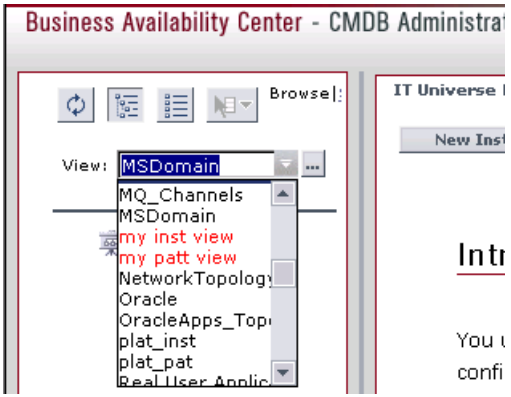
All new views are active by default, but you have the option of making a view inactive when editing the view properties. When a view is deactivated, the view is not built in the CMDB, so reducing the load on the system. You may want to deactivate a view, if, for example, the view is not working correctly.

Inactive views are shown in red in the Views table on the View Manager page, and some buttons are disabled.




You cannot edit an inactive instance view, but you can edit the TQL for an inactive pattern view.

In View Explorer, the inactive views are included in the View list in red, but they cannot be selected.



If you use the Search feature in View Explorer, the contents of an inactive view are not included in any search of the views; however, the CIs in the inactive view still exist in the CMDB, and can be found by doing a search of the entire CMDB. For more information on using the Search feature, see "Searching for Configuration Items" in *Working with the CMDB*.

---

**Note:** Views are automatically deactivated if the view is too big, meaning that they contain over the maximum number of CIs and relationships (100000).

This value is configured in the **<Mercury Business Availability Center server root directory>\mam_lib\server\appilogConfig.properties** folder, by modifying the value for **appilog.map.MaxObjectsInView**. However, it is recommended not to change the value without consulting Mercury Customer Support.

---

## Setting Permissions for Views

After creating a view, you should apply viewing and editing permissions to users and user groups for that view. You set permissions in **Admin** > **Platform** > **Users and Permissions** > **Permissions Management**.

Permissions are applied to views and not to folders. Every new view that is created is automatically added to the list of views in the Permissions Management page. You can set permissions at the **Views** level (the root folder) so that the permissions apply to all views. You can also set permissions for individual views.

For more information on setting permissions for views, see Chapter 22, "Configuring User Permissions" in *Platform Administration*.

---

**Note:** A view appears in IT Universe Manager, Dashboard, and Service Level Management only if specified in the Create View - View Properties dialog box when you created the view. For details, see "Creating and Editing Views" on page 19.

If a view is assigned to these applications but you still do not see the required view in View Explorer, it means that you have not been given permissions to view it. In this case, talk to your Mercury Business Availability Center administrator.

---

# Part II

## Instance Views

# 3

# Working with Instance Views

This chapter describes how you create an instance view. Instance views are used to build customized hierarchies of CIs.


| This chapter describes: | On page: |
|---|---|
| What Is an Instance View? | 45 |
| Creating a New View | 46 |
| Editing an Instance View | 47 |


## What Is an Instance View?

Instance views provide a framework for building customized hierarchies that meet your organization's business requirements and objectives, and that monitor what is important to you. There is no TQL for an instance view—you can add any CI that exists in the CMDB into an instance view, and you can manually create new CIs in the view.

Mercury Business Availability Center provides one factory instance view, the **First Service View**. You can create additional instance views as required, as described in the following section. You can also create new instance views by cloning an existing instance view, and editing the view contents (for details, see "Views Pane" on page 9).

# Creating a New View

There are two stages to the process of creating a new instance view in View Manager: defining the properties to create the view framework, and building the view by adding CIs from other views.

- Defining Properties to Create a New View – described below
- Adding CIs to the Created View – described on page 46

## Defining Properties to Create a New View

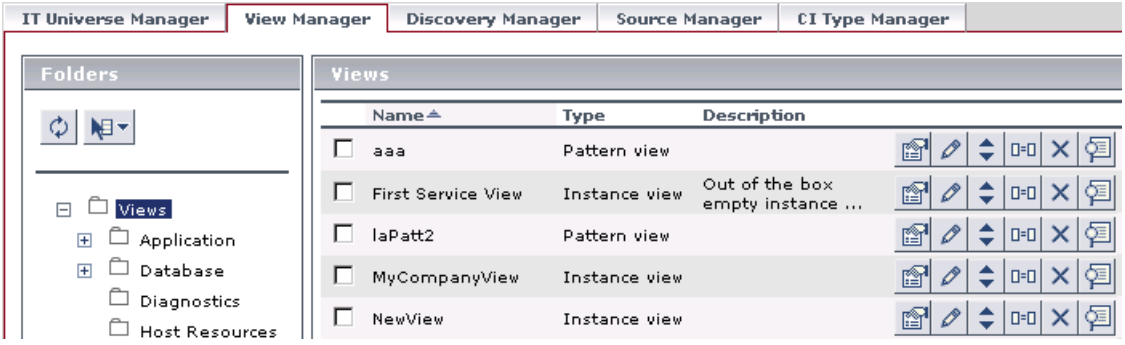
You can add a new instance view to any view folder. For details on defining the properties to create the view, see "Defining View Properties" on page 20.

After clicking **OK** in the Create View - View Properties dialog box, the Instance View Editor for the view you are creating is automatically opened. For details on using the Instance View Editor, continue with the following section.

## Adding CIs to the Created View

After defining properties for a new view (as described in the previous section), you have the option to continue the view creation process by defining new CIs to add to the view, and by adding existing CIs to the view. You do this in the Instance View Editor, as described in "Editing with the Instance View Editor" in *IT Universe Manager Administration*.

If you do not want to add any CIs to the view at this stage, click **OK** in the Instance View Editor. The view is added to the list of views in the selected folder in the View Manager tab.



You can then add CIs at a later stage by editing the view (as described in "Editing an Instance View" on page 47), or by using the options available in the IT Universe Manager tab. For details on how to do this, and more explanation of working with CIs in views in general, see *IT Universe Manager Administration*.

## Editing an Instance View



You can edit a view from the View Manager tab by clicking the **Edit view** button for the view. This opens the Instance View Editor, where you can define new CIs to add to the view, and add existing CIs to the view, as described in "Editing with the Instance View Editor" in *IT Universe Manager Administration*.

# Part III

## Pattern Views

# 4

# Working with Pattern Views

This chapter describes how you create a pattern view. Pattern views enable you to query the CMDB by defining Topology Query Language (TQL) queries. The pattern view you create displays the results of the TQL queries.


| This chapter describes: | On page: |
|---|---|
| About Pattern Views | 52 |
| Mercury Business Availability Center Topology Query System | 53 |
| The Roles of TQL | 53 |
| About Defining Pattern Views | 54 |
| Pattern View Workflow | 54 |
| Creating a Pattern View | 55 |
| Understanding the Create View Page | 58 |
| View Manager Tooltips | 60 |
| Adding Nodes and Relationships to a TQL Query | 60 |
| Setting TQL Node and Relationship Definitions | 69 |
| Using the TQL Node Wizard | 81 |
| Grouping CIs | 87 |
| Straightening a Relationship With an Angle | 93 |
| Adding Organization (Folding) Rules to Relationships | 94 |
| Defining Multiple Relationship Rules | 95 |
| Creating a Dependency Graph | 97 |
| Showing TQL Node Instances | 99 |

| This chapter describes: | On page: |
|---|---|
| Editing a Pattern View | 100 |
| Copying and Pasting TQL Nodes and Relationships | 100 |
| Saving the Pattern View | 101 |
| Using the Toolbar Options | 101 |
| Using the Toolbar Options | 101 |
| Printing the Contents of the Topology Map | 101 |
| Defining a View's Layout | 101 |
| Understanding Layout Options | 101 |


# About Pattern Views

A pattern view enables you to define TQL queries that extract information from the CMDB that is relevant to your needs (for details, see "Mercury Business Availability Center Topology Query System" on page 53). You define a pattern view and then create or attach a TQL query, which is a series of rules for defining how the query results are displayed in the editing pane, including which nodes and relationships are to be displayed. Once a TQL query has been created, it persists in the system memory and continuously generates updated results.

# Mercury Business Availability Center Topology Query System

The Topology Query System includes Topology Query Language (TQL). TQL is a language and tool for discovering, organizing, and managing IT infrastructure data. It is used to create queries that retrieve specific data from the CMDB and display that data. TQL queries constantly search the CMDB for changes that occur in the state of managed resources, and inform and update the relevant subsystems.

## The Roles of TQL

TQL is not in itself a stand-alone product. It is an integral part of Mercury Business Availability Center and it plays several roles:

- Enables you to draw conceptual relationships between nodes, which represent their actual interdependencies. This allows you to better understand and analyze the relationships between the nodes. You can use predefined operations to define different types of interconnections between the nodes, thus enabling you to represent your infrastructure design more accurately. This representation serves as a model for the discovery, arrangement, query, and management of your IT infrastructure.
- Creates queries that retrieve infrastructure data from the CMDB and displays it in a visual representation that consists of visual symbols that represent the resources and their interconnections. This graphic aspect simplifies data monitoring and managing.
- Constantly searches the CMDB for changes that occur in the state of managed resources. When such changes are detected, TQL continuously generates updated results accordingly.

## About Defining Pattern Views

When you create a pattern view, you must first name the managed view and then define how you want the query results displayed, including organization (folding) rules that determine the number of map levels to be generated, attribute conditions for each node and relationship and how many CI instances you expect to have at the end of a relationship in your query result.

You can either define the pattern view by manually dragging and dropping the TQL nodes manually from the Configuration Item Types pane or use the TQL Node Wizard (for details, see "Using the TQL Node Wizard" on page 81).

Once you have finished defining how you want the TQL results displayed, you activate the TQL query to retrieve the requested data from the CMDB.

## Pattern View Workflow

You create pattern views according to the following workflow:

- Define a new view.

  For details, see "Creating a Pattern View" on page 55.

- Add nodes and relationships to the query.

  For details, see "Adding Nodes and Relationships to a TQL Query" on page 60.

- Define node and relationship attribute conditions.

  For details, see "Setting TQL Node and Relationship Definitions" on page 69.

- Add organization (folding) rules to relationships.

  For details, see "Adding Organization (Folding) Rules to Relationships" on page 94.

- Define multiple relationship rules.

  For details, see "Defining Multiple Relationship Rules" on page 95.

➤ Create a dependency graph.

For details, see "Creating a Dependency Graph" on page 97.

➤ Save the TQL query.

For details, see "Saving the Pattern View" on page 101.

## Creating a Pattern View

The first step in creating a pattern view is to set the name and properties of the managed view in which you want to define the TQL query.

For details on defining the properties to create the view, see "Defining View Properties" on page 20.

After clicking **OK** in the Create View - View Properties dialog box, the Create View page for the view you are creating is automatically opened. For details on using the page, continue with "Understanding the Create View Page" on page 58.

Part of creating a view involves an option to merge identical instances as described in "Merging Identical Instances" on page 56.

---

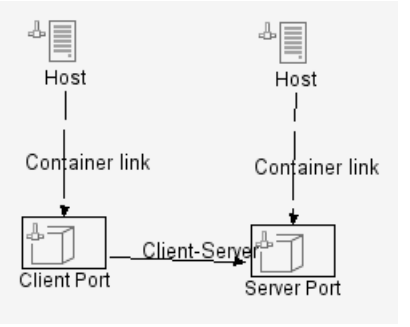
**Note:** TQL queries should only be created by administrators with an advanced knowledge of how the TQL system works.

---

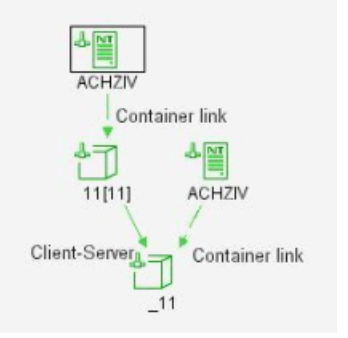
## Merging Identical Instances

You can decide to have identical instances that appear on the same layer in this view automatically merged into one symbol on the editing pane. This helps you remove redundant information and sharpen the view's focus.

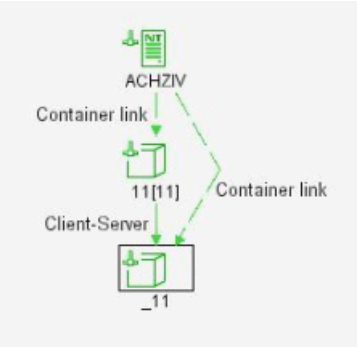
For example, suppose you want to discover and display hosts that are connected to other hosts in a server-client relationship, you can create the following TQL:




However, if both the server and the client are installed on the same host, one host appears twice.

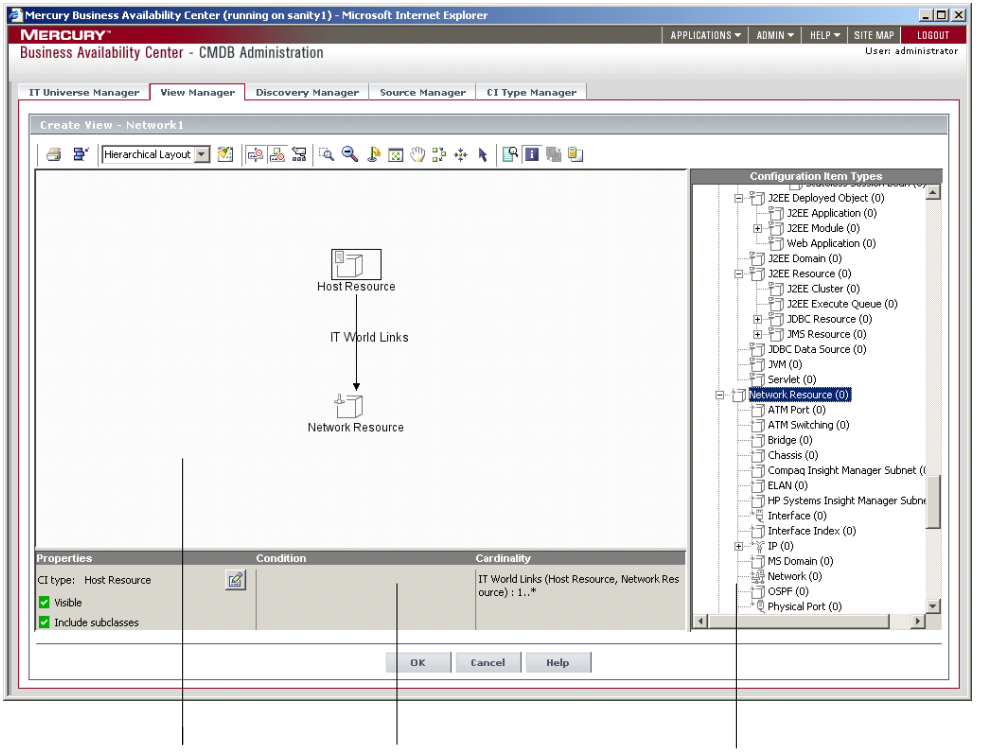


To overcome the redundant host display, you define that identical instances should be merged by selecting **Merge Identical Instances**:



# Understanding the Create View Page

This section describes how the Create View page is divided.




**Editing pane** – Displays the currently selected TQL query and the nodes and the relationships between them.

---

**Note:** Selecting the **Use different folding/grouping definitions for tree and topology presentations** option in the Create View Properties/Edit View Properties dialog box (for details, see "Defining View Properties" on page 20), changes the editing pane in the Create View/Edit View pages. When the option is selected, there are two versions of the editing pane, one for the topology presentation, and one for the tree presentation. When the option is not selected, there is one editing pane for the overall view. For details, see "Using the Different Folding/Grouping Definitions Option" on page 27.

---

**Configuration Item Types** – Represents the Configuration Item Type Model and contains icons for each CIT, as defined by the administrator (for details, see "Assigning an Icon to a CIT" in *CI Type Manager Administration*). The numbers that appear next to each CIT represent the number of instances there are of that CIT in the CMDB.

By clicking and dragging CITs to the editing pane and then defining the relationship between them, you can define a query and save it to the database. For example, you can drag the NT and IP CITs to the editing pane and then define the connection between them by adding relationships. For details, see "Adding Nodes and Relationships to a TQL Query" on page 60.

**Information pane** – Displays the attribute conditions you defined for the selected node and relationship. For details, see "Setting TQL Node and Relationship Definitions" on page 69.

## View Manager Tooltips

Hold the pointer over a node or relationship to view its tooltip. The tooltips contain the definitions of the selected nodes and relationships, for example, the attribute conditions, as described in "Setting TQL Node and Relationship Definitions" on page 69.

## Adding Nodes and Relationships to a TQL Query

Once you have created the pattern view, the next step is to add the nodes and relationships that define the query in the Create View page. The nodes represent the CITs, as defined in the CI Type Model. A relationship is a set of rules that defines the connection between two nodes. Relationships are defined one at a time for each pair of nodes in the query.

For a list of the relationships you can use to link two nodes and their definitions, see "Relationship Definitions" in *CI Type Manager Administration*.

This section contains the following topics:


- Adding Nodes and Relationships to Define a TQL Query – for details, see page 60
- Defining Function and Compound Relationships – for details, see page 63
- Sample Compound Relationship – for details, see page 66
- Deleting a Node or Relationship – for details, see page 68
- Editing a Relationship Type – for details, see page 68


### Adding Nodes and Relationships to Define a TQL Query

This section describes how to add nodes and relationships to the pattern view to define the TQL query.

**To add nodes and relationships to a define the TQL query:**

1. From the tree in the View Explorer, select the TQL query to which you want to add TQL nodes and relationships.
2. From the tree displayed in the Configuration Item Type pane, click and drag one or more required TQL nodes on to the editing pane. These are the TQL nodes that are included in the query.
3. To add a relationship between two nodes:

   ➤ Select the required TQL node(s) by holding down CTRL and clicking the TQL nodes, right-click and select **Add Relationship**.

   or



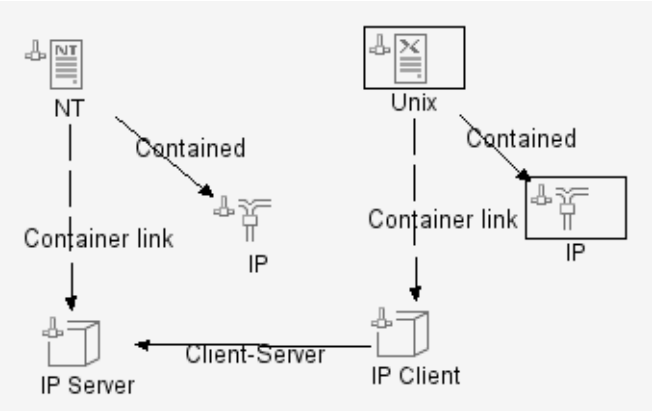
   ➤ Click the **Create Relationship** icon and draw a line between the required nodes.

The Add Relationship dialog box opens. It contains the following fields:

| Field | Description |
|---|---|
| **Node Label** | The label of the selected nodes. |
| **Relationship Direction** | The direction of the relationship that indicates which node is dependent on the other. |
| **Relationship Type** | A valid relationship that defines the connection between the selected nodes. |
| **Advanced** | ➤ Select **Relationship** to define the connection between two nodes using a child of the relationship in the **Relationship Type** box.<br>➤ Select **Function Relationship** to define the connection between two nodes using either a function or compound relationship. For details, see "Defining Function and Compound Relationships" on page 63.<br>➤ Define how you want to handle relationships between identical nodes or internal relationships, that is, a relationship that leads from a node to itself. For details, see step 4 in "Defining Function and Compound Relationships" on page 63. |

4. Select the required direction of the relationship.
5. If necessary, click **Advanced** to select either function or compound relationships.
6. Click **OK**. The selected nodes are linked by the relationship you have selected.

The direction of the relationship indicates which node is dependent on the other. The following example displays two hosts, an IP server, and an IP client that are linked to one another via a client/server connection.




## Defining Function and Compound Relationships

You can define the connection between two nodes using either a function or compound relationship.

- A function relationship (**Virtual join link**) describes a logical connection between two nodes, which is not represented in the CMDB.

  When you create a function relationship, you must define an attribute for each node, whose value will be used for comparison.

- A compound relationship (**Virtual compound link**) describes a connection that is composed of more than one relationship. If there are two nodes whose connection cannot be directly defined by one relationship, you can add several relationships, which act as stepping stones along the line, to create a connection.

**To define function and compound relationships:**

1. Select the nodes you want to connect with a function or compound relationship (for details on how to add nodes to a TQL query, see "Adding Nodes and Relationships to Define a TQL Query" on page 60).
2. Right-click and select **Add Relationship** to open the Add Relationship dialog box.

3 Click **Advanced**.

4 To define how to handle relationships between identical nodes or internal relationships (a relationship that leads from a node to itself) in the query results, select one of the following from the **Links** list:

- **Allow all links.** All relationships appear in the query results.
- **Allow self links only.** Only internal relationships appear in the query results.
- **Discard self links.** Internal relationships do not appear in the query results.

5 To define the connection between two nodes using a **Virtual Join link**, select **Function Relationship** from the **Relationship** list and do the following.

- Select **Virtual - Join link**.
-  In the **Join Definition** area, click the **Add an attribute definition** button to open the Conditions dialog box.
- From the **Attribute** lists, choose the two required attributes.
- From the **Operator** list, select one of the following operators:
  - Select **Equal** if you want the system to check whether the two selected attributes are equal.
  - Select **Not equal** if you want the system to check whether the two selected attributes are not equal.
  - Select **Sub string** if you the system to check whether the first attribute is a subset of the second.
  - Select **Sub string ignore case** if you want the system to check whether the first attribute is a subset of the second regardless of the string's case.
- Click **OK** to save your definition. The definition appears in the **Join Definition** area.
- To delete a definition, select the required definition and click the **Delete selected row** button.
- To edit a definition, select the required definition and click the **Edits selected row** button to open the Conditions dialog box. Edit the definition as described in step 5.

➤ Click **OK** to save your changes.

**6** To define the connection between two nodes using a **Virtual Compound link**, select **Function Relationship** from the **Relationship** list and do the following:

➤ Select **Virtual - Compound link.**



➤ In the **Compound Definition** area, click the **Add** button to open the Add dependency dialog box.

➤ From the **Source** list, select the source node.

➤ From the **Target** list, select the target node.

➤ From the **Relationship** list, select an available relationship connecting the two nodes. The list of available relationships appears only after defining both nodes.

➤ From the **Relationship Direction** list, select the required direction.

---

**Note:** You can create as many virtual compound definitions as necessary.

---

➤ Click **OK** to save the compound definition. The compound definition appears in the Compound definition dialog box.

---

**Note:** Each node represents a step in the chain leading from one node to the other. Ensure that <**node 1**> of the first row and <**node 2**> of the last row are the two selected nodes in the TQL query.

---



➤ To delete a definition, select the required definition and click the **Delete** button.



➤ To edit a definition, select the required definition and click the **Edit** button to open the Add dependency dialog box. Edit the definition as described in step 6.

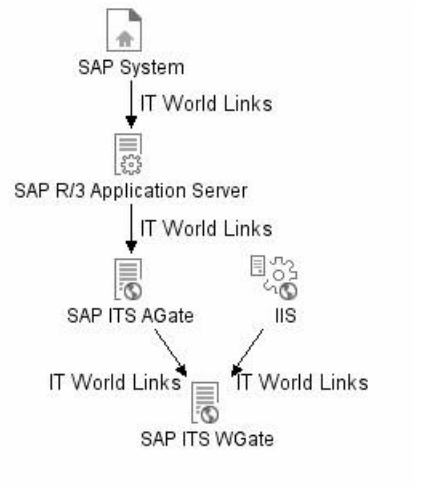
➤ In the **Depth** box, select a number that determines how many steps (that is, how many connecting nodes) in the TQL pattern you want included in the discovery process. The default is 5.

**7** Click **OK** to save your definitions.

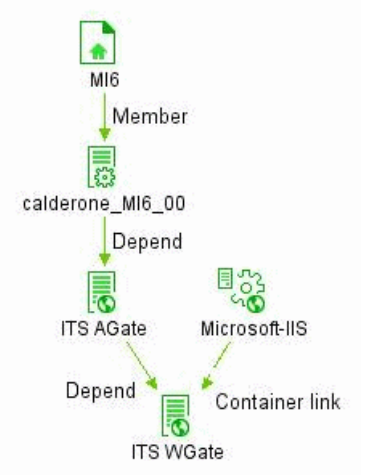
### Sample Compound Relationship

This section contains a sample compound relationship comprising several nodes and their connections, displayed in two different views:

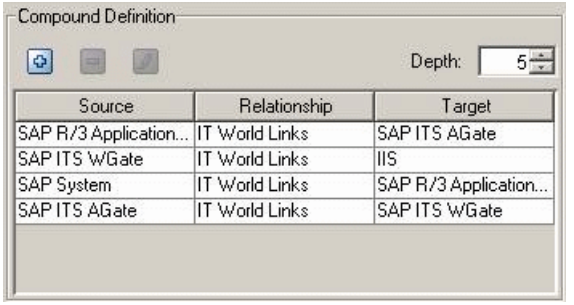
➤ A normal view in which each node in the compound relationship was dragged and dropped from the Configuration Item Types pane. (For details, see"Adding Nodes and Relationships to Define a TQL Query" on page 60.)



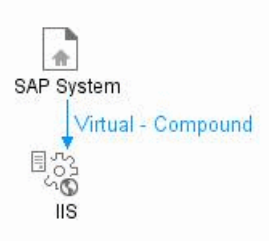
The normal view result is displayed below:



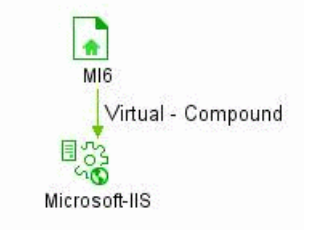

➤ A compound view in which the compound relationship was defined by using the **Compound Definition** area of the Add Relationship dialog box, as illustrated below. (For details, see "Defining Function and Compound Relationships" on page 63.)



The following compound view is created:




The compound view result is as follows:




## Deleting a Node or Relationship

This section describes how to delete a node or relationship.

**To delete a node or relationship:**

Right-click the TQL node or relationship you want to delete and select **Delete**.

## Editing a Relationship Type

This section describes how to edit an existing relationship. This option only appears if at least one of the following is true:

- You can change the direction of the relationship.
- You can select a descendent of the existing relationship.
- The selected relationship is function relationship (Compound or Join)

**To edit a relationship type:**

1. Right-click the required relationship and select **Edit Relationship Type**. The Edit Relationship Type dialog box opens.
2. Make the required changes. For details, see "Adding Nodes and Relationships to Define a TQL Query" on page 60 and "Defining Function and Compound Relationships" on page 63.
3. Click **OK** to save your changes.

# Setting TQL Node and Relationship Definitions

After you have added the TQL nodes and relationships required for your query, you can define their specific attribute conditions.

This section includes:


- Defining TQL Nodes and Relationships – for details, see page 69
- Changing the CI Type of the TQL Node – for details, see page 71
- Defining Attribute Conditions for Nodes and Relationships – for details, see page 71
- Filtering Query Results in the Attribute Condition Tab – for details, see page 74
- Defining Qualifier Conditions – for details, see page 76
- Defining Relationship Cardinality – for details, see page 77
- Filtering Query Results in the Cardinality Tab – for details, see page 78


## Defining TQL Nodes and Relationships

This section describes how to define TQL nodes and relationships.

**To define TQL nodes and relationships:**

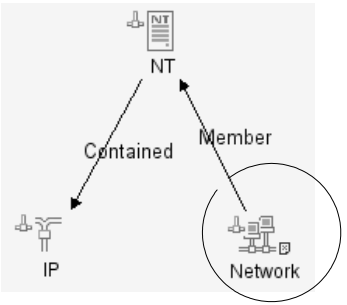
1. In the editing pane, right-click the TQL node or relationship whose attribute conditions you wish to define, and select **TQL Node Definition** or **TQL Relationship Definition** to open the TQL Node/Relationship Definition dialog box.



**Note:** Alternatively, you can select the required node/relationship, and click the **Go to Edit Dialog** button in the Information pane to open the TQL Node/Relationship Definition dialog box.

**2** (Optional) The **Element name** box contains the name of the selected node/relationship. By default, Mercury Business Availability Center assigns the CIT type as the element's name. You can rename a TQL node in the **Element name** box giving it a unique label. This can be helpful when there is more than one node of the same CIT type in the TQL query.

**3** To display the selected TQL node in the editing pane, select **Visible**. When **Visible** is cleared, an invisible box appears to the right of the selected TQL node in the editing pane:




Any query results pertaining to that TQL node are not displayed in the topology map. This can be useful when certain relationships or TQL nodes are required to build the query but are not needed in the results. For example, NTs are connected to the network by defining specific IPs, but in any query results you may want to view the IP elements only and not the NT elements

**4** To display both the selected CI and its descendents in the topology map, select **Include subclasses**.

### Changing the CI Type of the TQL Node

You can change the CI type of the TQL node after you have created a TQL.

**To change the CI type of a TQL node after creating a TQL query:**

1 In the editing pane, right-click the TQL node whose CI type you want to change. and select **TQL Node Definition** to open the TQL Node Definition dialog box.



2 Click the **Change CI type** button to open the Change CI type dialog box.

3 Select the required CI type and click **OK**.

---

**Note:** You can only change the CI type to a type from one of the CIT's subclasses, if one exists.

---

### Defining Attribute Conditions for Nodes and Relationships

This section describes how to define attribute conditions for nodes and relationships.

**To define attribute conditions:**

1 In the editing pane, right-click the TQL node or relationship whose attribute conditions you wish to define, and select **TQL Node Definition** or **TQL Relationship Definition** to open the TQL Node/Relationship Definition dialog box.

2 In the TQL Node/Relationship Definition dialog box, click the **Attribute Conditions** tab.



3 Click the **Add an attribute condition** button to open the Condition dialog box.

4 From the **Attribute Name** list, choose the required attribute.

5 From the **Operation** list, select the required operation.

The operations in the **Operation** list are:

| Operation | Description |
|---|---|
| **Equal** | Checks whether the attribute value is equal to the value specified in the **Value Comparison** box. |
| **Not equal** | Checks whether the attribute value is not equal to the value specified in the **Value Comparison** box. |
| **Is null** | Checks whether the attribute value is null. |
| **Like** | Uses a wildcard (%). Use **Like** when you are not sure of the complete name of what you are looking for. |
| **In** | Displays only the instances where this attribute value equals one of the selected values. For example, for CIs that have an operational state that equals Warning(2) and Warning(3), select the operation **In**, and select both Warning(2) and Warning(3) from the **Value comparison** list. |
| **Equal ignore case** | Checks whether the attribute value is equal to the value specified in the value box regardless of the case. |
| **Greater** | Checks whether the attribute value is greater than the value specified in the **Value comparison** box. |
| **Great than or equal** | Checks whether the attribute value is greater than or equal to the value specified in the **Value comparison** box. |
| **Less** | Checks whether the attribute value is less than the value specified in the **Value comparison** box. |
| **Less than or equal** | Checks whether the attribute value is less than or equal to the value specified in the **Value comparison** box. |
| **Changed during** | (Displayed only when you select the **Create Time** attribute.) Displays only the instances that changed during the period specified in the **Value comparison** box. |

| Operation | Description |
|---|---|
| **Unchanged during** | (Displayed only when you select the **Create Time** attribute.) Displays only the instances that did not change during the period specified in the **Value comparison** box. |
| **In list** | Equals one of the elements that appears in the list. For example: 320,4445,3483. |

**Note:** When using the **Not equal** or **Not like** operations, query results do not include data from the CI instances that were not assigned a value. For example, let us assume that your system contains three hosts. Host1 was assigned the value A, Host2 was assigned the value B, and Host3 was not assigned a value. If you created a query in which you want to retrieve all hosts that are **Not equal** to A, the results of your query only include Host2 because Host3 was not assigned a value.

**Note:** Mercury Business Availability Center supports both MS SQL and Oracle database servers. MS SQL Servers used by Mercury Business Availability Center are not case sensitive by default, unlike Oracle servers which are case sensitive. Consequently, if you are using MS SQL Server, the **Equal** operation retrieves the same query results as the **Equal ignore case** operation. For example, if you select the attribute **City**, the operation **Equal**, and type **NEW YORK** in the **Value comparison** box, case differences will be ignored and query results include **NEW YORK**, **New York** and **new york**.

**6** If you want to make a value comparison, do the following:

- Select **Value**.
- In the **Value Comparison** box, enter or select the value of the attribute.

---

**Note:** If you use a string in the **Value** box, it must appear in single quotes, as follows: 'YOKO', 'JASON'.

---

**7** If you want to create a condition in which you compare two attributes, do the following:

- ➤ Select **Attribute**.
- ➤ In the **Attribute Comparison** box, select the required attribute from the **Attribute Comparison** list.

**8** To filter the query results, see "Filtering Query Results in the Attribute Condition Tab" below.

**9** Click **OK** to save the changes you have made and close the Condition dialog box.

### Filtering Query Results in the Attribute Condition Tab

In the **Attribute Condition** tab, you can create an expression defining conditions that restrict the number of nodes that appear in the query.

**To filter your query results:**

**1** In the editing pane, right-click the TQL node or relationship whose attribute conditions you wish to define, and select **TQL Node Definition** to open the TQL Node Definition dialog box.

**2** Click the **Attribute Condition** tab.

**3** In the **Condition** area, create an expression using the following:

| Field | Description |
|---|---|
| NOT | Select **NOT** if you want the condition statement to do the opposite of what is defined.<br>**Note**: If you select **NOT**, query results do not include data from the CI instances that were not assigned a value. For example, let us assume that your system contains three hosts. Host1 was assigned the value A, Host2 was assigned the value B and Host3 was not assigned a value. If you created a query in which you want to retrieve all hosts that are equal to A and selected **NOT**, the results of your query only include Host2 because Host3 was not assigned a value. |
| Brackets  | Click inside the **Brackets** box to display a list of brackets you can use to build more complex logical statements. |
| Criteria | Contains the attribute condition definition as defined in the Condition dialog box (for details, see step 3 in "Defining Attribute Conditions for Nodes and Relationships" on page 71). |
| And/Or | Click inside the **And/Or** field and select either **And** or **Or** to link multiple conditions. |

**4** To delete an attribute condition, select the attribute condition you want to delete and click the **Delete selected row** button.

**5** To insert an attribute condition before a selected row, select the required row, and click the **Insert an attribute condition before a selected row** button to open the Condition dialog box.

**6** To edit an attribute condition, select the attribute condition you want to edit and click the **Edit selected row** button to open the Condition dialog box. For details on how to define an attribute condition, see "Defining Attribute Conditions for Nodes and Relationships" on page 71.

**7** To move a row up, select the row you want to move and click the **Move selected row up** button.

8 To move a row down, select the row you want to move and click the **Move selected row down** button.



9 Click **OK** to save the changes you have made.

### Defining Qualifier Conditions

This section describes how to define qualifier conditions for the selected node/relationship. For example, you can use a qualifier to define a CIT as abstract, meaning you cannot create instances from it.

**To define qualifier conditions:**

1 In the editing pane, right-click the TQL node or relationship whose qualifier conditions you wish to define, and select **TQL Node Definition** or **TQL Relationship Definition** to open the TQL Node/Relationship Definition dialog box.

2 Click the **Qualifier condition** tab.

3 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

---

**Note:** You can hold the CTRL key down to make multiple selections.

---

4 Click **OK** to save your definitions.

### Defining Relationship Cardinality

In this section, you define how many CI instances you expect to have at the end of a relationship in your query result.

**To define relationship cardinality:**

1 In the editing pane, right-click the TQL node whose conditions you wish to define, and select **TQL Node Definition** to open the TQL Node Definition dialog box.

2 Click the **Cardinality** tab.

---

**Note:** This tab only appears when a node is selected.

---



3 Click the **Add an attribute condition** button to open the Relationship Cardinality dialog box.

4 In the **Node** box, select the required relationship to which the selected node is attached. The list will contain all the relationships that are linked to the selected node.

In the following steps, you define the lower and upper limits for including the node at the other end of the relationship in the query results.

5 In the **Min** box, enter the value that defines the lower limits for including the node at the other end of the relationship in the query results.

6 In the **Max** box, enter the value that defines the upper limits for including the node at the other end of the relationship in the query results. You can use an asterisk (*) in the **Max** box to represent an infinite value.

For example, if <end_1> is IP and <end_2> is NT, entering 1 in the **Min** box and asterisk (*) in the **Max** box instructs the system to retrieve only those IPs that are connected to at least one NT. (The asterisk indicates an infinite value.) Entering 3 in the **Min** box and asterisk (*) in the **Max** box instructs the system to retrieve only those NTs that are connected to at least three IPs.

7 To filter the query results, see “Filtering Query Results in the Cardinality Tab” on page 78.

8 Click **OK** to save the changes you have made.

## Filtering Query Results in the Cardinality Tab

Mercury Application Mapping enables you to define relationship conditions that allow the relationship's connecting node to be included in the TQL query results. In the **Cardinality** tab, you can create an expression defining that condition. For an example, see "Sample Relationship Conditions" on page 79.

**To filter your query results:**

1 In the editing pane, right-click the TQL node or relationship whose qualifier conditions you wish to define, and select **TQL Node Definition** to open the TQL Node Definition dialog box.

2 Click the **Cardinality** tab.

3 In the **Cardinality** area, create an expression using the following:

| Field | Description |
|---|---|
| Brackets { } | Click inside the **Brackets** box to display a list of brackets you can use to build more complex logical statements. |
| Criteria | Contains the attribute condition definition as defined in the Condition dialog box (for details, see step 3 in "Defining Attribute Conditions for Nodes and Relationships" on page 71). |
| And/Or | Click inside the **And/Or** field and select either **And** or **Or** to link multiple conditions. |

4 To delete an attribute condition, select the attribute condition you want to delete and click the **Delete selected row** button.

5 To insert an attribute condition before a selected row, select the required row, and click the **Insert an attribute condition before a selected row** button to open the Condition dialog box.

6 To edit an attribute condition, select the attribute condition you want to edit and click the **Edit selected row** button to open the Condition dialog box. For details on how to define an attribute condition, see "Defining Attribute Conditions for Nodes and Relationships" on page 71.

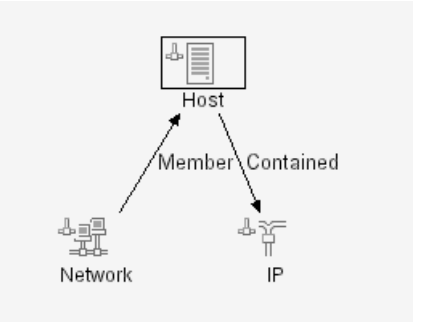
**7** To move a row up, select the row you want to move and click the **Move selected row up** button.

**8** To move a row down, select the row you want to move and click the **Move selected row down** button.

**9** Click **OK** to save the changes you have made.

### Sample Relationship Conditions

This section contains an example of relationship conditions based on the sample TQL query shown below.

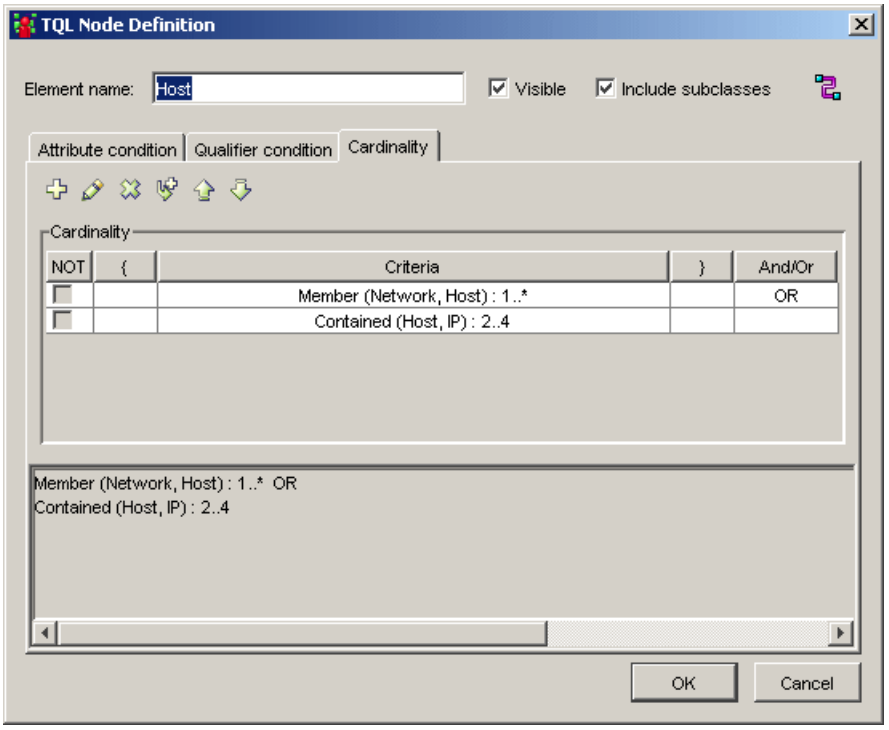



The following relationship conditions were defined for the TQL:

In the Relationship Cardinality dialog box, the following was defined (see "Defining Relationship Cardinality" on page 77):

- Member – **Min**: 1, **Max**: *
- Contained – **Min**: 2, **Max**: 4

The definitions appear in the **Cardinality** section, as displayed below.




- [contained (Host, IP)] OR [member (Network, Host)] – means that the host must either have between two and four IPs OR be a member of the network.
- [contained (Host, IP)] AND [member (Network, Host)] – means that the host must have between two and four IPs AND ALSO be a member of the network.

# Using the TQL Node Wizard

You can use the TQL Node Wizard to build a TQL query. The TQL Node Wizard allows you to:

- Add a node to a TQL query
- Add a relationship to a TQL query
- Define node conditions
- View all of the instances found for the TQL node in a table

This section includes:


- Adding a Node to the TQL Query – for details, see page 81
- Adding a Relationship to a TQL Query – for details, see page 84
- Setting TQL Node Definitions – for details, see page 86
- Showing TQL Node Instances – for details, see page 87

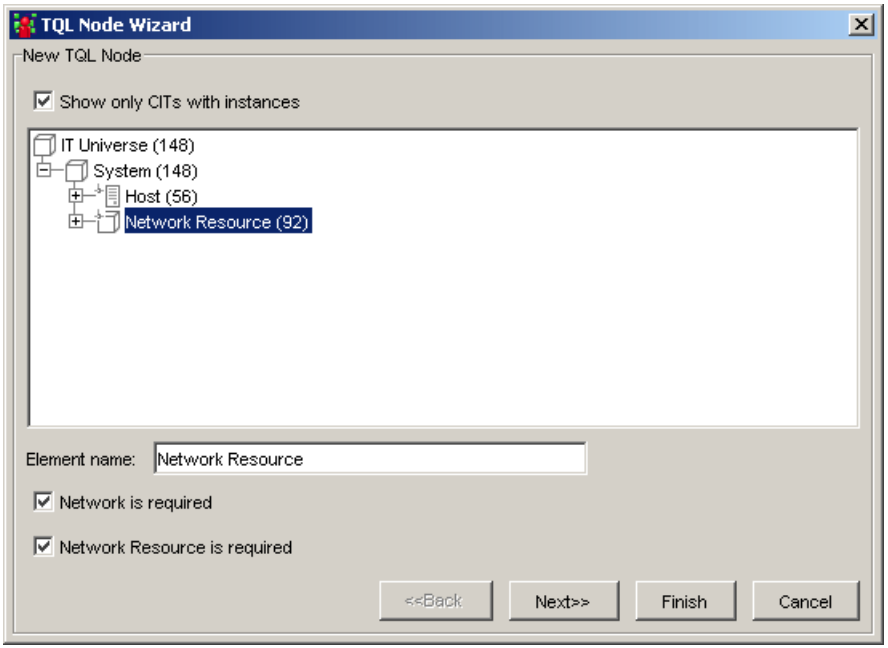

## Adding a Node to the TQL Query

This section describes how to add a node to the TQL query.

**To add a node to the TQL query:**

1 In the required view, if the view is empty, drag a TQL node on to the editing pane from the tree displayed in the Configuration Item Type pane.

**2** Right-click the required TQL node and select **TQL Node Wizard**. The New TQL Node page opens, allowing you to add a new TQL node to the query.




This page displays only the list of TQL nodes of the CIT type that have valid relationships to the selected (source) node.

To the right of each node, the number of CI instances that exist in the CMDB for that CIT type is displayed. The number of instances is only updated after you close the TQL Node Wizard and open it again.

---

**Note:** The first node in the hierarchy tree is selected by default.

---

**3** Select the **Show only CITs with instances** check box to display only the CITs of which there are instances in the CMDB.

---

**Note:** The **Show only CITs with instances** check box is selected by default.

---

**4** (Optional) The **Element name** box contains the name of the selected node. By default, Mercury Business Availability Center assigns the CIT type as the element's name. You can rename a TQL node in the **Element name** box giving it a unique label. This can be helpful when there is more than one node of the same CIT type in the TQL query.

**5** The **<node> is required** check box defines the relationship cardinality (for details, see "Defining Relationship Cardinality" on page 77).

- Select the **<node> is required** check box for each node to include a minimum of one instance of the node at the other end of the relationship in the query results. Selecting the check box gives the relationship a cardinality value of 1*.
- Clear the **<node> is required** check box to give the relationship a cardinality value of 0*.

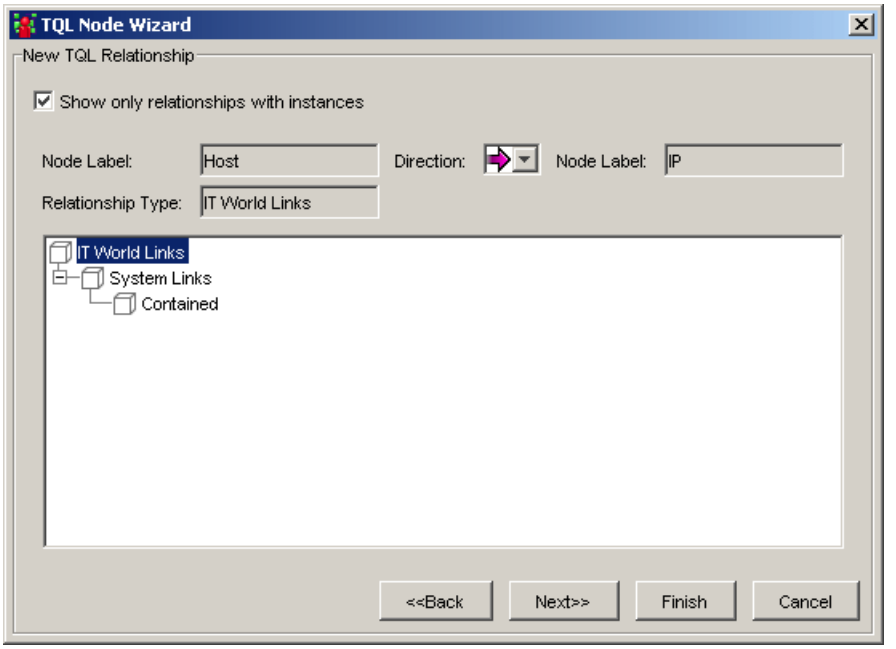
**6** Click **Finish** to create the TQL query with the node you have added. To add a relationship, see "Adding a Relationship to a TQL Query" on page 84.

## Adding a Relationship to a TQL Query

This section describes how to add a relationship to the TQL query.

**To add a relationship to the TQL query:**

**1** In the New TQL Node page (for details, see "Adding a Node to the TQL Query" on page 81), select **Next** to open the New TQL Relationship page.



The Add Relationship dialog box contains the following fields:

| Field | Description |
|---|---|
| **Show only relationships with instances** | Display only the relationships of which there are instances in the CMDB. |
| **Node Label** | The label of the source node. The default value is the CI Type display name. For details, see "Creating CITs" in *CI Type Manager Administration*.<br>**Note:** This field appears on the right hand side of the page. |
| **Direction** | The direction of the relationship that indicates which node is dependent on the other. |
| **Node Label** | The label of the node you selected in the New TQL Node page.<br>**Note:** This field appears on the right hand side of the page. |
| **Relationship Type** | A valid relationship that defines the connection between the selected nodes. The field displays the relationship you selected in the node hierarchy tree. |

**2** Select the required direction of the relationship.

**3** If required, select another valid relationship type to define the connection between the selected nodes.

---

**Note:** The relationship type that is selected by default is the highest in the hierarchy of the available, valid relationships.

---

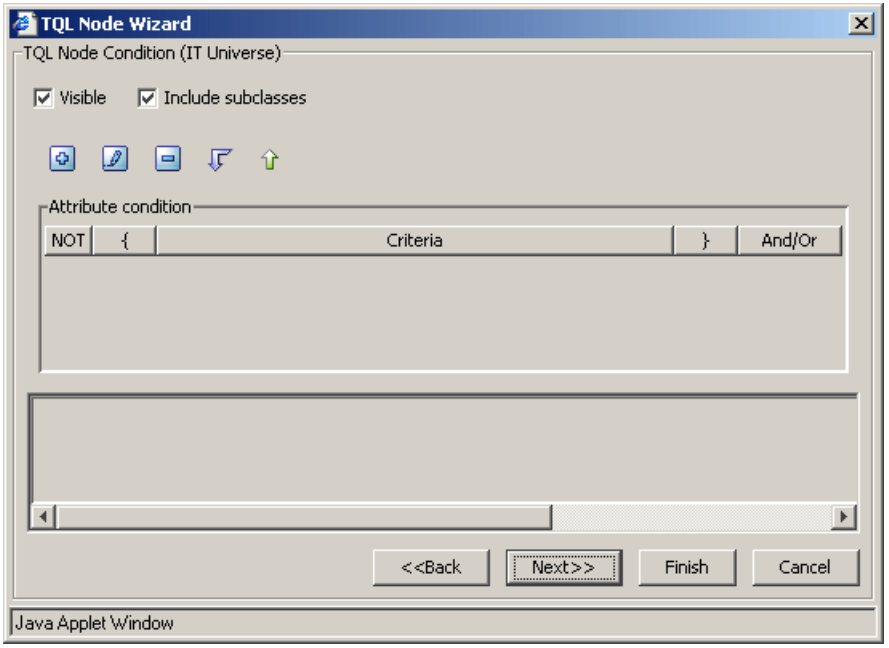
**4** Click **Finish** to create the TQL relationship you added to the query. The selected nodes are linked by the relationship you have selected. The direction of the relationship indicates which node is dependent on the other. To set node definitions, see "Setting TQL Node Definitions" on page 86.

## Setting TQL Node Definitions

This section describes how to define TQL node definitions.

**To define TQL node definitions:**

1 In the New TQL Relationship page (for details, see "Adding a Relationship to a TQL Query" on page 84), select **Next** to open the New TQL Node Condition page.




2 To define attribute conditions for the node you added, see "Defining Attribute Conditions for Nodes and Relationships" on page 71.

3 To filter query results in the **Attribute condition** area, see "Filtering Query Results in the Attribute Condition Tab" on page 74.

4 Click **Finish** to save the node conditions you have defined. To display the TQL instances that were found, see "Showing TQL Node Instances" on page 87.

### Showing TQL Node Instances

You can display all of the instances found for each TQL node in a table.

**To display all of the instances found for each TQL node in a table:**

1. In the TQL Node Condition page (for details, see "Setting TQL Node Definitions" on page 86), select **Next** to open the Element Instances page. For details, see "Showing TQL Node Instances" on page 99.
2. Click **Finish** to save your TQL query definitions.

## Grouping CIs

Mercury Business Availability Centerenables you to group the CIs in a pattern view according to defined criteria, for example, by CIT or by a particular attribute. You can apply the grouping option you defined to the layer of your view to organize the way the view is presented throughout Mercury Business Availability Center. For more details, see "Grouping Options (Pattern Views Only)" on page 29.

You can define whether different grouping definitions should be used for the topology and tree presentations of the view. To define different grouping options for topology and tree presentations, you select the **Use different folding/grouping definitions** option in the view properties dialog box (for details, see "Using the Different Folding/Grouping Definitions Option" on page 27).

You can apply grouping options for the overall view, or different grouping options can be defined to apply for the topology presentations and tree presentations of the view.

This section has the following topics:


- "Grouping CIs in the Topology Map" on page 88
- "Grouping By Attribute Properties" on page 91
- "Nested Groups" on page 92

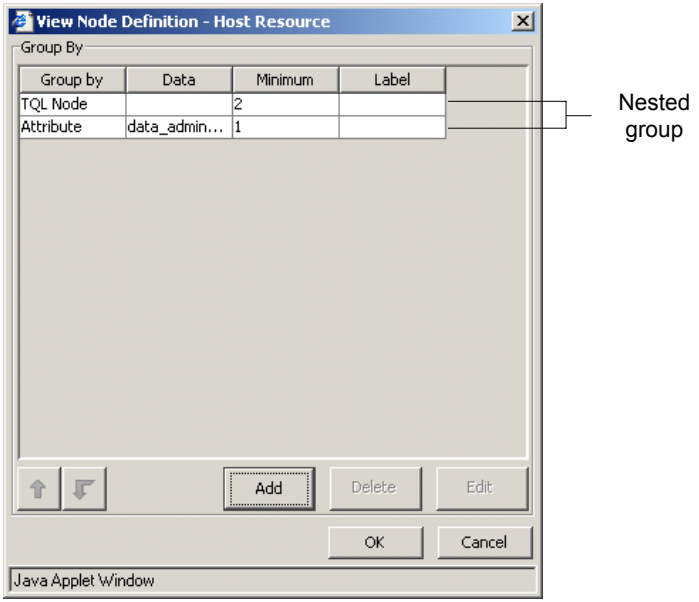
## Grouping CIs in the Topology Map

This section describes how to group CIs in the topology map.

**To group CIs in the topology map:**

1. In the editing pane, right-click the node for which you want to create groups according to selected criteria and select **View Node Definition**. The View Node Definition dialog box appears.
2. To define a group of nodes for this view, click **Add** to display the Group Configuration Items dialog box.
3. Define a group of nodes for this view by selecting one of the following options:
   - **Configuration Item Type –** Group CIs in the topology map by their CIT.
   - **TQL Node –** Groups all CIs of the same CIT that fulfill the node definition criteria. By default, these CIs appear in the topology map under a folder, containing the name of the selected TQL node and an automatically added identification number.
   - **Attribute –** Groups CIs from multiple CITs in the topology map, according to the attribute selected from the list. A CI can be a member of multiple groups. The **Attribute** option enables you to group CIs from different CITs. The **Configuration Type Item** and **TQL Node** options are used to group CIs from the same CIT. For details, see "Grouping By Attribute Properties" on page 91).
4. (Optional) In the **Minimum in group** box, enter the minimum number of matching CIs required to create the group. By default, there must be at least one CI in the group.
5. (Optional) In the **Group label** box, enter the label of the group.
6. You can create nested groups (using the **operation state** attribute), that is, a CI group inside another CI group. This option provides you with additional fine-tuning of the TQL results that are displayed in the topology map (for more details, see "Nested Groups" on page 92).

**7** To nest a group, create one group and return to the **Group By** dialog box (for more details, see "Nested Groups" on page 92). Click the **Add** button again and create another group. When you finish creating the second group, the **Group By** tab should look similar to the following figure:




You can repeat this procedure to create as many groups as needed.



**8** To change their nesting order by using the arrow buttons at the bottom of the View Node Definition dialog box.

**9** To delete a group, select the required group and click **Delete**.

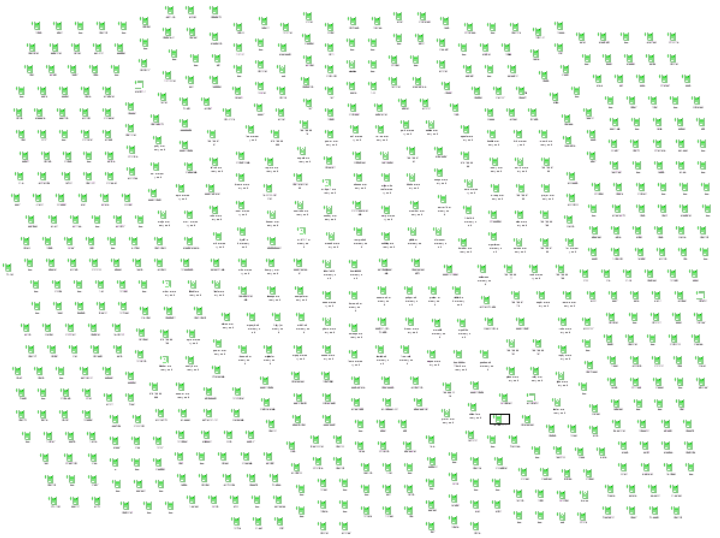
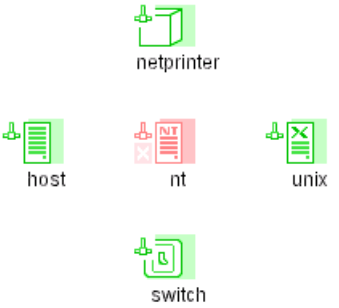
**10** To edit a group:

- Select the required group and click **Edit**.
- Repeat steps 2 to 4 in "Grouping CIs in the Topology Map" on page 88.

**11** Click **OK** to save the node's definitions.

**12** Click **OK** to save the definitions of the selected node.

This following contains examples showing how the host grouping affects the display in the topology map:

| | |
|---|---|
|  | Non-grouped hosts |
|   | Grouped hosts |

### Grouping By Attribute Properties

You can group nodes from multiple CITs in the topology map, according to the following attributes:

➤ **Mask to group by:**

Enter a regular expression, as follows:

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

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

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: |
|---|---|---|
| Group by zz | (.*[.].*[.])(.*)([.].*) | 2 |
| Group by yy | (.*[.])(.*)([.].*[.].*) | 2 |
| Group by aa | (.*)([.].*[.].*[.].*) | 1 |
| Group by mm | (.*[.].*[.].*[.])(.*) | 2 |

You can also group by the first or last letter of any attribute:

For example, enter a regular expression to group the nodes by the first or last letter(s) of the selected attribute as follows:

| To: | In the First Field Enter: | In the Second Field Enter: |
|---|---|---|
| Group by the first letter | (.)(.*) | 1 |
| Group by the last letter | (.*)(.) | 2 |
| Group by the first two letters | (..)(.*) | 1 |
| Group by the last two letters | (.*)(..) | 2 |

➤ **Group List** – Define the specific groups of the selected attribute to be created. For example, to display two groups of IP addresses that end between 0 and 100, enter 0-50, 51-100. Any IPs that fall outside this group are placed automatically into the **Others** group.

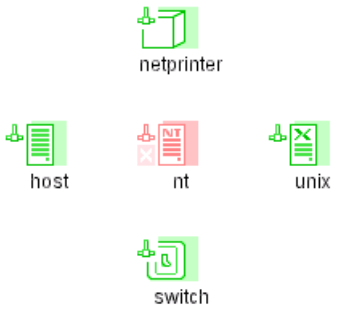
### Nested Groups

You can create nested groups (a node group inside another node group) to help fine-tune the TQL results that are displayed in the topology map.

For example, after grouping host nodes by their CIT, you can use their **operation state** attribute to create more focused groups.

In the topology map, nested grouping is displayed as follows:

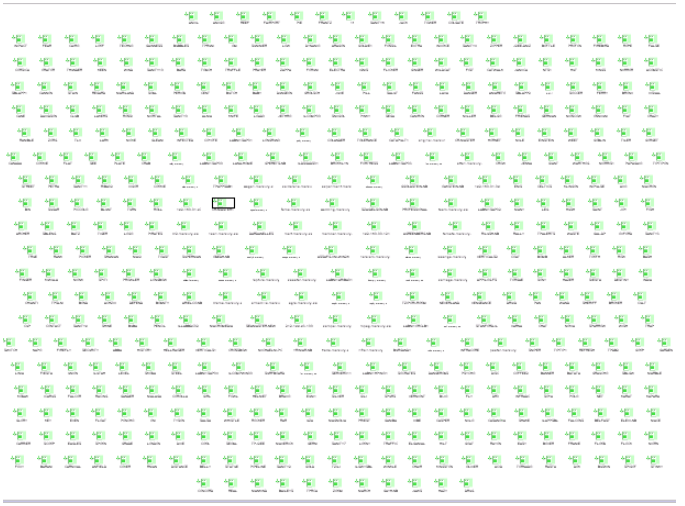
➤ The first layer shows host nodes grouped by their CIT:




➤ Drill down to the next layer to display host nodes grouped by **Oper State**:



➤ Drill down to the next layer to display host nodes belonging to a certain host CIT and with the same **Oper State**:



## Straightening a Relationship With an Angle

Mercury Business Availability Center allows you to straighten a relationship that has angles.

**To straighten a relationship with angles:**

Right-click the relationship that has an angle and select **Straighten**.

# Adding Organization (Folding) Rules to Relationships

After setting the node and relationship definitions of the view, you can add organization (folding) rules to selected relationships. This enables you to define the organizational structure of CIs in the topology map by displaying selected CIs at different levels. When no folding rules are defined, the topology map displays all CIs included in the query results on one level by default. For example, if the query results include hosts and networks, both CIs are displayed on the same level in the topology map.



If an icon appears with a shadow in the topology map, it means that there is an additional CI layer beneath it, created by a parent or child folding rule.

You can define whether different folding definitions should be used for the To define different folding options for topology and tree presentations, you select the **Use different folding/grouping definitions** option in the view properties dialog box (for details, see "Using the Different Folding/Grouping Definitions Option" on page 27).

**To add folding rules:**

1. In the editing pane, right-click the relationship for which you want to define a folding rule.
2. Select **Add Rule**.
3. Select one of the following options to define the organizational structure between the two CI types connected by the selected relationship:
   - **Parent** – Select this option to display Node1 as the parent of Node2. That is, double-clicking the Node1 element on one map level displays a new map level with Node2 elements.
   - **Right Sibling** – Select this option to display Node1 elements wherever Node2 elements are displayed. That is, both CIs are displayed on the same map level.
   - **Left Sibling** – Select this option to display Node2 elements wherever Node1 elements are displayed. That is, both CIs are displayed on the same map level.

- **Child** – Select this option to display Node1 as the child of Node2. That is, double-clicking a Node2 element on one map level displays a new map level with Node1 elements.
- **None** – Do not define any folding rules for this relationship.

## Defining Multiple Relationship Rules

As an alternative to adding folding rules (for details, see "Adding Organization (Folding) Rules to Relationships" on page 94), you can choose to define multiple relationship rules.

Whereas you define organizational rules to define the structure of two nodes connected to a specific relationship, you define relationship rules to define the structure between CITs in a view.

For example, if one of the view's nodes is a host, you can specify different relationships for NT, UNIX, router, and so forth. By assigning different folding rules to different CITs, views can encompass and represent additional query data.

---

**Note:** If both folding and relationship rules have been defined, the definitions for the folding rules override the definitions for relationship rules.

---

This section includes the following topics:


- Defining Multiple Relationship Rules – for details, see page 96
- Editing Relationship Rules – for details, see page 97
- Deleting Relationship Rules – for details, see page 97

## Defining Multiple Relationship Rules

---

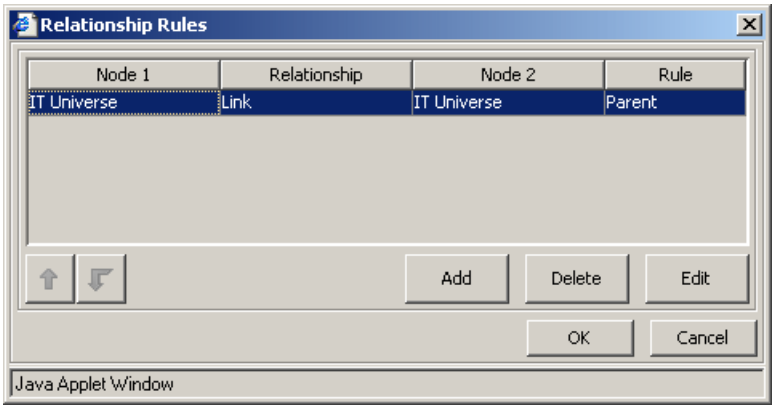
**Note:** You can define sibling and child rules for the same relationship between two CITs. However, you cannot define parent and child rules for the same relationship, or left sibling and right sibling rules.

---

**To define multiple relationship rules:**



**1** In the toolbar, click the **Relationship folding rules** button to open the Relationship Rules dialog box.




**2** Click **Add** to display the Add Rule dialog box.

**3** Select the required Node1 node in the **Node1** list.

**4** Select the required Node2 node in the **Node2** list.

**5** From the **Relationship** list, select an available relationship connecting the Node1 node to the Node2 node. (Enabled only after you defined both CITs.)

**6** From the **Rule** list, select a folding rule: **Parent**, **Right Sibling**, **Left Sibling**, or **Child**. For details, see "Adding Organization (Folding) Rules to Relationships" on page 94.

**7** Click **OK** to save the settings you have defined. The new rule is added to the Relationship Rules dialog box.

8 Repeat steps 2 to 7 to add another rule.

9 Click **OK**.

### Editing Relationship Rules

This section describes how to edit an existing relationship rule.

**To edit a relationship rule:**

1 In the Relationship Rules dialog box, select the rule you want to edit and click **Edit** to display the Add Rule dialog box.

2 Make the required changes.

3 Click **OK** to save the changes.

### Deleting Relationship Rules

This section describes how to delete an existing relationship rule.

**To delete a relationship rule:**

In the Relationship Rules dialog box, select the rule you want to delete and click **Delete**.

## Creating a Dependency Graph

You can create a graph that represents additional TQL query data related to a specific CI. The discovery pattern searches for the results from TQL query as well as the dependency graph definitions. The query recursively retrieves all the related CIs by a defined depth.

In the graph, you can define the relationship that is connected to a specific node. For example, if one of the nodes is a host, you can specify different relationships for NT, Router, and IP. You can also define attribute conditions for nodes. For details, see "Defining Attribute Conditions for Nodes and Relationships" on page 71.

The discovery pattern retrieves data that meets the criteria that is defined in the dependency graph.

This section contains the following topics:


- Defining a Dependency Graph – for details, see page 98
- Deleting a Dependency Definition – for details, see page 99
- Editing a Dependency Definition – for details, see page 99


## Defining a Dependency Graph

This section describes how to define a dependency graph.

**To define a dependency graph:**

1. Right-click the required node and select **Add SubGraph** to open the Dependency List dialog box.
2.  Click the **Add** button to open the Add Dependency dialog box.
3. From the **Node 1** list, select the desired node.
4. From the **Node 2** list, select the desired node.
5.  To define attribute conditions for the node you selected, click the **Condition** button to the right of the node. For information on how to define attribute conditions, see "Defining Attribute Conditions for Nodes and Relationships" on page 71.

---

**Note:** The **Condition** button is enabled only after you selected a node from the node list.

---

6. From the **Relationship** list, select an available relationship connecting Node 1 to Node 2. The list of available relationships appears only after defining both nodes.
7. Click **OK** to save the dependency map definition. The dependency map definition appears in the Dependency List dialog box.
8. In the **Depth** box, select a number that determines how many steps (that is, how many connecting nodes) in the TQL pattern you want included in the discovery process. The default is 5.
9. Click **OK** to save the changes you have made.

### Deleting a Dependency Definition

This section defines how to delete a dependency definition.

**To delete a dependency definition:**

1 From the Dependency List dialog box, select the row you want to delete.



2 Click the **Delete** button.

### Editing a Dependency Definition

This section defines how to edit a dependency definition.

**To edit a dependency definition:**

1 From the Dependency List dialog box, select the row you want to edit.



2 Click the **Edit** button to open the Add Dependency dialog box.

3 Edit the nodes and relationships as desired. For details, see "Defining a Dependency Graph" on page 98.



4 To define attribute conditions for a node, click the **Condition** button to the right of the node.

For details, see "Defining Attribute Conditions for Nodes and Relationships" on page 71.

5 Click **OK** to save the changes you have made.

## Showing TQL Node Instances

You can display all of the instances found for each TQL node in a table.

**To display all of the instances found for each TQL node in a table:**

1 In the editing pane, right-click the required TQL and select **Show element's instances** to open the Element instances dialog box.

The list of nodes 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 update the table, click the **Refresh** button.

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

## Editing a Pattern View

This section defines how to edit a pattern view.

**To edit a pattern view:**



1 In the Views pane of the **View Manager** tab, click the **Edit View** button of the view you want to edit. The Edit View page opens.

2 Make the required changes. For details, see "Pattern View Workflow" on page 54.

3 Click **OK** to save the changes you have made.

## Copying and Pasting TQL Nodes and Relationships

You can copy and paste TQL nodes and relationships and their definitions into the same view or another view definition.

**To copy and paste TQL nodes and relationships:**

1 Select the node(s)/relationship(s) you want to copy into the same view or another view. To select more that one item, hold down CTRL and click the required TQL nodes/relationships.

2 Right-click the selected items and click **Copy**.

3 In the view where you want to paste the selected item(s), right-click and select **Paste**.

## Saving the Pattern View

Once you have defined your pattern view, the last step is to save the TQL query to the CMDB so it can retrieve the data based on the definitions you have set. If you exit the View Manager without saving, a confirmation message is displayed, requesting that you save your changes.

**To save the pattern view:**

Click **OK** at the bottom of the Create View page. Mercury Business Availability Center takes you back to the **View Manager** tab where the view you have created is added to the list of views in the Views pane.

## 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*.

# Index


**A**

active views 39
attribute conditions
  defining for nodes and relationships 71
  defining in TQL queries 69
attributes
  grouping CIs by 91
  grouping objects by 88

**C**

CIs
  changing order in view 18
  grouping 88
CITs
  grouping by 88
configuration item types 59
Create View page
  understanding 58

**D**

deactive views 39
dependency definition
  deleting 99
  editing 99
dependency graph
  creating 97

**F**

folder pane 8
folders
  creating 16
  deleting 17
  managing 16
  predefined 12
  refreshing display 17
  renaming 17
folding options
  for views 26, 32
folding rules 35

**G**

getting more information vi
group by
  group label 88
  minimum in group 88
  nested groups 88, 92
grouping CIs 87
grouping options
  for views 26
  pattern views 29

**I**

instance views 5
  adding CIs to 46
  creating 46
  defining properties for new view 46
  editing 47
  understanding 45
  working with 45

**K**

KPIs
  selection, limiting in Dashboard views 25

**M**

minimum in group 88

## N

nodes
deleting 68

## O

organization rules
adding to relationships 94

## P

pattern views 5
creating 55
editing 100
workflow 54
working with 51
permissions
setting for views 40

## Q

qualifier conditions 76
query results
filtering in Attribute Condition tab 74
filtering in the Cardinality tab 78

## R

relationship cardinality
defining 77
relationship rules
defining multiple 95, 96
deleting 97
editing 97
relationships
defining 69
defining attribute conditions 69
deleting 68

## S

Service View Manager
defining multiple relationship rules 95
service views
understanding 4

## T

toolbars
for View Manager 101
Topology Query System
roles 53
Topology Query system
understanding 53
TQL nodes
adding to TQL queries 69
defining 69
defining attribute conditions 69
visible, in TQL queries 70
TQL queries
activating 101
adding nodes and relationships 60
attribute conditions 69
defining by adding nodes and relationships 60
visible TQL node 70
TQL relationships
adding to TQL queries 69
tree presentation 27

## V

View Manager
introduction 3
tab 8
tooltips 60
understanding 4
using toolbar options 101
working with 7
view properties
defining to create views 19
editing 19
views
active 39
creating 19
deactivating 39
different folding/grouping definitions 26
editing 19
folding options 32
folding rules 35
grouping options 29
in predefined folders 14

limiting KPIs to appear in Dashboard 25
managing 18
moving to different folder 18
name in red 39
predefined 12
topology/tree presentation 27
views pane 9
visible node 70

## W

who should read this guide vi