OPTIMIZE

MERCURY BUSINESS AVAILABILITY CENTER™

IT Universe Manager Administration

MERCURY™

BUSINESS TECHNOLOGY OPTIMIZATION

Mercury Business Availability Center

IT Universe Manager Administration

Version 6.5

Document Release Date: October 15, 2006

MERCURY™

Mercury Business Availability Center, Version 6.5
IT Universe Manager Administration

Mercury Interactive Corporation
379 North Whisman Road
Mountain View, CA 94043
Tel: (650) 603-5200
Fax: (650) 603-5300
http://www.mercury.com

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

AMLIB_ITUAD6.5/01

Table of Contents

Welcome to IT Universe Manager Administration

This guide describes how to work with the IT Universe Manager tab in CMDB Administration to build instance views and to manage the content of the IT universe model in the CMDB.

How This Guide Is Organized

The guide contains the following chapters:

Chapter 1   **Introduction to IT Universe Manager Administration**

Introduces the IT universe model and the concept of building views.

Chapter 2   **Getting Started with IT Universe Manager**

Describes the IT Universe Manager tab and the process of building views, and how to create new instance views.

Chapter 3   **Working with CIs in IT Universe Manager**

Describes how to build and modify views by adding CIs, and defining new CIs and relationships.

Chapter 4   **Configuration Item Properties**

Describes how to define and edit properties for a configuration item.

Chapter 5	**Working in Topology Map Mode**
	Describes how to work with CIs in the IT universe model and to build instance views when working in Topology Map mode.

Chapter 6	**Working with Dynamic Node Factory**
	Explains the concept of Dynamic Node Factory, and describes how to define and customize Dynamic Node Factory CIs.

Chapter 7	**Example Scenario for an Instance View**
	Provides a step-by-step scenario for defining an instance view for Dashboard.

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*.

Welcome

1

Introduction to IT Universe Manager Administration

This chapter introduces IT Universe Manager Administration and describes the origin of the configuration items in the IT universe model.

This chapter describes:	On page:
Using IT Universe Manager to Model Your IT World	1
Populating the IT Universe Model	2

Using IT Universe Manager to Model Your IT World

Mercury Business Availability Center enables you to build your own logical configuration in hierarchical tree format, according to how you view your organization. You can map key business processes at the top level of the tree to the relevant underlying IT infrastructure, drilling down to the leaves of the tree represent measurements collected by Business Process Monitor, SiteScope, or other monitoring tools.

The trees are contained in service views, and are built from the configuration items (CIs) in the Mercury Universal CMDB. Each CI in a tree can have one or more associated KPIs, which display different aspects of operational status for the CI when the view is deployed in the presentation layer.

Each CI and each relationship exists only once in the CMDB, as part of the IT universe model that mirrors your organization's IT world. You can add instances of the same CI to multiple service views, with a different relationship defined for the CI in each view.

The service views may be of type pattern or instance. (For an explanation of these view types, see "Instance Views and Pattern Views" in *View Manager Administration*.) You can build your own pattern views in the View Manager tab, by defining TQLs that display the hierarchy you want; however, when you require logical configurations that are tailored to the needs of your organization, you will generally build instance views in the IT Universe Manager tab.

The IT Universe Manager tab enables you to define and manage the CIs and relationships in your service views. When working in a service view in the IT Universe Manager tab, you can create new CIs, modify existing CIs, and delete CIs. All these actions have a direct affect on the whole IT universe model, not just the current view. For example, if you delete a CI in your view, that CI is deleted from the IT universe model—it will no longer appear in any of your views.

Populating the IT Universe Model

The IT universe model in the CMDB contains the CIs and relationships produced by the various CI-generating tools that operate within Mercury Business Availability Center. Most CIs and relationships originate from one of the following sources:

➤ the **source adapters**, which automatically create CIs and relationships for the monitor objects in Monitor Administration. For more information, see *Source Manager Administration*.

➤ the **discovery process**, which automatically creates CIs and relationships based on the discovered IT resources and topology in your company's network infrastructure. For more information, see *Discovery Manager Administration*.

➤ the **Dynamic Node Factory tool**, which automatically creates CIs and relationships based on data from incoming samples. For more information, see Chapter 6, "Working with Dynamic Node Factory."

➤ **IT Universe Manager**, which you use to manually create CIs and relationships that represent logical and physical elements of the business. For more information, see Chapter 3, "Working with CIs in IT Universe Manager."

➤ **Monitor Administration**, which enables you to manually create relationships between the CIs for monitor objects and CIs in the CMDB. For more information, see "Configuration Items and Monitor Objects" in *Working with Monitor Administration* and "Monitor Deployment Wizard" in *Working with SiteScope*.

In the IT Universe Manager tab, you can build service views where you map together the CIs and relationships from these different worlds contained in the CMDB. This functionality enables you to build views that meet your business requirements and objectives, and that monitor what is important to you.

2

Getting Started with IT Universe Manager

This chapter introduces the IT Universe Manager tab and functionality, and provides explanations for some of the concepts used when building views. It also describes how to create a new instance view when working in the IT Universe Manager tab, and provides an overview for building the view.

This chapter describes:	On page:
Working with the IT Universe Manager Tab	6
Working with Pattern and Instance Views in the IT Universe Manager Tab	8
Service View Tree Structure	13
Configuration Item Types	14
Overview for Building a Business View	19
Creating a New View	21

Working with the IT Universe Manager Tab

You access the IT Universe Manager tab from **Admin** > **CMDB** > **IT Universe Manager**.

The IT Universe Manager tab is divided into two panes:

➤ The left pane contains View Explorer, which enables you to select and display the required service view, and to search for configuration items (CIs) in views. For general information on working with View Explorer, see "Using View Explorer" in *Working with the CMDB*.

View Explorer lists the existing views (including empty views) that are defined as applicable for IT Universe Manager and for which you have viewing permissions (for details, see "Creating and Editing Views" and "Setting Permissions for Views" in *View Manager Administration*).

Note: View names shown in red text are inactive views and cannot be selected. For details, see "Activating or Deactivating Views" in *View Manager Administration*.

➤ The right pane displays information for the view currently open in View Explorer. The display format for this information depends on the selected viewing mode, as described in "Viewing Modes" on page 7.

Whichever viewing mode you are working in, either one or two buttons are displayed in the top-right of the pane, giving quick access to instance view functionality:
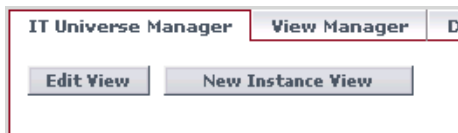
➤ **Edit View.** Displayed when an instance view is selected in View Explorer. Opens the Instance View Editor, where you can build and edit the view. For details, see "Editing with the Instance View Editor" on page 28.

➤ **New Instance View.** Opens the Create View - View Properties dialog box, where you define properties to create a new instance view. For details, see "Creating a New View" on page 21.

Viewing Modes

The IT Universe Manager tab works in two viewing modes: **Default** and **Topology Map**. When you first access the tab, it uses the default viewing mode. You can change to topology map viewing mode by selecting that option from the **Viewing Mode** list (top-right corner). The same functionality is available from both modes, the difference is in how the CI information is presented in the GUI:

➤ In **Default** viewing mode, the IT Universe Manager tab opens at the welcome page, which provides basic information on getting started, and provides links for creating new instance views (described in "Creating a New View" on page 21), and adding CIs without creating new relationships (described in "Working with CIs in IT Universe Manager" on page 23). The information and specific links presented depend on the view type (pattern or instance). For details, see "Working with Pattern and Instance Views in the IT Universe Manager Tab" on page 8.

The welcome page is displayed whenever you are working at the highest level of the view (you can return to the welcome page at any time by clicking the view name in View Explorer in the left pane).

In Default viewing mode, when you select a CI in View Explorer, the right pane displays information for the CI in two tabs:

➤ The **Properties** tab provides information on the properties for the selected CI, and enables you to edit the properties. For details, see "Configuration Item Properties" on page 71.

➤ The **Related Configuration Items** tab lists the relationships for the selected CI, and provides functionality to add additional related CIs, or to remove relationships. For details, see "Working with CIs in IT Universe Manager" on page 23.

➤ In **Topology Map** viewing mode, the right pane displays the CIs in the current view organized into single-level or multi-level topological maps. You can create new instance views, add CIs, add relationships, edit CI properties, and delete CIs and relationships, just as in default viewing mode. For details, see "Working in Topology Map Mode" on page 81.

Working with Pattern and Instance Views in the IT Universe Manager Tab

There are two types of service views: pattern and instance. Pattern views are defined in View Manager, while instance views are generally created and built in the IT Universe Manager tab. The functionality for working with the two view types in the IT Universe Manager tab differs somewhat, as described in the following sections.

For information on the different view types and how they function, see *View Manager Administration*.

Note: You must have **Change** permissions for a view in order to edit it. If you do not have the necessary permissions, contact your Mercury Business Availability Center Administrator.

This section includes the following topics:

➤ "Working in Instance Views" on page 9
➤ "Working in Pattern Views" on page 9

Working in Instance Views

Mercury Business Availability Center provides one default, empty instance view, the First Service View. Any further instance views you create will also be empty until you manually add CIs to the view.

You can add any sort of CI to an instance view, either by adding existing CIs or by creating new CIs within the view. When you create a new CI in the view, the CI (and any associated relationship) is also created in the Mercury Universal CMDB, and will automatically be added to any other view where the new CI is relevant. For example, if the parent of the created CI appears in another instance view and is set to show its child CIs, then the new child CI will automatically be added to the parent in that view; or if the new CI fits the requirements of the TQL query in a pattern view, then it will automatically appear in the pattern view.

The welcome page for an instance view (displayed when working at the highest level of the view) provides functionality for:

➤ creating new non-related CIs, as described in "Creating a New CI" on page 46.

➤ adding existing CIs to the view, and other edit functionality, as described in "Editing with the Instance View Editor" on page 28.

➤ creating new instance views, as described in "Creating a New View" on page 21.

Working in Pattern Views

CIs created by Discovery Manager or Source Manager will automatically be added to each pattern view where the CIs (and their relationships) fit the TQL definition for the view. When you open a pattern view in View Explorer, the TQL query for the view queries the CMDB for CIs that match the defined pattern. If there are no matches (for example, in the case of a view for a discovery process where the process has not yet been run) then the view will be empty.

When you create a new CI while working in any view in the IT Universe Manager tab, the CI (and any associated relationship) is created in the CMDB. The CI is also added to your current view if you are working in an instance view. However, if you are working in a pattern view, the new CI (and relationship) is added to the view only if it meets the TQL specifications for the view. If the created CI and relationship do not meet the TQL requirements, then the CI and relationship will be created in the CMDB and not added to the view. Similarly, if you try to attach an existing CI to a CI in the view, and the relationship does not meet the TQL requirements, then the attached CI will not appear in the view, but the relationship is still created in the CMDB.

The New CI Wizard, or a message box in the top-right corner in the CMDB without adding them to the view.

Summary

The following configuration items were created in the CMDB but will not appear in the current view:

Name	CI Type
host group	Group

The following relationships were created in the CMDB but will not appear in the current view:

Relationship Type	CI 1	CI 2
Depends On	host7	host group

Some of the relationships were created in the CMDB but are not part of the current view.

If a created CI and its relationship conform to the TQL queries defined for other pattern views, it will automatically appear in those views when the views are opened in View Explorer.

For example, a TQL defined for a pattern view in View Manager includes the nodes for **Group** and **Host** with relationship **Depends On**.
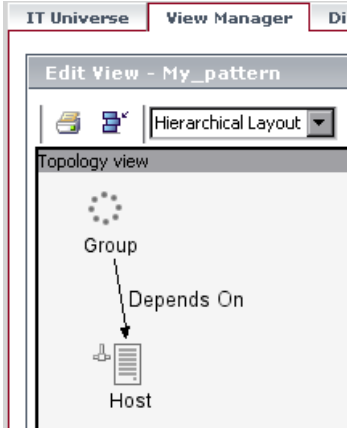
In an instance view in the IT Universe Manager tab, a Group CI, working group, is created with a **Depends On** relationship to a child CI of type host, host machine.
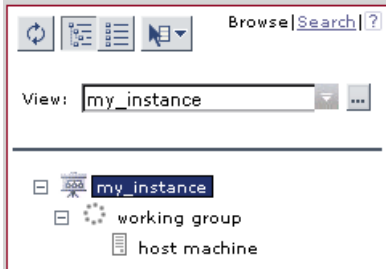
The new CIs and relationship fit the TQL definition in the pattern view, so they are automatically added to the pattern view.

The welcome page for a pattern view (displayed when working at the highest level of the view) enables you to:
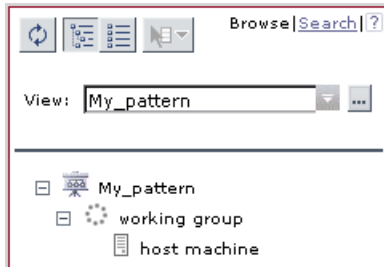
➤ open the New CI Wizard to create a new non-related CI, as described in "Creating a New CI" on page 46.

➤ create new instance views, as described in "Creating a New View" on page 21. (You cannot create new pattern views from the IT Universe Manager tab.)

For details on creating and working with pattern views and TQLs, see "Working with Pattern Views" in *View Manager Administration*.

Note: Changes to pattern views take longer to update to the view than changes to instance views.


Service View Tree Structure

Service views are organized into an hierarchical tree format; in View Explorer, this tree structure is shown inverted. The highest node in the tree shows the view name, displayed beside the view icon. The view name is not a CI, but rather acts as the containing folder for the CIs in the view.

At the next level of the View Explorer tree are the nodes for the root CIs. A root CI is the starting point for a view subtree. A view may contain any number of subtrees. Under the root CIs come the branch CIs, expanding down to the leaf CIs as the last nodes at the ends of each branch.

The CIs at each level in a subtree are considered to be attached to the CI on the branch directly above. This attachment generally depicts a dependant relationship, so that the higher branch (closer to the root CI) represents a component of your business that depends on the CIs on the next branch down (closer to the leaf CIs) in the business process. For example, an application depends on the servers running the application.
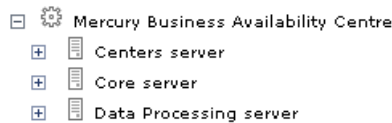
The attachment between a higher node and lower node is referred to as a parent-child relationship. The parent CI may have many attached child CIs, and the child CIs pass operational status to the parent. The status at each level is viewed in Dashboard.

When defining an instance view, you build your own topology for the subtrees, according to the business structure or process you want represented. For more information, see "Overview for Building a Business View" on page 19.

When working with pattern views, the tree topology is determined by the pattern defined for the view. For more information on defining pattern views, see "Working with Pattern Views" in *View Manager Administration*.

Note: If required, you can manually reorder CIs within a branch in the tree. For details, see "Managing Views" in *View Manager Administration*.


Configuration Item Types

Every CI must belong to a class of CIs, called a CI type (CIT). There are four main categories of CITs:

➤ **Business.** CITs that correspond to the logical elements of your business, such as processes and organizational groups.

➤ **System.** CITs that correspond to the physical elements (hardware and software) installed in your business environment.

➤ **Dynamic Node Factory.** CITs that dynamically generate CIs in response to incoming data from your data sources.

➤ **Monitor.** CITs that handle incoming metrics collected from your business universe:

  ➤ Monitor-type CIs under this category are added by the source adapters, as described in *Source Manager Administration*.

  ➤ UDX Measurement Filter CIs are added when you create a measurement filter, as described in "Working with Measurement Filters" in *Platform Administration*.

  ➤ Diagnostics Probe CIs are added when you integrate with Mercury Diagnostics, as described in *Mercury Diagnostics Installation and Configuration Guide*.

Monitor CIs cannot be manually created in the IT Universe Manager tab, but you can attach them to other CIs and edit some of their properties. For more information on the role of monitor CIs, see "Role of Monitor CIs in a View" on page 18.

Each CIT has a model that provides the template for creating the CI and defines the properties that are specific to that class of CIs. The CI type models are defined in the CI Type Manager, where you can edit the models, and define new CITs, if required. In the Edit Configuration Item Type dialog box for a CIT model, you can view a description for each CIT, and view its properties. For details, see "Managing CITs" in *CI Type Manager Administration*.

The different CIT categories are represented by various icons in View Explorer, New CI Wizard, and other pages, for example:
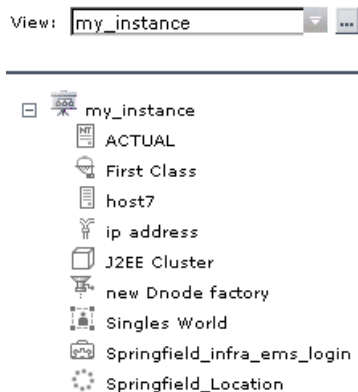

When defining a new CI using the New CI Wizard, on the first page you must select the CIT for the CI. The wizard provides a list of the CITs that are relevant at the location where you are adding the new CI. (For pattern views, you have the option to override this list and select from all possible CITs.)

The list of relevant CITs depends on:

➤ **the view definition.** The TQLs for pattern views define which CITs and relationships can be added to the view.

For example, when creating a new CI in a Host Resources (pattern) view, only the System CITs for the **IP** CIT group and for the **Host** CIT group, are relevant for the view.
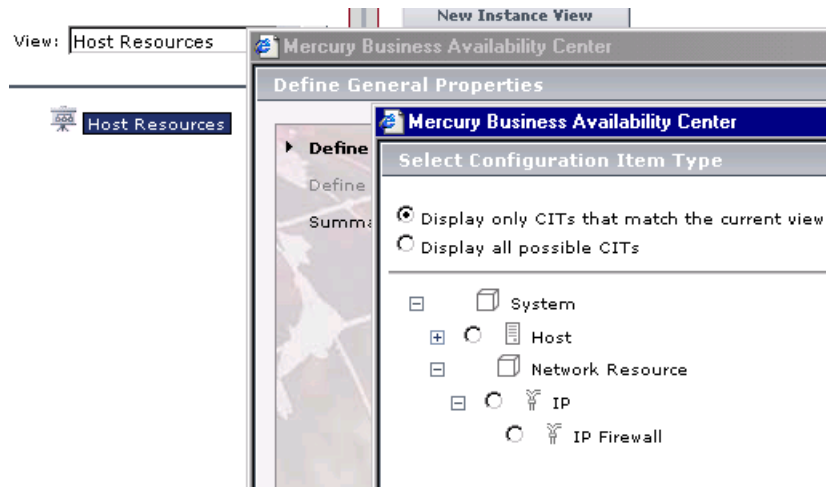
For more information on CIs in pattern views, see "Working in Pattern Views" on page 9.

➤ **the definition for the selected CI.** The CI Type Manager defines which other CITs can be logically related to each CIT. For example:

➤ If the selected CI is of type **Business Process Group**, the CIT list includes relevant CITs under the **Business** category, but excludes the **System** category. (System CITs would not be logically added to a business process group.)
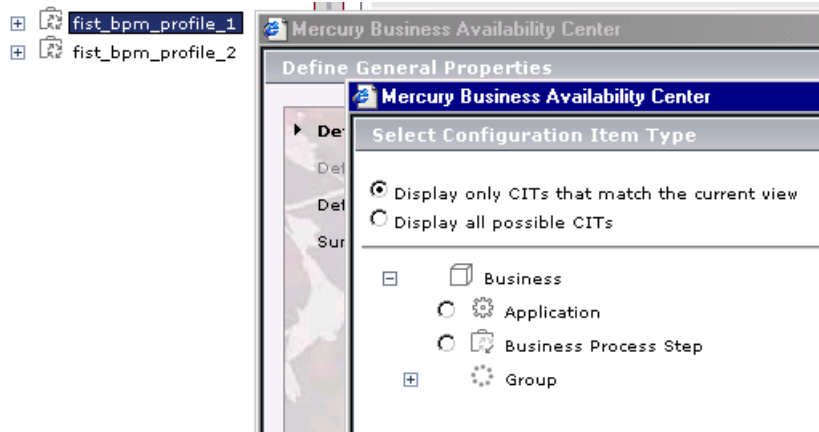
➤ If the selected CI is of type **J2EE Cluster**, the CIT list includes (under the System category) only the **J2EE Managed Object** CIT group, and certain relevant **Application Resource** CITs.
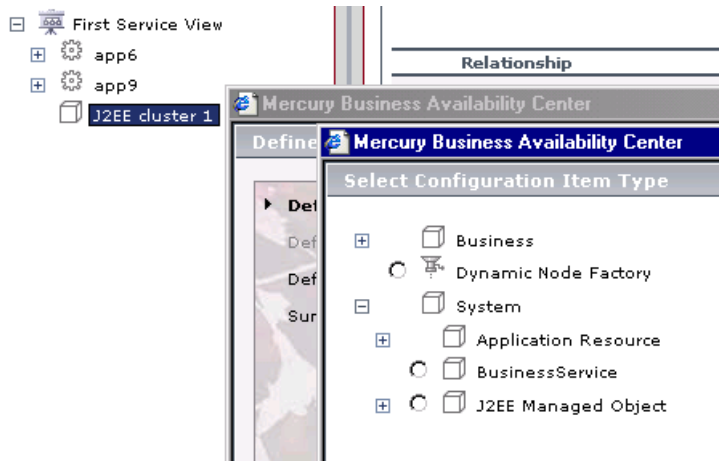
Most System CITs and the Dynamic Node Factory CIT have properties that are specific to the type (some with default values). When creating a new CI for one of these CITs, you define the values for these properties on the Define CIT-Specific Properties page of the New CI Wizard. Some of the properties are mandatory.

For CIs created by the source adapters or by the discovery process, the required CIT-specific properties are defined automatically. Some of these properties can be edited in the **Other Properties** area of the IT Universe Manager Properties tab.

Role of Monitor CIs in a View

In the Mercury Business Availability Center Dashboard and Service Level Management applications, the CIs display colored indicators (KPIs) to show operational status information for the components of your IT universe. To give status color to a KPI, it needs to have a dependency on one or more of the leaf CIs that receive actual monitoring information, for example, a SiteScope monitor or Business Process Monitor transaction. (For information on KPIs, see "Configuring KPIs" in *Application Administration*.)

The dependency can be direct (in which case the parent CI has a **monitored by** relationship with the monitor CI child), or through the branches of a subtree. For example, a logical CI for location could receive KPI status through either one of the following hierarchies:

➤ **Location CI -> monitor leaf CI**

➤ **Location CI -> group CI -> group CI -> monitor leaf CI**

Mercury Business Availability Center automatically creates a dependency connection to a monitor CI for all CIs added using the source templates in the Source Manager, or added using the Dynamic Node Factory template. However, CIs added by the discovery process, or that you create manually, are not automatically associated with monitor CIs; you must manually define the appropriate connections.

You can manually create a monitor dependency for a CI in one of the following ways:

➤ By attaching a monitor CI (or a subtree containing a monitor CI) to the parent CI, in the IT Universe Manager tab. For details, see "Attaching Existing CIs" on page 55.

➤ By making an association between a monitor object in the Monitor Administration tree and an existing CI. The monitor CI created for the monitor object will automatically be attached to the selected CI. For details, see "Configuration Items and Monitor Objects" in *Working with Monitor Administration*.

➤ By using Monitor What Matters to create a new SiteScope monitor for a specific CI. The created monitor will generate a new monitor CI in the CMDB, with an automatic attachment to the selected CI. For details, see "Monitor Deployment Wizard" in *Working with SiteScope*.

Overview for Building a Business View

You use the IT Universe Manager tab to build and manage the CI hierarchy of the service views. These views are used in Dashboard to monitor the business environment. The service views should provide the business users using Dashboard with logical representations of your company's business structure, processes, and goals.

To meet these requirements, you will generally build multiple instance views representing different aspects of your business; for example, you may build business views that revolve around:

➤ company structure, so that data is organized by the geographical location of branches, or by data centers, or so forth.

➤ business processes, so that data is organized by line of business, or by applications, or so forth.

➤ business goals, so that data is organized by sales, or by key customers, or so forth.

Of course, the views you create will depend on the particular demands of your organization, and the actual hierarchical structure of the views will reflect those demands.

Whatever aspect of the business a view is intended to represent, the most usual hierarchical format is as follows:

➤ **Top level.** The top-most levels of the view, consisting of the root CI and the branches below it, will normally be built from logical **Business** CI types such as **Location**, **Line of Business**, **Customer**, **Application**, and so forth.

➤ **Intermediate level.** The intermediate branches of the view will normally consist of: **System** CI types, representing actual hardware and software; and/or monitoring group CI types, representing groups and profiles containing monitors.

➤ **Bottom level.** The bottom-most level of the view, consisting of the leaf CIs at the end of the subtrees, is built from the **Monitor** CI types such as **BPM Transaction**, **SiteScope Measurement**, and so forth. The leaf CIs in a view should all be of type **Monitor**; these CIs receive the actual data from the data sources, used to determine KPI status all the way up the view subtrees, and so provide business context to the view.

In some cases, you may choose to use the **Dynamic Node Factory** CI type to dynamically build a hierarchy for you. For details, see Chapter 6, "Working with Dynamic Node Factory."

Summarized Steps for Creating and Building Your Instance View

The following gives a summary of the steps for creating a typical business view.

A step-by-step process chart for creating views is provided in "How Do I Reflect My IT Environment in Mercury Business Availability Center?" in *Getting Started with Mercury Business Availability Center*.

A full example scenario for creating a business view is provided in Chapter 7, "Example Scenario for an Instance View."

Steps to create and build an instance view:

1 Create a new instance view. For details see "Creating a New View" on page 21.

2 Create new (non-related) CIs to form the roots of your view subtrees. For details, see "Creating a Non-Related CI" on page 46.

3 For the intermediate branches of the subtrees, create new related CIs (for details, see "Creating a Related CI" on page 47), or attach existing CIs (for details, see "Attaching Existing CIs" on page 55).

4 For the leaf CIs of the subtree, attach monitoring CIs. For details, see "Attaching Existing CIs" on page 55.

Note: All the tasks described in steps 2 through 4 can be performed in the Instance View Editor. For details, see "Editing with the Instance View Editor" on page 28.

---

5 Define user permissions for the view. For details, see "Setting Permissions for Views" in *View Manager Administration*.

Creating a New View

From the IT Universe Manager tab you can create a new instance view. There are two stages to the process of creating a new view: First you define the properties to create the view framework, as described in the procedure below; then you build the view by adding CIs, as described in Chapter 3, "Working with CIs in IT Universe Manager."

You can also create new instance views and new pattern views in View Manager. For details, see "Creating and Editing Views" in *View Manager Administration*.

You can select to hide the view in 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).

Note: After creating the view, if you want to then modify the view properties, you must do so from View Manager. For details, see "Working with View Manager" in *View Manager Administration*.

To create an instance view from the IT Universe Manager tab:

1 In the **IT Universe Manager** tab, click the **New Instance View** button

New Instance View   at the top of the tab. (When working in an instance view, you also have the option of selecting the view name and clicking the **Create a new empty instance view** link in the right pane.)

2 Define the properties for the view. For details, see "Creating and Editing Views" in *View Manager Administration*.

Note the following:

➤ The view will be created in the same folder as the view currently displayed in View Explorer. If you want to create it in a different folder, click the folder name displayed in the **Create View in Folder** field.

➤ If you deselect the **IT Universe** check box, then after you close the Create View - View Properties dialog box, the view will not appear in the **View** list in the IT Universe Manager tab.

3 Click **OK**. View Explorer opens the new view in the IT Universe Manager tab, ready to add CIs.

3

Working with CIs in IT Universe Manager

This chapter describes how to use the IT Universe Manager tab to manage configuration items (CIs) and relationships in your service views, and to build views.

About Working with CIs in IT Universe Manager

The topics in this chapter describes the functionality available from the **Related Configuration Items** tab when working in default viewing mode. However, the same CI functionality is available when working in topology map viewing mode (as described in "Working in Topology Map Mode" on page 81).

Related Configuration Items Tab

The Related Configuration Items tab is displayed in the IT Universe Manager tab by clicking any CI in View Explorer and selecting **Related Configuration Items** in the right-hand pane.

The relationship table in the tab lists the relationships for the CI currently selected in View Explorer. The table only includes the relationships that exist in the current view for the selected CI; relationships for the CI that exist in the Mercury Universal CMDB but are not part of the CI definition in the view are not shown.

(You can view a list of the relationships that exist for a CI in all views, by right-clicking the CI in View Explorer and selecting **Show Related CIs**. For more information, see "Searching by Related Configuration Item" in *Working with the CMDB*.)

If the relationship list is very long, the entries are divided up between pages; in this case, the page scroll arrows in the top-right corner of the tab are enabled.

The relationship table contains the following columns:

➤ **Relationship.** Describes the type of relationship between the CI selected in View Explorer and the related CI. For more information, see "Working with Relationships" on page 63.

➤ **Direction.** Describes the direction of the relationship between the two CIs. The selected CI in View Explorer is to the left of the arrow, the related CI is to the right. For example, when the selected CI in View Explorer is Parent group, the relationship to Child group is shown in the table as follows:

The arrow shows that you read the relationship from left to right, meaning that Parent group depends on Child group.

When the selected CI in View Explorer is Child group, the relationship to Parent group is shown in the table as follows:

The arrow shows that you read the relationship from right to left, meaning, again, that Parent group depends on Child group.

Tip: Place the cursor over the direction arrow to see a tooltip with a descriptive summary of the relationship.

➤ **CI Name.** Displays the name of the CI related to the selected CI.

➤ **CI Type.** Displays the configuration item type (CIT) for the related CI.

➤ **Edit Relationship button.** Opens the Relationship Properties dialog box, where you can edit certain properties for the relationship. For details, see "Editing Relationship Properties" on page 65.

Tip: You can sort the information displayed in the table by clicking on a column header. You can switch the column between ascending and descending order by clicking the header again.

The buttons under the relationship table provide the following functionality:

➤ **Delete Relationship.** Enables you to delete relationships from the view and CMDB. For details, see "Deleting a Relationship" on page 68.

➤ **New Related CI.** Enables you to create a new CI that is related to the selected CI in View Explorer. For details, see "Creating a New CI" on page 46.

➤ **Attach Related CIs/Attach Monitor CIs.** Enables you to attach existing CIs to the selected CI in View Explorer. For details, see "Attaching Existing CIs" on page 55.

Methods for Adding CIs to a View

There are a number of ways in the IT Universe Manager tab for managing your view content. The functionality provided by each method is as follows:

➤ **Add CIs and perform most editing tasks.** The Instance View Editor provides a single edit tool from which you can perform all creating CI and relationship operations; the Instance View Editor also acts as a workspace for trying out changes to your views before the changes are updated to the CMDB. For details, see "Editing with the Instance View Editor" on page 28.

➤ **Create new CIs without creating relationships.** You can create new CIs that are added at the root level of the view, so that no new relationships are created. For details, see "Creating a Non-Related CI" on page 46.

➤ **Create new CIs and create relationships for them.** You can create new CIs and attach them to existing CIs in the view, so creating a new CI and a new relationship in the CMDB. For details, see "Creating a Related CI" on page 47.

➤ **Add existing CIs and create relationships for them.** You can copy CIs and CI subtrees from another view to the current view, and attach them to an existing CI in the view, so creating new relationships for existing CIs. For details, see "Attaching Existing CIs" on page 55.

Editing with the Instance View Editor

The Instance View Editor enables you to add and remove CIs in your instance views. The changes made to a view using the Instance View Editor are not applied in the system until you choose to save them; this enables you to build your view more quickly than with other IT Universe Manager editing tools, and puts less weight on the system during the view building process.

This section includes the following topics:

Accessing the Instance View Editor

The Instance View Editor is used only for instance views. If you do not have modifying permissions for the view, then the **Edit View** button that opens the editor is disabled. (For details, see "Setting Permissions for Views" in *View Manager Administration*.)

For information on understanding Instance View Editor functionality, see "Understanding the Instance View Editor" on page 30.

Note: While you are working in the Instance View Editor, other users can also make changes to the view. Changes that are saved may affect your work as follows:

➤ Changes made by other users are displayed in the view after you save your changes, or after closing the Instance View Editor.

➤ Changes made by other users may override your changes, or vice versa.

➤ Changes made by other users (for example, deleting CIs) may cause the save operation to fail (when you click **OK** or **Apply**).

To access the Instance View Editor:

1 In the **IT Universe Manager** tab, display in View Explorer the instance view you want to edit.

2 In the right pane, click **Edit View**.

The Instance View Editor opens, displaying the name of view to be edited in the title bar.

Note: The Instance View Editor for a view can also be opened from View Manager, and is opened automatically after you create a new view.

Understanding the Instance View Editor

The Instance View Editor contains two panes. In the View Explorer pane on the left, you can traverse the views that exist in Mercury Business Availability Center, and view information for CIs, such as CI properties. In the edit pane on the right, you build and modify the contents of your view.
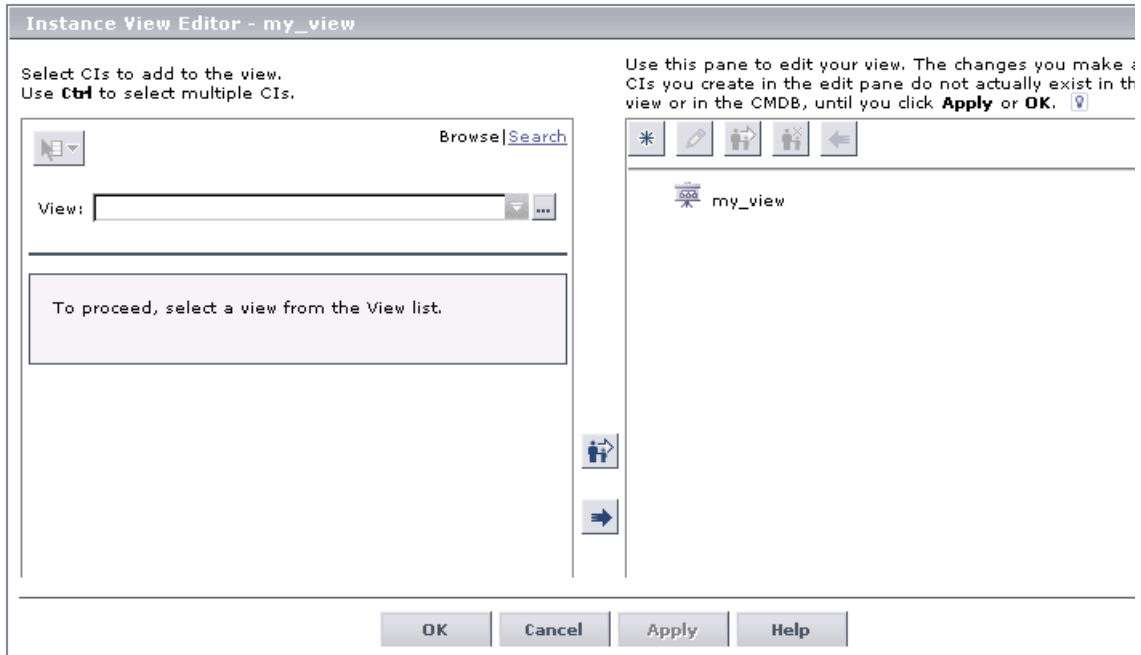
This section includes the following topics:

➤ "Left Pane Functionality" on page 31

➤ "Edit Pane Functionality" on page 31

Left Pane Functionality

You use the left pane to traverse the views in the system and select CIs, when you want to add CIs from other views to the view being edited. For more information, see "Adding Existing CIs to the View Using the Instance View Editor" on page 34.

The left pane uses View Explorer functionality for searching and viewing additional information for CIs. For more information, see "Using View Explorer" in *Working with the CMDB*.

Edit Pane Functionality

The right side of the Instance View Editor contains the edit pane for making changes to the view being edited. Your changes are not updated to the CMDB until you click **OK** (to save your changes and exit) or **Apply** (to save your changes and continue working in the Instance View Editor). If you make a mistake, you can click **Cancel** to exit without saving, and your unsaved changes will not be implemented.

Note: If you navigate away from IT Universe Manager tab without clicking **OK** or **Apply**, your changes are not updated to the CMDB.

The buttons and right-click menus provide access to the available edit functions, as described in "Adding Existing CIs to the View Using the Instance View Editor" on page 34, and "Additional Functionality in the Instance View Editor" on page 41.

Each CI in the edit pane has a tooltip showing:
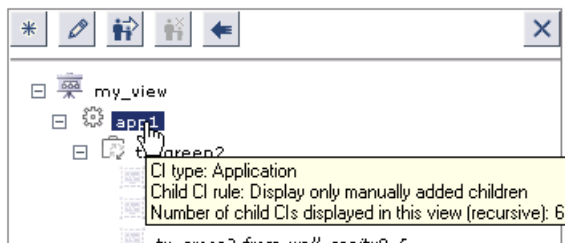
➤ the CI type, for example, Application.

➤ the child CI rule in effect for the CI. This states **Display all CMDB children** if the CI is set to include all its child CIs, or **Display only manually added children** if the CI is set to exclude all child CIs apart from those added to it within the current view. For more information, see "Child CI Rule Functionality" on page 32.

➤ the total number of CIs in the subtree under the CI, within the current view.

Child CI Rule Functionality

The child CI rule assigned to a parent CI determines what child CIs are displayed for the parent CI in the view.

When KPI status is calculated for a CI and displayed in Dashboard, the status is based on all child CIs that exist in the CMDB for the parent CI; however, in order to restrict the number of CIs actually displayed in the view, you have the option to hide all child CIs, or only display selected children. This has no affect on KPI status—status will still be calculated based on all child CIs, whether they are displayed in the view or hidden.

Displaying or hiding child CIs depends on the child CI rule that is applied to the parent:

➤ A CI defined as **Display all CMDB children** will recursively display in the view all child CIs that exist in the CMDB. The subtree is dynamically updated with every added child CI, whether it is added within the current view or in a different view.

In the edit pane of the Instance View Editor, all child CIs that are included recursively are displayed in italic text and with a faded icon.
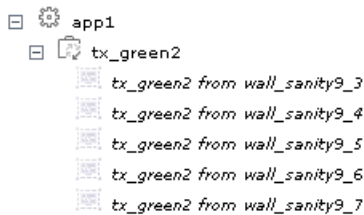
Note: The recursively-included child CIs are all marked in their tooltips as **Display only manually added children**, but in fact will dynamically display all child CIs in the CMDB.

➤ A CI defined as **Display only manually added children**, displays only those child CIs that were created or added to that CI within the current view. In the edit pane of the Instance View Editor, these child CIs are shown in the standard format used for CIs. In the following picture, CI **app1** is set as **Display only manually added children**:

However, the child CI of a parent with this rule can itself be set to use either rule. In the following picture, the rule for **tx_yellow2** has been changed to **Display all CMDB children**:

Note: In Dashboard, you can view the hidden children for a CI. For details, see "Find Visible and Hidden Child CIs Option" in *Using Dashboard*.

You can toggle the rule used by a parent CI, as described in "Changing the Display/Hide Child CIs Status" on page 43. For more information on how the child rules are applied in IT Universe Manager, see "Include Related CIs" on page 77.

Adding Existing CIs to the View Using the Instance View Editor

The Instance View Editor enables you to add any CIs in the CMDB to your view, by adding them from other views in the system.

When a CI is added to your view under another CI, a relationship is automatically created between the two CIs, using the default relationship for the two CI types. This will generally be a **depends on** relationship (for example, between two logical CIs) or a **monitored by** relationship (between a logical CI and a monitor CI). If there is no valid relationship between the two CIs (for example, between two monitor CIs), then a message is displayed saying that no valid link was found between the two items.

This section includes the following topics:

➤ "Example of Adding CIs with Children to the Edited View" on page 37
➤ "Example of Adding CIs without Children to the Edited View" on page 40

To add existing CIs to the edited view:

The following picture illustrates the procedure:
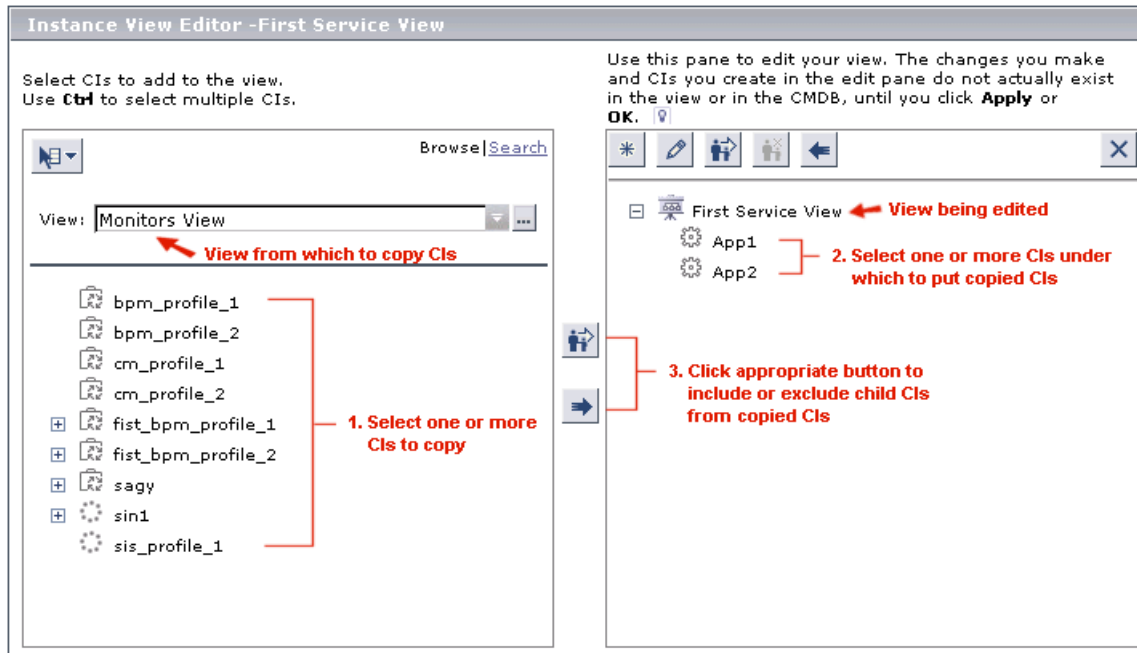
1 In the left pane of the Instance View Editor, select the required view from the **View** list (this is the view containing the CIs you want to add).

2 In the left pane, expand the tree and select the CI that you want to add. Use the CTRL key to select multiple CIs. You can select multiple CIs at different levels of the tree.

3 In the edit pane (the right pane of the Instance View Editor), select the CI that you want to add the CIs under. You can select multiple CIs using the CTRL key—the added CIs will be repeated under each selected CI.

Note: If you do not select a CI in the edit pane, then the CIs are automatically added at the root level in the view.

---

4 Add the selected CIs to the view being edited using one of the buttons between the panes:

➤ Click the recursive arrow button to add the CIs with **Display all CMDB children** status, so that the CIs are added to the edited view together with all recursive child CIs from the CMDB.

➤ Click the single arrow button to add the CIs with **Display only manually added children** status, so that the CIs are added to the edited view without any child CIs.

For more information on the significance of each status, see "Child CI Rule Functionality" on page 32.

After clicking one of the buttons, the CIs currently selected in the left pane are added as children to each selected CI in the edit pane (with or without the children of the added CIs, as appropriate).

Note: After adding a CI with its children to the edit pane, the edit pane displays the child CIs that were displayed in the left pane; there may be other recursive children for the added CI in the CMDB, but in order to give faster performance, Instance View Editor does not check the database until the changes are saved. After clicking **Apply**, the edit pane is updated to show all recursive CIs from the CMDB.
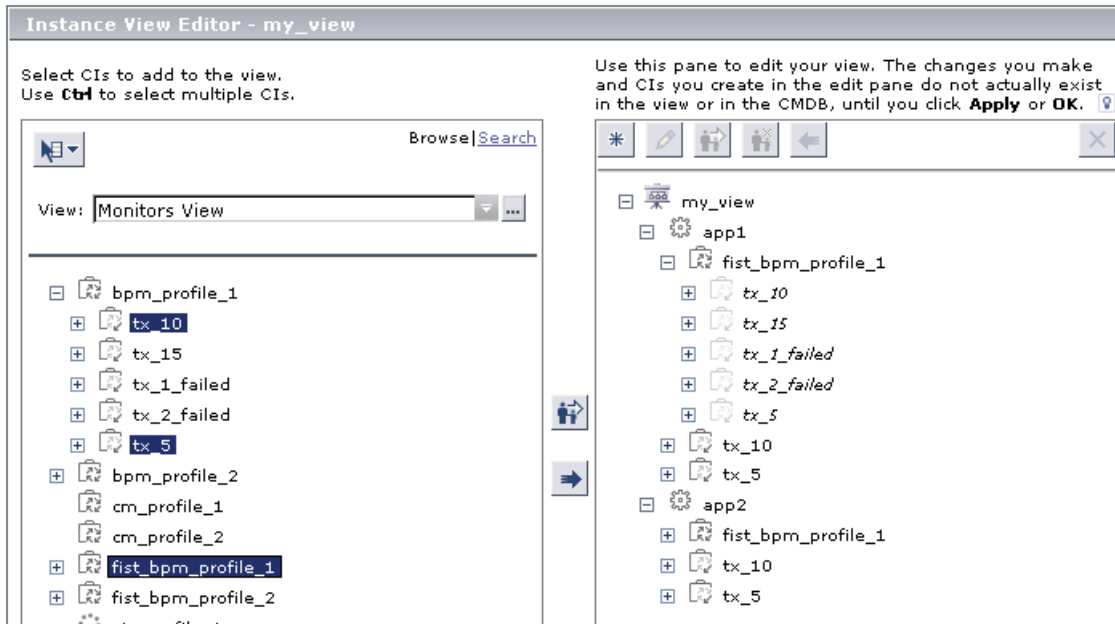
---

5 Click **OK** or **Apply** to update your changes to the CMDB.
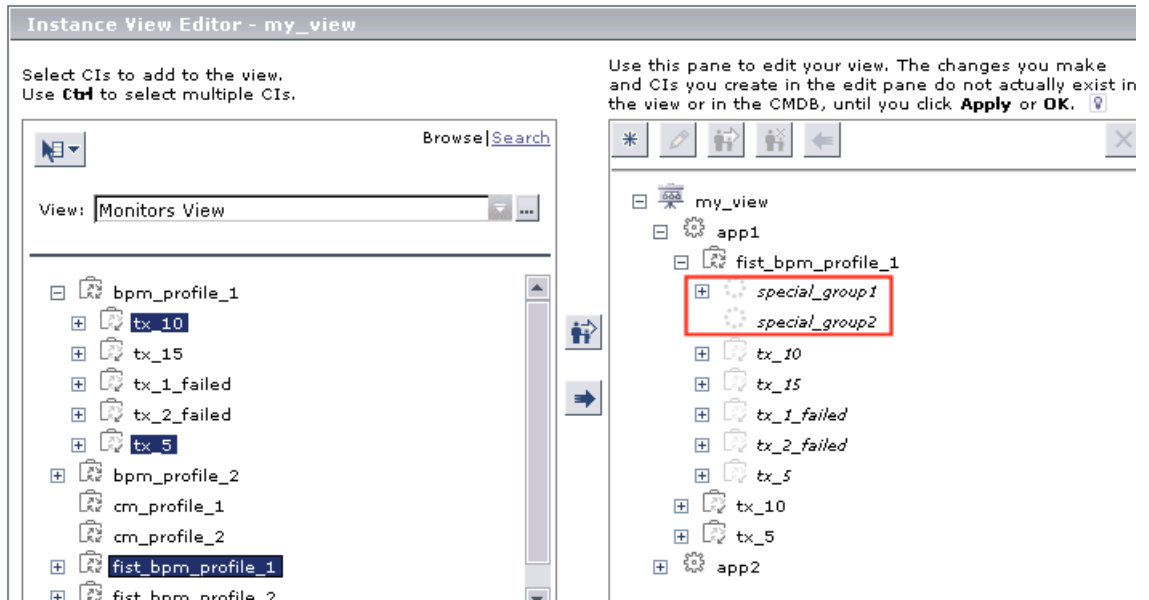
Example of Adding CIs with Children to the Edited View

➤ Select one or more CIs in the left pane and in the edit pane (using the CTRL key for multiple selection), then click the button to add with children:
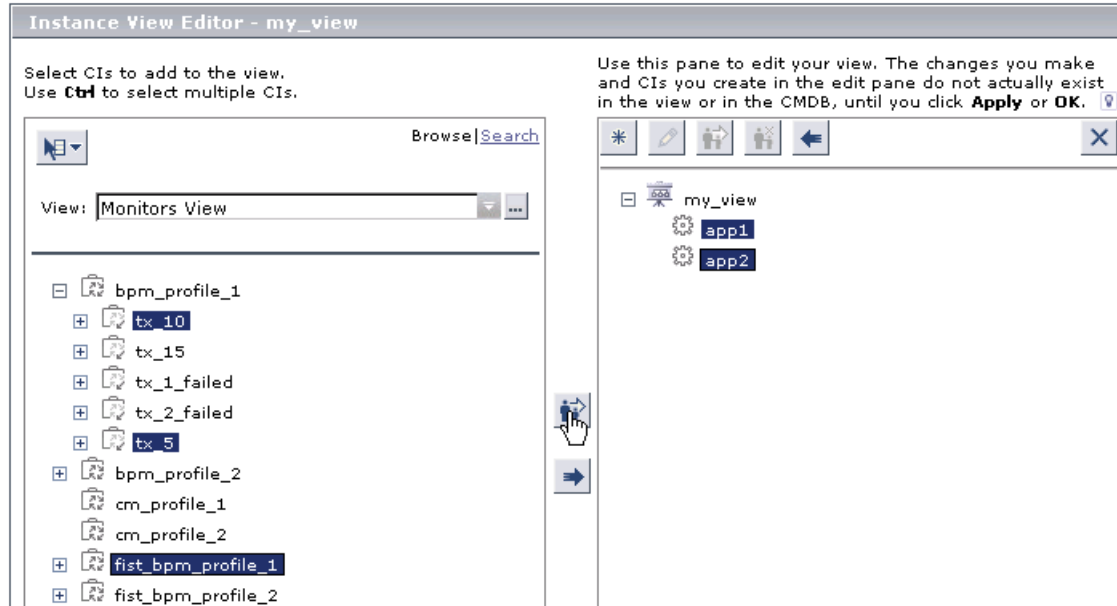
➤ The added CIs are displayed in the edit pane under the selected parent CIs, and the child CIs for the added CIs (with faded icons and italicized text) are also displayed in the edit pane, under the added CIs:

➤ Click **Apply** to update the view with any additional child CIs in the CMDB. In this example, fist_bpm_profile_1 had two additional children that were not visible when it was first added to the edit pane; after clicking **Apply** the Instance View Editor checks the database and adds them to the view.:

Example of Adding CIs without Children to the Edited View

➤ Select one or more CIs in the left pane and in the edit pane (using the CTRL key for multiple selection), then click the button to add without children:
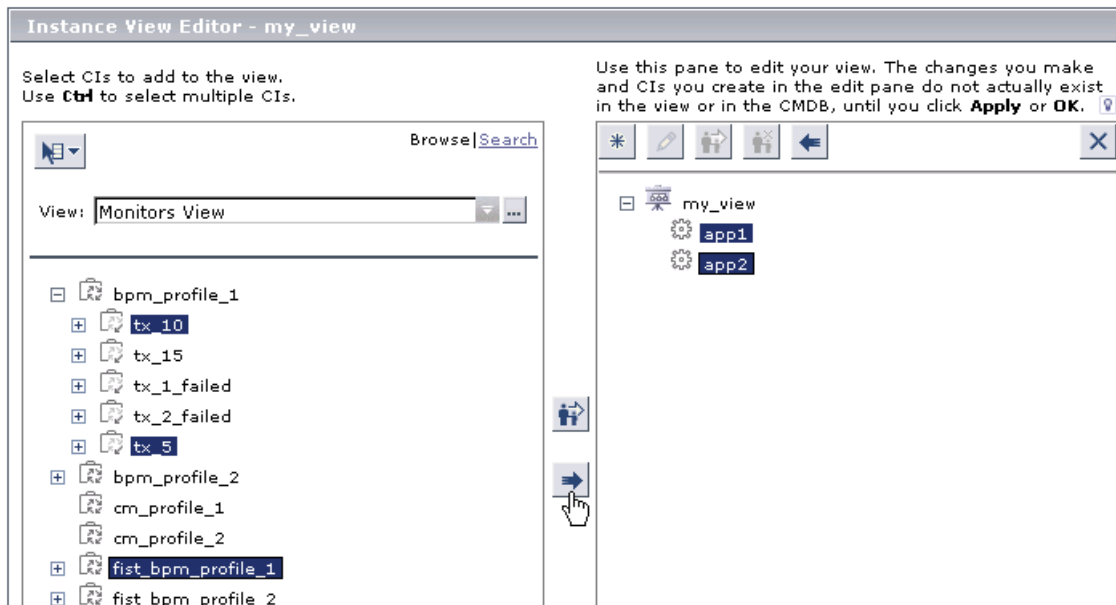
➤ The added CIs are displayed in the edit pane under the selected parent CIs.
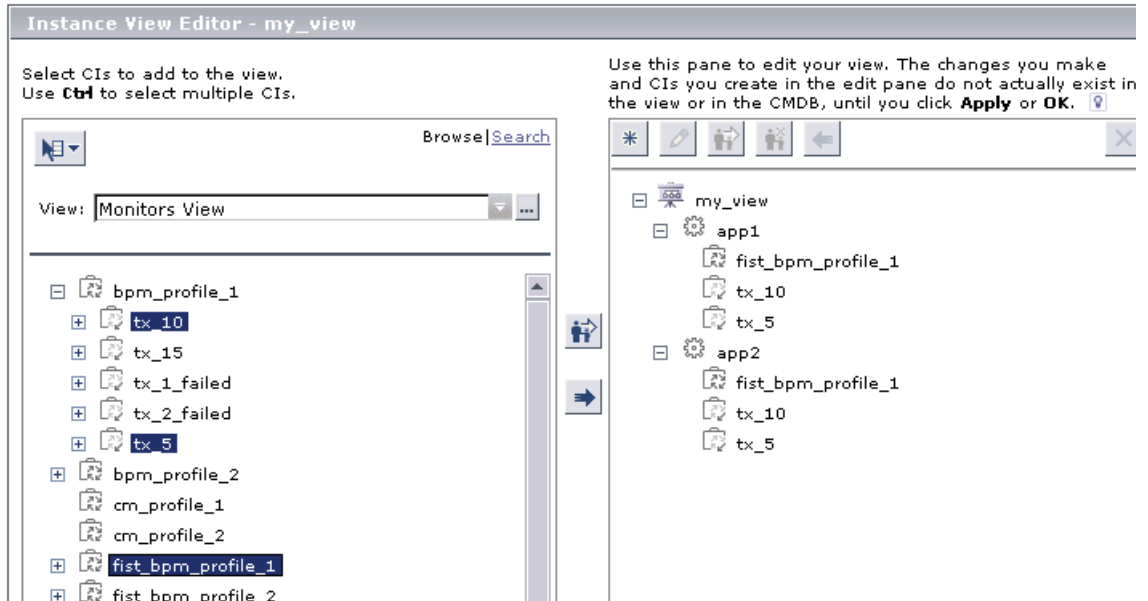
Additional Functionality in the Instance View Editor

The following functionality is available from the edit pane of the Instance View Editor:

➤ Create a new CI in your view and in the CMDB, as described in "Creating CIs" on page 42.

➤ Edit the properties for a CI, as described in "Editing CIs" on page 43.

➤ Change the display/hide all child CIs status for a CI, as described in "Changing the Display/Hide Child CIs Status" on page 43.

➤ Disconnect a CI from the parent CI, so removing the relationship between the two CIs in the IT universe, as described in "Deleting the Relationship to a Parent CI" on page 44.

➤ Delete a CI from the CMDB, as described in "Deleting CIs from the CMDB" on page 45.

Note: None of the above actions take effect in the CMDB until you click **OK** or **Apply** in the Instance View Editor.

Creating CIs

You can create new CIs from the Instance View Editor, at the root level of your view without a relationship, or related to one of the CIs in your view.

To create a new CI in the Instance View Editor:

1 In the edit pane of the Instance View Editor, decide under which CI you want to add the new CI, then do one of the following:

➤ Right-click the CI and select **Create New CI** from the displayed menu.

➤ Select the CI and click the **Create New CI** button.

The New CI Wizard opens.

2 Define the information for the new CI. For details, see "Using the New CI Wizard" on page 49.

3 After creating the CI using the New CI Wizard, the CI is added to the edit pane, at the appropriate point in the view. If the new CI was created related to another CI, it is shown in the view as a child of that CI, even if the parent CI is set to exclude all child CIs status. If the parent is set to include all child CIs status, then the new CI is added with an orange icon. (For more information, see "Editing with the Instance View Editor" on page 28.)

4 When you click **Apply** or **OK**, the changes are updated to the CMDB. The new CI is not physically created in the CMDB and added to the view until you save your changes in the Instance View Editor.

Editing CIs

You can edit some of the properties for a CI from edit pane of the Instance View Editor.

Note: From the left pane of the Instance View Editor you can view CI properties in read-only format.

To edit CI properties in the Instance View Editor:

1 In the edit pane, do one of the following:

➤ Right-click the required CI and select **Edit CI** from the displayed menu.

➤ Select the CI and click the **Edit CI** button.

The Edit CI Properties dialog box opens.

2 Edit the properties as required. For details, see "Configuration Item Properties" on page 71.

3 When you click **Apply** or **OK**, the property changes for the CI are updated to the CMDB.

Changing the Display/Hide Child CIs Status

You can toggle the status of some CIs in your view between displaying all child CIs and hiding the child CIs.

Toggling the status of the parent changes the child CI rule that it is using. For more information on the child CI rules, see "Child CI Rule Functionality" on page 32.

➤ If you select **Hide All CMDB Children**, in order to change the status of the parent CI from displaying all CMDB child CIs to hiding them, then all child CIs in the subtree under the parent CI are hidden, whether they were added manually or recursively.

If the subtree includes a CI created during the current Instance View Editor session and not yet saved, then that CI is removed completely and will never be created in the CMDB.

➤ If you select **Display All CMDB Children**, in order to change the status of the parent CI from displaying only manually added CIs to displaying all CMDB child CIs, then all child CIs that exist in the CMDB under the parent CI are added to the view. (You may have to click **Apply** to see the changes.)

All the child CIs are shown with faded icons and in italicized text, including any that were manually added to the parent before its status was changed.

To change the include/exclude child CIs status for a CI:

1 In the edit pane of the Instance View Editor, use one of the following methods to change the status:

➤ Right-click the required CI and select **Display All CMDB Children** or **Hide All CMDB Children**, as relevant, from the displayed menu.

➤ Select the CI and click the select **Display All CMDB Children** button or **Hide All CMDB Children** button, as relevant.

Note: One or both of these menu options and buttons are disabled when the status is not currently applicable for the CI.

---

2 When you click **Apply** or **OK**, the changes are updated to the CMDB.

Deleting the Relationship to a Parent CI

You can remove the relationship between a CI and its parent CI (or remove the relationship of a root CI to the view) by using the **Delete Relationship to Parent CI** option. Using this option disconnects the child CI from the parent and removes the CI from the view (unless it appears elsewhere in the view). When you save your changes in the Instance View Editor, the relationship is deleted from the CMDB, so the CI will no longer appear as a child of the parent CI in any view.

Any child CIs in the subtree under the disconnected CI are also removed from the view, (without affecting their relationship with the disconnected CI). If there are other instances of the disconnected CI (or any of its children) anywhere else in the view, these other instances are not affected, they remain in the view.

If you create a CI during the current Instance View Editor session, but then use the **Delete Relationship to Parent CI** option to remove it before saving it to the CMDB, the new CI is never created in the CMDB.

To delete the relationship between a CI and its parent CI:

1 In the edit pane of the Instance View Editor, do one of the following:

➤ Right-click a single CI and select **Delete Relationship to Parent CI** from the displayed menu.

➤ Select a single CI, or multiple CIs (using the CTRL key), and click the **Delete Relationship to Parent CI** button.

The relevant CIs are removed from the subtree.

2 When you click **Apply** or **OK**, the changes are updated to the CMDB.

Deleting CIs from the CMDB

From the Instance View Editor, you can delete single or multiple CIs from the CMDB, so that the CIs are deleted from all views. When you delete a CI, the subtree under the CI is removed from the current view and all other views (but the CIs in the subtree are not deleted from the CMDB).

To delete a CI from the CMDB:

1 In the edit pane of the Instance View Editor, use one of the following methods to delete a CI:

➤ Right-click a single CI and select **Delete CI from the CMDB** from the displayed menu.

➤ Select a single CI, or multiple CIs (using the CTRL key), and click the **Delete CI from the CMDB** button.

The relevant CIs are removed from the subtree.

2 When you click **Apply** or **OK**, the changes are updated to the CMDB.

Creating a New CI

In IT Universe Manager you can manually create new CIs. All new CIs are added to the CMDB; they also appear in the currently open view, if:

➤ it is an instance view.

➤ it is a pattern view, and the CI, together with the relationship defined for the CI, are appropriate for the pattern view definition.

This section contains the following topics:

➤ "Creating a Non-Related CI" on page 46 discusses the creation of a new CI without any current relationship to another CI.

➤ "Creating a Related CI" on page 47 discusses the creation of a new CI together with its relationship to an existing CI.

➤ "Using the New CI Wizard" on page 49 describes the procedure for creating a new CI (related or non-related).

Creating a Non-Related CI

When you build a new subtree within your view, you generally start by creating a root CI representing a high-level logical element in your business. You can add a root CI when working at the top level of the view (when you first access the view, or when the view name is selected), by clicking the **Create New CIs** link in the welcome page, or by selecting **Create New CI** in the Instance View Editor.

The **Create New CIs** link opens the New CI Wizard, where you define the properties for the new CI, as described in "Using the New CI Wizard" on page 49. The wizard creates the CI in the CMDB and adds the created CI as a root CI in your view (if you are working in a pattern view, the CI is added to the view only if it meets the pattern specifications).

A CI created in this way is added as an independent entity without any current relationship to any other CI existing in the CMDB. After creating the first CI, you can then create or add related CIs below the first CI.

For example, if you are building an instance view to show your lines of business, you may add root CIs representing different LOBs within your organizational structure.

Each Line of Business CI in the example is an independent entity, with no relations, ready to become the root CI for a subtree. You then attach CIs to build the subtree.

To build the branches of the subtree, you can attach new CIs that you create, or you can attach CIs that already exist in the CMDB. Each CI you attach is immediately related to the first CI:

➤ For details on creating new CIs under the CIs in the view, see "Creating a Related CI" on page 47.

➤ For details on attaching existing CIs, see "Attaching Existing CIs" on page 55.

Creating a Related CI

You can create a new CI to be added to your view as a child of an existing CI, using the New CI Wizard. Both the new CI and the relationship for it are created in the CMDB.

You first select the CI in your view to which you want to attach the new CI, then open the New CI Wizard. In the New CI Wizard you define the properties and relationship for the CI, as described in "Using the New CI Wizard" on page 49. The wizard creates the CI in the CMDB, and adds the new CI as a related CI in your view (if you are working in a pattern view, the CI is added to the view only if it meets the pattern specifications).

After creating the new child CI, it will also appear for other instances of the parent CI, wherever the subtree is set to use the include all child CIs rule. For information on the relationships between CIs, see "Working with Relationships" on page 63.

For example, if you are building an instance view to show your lines of business, then under your Line of Business CIs at the root of your subtrees you may want to add additional logical CIs representing different locations dealing with that business line within your organizational structure. Each location is added as a child of the Line of Business CI, so that the Line of Business CI depends on the Location CI.
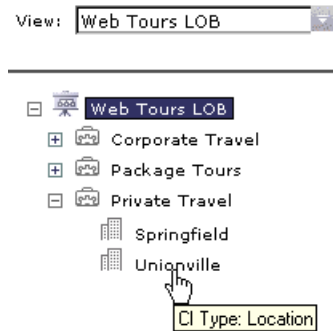
You can continue to build the branches of each LOB subtree by creating more related CIs to attach under the locations, and/or by attaching CIs or whole CI subtrees that already exist in the CMDB. For details on attaching existing CIs and CI subtrees, see "Attaching Existing CIs" on page 55.

Using the New CI Wizard

The New CI Wizard is used to define both non-related CIs (described in "Creating a Non-Related CI" on page 46) and related CIs (described in "Creating a Related CI" on page 47). The wizard content is dynamically updated during the definition process, according to your selections.

Some of the wizard pages contain a **Show Description** button in the top-right corner of the page. Click this button to display additional information for the CI properties, for example, a description of the property, or an alternative name used for the property. Click the button again to hide the descriptions.

To create a new CI:

 1 Open the New CI Wizard in one of the following ways:

  ➤ If you are working in the Instance View Editor, in the edit pane select the view name or a CI and click the **Create New CI** button. (For details on accessing Instance View Editor, see "Accessing the Instance View Editor" on page 28.)

➤ If you are working in the **IT Universe Manager** tab and you want to create a CI that is not related to an existing CI, select the view name in the view tree in View Explorer, then click **Create New CIs** in the right pane.
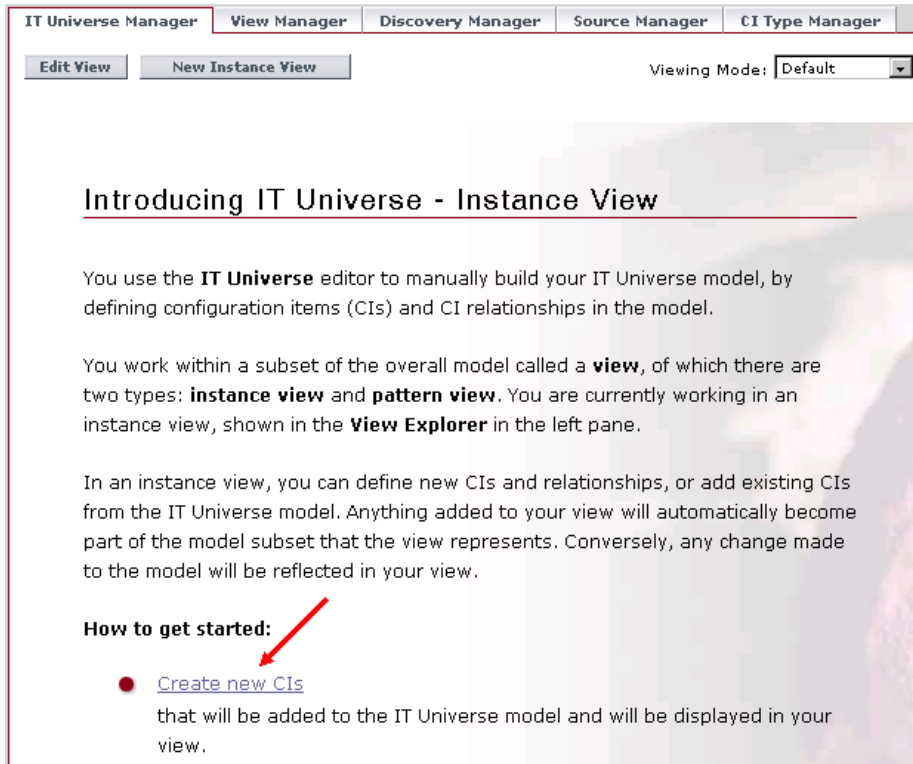
➤ If you are working in the **IT Universe Manager** tab and want to create a CI that will be the parent or child of an existing CI, select the existing CI in the view tree and click **New Related CI**.

Tip: You can also select the required option from the right-click menu for a CI.

The **New CI Wizard** opens at the Define General Properties page, where you select the CIT category for the new CI, and define the properties associated with every CI, such as name, geographical location, and appropriate context menu.

2 Select the CI type for the new CI. The **CI Type** property is context sensitive, permitting you to select only the CITs that are appropriate for the CI or view under which you are creating the new CI. You can select the CIT:

➤ from the **CI Type** dropdown list, which contains an abbreviated list of the most commonly used CITs, as well as CITs that you have previously used (as appropriate for the context).

➤ from the Select Configuration Item Type dialog box, opened by clicking the button beside the **CI Type** box. The Select Configuration Item Type dialog box contains a list of all available CITs that are appropriate for the context. You can toggle the display between **CI Types Tree** (which organizes the CITs by category) and **CI Types Table** (which displays an alphabetical list of all available CITs).

For pattern views, you have the option of toggling between the **Display only CITs that match the current view** list, and the **Display all possible CITs** list, where you can view all available CITs (not only the CITs that match the context). Select the appropriate radio button for the list you require. When you define a CI that does not match the context, it is created in the CMDB, but not added to the current view.

For more information on the available CITs, see "Configuration Item Types" on page 14.

3 Define the other general properties for the CI. Mandatory properties are marked with an asterisk. For information on the properties, see "General Properties for CIs" on page 72.

Note the following:

➤ Some CIs, for example **Group**, have a default context menu defined automatically. You can change the context menu, or define new ones, by clicking **Edit Menu List**. For details, see "Editing the Menu List" on page 75.

➤ CIs added to instance views generally include the **Include Related CIs** option:

　➤ If the **Include Related CIs** check box is selected, the CI is set to recursively display in the view all child CIs from the CMDB, and all CIs in the subtree for the CI are also displayed in the view.

　➤ If the check box is not selected, the CI is set to display only child CIs that are manually added within the view.

The **Include Related CIs** option may be non-editable. For more information on this option, see "Include Related CIs" on page 77.

Proceed as follows:

➤ If you are creating a non-related CI that does not require CIT-specific properties, then the **Finish** button is enabled. Click it and skip to step 6.

➤ If you are creating a CI that requires CIT-specific properties, the **Next** button is enabled. Click the button and continue with the next step.

➤ If you are creating a related CI that does not require CIT-specific properties, then the **Next** and **Finish** buttons are enabled. Continue according to the option you require:

　➤ Click **Next** to continue to the Define CI Relationship page, and skip to step 5.

　➤ Click **Finish** if you do not want to define relationship properties. In this case, IT Universe Manager applies the default relationship properties for the CI you are creating. Skip to step 6.

4 Some CITs require the definition of properties specific to that CIT category, in which case the wizard displays the Define CIT-Specific Properties page. Enter the required information in the property fields. Mandatory properties are marked with an asterisk. For information, see "CIT-Specific Properties" on page 79.

Click **Next** to define relationship properties, or click **Finish** and skip to step 6.

5 If you are defining a related CI, then the wizard displays the Define Configuration Item Relationship page. This page enables you to define the relationship between the new CI and the selected CI in the view, and the properties for the relationship. For more information on relationships, see "Working with Relationships" on page 63.

The **Relationship Definition** area of the page contains the following properties:

> ➤ **CI 1.** Displays the name of the CI on the first side of the relationship. This is the CI that you selected in the view.

> ➤ **CI 2.** Displays the name of the CI on the second side of the relationship. This is then new CI that you are creating.

> ➤ **Relationship Type.** Enables you to define the type of relationship that will be created between the two CIs. For most relationships, only the **Depends On** option is available. Select an option from the list.

The **Relationship Properties** area of the page contains properties that are applied to the relationship, such as the **Must** and **Weight** attributes. For information on the properties, see "Relationship Property Descriptions" on page 66.

Enter the required information and click **Finish**.

6 The new CI and relationship are created in the CMDB and the created CI is added at the specified point in the view tree, if the created CI is appropriate for the current view. If you are working in a pattern view, and the CI or relationship are not appropriate for the TQL definition for the view, then the Summary page is displayed, stating that the CI was not added to the view.

Tip: If the CI does not appear in the view, try refreshing the view using the **Refresh** button in View Explorer.

Attaching Existing CIs

In the IT Universe Manager tab, you can manually attach existing CIs in the CMDB to another existing CI that appears in the current view. When you attach a CI to another CI, you create a new relationship between them in the CMDB and in the view (where applicable). For information on the relationships between CIs, see "Working with Relationships" on page 63.

The IT Universe Manager tab provides two options for creating relationships between existing CIs:

➤ **Attach Related CIs.** This option enables you to create a relationship between the selected CI and one or more CIs in any view.

➤ **Attach Monitor CIs.** This option provides a shortcut for you to create relationships between the selected CI and one or more CIs in the Monitors View. This is useful when adding the monitor CIs at the bottom of your hierarchies.

Note: In addition, you have the option of adding any existing CIs to your view (automatically creating relationships) using the Instance View Editor. For more information, see "Adding Existing CIs to the View Using the Instance View Editor" on page 34.

You attach existing CIs using the Attach Related CIs wizard. The wizard has two main stages:

➤ Selecting the CIs that you want to attach.

➤ Defining the relationship between the original CI and the CIs to attach. (The information required at this stage is the same as in the Relationship Properties dialog box that opens when you draw a relationship line between two CIs in a view, when working in topology map viewing mode.) For information on relationships, see "Working with Relationships" on page 63.

When selecting the CIs to attach, you can select multiple CIs from a view to attach to the original CI. However, all CIs that you want to attach in one operation must have the same relationship types. For example, you cannot attach a CI that uses the relationship **Depends On** in the same operation as a CI that uses the relationship **Monitored By**.

When attaching CIs to a parent CI in an instance view, you can choose to add the CIs set to display all CMDB children, by using the **Add CIs and Their Children** button. For more information, see "Include Related CIs" on page 77.

When working in pattern views, the **Add CIs and Their Children** button is hidden. Both the added CI and its child CIs appear in the view only if the CI(s) and the relationship(s) conform to the requirements of the TQL for the view. For more information, see "Working in Pattern Views" on page 9.

Note: When copying a CI from a discovery process pattern view to an instance view, if you choose to display the child CIs, be aware that you may be adding hundreds of hidden child CIs that do not appear in the pattern view.

To attach existing CIs:

1 In the **IT Universe Manager** tab, access the required view in View Explorer.

2 In the view tree, select the CI for which you want to create a relationship.

3 In the right pane (**Related Configuration Items** tab), click **Attach Related CIs** to attach any CI, or **Attach Monitor CIs** if you want to attach a monitoring CI.

Tip: You can also select the required option from the right-click menu for the CI in View Explorer.

The Attach Related CIs wizard opens at the Select Related CIs page, which contains two panes: The left pane enables you to select the view from which you want to copy CIs; the right pane displays the CIs that you add to the view.

The left pane of the page uses View Explorer functionality. For information on the search capabilities of View Explorer, see "Using View Explorer" in *Working with the CMDB*.

 4 If required, select a different view from the **View** list in the left pane of the Select Related CIs page.

 5 In the view tree, select the check boxes for the CI(s) that you want to attach.

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

---

 6 Click one of the arrow buttons between the panes to move the selected CIs to the list in the right pane:

 ➤ Click the **Add CIs** (plain arrow) button if you do not want the selected CIs to be added with their child CIs. When you add a CI to the view in this way, it is set to display only children that are manually attached to it within the view—all other children are hidden.

 ➤ Click the **Add CIs and Their Children** (recursive arrow) button if you want the added CIs set to display all CMDB children. This means that all child CIs are recursively displayed in the view and the CI subtree is dynamically updated when child CIs are added or deleted, even if the change was made in a different view.

If you are adding CIs under a CI that is already set to display all CMDB children, then the child CIs are displayed, regardless of which child rule the CIs in the subtree are using.

The CIs are added to the list in the right pane, with the **Include Child CIs** column selected as relevant.
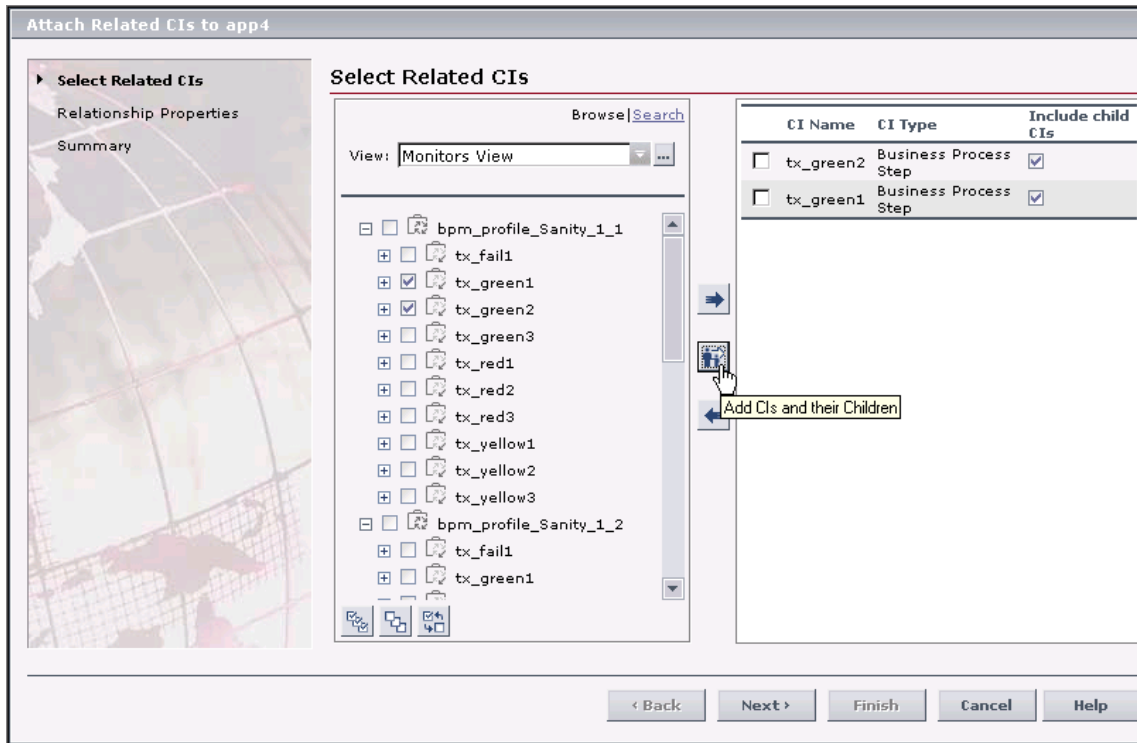
If you want to add both CIs set to display child CIs, and CIs set to hide child CIs, do this in two separate operations, each time copying them to the view using the appropriate button.

7 After adding the CIs, they are listed in the right pane. If you wish to remove one or more CIs from this list, select the check boxes for those CIs in the right pane and click the **Remove CIs** arrow button (this does not delete the CIs from the CMDB).
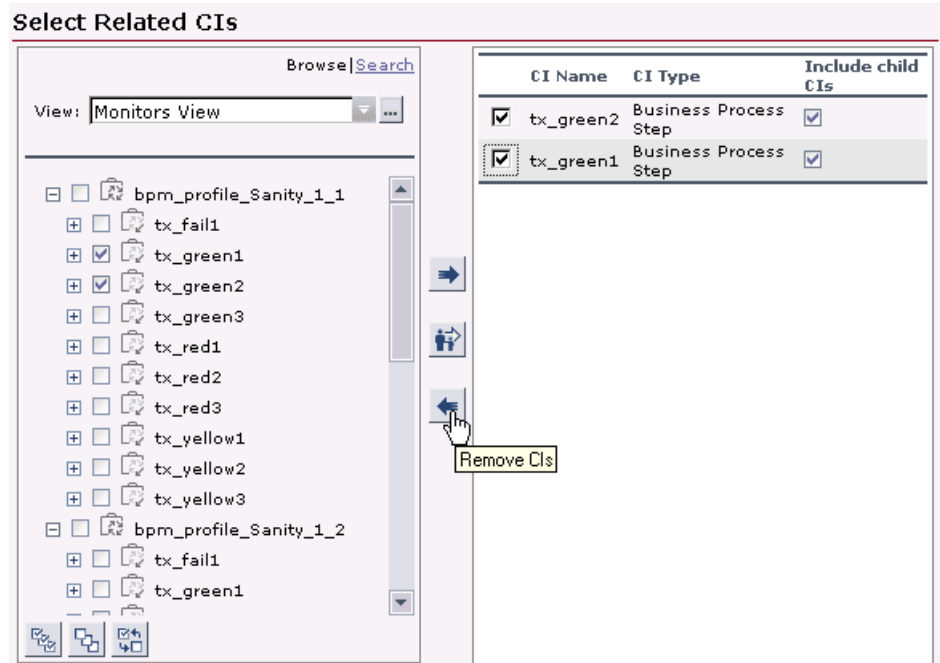
The selected CIs are removed from the list.

8 If you want to copy additional CIs from another view, select the view in the left pane and repeat the instructions in steps 5 and 6. All copied CIs should require the same relationship with the original CI.

9 When you are satisfied with the list in the right pane, click **Next**. If the selected CIs have the same relationship type, the wizard moves to the Relationship Properties page. Continue with step 10.

If there is a compatibility problem with the relationship types for the different CIs that you selected, then an error message is displayed saying that no valid relationships were found. In this case, select the CI(s) that you think are causing the problem from the list in the right pane, and remove them using the **Remove CIs** arrow button. Click **Next**.

10 In the Relationship Properties page, check the details in the **Relationship Definition** area and modify as required:

➤ **CI 1.** Displays the name of the CI on the first side of the relationship. This is the CI that you selected in the view.

➤ **CI 2.** Displays the name(s) of the CI(s) on the second side of the relationship, that will be attached to the original CI.

If you want to modify the list of CIs to attach, use the **Back** button to return to the previous page, and edit your selection.

➤ **Relationship Type.** Enables you to define the type of relationship created between the original CI and the CIs being attached. Select an option from the list. For most relationships, only the **Depends On** option or the **Monitored By** option is available, depending on the type of CIs for which you are creating a relationship.

➤ **Direction.** Enables you to define the direction of the relationship, by selecting the appropriate radio button. For example, if [**CI 1**] **Depends On** [**CI 2**] (generally the default option), then the new CIs will be created as child CIs of the originally selected CI. If [**CI 2**] **Depends On** [**CI 1**], then the new CI will be created as a parent to the originally selected CI.

For some relationship types (for example, **Container link**, or between a group CI and a monitor CI), only one direction is enabled.

11 In the **Relationship Properties** area, define the properties that you want to apply for these relationships, such as the **Must** and **Weight** attributes. For information on these properties, see "Relationship Property Descriptions" on page 66.

Note: When defining multiple relationships, if you need to define different descriptions and property values for each relationship, then skip the property definitions here and instead edit each relationship individually after it's created. For details, see "Editing Relationship Properties" on page 65.

---

12 Click **Finish**. The Summary page is displayed, summarizing the new relationships that are being created. If you are working in a pattern view, and the CIs or relationships are not appropriate, then the page tells you that the CIs are created in the CMDB but not added to the view.

13 Click **OK**. Where appropriate, the CIs are added to the view according to the defined relationships.

Deleting CIs

When you delete a CI, you are also deleting all relationships between that CI and its parent CIs. The CI is deleted from the CMDB and from all views where it appears.

Note: In addition to the procedure below, you can also delete CIs when working in the Instance View Editor. For details, see "Deleting CIs from the CMDB" on page 45.

To delete CIs from the CMDB:

1 In the **IT Universe Manager** tab, right-click the CI in View Explorer that you want to delete.

2 Select **Delete CI** from the displayed menu.

3 Click **OK** in the confirmation dialog box. The CI is removed from the view and from the CMDB.

Notes and Limitations for Deleting CIs

➤ Deleting a CI does not delete its child CIs. If the child CIs do not appear in any other view, then they will remain in the CMDB without appearing in a view. The child CIs may appear in the future in pattern views where they meet the pattern requirements.

One exception to this rule is when the relationship between a child and parent CI is a **Container link**; then, when the parent CI is deleted, the contained CI is also removed from the CMDB and the view. For example, when a host CI is the container for a child CPU CI, then when the host CI is deleted, the CPU is also removed from the CMDB.

➤ If you delete a CI without first deleting the relationship between the CI and any child CIs that were added within the view, then the child CIs are not removed from the view, they move up to the root level. This does not apply when deleting CIs from the view when working in the Instance View Editor.

➤ When you delete a CI that was created by the source adapters or the discovery process, the CI will be restored the next time that the CMDB is updated by that source (if the CI is still valid), and will reappear in all relevant pattern views.

Working with Relationships

This section contains the following topics.

➤ "About Relationships" on page 63

➤ "Editing Relationship Properties" on page 65

➤ "Relationship Property Descriptions" on page 66

➤ "Notes and Limitations for Creating Relationships" on page 67

➤ "Deleting a Relationship" on page 68

About Relationships

Many of the relationships between CIs in the CMDB are created automatically by Mercury Business Availability Center, for example, by the discovery process, the source adapters, and the Dynamic Node Factory. In addition, you can manually create relationships in IT Universe Manager in Monitor Administration.

When working in topology map mode, you also have the option to define a relationship between two CIs in the view by drawing a line between the two. This action directly opens the Relationship Properties dialog box, which is the same as the Relationship Properties page of the Attach Related CIs wizard (described in "Attaching Existing CIs" on page 55).

Attaching one CI to another CI in IT Universe Manager or Monitor Administration creates a hierarchical relationship (parent-child) between them, so that one CI is influenced by the other. Most relationships defined in this way can only be of relationship type **depends on** (if the relationship is between two non-monitoring CIs), or of relationship type **monitored by** (if the child CI in the relationship is a monitor-type CI); however, other relationship types are used between specific types of CIs. For example, if a relationship is defined between two Host CIs, then there are four possible relationship types.
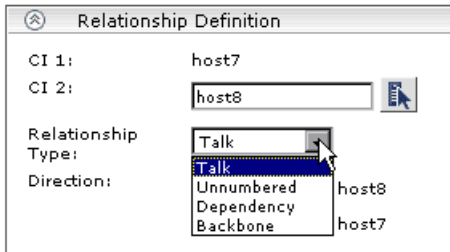
The possible relationship types between two CITs are defined in the CI Type Manager (for details, see *CI Type Manager Administration*).

When you create or attach a child CI to a parent CI in your view, you are also creating that relationship in the CMDB. The child will therefore impact on the parent CI in every view where the relationship exists, and will be displayed in every view where the parent CI is set to display all CMDB children (and for pattern views, where the child and relationship matches the TQL). For more information, see "Include Related CIs" on page 77.

You can create as many relationships as you want between a CI in the CMDB and other CIs. Each new relationship is its own entity in the CMDB, and has its own properties that can be edited in the IT Universe Manager tab (as described in "Editing Relationship Properties" on page 65). You can also create more than one relationship between the same two CIs, when each relationship is based on a different relationship type. In the following example, host7 has both a **Backbone** and a **Talk** relationship with host8.

The following lists the relationships types available in the CIT Manager (and used in TQLs) that the Business Logic Engine recognizes when performing status calculations:

➤ member

➤ applicationLink

➤ monitored_by_for_sap

➤ contained

➤ contains

➤ container_f

➤ depends_on

➤ monitored_by

➤ eum_links

➤ system_monitor_links

➤ monitor_links

Editing Relationship Properties

You can edit some relationship properties (but not the relationship definition).

To edit relationship properties:

1 In the **IT Universe Manager** tab, access the required view in View Explorer and select the parent or child CI in the relationship.

2 Open the **Related Configuration Items** tab in the right pane, and find the required relationship in the table.

3 Click the **Edit relationship** button for the relationship you want to edit. The Relationship Properties dialog box is displayed.

4 Modify the values in the **Edit Relationship Properties** area, as required. For information on these properties, see "Relationship Property Descriptions" on page 66.

5 Click **OK** to save your changes. After modifying relationship properties, the **Update Time** property will be updated with the time that you made the changes.

Relationship Property Descriptions

The following properties are displayed for relationships. Some properties can be defined while creating a relationship, others are defined automatically.

➤ **CI Type.** The type of relationship, for example, **Monitored By** or **Depends On**.

➤ **Create Time.** The date and time when the relationship was created.

➤ **Update Time.** The date and time when the relationship was last updated.

➤ **Description.** A description for the relationship.

➤ **Origin.** The user name of the administrator who manually created the relationship, or an ID for the source that automatically created the relationship.

➤ **Allow CI Update.** When the check box is selected, the discovery process will automatically update the relationship.

➤ **Weight.** When a weight value is entered in the box, the child CI is given more weight in percentage calculations, when percentage rule is used to calculate status for the parent CI. For example, if a child CI is given a weight of **3**, then it will have three times more impact on the parent status calculation than its unweighted siblings.

The weight defined for a relationship only applies for that relationship—it will not be used in any other relationships the child CI may have. For more details, see "Percentage Rule" in *Repositories Administration*.

➤ **Must.** When the check box is selected, the child CI is dominant when the percentage rule is used to calculate status for the parent CI. If the CI has low status, the parent must take the child CI status, if necessary overriding the percentage rule calculation. This is useful to give emphasis to an important CI. For example, you may want to define **Must** for a database CI, so that status will be **critical** all the way up the subtree if the database falls, regardless of the status of the other CIs in the subtree.

The must status defined for a relationship only applies for that relationship—it will not be used in any other relationships the child CI may have. For details, see "Percentage Rule" in *Repositories Administration*.

➤ **Network Address.** For relationships created by the discovery process, defines the discovered IP net address or the net address range.

Notes and Limitations for Creating Relationships

➤ If, when attaching multiple CIs to a CI, you define that the attached CIs will all be parents to the original CI (for example, so that the relationship direction reads that CI X, CI Y, CI Z Depends On CI 1), then a separate subtree will be created in the view for each parent, with the original CI appearing as a child in each subtree.

➤ You cannot create exactly the same relationship twice within a view. For example, if parent CI Italy depends on child CI Rome, Rome will appear under the CI Italy in the view. If you then again attach Rome to Italy with a **depends on** relationship, Rome will still only appear once under that parent CI. This is true even if you attach each instance of the child with different settings, for example, one instance set to display child CIs, and one instance set to not display them.

➤ An instance of a CI at the root level, with no relationships, is automatically removed when another instance of the CI is added to the view, this time as a child CI. If you later remove the relationship for the second instance, the first instance of the CI will reappear as a root CI in the view. (This does not apply when working with the Instance View Editor.)

➤ When you add an existing CI to an instance view, there may already exist other instances of that CI that are hidden in the view, because the parent CI is set to hide its children. After adding the CI to the view, all hidden instances of that CI in the view may then be displayed.

Deleting a Relationship

You can permanently remove the relationship created between two CIs. This will not remove either of the related CIs from the CMDB, but will remove the relationship between the two CIs from the CMDB and all views.

To delete relationships from the CMDB:

1 In the **IT Universe Manager** tab, select the CI in View Explorer for which you want to delete a relationship.

2 In the **Related Configuration Items** tab in the right pane, select one or more check boxes for the relationships that you want to delete.

3 Click **Delete Relationship**.
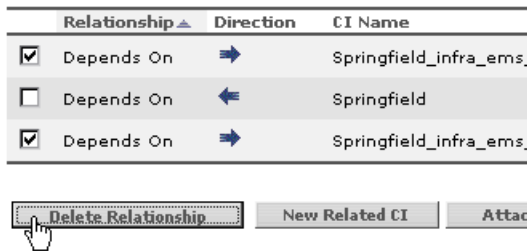
4 Click **OK** in the confirmation dialog box. The relationship(s) are removed from the list in the **Related Configuration Items** tab and from the CMDB.

Notes and Limitations for Deleting a Relationship

➤ The relationship table shows only the relationships for the CI that are part of the current view; however, the same relationships may also be part of other views, so the relationships will also be deleted from those views.

➤ Deleting a relationship does not remove from the view CIs that were added within that view. If, after deleting a relationship, the CI is no longer related to any other CI in the view, then the CI is shown at the top level of the view. To remove the CI completely from the view (without deleting it from the CMDB), use the **Delete Relationship to Parent CI** option in the Instance View Editor (as described in "Deleting the Relationship to a Parent CI" on page 44).

➤ When you delete a relationship that was created by the source adapters or the discovery process, the relationship will be restored the next time that the CMDB is updated by that source (if the relationship is still valid), and will reappear in all relevant pattern views.

➤ It may take a few minutes to delete subtrees, or multiple CIs and relationships, from a view.

4

Configuration Item Properties

Configuration items (CIs) have general properties, common to all CIs, and properties specific to the configuration item type (CIT). Some of these properties can be modified in the IT Universe Manager Properties tab.

This chapter describes:	On page:
Editing CI Properties	71
General Properties for CIs	72
CIT-Specific Properties	79

Editing CI Properties

You can edit different properties for CIs in the IT Universe Manager Properties tab or from the Instance View Editor.

Use the **Show Descriptions** toggle button (top-right corner of the tab) to see more information about the property fields or to see the internal name of the property.

To edit CI properties:

1 Either:

➤ In the **Admin** > **CMDB** > **IT Universe Manager** tab, select the required view and CI in View Explorer, then click the **Properties** tab in the right pane to display the properties for the selected CI.

Or:

➤ In the Instance View Editor for the view, right-click the required CI in the edit pane and select **Properties**. (For details on accessing the Instance View Editor, see "Accessing the Instance View Editor" on page 28.)

2 Edit the appropriate fields. For information on the properties, see "General Properties for CIs" on page 72 and "CIT-Specific Properties" on page 79.

3 Click **Apply** to save the changes. After modifying the CI properties, the **Update Time**, **Updated By**, and **Last Access Time** properties are immediately updated with the time that you made the changes and your user name.

General Properties for CIs

The CI general properties are common to all CIs. Some of the general properties are defined automatically during CI creation; others can only be defined manually in the General Properties area/page.

You can view properties for a CI from any application that uses View Explorer, by right-clicking the CI in View Explorer and selecting **Properties** from the displayed menu. Editing the properties can only be done in the IT Universe Manager Properties tab, as described in "Editing CI Properties" on page 71.

This section includes the following topics:

➤ "General Property Descriptions" on page 73

➤ "Removing Out of Date CIs Using the Aging Mechanism" on page 75

➤ "Editing the Menu List" on page 75

➤ "Include Related CIs" on page 77

General Property Descriptions

The General Properties area includes the following information:

➤ **CI Type.** The CIT for the CI (read-only value). For information about CITs in Mercury Business Availability Center, see *CI Type Manager Administration*.

➤ **Create Time.** The date and time when the CI was created (read-only value).

➤ **Update Time.** The date and time when the CI properties were last updated. This value is updated automatically by Mercury Business Availability Center.

➤ **Created By.** Where applicable, user name of the administrator who manually created the CI (read-only value).

➤ **Last Access Time.** (Only displayed for certain CIs.) The last time that the CI was accessed, meaning that there was an actual update to the CI, or that the CI was discovered by the discovery process but no information was updated. This parameter is used by the aging mechanism—for details, see "Removing Out of Date CIs Using the Aging Mechanism" on page 75.

➤ **Enable Aging.** (Only displayed for certain CIs.) When this option is selected, the aging mechanism is enabled for this CI. It is selected by default for discovery CIs. For details, see "Removing Out of Date CIs Using the Aging Mechanism" on page 75.

➤ **Name.** The name of the CI; it may be different from the **Display Label**.

➤ **Description.** A description of the CI.

➤ **Origin.** An ID for the source that automatically created the CI, or the application where the CI was created (read-only value).

➤ **Updated by.** The user name of the administrator who updated the CI properties. This value is updated automatically by Mercury Business Availability Center.

➤ **Allow CI Update.** When this option is selected, the discovery process can automatically update the CI properties with information that it discovers. If you change the value of a property supplied by the discovery process, the value will be overridden by the discovered value. When this option is cleared, the CI properties are not overridden by the discovery process.

➤ **Display Label.** The name of the CI as it appears in View Explorer (read-only value).

➤ **Country.** The name of the country where this CI is located. Select an option from the list. You must define a geographical location—country, state (if applicable), and city—if you want the CI to appear on the Geographical Map in Dashboard. For more information, see "Configuring the Geographical Map" in *Application Administration*.

➤ **State.** The name of the state - when applicable for the selected country. The list is automatically updated when you select a country. Select an option from the list.

➤ **City.** The name of the city. The list is automatically updated when you select a country (or state, when applicable). Select an option from the list.

Note: When Business Process Monitor is working in Transaction/Location hierarchy structure (as defined in the Business Process Monitoring source adapter on the **CMDB Administration** > **Source Manager** page), then Location CIs are added to the Business Process Monitor hierarchy displayed in the End User Locations view. If you require the same geographical location for all transactions from a location, you can set the geographical location for the Location CI, and the location is automatically propagated to all BPM Transaction from Location CIs in all profiles that fall under the Location CI.

➤ **Context Menu.** The context menu that is attached to the specific CI and displayed in Dashboard. Some of the CIs are automatically assigned a context menu; for example a Group CI is automatically assigned the Group Menu. To modify or add other context menus to the list, click **Edit Menu List**. For details, see "Editing the Menu List" on page 75.

➤ **Include Related CIs.** (Used in instance views only) When this option is selected, all CMDB child CIs for the selected CI are displayed in the current view. For details, see "Include Related CIs" on page 77.

Removing Out of Date CIs Using the Aging Mechanism

If a CI or relationship is not updated or refreshed over a period of time (for example, a discovery CI is not rediscovered by the discovery process, or no data is received for a CI from the source adapters, and no manual update occurs), then an aging mechanism exists to delete the CI from the CMDB.

When the aging mechanism is enabled, the CI is marked for deletion if the CI has not been accessed for a specified period of time (by default, 14 days). The **Last Access Time** parameter (displayed for in the General Properties area, is used to make the calculations. By default, the CI will be deleted 28 days after the last access time. These default values are set per CIT and can be changed in **Admin > CMDB > CI Type Manager.**

The aging mechanism for the CMDB is enabled/disabled from the Infrastructure Settings page: Select **Admin > Platform > Setup and Maintenance > Infrastructure Settings**, choose **Foundations**, select **CMDB**, and locate the **Is Aging Enabled** entry in the **Aging Settings** table. Change the value to **true** or **false**, as required.

When the aging mechanism is enabled for the CMDB in Infrastructure Settings, you can enable/disable aging and automatic deletion for individual CIs by selecting/deselecting the **Enable Aging** box in the CI General Properties (the box is selected by default for discovery process CIs). If the aging mechanism is disabled for the CMDB, the **Enable Aging** box is ignored and no CIs are deleted.

Editing the Menu List

You can define one or more context menus for a CI or modify the menu list. The menus are displayed for the CI in Dashboard.

When defining menus, you should make sure that the selected menus are logical for the CI.

For information on context menu content, see "Context Menu Items Repository" in *Repositories Administration*.

To edit the context menu list for a CI:

1 Open the IT Universe Manager **Properties** tab for the CI and click **Edit Menu List** (to the right of the **Context Menu** box).

2 In the displayed Select Context Menus dialog box, select the required menu(s) in the **Available Menus** area (you can select more than one using CTRL and SHIFT) and click the right arrow to move them to the **Selected Menus** area.
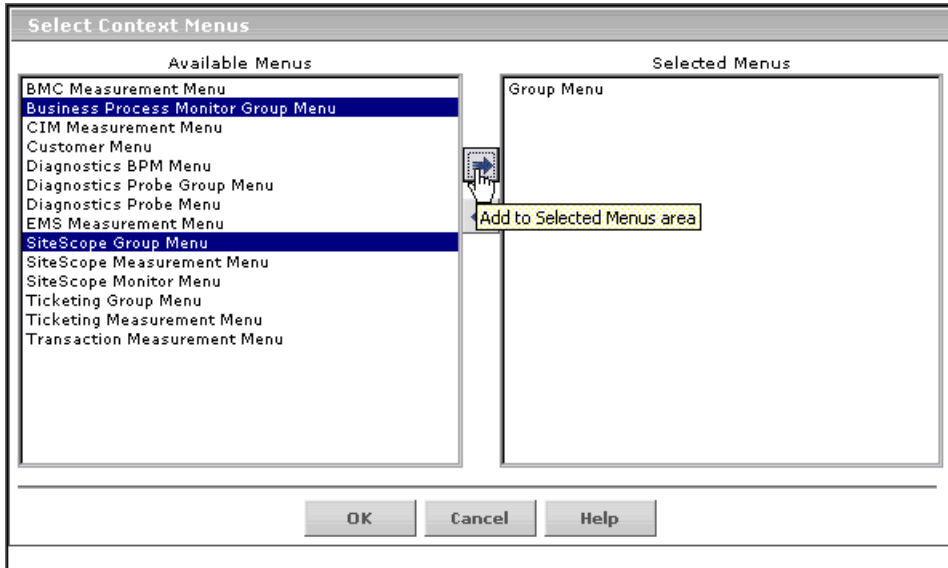
3 To remove menus from the menu list for the CI, select them in the **Selected Menus** box and click the left arrow.

4 Click **OK**. The selected menus are displayed in the **Context Menus** box on the Properties tab.
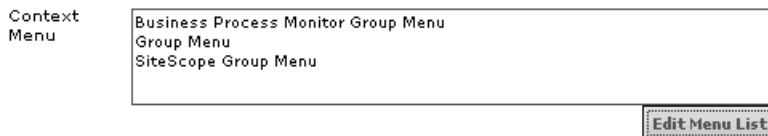
Note: You must click **Apply** in the Properties tab for the menu changes to be saved.

Include Related CIs

In instance views, CIs can be set so that they display all their child CIs from the CMDB (the full subtree), or they can display only the child CIs that are added to them within the view. This is determined by the child CI rule used for the CI. (In a pattern view, all child CIs are always displayed for a CI, as long as they match the TQL.)

The child CI rule assigned to a parent CI determines what child CIs are displayed in the view, as follows:

➤ A parent CI set to **Display all CMDB children** recursively displays in the view all child CIs that exist in the CMDB, and all the childrens' subtrees. The subtrees are dynamically updated with every added child CI, whether it is added within the current view or in a different view.

Once a parent CI is set to display all CMDB children, the child rule used for the CIs on the next branches in the subtree is irrelevant—all child CIs are displayed all the way down the subtree.

➤ A parent CI set to **Display only manually added children** initially has all CMDB children hidden in the view. Future child CIs that are added to that parent CI within the view, by attaching existing CIs or creating new CIs, will be displayed.

The child CIs themselves may be set to display or hide their children.

There are various ways that a CI can be set to display or hide child CIs:

➤ **Creating or editing the CI.** Select the **Include Related CIs** check box in the CI properties to set the child CI rule to display all CMDB children. Deselect the check box to set the child CI rule to display only future manually added children. (This option may be non-editable, depending on the definitions for the parent CI.) For details on editing properties, see "Editing CI Properties" on page 71.

➤ **Adding an existing CI to a view.** Use the **Add CIs and their children** (recursive) button to set the child CI rule to display all CMDB children. Use the **Add CIs** (single arrow) button to set the child CI rule to display only future manually added children. For details, see "Attaching Existing CIs" on page 55.

➤ **Working in the Instance View Editor.** Use the recursive arrow button or select the **Display All CMDB Children** option to set the child CI rule to display all CMDB children. Use the single arrow button or select the **Hide All CMDB Option** to set the child CI rule to display only future manually added children. For details, see "Editing with the Instance View Editor" on page 28.

Whether or not all child CIs are displayed in the view does not influence the status calculations made for the parent CI and the status displayed in Dashboard. The Business Logic Engine calculates status from all child CIs attached to that parent CI, across all views. The calculation is based on all the relationships for the CI in the CMDB, without connection to the child CIs displayed in any particular view.

For example, the following pictures show status in Dashboard for the same CI, Group1, in two views: In the first view, the CI is set to display all child CIs, and in the second, to hide all child CIs. The same status result is shown for the CI in both views, regardless of the child rule in use.

When a CI is set to hide some or all child CIs, then the KPI tooltip in Dashboard shows the number of hidden child CIs with KPIs attached, to indicate that the KPI status of the parent is based on CIs that are not in the view.
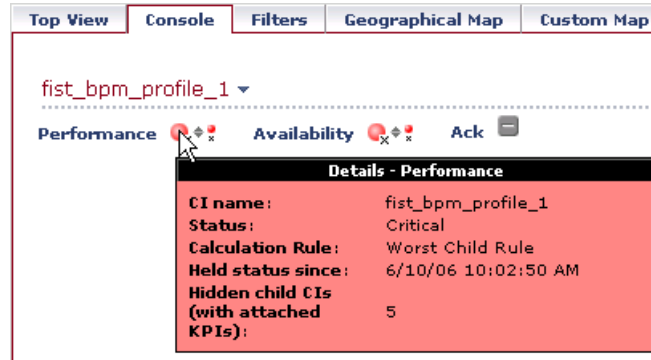
You can use the **Find Visible and Hidden Child CIs** option to view information on hidden children. For details, see "Find Visible and Hidden Child CIs Option" in *Using Dashboard*.

CIT-Specific Properties

The CIT-specific properties are created as part of the CIT definition in the CI Type Manager (for details, see *CI Type Manager Administration*). The values for the CIT-specific properties are taken from the data samples for the source adapter CIs, or discovered during the discovery process for the discovery CIs. The values can also be entered manually, when you manually create CIs to represent entities in your infrastructure.

The CIT-specific properties are displayed in the **Other Properties** area in the Properties tab. Some of the properties for the automatically created CIs can be edited; however, if the CI is set to **Allow CI Update** (in the general properties), then the discovery process may override your changes when it updates the CI.

Use the **Show Descriptions** toggle button (top-right corner of the tab or dialog box) to see more information on the CIT-specific properties.

Note the following regarding CIT-specific properties:

➤ The CIT-specific properties for the Dynamic Node Factory CI are described in Chapter 6, "Working with Dynamic Node Factory."

➤ For CIs generated by the source adapters, the only editable CIT-specific property is the **Selector Expression** for the monitor-type CIs. To edit the selector, click **Edit Selector Expression**. For details on editing selectors, see "Selectors for KPIs" in *Application Administration*.

When the Siebel discovery process is run, the CIs that appear in the Siebel Enterprises view have most of their properties' values automatically entered in the CMDB. Some of the properties related to the location of some packages or logs, must be entered manually in order for data to be inserted properly in the Siebel Enterprises view. For details, see "Changing the Default Timeout for the Execution of a SiteScope Monitor" in *Application Administration*.

After modifying the selector, the current CI will be updated according to the new selector definition, and a new CI will be generated by the source adapters from the incoming samples based on the original selector.

➤ Some of the CIT-specific properties for discovery process CIs are set by default to be monitored for changes. When one of these properties has been modified, information on the change can be viewed in a change report, and in some cases, and icon is displayed in Dashboard. For more information, see "Change Report" in *Using Dashboard*.

5

Working in Topology Map Mode

This chapter introduces the topology map. The topology map enables you to view single level or multilevel maps, which display the results of instance views and pattern views defined in the View Manager.

This chapter describes:	On page:
About the Topology Map	82
Displaying a View in Topology Map Mode	83
Viewing Layers with a Large Number of CIs	83
Understanding Topology Map Concepts	84
Using the Toolbar Options	86
Printing the Contents of the Topology Map	86
Defining a View's Layout	87

About the Topology Map

The topology map enables you to view the results of the views you created in the View Manager. Mercury Business Availability Center allows you to create two types of views: instance views and pattern views. For more information on creating views, see "Creating and Editing Views" on page 18 in *View Manager Administration*.

Instance views are created by defining the individual CIs and relationships that make up the view directly in the view. All the CIs in an instance view are seen on one level.

Pattern views are created from a TQL (Topology Query Language) query that defines the structure of the view. Pattern views allow you to define organization rules, enabling you to display topology maps in multiple layers. For more information on organization rules, see "Adding Folding Rules to Relationships" in *View Manager Administration*.

The topology map displays the results of the currently selected view in the View Explorer, and consists of CIs that are defined in the view and the relationships between them. You can change the way the maps are displayed by selecting one of the view layouts (**Hierarchical**, **Symmetric**, or **Orthogonal**). For details, see "Defining a View's Layout".

In addition, you can right-click a CI in the topology map to perform the following functions:

➤ **Edit CI properties.** For details, see "Editing CI Properties" on page 71.

➤ **Create a related CI.** For details, see "Creating a New CI" on page 46.

➤ **Attach a related CI.** For details, see "Attaching Existing CIs" on page 55.

➤ **Delete a CI.** For details, see "Deleting CIs" on page 62.

Displaying a View in Topology Map Mode

There are two modes in which you can display view information.

➤ **Default**

➤ **Topology mode**

When you first access open Mercury Business Availability Center, the view is automatically displayed in **Default** mode. This section describes how to display the view in **Topology Map** mode.

To display the view in Topology Map mode:

1 Select **Admin** > **CMDB** and then select the **IT Universe Manager** tab.

2 In the View Explorer, select the required view from the **View** list.

3 From the **Viewing Mode** list in the top-right corner of the **IT Universe Manager** tab, select **Topology Map**.

Viewing Layers with a Large Number of CIs

A layer cannot display more than 900 CIs in a topology map view. If the layer contains more than 900 CIs, it cannot be opened. If you try to open a layer that contains more than this amount, a message is displayed. In this case, it is recommended to use organization rules to spread the CIs among several layers. For using organization rules, see "Adding Folding Rules to Relationships" in *View Manager Administration*.

Understanding Topology Map Concepts

This section describes the following topology map features:

➤ "Understanding CI Icon Composition" on page 84

➤ "Selecting Multiple CIs" on page 85

➤ "Simultaneous Selection" on page 85

➤ "Tooltips" on page 85

➤ "CI Labels" on page 86

➤ "Navigating Between Different Map Levels" on page 86

Understanding CI Icon Composition

The icon that represents each CI in the topology map is defined by the administrator in the CI Type Manager. For information on how to assign an icon to a CIT, see "Assigning an Icon to a CIT" in *CI Type Manager Administration*.

CI icons are made up of the following components:

➤ An icon

➤ The severity for example, critical or major, appears in the lower left-hand corner.

If an icon appears with a downward arrow, as shown in the example above, it means that there is an additional CI layer beneath it, created by a parent or child folding rule defined in the View Manager. For information on how to set up organization rules, see "Adding Folding Rules to Relationships" in *View Manager Administration*. For information on how to navigate between layers, see "Navigating Between Different Map Levels" on page 86.

Selecting Multiple CIs

In the topology map, you can select multiple CIs.

To select multiple CIs:

Hold down the CTRL key and click the CIs or drag the mouse diagonally along the CIs as shown in the following figure:
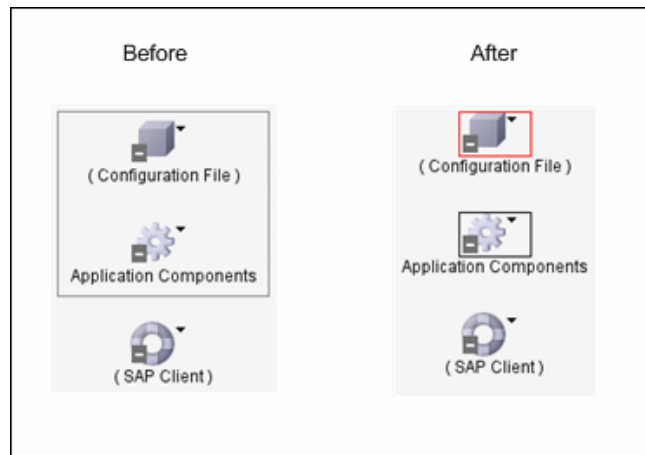
If you select multiple CIs, only one CI has a black selection rectangle around it, while all the other selected CIs have red selection triangles around them (see the **After** picture above). In this situation, when you right-click anywhere on the map to open the context menu, the option you choose is performed only on the CI surrounded by the black rectangle.

Simultaneous Selection

When you select a CI, it simultaneously selects the CI in both the View Explorer and the topology map. For more information on the View Explorer, see "Using View Explorer" in *Working with the CMDB*.

Tooltips

When you point to a CI in the topology map, a tooltip is displayed that contains the CI type of the selected CI as defined in the CI Type Manager.

CI Labels

Each CI on the topology map has a label that appears underneath it. The label definition can be customized to include different attribute values. For information on how to customize the label definition, see "Defining an Attribute for a CIT Label" in *CI Type Manager Administration*.

Navigating Between Different Map Levels

An icon with a shadow indicates that an additional CI layer lies beneath it, created by a parent or child organization rule defined in the View Manager.

For example, if the IP CI is defined as a child of the host CI in the View Manager, you can double-click any host icon in the topology map to display a map of IPs connected with that host.

To drill down to the lower level(s):

Double-click an icon with a shadow.

To move up one level in the topology map:

 1 Select a CI that has an additional CI layer above it.

 2 Click the **Up layer** button to move up to the next level.

The **Up layer** button is enabled only if there is an additional CI layer above it, created by a parent or child organization rule defined in the View Manager. For information on how to set up organization rules, see "Adding Folding Rules to Relationships" on page 94 in *View Manager Administration*.

Using the Toolbar Options

For a description of each toolbar option, see "Printing the Topology Map" 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" 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*.

6

Working with Dynamic Node Factory

This chapter describes the concept of the Dynamic Node Factory (DNode) in Mercury Business Availability Center. It also describes how to define and edit a Dynamic Node Factory configuration item (CI), whether this is done from the IT Universe Manager tab or in the source adapters in the Source Manager tab.

What Is Dynamic Node Factory?

The Business Logic Engine is able to dynamically create CI hierarchies in the Mercury Universal CMDB in run time, based on incoming data samples on the Mercury Business Availability Center Bus. These hierarchies are created using the **Dynamic Node Factory** CI.

The Dynamic Node Factory CI includes a template that defines the rules for creating and managing the generated hierarchy, and a selector that defines which data samples are relevant.

When a Dynamic Node Factory CI is added to a service view, it can immediately start generating CIs from the incoming data samples. The generated CIs are added to the IT universe model in the CMDB, and to the subtree under the Dynamic Node Factory CI in the view. These generated CIs are then available to add to other views like any other CI in the CMDB.

A Dynamic Node Factory CI is added to a view and the IT universe model in one of the following ways:

➤ **Automatically, by a source adapter.** Some of the Mercury Business Availability Center custom source adapters, for example, HP OpenView, include a Dynamic Node Factory definition. When the source is added in Source Manager, the Dynamic Node Factory CI is added to the out-of-the-box view for HP OpenView.

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

➤ **Manually, in IT Universe Manager.** You can create a new CI of type Dynamic Node Factory when building your view in the IT Universe Manager tab.

After a Dynamic Node Factory CI is added to a view, the CI properties can be edited in the IT Universe Manager tab. For details on creating a Dynamic Node Factory CI or editing its properties, see "Defining/Editing Properties for a Dynamic Node Factory CI" on page 94.

KPI status for the CIs in the Dynamic Node Factory hierarchy can be viewed in the Dashboard application as soon as the CIs are created.
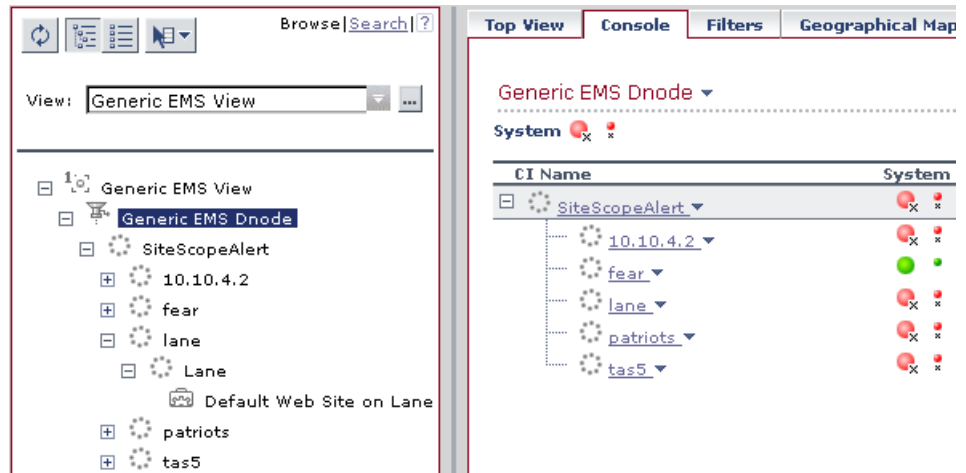
How Dynamic Node Factory Works

The XML definition file generated by the Dynamic Node Factory CI defines rules for listening to the samples that arrive to the system, using selectors to define areas of interest. The XML file guides the Business Logic Engine on how to build the topology in the CMDB based on the incoming data, and produces its own definitions for the generated CIs.

Dynamic Node Factory CIs use the **DNode definition key** property, which defines how the samples are processed by providing example values for the XML file. These values are used as the template for the dynamic hierarchy in the CMDB. When required, you can customize the DNode definition key to produce a hierarchy that displays the information you need. For details, see "DNode Definition Key Property" on page 100.

If there is no existing CI for a received sample in the CMDB, then the Business Logic Engine creates the CI and adds it to the subtree for the Dynamic Node Factory CI. If the CI already exists in the CMDB (for example, if a BPM Transaction CI that corresponds to the sample was already created by the Business Process Monitoring source adapter), then the existing CI is added to the Dynamic Node Factory subtree, creating a new relationship for that CI.

Once a CI is created, the CI definitions are stored as for any other CI, and it receives update samples from the Dashboard matcher based on the selector for the Dynamic Node Factory CI. The Dynamic Node Factory hierarchy is rebuilt only as necessitated by the sample contents. (You can also define additional composite selectors for the Dynamic Node Factory XML definitions, so that the generated CI is updated independently from the Dynamic Node Factory generators. For details, see "Defining Composite Selectors in the DNode Definition Key" on page 109.)

You can add the generated CIs to other views, modify the CI properties, and delete the generated CIs from the CMDB. (However, after deleting a CI, it may be recreated by the Dynamic Node Factory based on incoming data samples.)

When to Use Dynamic Node Factory

Dynamic Node Factory hierarchies are useful for the integration and mapping of samples from external systems (generally EMS systems), where the database structure for the source is unknown, and Mercury Business Availability Center does not have a predefined source adapter for handling the data. However, the decision of whether to set up a Dynamic Node Factory hierarchy or a new source adapter to import the information depends on the particular situation and your data requirements.

The Dynamic Node Factory CI may also be used to bring alternative hierarchies from adapter sources.

You should take into consideration the following points:

➤ The adapter provides more precise results.

➤ Implementation of a Dynamic Node Factory template to import the information is usually simpler than setting up a new adapter for the source.

➤ The Dynamic Node Factory is dynamic, reflecting changes that generate new CIs in run time, while the adapters reflect the changes only after synchronization. For example, if you create a Dynamic Node Factory CI for alerts, you can generate a hierarchy that displays alerts with event children, without knowing in advance what type of alert will be received. The Dynamic Node Factory hierarchy will display a new alert when it occurs, while an adapter for alerts will display a new alert only after synchronization occurs (by default, once an hour).

However, if an alert is removed at the source, the adapter will reflect this change at synchronization, while the Dynamic Node Factory may take up to 24 hours to reflect the change (depending on the Dynamic Node Factory cleaning definitions). Similarly, if a transaction name is changed or moved to a new profile in Monitor Administration, then the Dynamic Node Factory will create a new CI for it, but the old CI will remain in the hierarchy until it has not had any corresponding samples for 24 hours (default value, can be changed to a different value).

Defining/Editing Properties for a Dynamic Node Factory CI

You can edit the properties for a predefined Dynamic Node Factory CI in your view, or define a new Dynamic Node Factory CI to add to your view, in order to generate a hierarchy that fits your organization's needs. You can modify the mapping to the samples that create the hierarchy, and the rules and logic used to import the data and present it in Dashboard.

(It is also possible to customize the template for a predefined Dynamic Node Factory CI in the custom source adapter, but it is recommended that you customize the CI in the view, rather that in the adapter.)

Keep in mind that you affect two levels of CIs when you modify properties for a Dynamic Node Factory CI:

➤ The general properties for the Dynamic Node Factory CI, for example, location and context menu, apply only to that CI. Also, when you attach KPIs to the Dynamic Node Factory CI in Dashboard Administration, they are defined only for that CI. Modifying these properties has no influence on the generated hierarchy.

➤ The CIT-specific properties for the Dynamic Node Factory CI, for example, selector expression and KPI status for removal, define information relating to the CIs in the generated hierarchy under the Dynamic Node Factory CI. Modifying these properties changes the templates and handling rules for the hierarchy.

Changes to these properties should be correlated with the definitions within the DNode definition key. For example, the selector expression must identify samples that contain the keys defined in the generators, otherwise the Dynamic Node Factory hierarchy will be empty.

Note: If you change the DNode definition key or selector expression for an active (enabled) Dynamic Node Factory CI, then the previous Dynamic Node Factory hierarchy is deleted from the view and the hierarchy is recreated.

If the catchment area for a Dynamic Node Factory hierarchy is too wide, you can modify the selector expression for the Dynamic Node Factory CI to narrow down the definition, for example, to include only samples coming from a particular machine.

The following procedure describes how to define properties for a Dynamic Node Factory CI in the IT Universe Manager tab, whether you are creating or editing the CI.

If you require additional help with creating a Dynamic Node Factory hierarchy, contact Mercury Customer Support.

Note: There are no default KPIs defined for the Dynamic Node Factory CI; you must manually attach KPIs after creating the CI, or no status will be displayed in Dashboard.

To define/modify a Dynamic Node Factory CI:

1 To create a new Dynamic Node Factory CI: In the **IT Universe Manager** tab, open the New CI Wizard as described in "Creating a New CI" on page 46, and select **Dynamic Node Factory** in the Select Configuration Item Type page.

To edit properties for an existing Dynamic Node Factory CI: In the **IT Universe Manager** tab, select the CI in View Explorer and open the **Properties** tab.

2 Define properties in the Define General Properties page (or edit in the **General Properties** area), as required. These properties apply only to the Dynamic Node Factory CI, not to the CIs that will be created in the hierarchy. For details on these properties, see "General Properties for CIs" on page 72.

3 Define properties in the Define CIT-Specific Properties page (or edit in the **Other Properties** area), as required:

Define CIT-Specific Properties [i]

☑ Enable DNode

Max number of CIs to create   `1000`

DNode definition key

```
<?xml version="1.0" encoding="UTF-8"?><dnode><Generator
groupBy="szSessionName" itemTemplate="profile"
linkTemplate="depends_on"><Generator groupBy="szTransactionName"
itemTemplate="transaction" linkTemplate="depends_on"><Generator
groupBy="szLocationName" itemTemplate="tx_location"
```

Selector expression

[Edit Selector Expression]

☑ Create CI if groupBy value missing

KPI status for removal   `Nodata`

Time period for removal status   `24h`

⊛   Other Properties

DNode ID*   149896

☑ Enable DNode

Max number of CIs to create   `1000`

DNode definition key

```
<?xml version="1.0" encoding="UTF-8"?><dnode><Generator
groupBy="szSessionName" itemTemplate="profile"
linkTemplate="depends_on"><Generator groupBy="szTransactionName"
itemTemplate="transaction" linkTemplate="depends_on"><Generator
groupBy="szLocationName" itemTemplate="tx_location"
```

Selector expression   `((sampleType=="trans_t"))`

[Edit Selector Expression]

☑ Create CI if groupBy value missing

KPI status for removal   `Nodata`

Time period for removal status   `24h`

➤ **Enable DNode.** (Selected by default.) When the check box is selected, the Dynamic Node Factory hierarchy is updated as required based on the incoming samples. If the check box is deselected, the hierarchy is not updated (but the KPIs for the CIs will still receive status updates in Dashboard).

➤ **Max number of CIs to create.** Defines the limit for creating CIs in the Dynamic Node Factory hierarchy, default is**1000**. When a sample is received and the number of CIs in the hierarchy is below 1000, then the Dynamic Node Factory will add new CIs to the hierarchy as required by the sample, even if the number of CIs then exceeds 1000. However, if the hierarchy already has more than 1000 CIs when a sample is received, no new CIs will be created.

If you think that the number of CIs might be close to or exceeding the defined limit, check the **<Mercury Business Availability Center root directory>\log\EJBContainer\Trinity.log** file. If there is a problem, either change the limit in the **Max number of CIs to create** property, or modify the selector expression to narrow down the definition. (Also take into consideration the overall CI limit for the IT Universe Manager tab.)

➤ **DNode definition key.** Defines how the received data samples are processed to generate the hierarchy. For details, see "DNode Definition Key Property" on page 100.

Note: The default DNode definition key is for a Business Process Monitor source.

➤ **Selector expression.** Defines the filter that catches the samples on the Bus that are required for the Dynamic Node Factory. These samples are processed according to the generator definitions in the DNode definition key property. For example, if you want the generators to be applied against all Business Process Monitor samples, you would define a selector for sampleType "trans_t". If you want to narrow the catchment area, narrow the filter by adding additional selector expression rows.

To define or edit the selector, click **Edit Selector Expression**. For details on defining selector expressions, see "Selectors for KPIs" in *Application Administration*.

Note: If you change values in the DNode definition key property, you should change the selectors accordingly. The selectors must identify samples that contain the keys defined in the generators, otherwise the Dynamic Node Factory hierarchy will be empty.

➤ **Create CI if groupBy value missing.** (Selected by default.) Defines how to handle samples that are missing the grouping (**groupBy**) key. When the check box is selected a CI will still be created for the sample. (If the **nameBy** key is also missing, the CI is given the default name **Empty**.)

➤ **KPI status for removal.** Defines at which KPI status the generated CIs can be considered for removal from the CMDB (default is **Nodata**). If all KPIs for a CI have held this status for at least 24 hours (default period), the CI is eligible to be removed from the CMDB. Once an hour, the Dynamic Node Factory cleaner removes the CIs that are marked as removable.

This option enables automatic cleaning of CIs that have not received corresponding data samples for 24 hours. You can select a different status from the **KPI status for removal** list according to the characteristics of the generated CIs. For example, you may have generated ticket CIs that change to status green (**OK**) when the ticket is closed; after 24 hours at OK status these CIs are considered no longer relevant for your views.

➤ **Time period for removal status.** Defines the time period during which all KPIs for a CI must have the status defined in **KPI status for removal**, before the CI can be marked as removable (default is 24 hours). If you want to change the value, enter a number followed by one of the following suffixes: **s** - for seconds, **m** - for minutes, **h** - for hours, **d** - for days.

Note: The Dynamic Node Factory cleaner is set to review Dynamic Node Factory content and remove expired CIs once an hour. If the value set for the **Time period for removal status** property is less than an hour, then the actual removal time may still be up to an hour, when the next cleaner review takes place.

If required, you can change the default value for the **Dynamic Node Factory clean interval** parameter on the Infrastructure Settings page (**Admin > Platform > Setup and Maintenance > Infrastructure Settings** > context **Foundations** > select **Online Business Logic Engine** from the dropdown list). However, the cleaning values should be changed with care, to avoid disrupting the sensitive balance of the cleaning mechanism.

---

 4 If you are defining a new CI, complete the wizard as described in "Creating a New CI" on page 46.

  If you are editing a CI, click **Apply**.

 5 To define KPIs for the Dynamic Node Factory CI, move to **Admin > Dashboard > KPIs**. Define KPIs and business rules for the Dynamic Node Factory that are logical for the data source, for example, **Availability** and **Performance** for a Business Process Monitor source. You should select a group-type business rule for teach KPI.

  For information on defining KPIs, see "Configuring KPIs" in *Application Administration*.

Note: The dimensions defined for a parent Dynamic Node Factory CI should correlate with the dimensions defined within the DNode definition key property.

DNode Definition Key Property

The DNode definition key property defines how the received data samples are processed to generate the hierarchy under the Dynamic Node Factory CI. This section describes the DNode definition key structure and contents, and how to customize the property.

This section contains the following topics:

➤ "About the DNode Definition Key" below

➤ "Components of the DNode Definition Key" on page 101

➤ "Defining Composite Selectors in the DNode Definition Key" on page 109

Important note: Only advanced users should make changes to templates. An incorrect definition may lead to loss of data, the wrong hierarchy build, and redundancy in the CMDB.

About the DNode Definition Key

The DNode definition key defines the properties for the CIs created by the received samples. This includes generators to determine how to group the incoming samples, how many CI levels to create, and which templates to use when creating each CI (KPIs, business rules, and so forth). Each different data type needs an appropriate definition in the Dynamic Node Factory file.

You can customize the definitions in the DNode definition key to make them relevant for a specific data source and to build the hierarchy structure you require. This customization can be done by:

➤ editing the Dynamic Node Factory CI in the IT Universe Manager tab (recommended method). The procedure for editing a Dynamic Node Factory CI is described in "Defining/Editing Properties for a Dynamic Node Factory CI" on page 94.

➤ editing the template for the source adapter in Source Manager (only relevant for predefined Dynamic Node Factory CIs). For information on editing adapter templates, see "Working with Templates" in *Source Manager Administration*.

The default DNode definition key property (used when you create a new Dynamic Node Factory CI in the IT Universe Manager tab) contains definitions for a Business Process Monitor hierarchy:

DNode definition key

```
<?xml version="1.0" encoding="UTF-8"?><dnode><Generator
groupBy="szSessionName" itemTemplate="profile"
linkTemplate="depends_on"><Generator groupBy="szTransactionName"
itemTemplate="transaction" linkTemplate="depends_on"><Generator
groupBy="szLocationName" itemTemplate="tx_location"
```

If you are adding a Dynamic Node Factory CI for a different data source, for example, SiteScope, you must replace the definitions with ones appropriate for the source.

For explanations of the data samples and their fields, see "Samples" in *Reference Information*.

For descriptions of the definitions contained in the DNode Definition Key, see the following section.

Components of the DNode Definition Key

In order to customize a DNode Definition Key, you need some understanding of its various components.

All changes to the parameters in the DNode definition key property must be logical and relevant for the source.

Generators

The first part of the DNode definition key contains the **Generators** defining the hierarchy for the CIs. Each generator in the DNode definition key defines a level to be built in the Dynamic Node Factory hierarchy (the XML provides for definitions of up to three levels). For example, for Business Process Monitor, the Dynamic Node Factory may build levels for profile, transaction, and location.

For example, the generators in the default DNode definition key for Business Process Monitor are as follows:

```
<Generator groupBy="szSessionName" itemTemplate="profile"
linkTemplate="depends_on">
<Generator groupBy="szTransactionName" itemTemplate="transaction"
linkTemplate="depends_on">
<Generator groupBy="szLocationName" itemTemplate="tx_location"
linkTemplate="monitored_by"/>
```

➤ **groupBy.** Defines the property to look for in the samples, to see if the sample is relevant for building this level of the hierarchy. A CI is created for each unique value of this key. For example, all samples with a unique value for szSessionName will create a CI at the profile level of the hierarchy.

Note: If samples are missing this key, a CI is still created for the sample if the **Create CI if groupBy value missing** is selected for the parent Dynamic Node Factory CI. For more information, see "Defining/Editing Properties for a Dynamic Node Factory CI" on page 94.

➤ **itemTemplate.** Defines the name of the CI template to be used for creating each CI at this level. The CI templates are defined in the DNode definition key after the generators.

➤ **linkTemplate.** Defines the name of the relationship template to be used for each CI created at this level. The relationship template defines the relationship between the created CI and its parent CI. The relationship templates are defined in the DNode definition key after the CI templates.

Note: It is possible to use the DNode definition key property to build more complex dynamic hierarchies, for example, by using more than one generator at the same level. For more information, contact Mercury Customer Support.

CI Templates

The second part of the DNode definition key contains the **ItemTemplates**, the templates for creating the CIs. Each template defines attributes, KPIs, and business rules to be used for each generated CI. (If required, you can also add a composite selector to a monitor ItemTemplate. For details, see "Defining Composite Selectors in the DNode Definition Key" on page 109.)

For example, the following ItemTemplate creates Business Process Group CIs for the profiles in the default Business Process Monitor hierarchy:

```xml
<ItemTemplate template="profile" namespace="Mercury" class-
name="bp_group">

     <Attributes>
            <Attribute name="data_name" ref_value="szSessionName"/>
            <Attribute name="id" ref_value="u_iSessionId"/>
            <Attribute name="app_id" value="-1"/>
     </Attributes>

     <KPI>
            <Attributes>
                   <Attribute name="type" value="6"/>
                   <Attribute name="rule" value="1"/>
            </Attributes>
     </KPI>

     <KPI>
            <Attributes>
                   <Attribute name="type" value="7"/>
                   <Attribute name="rule" value="1"/>
            </Attributes>
     </KPI>

     <!--Specify objectives only if you want to override the default values for
objectives-->

</ItemTemplate>
```

Item Template Line

The first line of the ItemTemplate defines the following parameters:

```
<ItemTemplate template="profile" namespace="Mercury" class-
name="bp_group">
```

➤ **template.** Defines the name of the template.

➤ **namespace.** Defines the context for the class-name: **Mercury** is used for all CIs.

➤ **class-name.** Defines the configuration item type (CIT) name for the CIs generated by this template. This must be a valid CMDB CIT. The attributes for the created CIs are taken from the CIT definition.

This is the internal name used for the CIT, not the display name. For example, the internal name for Business Process Group CIs is **bp_group**.

To view the internal name for each CIT: Open the **Admin** > **CMDB Administration** > **CI Type Manager** page and select **Edit CIT** from the right-click menu for a CIT to display the Edit Configuration Item Type dialog box. In the **Information** tab, the internal name is displayed in the **Configuration Item Type Name** field.

Attributes Section

Most of the CITs have mandatory attributes. These attributes must be included in the **Attributes** section of the ItemTemplate (in the following example, the **id** and **app_id** are mandatory attributes).

The Attributes section defines the following parameters:

```
<Attributes>
        <Attribute name="data_name" ref_value="szSessionName"/>
        <Attribute name="id" ref_value="u_iSessionId"/>
        <Attribute name="app_id" value="-1"/>
</Attributes>
```

➤ **name.** Defines the internal name of an attribute included in the attributes list for the relevant CIT. For example, **data_name** is the internal attribute name corresponding to the **Name** property for the created CI.

To view the list of attributes defined for a CIT: Open the **Admin** > **CMDB Administration** > **CI Type Manager** page and select **Edit CIT** from the right-click menu for a CIT to display the Edit Configuration Item Type dialog box. Click the **Attributes** tab to see the list of attributes—the **Name** column lists the attributes by their internal name.

➤ **ref_value.** Defines a key in the data sample that will provide the value for the attribute. In the example <Attribute name="id" ref_value="u_iSessionId"/>, the **id** attribute for each CI is given the value from the property **u_iSessionId** in the incoming samples.

➤ **value.** Defines a static value for the attribute. This enables you to use a predefined value for all CIs generated by this ItemTemplate. For example, you can define a static value attribute for **data_name**, where you know that there will only be one CI at this level: <Attribute name="data_name" value="my_BPM_profile"/>

Note: The Business Process Group CI template includes the static value attribute <Attribute name="app_id" value="-1"/>. This **app_id** attribute corresponds to the Business Process Monitor source adapter, and is only required for Business Process Monitor hierarchies.

Every ItemTemplate should contain attributes defining a unique name and an appropriate ID for each generated CI (otherwise the CIs will be created on top of each other). You can define additional attribute keys for the template, as required.

If you want to define a context menu in the ItemTemplate, so that it will apply for all CIs generated by the template, you need to use a special string format. The value for the context menu attribute is presented as XML, and so must be "escaped" from the DNode definition key XML, as in the following example:

<Attribute name="contextmenu" value="<Values><Value>**groupMenu**</Value></Values>"/>

The value in bold shows the internal context menu name. (You can find the internal context names in: **<Mercury Business Availability Center server>\AppServer\webapps\site.war\bam\conf\ repositories\contextMenu\def.xml**)

KPI Section

The **KPI** section of the ItemTemplate defines the following parameters:

```
<KPI>
      <Attributes>
              <Attribute name="type" value="6"/>
              <Attribute name="rule" value="1"/>
      </Attributes>
</KPI>
```

➤ **name="type"** Defines the KPI to be attached to the CIs generated by this ItemTemplate. The value corresponds to the ID number for a KPI, for example, **6** is the ID number for the **Performance** KPI. The ID numbers for the KPIs are listed in the **Admin** > **Dashboard** > **Repositories** > **KPIs** page, under the **ID** column.

Each ItemTemplate in the DNode definition key should specify the same KPIs, and these should correlate with the KPIs defined for the Dynamic Node Factory CI itself.

➤ **name="rule"** Defines the business rule to be used with the KPI for the generated CIs. The value corresponds to the ID number for a rule, for example, **1** is the ID number for the **Worst Child Rule**. The ID numbers for the business rules are listed in the **Admin** > **Dashboard** > **Repositories** > **Business Rules** page, under the **ID** column.

The specified business rule should be a monitor rule if the ItemTemplate is defining leaf CIs, and a group rule if the ItemTemplate is defining non-leaf CIs. For more information on rule types, see "About Business Rules" in *Application Administration*.

If required, you can define additional parameters in the KPI section. Attributes can use **value** or **ref_value**, as described in the "Attributes Section" on page 104.

The following additional parameters are relevant for leaf monitor CIs defined in the DNode definition key, and should be declared as part of the KPI:

➤ **RuleArguments.** Defines parameters used by the business rule. When you add a rule argument to a KPI definition in the DNode definition key, it overrides the default value for the parameter defined in the Rules Repository. For example, the following definition will override the default value for the **No data timeout** parameter (used with, for example, the Transaction Availability Rule), and the generated CIs will use the new value for the KPI:

<RuleArguments></RuleArguments>

For details on the parameters used by the business rules, see "Business Rules Repository" in *Repositories Administration*.

Note: If you add definitions for **RuleArguments**, make sure to define the exact key name and to use the correct format for the value.

➤ **Objective.** Defines objective values that override the default objective parameter values declared in the Rules Repository. The objective values should be relevant for the rule.

For example, the following definition will override the default value for the
Critical (red) objective level (Objectives section shown in bold):

```
<KPI>
   <Attributes>
      <Attribute name="type" value="6"/>
      <Attribute name="rule" value="13"/>
   </Attributes>
   <RuleArguments/>
   <Objective>
      <Attributes>
         <Attribute name="critical" value="12"/>
      </Attributes>
   </Objective>
</KPI>
```

Note: The KPI definitions should be in the order shown above: KPI, business
rule, rule arguments, objectives.

Relationship Templates

The final part of the DNode definition file contains the **LinkTemplates**, the
templates for creating the relationships between the nodes.

For example, the LinkTemplates in the default DNode definition key for
Business Process Monitor are as follows:

```
<LinkTemplate template="depends_on" namespace="Mercury" class-
name="depends_on"/>
</LinkTemplate>

<LinkTemplate template="monitored_by" namespace="Mercury" class-
name="monitored_by">
</LinkTemplate>
```

➤ **template.** Defines the name of the template.

➤ **namespace.** Defines the context for the class-name: **Mercury** is used for all
CIs.

➤ **class-name.** Defines the internal name for the relationship generated by this template, for example, **depends_on**. The attributes for the created relationship are taken from the definitions in the CI Type Manager.

To view the internal name for each relationship: Open the **Admin > CMDB Administration** > **CI Type Manager** page and select **Edit CIT** from the right-click menu for a relationship to display the Edit Configuration Item Type dialog box. In the **Information** tab, the internal name is displayed in the **Configuration Item Type Name** field.

If required, you can define attributes for the relationships; the values will override the default values for the relationship properties.

Defining Composite Selectors in the DNode Definition Key

The Dynamic Node Factory uses the selector expression defined for the Dynamic Node Factory CI to receive samples from the Bus and create the CIs in the hierarchy. If required, you can define a **composite selector** for the monitor ItemTemplate (leaf level) in the DNode definition key, so that the new selector is used to receive update samples for KPI status, rather than the original selector that generated the CI.

Composite selectors are used when you want the created CI to be independent from the Dynamic Node Factory generators. This may be required when the original selector expression information is not included in later samples.

For information on defining selector expressions, see "Selectors for KPIs" in *Application Administration*.

You can use static or dynamic attributes in composite selector expressions. When using static attributes, you define a known value for a known referenced property, for example:

`<Selector key="sampleType" op="EQ" type="String" value="ss_t" />`

Note: The attributes in a selector are case sensitive, so take care to define the attributes exactly as in the samples.

If you do not know the value of a selector attribute in advance, you can use a dynamic selector expression. The value for the dynamic attribute is defined only when the sample is received.

You make an attribute dynamic by using the prefix **ref_** with the attribute name. For example:

```
<Selector key="u_iSessionId" op="EQ" type="Long" ref_value="u_iSessionId" />
```

In this example, the **value** attribute is dynamic, so the value for the referenced property will be taken from the **u_iSessionId** field in the received sample.

You can define **ref_value**, **ref_key**, **ref_type**, and **ref_op** in dynamic selector expressions, and each dynamic selector expression can contain more than one dynamic attribute.

Note: The dynamic selector expression option is not relevant when working with custom data (fully descriptive samples are received from the source, so there is no problem of insufficient information).

The following example shows part of the ItemTemplate for a SiteScope monitor with an added composite selector (shown in bold), including dynamic selector expressions:

```
<ItemTemplate template="monitor" namespace="Mercury" class-
name="sitescope_monitor">
    <Attributes>
        …
    </Attributes>
    <CompositeSelector logicalOp="AND">
        <Selector key="sampleType" op="EQ" type="String" value="ss_t" />
        <Selector key="u_iSessionId" op="EQ" type="Long"
        ref_value="u_iSessionId" />
        <Selector key="u_iMonitorId" op="EQ" ref_value="u_iMonitorId" />
    </CompositeSelector>
    <KPI>
        …
    </KPI>
</ItemTemplate>
```

Note: If necessary, you can define a different composite selector for a KPI in the ItemTemplate, if the KPI requires a different selector from the default for the monitor.

7

Example Scenario for an Instance View

This chapter provides a scenario for setting up an instance view to use in Dashboard and Service Level Management.

About the Example Scenario

The example scenario is for creating an instance view that shows KPI status for lines of business.

In this scenario, a new view is created for A.A. Web Tours to show real-time performance for the company's main lines of business, and over time impact of network problems on major customers.

Note that this scenario provides an example for setting up a line of business view with customized KPI settings. The actual steps performed and parameters defined when setting up a view will vary according to your organization's requirements.

Stage 1: Create the View

The first stage is creating the framework for the view. In this scenario, the new view is created from the IT Universe Manager tab.

Note: It is also possible to create instance views from the View Manager tab.

To create the new view in the IT Universe Manager tab:

1 In the **Admin** > **CMDB** > **IT Universe Manager** tab, click **New Instance View**.
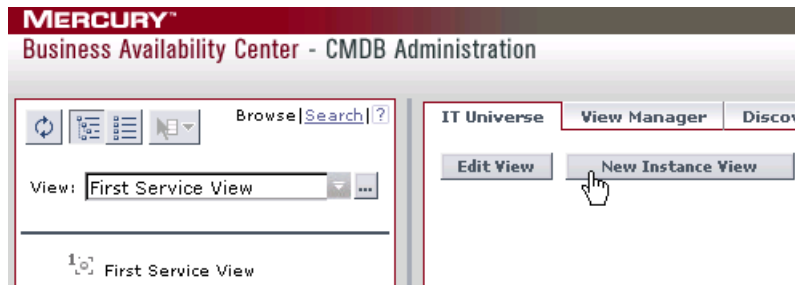
The Create View - View Properties dialog box opens.

2 Enter a logical **View name** (for example, Web Tours LOB) and a **Description**. In the **Assign view to** options, select **IT Universe** (so that the view can be edited in the IT Universe Manager tab) and deselect **Dashboard & Service Level Management** (so that the view will not be available in these applications until it is ready).

Click **OK**.

3 The view is created and automatically opens in View Explorer.
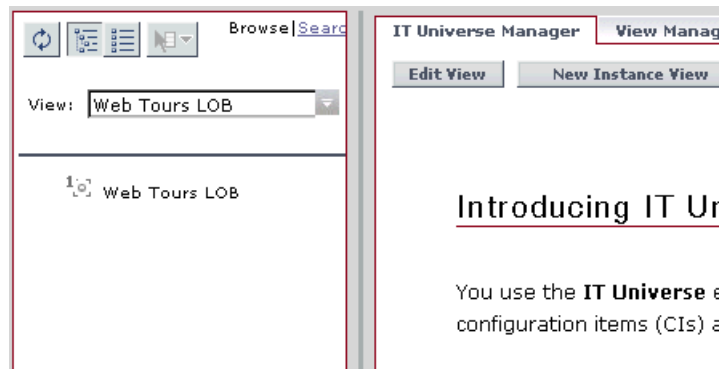
You can immediately continue with building the view, as described in "Stage 2: Build the View" on page 116.

> **Note:** The new view is created by default in the same folder as the original view that you had open in View Explorer. At any stage after creating the view, you can open the View Manager tab and move the view into a different folder.

Stage 2: Build the View

The second stage of the scenario involves populating the view with CIs.

For this scenario, the hierarchy created for A.A. Web Tours lines of business is:
Line of business > **customer** > **service** > **host** > **monitor**

The first three levels are logical CIs that are manually created in the view. The fourth level are CIs added to the CMDB by the discovery process. The final level are monitor CIs added to the CMDB by the source adapters.

This stage of the scenario describes how to add the CIs to the view using the Default viewing mode (all steps could also be performed using Topology Map viewing mode), and using the different edit options available in the IT Universe Manager tab. However, all the steps described in this stage could also be performed from the Instance View Editor (described in "Editing with the Instance View Editor" on page 28).

This section is divided into the following topics:

➤ Create Root CIs in the View, described below

➤ Create Logical CIs Related to CIs in the View, described on page 121

➤ Attach CIs from Discovery Process to CIs in the View, described on page 125

➤ Attach Monitoring CIs to CIs in the View, described on page 132

Create Root CIs in the View

This section describes the creation of the line of business CIs in the views, to be the root CIs for each subtree. These new logical CIs are added to the Web Tours view and the CMDB without any relationship to any other CI.

To create root CIs in the view:

1 In the **Admin** > **CMDB** > **IT Universe Manager** tab, open the created view in View Explorer (left pane).
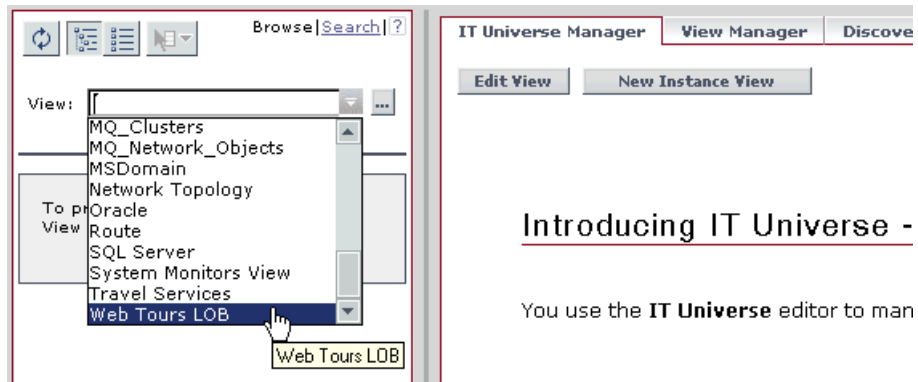
2 Create the Line of Business CIs for the root level of the hierarchy. With the view name selected, either select **New CI** from the right-click menu for the view name, or click **Create New CIs** in the right pane.
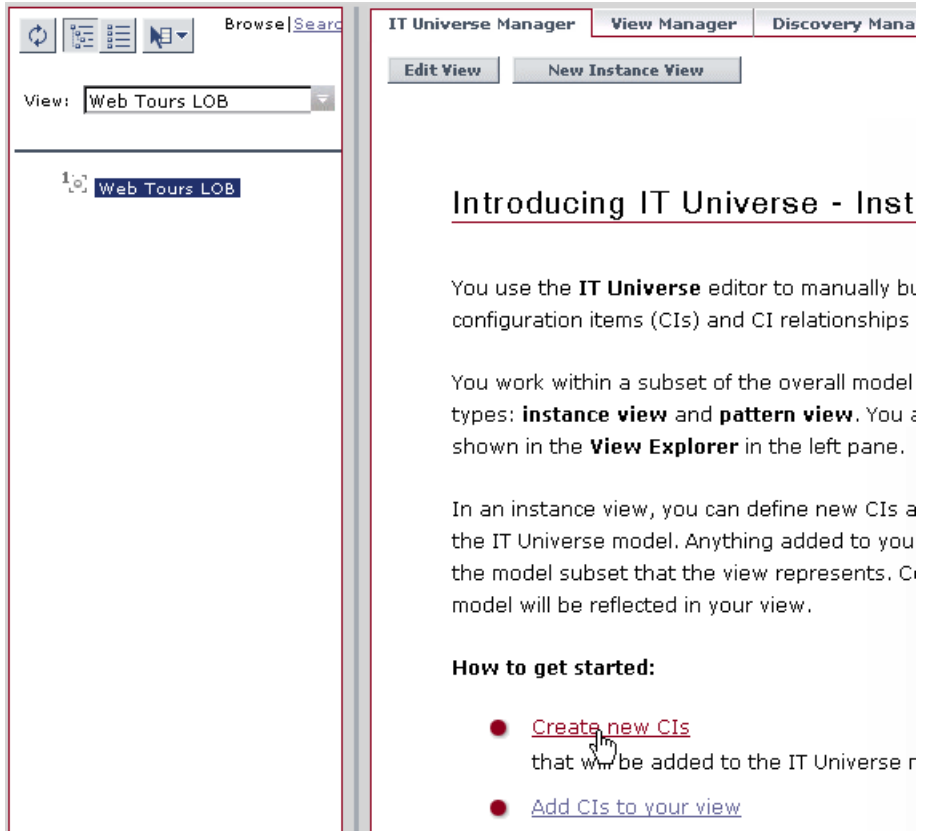
The New CI Wizard opens at the Define General Properties page.

3 Select **Line of business** from the **CI Type** list. Enter the CI **Name** (for example, Corporate Travel) and **Description**.

Click **Finish**.

4 The Corporate Travel CI appears as a root CI in the Web Tours LOB view.
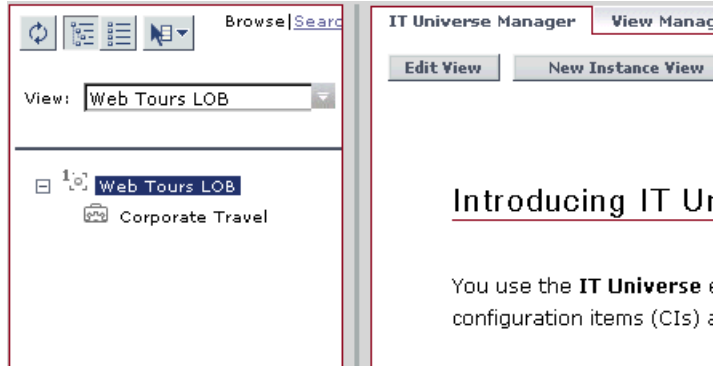
Repeat steps 2 and 3 to create root CIs for the other lines of business: Package Tours and Private Travel.
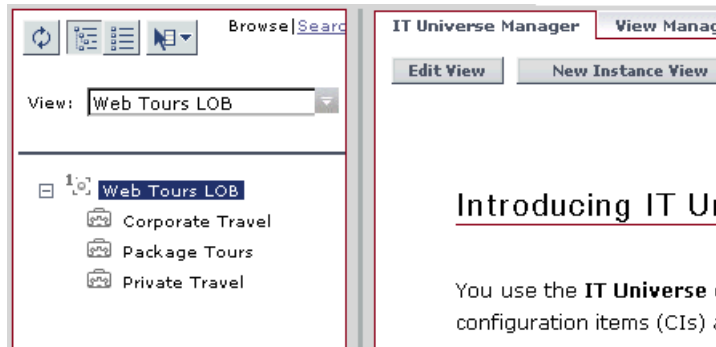
Create Logical CIs Related to CIs in the View

This section describes the creation of the customer CIs that are attached to the existing LOB CIs in the view, creating relationships between the CIs.

To create related logical CIs in the view:

1 Select the CI to attach to in View Explorer (for example, Corporate Travel), and then either select **New Related CI** from the right-click menu for the CI, or click the **New Related CI** button in the right-hand pane.
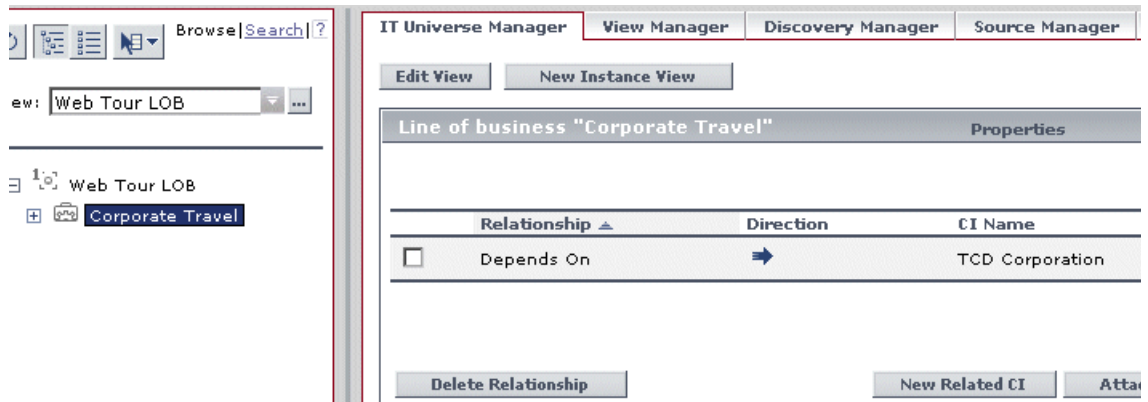
Note: You must be working in the IT Universe Manager **Related Configuration Items** tab to see the buttons in the right-hand pane.

The New CI Wizard opens at the Define General Properties page.

2 Define details for the customer. Select **Customer** from the **CI Type** list. Enter the CI **Name** (for example, TCD Corporation) and select the customer location from the dropdown lists.

Click **Next** to move to the Define CI Relationship page.

3 Check that the details are correct.

Click **Finish**.

4 The created Customer CI for **TCD Corporation** is shown in View Explorer attached as a child of the parent CI, **Corporate Travel**. The relationship for the attached CI is displayed in the **Related Configuration Items** tab.

| | | Browse|Sea | IT Universe | View Manager | Discovery Manager | Source Manager | CI Type Manager |

View: Web Tours LOB

Edit View   **New Instance View**   Viewing Mode: Def

Line of business "Corporate Travel"   Properties   Related Configura

1-1 of

	Relationship ▲	Direction	CI Name	CI Type	
☐	Depends On	➡	TCD Corporation	Customer	✎

Delete Relationship   **New Related CI**   **Attach Related CIs**   **Attach Mor**

Web Tours LOB
- Corporate Travel
  - TCD Corporation
- Package Tours
- Private Travel

Repeat steps 1 through 3 to create additional logical CIs in the view that represent the upper levels of Web Tours LOB hierarchies.

| | | Browse|Sea | IT Universe | View Manager | Discovery Manager | Source Manager | CI Typ |

View: Web Tours LOB

Edit View   **New Instance View**   Viewing

Customer "TCD Corporation"   Properties   Relate

	Relationship ▲	Direction	CI Name	CI Type
☐	Depends On	➡	Economy	Service
☐	Depends On	⬅	Corporate Travel	Line of bu
☐	Depends On	➡	First Class	Service

Delete Relationship   **New Related CI**   **Attach Related CIs**

Web Tours LOB
- Corporate Travel
  - New Eastern ANS
  - TCD Corporation
    - Economy
    - First Class
- Package Tours
  - Golden Years
  - Singles World
- Private Travel
  - Sunnyvale private
  - Unionville private

Attach CIs from Discovery Process to CIs in the View

This section describes the attachment of existing CIs in the CMDB to the LOB subtrees. These existing CIs are NT host machines that discovery process discovered in the network and added to the Network Topology view.

The host machines are running the travel service applications that are relevant for the LOB subtrees.

To attach discovery process CIs:

1 Select the CI to attach to in View Explorer (for example, **Economy**), and then either select **Attach Related CIs** from the right-click menu for the CI, or click the **Attach Related CIs** button in the right-hand pane.
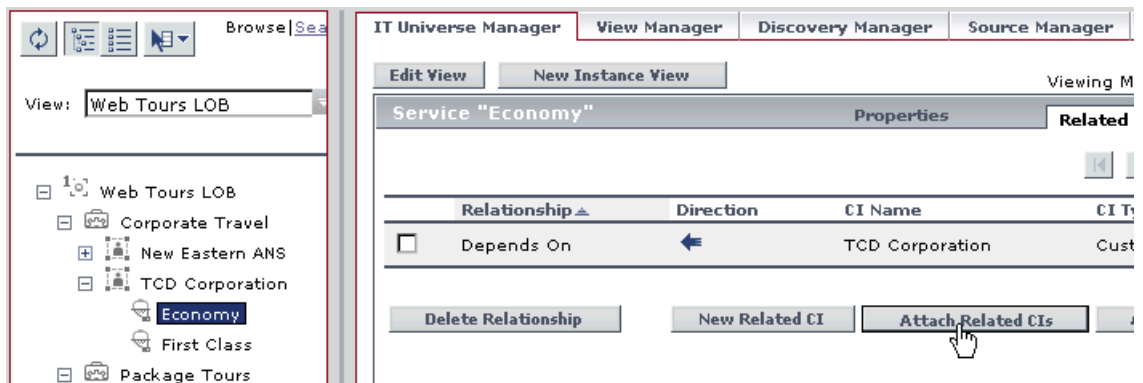
The Attach Related CIs wizard opens at the Select Related CIs page.

2 The left pane displays the **Web Tours LOB** view. Change to the **Network Topology** view by selecting it from the **View** list.
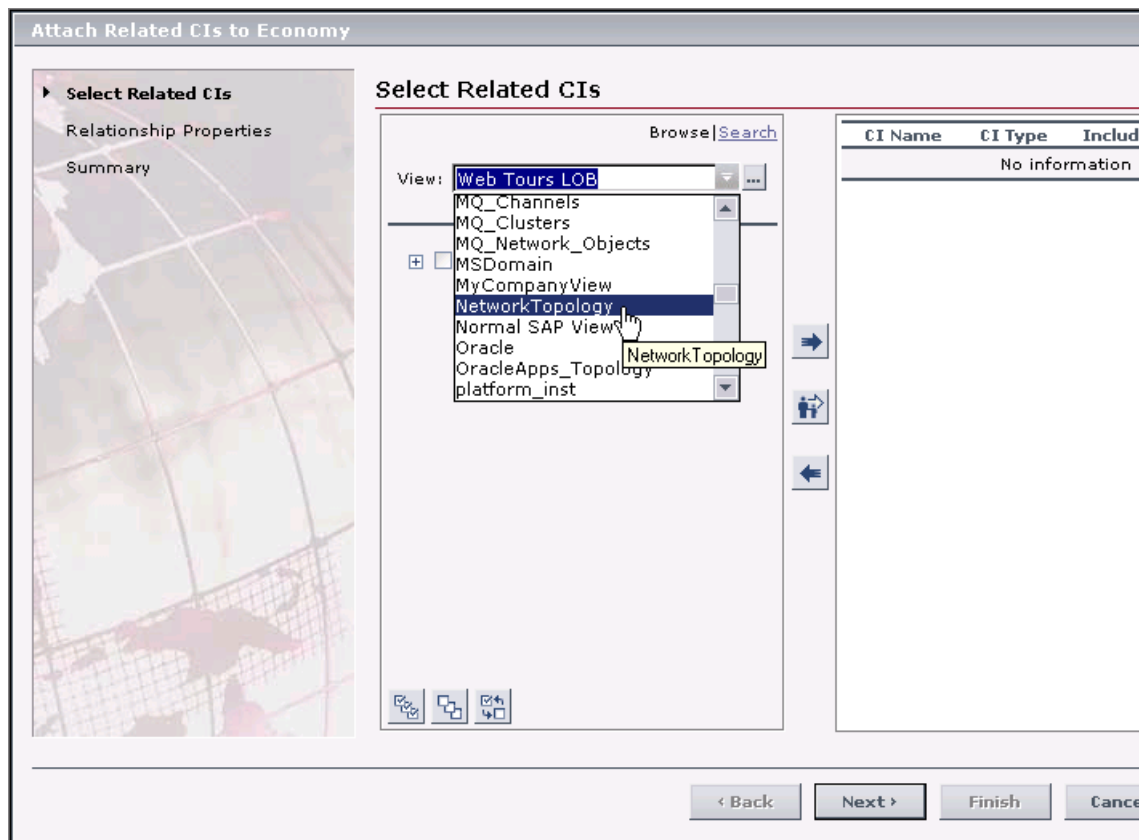
3 In the **Network Topology** view, select the check boxes for the required CIs—
CIs representing the host machines that are running the applications
relevant for this customer subtree (**TPRUM** and **sheriff**).

4 Click the top arrow button to add the selected CIs (without their child CIs) to the **Web Tours LOB** view.
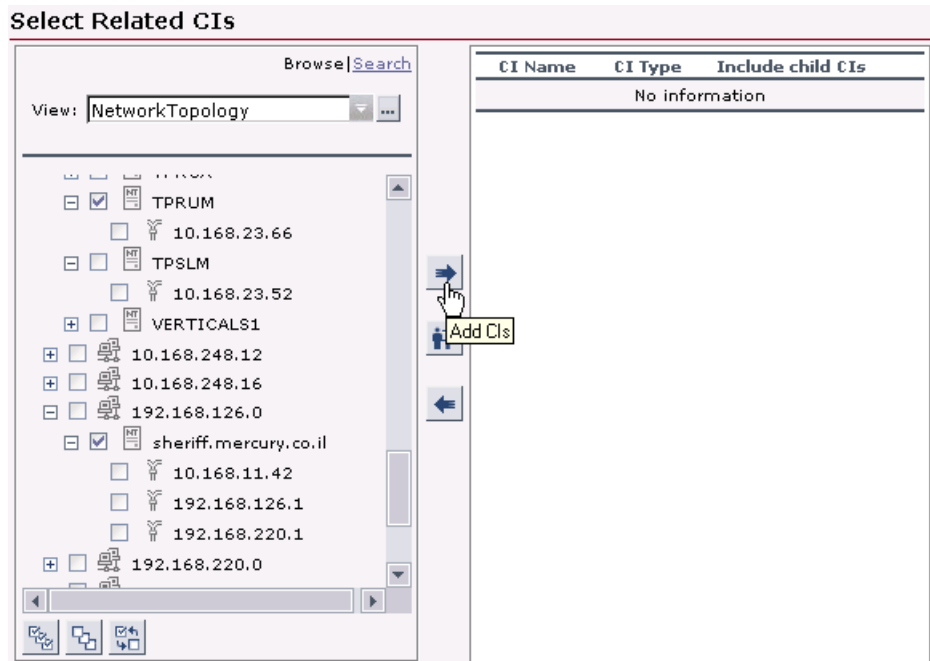
5 The selected CIs are listed in the right-hand pane.
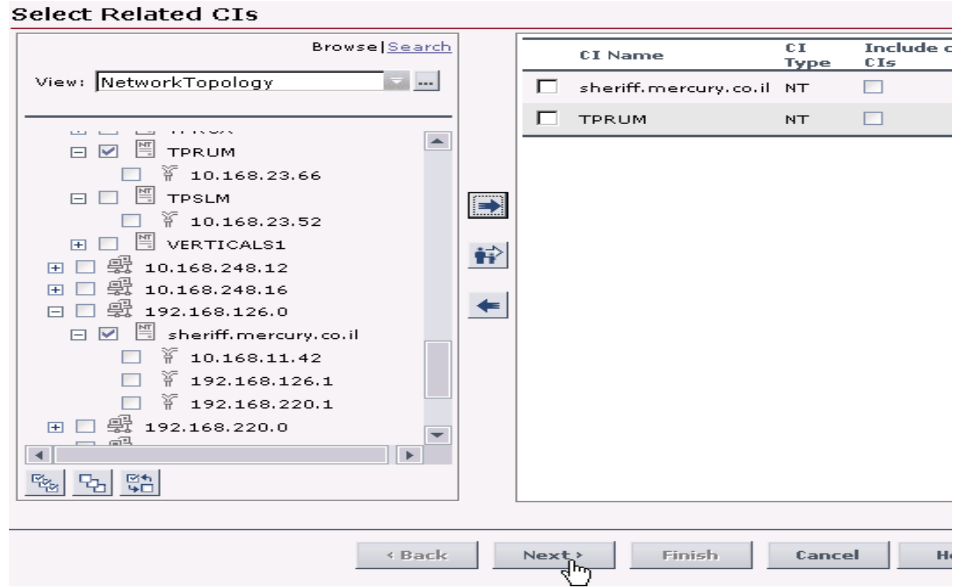
Click **Next** to move to the Relationship Properties page.

6 The Relationship Properties page defines the relationship to be created between the **Economy** CI and the CIs added from the Network Topology view.

Click **Finish** to move to the Summary page.

7 The Summary page summarizes the CIs added to the view.

Click **OK**.

8 The CIs you added to the view appear in View Explorer attached as children of the parent CI, **Economy**. The relationships for the attached CIs are displayed in the **Related Configuration Items** tab.
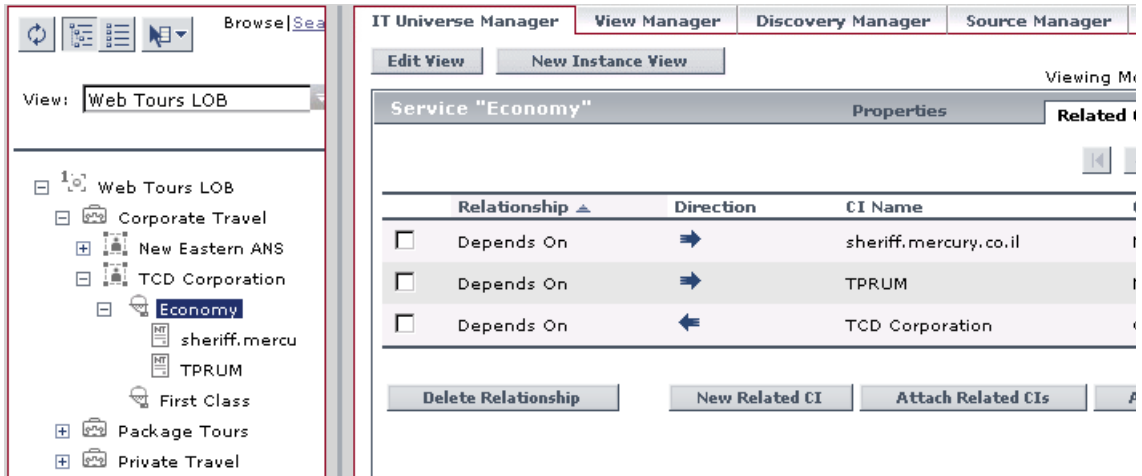
Repeat steps 1 through 7 to add additional existing CIs to the view.

Attach Monitoring CIs to CIs in the View

This section describes the attachment of monitor CIs or monitor subtrees to the host CIs. These monitoring CIs are checking network performance for the host machines, and monitoring system service on the machines.

Note that the individual monitor CIs and the monitor subtrees cannot be attached to a CI in a single operation, because they are added to the parent CI using different relationships (depends on, monitored by).

To attach monitoring CIs:

1 Select the CI to attach to in View Explorer (for example, **TPRUM**), and then either select **Attach Monitor CIs** from the right-click menu for the CI, or click the **Attach Monitor CIs** button in the right-hand pane.
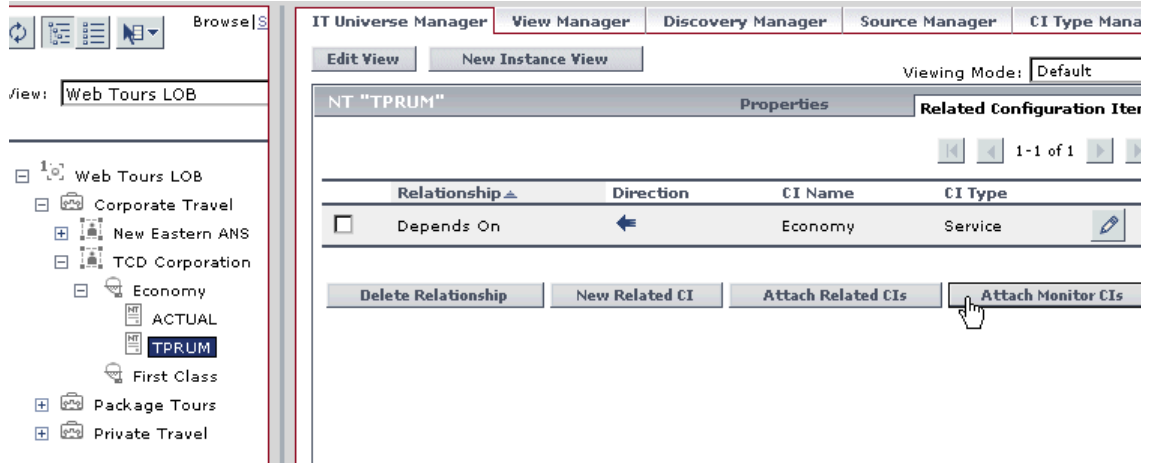
The Attach Related CIs wizard opens at the Select Related CIs page.

2 The Select Related CI page displays the **Monitors View** in the left-hand pane. Select group CIs to attach the required monitoring subtrees to **TPRUM**, for example, the SiteScope Measurement Group CI **Ping tprum.mercury.co.il**, then click the recursive (middle) arrow button (to include the child CIs).
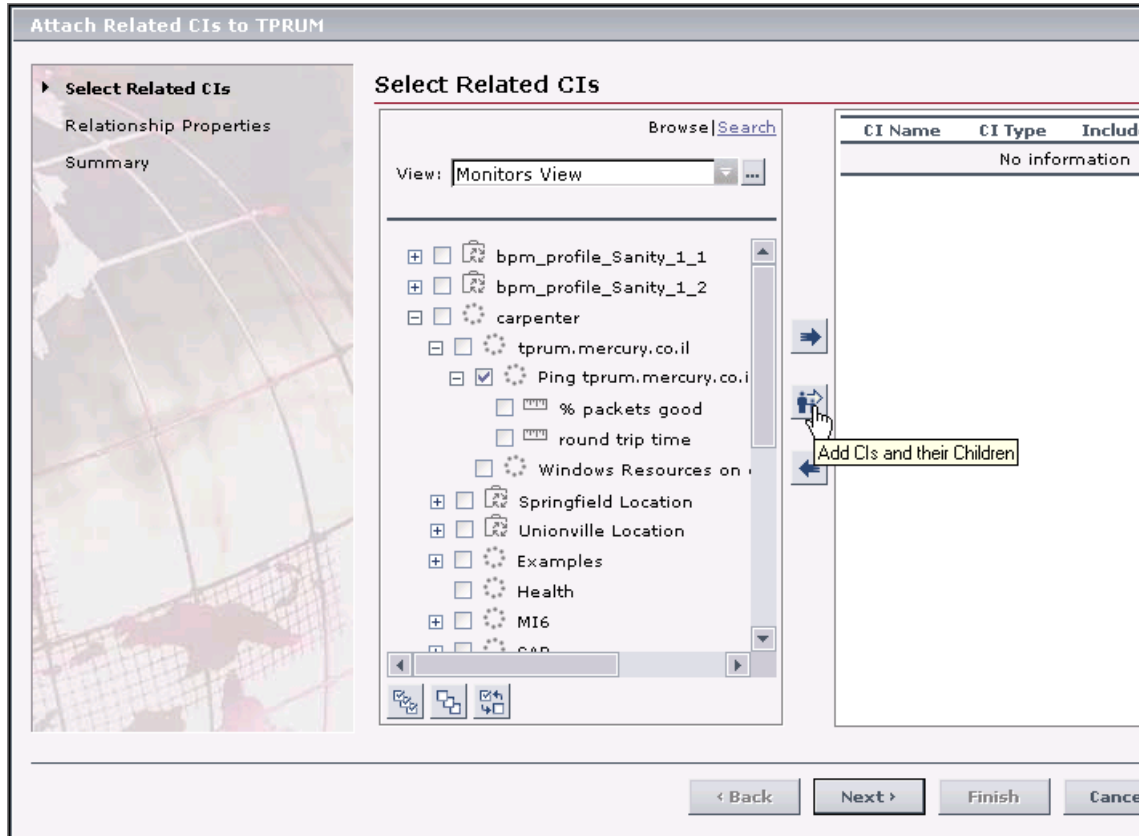
3 The selected CIs are listed in the right-hand pane.
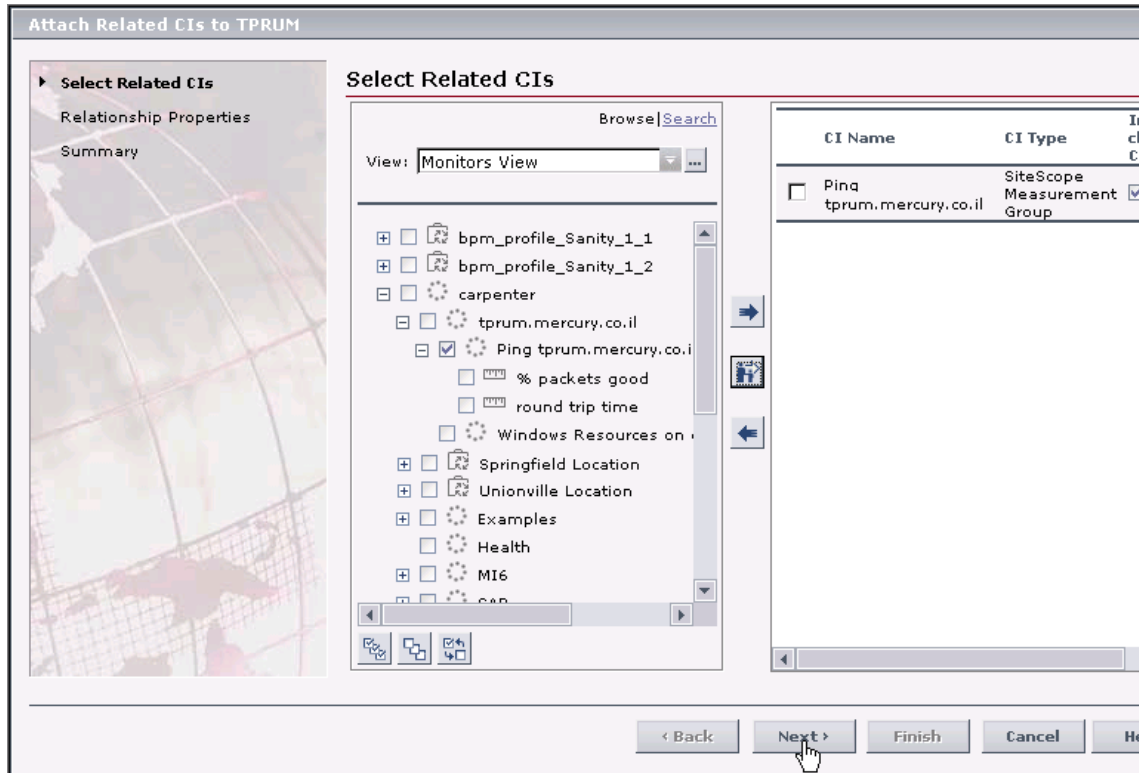
Click **Next** to move to the Relationship Properties page.

4 The Relationship Properties page defines the **Depends On** relationship to be created between the **TPRUM** CI and the **Ping tprum.mercury.co.il** CI.

Click **Finish** to move to the Summary page.

5 The Summary page summarizes the CIs added to the view.
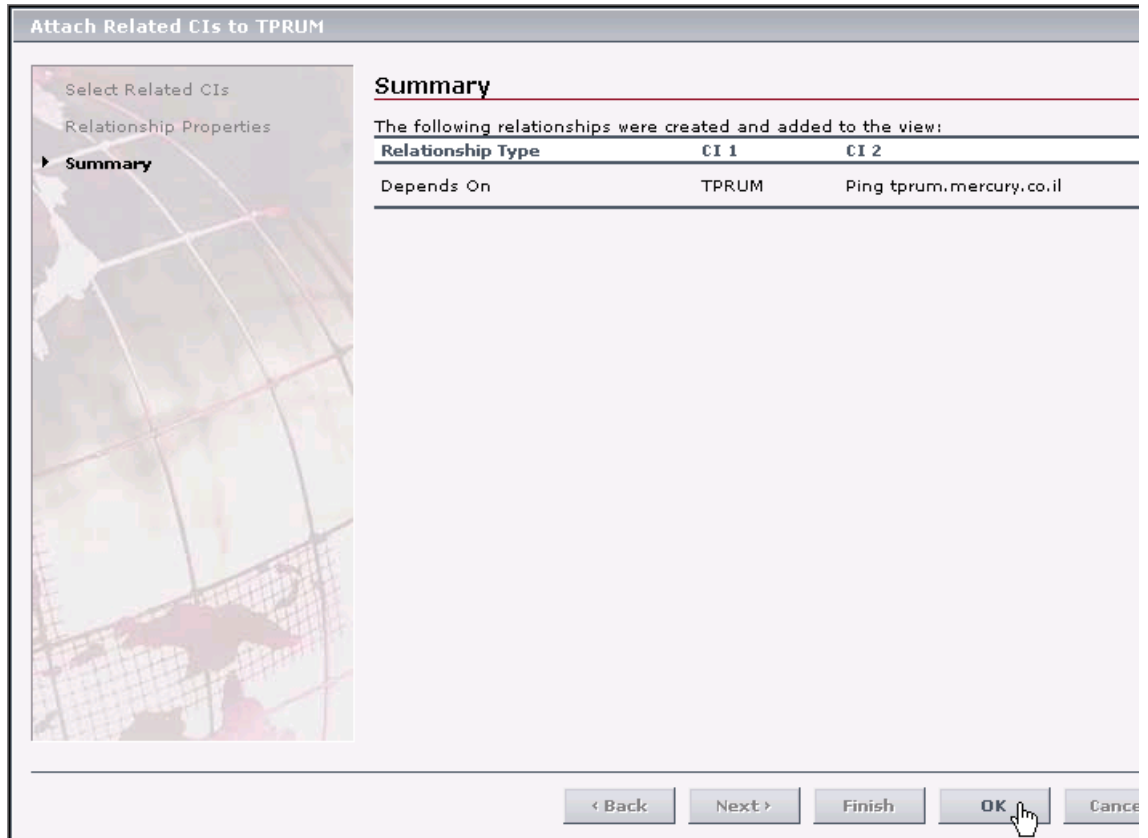
Click **OK**.

6 Repeat steps 1 and 2 to attach individual monitor CIs to **TPRUM**, for
example, the BPM Transaction CI **Springfield_travel_book_flight**. These CIs
are added as leaf CIs for the subtree.
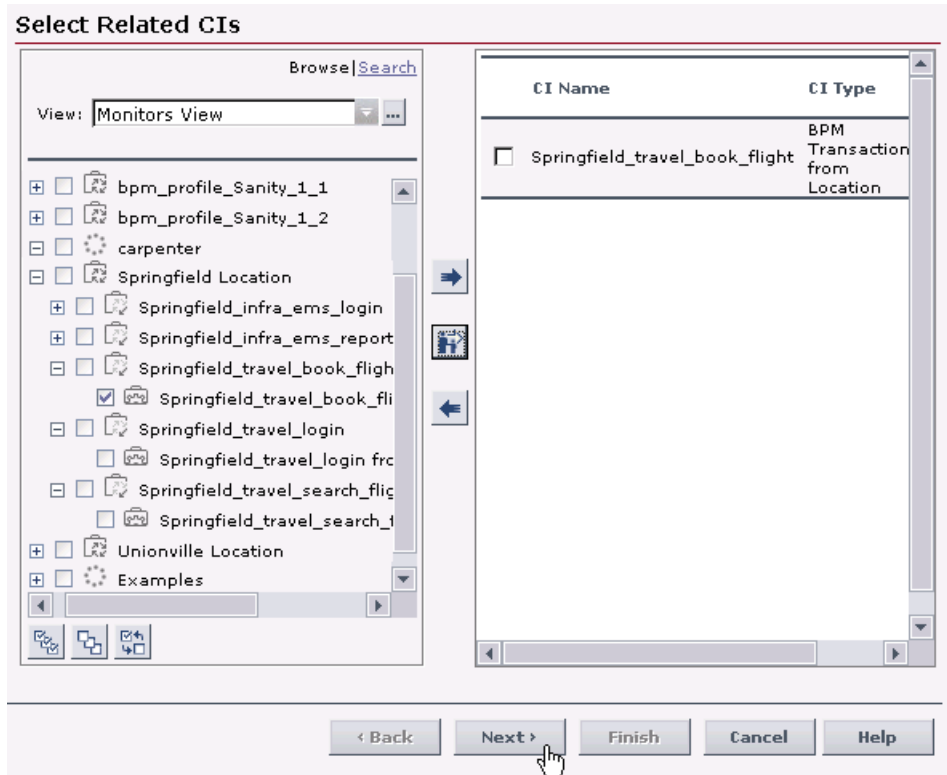
Click **Next** to move to the Relationship Properties page.

7 The Relationship Properties page defines the **Monitored By** relationship to be created between the **TPSLM** CI and the monitor CI.

Click **Finish** to move to the Summary page, then click **OK**.

8 The monitoring CI you added to the view appear in View Explorer as a child of the parent CI, **TPRUM**. The relationships for the attached CIs are displayed in the **Related Configuration Items** tab.
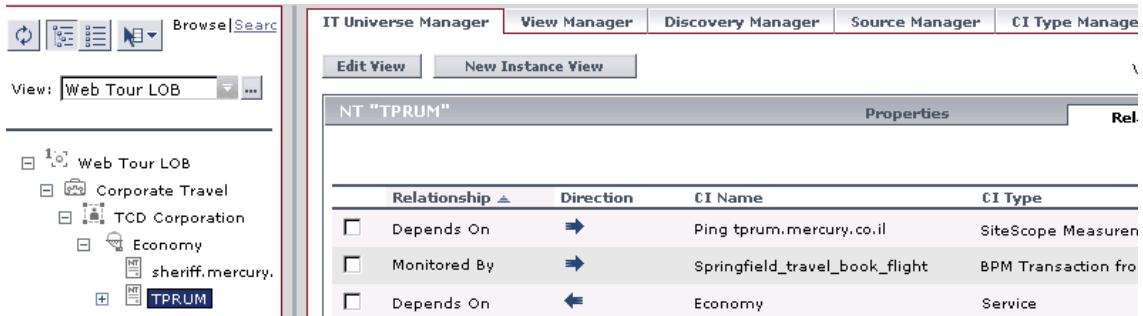
Repeat steps 1 through 7 to add additional existing CIs to the view.

Stage 3: Customize KPI Settings

The third stage of the scenario involves customizing the KPI setup for the view, to add an additional KPI category to the CIs, and to change the default rule calculation for some KPIs.

The KPIs for the monitor CIs automatically propagate up through the hierarchy as soon as the monitor CIs are attached to the view.

Customizing KPI settings is divided up into two sections:

➤ Define Over Time Impact KPI for CIs, described in the following section

➤ Define Weighted Percentage Rule, described on page 145

Define Over Time Impact KPI for CIs

For this scenario, A.A. Web Tours want the Web Tours LOB view to show over time impact metrics, based on availability of their network resources. The OT Impact KPI is defined for the BPM Transaction monitors (which by default have the Availability KPI attached). The OT Impact KPI settings are customized to represent the cost to the company when each resource is unavailable.

To define the OT Impact KPI:

1 In the **Admin** > **Dashboard** > **KPIs** tab, open the Web Tours LOB view in View Explorer (left pane).
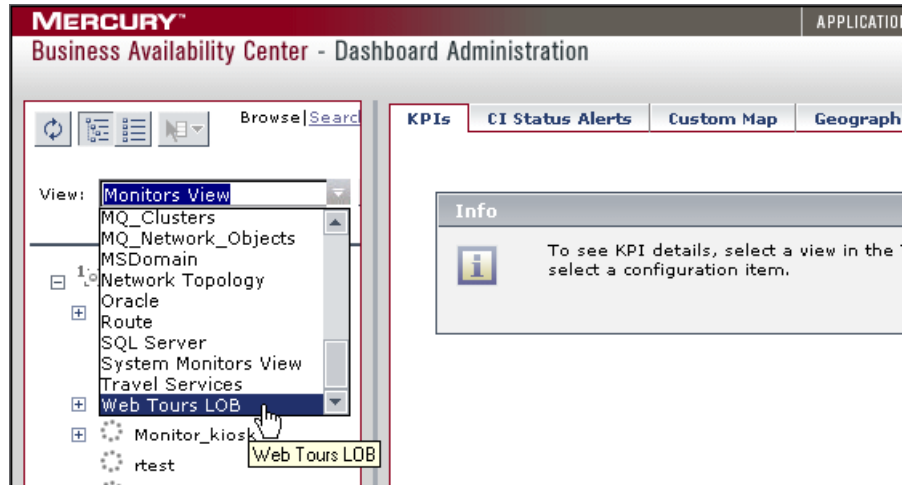
2 Select a monitor CI in the view, for example, **Springfield_travel_book_flight**, then click **New KPI** in the right-hand pane.
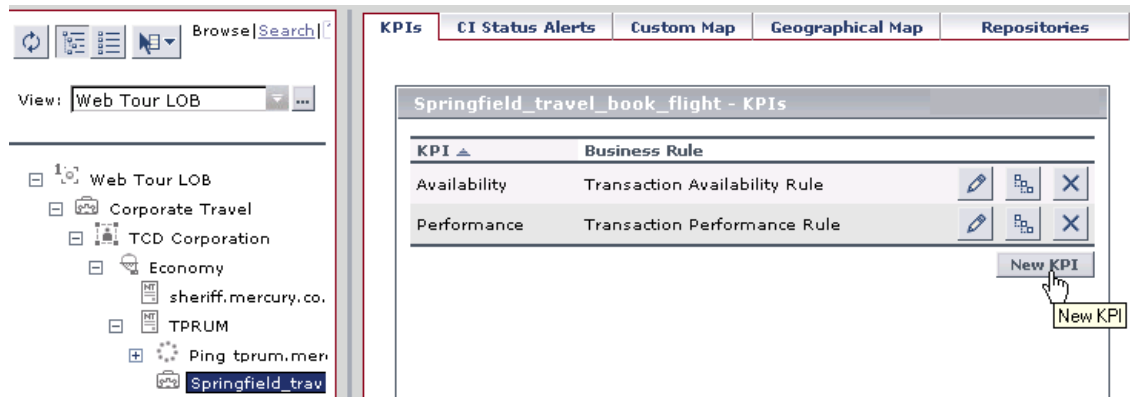
The New KPI dialog box opens.

3 Select **OT Impact** from the **KPI** list.
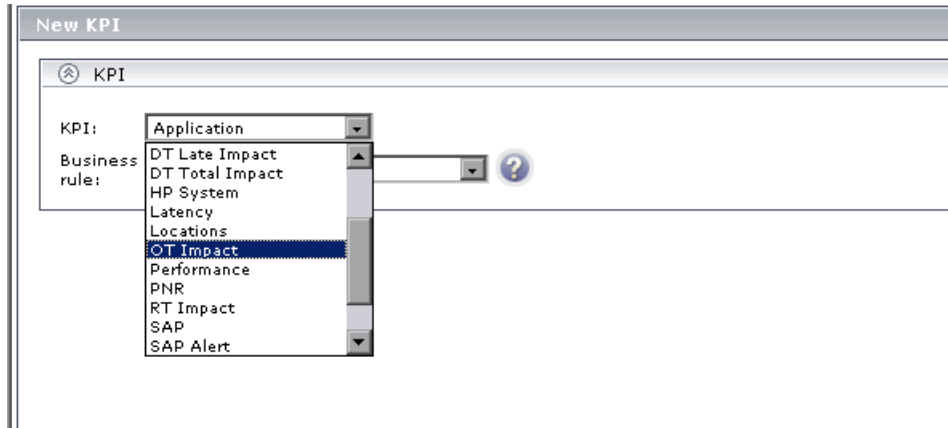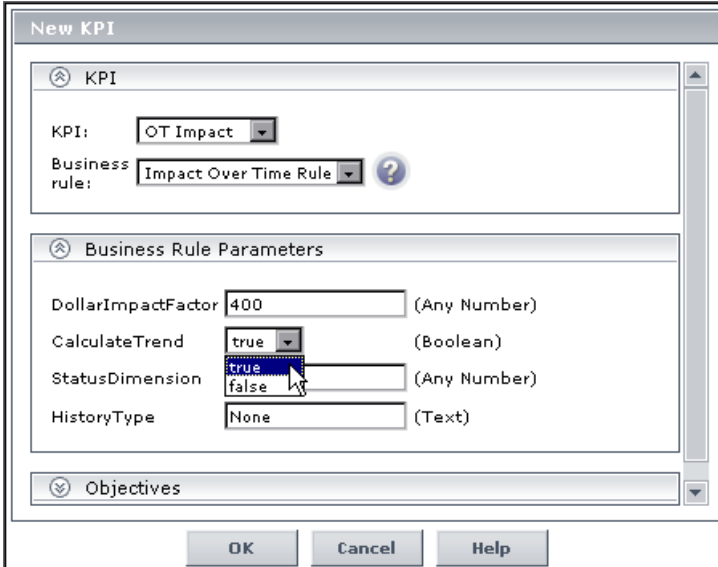
The dialog box is updated to show business rules and objectives for the OT Impact KPI.

4 Select the monitor rule for OT Impact, **Impact Over Time Rule**, from the **Business rule** list.
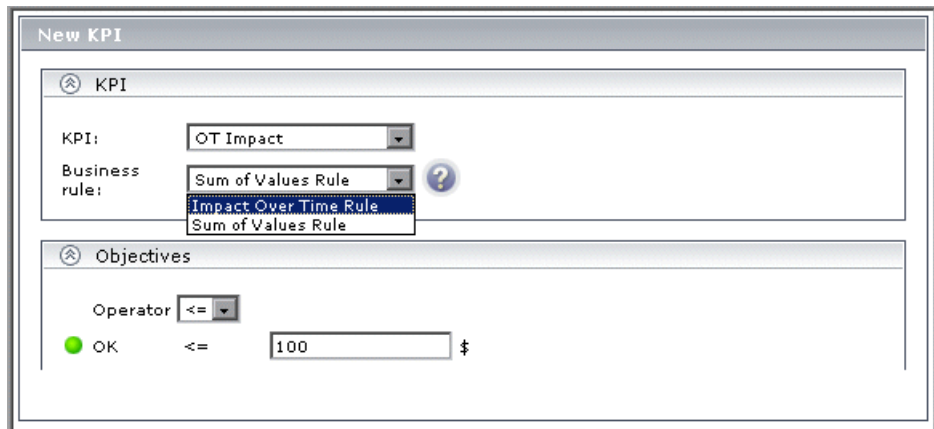
The dialog box is updated to include parameters for the rule.

 5 In the **Business rule parameters** area, modify the parameters as required:

 ➤ In the **DollarImpactFactor** box, enter the dollar financial loss per hour for non-availability of this CI - **100**.

 ➤ In the **CalculateTrend** box, change the value from **false** to **true**, so that trend status will be displayed for the KPI.

Click **OK** to save the changes.

6 The OT Impact KPI is added to the list of KPIs for the CI.

The KPI also automatically propagates up the hierarchy to the top level. For all CIs above the monitor CI, the OT Impact KPI uses the **Sum of Values Rule** by default.

Define Weighted Percentage Rule

For this scenario, A.A. Web Tours considers TCD Corporation to be a key customer, and they want TCD Corporation to have greater impact on the overall status calculations for the Corporate Travel line of business. To achieve this, the Percentage Rule is defined for Corporate Travel KPIs, and the metrics for TCD Corporation are given a heavier weight than the metrics of the other customers attached to Corporate Travel in the view.

To define weighted percentage rule:

1 In the **Admin** > **Dashboard** > **KPIs** tab, select **Corporate Travel** in View Explorer and then click the **Edit** button for the **Availability** KPI.

The Edit KPI dialog box opens.

2 Select **Percentage Rule** from the **Business rule** list.
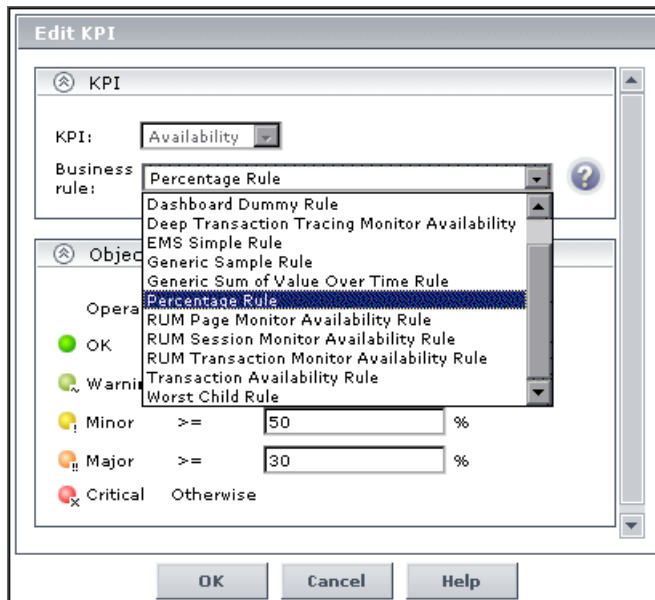
Click **OK**.

3 The new business rule is displayed for the **Availability** KPI in the KPIs list.
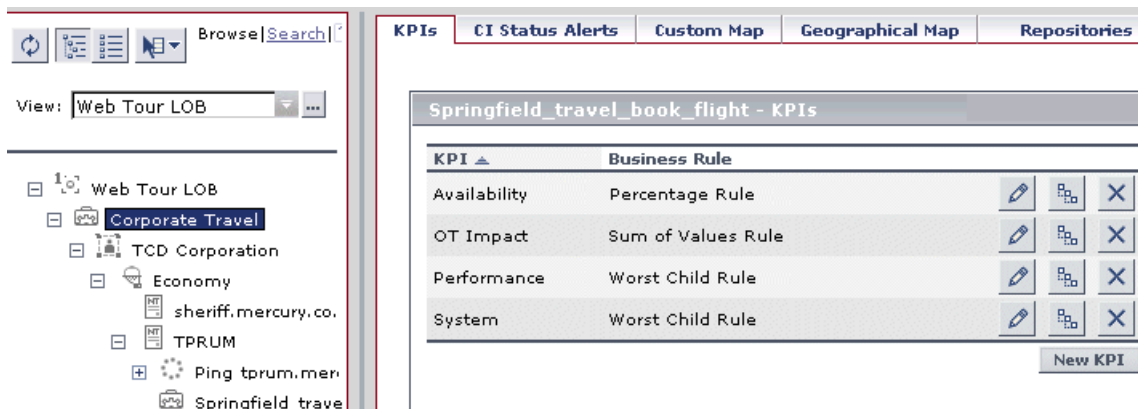
Repeat the procedure for each KPI for **Corporate Travel** that is to use weighted percentage.

4 Move to the **Admin** > **CMDB** > **IT Universe Manager** tab and open the **Web Tours LOB** view in View Explorer. Select the **TCD Corporation** CI in the view.
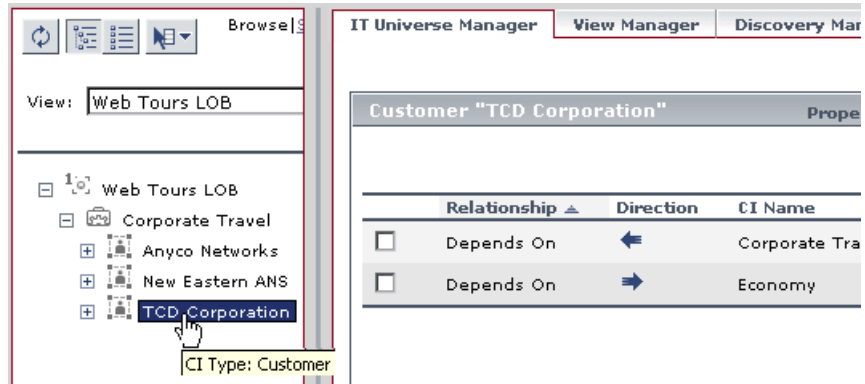
5 In the **Related Configuration Items** tab in the right pane, locate the row in the table for TCD Corporation's relationship with **Corporate Travel**. Click the **Edit relationship** button for the relationship.
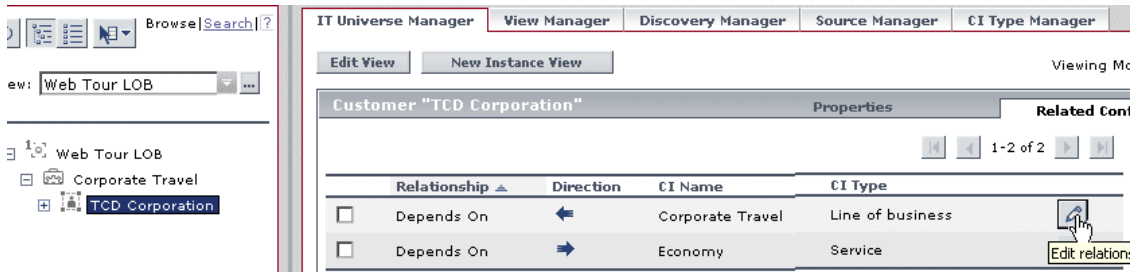
The Relationship Properties dialog box opens.

6 Enter the required weight (for example, **3**) in the **Weight** box.

Click **OK**.

Stage 4: Define Viewing Permissions

The fourth stage of the scenario is assigning the completed view to Dashboard, and defining viewing permissions for the view.

Defining viewing permissions is divided into two sections:

➤ Assign View to Applications, described below

➤ Define User Permissions, described on page 150

Assign View to Applications

Now that the view is ready for viewing in Dashboard and Service Level Management, the setting is changed in View Manager so that it is assigned to those applications.

To assign view to Dashboard and Service Level Management:

1 In the **Admin** > **CMDB** > **View Manager** tab, select the **A.A. Web Tours** folder in the **Folders** pane to display the list of views in the folder. Click the **Edit view properties** button for the **Web Tours LOB** view.

The Edit View Properties opens.

2 Select the check box for **Dashboard & SLM**.

3 Click **OK**.

Define User Permissions

For this scenario, the Web Tours LOB view is set to be available to user Kim for viewing purposes only. This means that Kim can see the view in all applications, but cannot change or delete the view.

To define user permissions for the view:

1 In the **Admin** > **Platform** > **Users and Permissions** > **Permissions Management** tab, select **CMDB** from the **Select context** list (left pane).

2 Expand the **Views** folder in the **Select context** pane, and select **Web Tours LOB**.
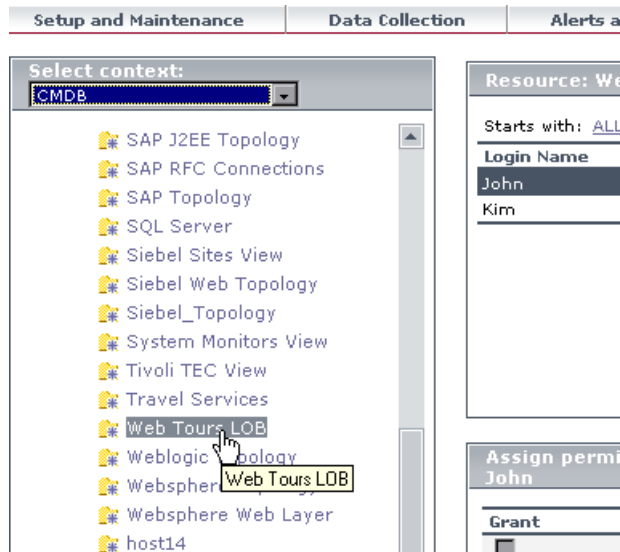
3 In the **Resource** pane (top right), select the user **Kim** from the list of users.
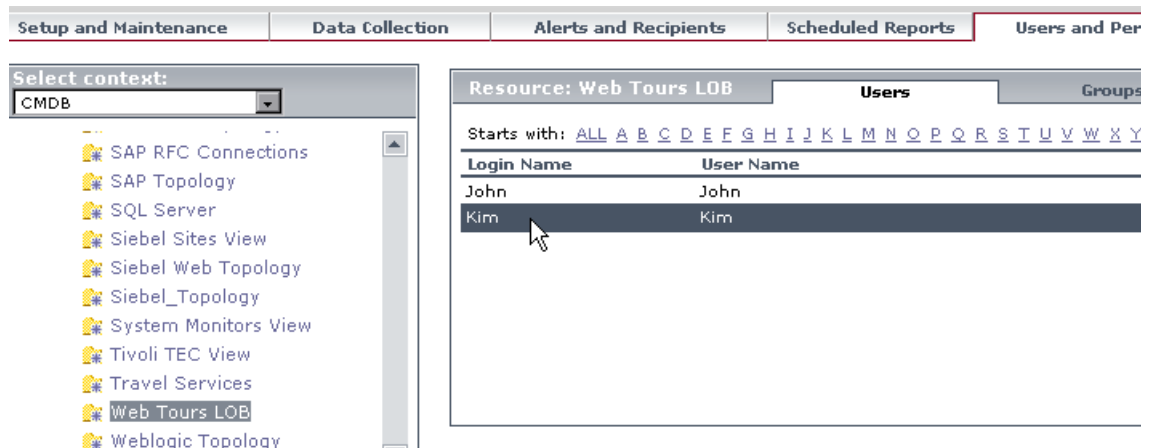
4 In the **Assign permissions to** pane (bottom right), select the **Operations** tab, and then select the **View** check box to assign user Kim permission for viewing only for the Web Tours LOB view.

5 Click **Apply Permissions** to save the permissions settings.

Stage 5: See KPI Status in Dashboard

The final stage of the scenario is using the view in Dashboard to monitor KPI status for A.A. Web Tours' main lines of business.

To view KPI status for the view in Dashboard:

1 Access the **Applications** > **Dashboard** > **Console** tab.

2 Select **Web Tours LOB** from the **View** list in View Explorer.

Index

Index