SERENA®

DIMENSIONS® RM 12.2

Legacy Windows Clients User's Guide

Serena Proprietary and Confidential Information

Trademarks

U.S. Government Rights

Publication date: November 2014

Table of Contents

Preface

This document describes the legacy desktop clients for Serena® Dimensions® RM. Some of these clients, such as RM Word, should no longer be used. Others, such as RM Concept and RM Explorer, should no longer be used for day-to-day operations, but can still be used for the limited set of operations described in this manual.

CAUTION! Performing certain operations with the legacy clients can cause unpredictable and undesirable outcomes! These clients are not fully compatible with current RM projects (thus the term, "legacy clients").

Do **NOT** use any legacy client feature unless it is listed as currently supported in this manual. Just as the legacy clients have not been updated to work with current projects, their functionality has not been limited to what would be safe to use on current projects.

Rather than the legacy clients, RM Browser should be used for most day-to-day operations. RM Browser fully supports the new features and project structures of recent RM releases. See the *Serena Dimensions RM Browser User's Guide*.

For most administration tasks, RM Manage is the tool to use. See the *Serena Dimensions RM Administrator's Guide*.

Objective

The purpose of this book is to describe how to use the features that are still supported in the legacy clients and to guide the user away from clients and client features that should no longer be used.

Audience

This document is intended for members of project teams who use Dimensions RM to create, manage, and track requirements during the lifecycle of a project.

Contacting Serena Technical Support

Serena provides technical support for all registered users of this product, including limited installation support for the first 30 days. If you need support after that time, contact Serena Support at the following URL and follow the instructions:

http://support.serena.com

Language-specific technical support is available during local business hours. For all other hours, technical support is provided in English.

You can use the Serena Support Web page to:

- Report problems and ask questions.

- Obtain up-to-date technical support information, including that shared by our customers via the Web, automatic e-mail notification, newsgroups, and regional user groups.
- Access a knowledge base, which contains how-to information and allows you to search on keywords for technical bulletins.
- Download updates and fix releases for your Serena products.

License and Copyright Information for Third-Party Software

License and copyright information for third-party software included in this release can be found as part of the software download available at:

http://support.serena.com/Download/Default.aspx

Chapter 1

Overview

Tools

The following sections briefly describe the Dimensions RM tools, *both* legacy and current. The Dimensions RM tools are the various interfaces you use to perform specific tasks.

CAUTION! Performing certain operations with the legacy clients can cause unpredictable and undesirable outcomes! The legacy clients are not fully compatible with current RM projects (thus the term, "legacy clients").

Do **NOT** use any legacy client feature unless it is listed as currently supported in this manual. Just as the legacy clients have not been updated to work with current projects, their functionality has not been limited to what would be safe to use on current projects.

RM Browser

RM Browser is the primary user interface. It fully supports current RM projects and is *not* a legacy client. It provides Web access to a core set of Dimensions RM functions. Using RM Browser, both remote and local users can use a Web browser to edit requirements, submit and review change requests, run reports, create filters and scripts, participate in discussions, run polls, create collections, move requirements into different categories, associate collections to Dimensions CM projects, and so on. RM Browser requires no installation.

The following sections describe the four views from which you can work in RM Browser. For detailed information about using RM Browser, see the *Serena Dimensions RM Browser User's Guide*.

Home Page

You can customize your Home page to include up to seven expandable sections. Each section contains the results of a query. Your Home page lets you quickly view and modify requirements that you refer to on a regular basis.

Requirements View

Requirements View is organized according to defined categories, and lets you run filters and scripts, create new queries, and view and modify requirements that are returned.

Documents View

Documents View provides a document-like presentation of requirements, with a table of contents, chapters, and subchapters. Documents View lets you easily add, delete, move, and edit documents, chapters, and requirements, and publish them as Microsoft® Word documents. Word documents that you imported through the RM Import tool are displayed in and can be modified in Documents View.

Note the following:

- Word uses unique, proprietary styles and formatting; therefore, the styles and formatting of the Word document (for example, some lists, bullets, and fonts) are not necessarily preserved in Documents View. However, RM Import always attempts to import the text to prevent data loss.

- You cannot import the changes you make in Documents View back to the original Word document. Roundtripping between Dimensions RM and Word is not supported.
- Documents in Documents View are **NOT** the same as the Word documents from which they were imported. Dimensions RM does not store the Word document; it parses the data into classes (requirements and so on) so the data is available to the entire project.

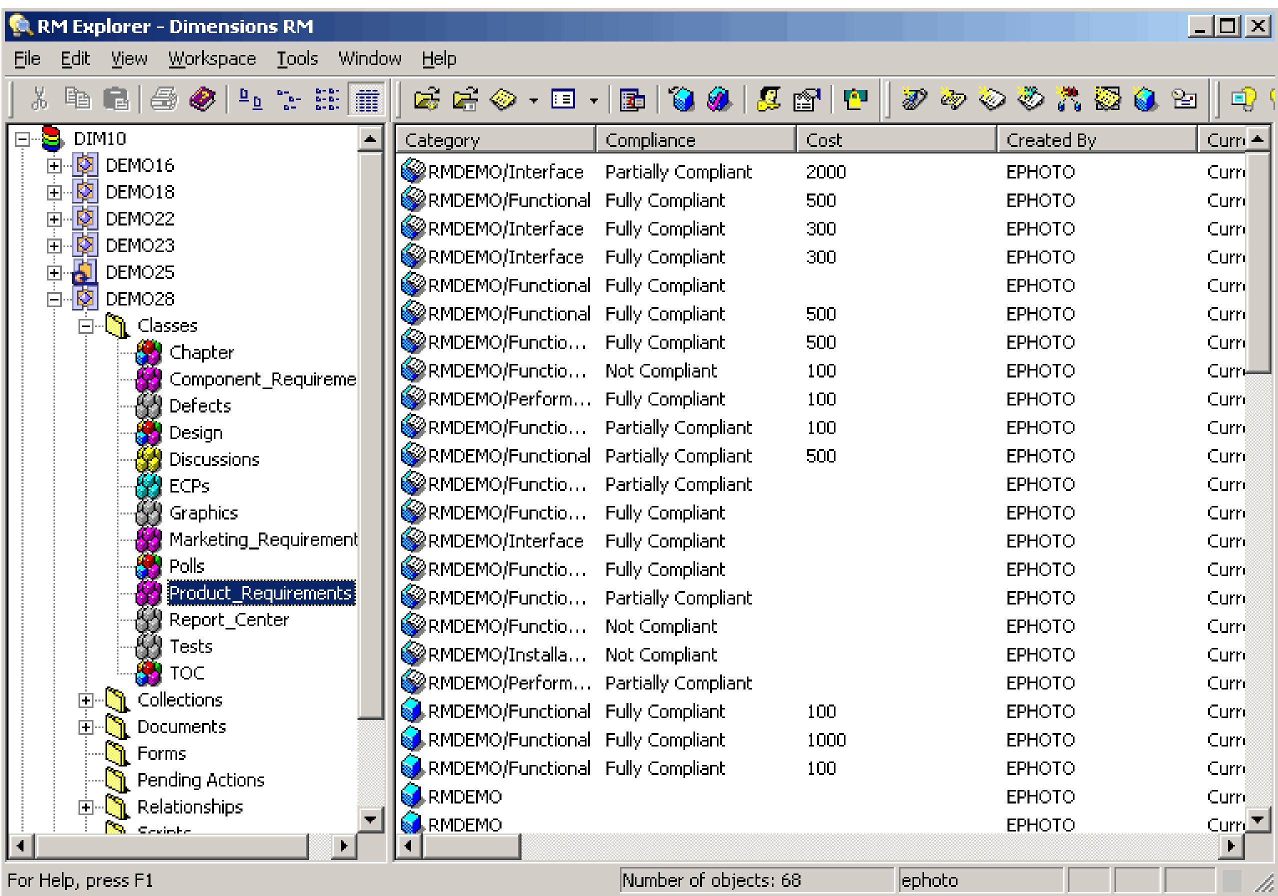
Traceability View

Traceability is a way to analyze the linkages between requirements. Traceability View provides a way to select the relationships you want to trace, browse through the requirements that are part of the relationships, and then print traceability reports that display the information in a visual format that is easy to analyze.

RM Explorer

CAUTION! RM Explorer is a legacy client. Do **NOT** use any legacy client feature unless it is listed as currently supported in this manual.

RM Explorer provides an integrated Windows Explorer-like view of a project, so you can see an entire project at once.



The following RM Explorer features are currently supported:

- AutoLink

- Batch update
- Creating parent containers
- Filter security
- Relationship security
- Script security
- Script renaming

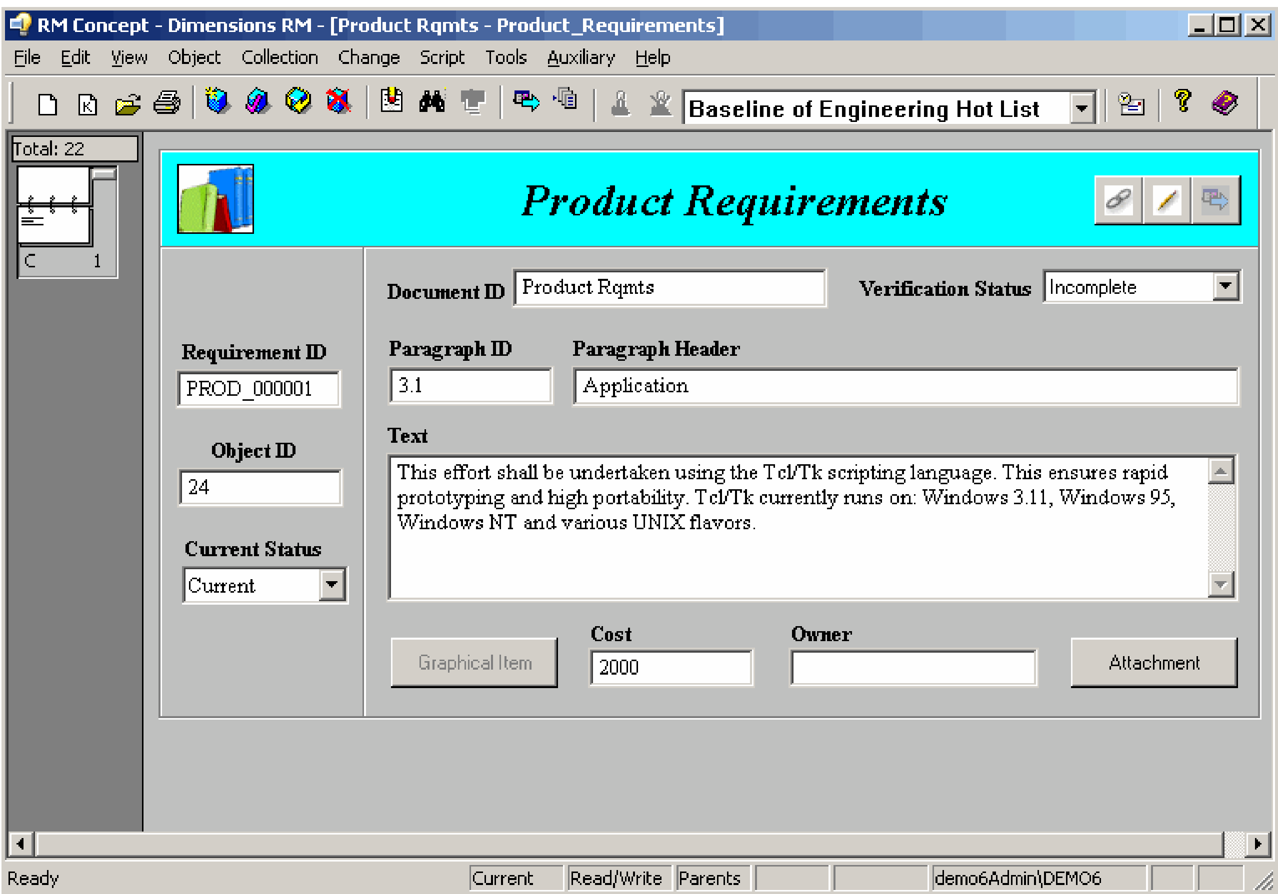
Do not use any other features of RM Explorer. Do not launch any other RM tools from RM Explorer (the Tools menu).

For information about using the supported RM Explorer features, see "RM Explorer" on page 45.

RM Concept

CAUTION! RM Concept is a legacy client. Do **NOT** use any legacy client feature unless it is listed as currently supported in this manual.

RM Concept tracks the requirements engineering and change processes. With RM Concept, you can capture requirements both manually and automatically. You can also customize views for reporting and information gathering.



The following RM Concept features are currently supported:

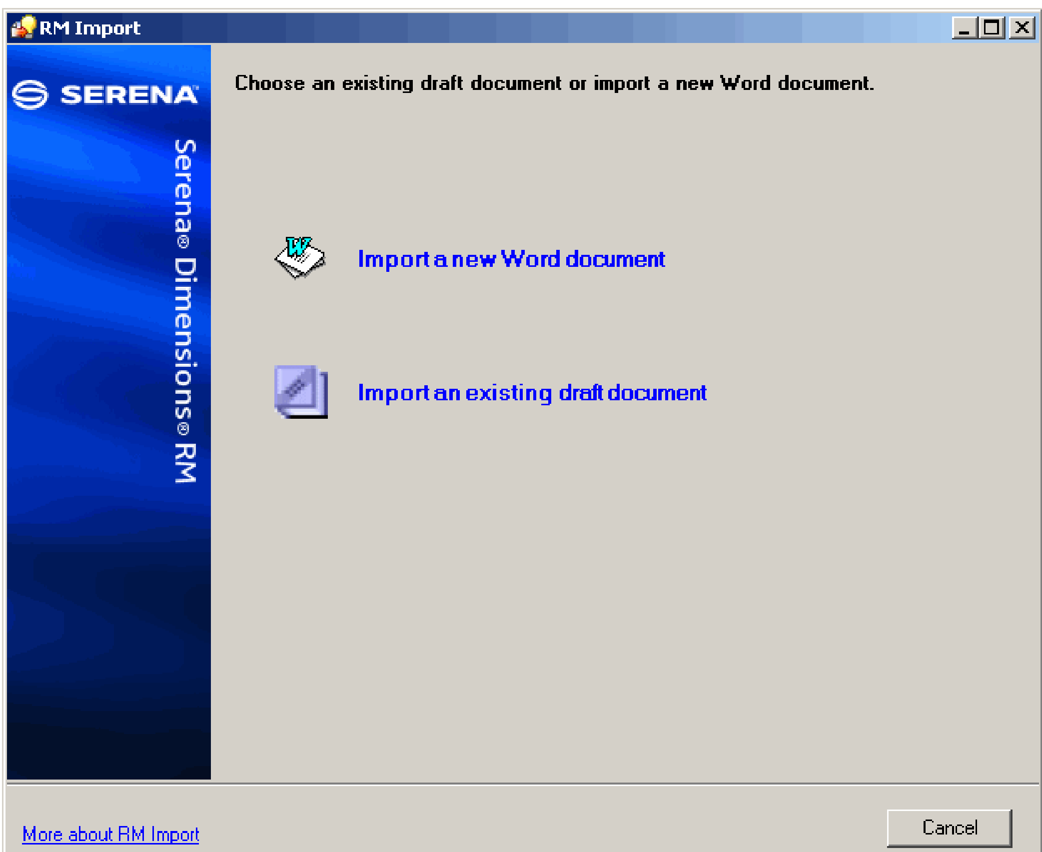
- All script functions
- Administering locks
- Batch update
- CSV import
- Creating parent collection
- Edit parent/child links
- E-mail registration

Do not use any other features of RM Concept. Do not launch any other RM tools from RM Concept (the Tools menu).

For information about using the supported RM Concept features, see "RM Concept" on page 19.

RM Import

RM Import allows users to preview a Microsoft® Word document as a draft document, change the description of chapters, reorganize the chapters, change the values of attributes, move attributes between chapters, and so on. When satisfied with the draft document, users can import the document into Dimensions RM as a Documents View "document."

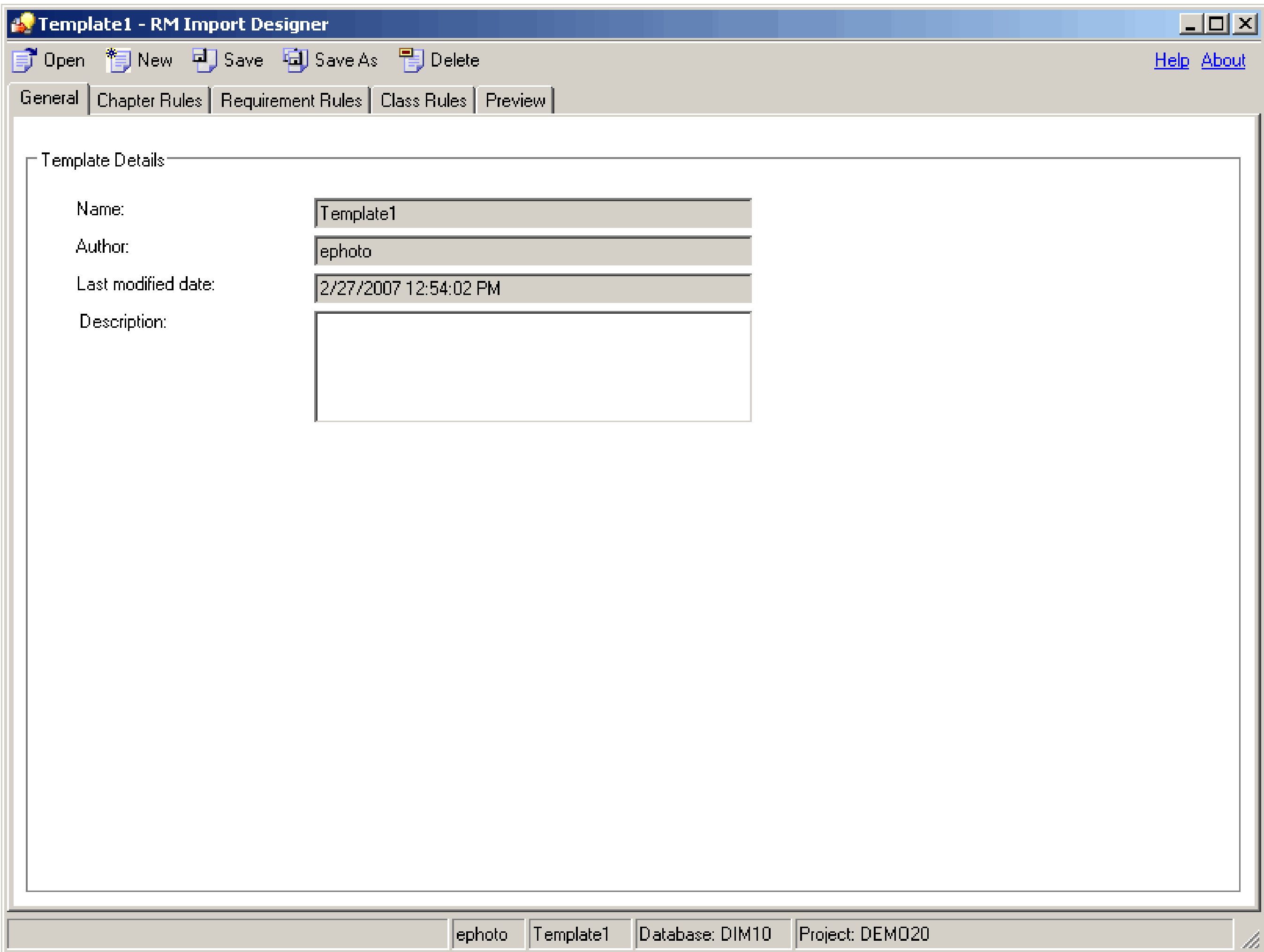


For information about using RM Import, see the *Serena Dimensions RM Browser User's Guide*.

RM Import Designer

RM Import Designer allows administrators to design templates that users select when importing Word documents using RM Import. The import process creates a "document" in Documents View.

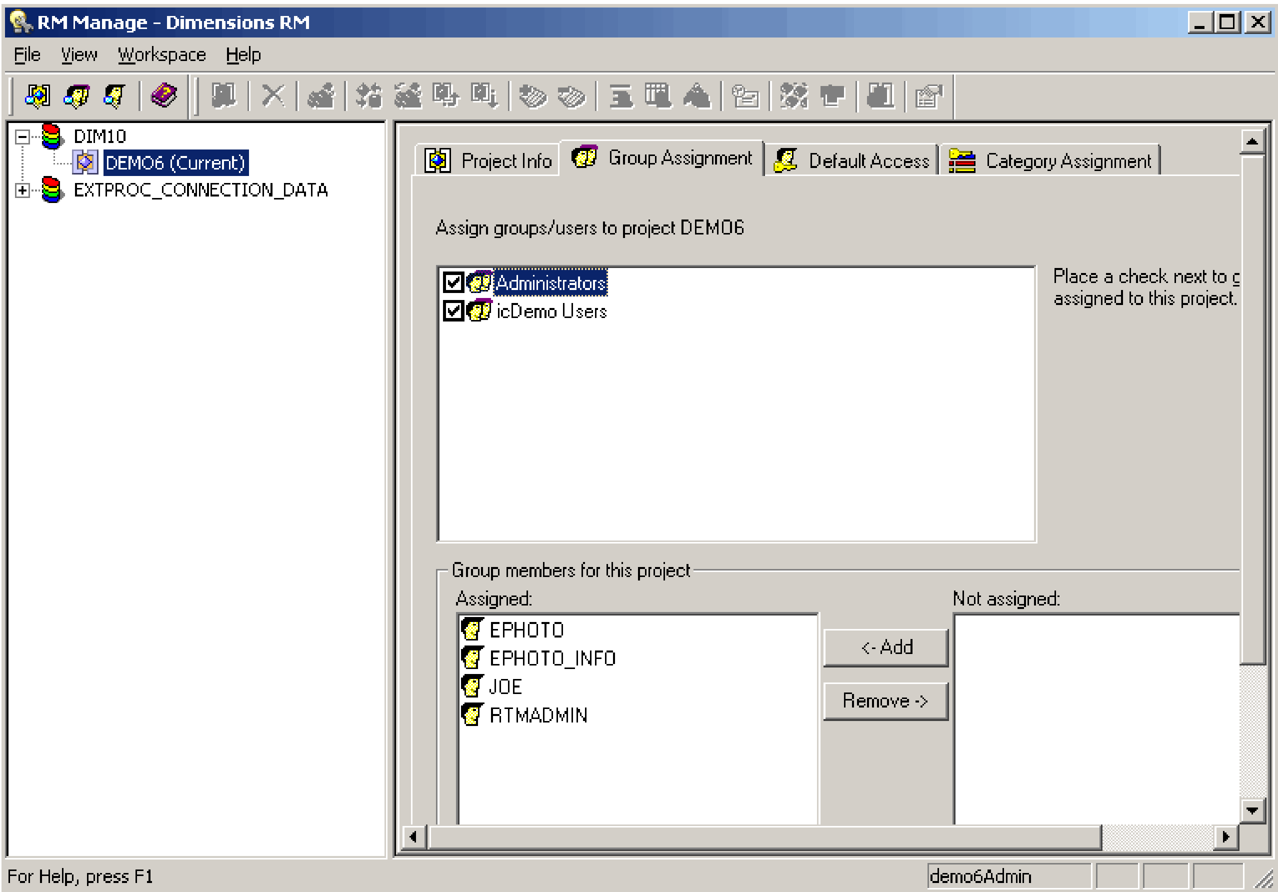
Templates define how to identify classes, attributes, chapters, requirements, and categories. Templates can be saved to the database for future use. Administrators who develop templates must be knowledgeable about class types and the structure of the Word documents that will be imported.



For information about using RM Import Designer, see the *Serena Dimensions RM Administrator's Guide*.

RM Manage

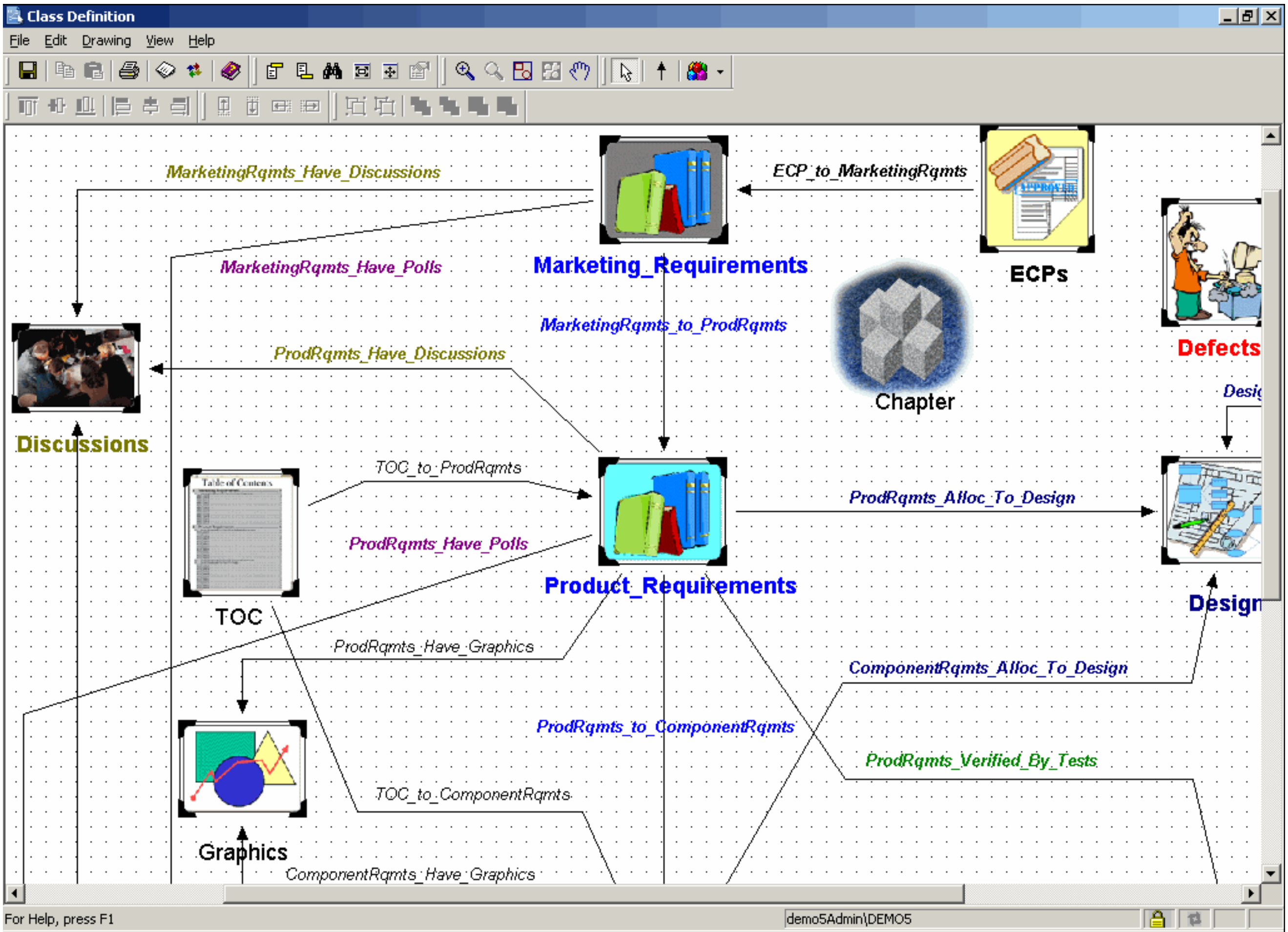
RM Manage allows project administrators to define users and groups, administer project security, configure the project database, organize data, and control user access and data routing.



For information about using RM Manage, see the *Serena Dimensions RM Administrator's Guide*.

Class Definition

Class Definition allows project administrators to set up project schemas in class definition diagrams. This involves defining classes, relationships, attributes, and security.



For information about using Class Definition, see the *Serena Dimensions RM Administrator's Guide*.

Chapter 2

RM Concept

Overview

CAUTION! RM Concept is a legacy client. Do **NOT** use any legacy client feature unless it is listed as currently supported in this manual.

The following RM Concept features are currently supported:

- All script functions
- Administering locks
- Batch update
- CSV import
- Creating parent collection
- Edit parent/child links
- E-mail registration

Do not use any other features of RM Concept. Do not launch any other RM tools from RM Concept (the Tools menu).

RM Concept User Interface

RM Concept provides a graphical user interface that gives you access to Dimensions RM project elements. Inside the main RM Concept window, nearly all data is presented in forms or spreadsheet-like grid views. The menu bar gives you access to most of the operations supported by RM Concept. The toolbar provides quick access to commonly used operations. Right mouse clicks in workspace areas cause RM Concept to display popup menus with view-specific operations.

Visual Status Indicators

Several visual indicators give you additional information about the objects presented on the screen:

- The Rolodex indicates the total number of objects above the “card” and the current “card number” at the lower right. When the Rolodex is “pointing” to an object that has been captured from a document, a “C” appears in the lower left corner of the Rolodex.
- When the object you’re viewing has links, comments, or pending change requests visual indicators may be included in both form and grid views. These indicators must be explicitly added to the form layout or grid column display selections.
- The status bar at the bottom of the window provides information about the currently selected object. The status bar is displayed by default. To hide it, uncheck **Status Bar** on the View menu. To display it, check **Status Bar** on the **View** menu.

From left to right, the status indicators display:

- Tool messages about most recent operation

- Object status information
- Loaded objects editing status
- Whether the loaded object has ancestor objects
- Whether the loaded object has descendant objects
- Which pane is active when in split screen mode
- Username\project for the current RM Concept session
- Caps Lock key indicator
- Numeric Lock key indicator

Menus

- The **File** menu provides operations that control interactions with the tool at a gross level. These operations include **Open** to allow you to choose the kinds of information that you will be working with, new form creation, layout operations to control grid displays, as well as the CSV import and batch update Wizards.
- The **Edit** menu offers the standard cut/copy/paste operations as well as object (un)locking for edit, and e-mail registration.
- The **View** menu provides operations that control the workspace appearance and its modes of object manipulation.
- The **Object** menu: Only the link editing features are supported in this menu.
- The **Collection** menu offers operations on collections.
- The **Change** menu: **No items in this menu are currently supported.**
- The **Script** menu offers creation, editing, and execution operations for scripts.
- The **Tools** menu: **Do NOT use the Tools menu.** Launching other RM clients from RM Concept is no longer supported.
- The **Help** menu includes the about menu and a link to the technical support web site. Help is no longer updated or provided for this tool.

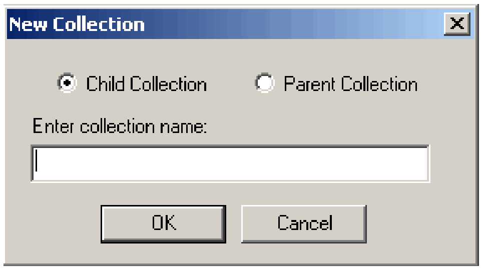
Creating Collections

There are two types of collections: parent and child. Parent collections are used to associate multiple child collections. Objects can only be linked to child collections. Use parent collections to create hierarchies of collections to make searches easier. Objects linked to child collections of a parent are displayed when the parent collection is opened in RM Concept.

To create a new collection:

1. Select **New** > **Collection** from the **File** menu or click the **New Collection** button .

2 The **New Collection** dialog opens.



3 Select **Child Collection** or **Parent Collection**.

4 Enter a name for the new collection

5 Click **OK** to create the collection.

6 For parent collections, the **Select Collection** dialog opens.

7 Select the collections to be associated with your parent collection. You must connect at least one child to the parent collection. As you select each collection, your choice is highlighted.

NOTE You can add other children to the parent collection later as described in "Viewing and Editing Parent/Child Links for Collections" on page 22.

8 Click **OK** to create the parent collection and link it to its child collections.

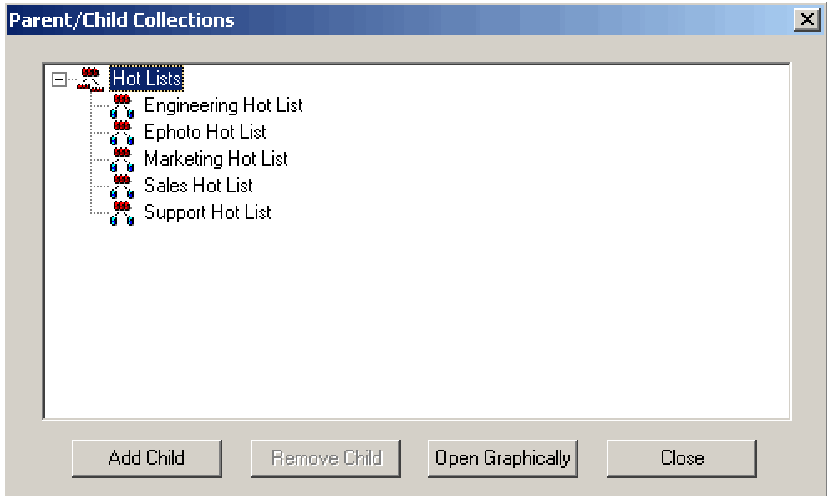
Viewing and Editing Parent/Child Links for Collections

Parent/child collection hierarchies give you a flexible mechanism to organize objects.

To view the child links for parent collections:

1 Select **View/Edit Parent-Child Links** from the **Collection** menu.

The **Parent/Child Collections** dialog opens.



NOTE If there is a recursive parent-to-parent collection relationship in the tree, the children for the parent collection will be displayed the first time it appears for a given branch, but will not be displayed again when the parent collection is displayed deeper in the tree.

2 Click the **+** next to the parent collection to display the child collections for the selected parent collection. The child collections for a selected parent can be a mix of other parent and child collections.

3 To add a new child to the selected parent, click **Add Child**, select the desired collections to be added from the displayed list, and click **OK** in the list dialog. The collections will be added as children to the selected parent and will be added to the tree anywhere the selected parent may appear in the tree.

4 To remove a child collection from a parent collection, select a child and click **Remove Child**. This will remove the parent-child linkage.

Working with Scripts

Scripts are used in RM Concept to retrieve data from the database and format the view of the data retrieved. The data are originally displayed in a spreadsheet, but can be saved to multiple formats that can then be loaded in the format's native application (for example, Microsoft® Word or a Web browser).

The script generator wizard can create scripts for most types of reports. There are elements of the script language syntax that cannot be generated by the wizard. For example, the script wizard does not handle prompting, script variables, script comments, or simple calculation operations. Scripts requiring these features must be manually edited.

Scripts that you can create in the wizard retrieve objects of a single class, and objects participating in specific relationships. You can also combine these things in a single script.

NOTE

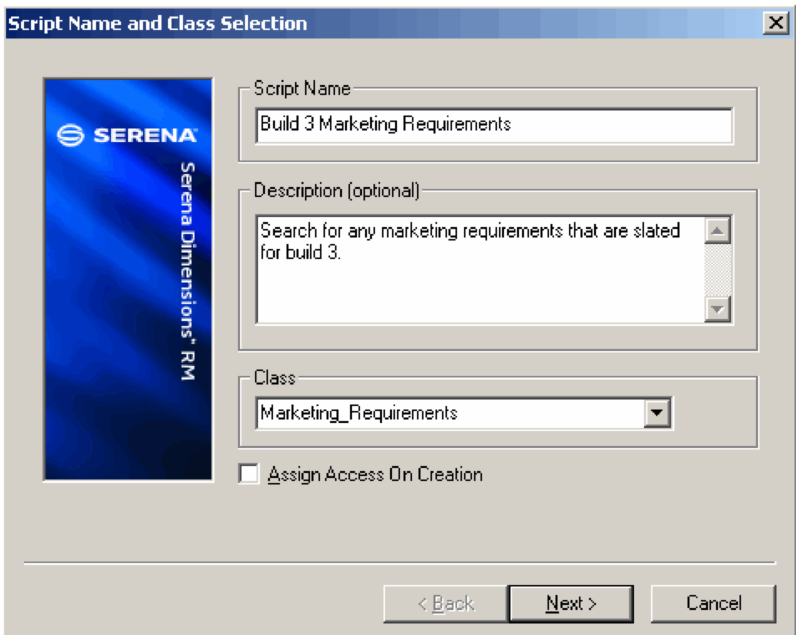
- Traceability reports are presented as scripts in RM Concept.
- Filters are scripts that include only one class. RM Concept does not support filters that contain runtime values (prompting).

Creating a Script

NOTE If you do not have "Create" permission for scripts, you can create a script; however, you cannot save it.

To create a new script:

1 Select **Create** from the **Script** menu. This starts the Script Generator wizard.

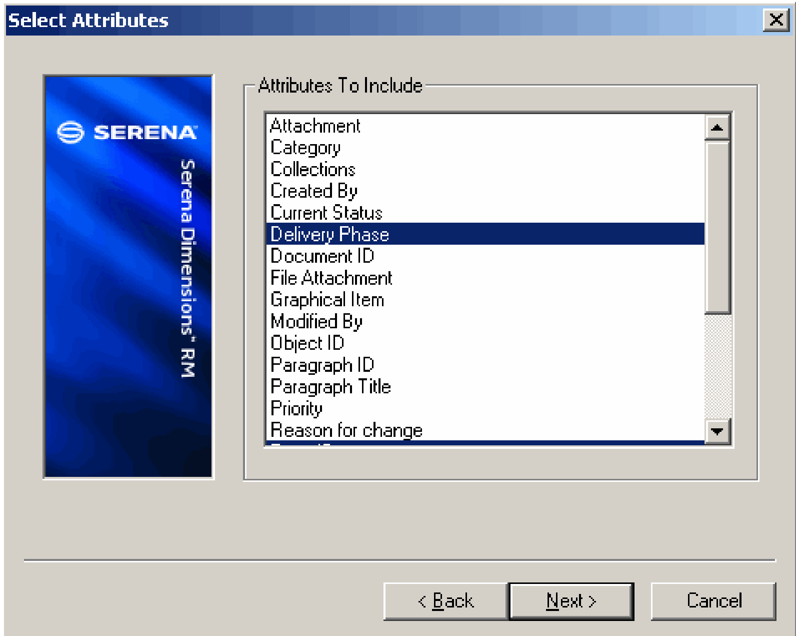


2 In the **Script Name and Class Selection** dialog, enter the script name, a description, and select the first class of interest in the script.

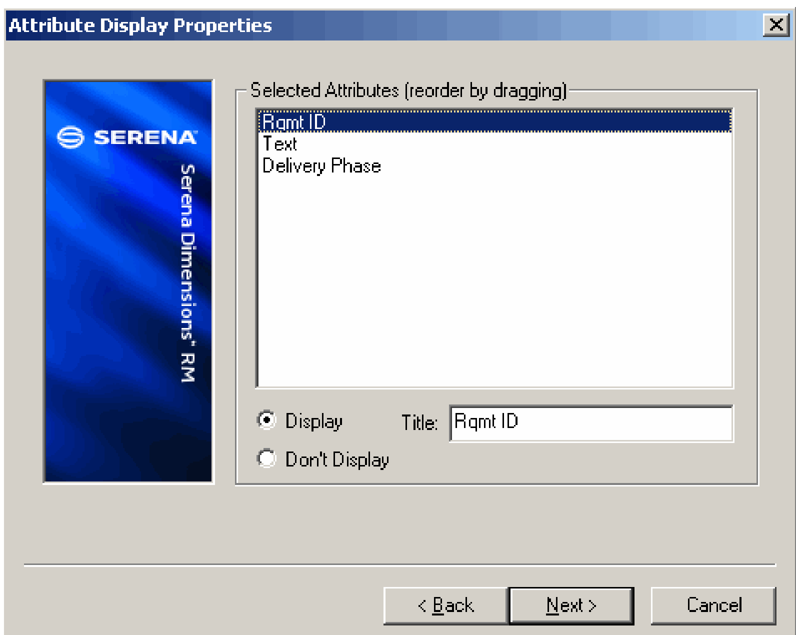
3 Click **Next** to continue.

4 The next step is to select the attributes to be retrieved for each object to be selected by the script. The **Select Attributes** dialog offers the display names (as defined in

Class Definition) of each attribute. Select each desired attribute by left clicking on its display name in the list.

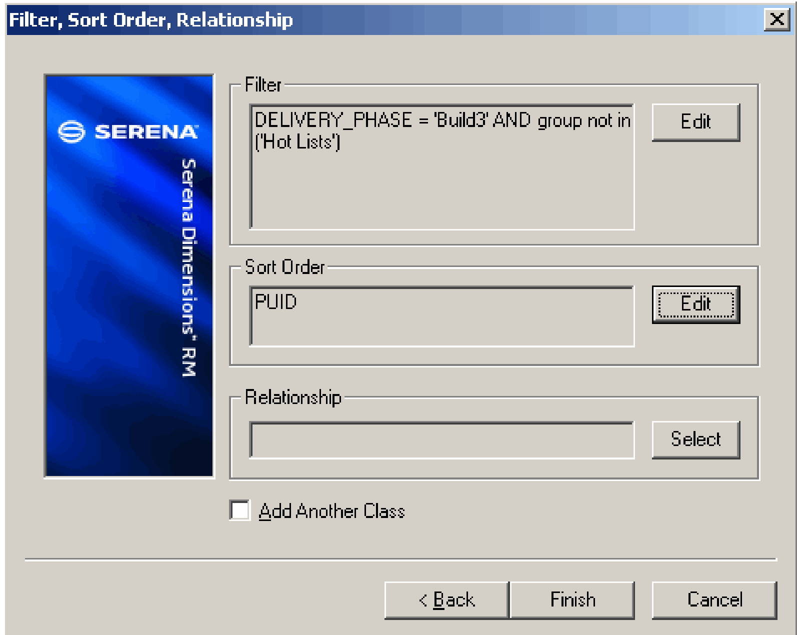


5 After you have selected the attributes, use the **Display Properties** dialog to specify the order in which your selections are to be reported in the script output. The attribute display names as listed in the **Display Properties** page are used as column headings in the script output. You can rearrange them by selecting individual names and dragging them up or down to new positions. In addition, you have the options of changing the column heading to be used or disabling the display of each attribute.



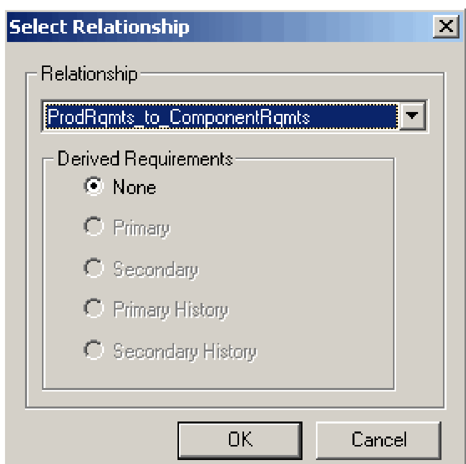
NOTE The display names also translate to document style names for script output formats such as RTF. Using this feature, you can generate reports as Microsoft Word documents with all occurrences of particular attributes already in a specific style.

6 The final dialog in the wizard, **Filter, Sort Order, Relationship** allows you to specify filtering and sorting options, cross references to related classes, and to add more classes to your report.



- Specify the filtering options just as you specify filters for complex find operations.
- Specify the sorting options as you would for simple finds.

7 To trace a relationship from your first class selection to another class, click the **Select** button at the right of the **Relationship** box. The **Select Relationship** dialog includes only those relationships that have been defined for the class that you selected in the first wizard dialog (**Script Name and Class Selection**).



After you select a relationship in the **Relationship** drop-down list, click **OK**, and then click **Next** in the wizard, the wizard returns to the **Select Attributes** dialog. This dialog lets you select the attributes to be displayed for the linked objects of the related class.

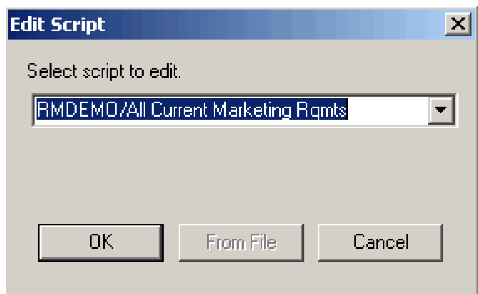
8 To add another class to your report, select the **Add Another Class** check box at the bottom of the **Edit, Sort Order, Relationship** dialog. You can use this mechanism to build scripts that concatenate otherwise separate reports into a single report.

NOTE You can also create a script by loading it and making changes directly to the script in the edit box where the script is displayed. Select **Save As** from the **Script** menu. This is recommended only for advanced users because the script syntax is not checked until the script is run, so errors can be made in scripts that are edited by this method.

Editing a Script

To edit a script:

1 Select **Edit** from the **Script** menu. The **Edit Script** dialog opens.



2 From the drop-down list, select the script to be edited.

CAUTION! If you edit and save a script that is a traceability report, the traceability report becomes a regular script.

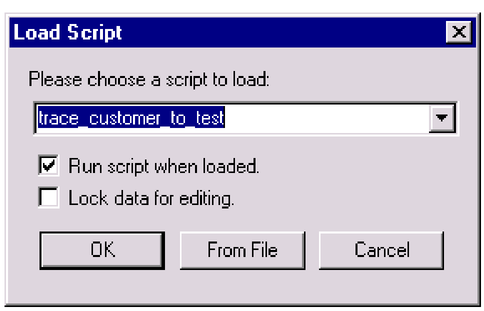
3 Click **OK**. This starts the Script Generator wizard for editing the selected script.

NOTE You can also edit a script by loading it and making changes directly to the script in the edit box where the script is displayed. However, you should do this only if you are very familiar with the scripting syntax. This is recommended only for advanced users because the script's syntax is not checked until the script is run, so errors can be introduced in scripts that are edited by this method.

Loading a Script

To load a script:

1 Select **Load** from the **Script** menu. The **Load Script** dialog opens.



2 Select the script to be loaded from the drop-down list.

3 Click **OK**.

- If the **Run script when loaded** box is checked, the script will be run when it is loaded into the script view in RM Concept.
- If the **Lock data for editing** box is also checked, the data will be loaded into the grid with each record being locked. This will lock other users from editing any of the data returned by the script, so select this option carefully. The script will be displayed in the edit box at the top of the view and the results of the script will be loaded into a grid.

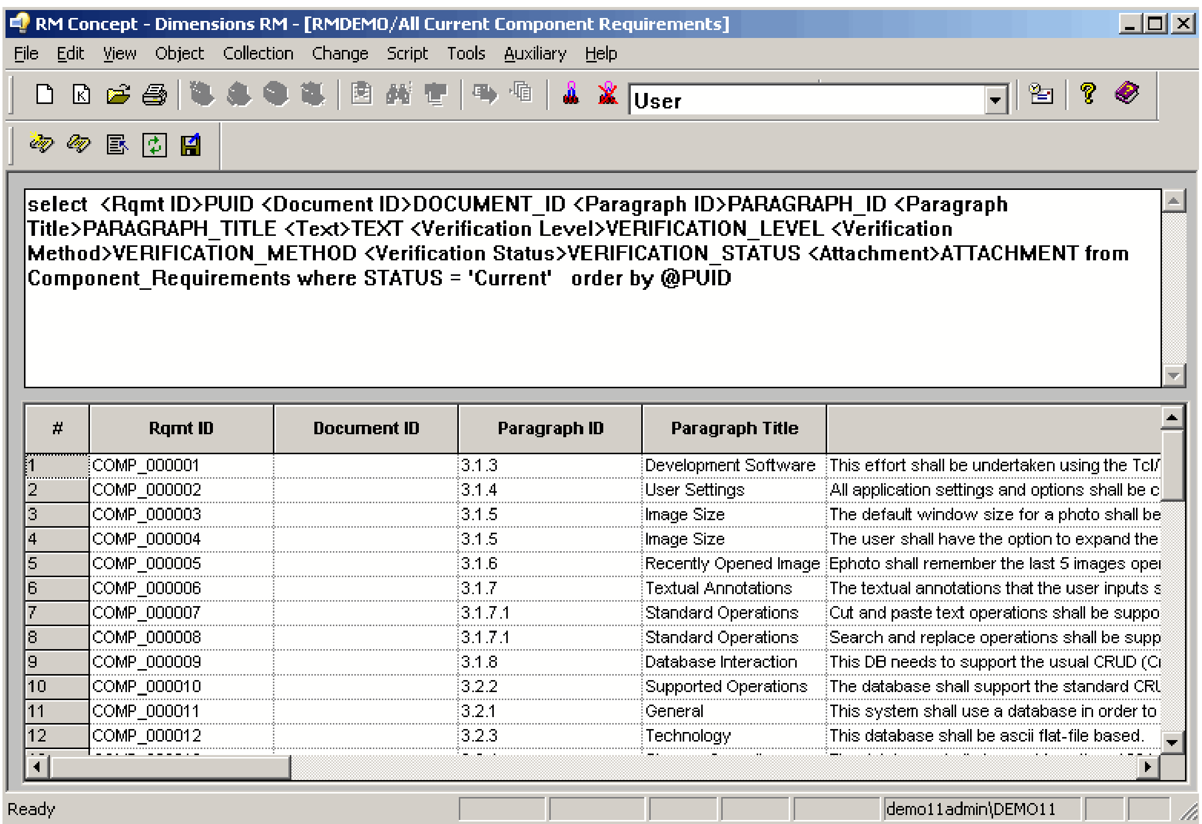
Running a Script

If the **Run script when loaded** check box on the **Load Script** dialog is unchecked, the script is loaded into the script view's edit box, but the script will not be run.

To run a script:

- Select **Run** from the **Script** menu or click the run script button .

  This runs the script that is currently in the edit box of the script view. The results will be loaded into the grid. This option will also be used to run the script if the script has been modified directly in the edit box.



Saving a Script

NOTE If you do not have "Create" permission for scripts, you can create a script; however, you cannot save it.

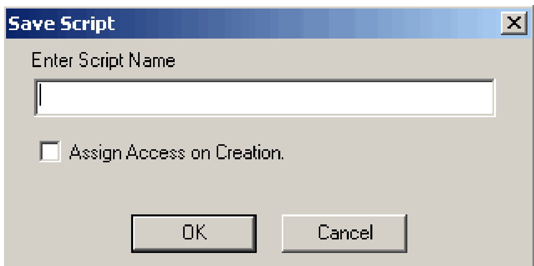
To save a script that has been modified directly in the edit box of the script view:

- Select **Save** from the **Script** menu or click the save script button . This saves any modifications to the loaded script in the edit box.

To save a script with a different name

1 Select **Save As** from the **Script** menu.

The **Save Script** dialog opens.



2 Enter the new script name and click **OK**.

CAUTION! If you edit and save a script that is a traceability report, the traceability report becomes a regular script.

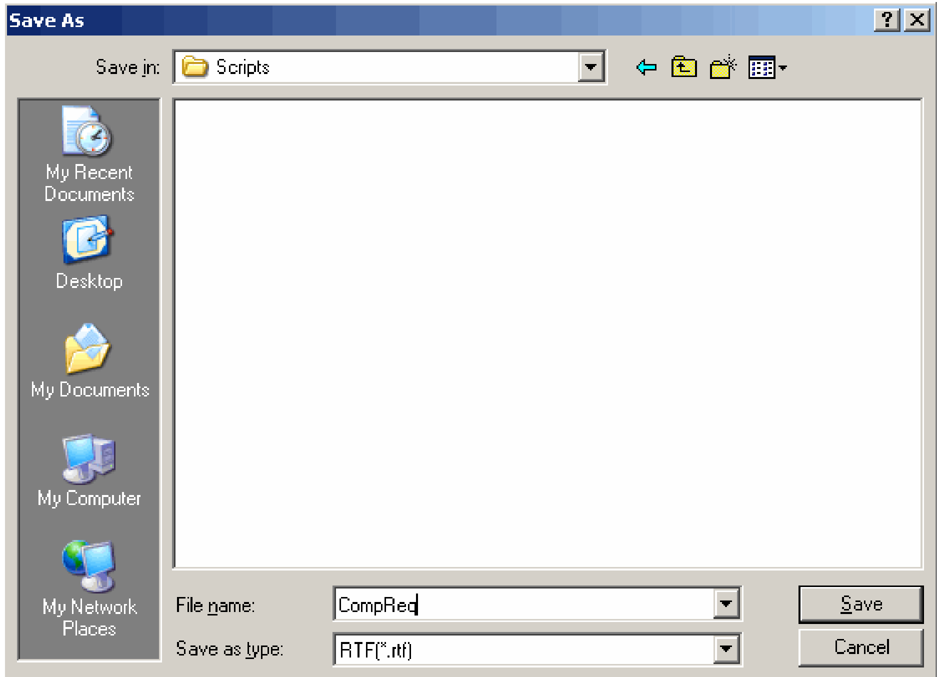
Saving a Script's Output to a Different Format

After a script has been loaded, you can run the script and save the data retrieved by the script. The results will be stored in the file system and not displayed in the RM Concept workspace.

To save the output of a script to a different format:

1 Select **Save As** from the **File** menu.

The **Save As** dialog opens.



2 Select a filename, a file location, and a format.

> **NOTE** The following format types are available: ASCII Table, Comma Separated Value (CSV), No Markup Type, RTF, RTF Table, SGML, SGML Table, HTML and HTML Table. The script text itself can be saved using the rmdoc format. When you save the script text, the script is not run.

3 Click **OK**.

CSV Import

The CSV (comma separated value) Import utility is designed to allow you to import data from a comma separated value file into the Dimensions RM project database. The data can be imported into either a class or relationship for the selected project. For example, you may have an Excel spreadsheet that you want to export to Dimensions RM. You can save the spreadsheet in .csv format and bring it into Dimensions RM through the CSV Import wizard.

> **CAUTION!** Dimensions RM generates and requires upon import a comma-quote delimited file. If the generated .csv file is modified in Microsoft® Excel®, Excel saves the file in comma-delimited format (without quotes). The CSV Import wizard will not accept such a file and will report that there are errors in the format of the file.

CSV Operations

When importing into a class the following operations are available:

- Create a new object for each row in the file.
- Update matching objects in the project database.
- Update matching objects in the project database—create a new object if no match is found.
- Replace matching objects in the project database.
- Replace matching objects in the project database—create a new object if no match is found.
- Delete matching objects in the project database.
- Undelete matching objects in the project database.
- Remove matching objects in the project database.

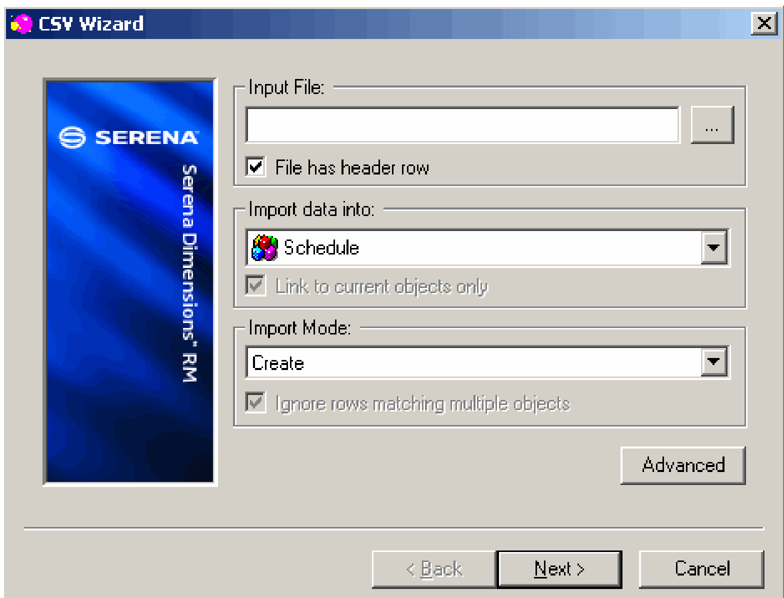
When importing into a relationship the following operations are available:

- Create a new link object for each row in the file.
- Break or unlink matching link objects in the project.

Creating Objects from a CSV File

To create new objects in the project database:

1 Select **Import CSV Data** from the **File** menu. The CSV wizard starts.



2 Type the path of the input file or click **Browse** to select an input file.

The input file is usually a Comma Separated File (`.csv`), which is the most common format you will import. Most spreadsheet applications have the capability to output the data in this format. CSV files use commas to separate data in cells; for example, red, green and blue in the cell is formatted as **red,green,blue** in a text string.

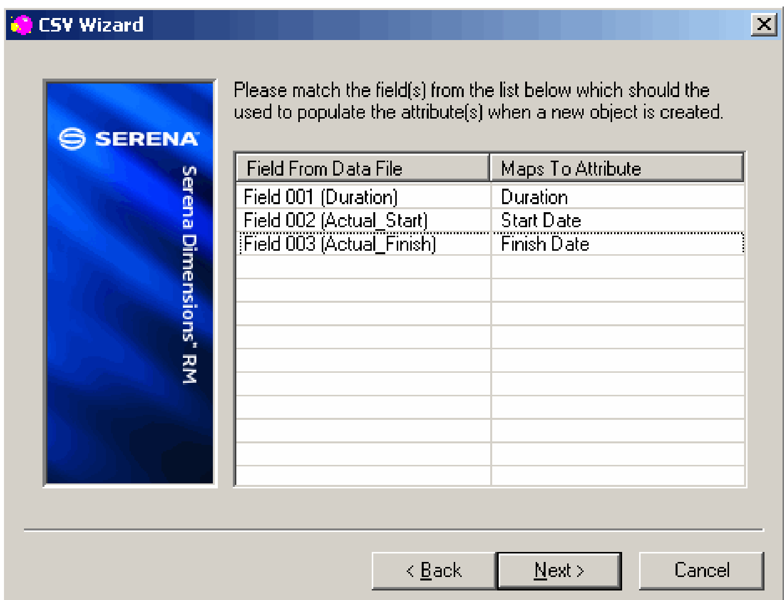
You also can use other formats for importing files by clicking the **Advanced** button. For example, you can save the file as a "tab delimited" file, which uses tabs to separate data instead of commas. These files have a `.txt` extension. The CSV import wizard allows you to specify which delimiter is used.

3 Select the class into which the data is to be imported.

4 Select the operation to be performed on the data in the import file from the drop-down menu. The following options are available:

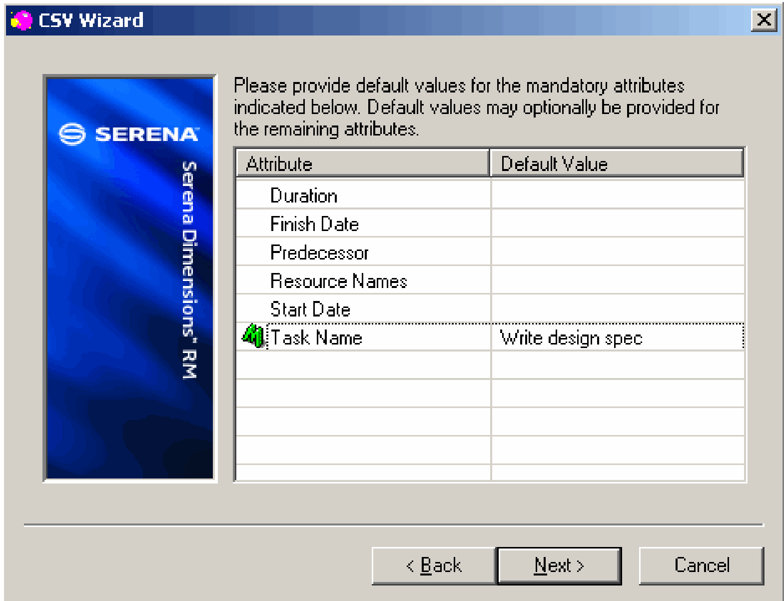
- **Create**—Create a new object for each row in the file.
- **Update Matching Records Only**—Update matching objects in the project database.
- **Update/Create When no Match Found**—Update matching objects in the project database, or create a new object if no match found.
- **Replace Matching Records Only**—Replace matching objects in the project database.
- **Replace/Create When no Match Found**—Replace matching objects in the project database or create a new object if no match found.
- **Delete**—Delete matching objects in the project database.
- **Undelete**—Undelete matching objects in the project database.
- **Remove**—Remove matching objects in the project database.

5 Click **Next**. The attribute constraint dialog opens.



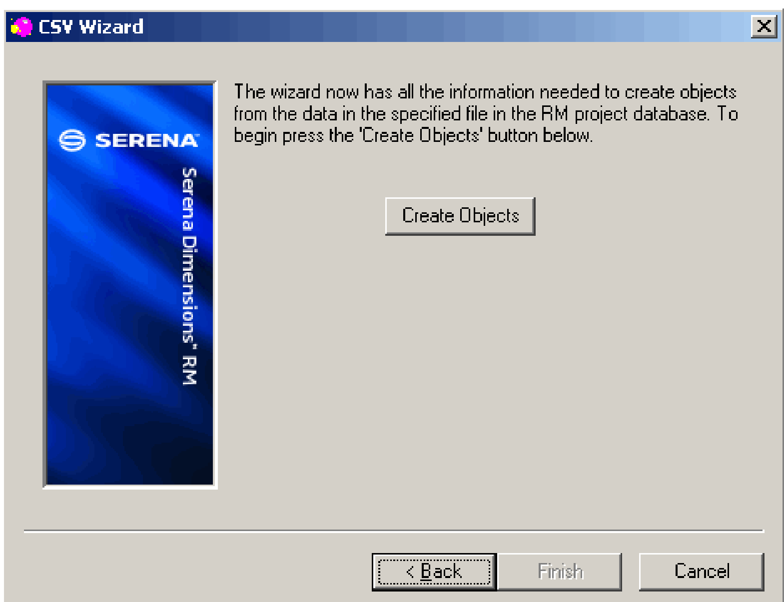
The attribute constraint dialog maps the fields from the CSV file to the attributes of the object. The left column contains an entry for each field read from the input file. The right column displays the attribute that should be constrained with the value at that position in each row of the file. To specify which attribute should be constrained by a given field in the file, click the **Maps To Attribute** cell that is opposite the field and select the attribute from the drop-down menu. Note that an attribute can only be mapped to a single field in the file. Any fields that are not be used as search constraints should be left blank.

6 Click **Next**. The default value assignment dialog opens.



The default value assignment dialog is used to assign default values to each of the attributes for the object. If an attribute is defined as mandatory, indicated by the symbol next to the attribute name, a default value must be provided so that the object can still be created, even if a value is not present in the file. You may optionally provide default values for all other attributes that will be used if the field in the file assigned to the object contains no value.

7 Click **Next**. The perform import dialog opens.



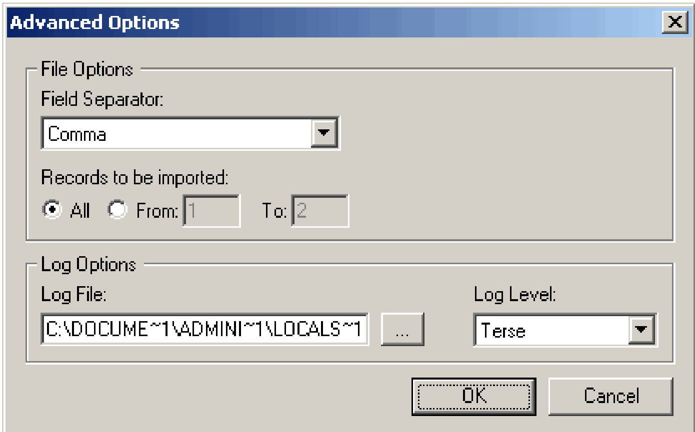
The wizard now has all the information it needs to read the data from the CSV file and modify the objects identified by each row in the file.

8 Click **Create Objects** to begin the import.

Once the import has completed, you will have the option to view the log information that displays the details of the objects that have been created. Click **Finish** to close the import wizard.

CSV Import Advanced Options

To display the **Advanced Options** dialog, click the **Advanced** button on the first page of the CSV Import wizard.



The **Advanced Options** dialog allows the following configuration parameters to be set for the CSV Import wizard:

- **Field Separator**—Select the character to be used as the field separator in the input file from the list of options available on the drop-down menu.
- **Records to be Imported**—In the **From** and **To** fields, specify whether all the rows from the file are to be imported, or a selected row.
- **Log File**—Specify the log file.
- **Log Level**—Configure the log output. The default value, **Terse** provides only a minimal description of the objects changed, while **Verbose** displays all attribute values for each object modified. Note that when using **Verbose**, the log file can grow to be very large.

Batch Update Wizard

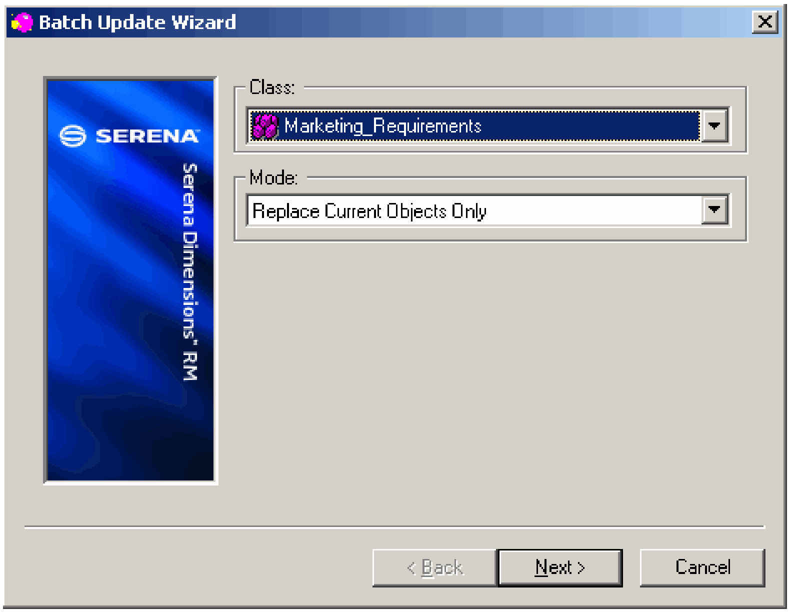
The Batch Update Wizard is designed to allow attribute values to be changed for all objects that match a set of attribute constraints for any one class defined in the project schema. When running the Batch Update Wizard the following operations are available:

- **Update Current Objects Only—**Updates all current objects that match the specified attribute constraints to the new attribute values provided.
- **Update All Objects—**Updates all objects that match the specified attribute constraints regardless of status with the new attribute values provided. You can update all objects only if this privilege has been given to you by the system administrator.

- **Replace Current Objects Only—**Replaces all current objects that match the specified attribute constraints and updates the attribute values to the new attribute values provided.
- **Delete Objects—**Deletes all current objects that match the specified attribute constraints.
- **Undelete Objects—**Undeletes all deleted objects that match the specified attribute constraints.
- **Remove Objects—**Removes all current objects that match the specified attribute constraints. This option destroys the matching objects permanently and cannot be undone.

To update or replace objects in the project database:

1 Select **Batch Update Data** from the **File** menu. The Batch Update wizard opens.



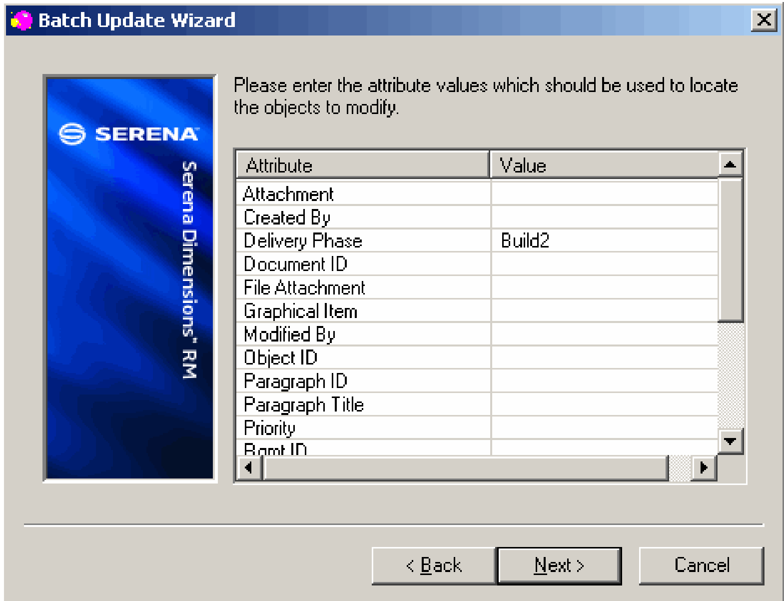
2 Select the class that contains the objects you want to modify from the **Class** list.

3 Select the operation to be performed on the objects from the **Mode** list.

The following options are available:

- **Update Current Objects Only**—Updates all current objects that match the specified attribute constraints with the new attribute values provided.
- **Update All Objects**—Updates all objects that match the specified attribute constraints regardless of status with the new attribute values provided.
- **Replace Current Objects Only**—Replaces all current objects that match the specified attribute constraints and with the new attribute values provided.
- **Delete Objects**—Deletes all current objects that match the specified attribute constraints.
- **Undelete Objects**—Undeletes all deleted objects that match the specified attribute constraints.
- **Remove Objects**—Removes all current objects that match the specified attribute constraints. This destroys the matching objects permanently and cannot be undone.

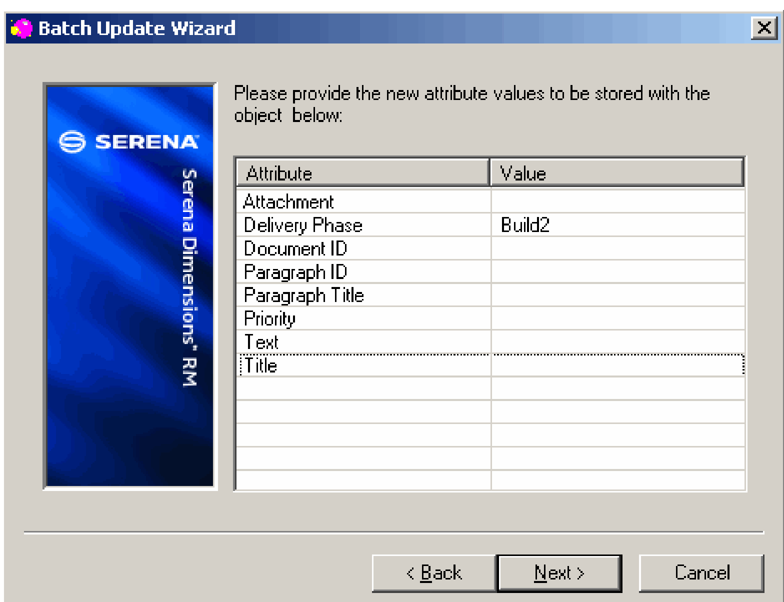
4 Click **Next**. The attribute constraint dialog opens.



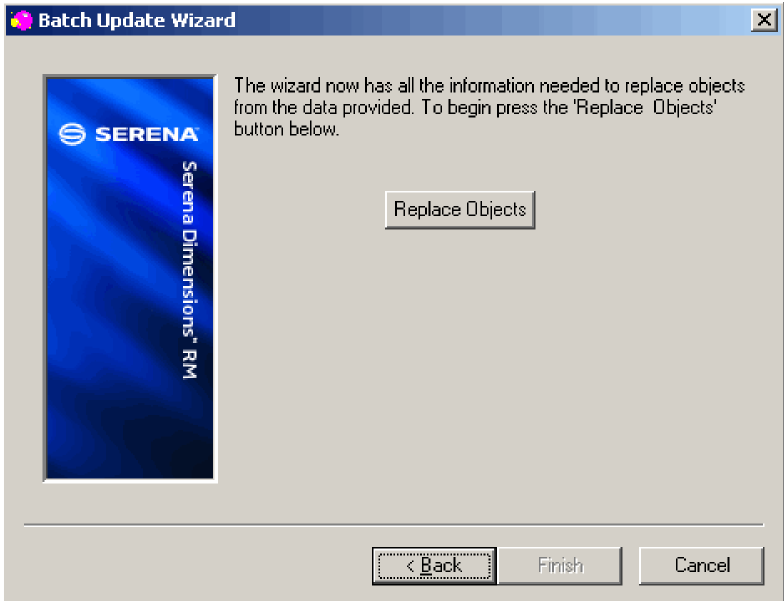
Use the attribute constraint dialog to provide values for the attributes used as constraints when searching for objects. Only certain attribute types can be constrained (for example, text in the Description attribute cannot be constrained). The left column contains an entry for each attribute defined for the class. The right column allows the constraint value to be defined for the attribute.

5 To add a constraint to an attribute, click the **Value** cell next to the attribute and enter the appropriate value. Any attributes that are not to be used as constraints when locating the objects should be blank.

Use the attribute value allocation dialog to specify values for each attribute to be changed. To set the attribute value, click the cell next to an attribute and select a value from the drop-down menu. Attributes left blank will not be changed when the object is updated.



6 Click **Next** to access the perform update dialog.



7 The wizard now has all the information it needs to update the objects. Click **Replace Objects** to begin the update. Once the update has completed, you can view the log that displays information about the objects that have been modified.

NOTE The button on the perform update dialog differs depending on the type of operation you are performing; for example, it will display **Update**, **Remove** or **Delete** when you perform these functions.

8 Click **Finish** to close the Batch Update wizard.

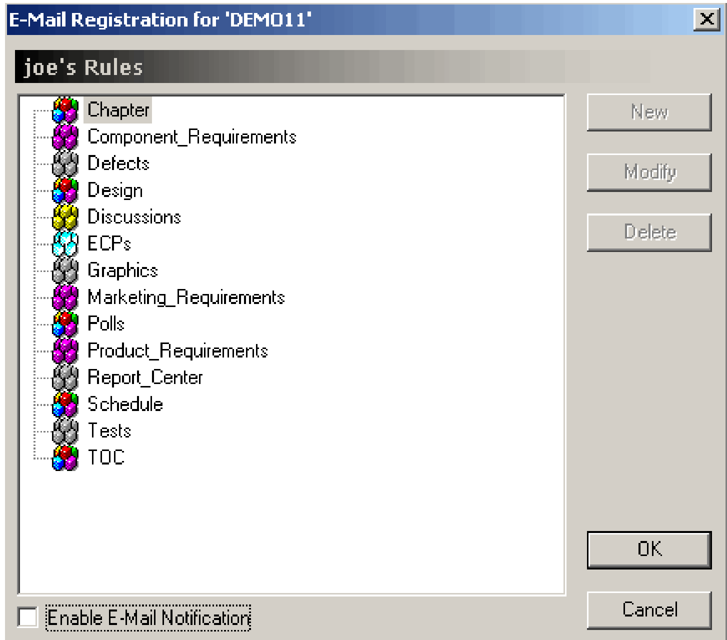
E-Mail Notification

The e-mail notification feature allows you to register interest in certain types of changes within the project data and to receive electronic notices of those changes. Project administrators can enable or disable e-mail registration for entire projects, classes, and users. Administrator settings override individual settings made in RM Concept.

E-Mail Registration For a Project User

To edit e-mail registrations:

1 Select **e-mail Registration** from the **Edit** menu. The **Mail Registration** dialog opens.



The dialog displays a list of classes to which the user has read access. The plus (+) symbol indicates that a class has one or more associated rules. To see the rules defined for a class, click on the plus symbol to expand the branch.

2 Rules that are enabled have a check mark in the check box next to them, while rules that are disabled have an empty check box next to them. To enable or disable e-mail rules for a specific user, click in the check box to the right of the desired rule.

3 To enable or disable a rule, check or clear the **Enable E-Mail Notification** check box. Disabling rules does not delete the rules; it simply prevents the mail service from processing them.

Creating a New Rule

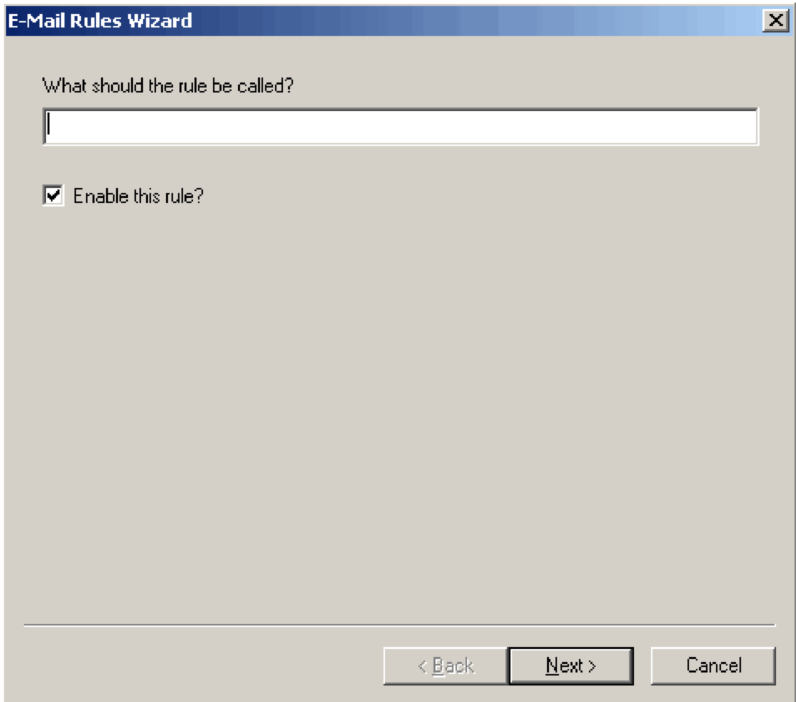
The following sections describe the E-Mail Rules wizard, which guides you through the creation of a new rule.

1 In the **Mail Registration** dialog, select the class for which the rule should be created.

2 Click the **New** button to start the E-Mail Rules wizard.

Naming the Rule

1 Provide the name for the rule in the naming rules dialog.



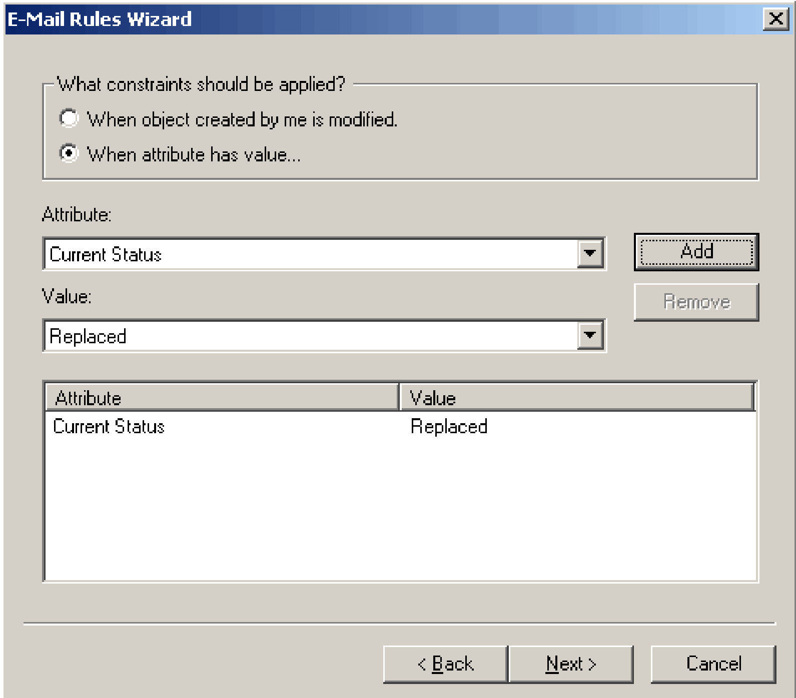
2 Configure whether the rule should be initially enabled or disabled.

3 Once all the values have been filled in, click **Next** to proceed to the next screen.

4 To return to a previous step and make changes, click **Back**. Click **Cancel** to cancel the new rule operation.

Defining the Trigger Condition

The second step of creating a new rule is to define the triggering condition for when the e-mail should be sent.



The triggering condition falls into the following two broad categories:

- **When object created by me is modified**

  This option should be chosen if you want to be notified whenever an object that you created has been changed. This will cause an e-mail to be sent to you regardless of what the change was. If this option is selected the remainder of the items on the page are hidden since they are no longer applicable.

- **When attribute has value...**

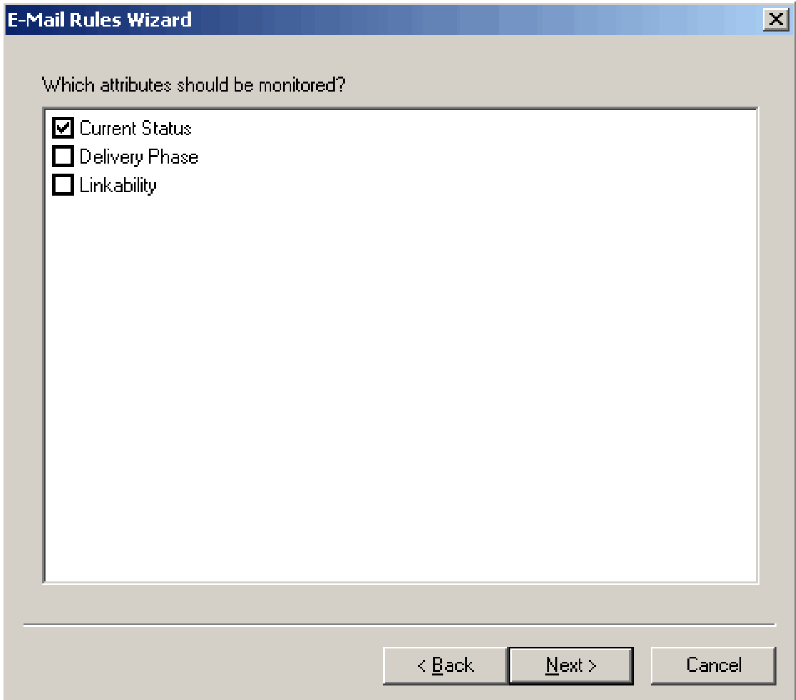
  This option allows you to specify that an e-mail should only be sent if the specified set of constraints are met. To add constraints, select the attribute from the **Attribute** drop down menu, and then the value that the object needs to have before an e-mail will be sent from the **Value** drop down menu. Multiple constraints can be added, and these constraints will be logically "and-ed" together to determine whether an e-mail should be sent. Constraints can only be added to list attributes.

Defining the Attributes to be Monitored

If **When attribute has value...** was chosen in the previous step, it is now necessary to determine which attribute should be monitored for a change in value. The mail service will only process a rule if the values for the attribute(s) selected in this step change.

In this example the rule will only be processed if the value of the **Current Status** attribute changed.

Note this page will not be displayed if you selected **When object created by me is modified** in the previous step, because an e-mail is then sent regardless of what attributes changed.



Selecting the Attribute to Include in the E-mail

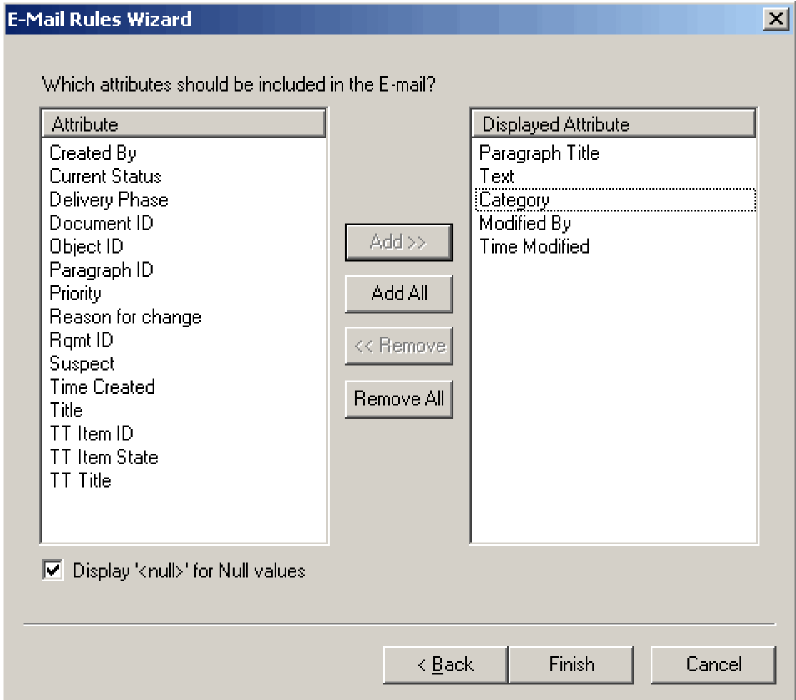
Finally you need to specify which attributes are to be included in the e-mail. The list on the left contains a list of attributes that may included with the e-mail, while the list on the right displays a list of attributes that have been selected for inclusion.

To add an attribute to the list of attributes included with the e-mail, either double-click on the attribute in the left hand list, or select the attribute and click **Add**. To remove an attribute from the list of attributes included with the e-mail either double click the attribute on the right hand list, or select the attribute and click **Remove**.

Dragging and dropping the attributes in the **Displayed Attribute** list will change the order in which the attributes are included in the e-mail. Note that the attributes will be included in the e-mail in the order they appear in the **Displayed Attribute** list.

The values for administrator-defined attributes will be included in the e-mail before the attributes that you select to display when creating a rule.

If **Display <null> for Null values** check box is checked, the string "<Null>" will be included next to the attribute display name in the e-mail if the attribute does not have a value assigned to it. If it is unchecked, nothing will be displayed after the attribute display name.



In summary, the rule created here will only be processed if the **Current Status** attribute has been changed to **Deleted**. If this is true, an e-mail will be sent containing the values for the attributes **PUID**, **Paragraph ID**, and **Text** along with the values for the attributes that the administrator selected for inclusion in the e-mail.

Modifying a Rule

To modify a rule, locate the rule in the tree and select it. Press **Modify** to run the E-Mail Rules Wizard described in the preceding sections.

Deleting a Rule

To delete a rule, select the rule in the tree, and click **Delete**.

Lock Management

Dimensions RM classes, objects, collections and source documents are implemented as sets of values within database tables. The set of values that defines a single object or collection is referred to as a *record*. When records are locked, they cannot be modified. Records may become locked after a system crash. Lock management is provided as a means of freeing records that have been locked as a result of a system crash. It is also desirable to lock records temporarily to prevent others from changing the contents while other actions take place. Lock management also provides a way to lock those records that should not be changed. CM Lock operations freeze objects so that they cannot be modified.

For security, only those records that are locked by the current user are available to unlock. If the current user has been granted Unlock privileges, all the user-locked records are available to unlock. Only users who have been granted CM Lock privileges are allowed to lock records for CM purposes. Only users who have been granted CM Lock privileges are allowed to unlock CM locked records.

NOTES You cannot use CM Lock to unlock objects that are baselined.

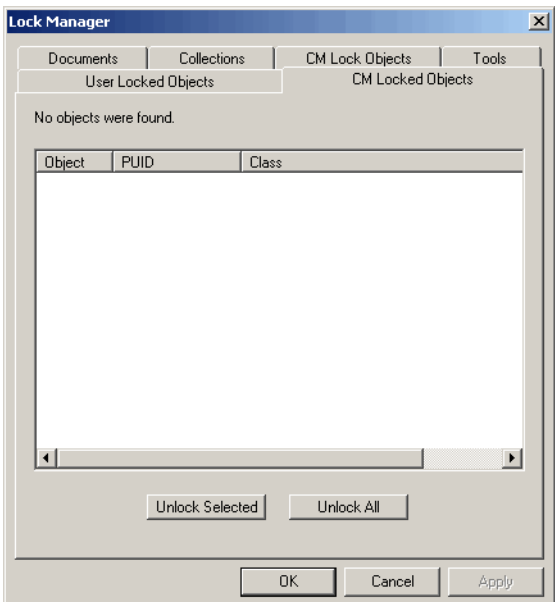
Locking Objects

The Configuration Management (CM) Lock makes objects read-only and stops records from being updated.

To lock objects:

1. Select **Administer Locks** from the **File** menu.

   The **Lock Manager** dialog opens.

2. Select the **CM Lock Objects** tab.
3. Select a class from the class list.
4. Select the objects you want to lock from the object list.
5. Select **CM Lock Selected** to lock only the selected objects. Select **CM Lock All** to lock all of the objects for the selected class and click **OK**.

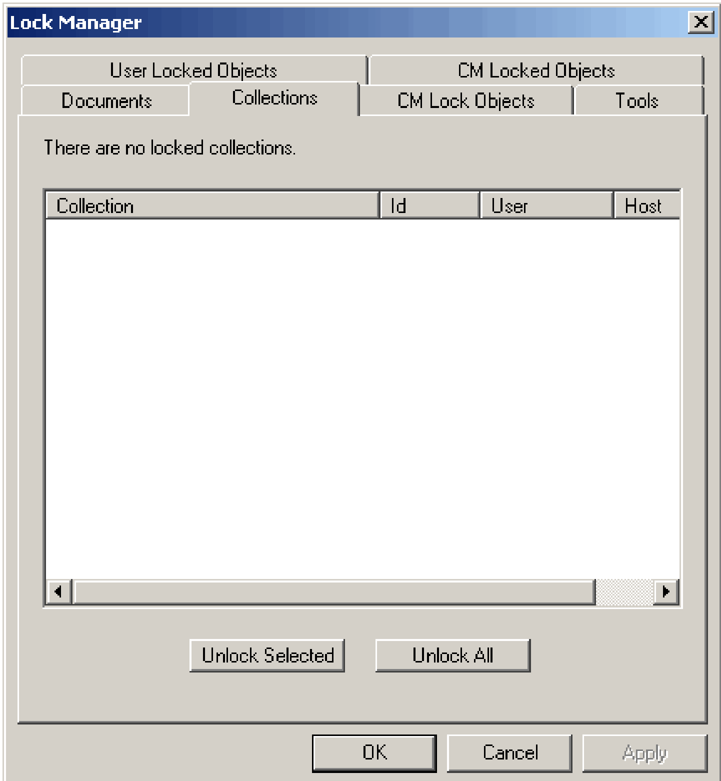
NOTE You can also CM Lock objects when you are viewing them as script results. To CMLock objects in a grid view of script results, select each row that you wish to lock and select **CM Lock Records** from the **Edit** menu.

Unlocking Objects, Documents, Collections, and Tools

To unlock objects, documents, collections, and tools:

1 Select **Administer Locks** from the **File menu**.

The **Lock Manager** dialog opens.



2 Select the **User Locked Objects, CM Locked Objects, Documents**, **Collections** or **Tools** tab as appropriate.

3 Select the locked items you want to unlock from the list.

4 Select **Unlock Selected** to unlock the selected items.

5 Select **Unlock All** to unlock all the locked items.

Chapter 3

RM Explorer

Overview

CAUTION! RM Explorer is a legacy client. Do **NOT** use any legacy client feature unless it is listed as currently supported in this manual.

The following RM Explorer features are currently supported:

- Auto Link
- Batch update
- Creating parent containers
- Filter security
- Relationship security
- Script security
- Script renaming

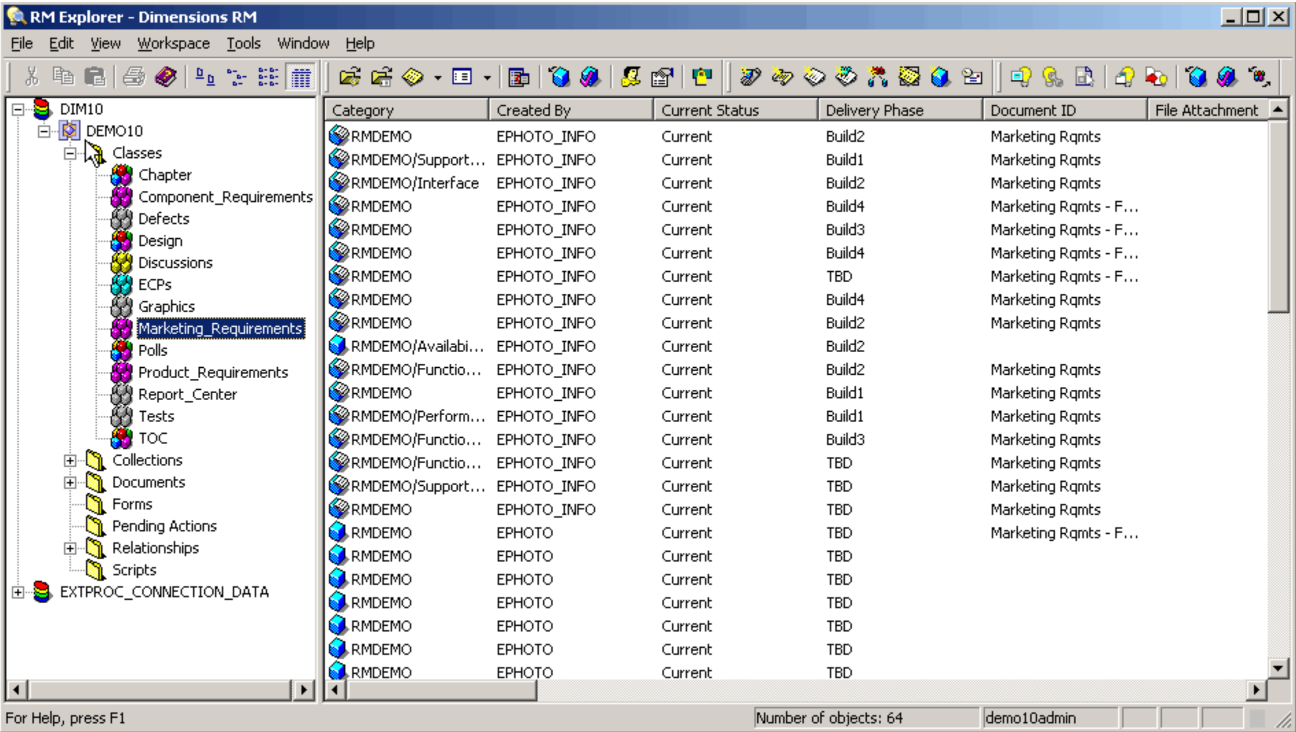
Do not use any other features of RM Explorer. Do not launch any other RM tools from RM Explorer (the Tools menu).

Navigation

The RM Explorer workspace uses folders and icons that make navigation and file management easy to use. Similar to Windows Explorer, you can double-click to open an item and use toolbars to view tools. You also can view properties of folders by right-clicking the mouse or change the appearance of items. The following describes some of the common navigation features.

- Click a folder on the left side of the window to display its contents on the right side.
- Click the plus signs (**+**) to expand the tree; click the minus signs (**-**) to collapse the tree.
- To quickly open a folder and display the folders inside, double-click the folder on the left side of the window.
- To quickly open an item, double-click the item on the right side of the window.
- To change the size of either side of the window, drag the bar that separates the two sides.
- To change the size of a column on the right, drag the column heading.
- To change the appearance of the list on the right, click the **View** menu. Choose **Large Icons** , **Small Icons** , **List** , or **Details** .
- To hide or show the main toolbar, select **Toolbar** from the **View** menu.
- To hide or show the status bar, select **Status Bar** from the **View** menu.

- To refresh the view from the database, select **Refresh** from the **View** menu or press the F5 key.

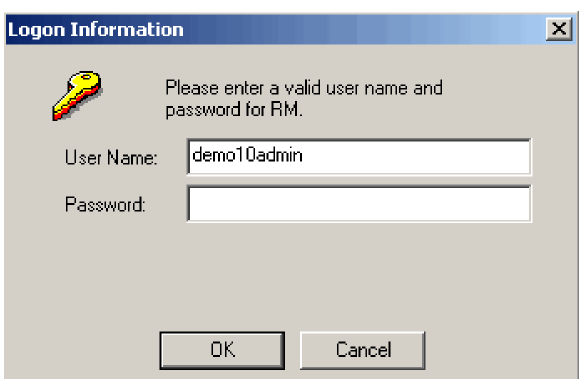


Logging on to a Database

To log on to a database:

1. Click a database name or expand the database folder.
2. Enter a valid ID and password.

   If you are required to change your password at the next logon, Dimensions RM prompts you to enter a new password at this time. If your user ID has been validated, the assigned projects are available.



Logging On as a Different User

To log on as a different user:

1 In the left window of RM Explorer, select the database to which you want to log on.

2 Select **Change User** from the **File** menu, or click the **Change User** button .

3 Enter the new user ID and password.

4 Click **OK.**

Filter Security

Setting Filter Security for a Group

To set filter security for a group:

1 Select the class that has the filter for which you want to set security.

2 Select **Filters** > **Define Filters** from the **File** menu, click the **Define Filters** button , or right-click the class and select **Define Filters**.

The **Define Filters** dialog opens.

3 Select the filter for which you want to set security.

4 Click the **Security** button .

The **Security** dialog opens.

5 Expand the **Groups** folder by clicking **+**.

6 Select the group for which you want to set security.

7 Change the access rights for a transaction by selecting the transaction:

- Explicit access granted is indicated by a checked box .
- Inherited access is indicated by a gray box .
- Explicit access denied is indicated by an unchecked box .

Setting Filter Security for a User

To set filter security for a user:

1 Select the class that has the filter for which you want to set security.

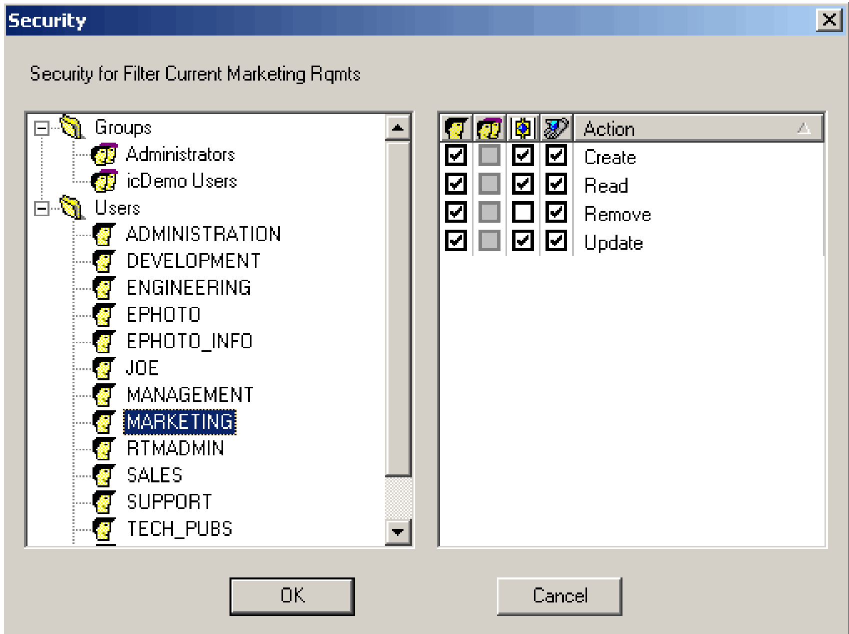
2 Select **Filters** > **Define filters** from the **File** menu, click the **Define Filters** button , or right-click the class and select **Define filters**.

The **Define Filters** dialog opens.

3 Select the filter for which you want to set security.

4 Click the **Security** button .

The **Security** dialog opens.



5 Expand the **Users** folder by clicking **+**.

6 Select the user for which you want to set security.

7 Change the access rights for an action by clicking the check box next to it. The following describes the identification for each access right:

- Explicit access granted is indicated by a checked box ☑.
- Inherited access is indicated by a gray box .
- Explicit access denied is indicated by an unchecked box ☐.

Batch Update Wizard

The Batch Update wizard is designed to allow attribute values to be changed for all objects that match a set of attribute constraints for any one class defined in the project schema. When running the Batch Update wizard the following operations are available:

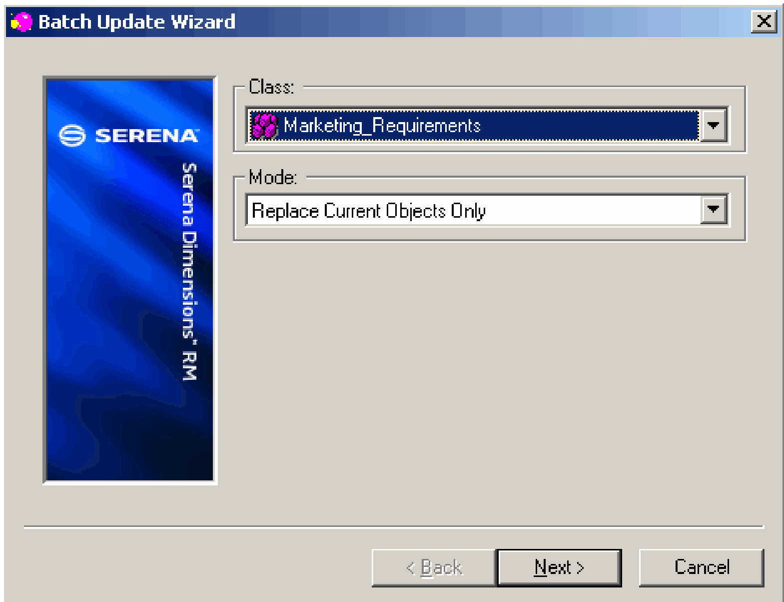
- **Update Current Objects Only—**Updates all current objects that match the specified attribute constraints and updates the attribute values to the new attribute values provided.
- **Update All Objects—**Updates all objects that match the specified attribute constraints regardless of status and updates the attribute values to the new attribute values provided. You can update all objects only if this privilege has been given to you by the system administrator.
- **Replace Current Objects Only—**Replaces all current objects that match the specified attribute constraints and updates the attribute values to the new attribute values provided.
- **Delete Objects—**Deletes all current objects that match the specified attribute constraints.

- **Undelete Objects—**Undeletes all deleted objects that match the specified attribute constraints.
- **Remove Objects—**Removes all current objects that match the specified attribute constraints. **Remove Objects** destroys the matching objects permanently and cannot be undone. This is a good way to remove all objects captured from a specific document (if the document IO field exits).

To update or replace objects in the project database:

1 Select **Batch Update Data** from the **File** menu, click the **Batch Update Data** button , or right-click a class and select **Batch Update Data**.

The **Batch Update wizard** opens.



2 Select the class that contains the objects you want to modify from the **Class** field.

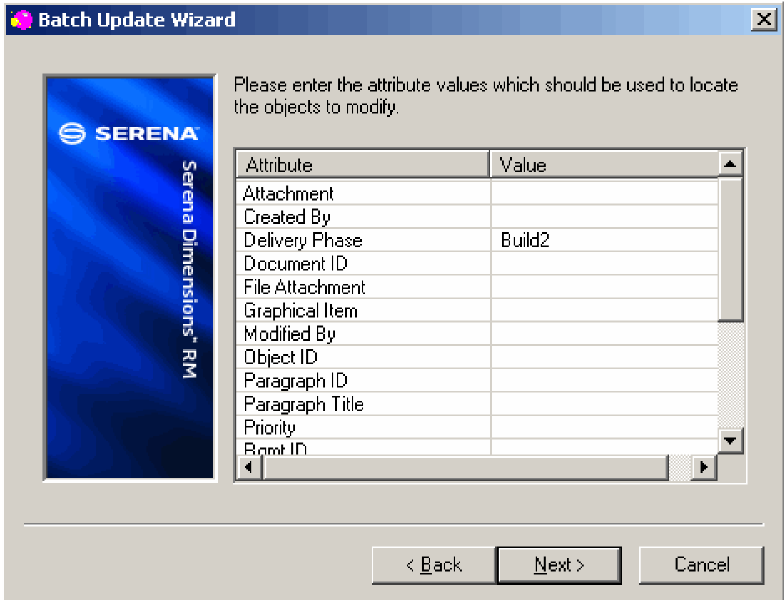
3 Select the operation to be performed on the objects from the **Mode** field.

The following operations are available:

- **Update Current Objects Only**—Updates all current objects that match the specified attribute constraints and updates the attribute values to the new attribute values provided.
- **Update All Objects**—Updates all objects that match the specified attribute constraints regardless of status and updates the attribute values to the new attribute values provided.
- **Replace Current Objects Only**—Replaces all current objects that match the specified attribute constraints and updates the attribute values to the new attribute values provided.
- **Delete Objects**—Deletes all current objects that match the specified attribute constraints.
- **Undelete Objects**—Undeletes all deleted objects that match the specified attribute constraints.

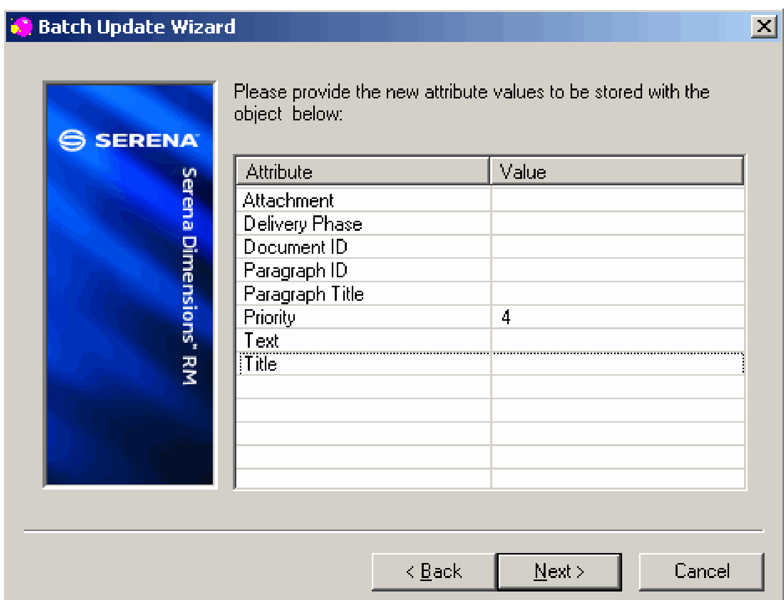
- **Remove Objects**—Removes all current objects that match the specified attribute constraints. This destroys the matching objects permanently and cannot be undone.

4 Click **Next** to advance to the attribute constraint dialog.



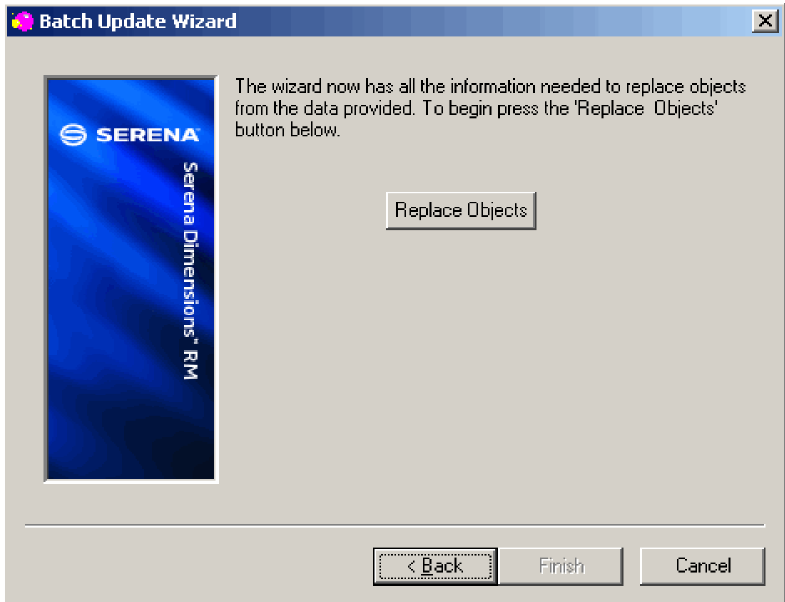
Use the attribute constraint dialog to provide values for the attributes used as constraints when searching for objects. Only certain attribute types can be constrained; for example, text contained in the **Description** attribute cannot be constrained. The left column contains an entry for each attribute defined for the class. The right column allows you to define the constraint value for the attribute.

5 To add a constraint to an attribute, click the **Value** cell next to the attribute and enter the appropriate value. Leave this cell blank for any attributes that are not to be used as constraints when locating the objects.



Use the attribute value allocation dialog to specify values for each attribute to be changed. To set the attribute value, click the cell next to an attribute and select a value from the drop-down menu. Attributes that are left blank are not changed when the object is updated.

6 Click **Next** to advance to the perform update dialog.



7 The wizard now has all the information it needs to update the objects. Click **Replace Objects** to begin the update. Once the update has completed, you can view the log that displays information about the objects that have been modified.

NOTE The button on the perform update dialog differs depending on the type of operation you are performing; for example, it displays **Update**, **Remove**, or **Delete** when you perform these functions.

8 Click **Finish** to close the **Batch Update wizard**.

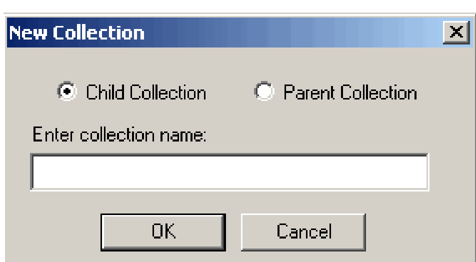
Creating a New Collection

To create a new collection:

1 Select the collection folder in which you want to add the new collection.

2 Select **New** > **Collection** from the **File** menu, click the **New Collection** button, or right-click the **Collection** node and select **New**.

The **New Collection** dialog opens.



3 Select **Child Collection** or **Parent Collection**.

4 Enter the name of the collection and click **OK**.

5 Press F5 to show the new keyword in the tree.

Form Security

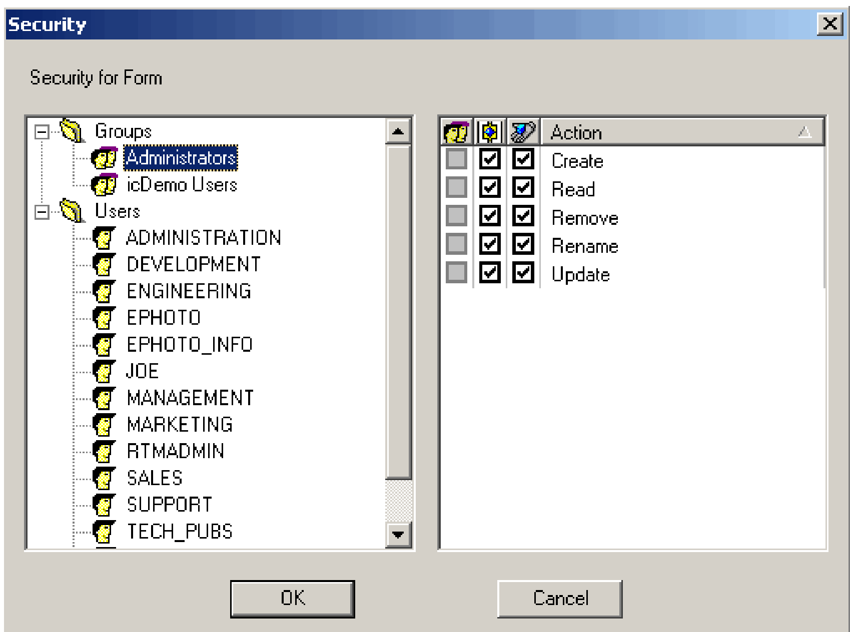
Setting Form Security for a Group

To set form security for a group:

1 Select a form.

2 Select **Security** from the **File** menu, click the **Security** button , or right-click the collection and select **Security**.

The Security dialog opens.



3 Expand the **Groups** folder by clicking **+**.

4 Select the group for which you want to set security.

5 Change the access rights for a transaction by selecting the transaction:

- Explicit access granted is indicated by a checked box .
- Inherited access is indicated by a gray box .

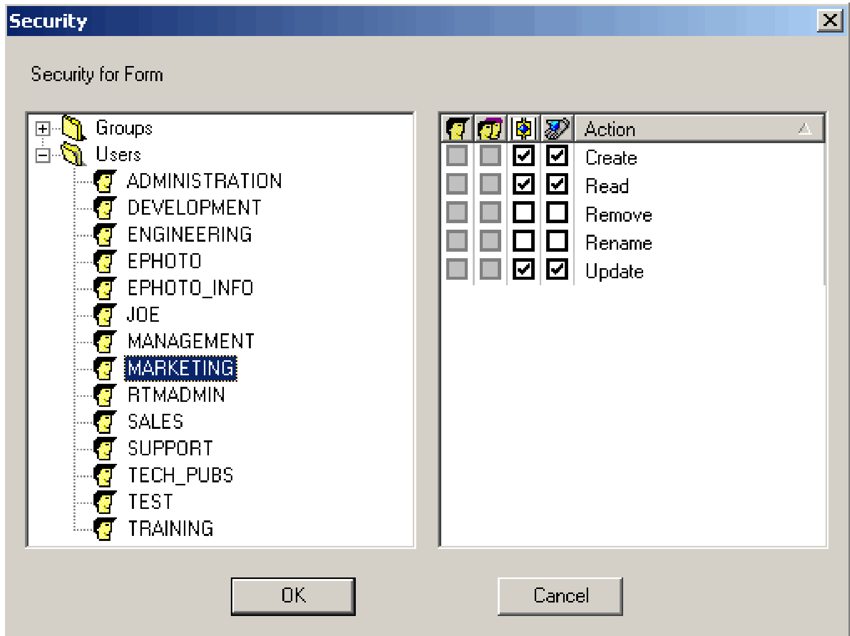
Setting Form Security for a User

To set form security for a user:

1 Select the form for which you want to set security.

2 Select **Security** from the **File** menu, click the **Security** button, or right-click the collection and select **Security**.

The **Security** dialog opens.



3 Expand the **Users** folder by clicking **+**.

4 Select the user for which you want to set security.

5 Change the access rights for a transaction by selecting the transaction:

- Explicit access granted is indicated by a checked box.
- Inherited access is indicated by a gray box.
- Explicit access denied is indicated by an unchecked box.

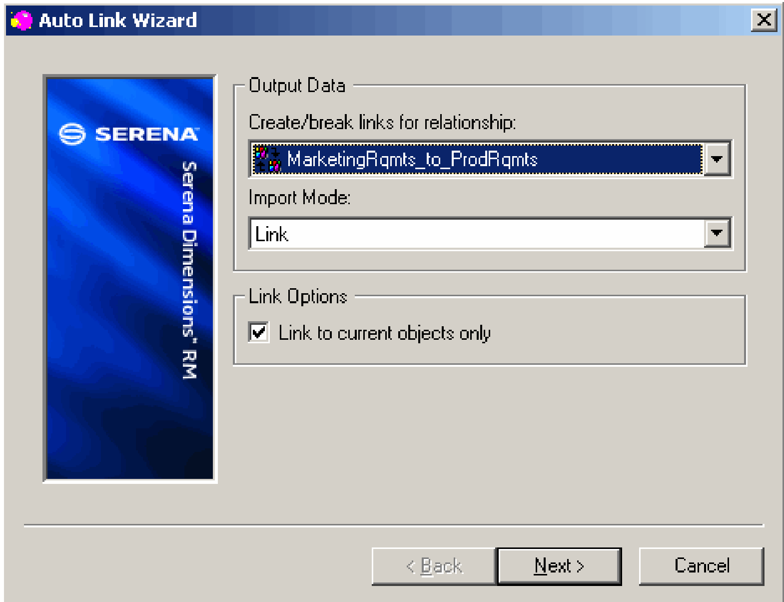
Auto Link

The Auto Link wizard is used to either create or break links between objects of the primary and secondary class connected to the selected relationship. Constraints are provided for the primary and secondary classes to select the objects that are to be linked. Attribute values can also be provided for the links that are created.

To create links:

1 Select the relationship for which you want to automatically create links.

2 Select **Auto Link** from the **Tools** menu or click the **Auto Link** button . The Auto Link wizard opens.



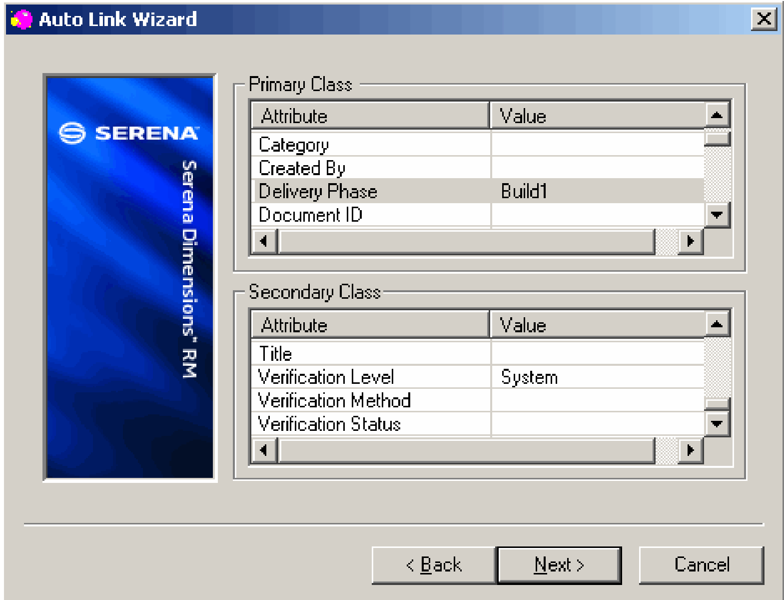
3 Select a relationship from the **Output Data** field. This contains the Create/Break links for the relationship drop-down list.

4 Select an import method. This creates or breaks links in a list of relationships that you could reference. The following methods are available:

- **Create**
- **Delete**
- **Undelete**
- **Remove**

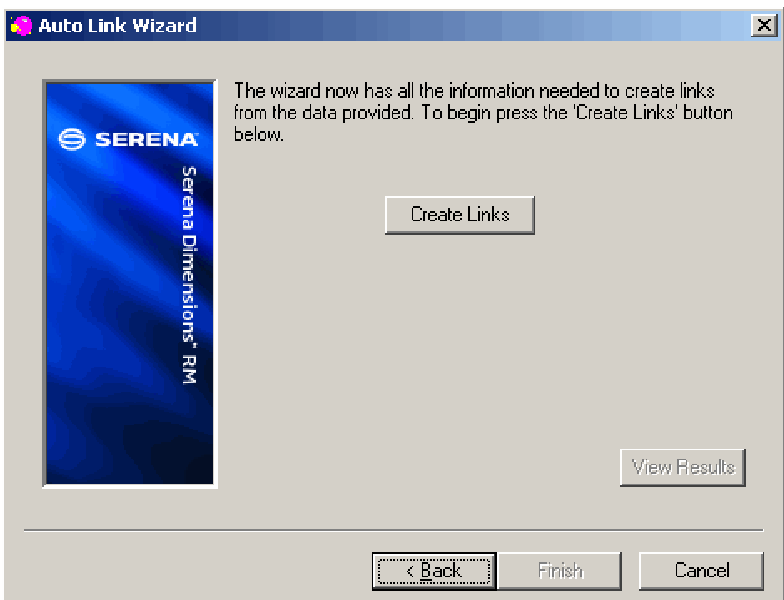
5 Click **Link to current object only** to creating links only among current objects.

6 Click **Next** to advance to the attribute constraint dialog. Use this dialog to provide values for the attributes of the primary and secondary classes used as constraints.



7 To set an attribute constraint, click the cell next to an attribute and enter a constraint value or select one from the drop-down menu. Attributes that are not used as search constraints should be left blank.

8 Click **Next** to advance to the perform link dialog.



The wizard has all the information needed to create a new link among the objects.

9 Click **Create Links** to begin the link creation process. When the import is completed you will have the option to view the log containing details of the links.

10 Click **Finish** to close the Auto Link wizard.

Renaming a Script

To rename a script:

1. Select the script you want to rename.
2. Select **Rename** from the **File** menu, click the **Rename** button, or double-click the script name.
3. Enter the new script name.

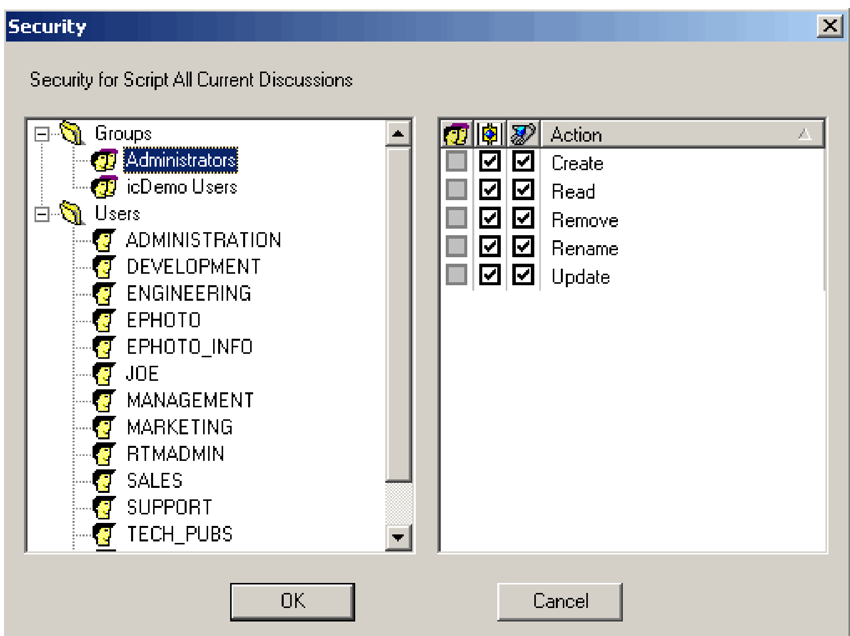
Script Security

Setting Script Security for a Group

To set script security for a group:

1. Select the script for which you want to set security.
2. Select **Security** from the **File** menu or collection context menu, or click the **Security** button .

   The **Security** dialog opens.



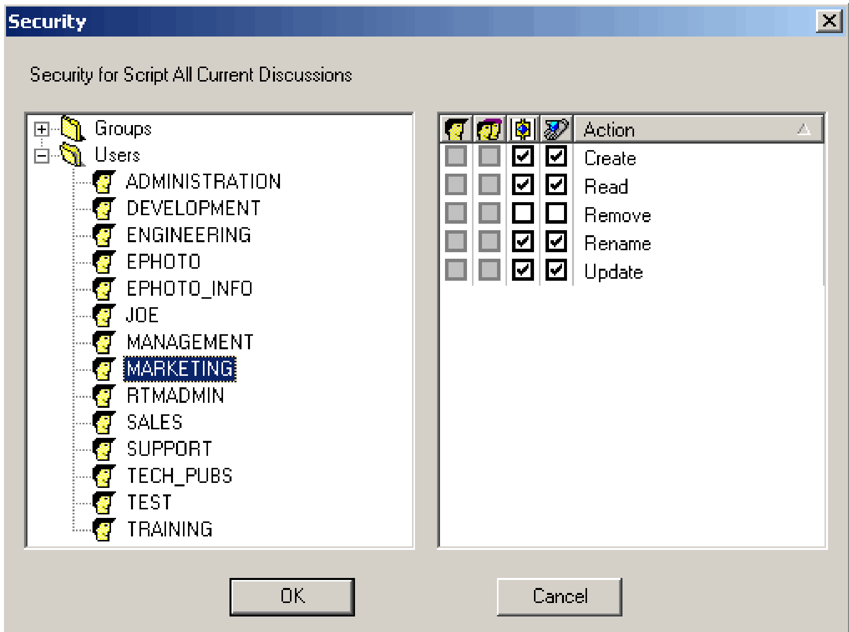
3. Expand the **Groups** folder by clicking **+**.
4. Select the group for which you want to set security.
5. Change the access rights for a transaction by selecting the transaction.
   - Explicit access granted is indicated by a checked box .
   - Inherited access is indicated by a gray box .

Setting Script Security for a User

To set script security for a user:

1. Select the script for which you want to set security.
2. Select **Security** from the **File** menu, click the **Security** button, or right-click the script and select **Security**.

   The **Security** dialog opens.



3. Expand the **Users** folder by clicking **+**.
4. Select the user for which you want to set security.
5. Change the access rights for a transaction by selecting the transaction:
   - Explicit access granted is indicated by a checked box.
   - Inherited access is indicated by a gray box.
   - Explicit access denied is indicated by an unchecked box.

Chapter 5

Command Line Tools

Overview

Serena® Dimensions® RM provides doc_out as the command line tool. doc_out allows you to run scripts from the DOS prompt. The command line tool can be run when you want to create large reports on links among entities created in or removed from the database.

Setting Your Path

doc_out is normally installed in *RTM_HOME\bin*. This directory must be in your path for the tools to be visible. Check with your project administrator for the location in your installation. Alternatively, you can search for the tool (*doc_out.exe*) using the **Search** command from the Windows **Start** menu.

To add the Dimensions RM bin directory to your path:

1 Right-click the **My Computer** icon on your desktop.

2 Select **Properties**.

3 Click the **Environment** tab.

4 Select the Path variable from the system variables list.

5 In the **Value** box at the bottom of the page, insert the Dimensions RM bin path into the value string. Separate that path from the next one with a semi-colon (;).

6 Click **Set**, **Apply**, and then **OK**.

Using doc_out

The doc_out tool allows you to run a Dimensions RM reporting script from the command line and to generate the output to a variety of formats. Using doc_out enables you to extract data from Dimensions RM without invoking a Windows-based Dimensions RM session. Scripts can be prepared before extraction for batch processing of reports.

Invoking doc_out

To invoke the doc_out tool:

Type the following command at the command line prompt:

```
doc_out   required_arguments   [optional_arguments]
script_identification
```

where:

`required_arguments` are:

```
-user username -pwd password -project projectname -location dblocation
```

- `username` is your Dimensions RM user account name.
- `password` is your Dimensions RM user account password.
- `projectname` is the project from which you want to extract data.
- `dblocation` is the database name in which the project resides. Your local *tnsnames.ora* file contains the database names of each database to which you have visibility.

`optional_arguments` are:

`[-m format] [-#variablename value]* [-o outputfile] [-e] [-v] [-s[n|u terminatorstring]`

- `format`–The format in which you want the output to be represented. The accepted formats are described in the following table.

Format	Description
csv	Comma Separated Values format
rtf	Rich text format used by Microsoft Word
rtf_table	Rich text table format
sgml	SGML markup format
sgml_table	SGML table format
html	HTML format
html_table	HTML table format
none	No markup (that is, plain ASCII)
ascii_table	ASCII text with spacing to create a tabular effect. **NOTE:** This is the default output format.

- `variablename` is the name of a variable defined in the script and **value** is the actual value that you want substituted for that variable. You will need to define **- #variablename value** pairs for each variable defined in the script.
- `outputfile` is the filename or path in which the results are to be stored. Enclose the path in double quotation marks when spaces or punctuation characters are included in the filename.
- `-e` suppresses word processor markup at the beginning of the outputfile (not accepted for RTF or SGML formats). Omitting markup is useful when you will be appending the output to an existing file.
- `-v` requests program arguments be echoed to the console or DOS window.
- `-s` (used only with the -m none option) with no argument causes a standard end-of-record marker (“#### End of Record ####”) to be appended to each record in the report.
- `-su` (used only with the -m none option) terminatorstring provides a textual end-of-record marker for the -su option. Enclose terminatorstring in double quotation marks when spaces or punctuation characters are included the string.
- `-sn` (used only with the -m none option) eliminates blank lines in the unformatted output. Blank lines are otherwise inserted for empty attribute values. -sn can be used in combination with -su, but overrides -s.

`script_identification` is:

`scriptfilename | -n scriptname`

- `scriptfilename` is the path of the script stored in the file system. You can save scripts in RM Concept as `.rmdoc` format.
- `-n scriptname` identifies a script stored in the project. Scriptname is required with the -n option. If the script name in the project includes blanks or punctuation characters, you will need to enclose the name in double quotes (for example, `-n “my favorite script”`).

Using doc_out from the Command Line

Various ways to use the doc_out tool from the command line are illustrated in the following DOS examples.

Example 1

Minimum arguments. Assumes that doc_out is in the path, the script text is in the current folder, and the results are to be displayed on the screen. The output format defaults to ascii_table.

```
C:\>doc_out -user fred -pwd fredspassword
-project myproject -location bigserver.ic
getCustomerRequirements.rmdoc
```

Example 2

Similar to Example 1, but references a script inside the project and an output file in the current folder in which to save the results.

```
C:\>doc_out -user fred -pwd fredspassword
-project myproject -location bigserver.ic
-n “Get Customer Requirements” -o CustomerRequirements.txt
```

Example 3

Similar to Example 2, but requests the output in a csv file that can then be imported into other tools.

```
C:\>doc_out -user fred -pwd fredspassword -project
myproject -location bigserver.ic -n “Get Customer
Requirements” -m CSV
-o CustomerRequirements.csv
```

Example 4

Similar to example 2, but specifies no markup and uses the date as a record separator.

```
C:\>doc_out -user fred -pwd fredspassword -project
myproject -location bigserver.ic -n “Get Customer
Requirements” -m none -su “09/15/2000”
-o CustomerRequirements.txt
```

Example 5

Similar to Example 3, but uses a variable to specify a collection that identifies a subset of the objects

```
C:\>doc_out -user fred -pwd fredspassword -project myproject
-location bigserver.ic -#LookIn Baseline1 -n "Get
Customer Requirements" -m CSV -o CustomerRequirements.csv
```

Using Script Variables with doc_out

Variables can be used within a Dimensions RM reporting scripts only when the script is run by doc_out. Attempting to use them in RM Concept or RM Browser results in erroneous results because the variable values are substituted in doc_out before submitting the script to Dimensions RM.

To use variables in scripts, follow these rules:

- Use UPPER CASE for the variable name in both the script and the command line reference.
- Enclose spaces or punctuation in values in double quotation marks.
- In the command line, precede the variable name with a '#', but in the script, include the variable name without any special markings.

The following example is a parameterized version of the baseline check script provided in the RMDEMO sample project. The script identifies three subsets of objects given two collections used as baselines. The first subset is (the first select statement) the set of objects in the earlier baseline that are not deleted and are still in the later baseline (as identified by the second select statement). The second subset is those objects that are in the later baseline, but were not in the earlier baseline. The third subset is those objects that were in the earlier baseline and were deleted.

```
select  <PUID>PUID <Earlier>TEXT from CustomerRequirements where
group in ('EARLY') and STATUS != 'Deleted'
xref source secondary_history
select  <PUID>PUID <Later Modifications>TEXT from
CustomerRequirements where group in ('LATE')
plus
select  <PUID>PUID <Later Additions>TEXT from CustomerRequirements
where group in ('LATE')  and group not in  ('EARLY')  and
NOT_SECONDARY_IN immediate
plus
select  <PUID>PUID <Earlier Deletions>TEXT from
CustomerRequirements where group in ('EARLY') and STATUS='Deleted'
```

The baseline check script is invoked to compare Baseline 1 and Baseline 2 from the DOS command line as:

```
C:\>doc_out -user fred -pwd fredspassword -project rmdemo
-location bigserver.ic baselinecheck.rmdoc -#EARLY
"Baseline 1" -#LATE "Baseline 2"
```

Glossary

Accept — A command that accepts a proposed change. The current status of the proposed requirement becomes "accepted," and a copy of the requirement is created with the Current Status of "current."

accepted — Current Status of a change request that was accepted.

access rights — The set of tasks that can be performed on a resource by a user.

action attribute — A pointer to a file that is held internal or external to Dimensions RM, and indicates the method of accessing the file.

alias — A set of keywords defined as variants or synonyms of a main keyword. For example, aliases of the keyword "calibrate" might be "calibrated," "calibrating," or even the wild card string "cal*". Unlike pseudonyms, which exist only while a particular Dimensions RM tool is active, aliases exist for the duration of the project or until they are deleted.

alphanumeric attribute — An attribute that represents one line of alphanumeric text, such as the title of an acceptance test. It can be up to 1000 characters in length.

alphabetic sort — A simple alphabetical sort. Contrast with *numeric sort*.

attribute — Information that is logically associated with a class of information to further specify the information content. See also *class attribute*, *evaluated attribute*, *implicit attribute*, *relationship attribute*, *user-defined attribute*.

attribute constraint — A rule that permits a link to be created only if some attribute of the primary or secondary object obeys a specific constraint. See also *primary object*.

attribute type — The nature or data type of an attribute: alphanumeric string, free text field, and date.

audit trail — An historical trace of the various versions of requirements that lets you reconstruct requirement evolution. The Visual Network tool lets you view the audit trail graphically.

auto link — A utility that lets you create or break links between objects of the primary and secondary class in the selected relationship.

baseline — A stable, unchangeable group of requirements. Baselining a collection ensures that the collection will never change.

batch capture — A method of capturing objects that is performed using the batch capture utility in RM Word. This method is particularly suitable for capturing requirements from large and structured documents. See also *manual capture*.

batch update — A utility that lets you change attribute values for all requirements that match a set of attribute constraints for any one class defined in the project schema.

capture	The process of capturing an object when it is extracted from the original source document and entered in the Dimensions RM database. See also *source document*.
category	A method for organizing objects so you can create views of requirements, scripts, and filters for a subset of users. A requirement can belong to only one category.
cardinality rule	A rule that specifies the maximum number of links that can lead to and from primary or secondary objects. For example, a cardinality of 2:3 means that no more than two links can lead to a secondary object, and no more than three to the primary object.
change request	A proposal to change one or more requirement attributes.
child collection	Object hierarchies are created from the top down, from parent to child, while collection hierarchies are created in the opposite direction, by grouping child collections to form a parent collection, and so on. A child collection may be directly linked to an object. When a collection is created, it is a child collection by default.
child database	Used in the context of database partitioning only. Contrast with *parent database*.
child object	Whenever an object is edited and replaced, a new object is produced. The original object is called the parent object, and the new object is its child object. If this process is repeated, a child can itself be a parent of another child.
class	A container for related types of information. After classes are defined, requirements are entered or captured into the class.
class attribute	A property of a class, as defined by the project administrator, that further breaks down the information in the class. By specifying attributes for a class, the project administrator can define the exact nature of the information represented by the class. This allows Dimensions RM users to make complex searches on their project information. Without defined attributes, classes model information at only a relatively high level of abstraction and lack internal detail. In such cases, the resulting diagram simply shows an overview of the abstract data types pertaining to the project, together with their interdependencies.
class definition	The initial Dimensions RM information modeling procedure the project administrator uses to configure Dimensions RM with respect to the information to be generated and traced within the project. This is represented graphically as a class definition diagram.
class definition diagram	A graphical representation of the information classes that exist in a project, along with the relationships between the classes.
Class Definition	A Dimensions RM tool that lets users with special privileges (such as project administrators) define various classes of information, attributes of those classes, and the relationships between the classes. By specifying the project structure in this way, a class definition both constrains and supports the systems engineer in the way that instances of classes, attributes and relationships can be created during the lifetime of the project. Systems engineers and other ordinary users can use Class Definition to view the class definition diagram for the project. See also *class definition diagram*.
CM Lock	Configuration Management Lock. A security feature that makes objects read-only and stops them from being updated. You can lock requirements, collections, and documents.

collection	A way to group requirements of any class. Once a collection is created, it can be associated with a requirement by linking the requirement to the collection. Each requirement can be linked to many different collections, and each collection can be linked to many different requirements. Parent collections contain child collections. Child collections contain requirements. Parent collections are not directly linked to any requirements, only linked indirectly through their child collections. See also *baseline*.
collection linkability	An occurrence of the association defined by a relationship. It connects two objects.
command line tools	Dimensions RM tools (doc_out) that let you to run scripts from the DOS prompt. Doc_out lets you run a Dimensions RM reporting script to generate output to a variety of formats. With doc_tool, you can prepare scripts before extraction for batch processing of reports.
compliance check	A process in which Dimensions RM searches the database and produces a report specifying which objects do not contain links across a defined relationship.
compliance report	A report that lists requirements that are or are not linked to objects in the other class in a relationship. A full compliance report lists all requirements in the primary and secondary class, whether or not they are linked to each other. A compliance-only report lists either all matching requirements in the primary class that have links to matching requirements in the secondary class, or all matching requirements in the secondary class that have links to matching requirements in the primary class. A non-compliance report lists either all matching requirements in the primary class that have no links to matching requirements in the secondary class, or all matching requirements in the secondary class that have no links to matching requirements in the primary class.
CSV import	A utility that lets you import data from a comma separated value file into the Dimensions RM project database.
current	Current Status of a requirement that is the most recent or current version.
Current Status	A special implicit attribute that identifies the state of a requirement.
cyclic relationship	Relationship in which the relationship points from one class back to the same class.
database	In the Dimensions RM environment, an instance of Oracle. The databases that are displayed in the Dimensions RM tools are determined by the contents of the `tnsnames.ora` file (an Oracle file).
database partitioning	The division of project data between a contractor and subcontractors. This is used for large projects undertaken by several groups or organizations. The original objects are retained in the main contractor's parent database, and allocated objects in the subcontractor's child database.
date attribute	A user-defined attribute type that stores values that are based on user-defined date formats.
Delete	A command that changes the Current Status of a requirement to "deleted," but leaves the requirement in the project.
deleted	Current Status of a requirement that was deleted. A deleted requirement remains in the project. The prior version of the requirement, if any, receives a status of "current."

derivation	The analysis process in which an object is changed or translated into a form suitable for lower-level analysis and design.
derived object	A lower-level object that is necessary for the implementation of a higher-level object. When an object changes form, it becomes a derived object. In general, a derived object is directed toward some sub-element and is more specific than the original object.
Dimensions RM	A suite of multi-user, configurable tools that support the capture, management, traceability and documentation of systems engineering information.
Dimensions RM project administrator	The person responsible for maintaining the data that is accessible to a particular project. This includes using the Class Definition tool to implement the information model and, in some cases, the Database Management Utility to manage project information.
Dimensions RM third-party integrator	Person responsible for using the API functions to integrate third-party tools with Dimensions RM.
doc_out	A command line tool that lets you run a reporting script and generate the output to a variety of formats.
Doctool	A Dimensions RM tool that interprets a documentation script and generates an on-screen report.
document	In the Documents View perspective in RM Browser, a hierarchical arrangement of chapters and requirements that can be edited. You can create a document from within Documents View or open one that was imported by the RM Import tool. From Documents View, you can publish a document as a Microsoft Word document.
Documents View	An RM Browser view that provides a document-like presentation of requirements, with a table of contents, chapters, and subchapters. Requirements are contained within the chapters and subchapters. Document View allows you to easily add, delete, move, and edit chapters and requirements. Microsoft Word documents that you imported through RM Import are displayed in Document View. You can easily add, delete, move, and edit chapters and requirements from the imported Word document in Document View. See also *Requirements View*, *Home View*, *Traceability View*.
ECP	Engineering Change Proposal. A class type. As you create a set of related proposed requirements, you can link them to an ECP object so that they are easily accessible for review as a group.
e-mail notification	A feature that lets you register interest in certain types of changes within the project data and to receive electronic notices of those changes.
evaluated attribute	An attribute that takes its value from the external environment. Such an attribute can be specified as the default value for alphanumeric, numeric or date attributes. At run time, the specified script or command is executed and the resulting value is set for the attribute.
expanding	A process in which a single parent object is edited to produce one or more child objects.

export utility	A utility that can be used to back up a project or database. In the context of database partitioning, it generates a data partition package containing all the data and external files associated with a data partition. The package can be created as a collection of files in a single directory or as a single file, ready for transfer to the destination site.
file attachment attribute	A user-defined attribute type that holds a single file that can be accessed through RM Browser
filter	A query against a single class. Form filters are the simplest kind of filter, but they are limited in terms of the complexity of the selection criteria. Complex filters provide greater expressive power for the selection criteria. With complex filters, you can construct detailed logical expressions that use attribute values, ranges of attribute values, and membership in relationships as selection criteria. Contrast with *script*.
flowdown	A systematic process in which objects are decomposed into allocated and derived objects, and then assigned to low-level model components. This flowdown process generates a hierarchical structure of refined objects derived from the objects captured for the system.
focusing	A process in which two or more parent objects are edited to produce a single child object.
form	A structure that displays requirement information for classes and relationships. A form is created for each class and relationship. You can create new forms by customizing the form that Dimensions RM generates, and can designate any form to be used as the default form.
form view	In RM Browser and RM Concept, a view that displays requirements one at a time. From the form view in RM Browser and RM Concept, you can edit requirement attributes. From the form view in RM Concept, you can also find objects.
genealogical links	Links between parent objects and child objects, or between parent collections and child collections.
generic links	Links that must span a relationship.
graphic attribute	A user-defined attribute type that holds either graphic images or OLE data captured from RM Word. See also *OLE*.
grid view	In RM Browser and RM Concept, a view that lets you view multiple requirements in a table-like list. The column headings represent attributes of the requirements.
group	A collection of individual users grouped into a functional category. Access rights can be assigned to a group and all members of the group. If users have been assigned to a project through a group, they inherit the group access rights, unless they have been explicitly granted or denied access.
Home View	An RM Browser view that you can customize to include up to seven expandable sections. Each section contains the results of a query. RM Home allows you to quickly view and modify requirements that you refer to on a regular basis. See also *Documents View*, *Requirements View*, *Traceability View*.
immediate child	The object that was created when the original object was replaced, focused, or expanded. Immediate children are the next version of objects in the line of descent and may be current objects or objects with another status.

immediate parent	The object that was used to create the currently selected object. Parent objects never have a status of "current."
immediate relationship	A relationship that refers to the immediate predecessor or successor of an object. Contrast with *source relationship*.
implicit attribute	An attribute that is used to maintain the integrity of project information. Implicit attributes include intrinsic information such as the project unique identifiers (PUIDs), object IDs, and modification times. You cannot modify implicit attributes. Implicit attributes are supplied for each class and relationship. Contrast with *user-defined attribute*.
import utility	A utility that can be used to restore a project or database from backup.
lifecycle	The phases of a project from its initial requirements specification through its implementation.
link	An instance of a relationship. You can link two requirements together if a relationship between their corresponding classes is defined.
list attribute	A user-defined attribute type that provides a list of values from which the Dimensions RM user can make a selection. For example, if you require the Dimensions RM user to choose one of a given set of values for the attribute *test_result*, specify the attribute as a list attribute, and define *pass*, *fail*, and *untested* as the set of allowed values.
lock manager	A Dimensions RM tool used to lock and unlock Dimensions RM database elements such as source documents, tools, objects, and collections.
lowest level child	A current object that is descended from the selected object. The objects contained in the lowest-level children list may skip generations of an object; that is, they need not be immediate children of the selected object.
mandatory attribute	An attribute for which users must specify values. Contrast with *optional attribute*.
manual capture	A method of capturing objects one at a time using the RM Capture tool. This method is particularly suitable for capturing objects from documents that are short or not formally structured. Contrast with *batch capture*.
NOT_PRIMARY_IN	A relationship operator that is used to identify requirements that can be at the origin of a link, but do not.
NOT_SECONDARY _ IN	A relationship operator that is used to identify requirements that can be at the termination of a link, but do not.
numeric attribute	A user-defined attribute type that holds numeric values, such as reference numbers. The numbers can include decimal points.
numeric sort	A method of sorting that is used for alphanumeric attributes such as paragraph numbers in outlines. For example, with a numeric sort, the numbers (10, 20, 1, and 2) are sorted as (1, 2, 10, 20) instead of (1, 10, 2, 20). Contrast with *alphabetic sort*.
object	Synonymous with *requirement*.

Object Editor	A Dimensions RM dialog used to modify attributes and edit, focus, and expand class information.
OLE	Object Linking and Embedding. A technology for transferring and sharing information among applications.
optional attribute	An attribute for which you can use the default values or leave blank. Contrast with *mandatory attribute*.
ORACLE_HOME	The logical pathname of the file system or network location of your Oracle installation.
parent collection	A collection that links child collections. Parent collections cannot be linked directly to an object.
parent database	A database at a main contractor's site used to hold the project's original objects. Used in the context of database partitioning only. Contrast with *child database*.
parent object	An original object that produces a new object when the original object is edited. The original object is called the immediate parent object, and the new object is its immediate child object. If this process is repeated, a child object can itself be a parent object of another child object. In this way, the original parent object can spawn multiple levels of descendants, including both immediate child objects and lowest-level child objects. One or more parent objects can produce one or more child objects.
pending change request	A change request that has not yet been accepted or rejected. A pending change request has a Current Status of "proposed."
polling	A feature of RM Browser that lets you solicit feedback about a requirement from selected users. Polls are typically used to decide whether a specific requirement should be accepted, or to reach consensus concerning the content of a requirement.
primary class	The first class in a direct relationship between two classes. For example, in the relationship *Is_Tested_By* that connects class *Code_Module* and class *Acceptance_Tests*, *Code_Module* is the primary class, *Is_Tested_By* is the relationship, and *Acceptance_Tests* is the secondary class. The direction of the relationship arrow is always from the primary to the secondary class in the class definition diagram. This direction and positioning on the diagram defines the direction of the relationship. Contrast with *secondary class*.
PRIMARY_IN	A relationship operator that is used to identify requirements that are at the origin of a link.
primary object	An instance of a primary class to which objects are linked.
project	A Dimensions RM work area where information is created and maintained.
proposed	Current Status of a requirement for which a change request has been made to either change the current requirement or create a new requirement.
pseudonym	Text pattern used to locate objects to be linked to a collection. For example, pseudonyms of the keyword "calibrate" might be "calibrated," "calibrating," or even the wildcarded string "cal*". Unlike aliases, which exist for the duration of the project or until they are deleted, pseudonyms exist only during the linking process. See also *alias*.

PUID Project Unique Identifier. An intrinsic attribute.

query A filter or script, expressed in terms of the project schema, that you use to retrieve selected requirements.

quick search A feature of the Requirements View in RM Browser that lets you quickly create a query to see the content of any category.

Reject A command that rejects a proposed change. The Current Status of the proposed requirement becomes "rejected," and a copy of the requirement is created with the Current Status of "current."

rejected Current Status of a change request that has been rejected.

relationship An association between two classes. It is also an entity in its own right, in terms of having its own attributes and associated user access rights. See also *link*.

relationship attribute A property of a relationship, such as its cardinality and its inheritance characteristics. Relationship attributes, defined by the Project Administrator, can control how traceability is established across different relationships. Project Administrators can specify that links be created between two objects according to the value of one or more of the class attributes. For example, it can be specified that links can be created from a change request object to a requirement object only if the value of the change request object's attribute APPROVAL_STATUS is APPROVED. See also *cardinality rule*.

relationship rule Circumstances under which links between objects will be permitted. See also *cardinality rule*.

Remove A command that physically removes a requirement from a project. Only requirements with a status of "current" can be removed.

Replace A command that creates a new version of a requirement with the changes you made. The Current Status of the original requirement is changed from current to replaced, a parent-child link is created from the original requirement to the new requirement, and the current status of the new requirement is set to current.

replaced Current Status of a requirement that has been replaced by a newer version.

resource category A grouping of resources into a class of items. For example, a unique document name falls into the category of Documents. Resource categories are important when assigning default permissions because defaults are assigned to entire class of resources rather than an individual resource. Resource categories are also important when assigning appropriate transaction for a class of resources. Certain transactions are appropriate only for certain categories of resources.

requirement An instance of a class. A description of a set of conditions applicable to a product or process; this description must be capable of being validated for success. A requirement object is satisfied by a product or process if a test reveals that the described conditions are met by the product or process. Synonymous with object.

Requirements View An RM Browser view that lets you view and modify requirements that are organized by category. From this view, you can run an existing filter, run an existing script, run a quick search to create a new query, and add queries that you access frequently to a favorites folder. See also *Documents View*, *Home View*, *Traceability View*.

RM Browser A Dimensions RM tool that provides Web access to a core set of Dimensions RM functions.

RM Capture A Dimensions RM tool that lets users extract objects from ASCII documents and pass them to the Dimensions RM database, where they can be analyzed and engineered as required.

RM Concept A Dimensions RM tool that tracks the requirements engineering and change processes. In RM Concept, you can capture requirements both manually and automatically, and can customize views for reporting and information gathering. To support change management, RM Concept manages requirements, reports changes, generates customized reports, and compares versions. You can run multiple instances of RM Concept simultaneously. RM Concept provides locking at the data level.

RM Explorer A Dimensions RM tool that provides an integrated Windows Explorer-like view of a project, so you can see and organize an entire project at once, not just one element at a time. RM Explorer provides a single interface for starting other Dimensions RM tools. You do not have to log in to the other tools from RM Explorer; however, if you start other tools from outside of Dimensions RM, you have to log in to each one individually.

You can use RM Explorer to manage and organize all the elements and source files of your project. RM Explorer lets you see the hierarchy of classes, collections, documents, files, and folders in your project. You can import and update data with RM Explorer, and view the attributes of all the components of your project. You can also use RM Explorer to create custom views and filters.

RM Import A Dimensions RM tool that lets you preview a Microsoft Word document as a draft document, change the description of chapters, reorganize the chapters, change the values of attributes, move attributes between chapters, and so on. When satisfied with the draft document, you can import the document into Dimensions RM as a document that can be viewed and modified in the Document View of RM Browser.

RM Import Designer A Dimensions RM tool that lets administrators design templates that users select when importing Word documents from RM Import. Templates define how to identify classes, attributes, chapters, requirements, and categories.

RM Manage A Dimensions RM tool that lets project administrators define users and groups, administer project security, configure the project database, organize data, and control user access and data routing.

RM Word A Dimensions RM tool that lets you capture, delete, and edit data directly from Microsoft Word. In RM Word, you can register documents in the database and synchronize requirements in a document with requirements in the database.

RTM_HOME A logical name for the file system location of Dimensions RM programs and data.

schema diagram See *class definition diagram*.

script A query against one or more classes. Scripts are the most comprehensive way to perform searches. They can be used to combine the selection criteria capabilities provided by complex filters, with complex link traversal, parameterization, basic calculations, and output formats. Contrast with *filter.*

script generator wizard A Dimensions RM wizard that provides a graphical interface allowing the user to specify the contents of a given report.

secondary class	An object class that is the destination of the relationship arrow from a primary class in a class definition diagram. The relationship arrow points to the secondary class. Contrast with *primary class*.
SECONDARY_IN	A relationship operator that is used to identify requirements that are at the termination of a link.
snapshot	In the Document View of RM Browser, a read-only baseline of a document.
source document	A document, typically provided by the customer, which is input to the system being developed. A source document can also be written in Dimensions RM by using an empty document and inserting objects.
source parent	The original object that was either captured from a document or input into the system. Source parent objects never have a "current" status.
source relationship	A relationship that refers to the original object in a chain of versions. Contrast with *immediate relationship*.
suspect link	A link that becomes questionable after one of the requirements in the link changes. The change could render other requirements questionable, or "suspect."
tablespace	A logical storage unit. Your project data is physically stored in one or more data files associated with a tablespace. Initially, only one file is associated with the tablespace, but you can add more files as you need them. The size of a tablespace is determined by the size of the data file or combined data files that make up the tablespace.
template	A set of rules defined by an administrator in RM Import Designer that determines how a document will be imported into Dimensions RM using the RM Import tool.
text attribute	A user-defined attribute type that holds up to 64 KB of alphanumeric, ASCII text that can span more than one line. It is suitable for long descriptions, such as the description of an acceptance test.
transactions	Actions associated with a category of resources that represent what can be done with that resource. For example, an update transaction is associated with a class definition. A user that has the update transaction for a particular class definition can change characteristics of that class definition and store them in the database.
traceability	The process of making explicit links between requirements and other entities. Traceability lets you trace the evolution of a project.
Traceability View	An RM Browser view that provides a way to select the relationships you want to trace, with requirements limited to selected baselines, documents, collections, or categories; browse through the requirements that are part of the relationships; and then print traceability reports that display the information in a visual format that is easy to analyze. See also *Documents View*, *Home View*, *Requirements View*
type	A definition of the basic properties of a set of instances of a class, relationship, or attribute.
Undelete	A command that changes a the Current Status of a requirement from "deleted" to "current."

Update A command that overwrites the content of the requirement without maintaining a record of what was changed. This option is not recommended if you need to maintain a history, or audit trail, of changes to requirements over time. All other attributes, including Current Status, remain intact.

user An individual responsible for performing basic information management tasks, such as capturing objects, including generic and CASE tool data in the Dimensions RM database, creating traceability links among requirements and other data, engineering and categorizing objects, and producing reports. An individual Dimensions RM user.

user-defined attribute An attribute that you can create for use in a specific class. See also *action attribute*, *alphanumeric attribute*, *date attribute*, *file attachment attribute*, *graphic attribute*,*list attribute*, *numeric attribute*, *text attribute*. Contrast with *implicit attribute*.

Index

S

T