Rational. Dashboard

Configuration Guide

Before using this information, be sure to read the general information under the "Notices" section on page 103.

This edition applies to **VERSION 4.0**, **Rational Dashboard** and to all subsequent releases and modifications until otherwise indicated in new editions.

TABLE OF CONTENTS

1.1 Audience

This document is intended for IBM Rational Dashboard administrators, application administrators, measurement leads, process analysts, and managers who need to understand the extent to which the IBM Rational Dashboard can be applied and tailored within their organization. All readers should be familiar with the IBM Rational Dashboard Portal. For some topics, it is helpful to have a good understanding of the integration mechanism provided by an external application, such as IBM Rational DOORS, particularly the manner in which it has been tailored within your organization.

An overview of IBM Rational Dashboard and terminology is provided in the "Getting Started Guide". It is recommended that you read that document before this Configuration Guide.

1.2 Purpose

Many organizations do not have an effective metric or measurement process, forcing managers to manage without data. Requirements metrics are key indicators of project scope, growth, stability and progress. Managers who use requirements engineering measures can spot trouble before a software project becomes a death march. By better managing requirements engineering, managers ensure that they deliver a quality product that completely satisfies customer expectations.

IBM Rational Dashboard puts software management best practices within reach of every organization and every manager. IBM Rational Dashboard contains a Built-In Library of Best Practices for project planning, Automated Collection for management reporting, and Compliance Tracking capability to allow managers to manage by exception.

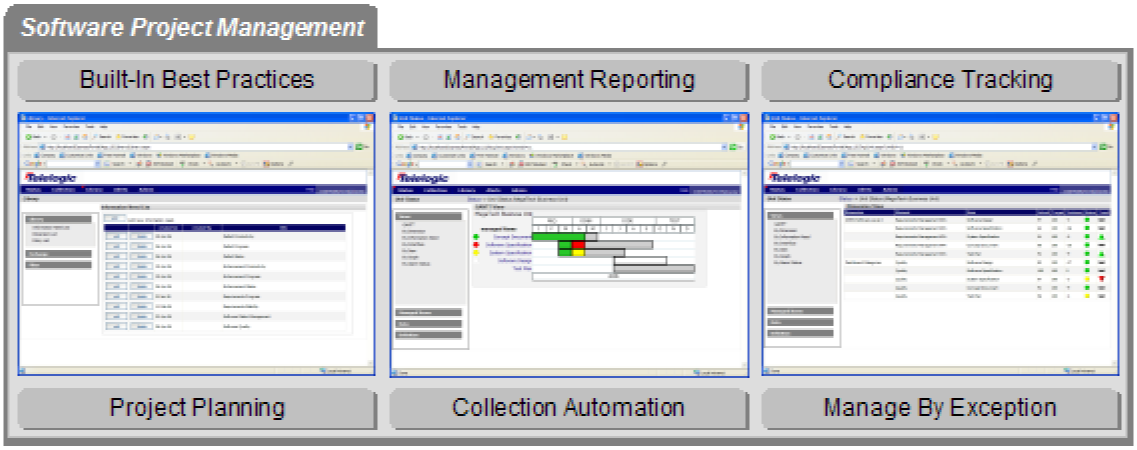


Figure 1

1.3 Architecture

IBM Rational Dashboard consists of two components which perform the functions required to collect, analyze, and report measurement data. These two components are the Portal and the Collector. The following figure depicts the relationship between these two components.

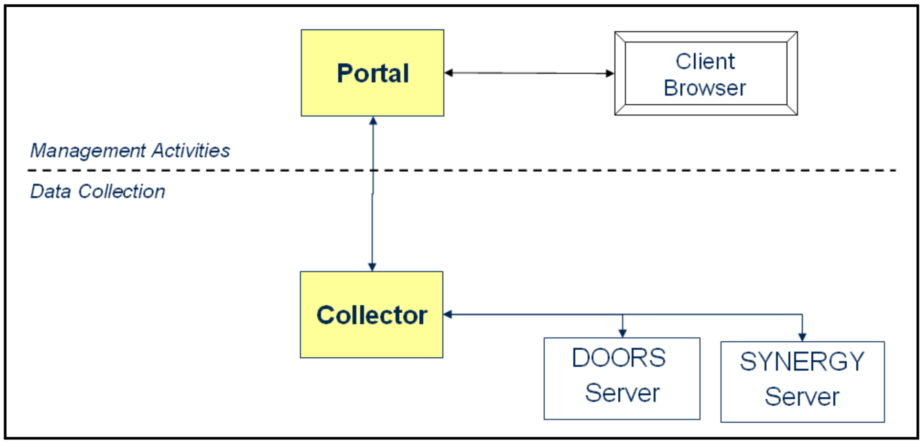


Figure 2

2 Review of Portal and Collector

This chapter provides a brief review of the functions involved in configuration. If you have not done so, it is strongly recommended that you become familiar with the Portal and the Collector.

One of the strengths of IBM Rational Dashboard is that it requires no programming or script development in order to collect data needed by managers. Instead, IBM Rational Dashboard allows you to select (or define) the reports you need, and then it automatically builds the integration needed to validate and collect the needed data.

2.1 Review of the Portal

The following subsections review the tabs in the Portal.

2.1.1 The Status Tab

The Status tab contains the organization tree, and the administrative tools for maintaining it. This organization tree contains folders, units, and items.

The tree should be built based on a company's management structure. This structure should be familiar to users, making it easy for managers to find items they need to manage. Folders and units should be organized based on who will manage items within them. Keep in mind many items are being assigned to a single manager.

An example tree for a company with five business areas, three departments in each business area is shown below:

- **Folder** – *area 1*
  - **Sub-Folder** – *department 1*
    - **Unit** – *project 1*
      - **Item** – *IBM Rational Change Defects*
      - **Item** – *DOORS Requirements Module*
      - **Item** – *Test Plan*
    - **Unit** – *project 2*
    - **Unit** – *project 3*
    - **Unit** – *project 4*
    - **Unit** – *project 5*
  - **Sub-Folder** – *department 2*
  - **Sub-Folder** – *department 3*
- **Folder** – *area 2*
- **Folder** – *area 3*
- **Folder** – *area 4*
- **Folder** – *area 5*

It is important that there are enough granularities in the tree. The right manager needs to be assigned the right items to manage. Typically, a tree defined by the work break-down structure, would provide too much granularity. The data organization is separate from the tree organization. Data can be assigned to one or more units. There is no need to define sub-tree areas for things like CMMI compliance. This is built into IBM Rational Dashboard using Dimensions.

To make access even simpler, managers see only the folders and units that they have permission to view.

The design of the organization tree has a big impact on usability by managers as well as the overall IBM Rational Dashboard performance. Below are a few recommendations. However, the Portal allows users to organize their tree in any way they feel is most beneficial to their organization. Keeping the tree as flat and as simple as possible will allow end-users to navigate and review status more efficiently.

Folders:

1. Limit sub-folder levels to 4 or fewer nested levels of folders
2. Limit top level folders to 20 (when collapsed, the tree should have less than 20 nodes)
3. No more than 200 total folders

Units:

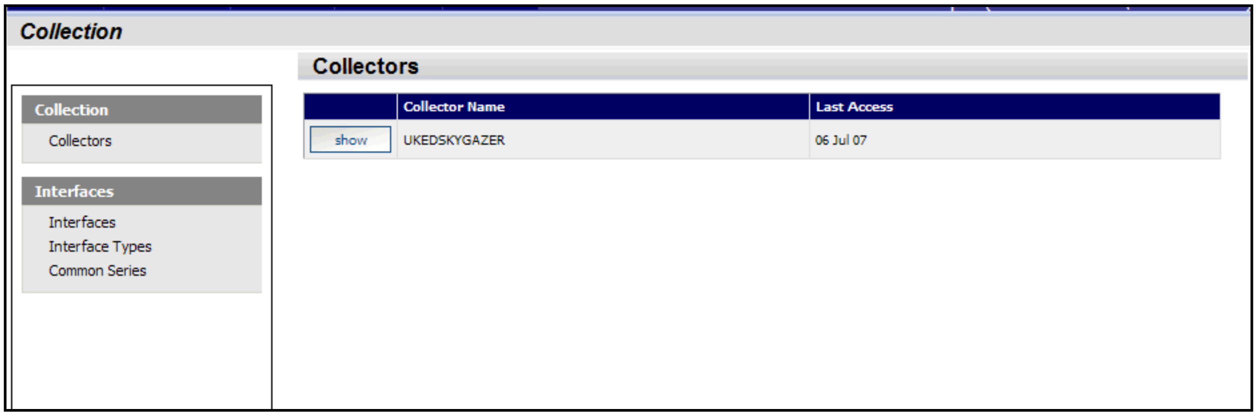
4. Limit total number of basic and summary units to less than 1000

Items:

5. Limit items to 25 items per unit

2.1.2 The Collection Tab

The Collection tab allows the review and configuration of three primary components in the data collection process: collectors, interfaces and interface types. Access to this tab is granted to all Administrator users, as well as users belonging to the "Collection Role".



Collectors

	Collector Name	Last Access
show	UKEDSKYGAZER	06 Jul 07

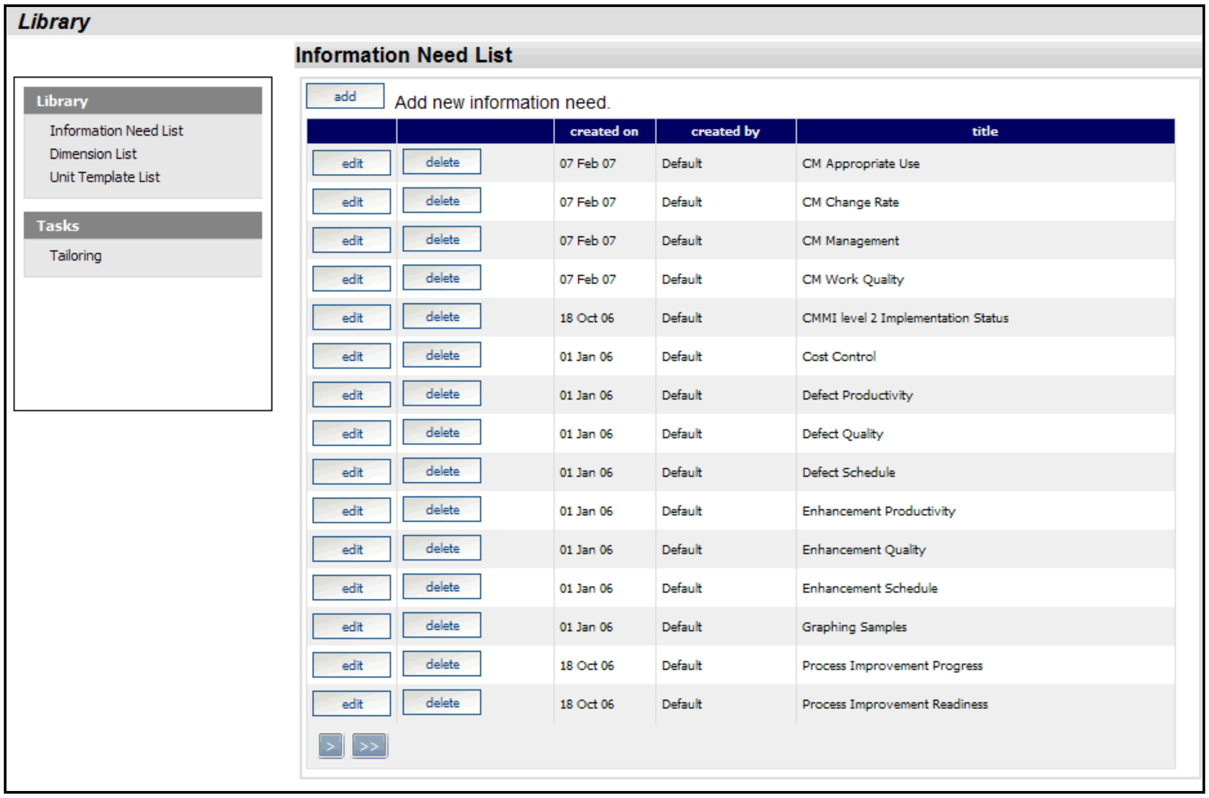
Figure 4

A Collector is a Windows Application that provides the collection interface to an external tool. Each time a Collector starts, its name is registered in the list of Collectors in the Portal.

An Interface contains the instructions for translating the information needed for graphs into the interface or protocol of an external tool. The interface describes what we want to collect from each type of external application. For example, an Interface for IBM Rational Doors defines the attributes needed for collection – the Collector translates this list of attributes into DXL and runs it so that no programming is required.

2.1.3 The Library Tab

The Library of Best Practices allows users to define or select Information Needs. Dimensions provide a method for organizing the status of information needs, for example, by maturity model areas, strategic initiative or corporate focus area. Information needs can be added to Unit Templates, along with attributes, forms, roles, users, default schedules and phases, creating samples or organizational measurement policies. Using unit templates simplifies the process of deploying new projects.

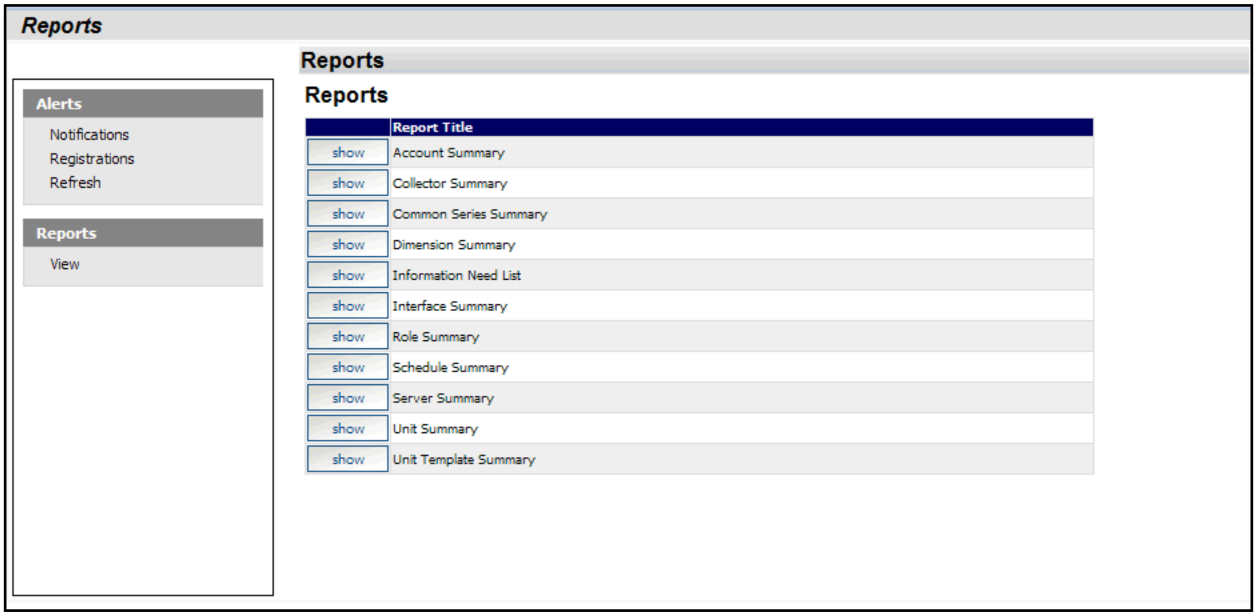


		created on	created by	title
edit	delete	07 Feb 07	Default	CM Appropriate Use
edit	delete	07 Feb 07	Default	CM Change Rate
edit	delete	07 Feb 07	Default	CM Management
edit	delete	07 Feb 07	Default	CM Work Quality
edit	delete	18 Oct 06	Default	CMMI level 2 Implementation Status
edit	delete	01 Jan 06	Default	Cost Control
edit	delete	01 Jan 06	Default	Defect Productivity
edit	delete	01 Jan 06	Default	Defect Quality
edit	delete	01 Jan 06	Default	Defect Schedule
edit	delete	01 Jan 06	Default	Enhancement Productivity
edit	delete	01 Jan 06	Default	Enhancement Quality
edit	delete	01 Jan 06	Default	Enhancement Schedule
edit	delete	01 Jan 06	Default	Graphing Samples
edit	delete	18 Oct 06	Default	Process Improvement Progress
edit	delete	18 Oct 06	Default	Process Improvement Readiness

Figure 5

2.1.4 The Reports Tab

The Reports tab allows users to register for notifications that will provide them with specific alerts following a collection. Alert types include: collection complete, collection complete with errors or with new items, folders or Units created or deleted and alarm change to yellow or to red. An Alert notification list is present in the Alert Tab following a collection if the user has registered for alerts.



	Report Title
show	Account Summary
show	Collector Summary
show	Common Series Summary
show	Dimension Summary
show	Information Need List
show	Interface Summary
show	Role Summary
show	Schedule Summary
show	Server Summary
show	Unit Summary
show	Unit Template Summary

Figure 6

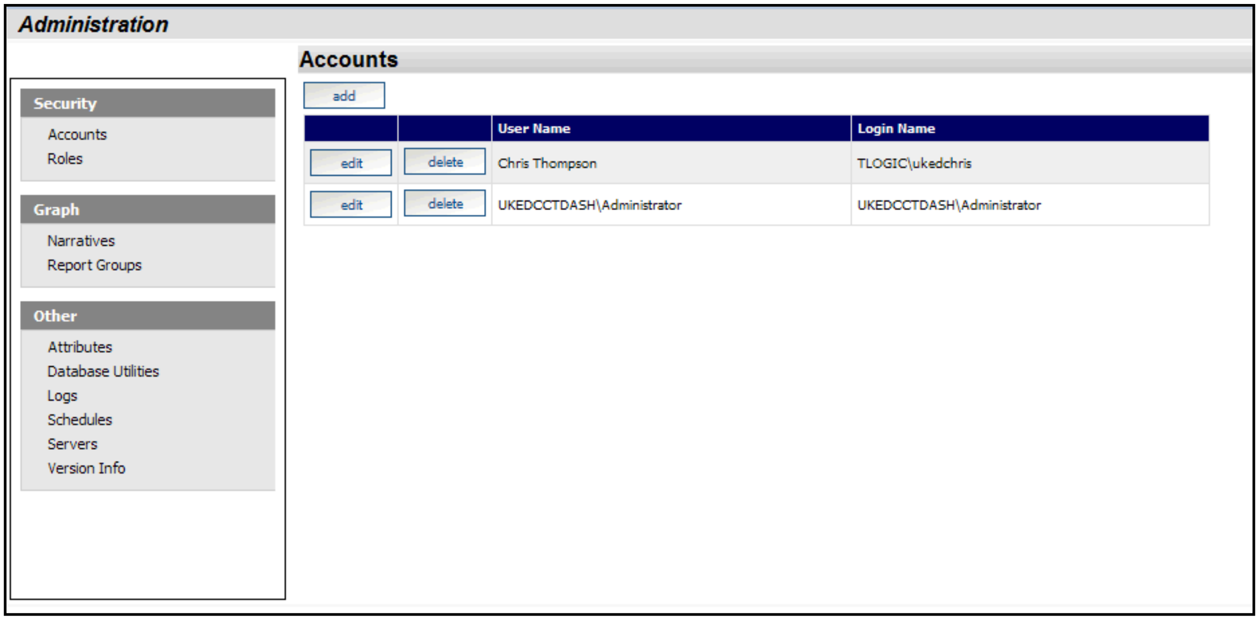
Alerts can be sent via email using SMTP. Each Alert email contains a URL that navigates to a page providing additional information on the associated alert.

IBM Rational Dashboard provides a set of built-in reports that provide visibility into the definition and configuration of the measurement process. The reports give up to the minute information on all aspects of IBM Rational Dashboard such as: Interfaces, Collectors, Information Needs, Units, Unit Templates, Accounts, Roles, and Servers. Reports can be printed to a Windows printer or exported to RTF, Excel, or Acrobat/PDF format. Access to reports can be restricted to users with particular Roles.

To add custom reports to the Portal they must first be designed using Crystal Reports Designer 11.5 (which is not included with the Portal) and then placed in the Accessories\Reports folder within the IBM Rational Dashboard website on the IBM Rational Dashboard server. A facility is provided on the Admin Tab to manage Report Groups (which are containers for these report design files) and to setup the associated access controls.

2.1.5 The Admin Tab

The Admin tab allows the administrator to customize the application to represent the unique needs of their organization. Administrative functions performed on this tab include: adding accounts, roles for security, narratives to capture information on graphs, report groups, attributes to be used for filtering information, schedules and servers. The administrator can also view version information and the Portal and Web Services log files for important troubleshooting information.



		User Name	Login Name
edit	delete	Chris Thompson	TLOGIC\ukedchris
edit	delete	UKEDCCTDASH\Administrator	UKEDCCTDASH\Administrator

Figure 7

2.2 Review of Collector

The IBM Rational Dashboard Collector is responsible for retrieving data from all applicable data sources: IBM Rational Doors (Requirements Management), IBM Rational Change, IBM Rational Synergy and other external tools. The Collector is a Windows application that provides automation to collect data from external applications. In IBM Rational Dashboard, you do not write scripts, extractors or integration code – the Collector generates and runs the appropriate integration method for the external applications that you define.

You can install as many Collectors as required by your computing infrastructure such that each Collector provides access to the required external applications. For example, you may use one Collector to collect defect and requirements data and another collector to collect cost and schedule data.

It is important to understand that the Collector follows collection “rules” that have been defined in the Portal. For example, the information for **what** to collect is provided to the Collector by **Interfaces** defined in the Portal. Information on **when** to collect is provided to the Collector by **schedules** defined in the Portal. Finally, the list of **units** (or projects) to collect **data** for is defined in the organization tree in the Portal.

The Collector requires a DOORS and/or Synergy Windows client from which to extract data. The DOORS or Synergy repository and server can exist on any of the other IBM Rational supported platforms (e.g. Linux or UNIX). IBM Rational Dashboard will require one DOORS or Synergy license to extract data.

2.2.1 Data Sources

In the Collector, a Data Source represents an Interface configured for data collection. Each data source allows the Collector to collect from an interface, such as DOORS or Microsoft project, by providing any required configuration data. This configuration data could include a user name, password or the location of an executable. Each type of interface requires different configuration data which is then stored in a data source.

For example, you might create a data source to collect peer review data from a set of CSV files located in one folder. You might create another data source to collect cost data from a set of CSV files in another folder. You create a data source for each unique type or set of data that you have to collect.

Data sources can be run interactively or on a periodic basis (e.g. every Saturday). The Collector command line allows you to specify the name of a data source so that you can run them one at a time.

2.2.2 Collection

The Collectors primary purpose is to connect to external applications, collect data and forward that data to the Portal via Web Services. The collection commands in the Collector allow you to select which data source(s) to run, to start and stop a collection, and to select which date, or dates, to run the collection for.

When the Collector is run, at least one calendar date must be specified. When the calendar date matches a collection date, the Collector collects and stores data from one or more external applications. Briefly, a collection date is one of three dates in what is called a schedule period, which consists of a start period date, and end period date and a collection date. The Collector allows you to use today’s calendar date, to select a single date or to run for a range of dates between and start and stop date.

2.2.3 Logging

The Collector provides a user interface that identifies whether errors occurred during collection. In addition to the information displayed on screen, there is a detailed logging system that captures information that occurs as collection is performed. The Collector log file is located in the main folder of the Collector and is called “logfile.txt”. It can be viewed using the File -? Logging command and also opened using a text editor.

3 Overview of the Collection Process

Data collection is one of the key aspects of the measurement process. Configuring and monitoring data collection involves a number of concepts explained in this section.

To start this section, the collection process is presented in terms of the control and then data flow.

3.1 Control Flow in the Collection Process

This subsection describes how the collection processed is controlled. By “control”, we mean the mechanisms that determine which external data sources to assess and which cause data collection to occur.

Control flow involves the following steps:

1. The users runs the Collector to perform data collection process, either using the user interface or through the command line. When performing data collection, the user selects which data source(s) to run (this is called the “selected data source”) and which date is to be used as the collection date (this is called the “selected collect date”).
2. The Collector requests from the Portal the interface associated with the selected data source. This step ensures that the Collector knows the required fields to collect for the selected data source.
3. The Collector initializes the selected data source(s), validating parameters and checking the system configuration for required components and files (when possible).
4. The Collector requests that the selected data source connect to the external application and retrieve a list of containers and items. (Note: some external applications do not support the concept of containers.) This is the “current list of items.”
5. The Collector requests from the Portal a list of items that have been previously (this is the “previous list of items”) found for the associated data source. This list indicates contains all items and identifies those items that have a schedule period where the period collect date matches the selected collect date.
6. The Collector compares the current list of items to the previous list of items. The Collector then determines which of the current list of items are new, that is, have not been found/collected before.
7. The Collector then builds a list of items to be collected, which contains new items and those that have a schedule period whose collection date matches the selected collect date.
8. For each item in the list of items to be collected, the Collector requests that the selected data source connect to the external application and collect data for the fields specified

in the interface. The Collector generates whatever script or interface mechanism is required to perform this task.

9. For each item in the list of items to be collected, the Collector sends to the Portal the data for the associated item.
10. The Collector requests that the Portal perform a data refresh of all units that contain items that were collected (i.e. contained in the list of items to be collected). The Portal performs a refresh for the selected collection date.
11. The Collector sends the Portal a run log that identifies the data source(s) collected (in case there was more than one) and summary information about the items and data found during the collection, as well as any errors encountered.

3.2 Data Flow in the Collection Process

This subsection describes the flow of data through the process of collection.

Data flow involves the following steps:

1. Each interface in the Portal Collection tab contains a definition of which transform database will be used to store data, and a list of field sets that define a table (in the transform database), attributes/fields in the external application and the name of a transform table field to store the attributes. Notice that the interface definition
2. As the Collector is run and performs the collection process, it returns a list of items that were found from each data source. This list is shown in the assign sub-tab of the Status page where each item is assigned to one or more basic units (which represent projects or programs).
3. In the Status tab, the Portal is used to assign a schedule to each item assigned to a unit. If the unit is using “basic scheduling mode, then all items and graphs are using the same schedule. If the unit is using advanced schedule mode, then each item and graph could have a different schedule.
4. As the Collector obtains a list of items for each data source, and then collects data for each one, it forwards the data to Web Services after each item is collected.
5. For each item from the Collector, the Web Services process the data from the Collector row by row. The Web Services validate the connection information for the transform database and opens a connection to the associated transform database.
6. The Web Services uses the definition of the associated Interface (from the Portal) to map each attribute/field in the data set to a table field in the transform database. The Web Services coerces the data type in the attribute/field into that of the transform database table field. The Web Services also inserts default values when a null value is detected in the data.
7. The Collector then requests that the Web Services performs a data refresh for all affected items, which causes the Web Services to open each affected basic unit and refresh the data in the order that was specified in the Unit Properties. Refresh includes

evaluating and running the queries, then the equations in data series, then any region equations and then finally the equations in alarm series.

3.3 Populating a Basic Unit from The Library

The Collector gathers data from an external tool. Once collected, the data is organized together with Information Needs and analyzed by managers in the Organization Tree.

Below is a high-level textual description of the flow:

- The Collector gathers data from outside sources and stores it in the Portal Database.
- Interfaces allow users to define the data coming in from the Collector and associated Queries
- Information Needs are a set of Graphs, series, status and measurement guidance that can be associated with a Unit. A Unit represents a real world item (e.g. a project) and a Unit can be associated with multiple Information Needs
- The Organization Tree allows users to display and analyze data for different Units arranged hierarchically

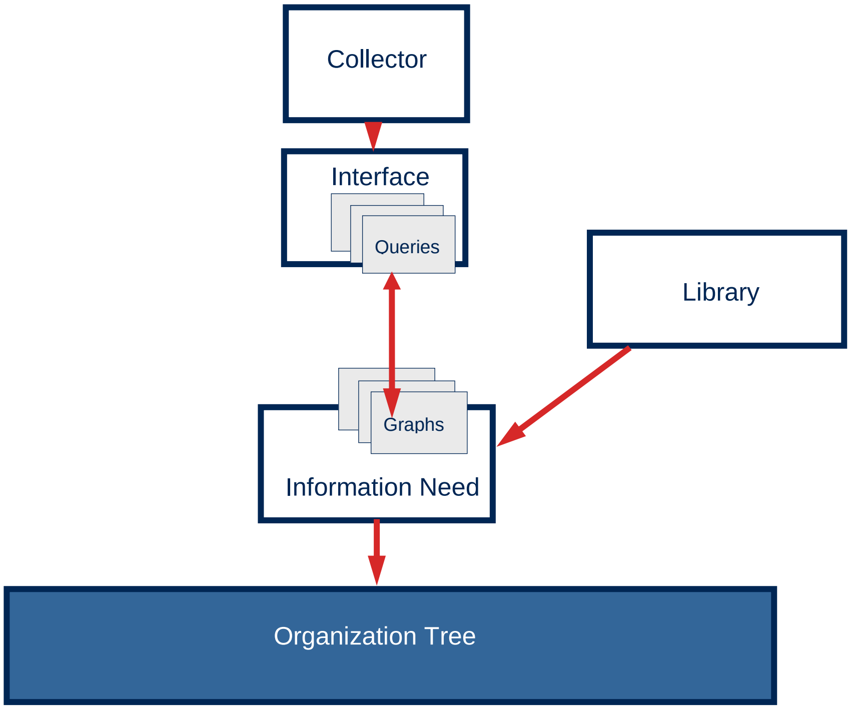


Figure 8

The first time that a collection is run, the collector will retrieve all items that could possibly have data collected. These items are considered 'new'. This results in a list of items that can be chosen to monitor in the status tab. This is the only time that data will be retrieved from all items in a tool source. Every time after that the collector will check whether the item has been assigned to a unit and whether the day of collection is one of the scheduled collection dates.

If a new item is added to the tool (such as a new module or new release) after the first collection has been run, data will be collected for the new item.

3.4 Databases

Databases are used to store the data required for the Portal and Web Services. There are two default IBM Rational Dashboard databases:

1. Portal database
2. Transform database

A portal database stores the data needed to run the Portal and Web Services, such as accounts, units, security, alerts, graphs and processed graph data. There should only be one Portal database for each IBM Rational Dashboard installation. When using a Web Farm, multiple IBM Rational Dashboard Portal servers should share the same Portal database.

A transform database acts as a data mart, storing data collected from external applications by the Collector. The transform database is used by the Portal as its source of data when performing data calculations or analysis. For example, when using a Defect Growth information need, collected defect data is stored in a transform database, which is then counted to construct (for example) the number of new defects during the last month. The resulting "counted" data is stored in a portal database.

The structure of a Transform database is maintained by the Portal. As you define data in the Portal (by adding fields to interfaces for example), tables and fields are automatically added, modified or dropped from the Transform database. You can turn off this feature in the Portal user interface if you want to maintain the Transform database manually.

You may define more than one transform database to separate data based on any appropriate factors, including capacity, performance or security. Initially, you should use just one transform database. Over time, you can add transform databases. The only limitation of using multiple transform databases is that you cannot combine data which resides in different transform databases, unless you manually create Views.

3.5 Interfaces – What To Collect and How To Process

An interface describes what data to collect, where to store the collected data and how to analyze (i.e. query) the stored data in order to generate one or more data points. Each interface is associated with an external application or file format, called an interface type, such as Microsoft Project, IBM Rational Doors or a set of CSV files.

The main communication between the Portal and the Collector is defined by **Interfaces.** An Interface essentially tells the collector ***what*** to collect and ***how*** to use the information once it is collected. You create an interface for each external application you're using and for each variation of fields within the same external application.

For example, in DOORS, different projects may be using differently structured DOORS modules to capture requirements (e.g. different attributes, types or values). These projects could require different Interfaces to support the same Information Need (e.g. Requirements Progress).

Before data can be collected, there must be an Interface describing the data to be collected that can be provided for to the Collector. The Collector reads the interface definition, generates the appropriate integration, validates that the data is available and then collects data from the external application. This mechanism results in not having to perform any script or software development in order to gather data for the measurement process.

More information on how interfaces are used is contained in the next section.

3.5.1 General Properties of the Interface

Each interface contains the following general properties.

Name	description
Title	a unique name that identifies the interface within the Portal
interface type	indicates the type of external application, database or file format that this interface will be used to process data
database server	selects a SQL database (defined on the Admin tab) where data will be stored. Notice that the table within the selected database is specified in each field set (since there may be more than one field set within an interface)

The interface type must be selected from a list of available types within the Portal. The list of available interface types is read-only and cannot be modified except by update of the Portal and Collector software. New releases often contain new interface types, so check with the support site for new/modified interface types.

To simplify creating a new interface, the Portal allows you to copy the field sets, fields and queries from an existing interface that you select. When you copy an existing interface, all the field sets, fields and queries that exist in the current interface are deleted and the new ones created from the selected interface. Some comments on using the copy interface:

- Before using the copy command, you should select the interface type and save the interface.
- You cannot change an interface type after the initial save, but you can use the copy command to copy one interface into another one of a different type.

3.5.2 Overview of Field Sets

The Field Set defines what source attributes or fields are to be collected and where the resulting data is stored. The field set consists of the following:

- A title
- A database table where the data for this field set will be stored
- A replacement tag
- A list of fields

The interface provides the database server and the table name is defined in the field set. Each field set should use a different database table name to ensure that data is not combined. When the interface is saved (and the "add" and "update" checkboxes are selected), the Portal will create the specified field set table, and then verify that each field in the list of fields.

The field set definition is used by the Collector when collecting data from a data source. For example, you may decide to collect field1, field2 and field3 and store this data in table called 'Defects'. In this case, the field set could be called "Defect Set", the table "Defects" and there would be three fields defined in the field set.

An interface may contain more than one field set. The purpose of having multiple field sets is to allow the collection of multiple sets of data from the same source. For example, you would define two field sets to collect both cost and resource data from Microsoft project, or defect and workspace data from IBM Rational Synergy. The Collector will collect as many sets of data for each interface as defined by the number of field sets.

3.5.3 Overview of a Field Set Parameter Replacement Tag

Each field set in an interface has an optional parameter replacement tag, which can save a text string. The parameter replacement tag is a mechanism to pass data from the portal to the collector at collection run time. The use of this tag is optional -- many field sets do not use them. When the parameter replacement tag is used, the value set in the interface is passed to the Collector which inserts the parameter replacement tag text into the 'Get Details SQL' field in place of the !SETTAG! reserved word. A different parameter replacement tag can be entered for each field set in an interface.

For example, the parameter replacement tag is used with a IBM Rational Synergy interface to differentiate between the ccm queries used for each of the five field sets. When the Collector collects for a IBM Rational Synergy data source, it will process each one of the five sets individually. When the Collector gets the details for the first field set, it replaces !SETTAG!, saved in the 'Get Details SQL' field, with the parameter replacement tag text for the first IBM Rational Synergy field set. When the Collector gets the details for the second field set, it replaces! SETTAG! with the parameter replacement tag text for the second IBM Rational Synergy field set. It proceeds in this manner until all five field sets for the interface have been processed.

The parameter replacement tag is also beneficial for other tool types where it is necessary to collect multiple sets of information from the same database.

3.5.4 Overview of a Field

Each field set consists of one or more fields. The field performs three purposes, as follows:

1. Identifies the name of an attribute, field or common in an external application that is required for data collection
2. Describes the data type and validation criteria for the field
3. Identifies a database table column (in an IBM Rational Dashboard transform database) where this field is to be stored

Each field that you create will be extracted from an external application (unless the "Do not collect?" option is selected, which is described below). The source attribute name must exactly match the name given to the attribute in the external application. For example, if the field in Microsoft Project Server 2003 is called "task_act_start", then this field must be defined the same. In IBM Rational Doors for example, some object attributes have spaces in the name, such as "Object Heading" – the field in IBM Rational Dashboard should be entered with the embedded space. Do not use leading or trailing spaces or other white space. The order in which the fields are defined does not matter.

Each field also contains a table field name. This is the name of the SQL table field where the resulting data will be stored. Each field in the external application is stored in a table field whose name is specified here. The table field name allows you to use more meaningful or descriptive names (instead of the name used by the external application) In addition, the table field name insulates your queries from name changes in the external application, since all the queries are written against the table field name and not the external attribute names.

Using the "null" option controls whether the transform database table will be defined to allow a null value or not. If a null value is not allowed, then the external application must provide a value or you must select the option to provide a default value. If the value from the external application is null and no default is provided, then the entire record (that is, the data for all fields) will fail when it is inserted.

The field contains an option for “Do not collect?” – this option is used to define and validate your interface, fields and queries in manageable pieces. For example, you might define all the fields in your external application and then set all but two of them to “Do not collect?”. Once you have validated the collection of data and checked the queries, you could turn on other fields using the “Do not collect?” option. For example, you might then work on the date fields in your external application. By using this flag, you can methodically work through all the aspects of an integration.

A field set with no fields is simply a place-holder. If this field set is run by the Collector, no data will be collected. Except in initial setup, you should define one or more fields in a field set.

When the interface is saved (using the “Save” button at the bottom left of the page, IBM RATIONAL DASHBOARD will make sure that the fields in the interface are contained in the transform database.

3.5.5 Interface Queries

In addition to fields, each interface contains a set of queries. There is one set of queries for all field sets and fields in the interface. An interface query defines the processing needed to generate a data point. A query is the first opportunity in the Portal to analyze data and generate a data point.

To allow the rules for generating a data point to be modified, queries use the SQL language. SQL is a fairly simple language to use and provides excellent capabilities for data analysis. Also, the Portal provides a query builder that helps develop basic SQL queries, and also an editor for writing complex SQL queries.

There are two types of queries, described in the following table.

Type	Purpose
Single series	Generates a data point in a schedule based graph
	Generates a set of records in a snapshot grid
Multi series	Generates a set of series (dynamically) based on the contents of the data, and then generates a data point for each series

During data processing, Web Services will automatically execute the queries that are associated with each interface. For example, the Collector finds a list of items from an interface and sends them to the Portal. An administrator will assign that item to a unit. In the Unit, the information needs associated with the item contain graphs whose series are queries. Once this set of assignments is made (items to a unit, and queries to an interface), IBM Rational Dashboard takes care of running the queries automatically as new item data is collected.

Queries are associated with series in Library graphs. For example, in the Library, The Information Need for "Defect Status" contains a graph called "Defect Growth" that contains a series called "Total Defects". In addition, there are two interfaces, IBM Rational Change and an Excel file, that can provide defect data for this information need. To configure both interfaces to provide data for the "Total Defects" series, we associate the queries in each interface with series in the Information Need. This allows you to define Information Needs in the Library without making them dependent on specific interfaces or queries.

3.5.5.1 Query Builder

The query builder allows you to quickly and easily build queries that have a fairly basic definition. As shown below, the Edit Query page provides various selectable options to build a SQL query statement.

The first line, "1 Data from:", displays the name of the table in the first field set in the interface.

The second line, "2 Result is", allows you to specify whether the records resulting from the SQL query should be the count of the records, or the sum of a selected field.

The "count" option is the most common, where you are interested in counting the number of data records that match a set of filter criteria (in SQL these appear in the "WHERE" clause). When you select "count", the resulting value is the number of records that match.

The "sum" option is used when the value you want is the sum of one of the fields in your data set. You create needed filter criteria (to select the appropriate records) and the "sum" a single field to create the resulting value. For example, you could apply the "sum" option to the field "planned_hours" to generate the total number of planned hours for a period. If you need to sum more than one field, you must use the Query Edit sub-tab and enter the SQL in by hand.

The third line, "3 Filters are:", lets you add common filters to the SQL query.

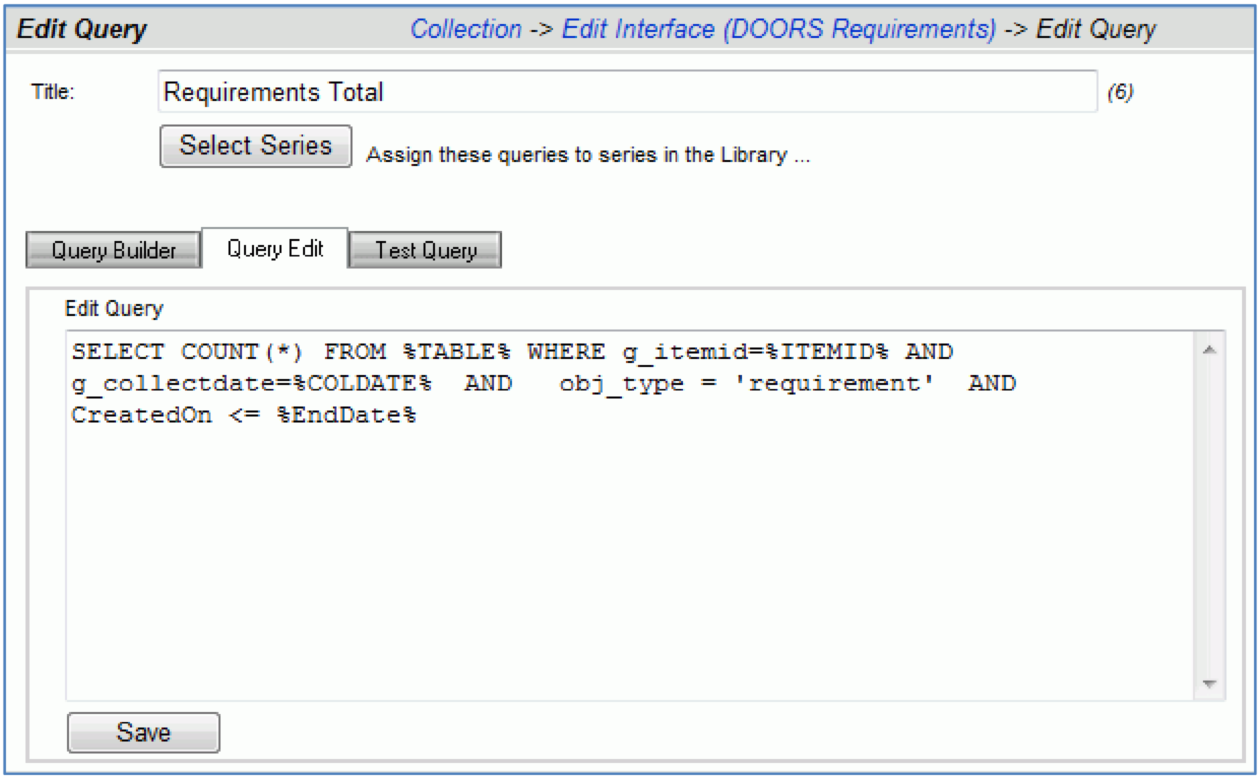
The option "the current item" adds a WHERE clause term to the SQL statement that selects only records that were collected from one item, the item assigned to the current unit.
The option "the current period" adds a WHERE clause term to the SQL statement that selects only the records that were collected during one collection period.

The "4 With terms:" section allows you to add additional terms tot eh SQL WHERE CLAUSE using the "AND" operator. To add a term, you select a field from the drop down titled "Add a new term". Then, you select a comparison operator and provide a value. A value can be provided by either entering a value into the edit box either or by selecting a tag and then pressing the "Insert" button. To place the new term into the query, press the "Add" button on the left.

To use an "OR" operator, you must use the Query Edit sub-tab.

3.5.5.2 Query Edit Page

The Query Edit page, accessible from the Query Edit sub-tab, allows you to directly enter or modify the SQL for an interface query. The figure below shows the Query Edit sub-tab for the query created in the Query Builder.

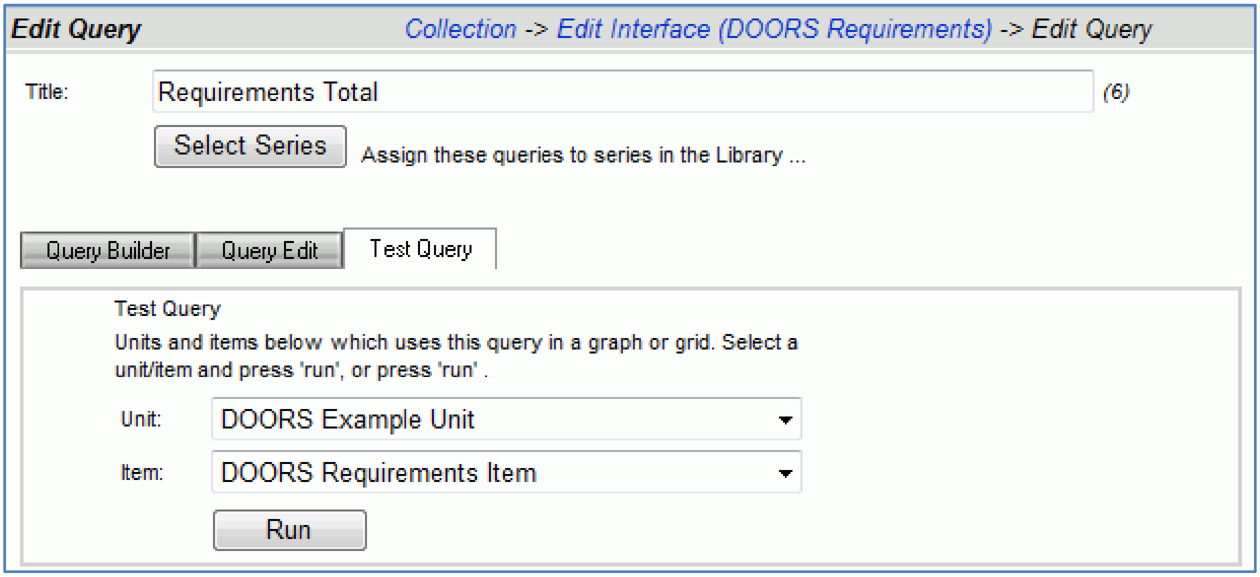

Edit Query
Collection -> Edit Interface (DOORS Requirements) -> Edit Query
Select Series
Assign these queries to series in the Library ...
Query Builder
Query Edit
Test Query
Edit Query
Save

Title: Requirements Total (6)

```
SELECT COUNT(*) FROM %TABLE% WHERE g_itemid=%ITEMID% AND
g_collectdate=%COLDATE%  AND   obj_type = 'requirement'  AND
CreatedOn <= %EndDate%
```

3.5.5.3 Query Test Page

The Query Test page allows you to test an interface query, and to explore the database table used by the interface. In order to test a query, the query should be run against a table that contains data. To get actual data, you must select a unit and item that uses the same interface that contains the selected query. As shown in the screen below, the Test Query sub-tab requires that you select a unit and item before trying to test the query.



Edit Query — Collection -> Edit Interface (DOORS Requirements) -> Edit Query

Title: Requirements Total (6)

Assign these queries to series in the Library ...

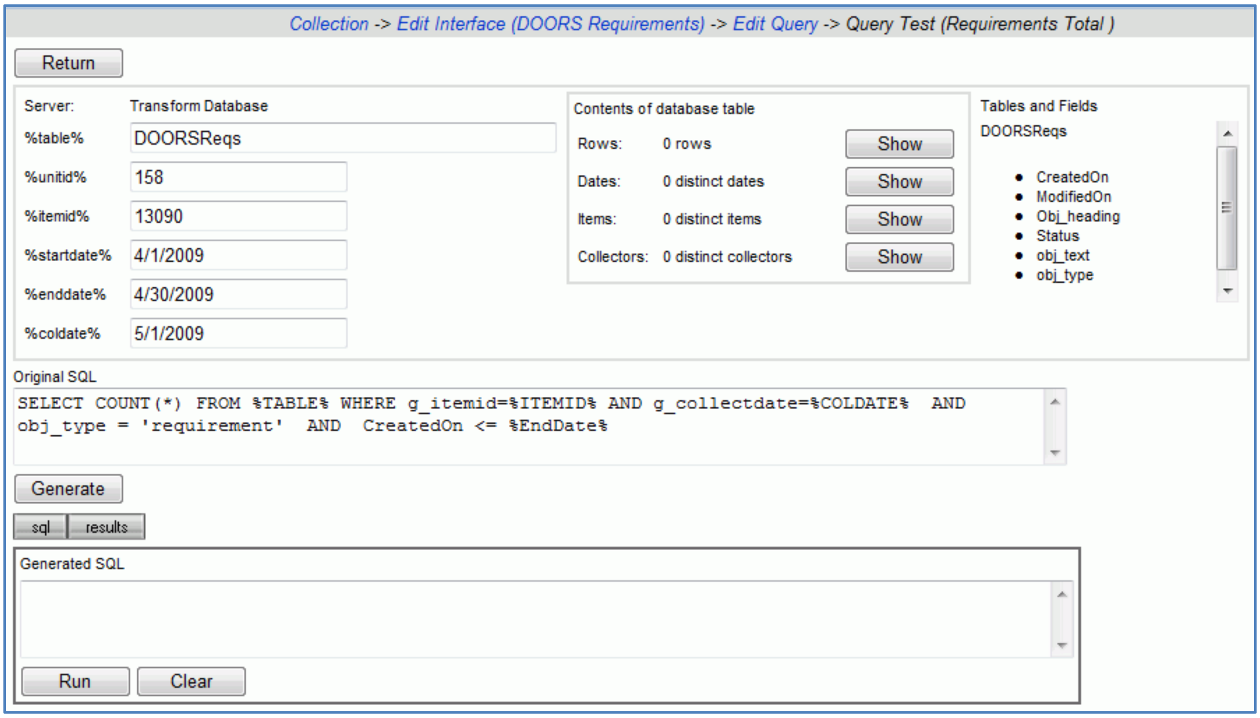
Test Query

Units and items below which uses this query in a graph or grid. Select a unit/item and press 'run', or press 'run' .

Unit: DOORS Example Unit

Item: DOORS Requirements Item

Once you have pressed the “Run” button, the query test page is shown, as depicted below. The buttons on the top left labeled “Show” allow you to explore the data that is in the specified table. These buttons help you see what data was collected, for which dates and items and which collectors collected the data. These buttons save you time when testing or debugging a query.



Collection -> Edit Interface (DOORS Requirements) -> Edit Query -> Query Test (Requirements Total)

Server: Transform Database

%table%: DOORSReqs

%unitid%: 158

%itemid%: 13090

%startdate%: 4/1/2009

%enddate%: 4/30/2009

%coldate%: 5/1/2009

Contents of database table

Rows:	0 rows
Dates:	0 distinct dates
Items:	0 distinct items
Collectors:	0 distinct collectors

Tables and Fields

DOORSReqs

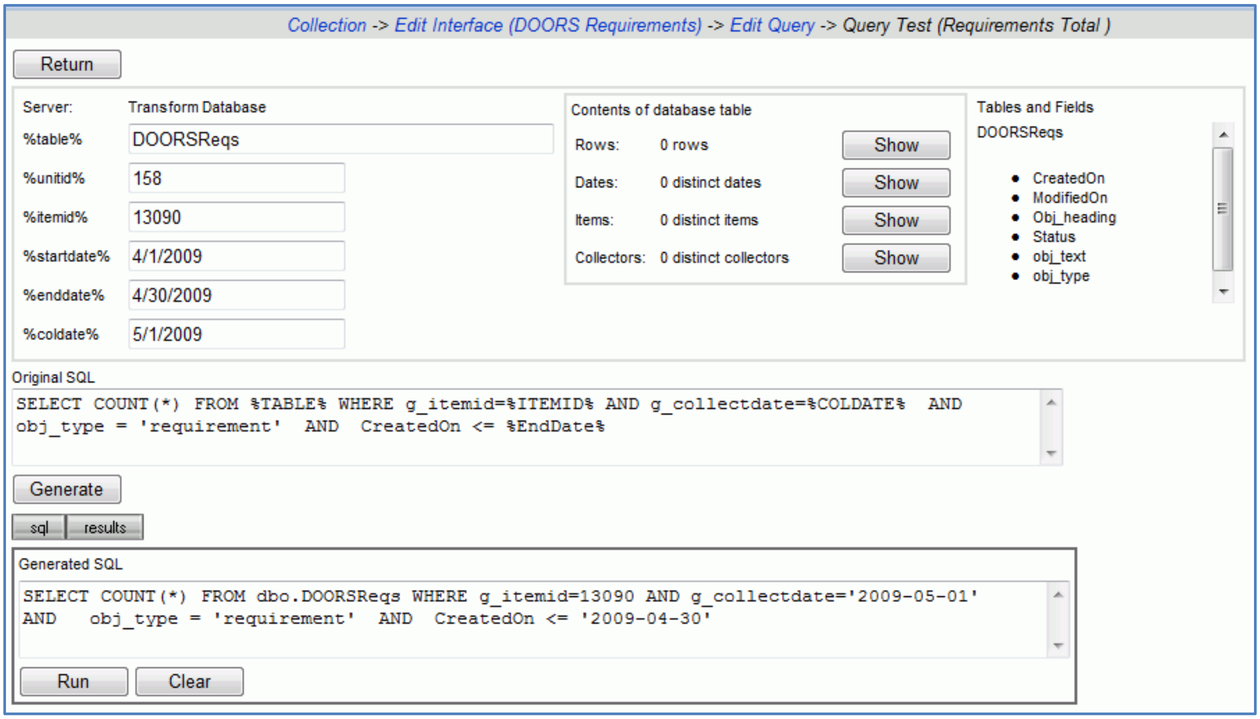
- CreatedOn
- ModifiedOn
- Obj_heading
- Status
- obj_text
- obj_type

Original SQL

```
SELECT COUNT(*) FROM %TABLE% WHERE g_itemid=%ITEMID% AND g_collectdate=%COLDATE%  AND
obj_type = 'requirement'  AND  CreatedOn <= %EndDate%
```

Generated SQL

Notice that all the common query tags are shown in two columns on the left, along with the original SQL. You may modify any of the tag values by simply changing the appropriate edit box. To see the original query with tags replaced, you must press the “Generate” button. This creates a SQL statement where all tags have been replaced. The figure below shows the query with the tags replaced.



Collection -> Edit Interface (DOORS Requirements) -> Edit Query -> Query Test (Requirements Total)

Server:	Transform Database
%table%	DOORSReqs
%unitid%	158
%itemid%	13090
%startdate%	4/1/2009
%enddate%	4/30/2009
%coldate%	5/1/2009

Contents of database table

Rows:	0 rows	Show
Dates:	0 distinct dates	Show
Items:	0 distinct items	Show
Collectors:	0 distinct collectors	Show

Tables and Fields

DOORSReqs

- CreatedOn
- ModifiedOn
- Obj_heading
- Status
- obj_text
- obj_type

Original SQL

```
SELECT COUNT(*) FROM %TABLE% WHERE g_itemid=%ITEMID% AND g_collectdate=%COLDATE%  AND
obj_type = 'requirement'  AND  CreatedOn <= %EndDate%
```

Generated SQL

```
SELECT COUNT(*) FROM dbo.DOORSReqs WHERE g_itemid=13090 AND g_collectdate='2009-05-01'
AND   obj_type = 'requirement'  AND  CreatedOn <= '2009-04-30'
```

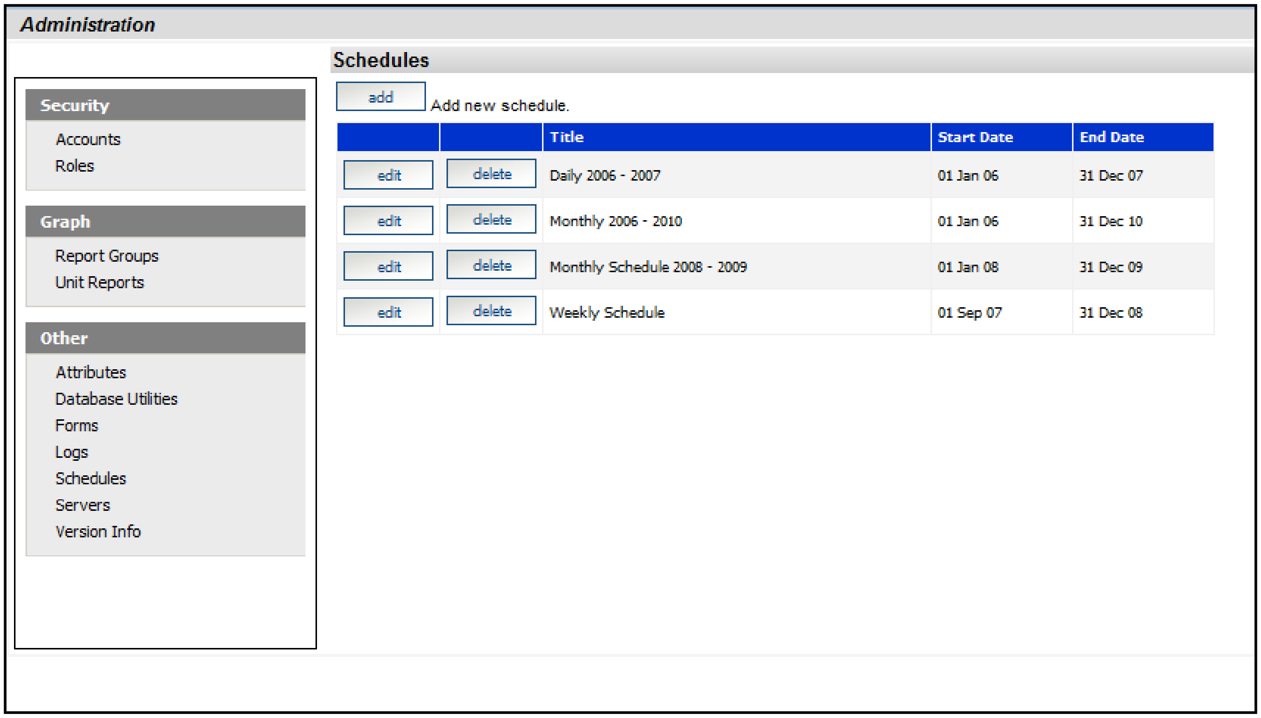
Then, you can press the “Run” button to actually run the query. The results are shown on the “results” sub-tab, either as a single record in the case of a query with a “count(*)” or “sum(field)” term, or a data table in the case of a query that returns records such as a query for a snapshot grid.

3.6 Schedules - When to Collect

Schedules are important in determining *when* data is going to be collected from a source. You may create schedules with any frequency; common options are to have collections on a daily, weekly, or monthly frequency. Each period for a schedule consists of the starting and ending dates for the period and the date that the data is actually being collected or a ‘collection date’. If a collection is run for a source, but it is not a date that any schedule has a collection date for, then no data will be collected. A schedule must be set for an item when it is created and must relate to when the data is going to be collected.

Performance Measurement data is generally gathered on a periodic basis (e.g. weekly or monthly, over the life of a project or a program). Each period has a start date and an end date. In IBM Rational Dashboard, the collection of periods and dates is called a schedule. Schedules can be defined monthly, weekly, daily, or individual dates may be added as needed.

IBM Rational Dashboard schedules are defined by administrators and available to all users to assign to a unit or an item. Some schedules may be regular (monthly or weekly) and others may be more irregular (a schedule based on project milestones or phases).



Add new schedule.

		Title	Start Date	End Date
edit	delete	Daily 2006 - 2007	01 Jan 06	31 Dec 07
edit	delete	Monthly 2006 - 2010	01 Jan 06	31 Dec 10
edit	delete	Monthly Schedule 2008 - 2009	01 Jan 08	31 Dec 09
edit	delete	Weekly Schedule	01 Sep 07	31 Dec 08

Figure 22

4 "How To" Perform Common Configuration Tasks

This section describes the how to perform various configuration tasks involving the Portal and Collector.

To successfully configure a collection these steps are required:

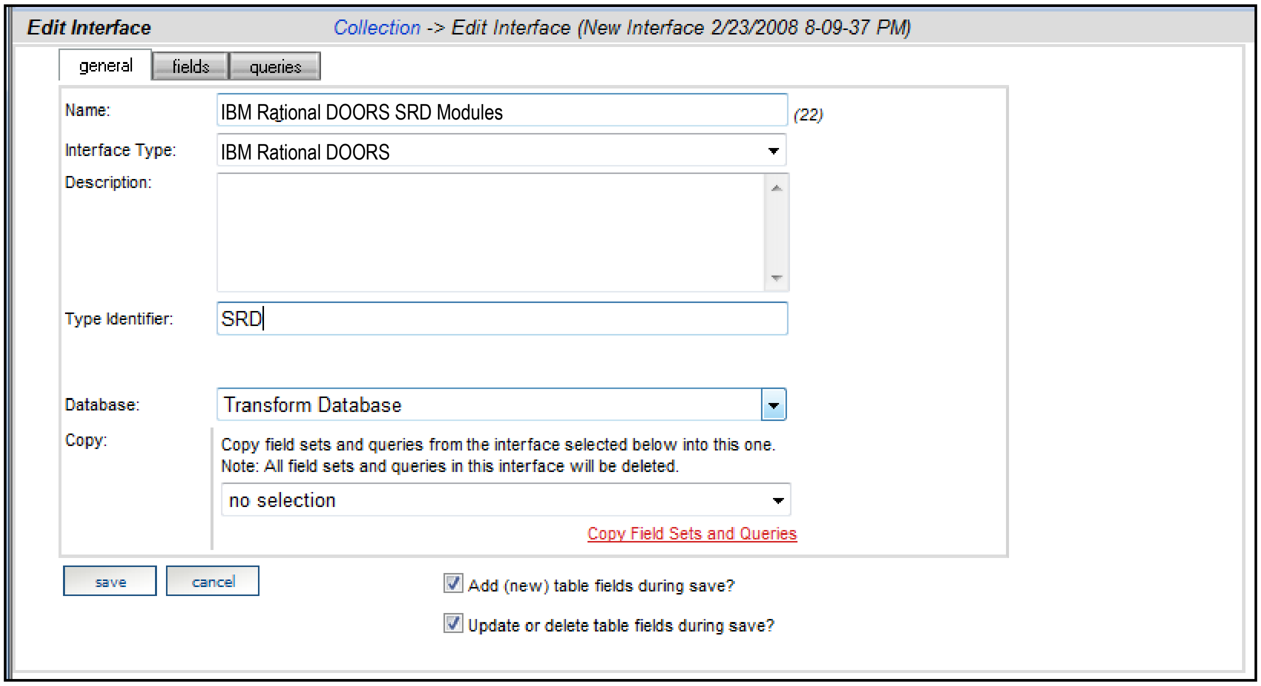
1. Modify/Define an Interface
2. Configure the Collector
3. Perform initial collection of data
4. Create a Unit
5. Assign collected data to the Unit
6. Perform subsequent collections

4.1 Defining Interfaces

The next three subsections discuss how to configure a DOORS, Synergy and Change interface. For additional information on configuring and running collection, please consult one of the Walkthrough documents.

4.1.1 How to Define a DOORS Interface

1. Login to the Portal
2. Click on the **Collection** tab
3. Click the **Interfaces** link
4. Click **add** to add a new interface
   In the Interface, information is stored about the type of data being retrieved and where it is stored (in this example a specific interface is created to collect data from modules that are based on a particular DOORS template)



Collection -> Edit Interface (New Interface 2/23/2008 8-09-37 PM)

Name: IBM Rational DOORS SRD Modules (22)

Interface Type: IBM Rational DOORS

Description:

Type Identifier: SRD

Database: Transform Database

Copy: Copy field sets and queries from the interface selected below into this one.
Note: All field sets and queries in this interface will be deleted.

no selection

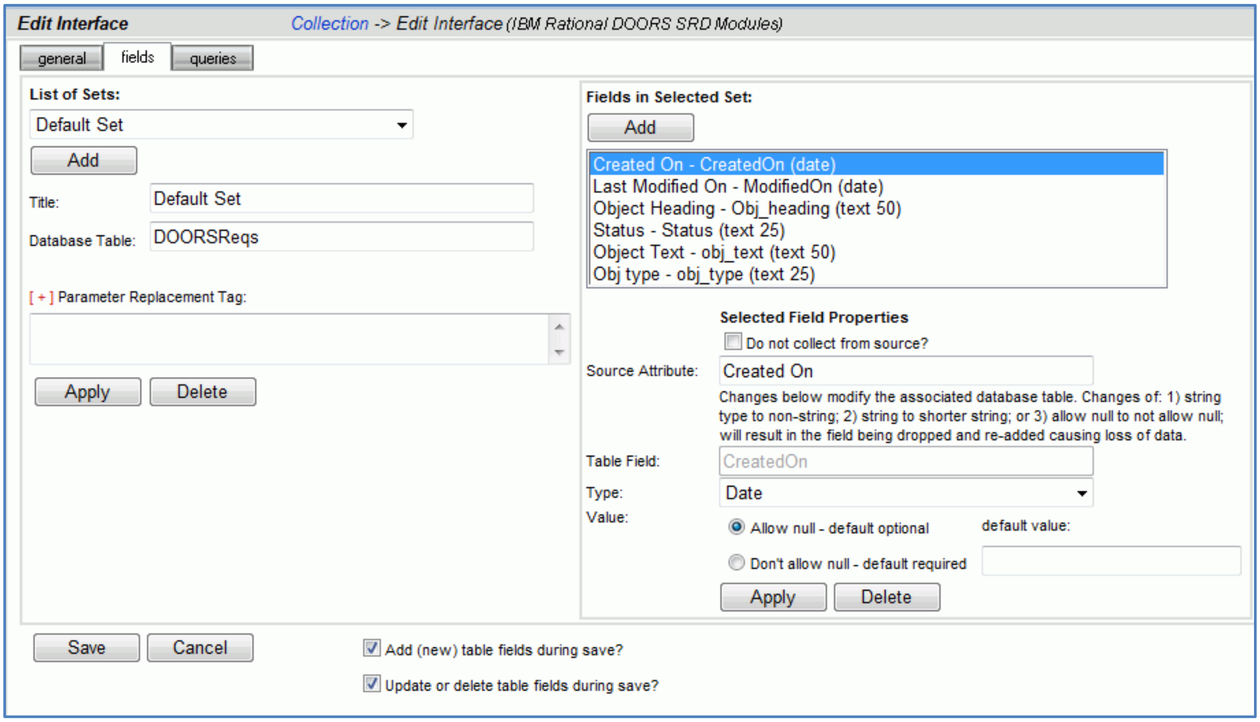
☑ Add (new) table fields during save?

☑ Update or delete table fields during save?

Figure 9

5. In the **general tab**, enter the following information for your interface:
    1. **Title** – name of the interface (e.g. DOORS System Requirements Document)
    2. **Interface Type** – the application you are gathering data from
    3. **Description** – information on this new interface
    4. **Type Identifier** – this value is used to match the Interface definition against the DOORS modules that are available for collection. The value specified here (e.g. “SRD”) is compared against the actual value of a Module string attribute (e.g. “DashboardType”) in the DOORS module. If a match is found then this Interface is used to perform the collection. The name of the Module attribute to be used (e.g. “DashboardType”) is specified in the Collector. Refer to the section on *Configuring IBM Rational Change and DOORS for Collection* for further details.
    5. **Database** – choose a database. In most cases, this is called the Transform Database. It is possible to have multiple Transform databases so it is important to choose the correct database where the information for this interface will be stored.
    6. The **Copy** option provides the ability to re-use field and query definitions from an existing interface. Choose the Interface from the drop-down list and click **apply** to delete any fields or queries already defined for the Interface and replace them with definitions from the chosen Interface.
    7. **Ensure that the checkbox “Verify and update table fields during save?” is checked**
6. Click **save**

7. Return to the interface, and select the **fields tab:**



Edit Interface   Collection -> Edit Interface (IBM Rational DOORS SRD Modules)

List of Sets: Default Set

Title: Default Set

Database Table: DOORSReqs

[+] Parameter Replacement Tag:

Fields in Selected Set:

Created On - CreatedOn (date)
Last Modified On - ModifiedOn (date)
Object Heading - Obj_heading (text 50)
Status - Status (text 25)
Object Text - obj_text (text 50)
Obj type - obj_type (text 25)

Selected Field Properties

□ Do not collect from source?

Source Attribute: Created On

Changes below modify the associated database table. Changes of: 1) string type to non-string; 2) string to shorter string; or 3) allow null to not allow null; will result in the field being dropped and re-added causing loss of data.

Table Field: CreatedOn

Type: Date

Value: ● Allow null - default optional   default value:

○ Don't allow null - default required

☑ Add (new) table fields during save?

☑ Update or delete table fields during save?

Figure 10

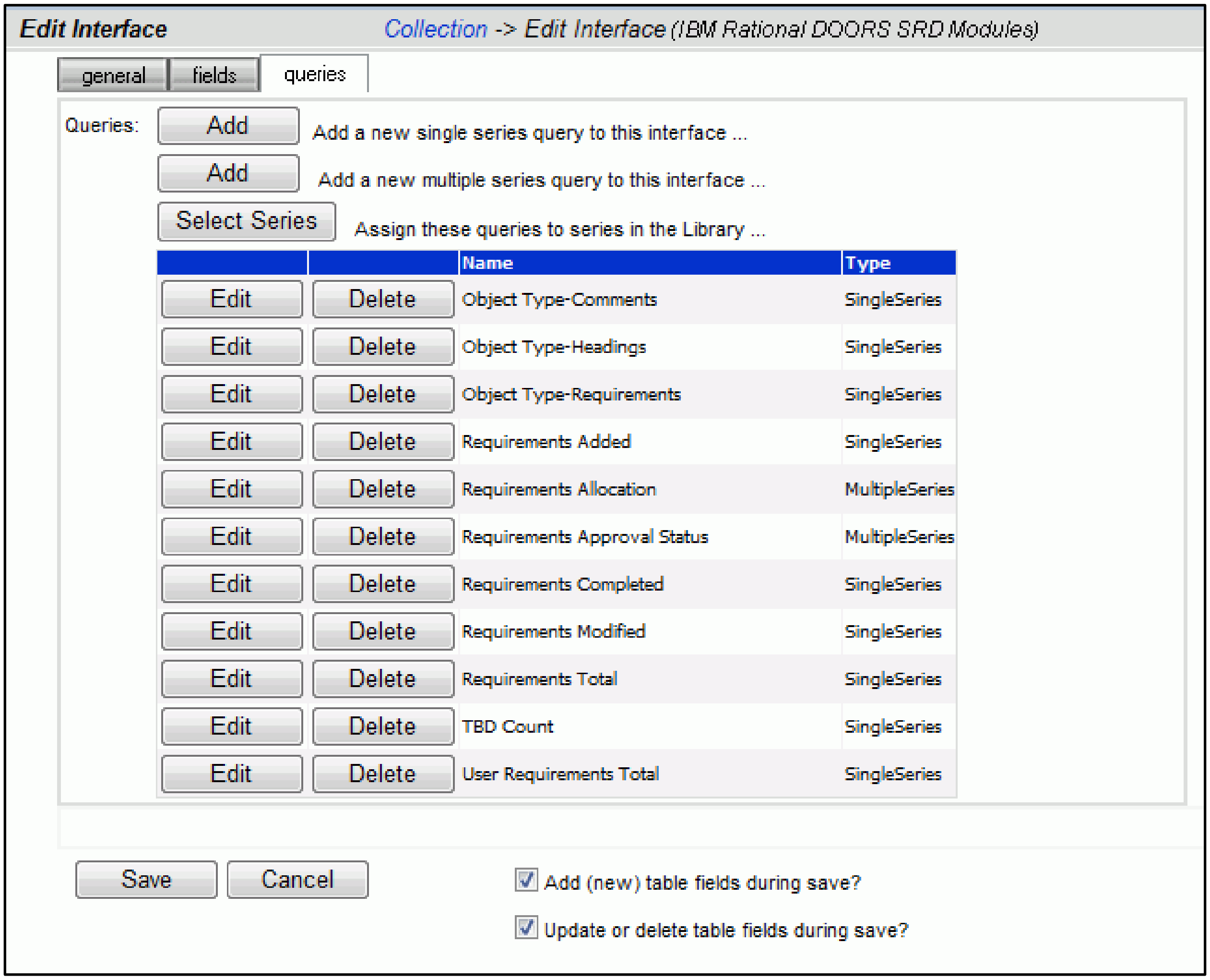
8. **Add** fields from the data source that the Interface will be collecting. *In this example these field names match DOORS attribute names in the source DOORS modules. You can use standard DOORS attributes like 'Object Heading', 'Object Text' and 'Last Modified On' as well as custom attributes e.g. 'Obj type'.*

   The Collector can obtain one or more related sets of data from an Interface. Each set of data (Field Set) is stored in a separate database table and may be queried (or joined with other tables) to provide the desired metrics. The **Title** field designates the name of the set while the **Database Table** field indicates which table in the Transform Database stores the data collected from the set. The **Parameter Replacement Tag** field is used to create an individual query for each set. When this field is used, the! SETTAG! tag is used in the Get Details query in the collector.

   For each field, you must provide a **Source Attribute** and database table field (**Table Field**) name. The Source Attribute should match the name of the column or field being collected. For example, if you are collecting from DOORS, "Object Heading" might be the name of a Source Attribute. The Table Field name is the name of a field (in a database table) where the data for this Source Attribute will be stored. The table for each field set is created automatically, provided the checkbox **"Verify and update table fields during save?"** is checked.

9. Click **save**

10. Return to the interface, and select the **queries tab:**



Edit Interface
Collection -> Edit Interface (IBM Rational DOORS SRD Modules)
general
fields
queries

Queries: Add — Add a new single series query to this interface ...

Add — Add a new multiple series query to this interface ...

Select Series — Assign these queries to series in the Library ...

		Name	Type
Edit	Delete	Object Type-Comments	SingleSeries
Edit	Delete	Object Type-Headings	SingleSeries
Edit	Delete	Object Type-Requirements	SingleSeries
Edit	Delete	Requirements Added	SingleSeries
Edit	Delete	Requirements Allocation	MultipleSeries
Edit	Delete	Requirements Approval Status	MultipleSeries
Edit	Delete	Requirements Completed	SingleSeries
Edit	Delete	Requirements Modified	SingleSeries
Edit	Delete	Requirements Total	SingleSeries
Edit	Delete	TBD Count	SingleSeries
Edit	Delete	User Requirements Total	SingleSeries

☑ Add (new) table fields during save?

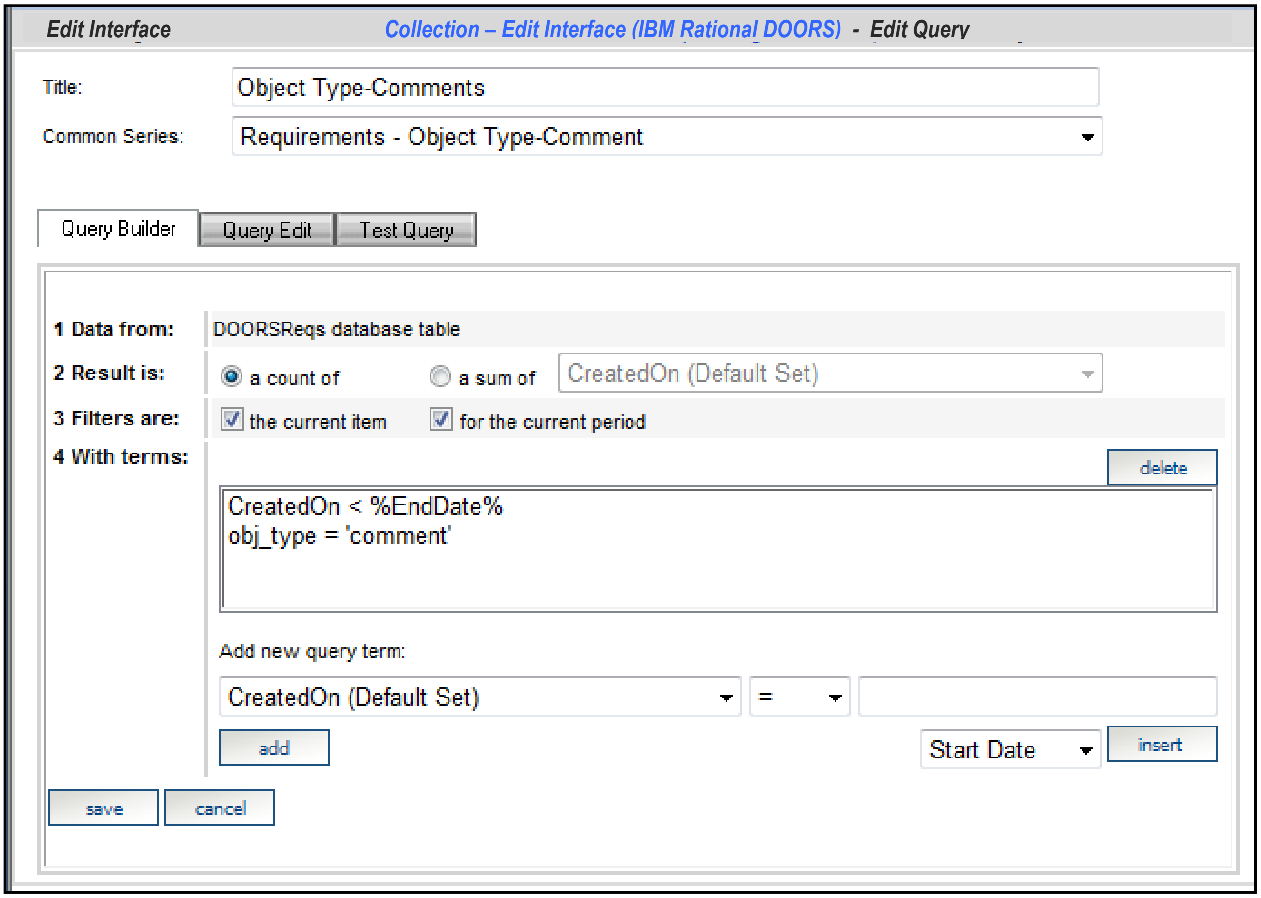
☑ Update or delete table fields during save?

Figure 11

8. The **queries** tab contains a list of queries that are defined for the interface.
9. The **add** buttons add a new query to the list which can then be edited to meet the requirements for the interface
10. The **edit** button opens the query for editing via either the Query Builder or the Query Edit dialogs. The Query Builder is the easy way to set up queries on the collected data (Interface Queries) and associate them with series in an Information Need Graph from the Library.

The following example shows how to change the way "comments" (in collected DOORS modules) are counted in this Interface.

11. Click on the **edit** button beside the query "Object Type-Comments"



Edit Interface — Collection – Edit Interface (IBM Rational DOORS) - Edit Query

Title: Object Type-Comments

Common Series: Requirements - Object Type-Comment

1 Data from: DOORSReqs database table

2 Result is: ● a count of ○ a sum of CreatedOn (Default Set)

3 Filters are: ☑ the current item ☑ for the current period

4 With terms:

```
CreatedOn < %EndDate%
obj_type = 'comment'
```

Add new query term: CreatedOn (Default Set) =

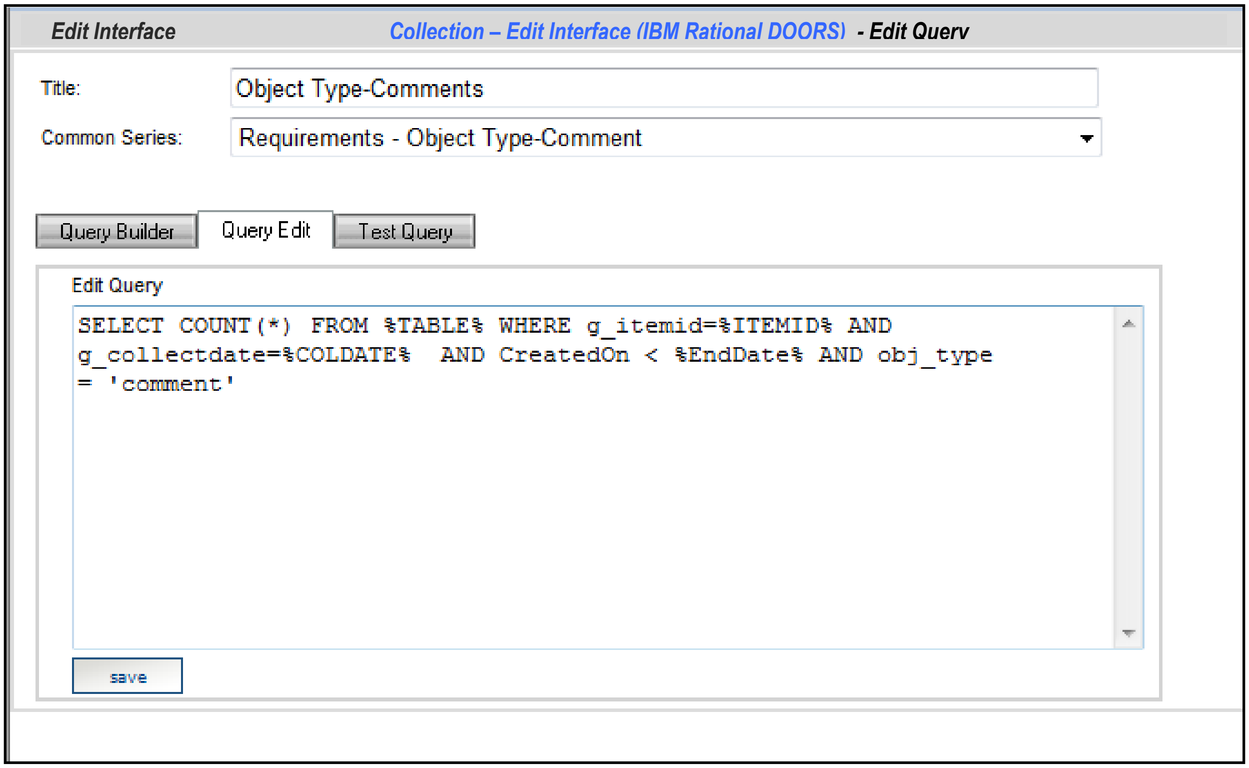
Start Date

Figure 12

12. The **Query Builder** or the **Query Edit** sub-tabs can be used to modify the query. The **Query Builder** generates SQL statements that operate on the data collected from this Interface. The **Query Edit** sub-tab may be used if the user is proficient in SQL statements or to make simple changes to the query terms. The **Query Edit** sub-tab requires correct SQL syntax for all statements.

- To generate or modify a SQL statement using the Query Builder:
    1. Select either a **count of** or a **sum of.** *In this example we want a **count of** DOORS objects which meet our criteria*.
    2. Select a **filter**, by current item and/or by schedule period. The "the current item" filter will limit the query to just the item selected. The "the current period" filter limits the query to just the data that had been collected for the time period being calculated. *In this example we want to query the **current item** (DOORS module) in the **current period**.*

3. Add (or delete) **terms** to the query. *In this example we want to modify the query to count objects where the table field obj_type = 'comments' rather than the pre-defined 'Comments' and so we will use the **Query Edit** sub-tab*
4. Click on the **Query Edit** sub-tab


Edit Interface
Collection – Edit Interface (IBM Rational DOORS) - Edit Query
Query Builder
Query Edit
Test Query
Edit Query
save

Title: Object Type-Comments

Common Series: Requirements - Object Type-Comment

```
SELECT COUNT(*) FROM %TABLE% WHERE g_itemid=%ITEMID% AND
g_collectdate=%COLDATE%  AND CreatedOn < %EndDate% AND obj_type
= 'comment'
```

Figure 13

5. Change the text 'Comment' to 'comment'
6. Click **Save**

This query will now match objects where the table field 'obj_type' (which is mapped to a custom DOORS object attribute "Obj type") contains 'comment'. If you have already collected and assigned data you may now test the query using the **Test Query** sub-tab

Two of the available SQL operators may not be familiar, "LIKE" and "IN":

- The LIKE operator enables a match to be part of a word. For example, ObjectTitle LIKE '%a' would match any object title that ended with an 'a'.
- The IN operator can be used to match multiple terms to a field. For example, ObjectTitle IN ('one', 'two') would match either an Object Title of 'one' or 'two'.

4.1.2 How to Define a IBM Rational Change Interface

1. Click on Collection tab in the IBM Rational Dashboard Portal
2. On the left hand side of the page, click on **Interfaces**
3. Click on the **edit** button to the left of the IBM Rational Change Interface. The Edit Interface page will open.
4. Click the **fields** tab
5. Click on the **add** button
6. Enter the name of the **Source Attribute** (e.g. Issues Assigned)
7. Enter the name of the **Table Field** (e.g. Issues_Assigned)
8. Enter the Type of field (e.g. text)
9. Return to the top of page and click the **queries** button
10. Click on the **add** new single series query button
11. Return to the grid and click the **edit** button for the **New Query**. The Edit Query page will open.
12. Enter the **Title** of the new query (e.g. Issues Assigned)
13. In the Query Builder, enter the following information:
    - Select the desired **result** (e.g. Count)
    - Selected **filters** (e.g. current item and current period).
14. Click on **Save** button

    Selecting these options will generate a query: "SELECT COUNT (*) FROM %TABLE% WHERE g_itemid=%ITEMID% AND g_collectdate=%COLDATE%".

    To verify the query, click on the **Edit Query** link directly above the query text box.

Now we are going to add additional filters, a start and end period. To add additional filters, use the terms option.

1. In the drop down box click **add** new query **term**
2. Select the **field name** that will be queried.
3. Select the **Operator** and then select the value from the drop down list and click **insert**. When all the values are entered, click **add**. The new query filters will display in the terms text box.
4. Click **save**.
5. To view the entire query, click on the **Edit SQL**.

In this example we have added a start and end date, the final query looks like this:

"SELECT COUNT (*) FROM %TABLE% WHERE g_itemid=%ITEMID% AND g_collectdate=%COLDATE% AND assignment_date >= %StartDate% AND assignment_date <= %EndDate%".

4.1.3 How to Define a IBM Rational Synergy Interface

1. Click on Collection tab in the IBM Rational Dashboard Portal
2. On the left hand side of the page, click on **Interfaces**
3. Click on the **edit** button to the left of the IBM Rational Synergy Interface. The Edit Interface page will open.
4. Click the **fields** tab
5. Click on the **add** button
6. Enter the name of the **Source Attribute** (e.g. Issues Assigned)
7. Enter the name of the **Table Field** (e.g. Issues_Assigned)
8. Enter the Type of field (e.g. text)
9. Return to the top of page and click the **queries** button
10. Click on the **add** new single series query button
11. Return to the grid and click the **edit** button for the **New Query**. The Edit Query page will open.
12. Enter the **Title** of the new query (e.g. Issues Assigned)
13. In the Query Builder, enter the following information:
    - Select the desired **result** (e.g. Count)
    - Selected **filters** (e.g. current item and current period).
14. Click on Save button

Selecting these options will generate a query:
"SELECT COUNT (*) FROM %TABLE% WHERE g_itemid=%ITEMID% AND g_collectdate=%COLDATE%".

To verify the query, click on the **Edit Query** link directly above the query text box.

Now we are going to add additional filters, a start and end period. To add additional filters, use the terms option.

1. In the drop down box click **add** new query **term**
2. Select the **field name** that will be queried.
3. Select the **Operator** and then select the value from the drop down list and click **insert**. When all the values are entered, click **add**. The new query filters will display in the terms text box.
4. Click **save**.
5. To view the entire query, click on the **Edit SQL**.

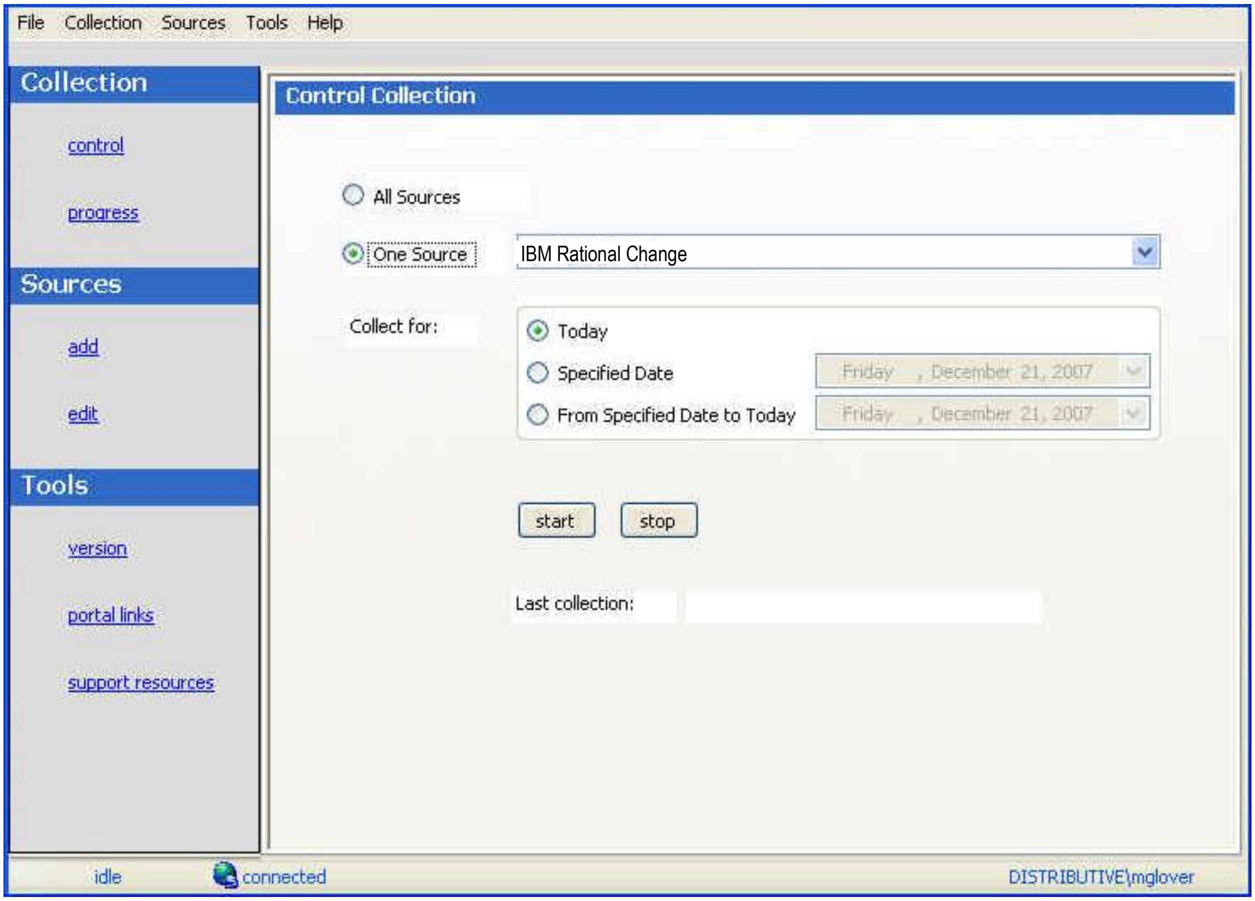
In this example we have added a start and end date, the final query looks like this:

"SELECT COUNT (*) FROM %TABLE% WHERE g_itemid=%ITEMID% AND g_collectdate=%COLDATE% AND assignment_date >= %StartDate% AND assignment_date <= %EndDate%".

4.2 *Setup the Collector*

The Collector is a tool outside of the Portal that enables collection of data from a tool such as Microsoft Project, IBM Rational Change, IBM Rational Synergy or DOORS. This data is then viewed and configured in the Portal. When you have defined your interface, you can begin the collection.

The IBM Rational Dashboard Collector can be accessed by going to:
Open the Collector using Start > Programs > IBM Rational > IBM Rational Dashboard > Dashboard Collector.



○ All Sources

● One Source: IBM Rational Change

Collect for:
● Today
○ Specified Date: Friday , December 21, 2007
○ From Specified Date to Today: Friday , December 21, 2007

Last collection:

idle | connected | DISTRIBUTIVE\mglover

Figure 14

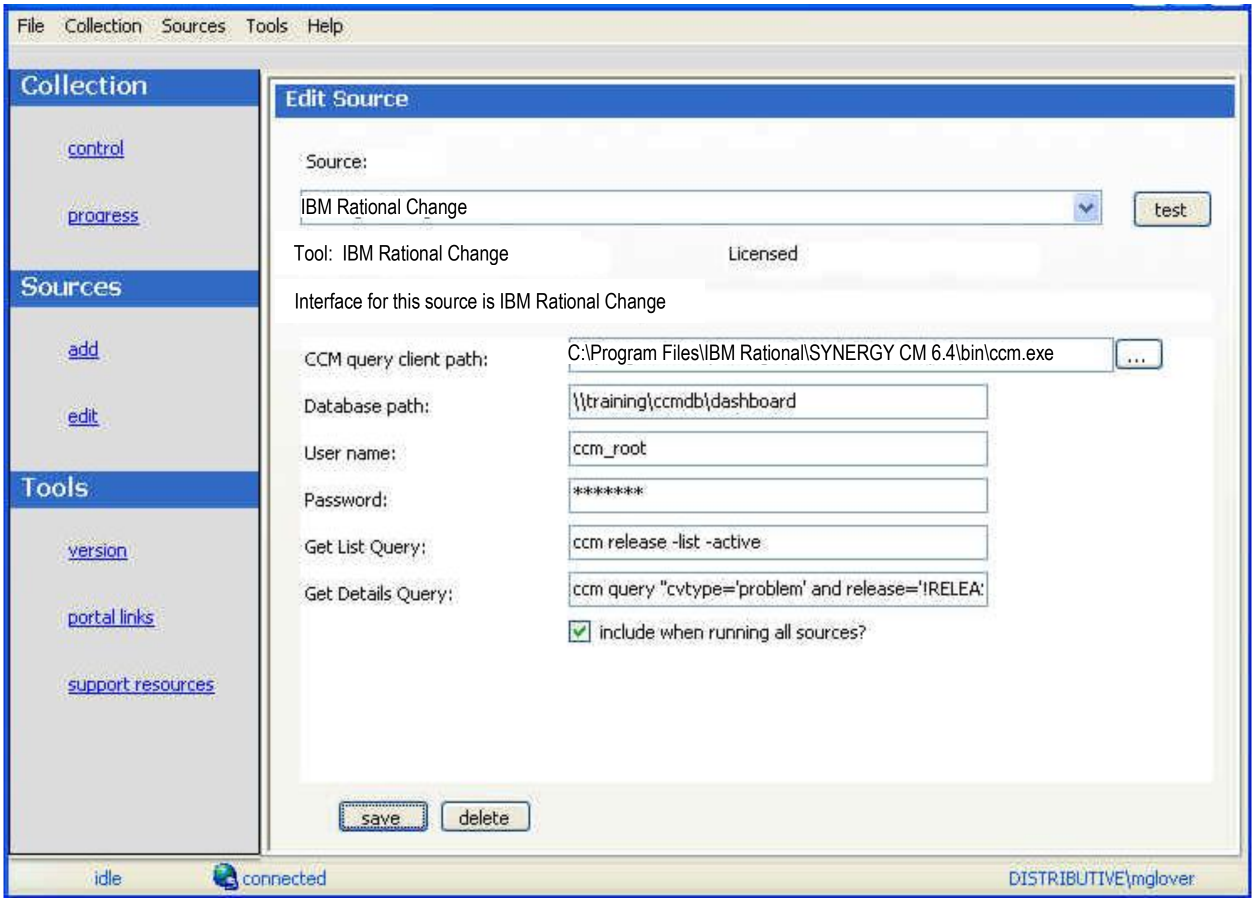
Once an interface has been set up, you can create a source in the Collector. Usually one source is enough for one external application, but if there is not one created, you should create one.

But first we have to configure IBM Rational Change or DOORS for the collection.

4.2.1 Configuring IBM Rational Change

Before collecting from IBM Rational Change, there are several prerequisites. First, you should verify that the IBM Rational Command Line Interface has been installed and is operational. More information on IBM Rational Change is available in the IBM Rational Change help documentation.

Second, in the Portal, an IBM Rational Change interface must be created before a collection can be set up. This is so that the collector will be able to delineate between different instances and databases within IBM Rational Change. Interfaces are created in the Collection tab in the portal. Once the interface is created, the source can be added to the Collector.



Source: IBM Rational Change

Tool: IBM Rational Change Licensed

Interface for this source is IBM Rational Change

CCM query client path: C:\Program Files\IBM Rational\SYNERGY CM 6.4\bin\ccm.exe

Database path: \\training\ccmdb\dashboard

User name: ccm_root

Password: *******

Get List Query: ccm release -list -active

Get Details Query: ccm query "cvtype='problem' and release='!RELEA

☑ include when running all sources?

Figure 15

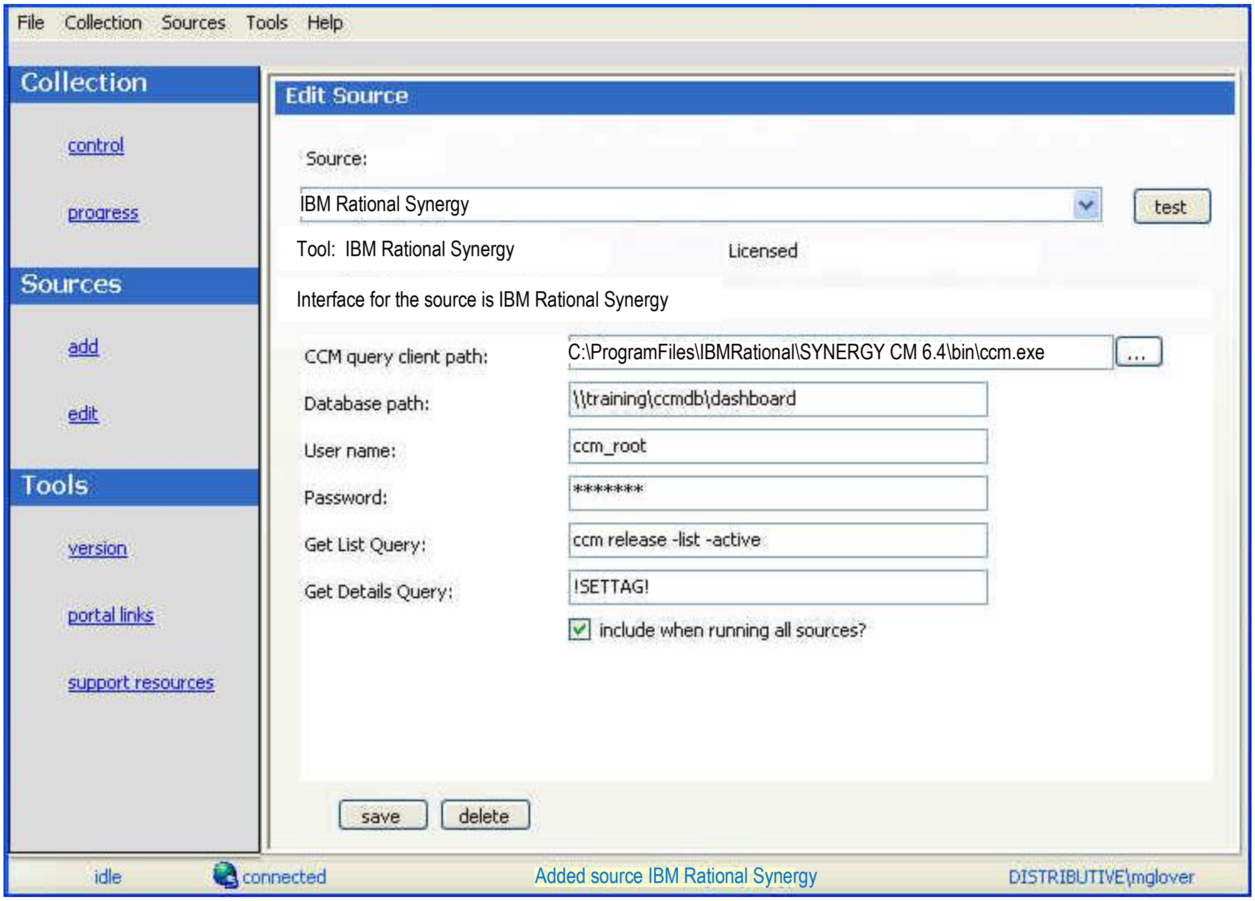
When creating an IBM Rational Change data source (in the Collector), an interface must be selected. In addition, there are several fields to fill in, as seen by the image above. The "CCM query client path" is the location of the ccm.exe file on the current computer. Click the button to the right side to browse for the file on the computer. The "Database path" is the location of the database from which data is being collected. The "User name" and "Password" are the user name and password for a database administrator who has access to all of the releases and CRs that need to be collected.

The "Get List Query" and "Get Details Query" are CLI commands to retrieve Items and Item details. The default values select all of the releases in the database and then collect from those releases all items that are listed as defects or enhancements.

4.2.2 Configuring IBM Rational Synergy

As with IBM Rational Change, before beginning a collection from IBM Rational Synergy, there are several prerequisites. First, verify that the IBM Rational Command Line Interface is operational. More information on IBM Rational Synergy is available in the IBM Rational Synergy help documentation.

Second, in the Portal, an IBM Rational Synergy interface must be created. This is so that the collector will be able to delineate between different instances and databases within IBM Rational Synergy. Interfaces are created in the Collection tab in the portal. Once the interface is created, the source can be added to the Collector.



Edit Source

Source: IBM Rational Synergy

Tool: IBM Rational Synergy Licensed

Interface for the source is IBM Rational Synergy

CCM query client path:	C:\ProgramFiles\IBMRational\SYNERGY CM 6.4\bin\ccm.exe
Database path:	\\training\ccmdb\dashboard
User name:	ccm_root
Password:	*******
Get List Query:	ccm release -list -active
Get Details Query:	!SETTAG!

☑ include when running all sources?

Added source IBM Rational Synergy

Figure 16

When creating an IBM Rational Synergy source, an interface must be selected. In addition, there are several fields to fill in, as seen by the image above. The "CCM query client path" is the location of the ccm.exe file on the current computer. Click the button to the right side to browse for the file on the computer. The "Database path" is the location of the database from which data is being collected. The "User name" and "Password" are the user name and password for a database administrator who has access to all of the releases and CRs that need to be collected.

The "Get List Query" and "Get Details Query" are CLI commands to retrieve Items and Item details. The default values select all of the releases in the database and then collect from those releases all items that are listed as defects or enhancements. You will notice that the “Get Details Query” defaults to the value “! SETTAG!” The “! SETTAG” reserved word is replaced at collection time with the value in the ‘Parameter Replacement Tag’ field in each field set of the interface that this source uses. The ‘Parameter Replacement Tag’ is useful because it allows you to specify a different ‘Get Details’ query for each different field set that you are collecting.

4.2.3 Configuring IBM Rational Doors

The DOORS collection requires two attributes to be added into the DOORS database. The attributes must be added at the ***module*** level. Adding the attribute at the DOORS Object level will cause the collection to fail. The two attributes are termed Collect Attribute and Type Attribute in IBM Rational Dashboard, but you can give them any name you like.

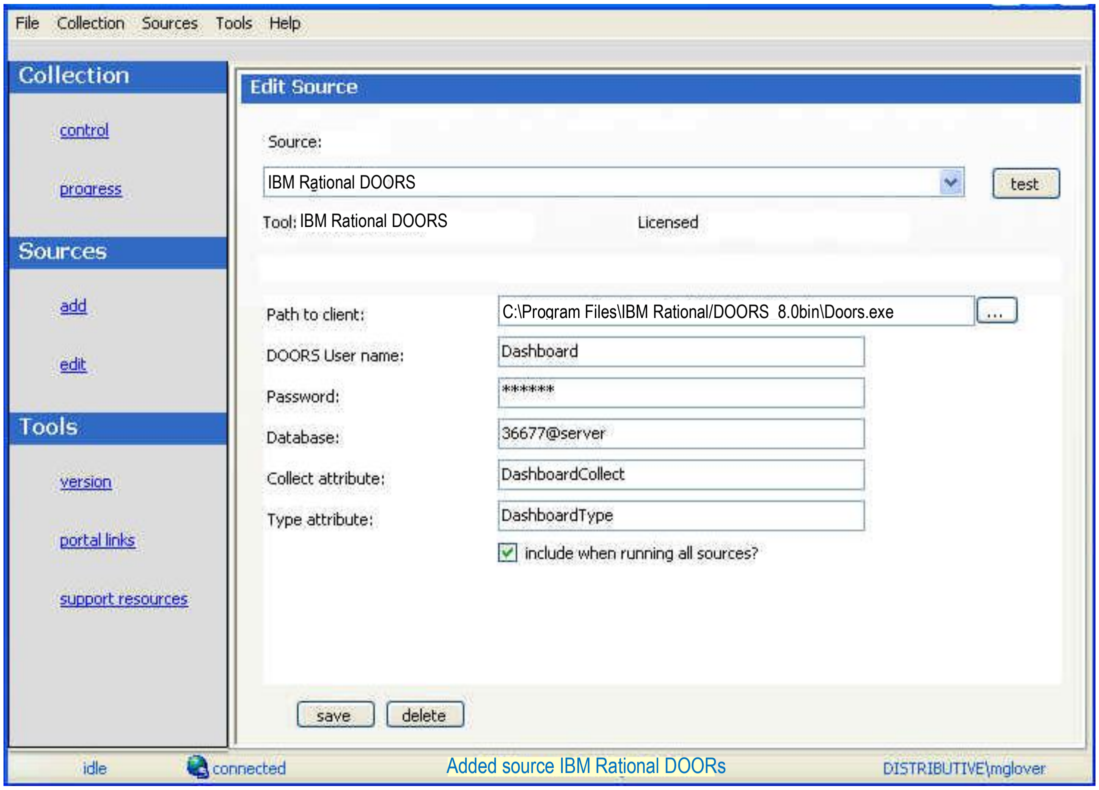
For example, you can call the Collect Attribute *DashboardCollect* and the Type Attribute *DashboardType*.

The Collect Attribute should be a Boolean value. The Collector will only collect those modules that have the Collect Attribute set to *True*. Therefore you should set the Collect Attribute value of those DOORS modules you want to collect to True.

The Type Attribute in the DOORS database should be a text value. The value should match what is put in the **Type Identifier** field in the Interface details (please refer to Interface section for more information).

The other components of the DOORS source are relatively self explanatory.

- The "Path to client" is the location of the DOORS client on the current machine. Click the button to the side to browse for the file on the computer.
- The "DOORS User name" and "Password" should be a DOORS Administrator or a user who has access to all of the projects and modules from which data needs to be collected.
- The "Database" is the port number and the server name in the format <port number>@<server>.



Source: IBM Rational DOORS

Tool: IBM Rational DOORS Licensed

Path to client: C:\Program Files\IBM Rational/DOORS 8.0bin\Doors.exe

DOORS User name: Dashboard

Password: ******

Database: 36677@server

Collect attribute: DashboardCollect

Type attribute: DashboardType

☑ include when running all sources?

Added source IBM Rational DOORs

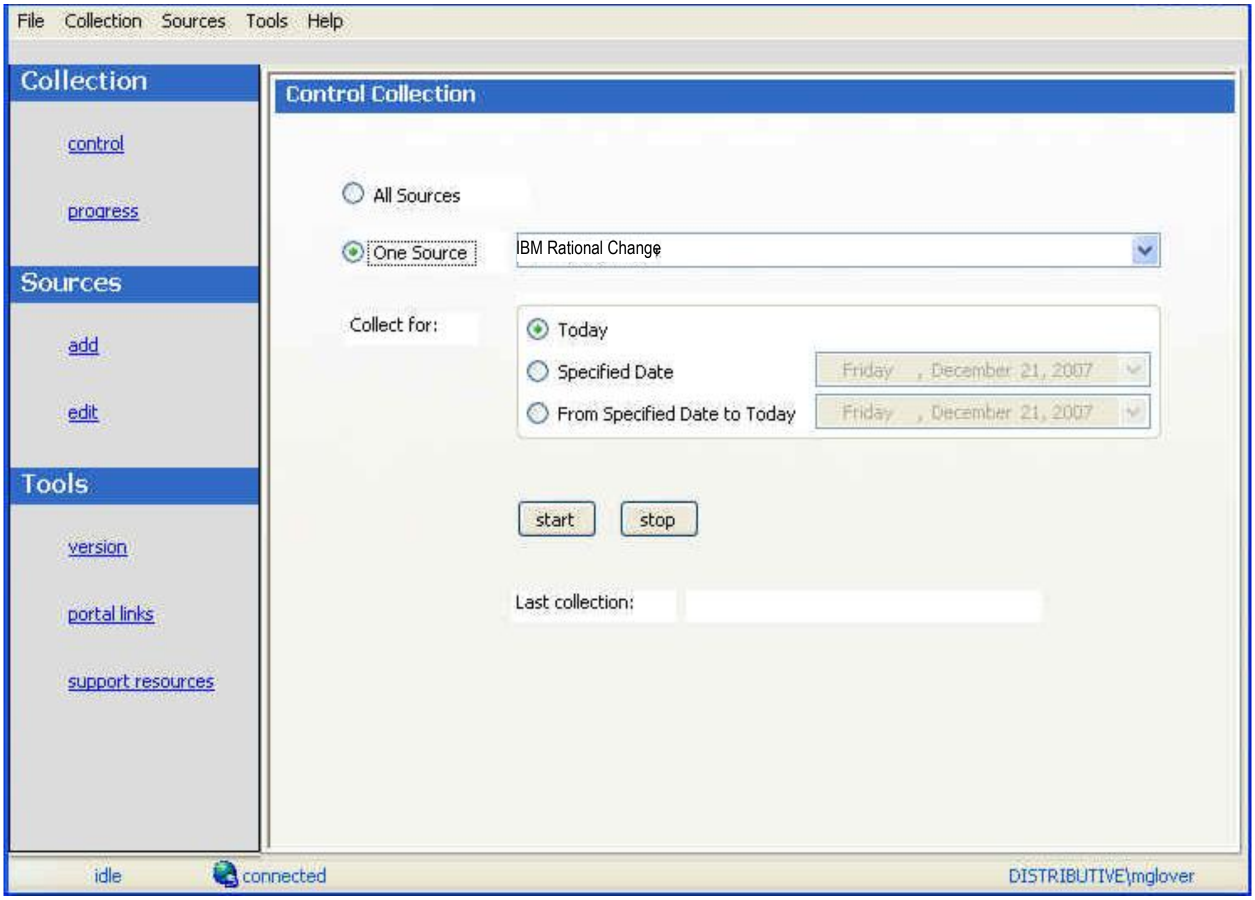
DISTRIBUTIVE\mglover

Figure 17

4.3 Perform Initial Collection

Once the interfaces and sources have been defined, we can run the collection. In the IBM Rational Dashboard Collector:

1. Click on the **Collection** tab on the left navigation pane
2. Click on the **control** link. This will open the Control Collection on the right side of the window.
3. Chose the **source** you defined for this collection
4. Click **start**. The collection will begin.



Control Collection

○ All Sources

● One Source: IBM Rational Change

Collect for:

● Today

○ Specified Date: Friday , December 21, 2007

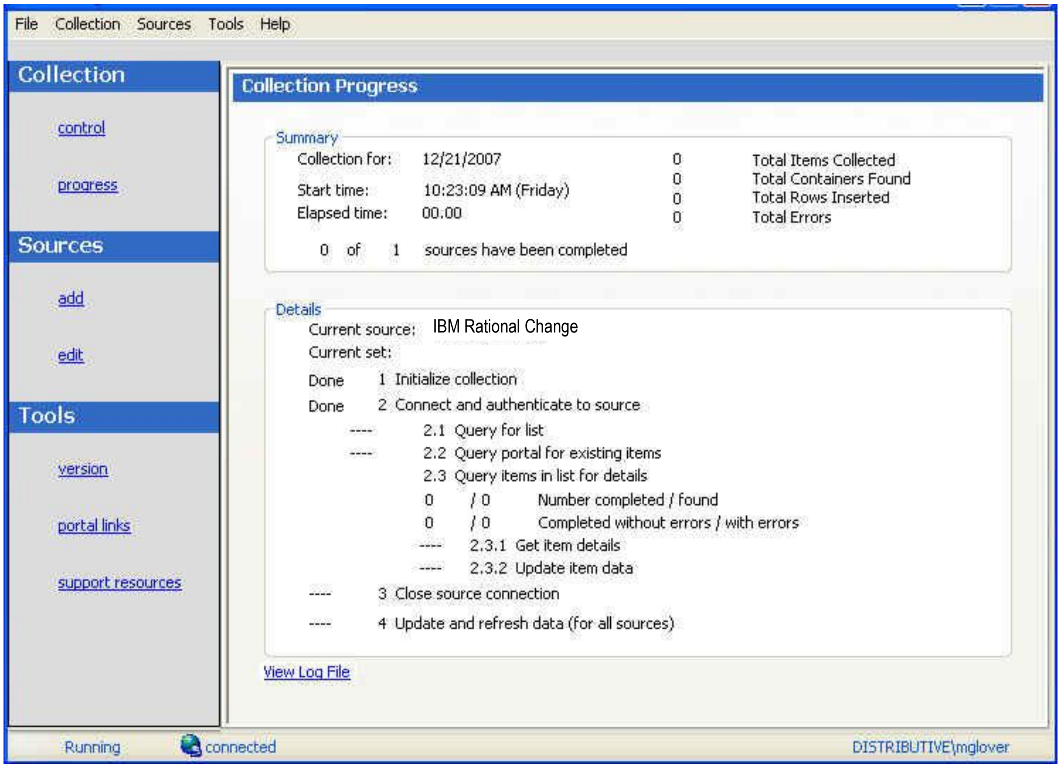
○ From Specified Date to Today: Friday , December 21, 2007

Last collection:

Figure 18

You can look at the **progress** page while the collection is running. When collection is finished, you will see the word **Done** by step #4 “Update and refresh data (for all sources).”

Once you see this, you are finished with your collection. The items will have been pulled into IBM Rational Dashboard, and now you can go into the IBM Rational Dashboard Portal to ensure that your unit has been populated with data.



Summary

Collection for: 12/21/2007
Start time: 10:23:09 AM (Friday)
Elapsed time: 00.00

0 Total Items Collected
0 Total Containers Found
0 Total Rows Inserted
0 Total Errors

0 of 1 sources have been completed

Details

Current source: IBM Rational Change
Current set:

```
Done    1  Initialize collection
Done    2  Connect and authenticate to source
  ----       2.1  Query for list
  ----       2.2  Query portal for existing items
             2.3  Query items in list for details
             0     / 0       Number completed / found
             0     / 0       Completed without errors / with errors
           ----    2.3.1  Get item details
           ----    2.3.2  Update item data
----    3  Close source connection
----    4  Update and refresh data (for all sources)
```

Figure 19

4.4 Create Unit

If you do not have a Unit created for this collection, you must do so now. For instructions on how to create a Unit please refer to the Online Help or the IBM Rational Dashboard Getting Started Guide.

4.5 Assign an Item to a Unit

To populate your unit with data, you must first assign the item to a unit.

1. Go to IBM Rational Dashboard **Portal → Status tab**
2. Highlight your Unit.
3. Click the **pencil icon** at the top of the Status Page to edit your Unit
4. Click on the **by date** or **by source** tab on the right side (by date to select an item based on the collection date; by source to select an item based on the collection source)
5. Click whether to show **all items** or to show **containers** (containers are collection of items according to the structure of the source, and if selected will assign multiple items to the unit)
6. Select the item to **assign** to the unit from the item list

7. Highlight the item to be assigned and click the **Assign** button

4.6 Run Data Collection to Collect Data

Now that the Collector has found items and at least one of them has been assigned to your unit, you can collect and populate data.

1. Go to the IBM Rational Dashboard Collector
2. Under **Collection**, click **Control** (the Control Collection page displays)
3. Set the Collection as follows:
   - **One Source**: Choose the last source you created
   - **Collect for**: <a collection data> for example: Saturday, September 30, 2006.
     Please refer to the Schedule Section in this document for more detailed on setting up proper schedules.

     Note: If functionality in the Collector is disabled, the connection to Web Services may be disconnected. Check the bottom status bar to see if it indicates connected or disconnected. If disconnect, select Connect from the File menu to connect to Web services.

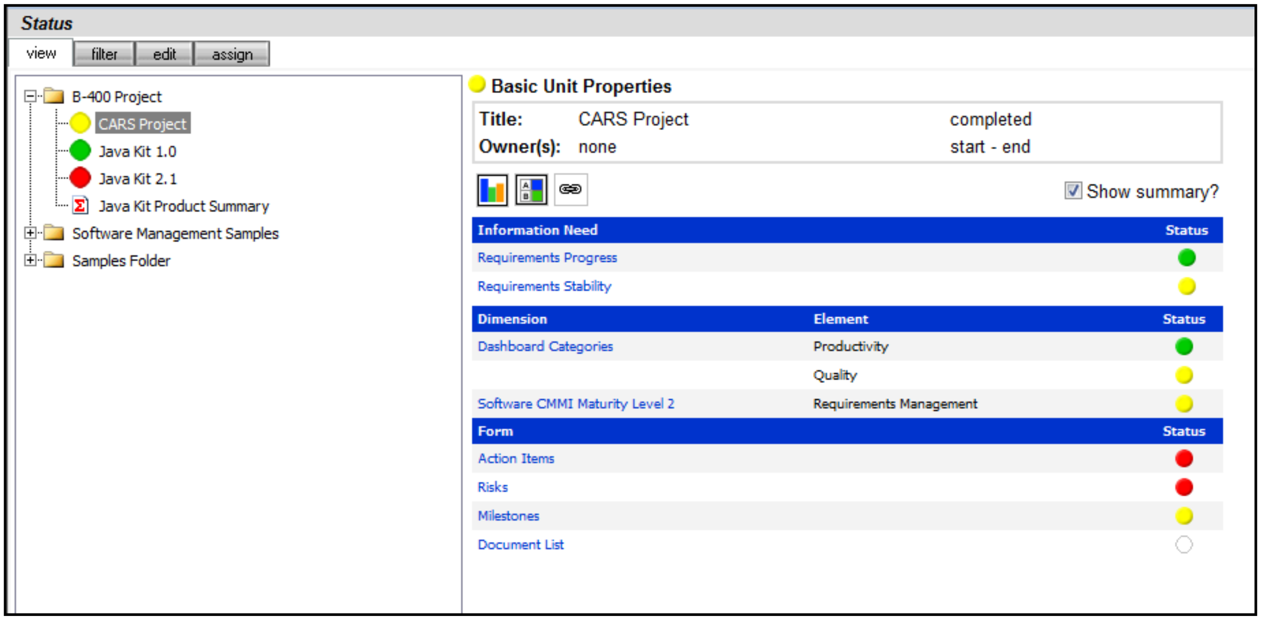
4. Click the **Start** button to begin the collection
   The Collection Progress page displays and shows the progress of collection as it is performed. Once collection is complete, the Collection Progress page indicates at the bottom that all 4 Details steps are “done” and shows collection results in the summary at the top.

5. From the **File** menu, select **Exit**.

4.7 *View the collected data in your unit*

Now view your data:

First, go to the **Status** tab and select a unit from the Organization Tree as shown below.



Basic Unit Properties

Title:	CARS Project	completed
Owner(s):	none	start - end

☑ Show summary?

Information Need	Status
Requirements Progress	●
Requirements Stability	●

Dimension	Element	Status
Dashboard Categories	Productivity	●
	Quality	●
Software CMMI Maturity Level 2	Requirements Management	●

Form	Status
Action Items	●
Risks	●
Milestones	●
Document List	○

Figure 20

To access the status and definition once you have selected something from the tree, press the **details** button and you will see the Gantt view of the selected unit.

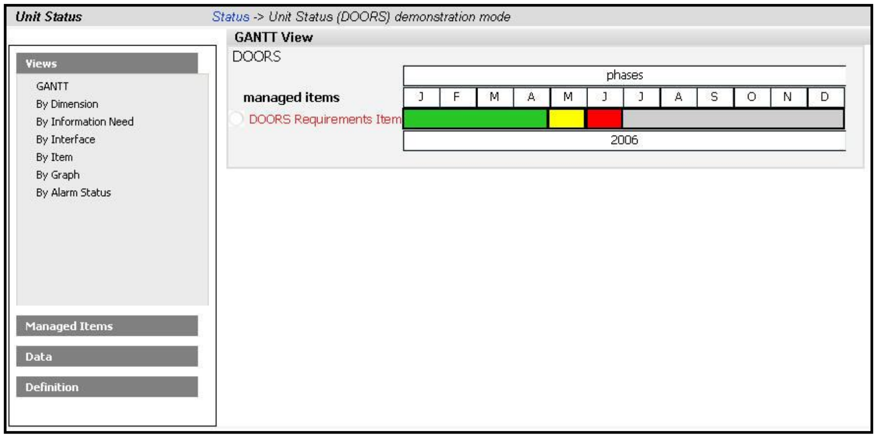


Figure 21

If the graphs are not populated, try refreshing the graphs, by going under Definition → Unit Properties. When your Unit is brought up, click the **refresh** button in the middle of the page. A new page will open up. Give a **Start date** approximately 10 days prior to today and an end date. Click **Run** and then **Cancel**. Check your graphs again.

If the graphs are still not populated, please refer to the Troubleshooting section at the end of this document.

4.8 Adding a Schedule

To add a schedule:

1. Click on the **Admin** tab in the IBM Rational Dashboard Portal
2. Click **Schedules** from the left hand column
3. Click **add** on top of the grid. The *Edit Schedule* page will display
4. In the Edit Schedule page:
   - Enter a **title** for the schedule
   - Enter a **start** date
   - Enter an **end** date. The end date cannot precede the start date.
   - Click **save**. The schedule start and end date must be saved before entering individual collect dates.
   - Click **cancel** to return to the previous page.

To add schedule collect dates:

1. Click on the **Admin** tab in the IBM Rational Dashboard Portal
2. Click **Schedules** from the left hand column
3. Click **Edit** to the left of the schedule name. The *Edit Schedule* page will display
4. Click **add Periods** button schedule item link. The schedule options will display
5. Click **radio** button for add single item or add multiple items
6. For a single item:
   - Enter a **start** date
   - Enter **end** date
   - Enter **collect** date
   - Click **add**
7. For multiple items:
   - Enter a **first** date.
   - Select a **frequency**.
     - For monthly, select the day of the month when the collection should begin. For example, if the 1st is entered, the collection will run from the first of the month through the end of the month. If the 15th is entered, the collection will run from the 15th of the month through the 14th of the next month.
     - For weekly, select the day of the week the collection should run. For example if Monday is selected, the collection will run Monday through Sunday of that week.
     - For Daily, select the number of days should past before a new collection begins. For example, if 3 is selected then a collection will run every three days.
   - Enter a **last** date.

- Enter **collection** offset. An offset will delay the collection by the number of days that is selected. For example, the frequency is set for month, with the collection starting on the 1st of the month. By adding an offset of 5 days, the data collection would begin on the 5th of the month instead of the 1st of the month but it would still collect for the entire month.
- Click **add**.

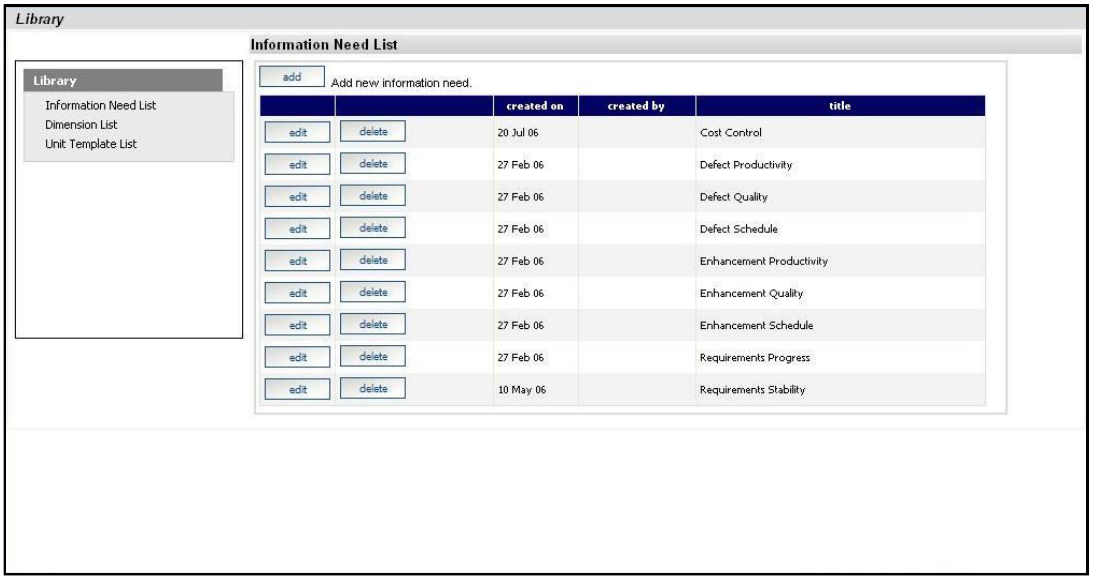
4.9 Tailoring for a New Attribute

This section describes how to add and use a field or attribute from a data source that was not part of the default installation.

4.10 Tailor the Library

This section describes how to tailor an information need in the library such that the modified information need refers to a new field from IBM Rational Change or DOORS.

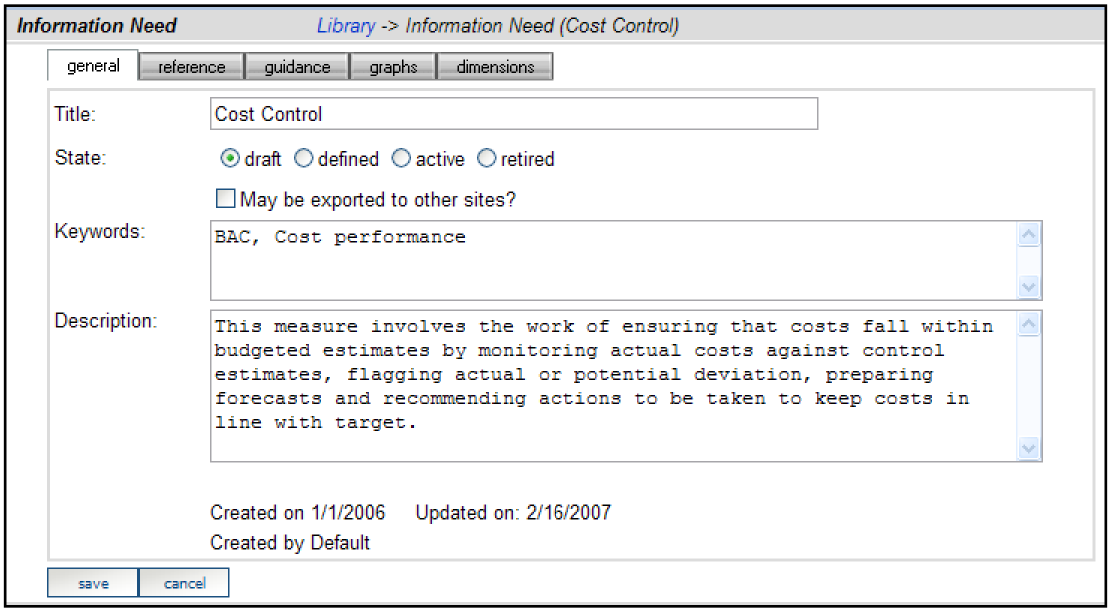
1. Click on the **Library** tab in the IBM Rational Dashboard Portal
2. Click on the **Information Need List**
3. Click on the **edit** button one of the Information Needs (e.g. Cost Control)



		created on	created by	title
edit	delete	20 Jul 06		Cost Control
edit	delete	27 Feb 06		Defect Productivity
edit	delete	27 Feb 06		Defect Quality
edit	delete	27 Feb 06		Defect Schedule
edit	delete	27 Feb 06		Enhancement Productivity
edit	delete	27 Feb 06		Enhancement Quality
edit	delete	27 Feb 06		Enhancement Schedule
edit	delete	27 Feb 06		Requirements Progress
edit	delete	10 May 06		Requirements Stability

Figure 23

The selected Information Need is shown below with the **general** sub tab information displayed.



Information Need *Library -> Information Need (Cost Control)*

Title: Cost Control

State: ● draft ○ defined ○ active ○ retired

□ May be exported to other sites?

Keywords: BAC, Cost performance

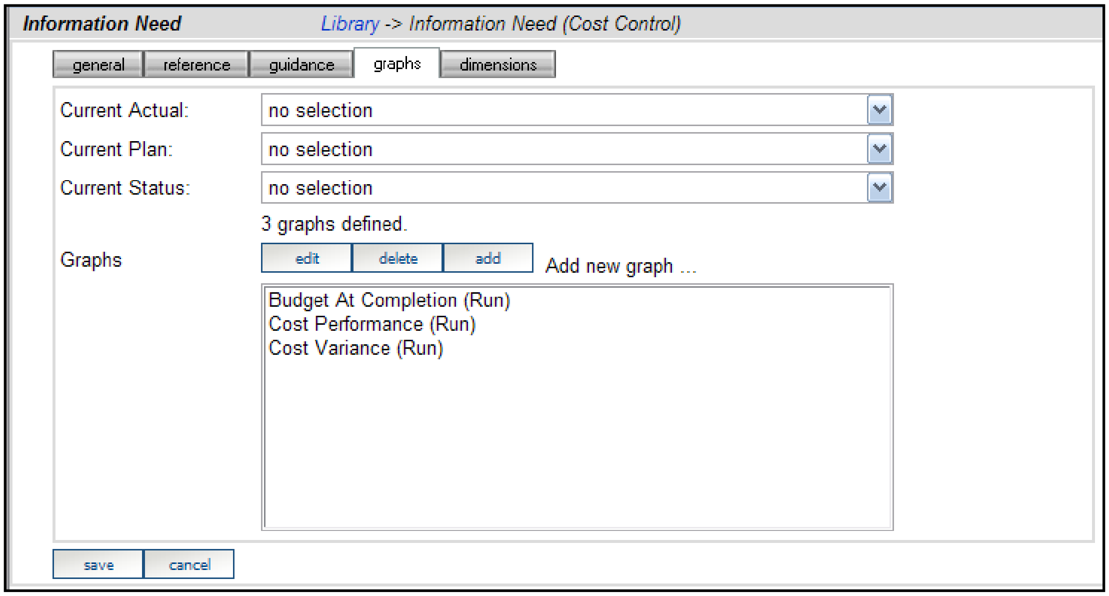
Description: This measure involves the work of ensuring that costs fall within budgeted estimates by monitoring actual costs against control estimates, flagging actual or potential deviation, preparing forecasts and recommending actions to be taken to keep costs in line with target.

Created on 1/1/2006 Updated on: 2/16/2007

Created by Default

Figure 24

4. Switch to the **graphs** sub tab to see the list of graphs currently defined for this information need, as shown below.



Information Need *Library -> Information Need (Cost Control)*

Current Actual: no selection

Current Plan: no selection

Current Status: no selection

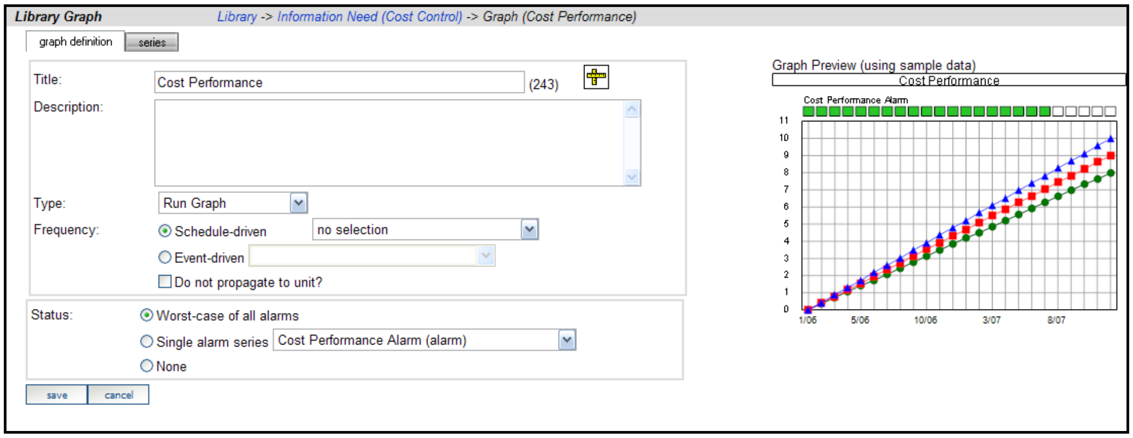
3 graphs defined.

Graphs Add new graph ...

Budget At Completion (Run)
Cost Performance (Run)
Cost Variance (Run)

Figure 25

5. Select a **graph** from the list
6. Press the **edit** button to display information about the selected graph as shown below.



Library Graph Library -> Information Need (Cost Control) -> Graph (Cost Performance)

graph definition | series

Title: Cost Performance (243)

Description:

Type: Run Graph

Frequency: ● Schedule-driven no selection

○ Event-driven

□ Do not propagate to unit?

Status: ● Worst-case of all alarms

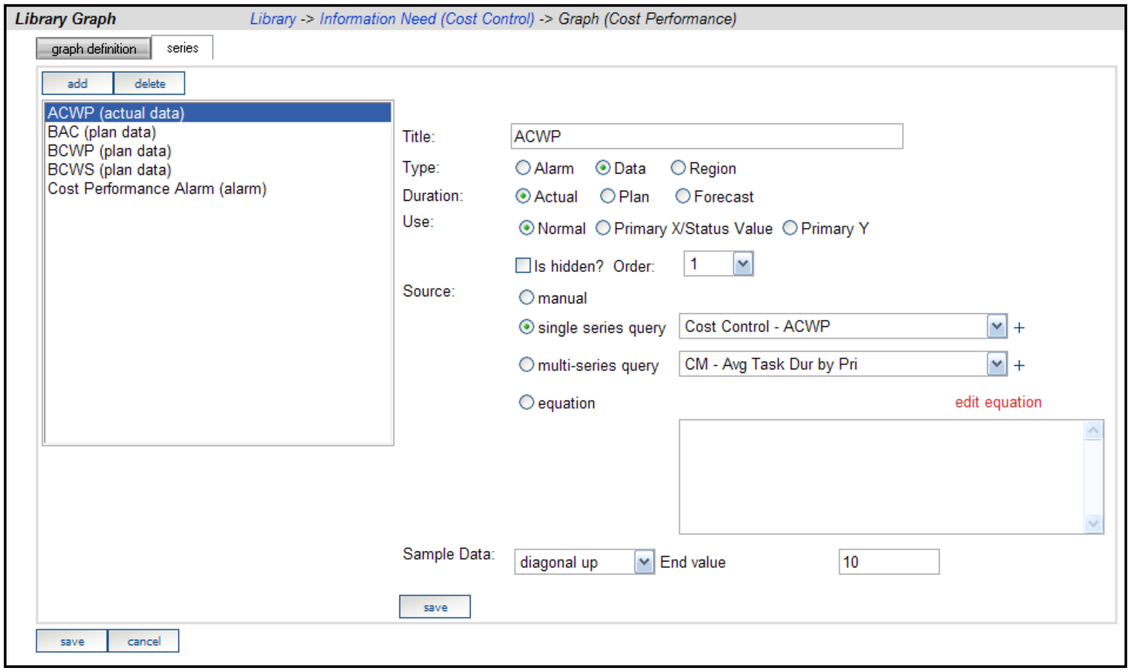
○ Single alarm series Cost Performance Alarm (alarm)

○ None

save | cancel

Figure 26

7. Switch to the **series** sub tab
8. Press the **add** button to create a new series



Library Graph Library -> Information Need (Cost Control) -> Graph (Cost Performance)

graph definition | series

add | delete

ACWP (actual data)
BAC (plan data)
BCWP (plan data)
BCWS (plan data)
Cost Performance Alarm (alarm)

Title: ACWP

Type: ○ Alarm ● Data ○ Region

Duration: ● Actual ○ Plan ○ Forecast

Use: ● Normal ○ Primary X/Status Value ○ Primary Y

□ Is hidden? Order: 1

Source: ○ manual

● single series query Cost Control - ACWP +

○ multi-series query CM - Avg Task Dur by Pri +

○ equation edit equation

Sample Data: diagonal up End value 10

save

save | cancel

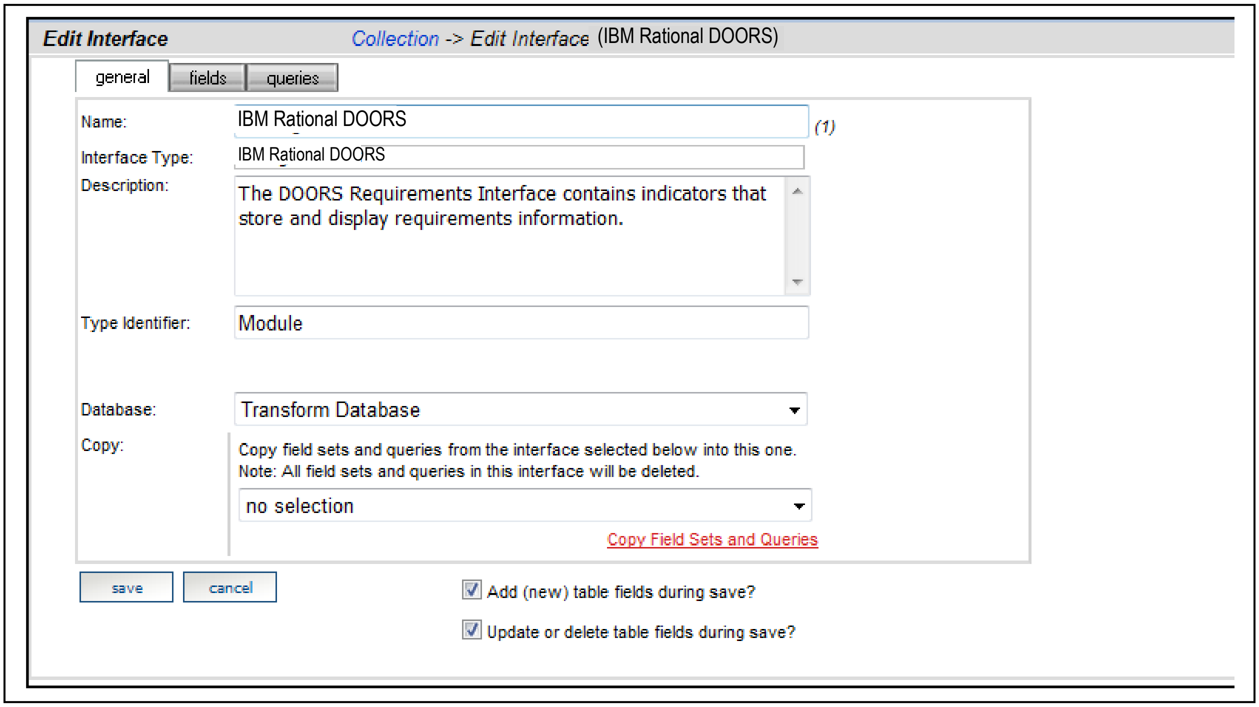
Figure 27

A new blank series is created for you, which you will give a new name. Simply set properties as needed.

4.11 Modify an Interface to Collect a New Field

Now that we have modified the Library, let's modify the Interface to collect a new field.

1. Go to the **Collect** tab
2. Select an interface from the list
3. Click on the **Edit** button to bring up information for the selected interface. The *general* sub tab is shown below.



Edit Interface Collection -> Edit Interface (IBM Rational DOORS)

Name: IBM Rational DOORS (1)

Interface Type: IBM Rational DOORS

Description: The DOORS Requirements Interface contains indicators that store and display requirements information.

Type Identifier: Module

Database: Transform Database

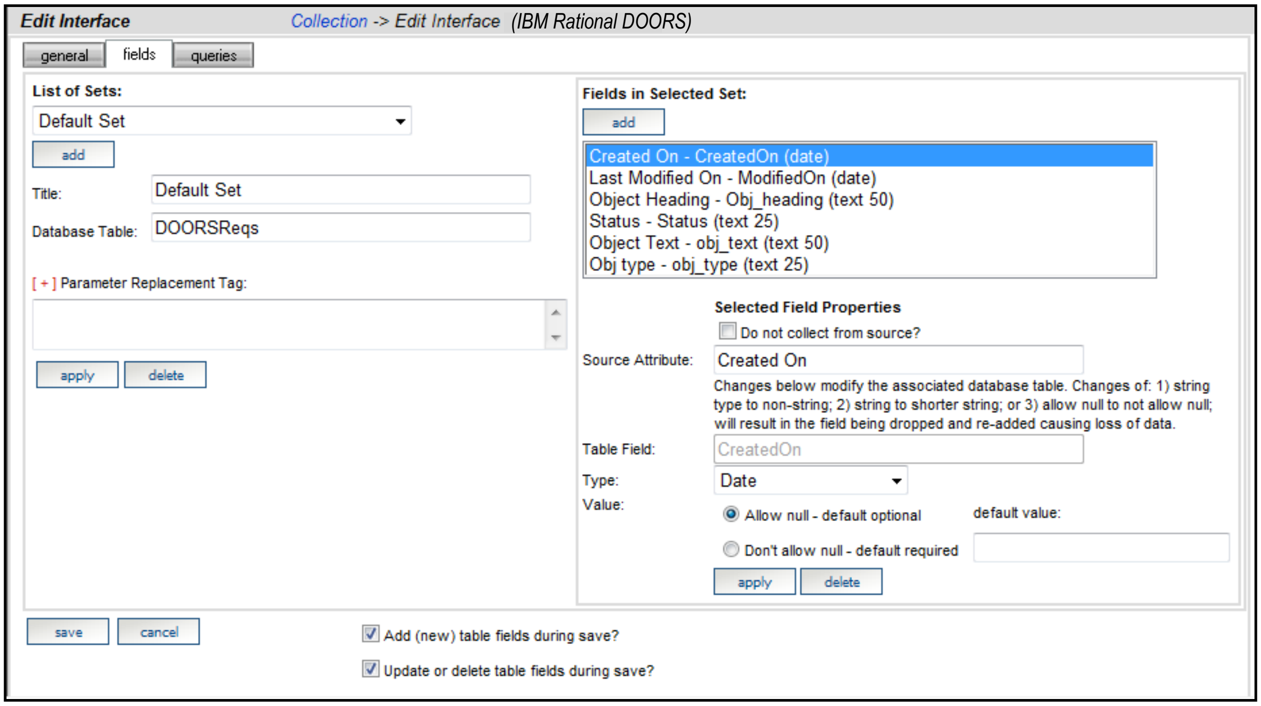
Copy: Copy field sets and queries from the interface selected below into this one.
Note: All field sets and queries in this interface will be deleted.
no selection

☑ Add (new) table fields during save?

☑ Update or delete table fields during save?

Figure 29

4. Click on the **fields** sub tab to review the list of fields. In the figure below, we are adding a field called **Created On** to the DOORS interface. Notice above, that you must specify the **name of the attribute** in the **source tool** (**Source Attribute**) as well as the name of a **Table Field** in IBM Rational Dashboard Transform Database to hold the data.



List of Sets: Default Set

Title: Default Set

Database Table: DOORSReqs

[+] Parameter Replacement Tag:

Fields in Selected Set:

- Created On - CreatedOn (date)
- Last Modified On - ModifiedOn (date)
- Object Heading - Obj_heading (text 50)
- Status - Status (text 25)
- Object Text - obj_text (text 50)
- Obj type - obj_type (text 25)

Selected Field Properties

□ Do not collect from source?

Source Attribute: Created On

Changes below modify the associated database table. Changes of: 1) string type to non-string; 2) string to shorter string; or 3) allow null to not allow null; will result in the field being dropped and re-added causing loss of data.

Table Field: CreatedOn

Type: Date

Value: ● Allow null - default optional ○ Don't allow null - default required

default value:

☑ Add (new) table fields during save?

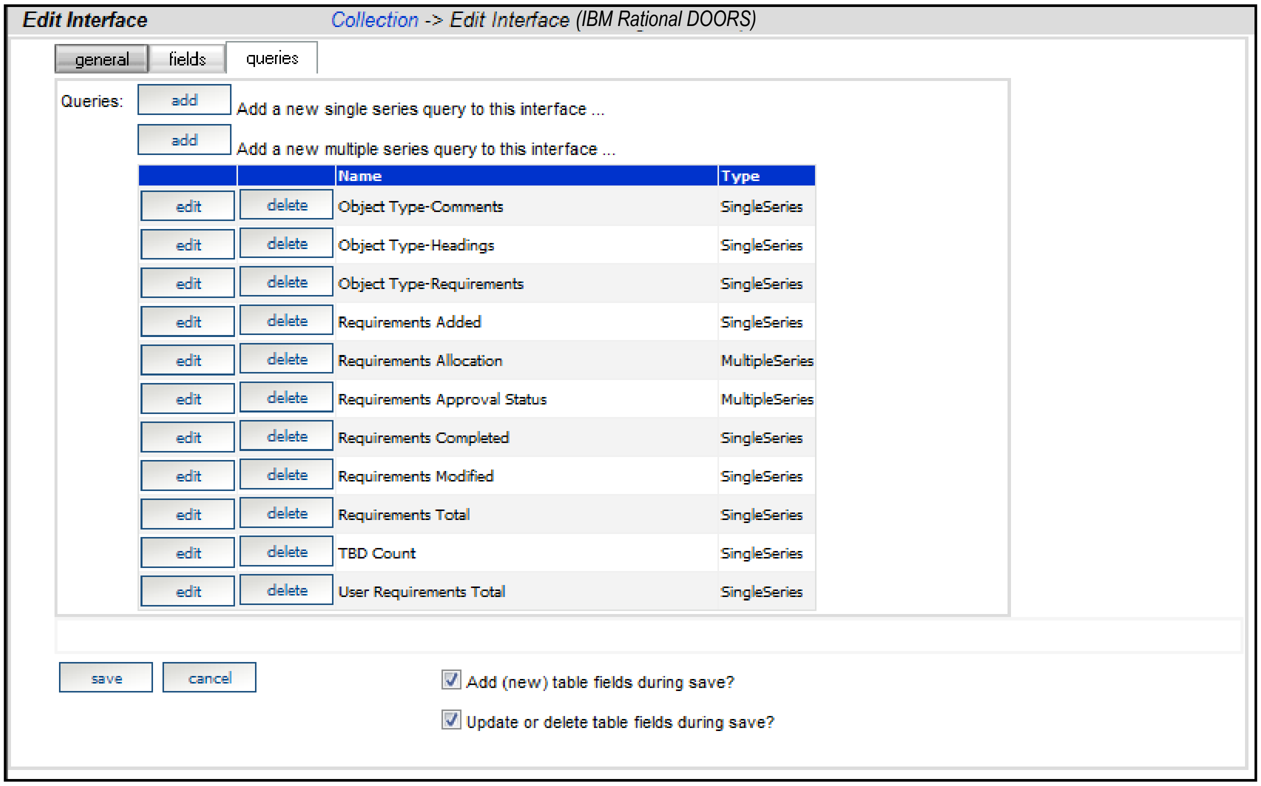
☑ Update or delete table fields during save?

Figure 30

5. Pick an appropriate data type.

   Now that we have defined a new field of data to collect, we must create a Query that will count the data from this new field. (Alternatively, you could modify an existing Query to use the new field.)

6. Switch to the **queries** sub tab to review the list of queries as shown below.



Add a new single series query to this interface ...

Add a new multiple series query to this interface ...

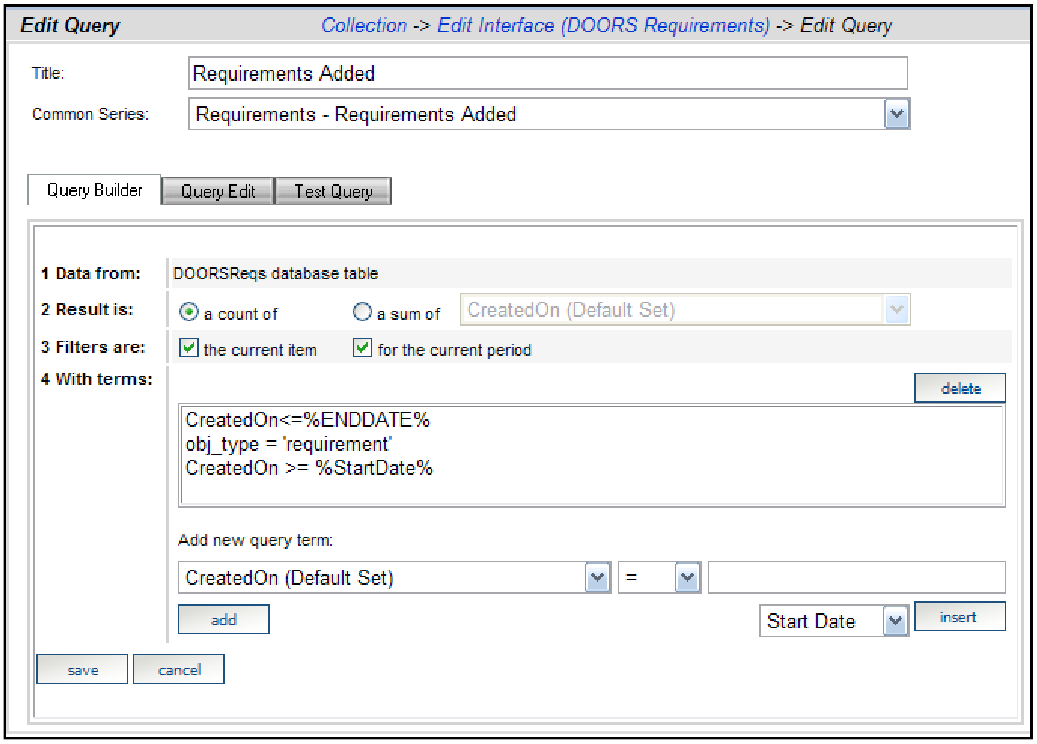
		Name	Type
edit	delete	Object Type-Comments	SingleSeries
edit	delete	Object Type-Headings	SingleSeries
edit	delete	Object Type-Requirements	SingleSeries
edit	delete	Requirements Added	SingleSeries
edit	delete	Requirements Allocation	MultipleSeries
edit	delete	Requirements Approval Status	MultipleSeries
edit	delete	Requirements Completed	SingleSeries
edit	delete	Requirements Modified	SingleSeries
edit	delete	Requirements Total	SingleSeries
edit	delete	TBD Count	SingleSeries
edit	delete	User Requirements Total	SingleSeries

☑ Add (new) table fields during save?

☑ Update or delete table fields during save?

Figure 31

7. Select a **query** that you want to modify to use the new field or create a new **single-series query**. The resulting page is shown below:



Edit Query *Collection -> Edit Interface (DOORS Requirements) -> Edit Query*

Title: Requirements Added

Common Series: Requirements - Requirements Added

1 Data from: DOORSReqs database table

2 Result is: ● a count of ○ a sum of CreatedOn (Default Set)

3 Filters are: ☑ the current item ☑ for the current period

4 With terms:

```
CreatedOn<=%ENDDATE%
obj_type = 'requirement'
CreatedOn >= %StartDate%
```

Add new query term: CreatedOn (Default Set) =

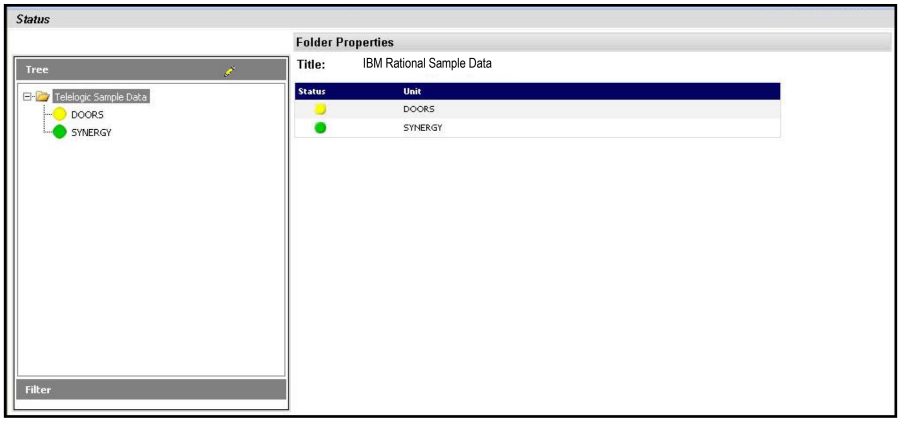
Start Date

Figure 32

8. Once you have configured the Information Need, simply press the **save** button.

4.12 Assign the Tailored Information Need

1. Go to the **Status** tab in IBM Rational Dashboard Portal
2. Select your **Unit** from the Organization Tree as shown below.



Folder Properties

Title: IBM Rational Sample Data

Status	Unit
●	DOORS
●	SYNERGY

Figure 33

3. Press the **Details** button and you will see the Gantt view of the selected unit.

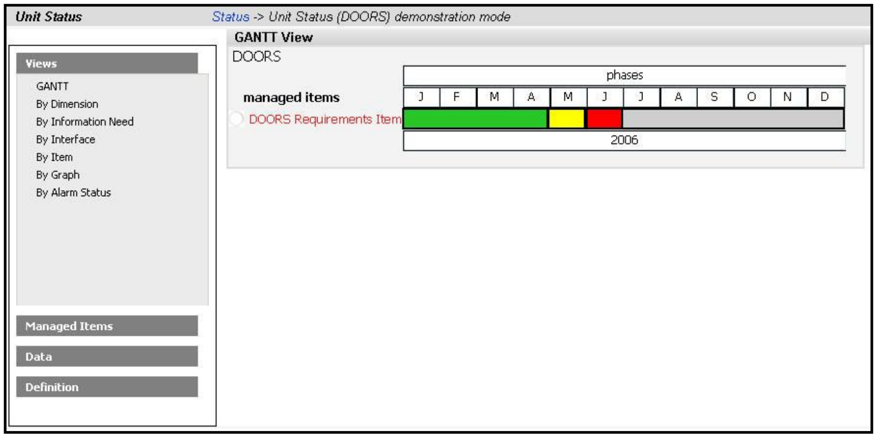
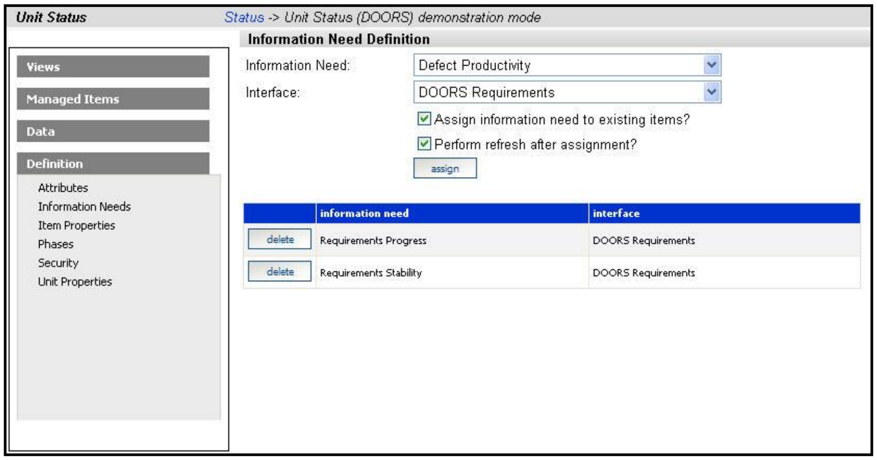


Figure 34

4. To assign the information need (modified previously), select the **Definition** group on the left hand side
5. Select the **Information Needs** command to display the following page.



Information Need Definition

Information Need: Defect Productivity

Interface: DOORS Requirements

☑ Assign information need to existing items?

☑ Perform refresh after assignment?

	information need	interface
delete	Requirements Progress	DOORS Requirements
delete	Requirements Stability	DOORS Requirements

Figure 35

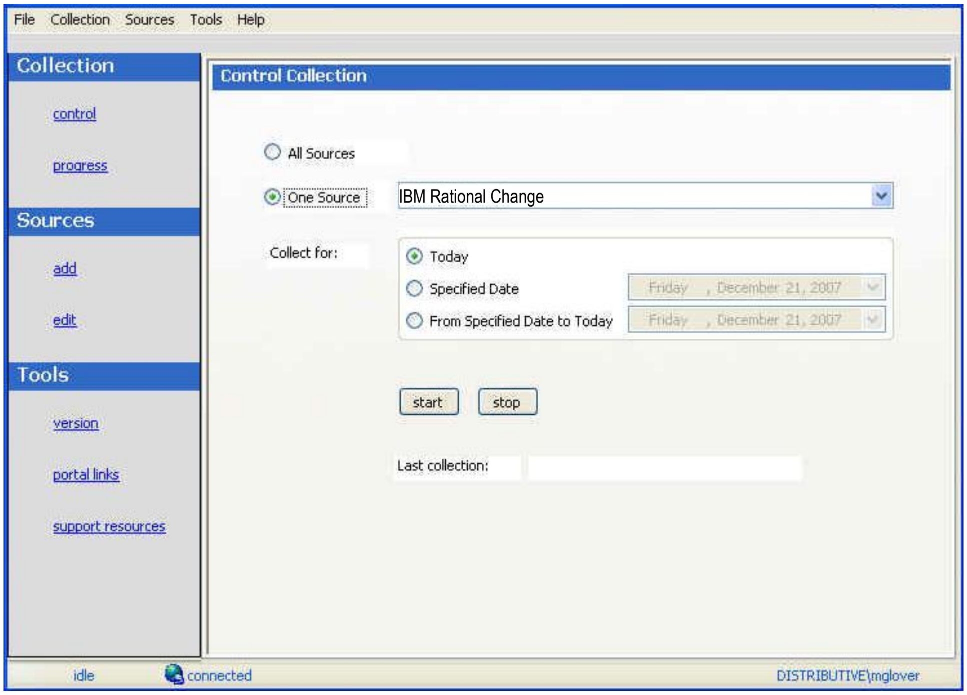
6. Above, you should remove the information need that you modified if it is already in the list.
7. Select the Information Need and the Interface, enable options for assigning to existing items and performing a refresh
8. Click the **assign** button

   Now, you have successfully configured the system to collect, process, and display the new field.

4.13 Run Collection

To run a collection again:

1. Start the IBM Rational Dashboard Collector using the command on your Start menu. It will start as shown in the following diagram.



○ All Sources

● One Source: IBM Rational Change

Collect for:

● Today

○ Specified Date: Friday , December 21, 2007

○ From Specified Date to Today: Friday , December 21, 2007

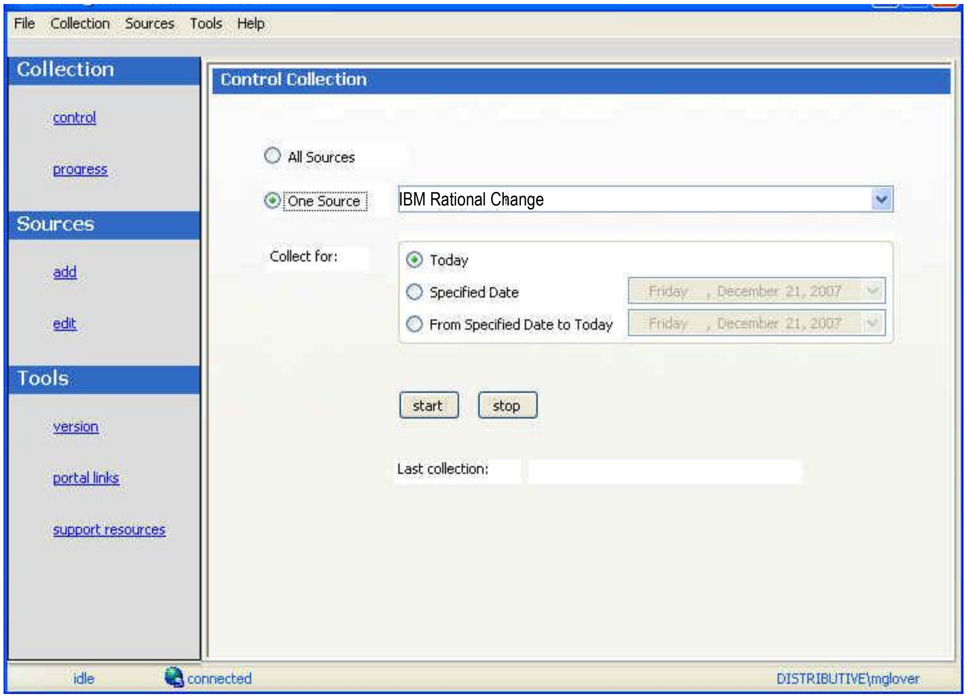
Last collection:

Figure 36

2. To review the parameters (previously entered) for a data source: Press the **edit** command in the **Sources** area on the left-hand side. The following window is displayed.

Figure 37

3. Switch to the **control** page by clicking on the command on the left hand side.



Control Collection

○ All Sources
● One Source IBM Rational Change

Collect for:
● Today
○ Specified Date Friday , December 21, 2007
○ From Specified Date to Today Friday , December 21, 2007

Last collection:

idle connected DISTRIBUTIVE\mglover

Figure 38

4. To start collection, select the **One source** options
5. Select the **desired data source**
6. Click the **start** button. Progress is displayed as collection proceeds.
7. To review new data, check that the Unit and graphs that you modified in the previous steps.

4.14 Task Wizard

For users who would rather skip most of the above steps to tailor an Information Need graph, IBM Rational Dashboard provides a wizard that can help.

The "Add Series to Graph Task" wizard provides the user with a simple way to tailor an existing Information Need graph. It is located in the IBM Rational Dashboard Portal, in the Library tab, in the Tasks link.

The wizard interfaces allows users to add a single query-based series to a graph quickly without having to do the steps manually, and it walks the user through the steps required to correctly configure the new series.

- Go into **Library** tab
- **Click** on the **Add Series to Graph Task**

The Add Series to Graph Task is divided into four steps:

1. **Choosing the Information Need and Graph to modify**.
   After selecting the Information Need and graph, click the **next** button and a new graph series is created in the selected Information Need graph.

2. **Configure the newly added series which includes: the series name, duration, use and source.**
   The series source is an SQL query. The source may be a single or multi-series query and may use an existing query or a new one.

3. **Choose the existing interface in which a new field and SQL query will be added.**
   If a new interface field is not needed for this series and query, uncheck the “Create new interface field” check box. Select the current Interface and click the next button to move to the final step.

4. **Configure the new field added to the interface and edit the new query added to the interface.**

   The name of the attribute in the source should be placed in the Source Attribute field and the name of the field in the transform database table for this interface should be placed in the Table Field field.

   Finally enter the SQL required to gather the data from the new series. Click **edit** to save field changes and go to the Interface Edit Query page and create the required SQL.

   Upon completion of the SQL query editing, the wizard completion page is displayed. The task is now completed.

4.15 Add a Graph to the Library

IBM Rational Dashboard provides a standard set of graphs based upon Information Needs. Users may have additional graphical needs that are not available out of the box with IBM Rational Dashboard. IBM Rational Dashboard provides two types of graphs: Unit graphs and Library graphs. The Unit graph applies only to that specific Unit. Changes to a Library graph will be propagated to any Unit where the Information Need is newly assigned. If there are units that currently use a graph and you wish to obtain the graph changes made in the Library, you must dissociate the Information Need containing the graph from the Unit and the re-associate it.

New graphs can be added in two ways:

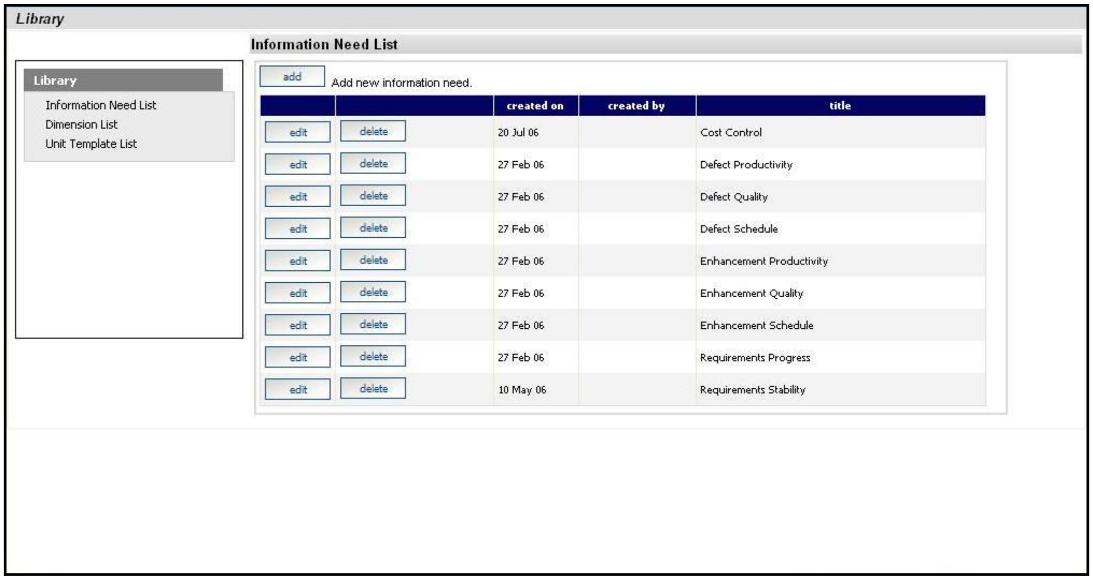
1. By adding a new graph to an existing Information Needs
2. By adding an Information Need with the new graph

For example, say a unit has the Information Need "Cost Control" assigned. When reviewing that Information Need, a manager will see that three graphs are assigned to that need: Budget at Completion, Cost Performance, and Cost Variance. But the manager would like to also see a graph regarding Scheduled Performance. To see this additional graph, a user with a correct administrative right must add it to the Cost Control Information Need.

Or they can create a new Information Need and assign the Schedule Performance Graph (and any others) to (new) that Information Need.

Adding an additional graph to a current Information Need will not affect other units that previously had the Information Need assigned to them. Units only display graphs that are included in the Information Need at the time the Information Need was assigned to them.

This section describes how to tailor an Information Need in the library. First, we use the **edit** button to open an information need from the Library tab shown on the following figure.



Information Need List

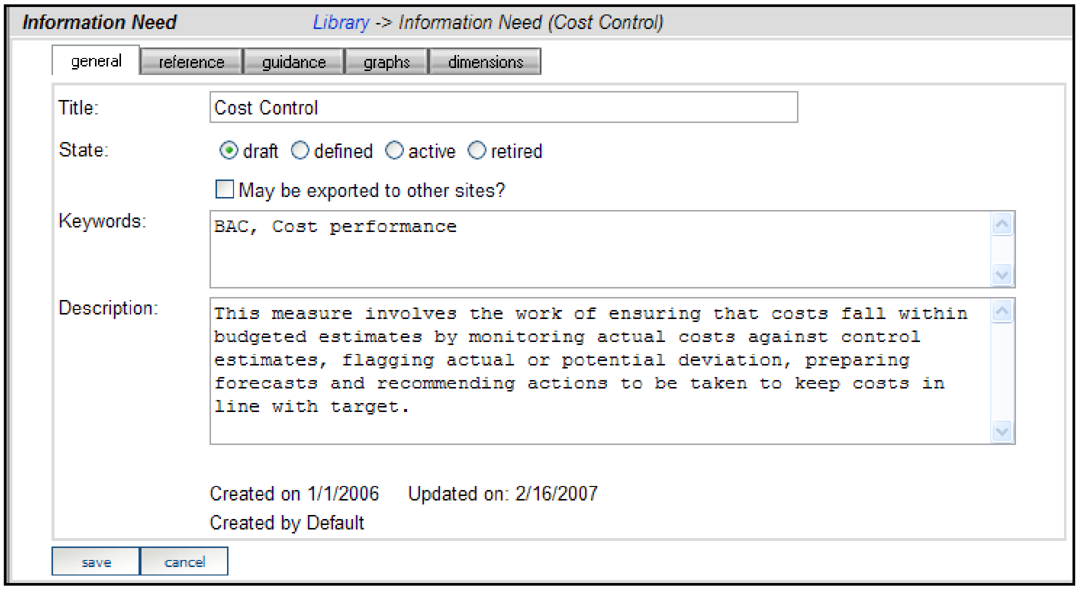
Add new information need.

		created on	created by	title
edit	delete	20 Jul 06		Cost Control
edit	delete	27 Feb 06		Defect Productivity
edit	delete	27 Feb 06		Defect Quality
edit	delete	27 Feb 06		Defect Schedule
edit	delete	27 Feb 06		Enhancement Productivity
edit	delete	27 Feb 06		Enhancement Quality
edit	delete	27 Feb 06		Enhancement Schedule
edit	delete	27 Feb 06		Requirements Progress
edit	delete	10 May 06		Requirements Stability

Figure 39

1. Click on the **Library** tab
2. Click on **Information Need List** in the left hand column
3. Click on the **edit** button to the left of the Information Need you want to add a graph to (e.g. Cost Control). The Information Need page will load.

   The selected Information Need is shown below with the **general** sub tab displayed.



Title: Cost Control

State: ● draft ○ defined ○ active ○ retired

□ May be exported to other sites?

Keywords: BAC, Cost performance

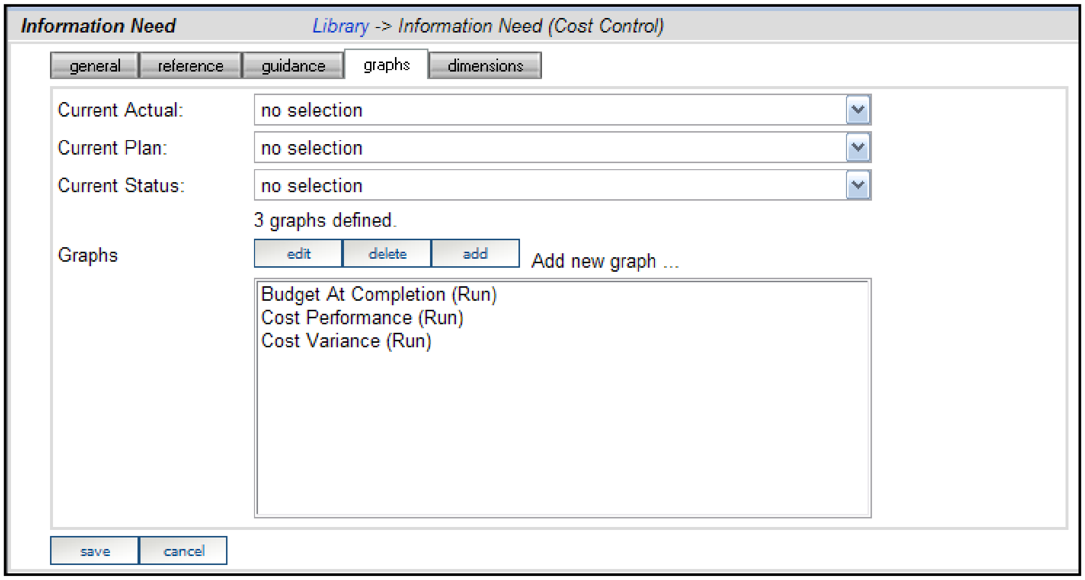
Description: This measure involves the work of ensuring that costs fall within budgeted estimates by monitoring actual costs against control estimates, flagging actual or potential deviation, preparing forecasts and recommending actions to be taken to keep costs in line with target.

Created on 1/1/2006 Updated on: 2/16/2007

Created by Default

Figure 40

4. Switch to the **graphs tab** to see the list of graphs currently defined for this information need, as shown below.



Current Actual: no selection

Current Plan: no selection

Current Status: no selection

3 graphs defined.

Graphs Add new graph ...

Budget At Completion (Run)
Cost Performance (Run)
Cost Variance (Run)

Figure 41

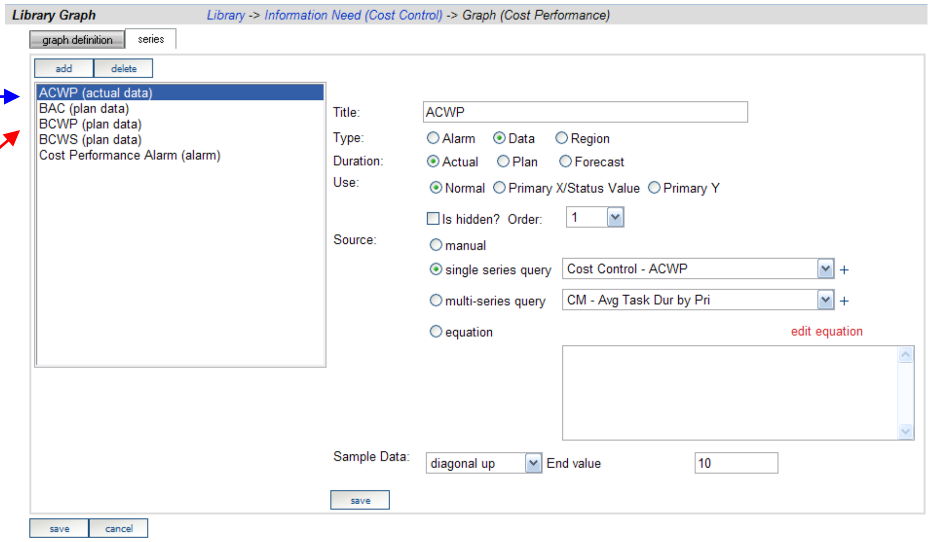
5. To the right of graphs, click on the **add** button. The **Library Graph** page will load.
6. Enter the following information for the graph:
   - **Title**
   - Enter a **Description** of the graph (*optional*)
   - **Type** – select a graph type from the drop down list
     Note: The Alarm and Data types are the most common. Alarm creates colors for status of the alarm and Data provides numbers into the graph. Region has two

values (edit upper and edit lower) which define the thresholds for colors of regions to change (e.g. edit upper: 100 and edit lower: 90). Multiple regions can also be set up.

- **Frequency** – select a frequency type such as: schedule driven or event driven (if schedule driven is selected, you must select a schedule from the drop down list).
- **Status** – select an alarm status

Now you have to add a new series to this graph.

1. Click on the **series** tab



Library Graph — Library -> Information Need (Cost Control) -> Graph (Cost Performance)

graph definition | series

add | delete

ACWP (actual data)
BAC (plan data)
BCWP (plan data)
BCWS (plan data)
Cost Performance Alarm (alarm)

Title: ACWP

Type: ○ Alarm ● Data ○ Region

Duration: ● Actual ○ Plan ○ Forecast

Use: ● Normal ○ Primary X/Status Value ○ Primary Y

□ Is hidden? Order: 1

Source: ○ manual

● single series query: Cost Control - ACWP +

○ multi-series query: CM - Avg Task Dur by Pri +

○ equation — edit equation

Sample Data: diagonal up — End value: 10

save

save | cancel

The *Actual* vs. the *Plan*

Figure 42

2. Click on the **add** button directly under the series tab
3. Enter the following information for the series:
   - Enter a **Title** for the series
   - Select the **series order**
   - **Type** – select the type of series
     - Alarm
     - Data
     - Region

   Note: Alarm and Region are always equations. With Data Series you can select the type you want.
   - **Duration** – select the series duration

    - ♦ Actual
    - ♦ Plan
    - ♦ Region
- **Options** (determines display in graph)
- **Source** – determine where the series information originates
    - ♦ Manual
    - ♦ Single Series Query
    - ♦ Multi Series Query
    - ♦ Equation

When you click the **add** button to create a new series, a new blank series is created for you, which you will give a new name. Simply set properties as needed.

Note: For Multiple Series Queries, there is a choice of querying the database for the values or entering specific values to run the query from. When querying the database use the syntax SELECT DISTINCT <field> FROM <table>, where <field> is the name of the Table Field and <table> is the name of the table to be queried from.

4.16 Add an Information Need

You can also add an information need if you have a group of measures that do not fit into the existing information needs.

To add *a new* Information Need:

1. Click on the **Library** Tab
2. Click the **add** button on the **Information Need List**
3. Click **cancel** to return to the previous page
4. In the **Information Need page**:
    - Go to **general** tab
    - Enter **title** name
    - Select a radio button to determine state
    - Select **Options**. Check box if information need can be exported to other sites. Check box if information need is restricted to one use per unit/project.
    - Enter **Keywords**
    - Enter **Description**
    - Click **save** to add new information need
    - Click **cancel** to return to the previous page.
    - Click on Edit button beside newly created Information need
    - Go to **Reference tab** - in the Reference tab, administrators may enter unit templates, standards, URLs or SME for the information need. A Unit Template provides a predefined list of information needs and dimensions. Industry standards can also be assigned, for example, CMMI or SEI standards. URLs can

be added that will link to information needs. SME (subject matter expert) can also be referenced in this tab.

- Click **save** button to save the changes in the information need.
- Click **cancel** to return to the previous page.
- Go to **Guidance** tab - in the guidance tab, administrators may enter planning, monitoring and control parameters. Planning provides an area where administrators may enter guidelines about what should be expected from the information need, what the user should plan for and what the user should see. Monitoring provides guidelines of what the user should look for in the information need. Control provides guidelines of how the user should control the information need.
- Enter **Planning**.
- Enter **Monitoring**.
- Enter **Control**.
- Click **save button** to save the changes in the information need.
- Click **cancel** to return to the previous page.

To add a graph to this information need please see previous section. After you have added a graph, return here, and assign a dimension to the Information Need.

4.17 Add a Dimension

Dimensions represent categories of status of interest to your organization. To add a new dimension to the Library, follow these steps:

1. In the IBM Rational Dashboard Portal, click on the **Library tab**
2. Under Library, click on the **Dimension List**
3. Click the **Add** button to add a new Dimension
4. Setup the new Dimension based on your process
5. Click the **Save** button to save the new Dimension.

4.18 Assign a Dimension to an Information Need

In the **dimensions** tab, administrators assign dimensions to Information Needs. Dimensions contain elements that act as categories to display the status of the information need for each Unit in the Status tab.

1. To assign a dimension to the information need, press the **assign** button (if the dimension doesn’t exist, please see the next section).
2. Select a **dimension** for the dimension list.
3. Select an **element** from the element list.
4. Press **apply** to save changes.
5. Press **cancel** to return to the previous page.

5 Advanced Topics

This section presents a variety of advanced topics. Information on specific web pages, commands and functions is contained in the Portal online help.

5.1 Sample Queries

This subsection contains sample SQL queries that demonstrate various SQL techniques for counting data in a transform database.

The following query counts the number of records in a table:

```
SELECT count(*) FROM ReqData
```

The following query counts everything in a table where the table name is determined just before the query is run with the %TABLE% tag:

```
SELECT count(*) FROM %TABLE%
```

The following query counts all objects for a single period:

```
SELECT count(*) from %TABLE% where (g_collectdate = %COLDATE%)
```

The following query counts all objects for a single period in a single module:

```
SELECT count(*) from %TABLE% where (g_collectdate = %COLDATE%) and
(g_itemid = %ITEMID%)
```

The following query counts all objects that were created during a single period

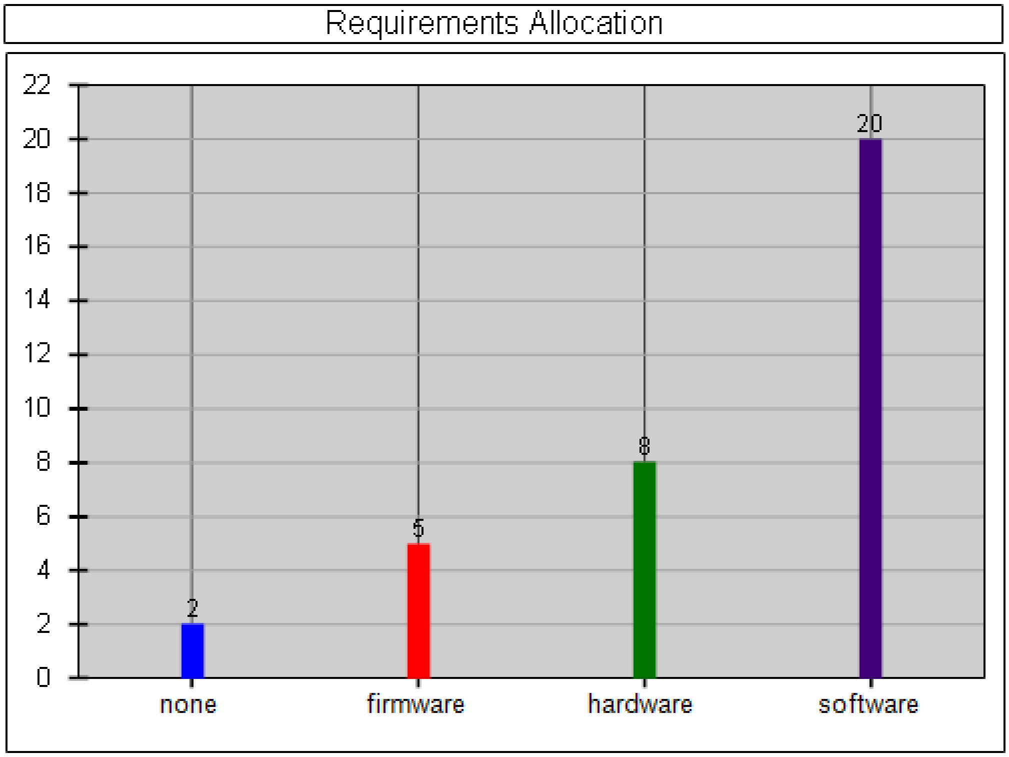
```
SELECT count(*) from %TABLE% where (g_collectdate = %COLDATE%) and
(created_on >= %STARTDATE%) and (created_on <= %ENDDATE%) and
(g_itemid = %ITEMID%)
```

The following query counts requirements that were edited in a single period

```
SELECT count(*) FROM %TABLE% WHERE (g_collectdate=%COLDATE%)
and (ObjType='R') and (modified_date IS NOT NULL) and
(modified_date >= %STARTDATE%) and (modified_date <= %ENDDATE%)
and (g_itemid = %ITEMID%)
```

5.2 Using Multi-Series Queries

A multi-series query is used to simplify setting up a graph that has more than one series which uses similar queries. For example, consider a graph of requirements by allocation type where there are four types of allocation (“none”, “software”, “hardware” and “firmware”). The graph would have four series, one series for each of the allocation type, and look like the following:



The query used for each series would be very similar (almost identical), except that each query would contain a slightly different WHERE clause such as “WHERE Allocation=’none’”, “WHERE Allocation=’software’” and so on. In this example, you can see that it would be time consuming to set up four series and then four single-series queries for the allocation types. The multi-series query simplifies this scenario.

In IBM Rational Dashboard, queries are SQL database statements that create numeric values from data stored in a transform database. Queries are run at the end of the data collection process (after data has been inserted and updated in the database) and also when a refresh is performed through the user interface. When the query is executed, it returns a numeric value which is stored in the Portal database and (usually) plotted as a series value on a graph.

There are two types of queries: single-series query and multi-series query. The most common, the single-series query, returns one value (for a *single series*) when it is run. To use a single-series query, you create a data series and assign a single-series query as the source of data. The single-series query returns one numeric value per period as the query is run for each period. For example, the "number of new defects" is a common single-series query which calculates the count of new defects created during each period.

5.2.1 How the Multi-Series Query Works

The multi-series query performs the counting operation of the single-series query but also creates other series in the graph where it is defined. In the above example, multi-series would create 4 series, one for each allocation type. A multi-series query is assigned to a series in the graph, and the "real" allocation type series are created automatically when the multi-series is run. The series created by the multi-series are called "reference series". It is recommended that you only have one multi-series query per graph.

Compared to the single series query, the multi-series requires a little more configuration, as there must be rules for determining how to create the reference series. The edit page for the multi-series query supports two options for determining how to create the reference series:

- Query database for series
- Specify series in a list

<u>Query Database for Series</u> - When the "query database for series" is used, the multi-series will create as many series as there are different (in SQL query language "distinct") values in the database field where the data is stored. In this option, you must specify a secondary SQL statement that determines the distinct values. Using the example of requirements by allocation type, the multi-query series would query the database for the allocation types that are currently in the database. For each distinct allocation type found, a new series is created in the graph. For example, consider the following collected data:

Allocation	Requirement Title
Software	Capture system events in memory log
Software	Periodically check available memory
Hardware	Provide warm restart button

With data shown above, notice that the multi-series query will find and create two distinct series, for "software" and "hardware". Even though we know there are four possible allocation types, the multi-series query found 2, since the data only contains those 2. As additional data is collected (and requirement records are added with additional allocation types), the multi-series query will create additional series for them.

When the list of values dynamically grows and you want to see them all in a graph, this option, "query database for series" is the best choice.

<u>Specify Series in List</u> – This option allows you to specify the list of reference series for the multi-series query. Each time the multi-series query is run, the reference series in the list will be

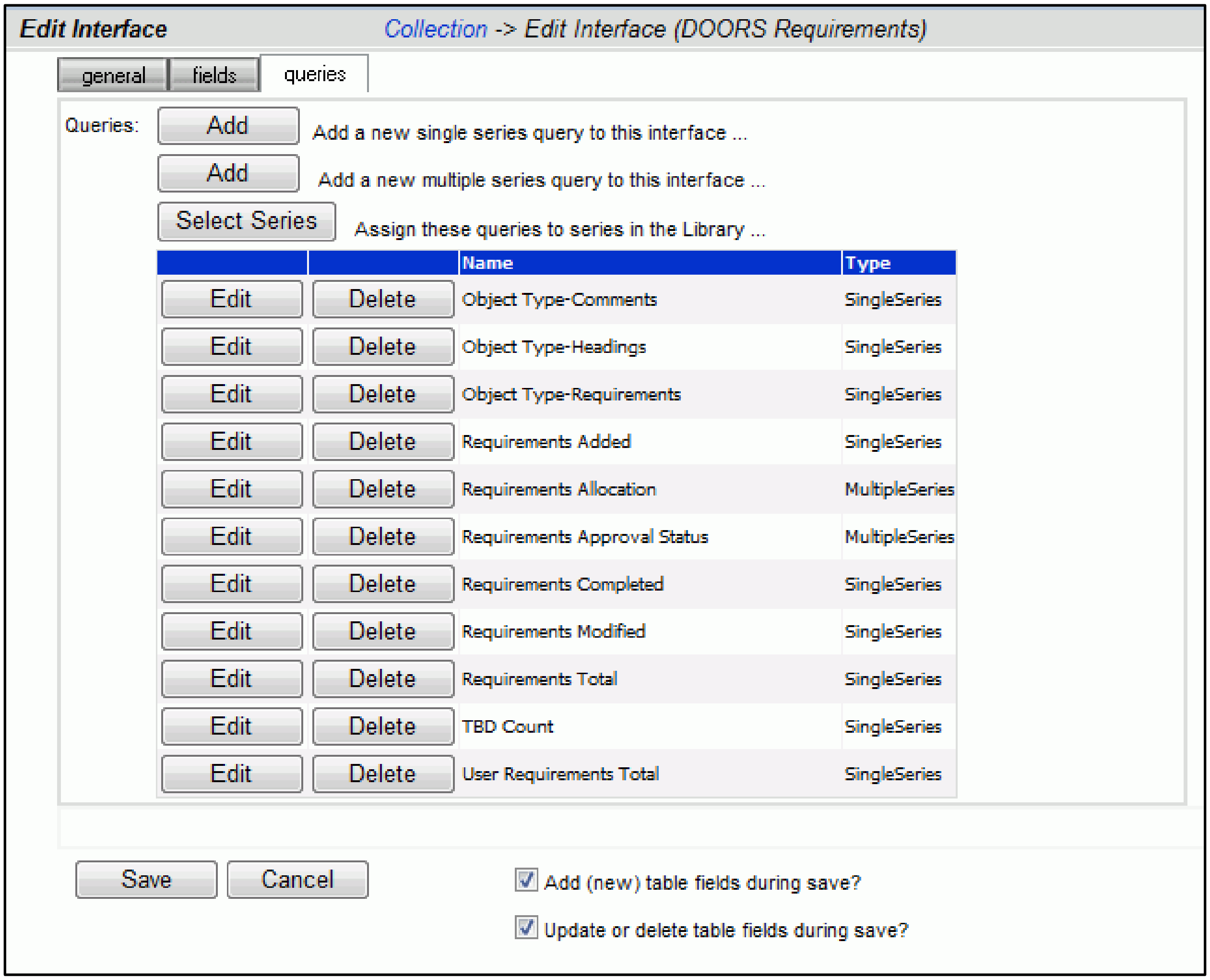
created in the corresponding graph. If you change the list of reference series, the graphs will be updated next time data collection is performed or the graph is refreshed.
In this option, you enter a list of the reference series. Each reference series consists of a series name and a value. Each entry in the list becomes a series in the graph that contains the multi-series query.

Since the list specifies the reference series to expect, there is the chance that the data in the associated table contains more than just the values you specify. For example, if we enter the four allocation types in our example, the table could contain a data value for “operator” (which was not in our list of four in this example). To detect data such as this, the multi-series query supports something called the “other” series. When this option is enabled (using a checkbox titled “Include ‘other’ series?”), the multi-series query will create a series titled ‘other’ and include a count of data items that were not counted using the reference series. When this option is used, you must enter a SQL statement that selects and counts the records to be considered ‘other’.

When the list of values does not change often, this option, “specify series in list” is the best choice.

Example Multi-Series Query Using Database Values

A multi-series query is created as a query in *a*n interface – an interface such as the 'IBM Rational Doors' interface typically contains both single-series and multi-series queries as shown below.

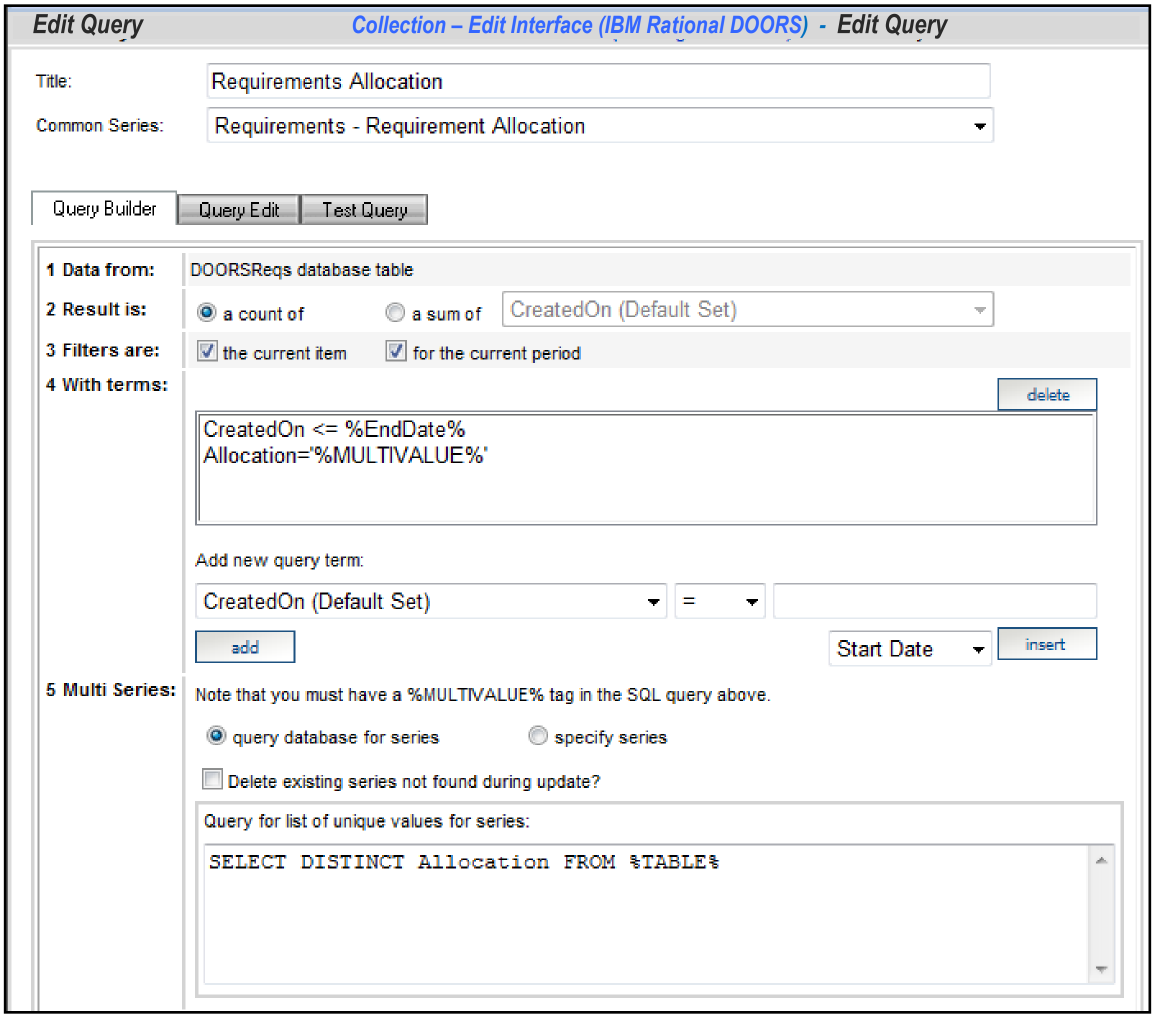

Edit Interface
Collection -> Edit Interface (DOORS Requirements)
general
fields
queries
Queries:
Add
Add a new single series query to this interface ...
Add
Add a new multiple series query to this interface ...
Select Series
Assign these queries to series in the Library ...
Save
Cancel

		Name	Type
Edit	Delete	Object Type-Comments	SingleSeries
Edit	Delete	Object Type-Headings	SingleSeries
Edit	Delete	Object Type-Requirements	SingleSeries
Edit	Delete	Requirements Added	SingleSeries
Edit	Delete	Requirements Allocation	MultipleSeries
Edit	Delete	Requirements Approval Status	MultipleSeries
Edit	Delete	Requirements Completed	SingleSeries
Edit	Delete	Requirements Modified	SingleSeries
Edit	Delete	Requirements Total	SingleSeries
Edit	Delete	TBD Count	SingleSeries
Edit	Delete	User Requirements Total	SingleSeries

☑ Add (new) table fields during save?

☑ Update or delete table fields during save?

After you press the "Add" button for adding a multi-series query, you will see the "Edit Query" page, as shown below for the "Requirements Allocation" query which is a multi-series query in the "IBM Rational Doors" interface.


Edit Query
Collection – Edit Interface (IBM Rational DOORS) - Edit Query
Query Builder
Query Edit
Test Query
delete
add
insert

Title: Requirements Allocation

Common Series: Requirements - Requirement Allocation

1 Data from: DOORSReqs database table

2 Result is: ● a count of ○ a sum of CreatedOn (Default Set)

3 Filters are: ☑ the current item ☑ for the current period

4 With terms:

```
CreatedOn <= %EndDate%
Allocation='%MULTIVALUE%'
```

Add new query term: CreatedOn (Default Set) =

Start Date

5 Multi Series: Note that you must have a %MULTIVALUE% tag in the SQL query above.

● query database for series ○ specify series

□ Delete existing series not found during update?

Query for list of unique values for series:

```
SELECT DISTINCT Allocation FROM %TABLE%
```

The first four sections (1 -4 on the left side) of the Edit Query page are the same as the single-series query, and provide a wizard-driven method for specifying the query SQL statement. The fifth section is unique to the multi-series query and allows you to specify how the reference series are created. You use the Query Builder sub-tab to build the SQL statement, or switch to the Query Edit sub-tab and enter it manually.

Note that there is one replacement tag, %MULTIVALUE% which is not used for a single series query. Since the purpose of the multi-series is to simplify creating and counting multiple series when they differ by only one term – it is the %MULTIVALUE% replacement tag that performs this function. When the SQL query for a multi-series query is run, it will be run multiple times, one for each reference series with the value of the %MULTIVALUE% tag replaced by the corresponding value from the reference series.

To understand how this works, let's walk through using the requirements allocation example. For this example, assume the list of reference series is obtained by querying the database. The SQL statement to do this is contained in the edit box labeled "Query for list of unique values for series” and is run before the multi-series query to obtain the list of reference series. From the image above, the SQL statement to get a list of allocation status values is as follows:

```
SELECT DISTINCT Allocation from %TABLE%
```

In this SQL statement, the “DISTINCT” keyword will return a list of unique values for the data in the Allocation field. This list becomes the reference series for our multi-series query. In this example, assume that the query returns the following list of string values for the allocation field:

“None”, “Software”, “Hardware”, “Firmware”

Once the list of reference series is obtained, the multi-series query in the wizard will be run. For this example, the “raw” SQL for the multi-series query uses the following:

```
SELECT COUNT(*) FROM %TABLE% WHERE
  Allocation=’%MULTIVALUE%’ AND
  CreatedOn<=%ENDDATE% AND
  g_item=%ITEMID%
```

So, using the results of the first query (which returns the list of reference series), IBM Rational Dashboard will execute the multi-series query four times, one for each of the four reference values. The resulting queries looking like the following:

```
SELECT COUNT(*) FROM DoorsReq WHERE Allocation=’none’ AND CreatedOn<=’20080601’ AND g_item=123
SELECT COUNT(*) FROM DoorsReq WHERE Allocation=’software’ AND CreatedOn <=’20080601’ AND g_item=123
SELECT COUNT(*) FROM DoorsReq WHERE Allocation=’hardware’ AND CreatedOn <=’20080601’ AND g_item=123
SELECT COUNT(*) FROM DoorsReq WHERE Allocation=’firmware’ AND CreatedOn <=’20080601’ AND g_item=123
```

(Other tag values, such as %ITEMID%, have sample data.)

The resulting graph will have four series automatically added, for a total of five with the multi-series query (which is also technically a series). Each of the four reference series will have a value set for the corresponding collection period. The graph may contain other series, such as equations and alarms that analyze the data in the multi-series query.

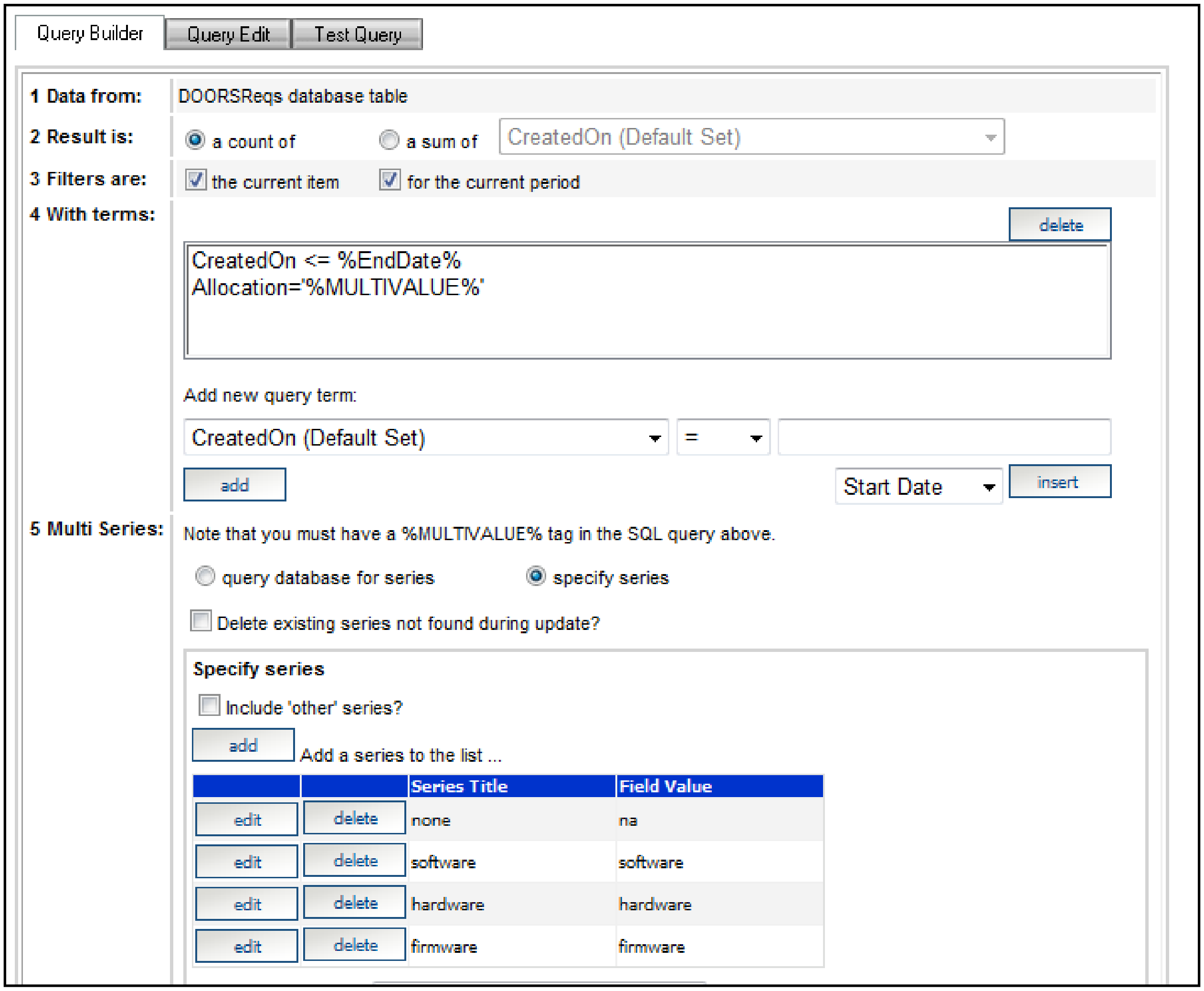
For example, the requirement by allocation type graph may contain a data series with a source of equation that uses the equation graphsum (“\”), which provides a sum of all actual data series in the graph. As reference series are added and removed, the graphsum equation adds up all available series, and does not require any maintenance.

5.2.2 Example Multi-Series Query Using Specified Series

When using the option for “specify series”, the multi-series query provides a few additional options than when using the option for “query series from database”. In particular, this option allows you to:

- Provide a series title rather than using the value which comes from the database
- Limit the series which are created to those that you want or expect
- Specify a single value to use when substituting the %MULTIVALUE% tag
- Automatically generate an ‘other’ series which counts the number of records which are not in the list

As shown below, the example from above has been modified to use the “specify series” option rather than the “query database for series” option.



1 Data from: DOORSReqs database table

2 Result is: ● a count of ○ a sum of CreatedOn (Default Set)

3 Filters are: ☑ the current item ☑ for the current period

4 With terms:

```
CreatedOn <= %EndDate%
Allocation='%MULTIVALUE%'
```

Add new query term: CreatedOn (Default Set) =

Start Date

5 Multi Series: Note that you must have a %MULTIVALUE% tag in the SQL query above.

○ query database for series ● specify series

□ Delete existing series not found during update?

Specify series

□ Include 'other' series?

Add a series to the list ...

		Series Title	Field Value
edit	delete	none	na
edit	delete	software	software
edit	delete	hardware	hardware
edit	delete	firmware	firmware

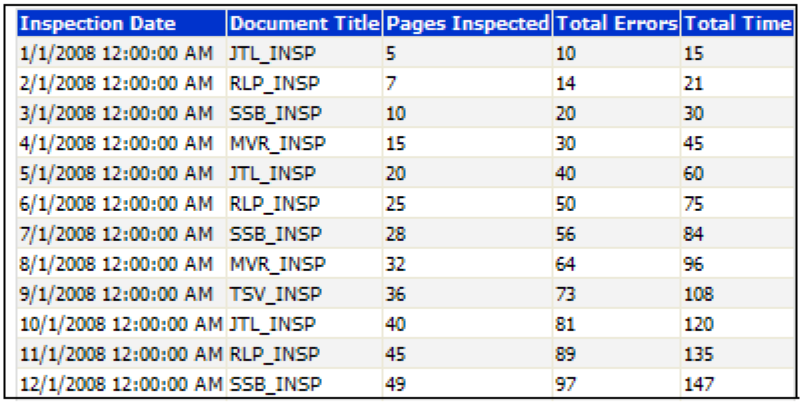
Notice that at the bottom of the screen, there are four reference series specified for “none”, “software”, “hardware” and “firmware”. Along with each reference series, there is a value which will be used for the %MULTIVALUE% tag. The values in the “field value” column are string values but do not contain the enclosing single quote mark. (In the SQL language, when you check whether a field contains a string value, you would enter fieldname=’value’, with the value surrounded by single quote marks.) This is because the term shown at item 4 (in the image next to “4. With Terms”) is defined as Allocation = ‘%MULTIVALUE%’ which provides the enclosing quote marks.

Notice that the first reference series in the list contains a title of “none” but uses a different value “na”. This allows you to substitute a value to search for different than the series title. You could also substitute a more readable or shorter title for the series if needed.

The operation of the multi-series query is much the same as the previous example, except instead of running the first query to obtain a list of distinct values, the values are simply extracted from the reference series list.

5.3 Setting Up a Snapshot Grid to Show Tabular Data

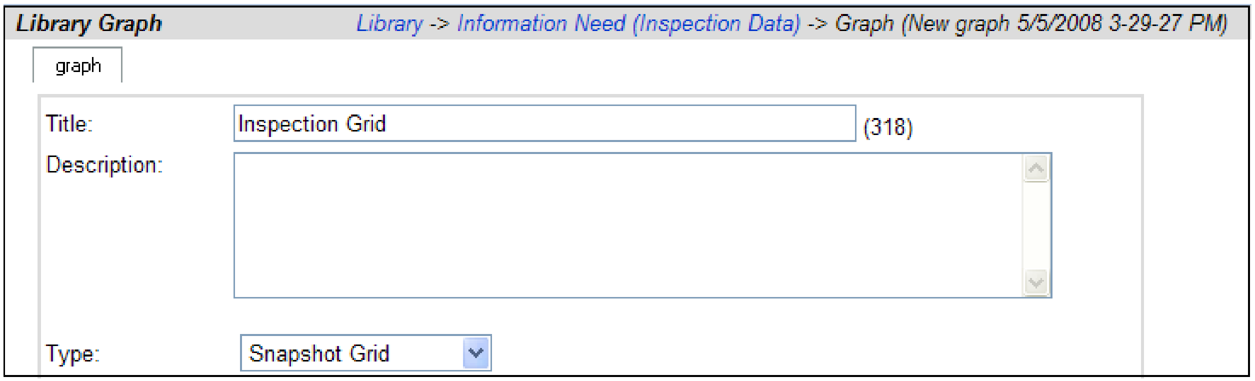
Tabular data can be displayed through a graph type called a Snapshot Grid. The snapshot grid is a data table which displays a set of records of interest to managers. An example of an inspection grid is shown below:



Inspection Date	Document Title	Pages Inspected	Total Errors	Total Time
1/1/2008 12:00:00 AM	JTL_INSP	5	10	15
2/1/2008 12:00:00 AM	RLP_INSP	7	14	21
3/1/2008 12:00:00 AM	SSB_INSP	10	20	30
4/1/2008 12:00:00 AM	MVR_INSP	15	30	45
5/1/2008 12:00:00 AM	JTL_INSP	20	40	60
6/1/2008 12:00:00 AM	RLP_INSP	25	50	75
7/1/2008 12:00:00 AM	SSB_INSP	28	56	84
8/1/2008 12:00:00 AM	MVR_INSP	32	64	96
9/1/2008 12:00:00 AM	TSV_INSP	36	73	108
10/1/2008 12:00:00 AM	JTL_INSP	40	81	120
11/1/2008 12:00:00 AM	RLP_INSP	45	89	135
12/1/2008 12:00:00 AM	SSB_INSP	49	97	147

Any data records (in one of the data mart databases) can be displayed in a Snapshot Grid. For example, if an information need contained a run graph of software inspections held each week, you could define a snapshot grid to show the details for all completed inspections.

A snapshot grid is actually a type of graph, with the graph type set to Snapshot grid. In the Library for example, the snapshot grid is one of the available Graph types (as shown in the figure below). You can add any number of snapshot grids to an Information Need in the Library tab.



Library Graph   Library -> Information Need (Inspection Data) -> Graph (New graph 5/5/2008 3-29-27 PM)

Title: Inspection Grid (318)

Description:

Type: Snapshot Grid

To display a snapshot grid, follow three steps:

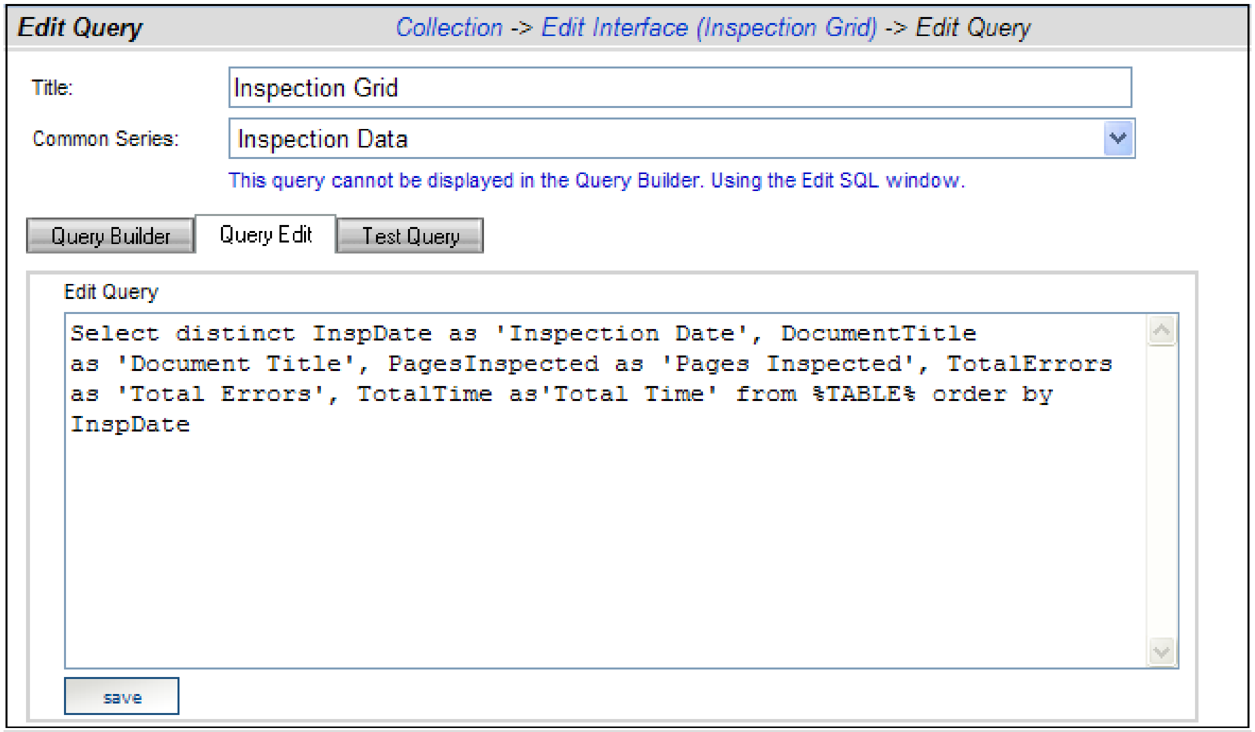
1. Create a SQL query in an interface (Collect tab)
2. Create a snapshot grid in an Information Need (Library tab)
3. Assign the Information need to your Unit (Status tab)

Once the snapshot grid and associated query is defined, you assign an information need to your unit and appropriate data (that is, data which meets the conditions of your query) will be displayed. Detailed instructions for each step are below.

5.3.1 Step 1: Creating a SQL Query within an Existing Interface.

To add a SQL Query to the Interface, follow these steps:

1. Click on the Collection tab.
2. In the gray panel, click on Interfaces.
3. Click on the edit button for the Interface in which you want to add a snapshot grid.
4. In the Edit Interface page, click on the queries tab.
5. Click on add (add a new single series query to this interface) button.
6. Enter a Title for the snapshot grid query.
7. Enter the query by using the query builder or the query edit tab.
8. Save your changes.


Edit Query
Collection -> Edit Interface (Inspection Grid) -> Edit Query
Query Builder
Query Edit
Test Query
save

Title: Inspection Grid

Common Series: Inspection Data

This query cannot be displayed in the Query Builder. Using the Edit SQL window.

Edit Query

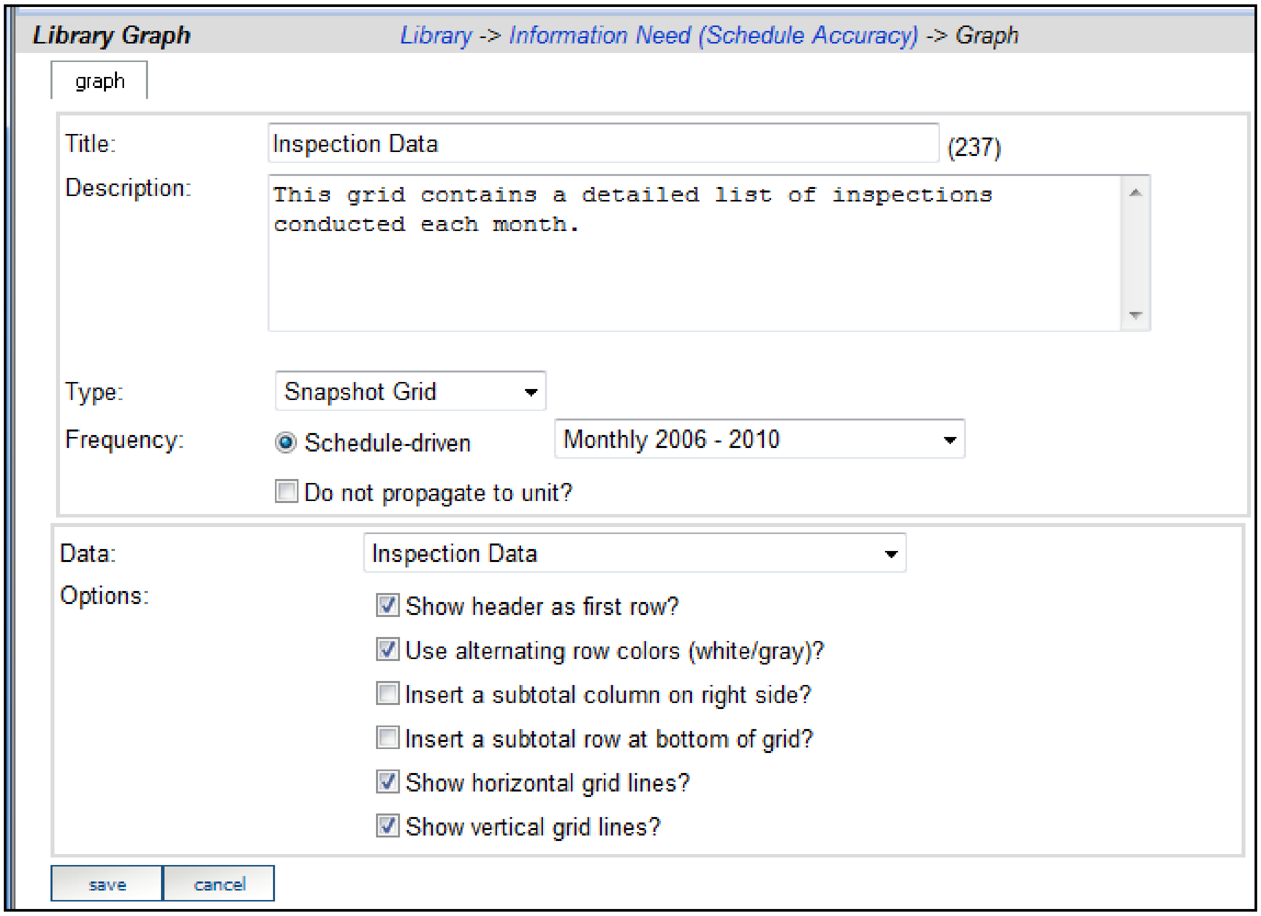
```
Select distinct InspDate as 'Inspection Date', DocumentTitle
as 'Document Title', PagesInspected as 'Pages Inspected', TotalErrors
as 'Total Errors', TotalTime as'Total Time' from %TABLE% order by
InspDate
```

In the above diagram, notice that the fields (or columns) are defined using the "as" clause so that the headings in the snapshot grid have a more readable/understable column heading.

5.3.2 Step 2: Adding a Snapshot Grid to an Information Need.

To add a snapshot grid to your information need, follow these steps:

1. Click on the library tab. Click on the Edit button for the Information Need where you want to add the data table.
2. Click on the graphs tab in the Information Need. Click on the add button.
3. In the graph tab, enter the following information:
   a. Title - enter the name for the snapshot grid.
   b. Type – select snapshot grid from the drop down. (This action will cause the page to reload with only the graph tab available).
   c. Frequency – click on the Schedule-driven option.
   d. Measure – click on the measure that is associated with the information need.
   e. Options – check the box to engage the option.
4. Save your changes.



Title: Inspection Data (237)

Description: This grid contains a detailed list of inspections conducted each month.

Type: Snapshot Grid

Frequency: ● Schedule-driven Monthly 2006 - 2010

□ Do not propagate to unit?

Data: Inspection Data

Options:

☑ Show header as first row?

☑ Use alternating row colors (white/gray)?

□ Insert a subtotal column on right side?

□ Insert a subtotal row at bottom of grid?

☑ Show horizontal grid lines?

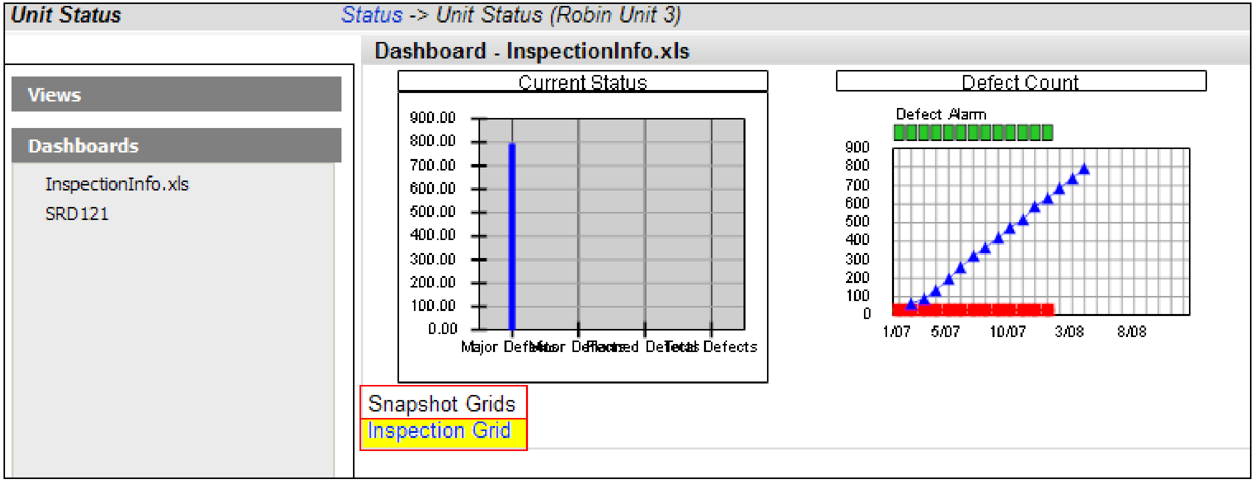
☑ Show vertical grid lines?

5.3.3 Step 3: Assigning the Information Need to your Unit

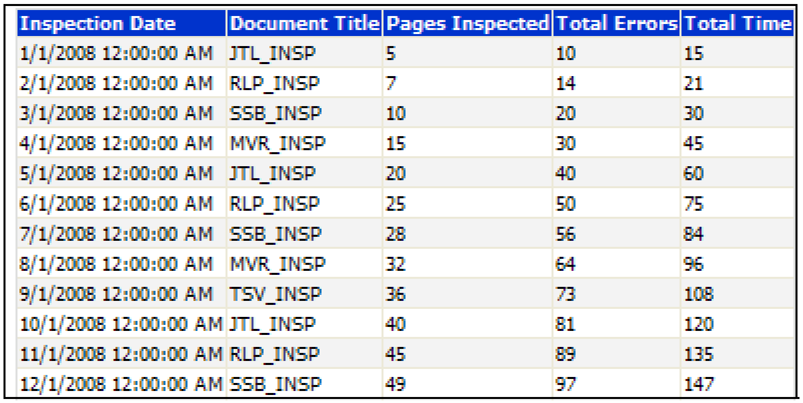
To see the Snapshot Grid in your Dashboard, follow these steps:

1. Click on the Status tab and click on the Unit in which you wish to add the snapshot grid.
2. Click on the Details icon to open Unit.
3. In the navigation panel on the left side, click on the Definition group and then the Information Needs command.
4. From the drop down list, select the Information Need that contains the snapshot grid.
5. Select the Interface associated that contains the query for the snapshot grid.
6. Check boxes: Assign information need to existing items and Perform refresh after assignment.
7. Click on the assign button.
8. The new Information Need will be listed in a grid directly below the Information Need Definition.
9. In the navigation bar on the left hand side, click the Dashboard group and then click the command that contains the information need.
10. In the Dashboard, under Snapshot Grids, click on the link. This will open the snapshot grid.

Each dashboard contains a set of graphs, followed by a list of snapshot grids. Each item in the list is a hyperlink that will open the selected grid. In the image below, there is one snapshot grid, called “Inspection Grid” in the “InspectionInfo” dashboard.



Once you click on the hyperlink for the grid, the snapshot grid for inspection data will be displayed as shown below (if you used the same SQL query as provided above).



Inspection Date	Document Title	Pages Inspected	Total Errors	Total Time
1/1/2008 12:00:00 AM	JTL_INSP	5	10	15
2/1/2008 12:00:00 AM	RLP_INSP	7	14	21
3/1/2008 12:00:00 AM	SSB_INSP	10	20	30
4/1/2008 12:00:00 AM	MVR_INSP	15	30	45
5/1/2008 12:00:00 AM	JTL_INSP	20	40	60
6/1/2008 12:00:00 AM	RLP_INSP	25	50	75
7/1/2008 12:00:00 AM	SSB_INSP	28	56	84
8/1/2008 12:00:00 AM	MVR_INSP	32	64	96
9/1/2008 12:00:00 AM	TSV_INSP	36	73	108
10/1/2008 12:00:00 AM	JTL_INSP	40	81	120
11/1/2008 12:00:00 AM	RLP_INSP	45	89	135
12/1/2008 12:00:00 AM	SSB_INSP	49	97	147

5.3.4 Writing Queries for Snapshot Grids

When setting up a measurement program, it often helpful to see a table of supporting data that explains what caused the behavior of a series. For example, if the series “number of tasks started late” goes up one month unexpectedly, then a manager may want to see which tasks are late. In IBM Rational Dashboard, a snapshot grid can display data in a tabular form, such as a list of late tasks. Snapshot grids can be added to information needs, along with graphs, to provide supporting details for managers.

A snapshot grid uses a single query as its data source. The query is added to the interface, along with other single-value queries. The snapshot grid query uses the same format and tags as other queries. The query for a snapshot grid must define the data columns (or fields) in the order that you want them to appear in the snapshot grid, and also order the resulting records in the desired fashion. Unlike a single-value query for a data series, the query for a snapshot grid does not use a SUM, COUNT or other aggregate function to return a single value.

For example, below is a snapshot query that returns the tasks which started late in the last period.

SELECT task_title, task_start_date as PlannedStart, task_act_start as ActualStart FROM %TABLE% WHERE g_itemid=%ITEMID% AND (task_start_date BETWEEN %STARTDATE% AND %ENDDATE %) and task_start_date < task_act_start ORDER BY task_start_date

The query returns all records in a schedule that were planned to start in the last period and were started late (that is, the actual start is after the planned start date). Since the column names are used as headers in the snapshot grid, the "AS" clause is used to provide more readable headers than the column names. Finally, the "ORDER BY" clause sorts the records by planned start date from the earliest (i.e. start of the period) to the latest (i.e. end of the period).

5.4 Understanding the! FIELD! Tag in the Collector

When the Collector is run, it obtains the interface definition for each data source that is being run. The interface definition contains a list of field sets, where each field set contains a table name; a field set replacement tag and a list of fields. The Collector then builds whatever interface code/script/API is required for the data source and runs a process to collect the desired data. For example, when collecting from IBM Rational Doors, the Collector collects object attributes using DXL. When collecting from a SQL database, the Collector collects data base fields (sometimes called columns) using a SQL statement.

Consider when the Collector runs to collect data from a SQL, Oracle or ODBC database: after the Collector obtains a list of items, the Collector must collect data for each item by generating a SQL statement. This SQL statement must request the fields that were defined in the interface. To simplify the coding of the SQL statement, the Collector provides a replacement tag that looks like! FIELDS!. This tag is replaced with a comma separated list of all the field names that were defined in the interface.

Looking at an example of the !FIELDS! tag, if an interface for a SQL database contains three fields (TaskName, StartDate and EndDate), the Collector could be configured with a "Get Details" query that looks like the following:

```
      SELECT !FIELDS! FROM TaskTable WHERE ProjectId=!ITEM!
```

The two tags above, !FIELDS! and !ITEM!, are filled in at run-time by the Collector, to look like the following (though the ITEM value depends on actual data):

```
SELECT TaskName, StartDate, EndDate FROM TaskTable WHERE
ProjectId=1
```

The! FIELDS! tag is very useful when the syntax of the collection allows for a comma separated field list, as is the case with a SQL statement.

5.5 Understanding the !SETTAG! in the Collector

In an interface, the field set replacement tag is an optional field in a field set. Each field set has it's a replacement tag, a table name and a set of fields. During collection, field set information is passed to the Collector. Some parameters in a data source are replaced at run-time by the field set replacement tag. In particular, the replaceable parameters are the "getlist" and "getdetails" queries from each type of interface.

In the Collector data source, you may enter the text! SETTAG!, which signals that the Collector should replaced the text! SETTAG! with the text entered into the that was entered into the Portal. You may use the !SETTAG! to replace either part of the data source parameter, as in:

```
SELECT * FROM ProjectList WHERE !SETTAG!
```

Or, you may use the! SETTAG! to replace the entire parameter, as in:

```
!SETTAG!
```

If there are errors with this tag replacement, you should consult the Collector log file for more details.

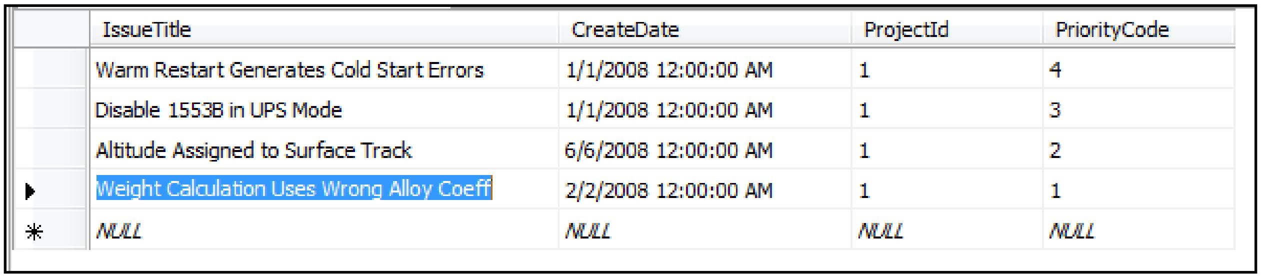
5.6 Using the Field Set Replacement Tag for SQL Joins

When you need to combine data from more than one table, you are typically performing what is called a join query. In a join query, you must use an identifier (which is usually not human readable) to look up a readable or descriptive string from another table. The other table(s) is frequently referred to as a look-up table.

To use a JOIN statement, you use the field set replacement tag,! SETTAG!. The! SETTAG! is basically a method to write more advanced SQL statements without being encumbered by the simplicity of the! FIELDS! tag. An example, presented in the next subsections, walks through how to configure and run a SQL join using the! SETTAG!.

5.6.1 A Sample Database

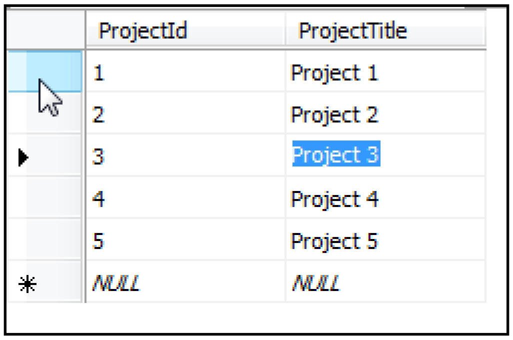
In this sample, the external application is a Microsoft SQL database. The database contains a table called “IssueData”, from which we would like to collect data. The IssueData table has four fields and there are four records in the table, as shown below.



IssueTitle	CreateDate	ProjectId	PriorityCode
Warm Restart Generates Cold Start Errors	1/1/2008 12:00:00 AM	1	4
Disable 1553B in UPS Mode	1/1/2008 12:00:00 AM	1	3
Altitude Assigned to Surface Track	6/6/2008 12:00:00 AM	1	2
Weight Calculation Uses Wrong Alloy Coeff	2/2/2008 12:00:00 AM	1	1
NULL	NULL	NULL	NULL

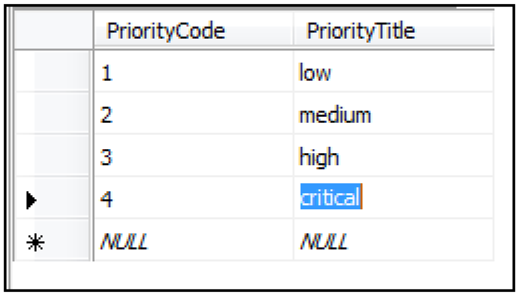
Notice that the table has a “ProjectId” field but we cannot tell the Project name or title. Also, there is a “PriorityCode” field but we cannot tell what the priority is (such as low, medium or high).

The database also has two lookup tables, one for the project title and one for the priority title. Both tables contain an Id field and a text field. The project look-up table is shown below.



ProjectId	ProjectTitle
1	Project 1
2	Project 2
3	Project 3
4	Project 4
5	Project 5
NULL	NULL

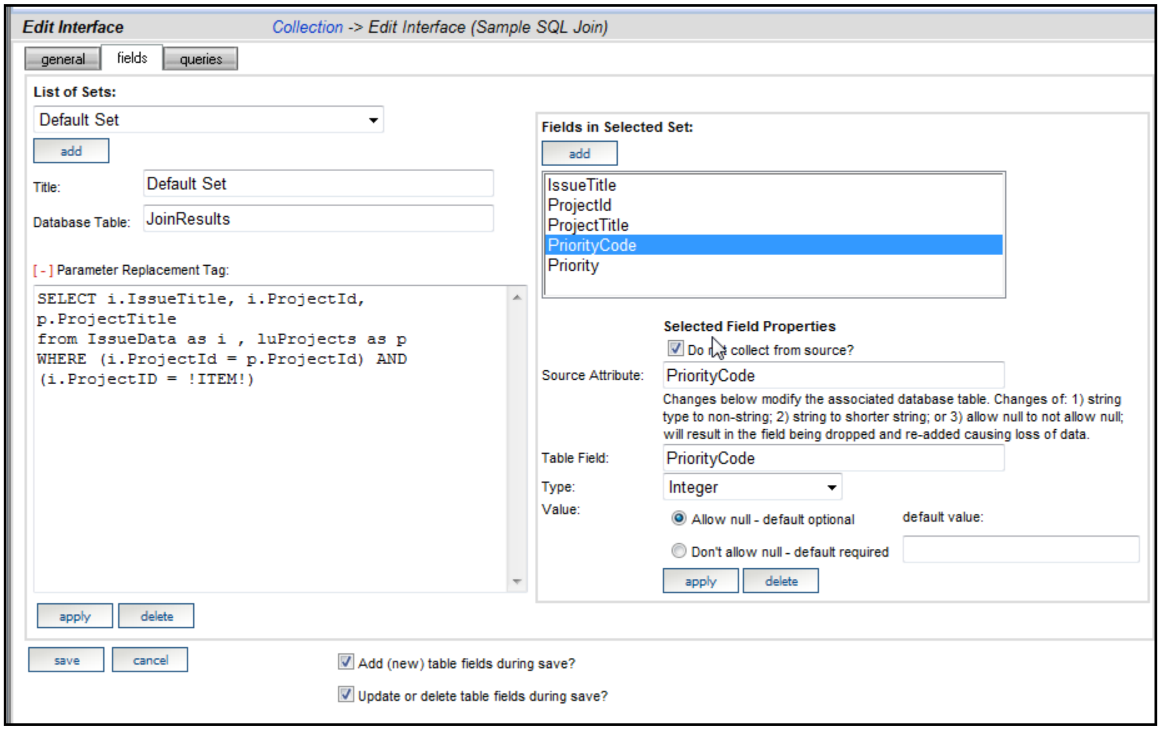
Similar to the project table, a priority look-up table is shown below. The priority table is not used in the sample (described in subsequent paragraphs), but can be used as a follow-on to expand the sample from a two-table to a three-table join.



	PriorityCode	PriorityTitle
	1	low
	2	medium
	3	high
▶	4	critical
✱	NULL	NULL

5.6.2 Configuring an Interface to Perform a Join

The IBM Rational Dashboard Portal provides the means to define an interface, the field sets within the interface and the fields within each field set. An interface called “Sample SQL Join” is shown below with the default field set being configured to perform a SQL join on one table.



List of Sets: Default Set

Title: Default Set

Database Table: JoinResults

[-] Parameter Replacement Tag:

```
SELECT i.IssueTitle, i.ProjectId,
p.ProjectTitle
from IssueData as i , luProjects as p
WHERE (i.ProjectId = p.ProjectId) AND
(i.ProjectID = !ITEM!)
```

Fields in Selected Set:

- IssueTitle
- ProjectId
- ProjectTitle
- PriorityCode
- Priority

Selected Field Properties

☑ Do not collect from source?

Source Attribute: PriorityCode

Changes below modify the associated database table. Changes of: 1) string type to non-string; 2) string to shorter string; or 3) allow null to not allow null; will result in the field being dropped and re-added causing loss of data.

Table Field: PriorityCode

Type: Integer

Value: ● Allow null - default optional ○ Don't allow null - default required

default value:

☑ Add (new) table fields during save?

☑ Update or delete table fields during save?

In the interface (above), the fields defined for the field set named “Default Set” are shown in the following table.

Collect?	Source Attribute	Table Field	Type

Yes	IssueTitle	IssueTitle	integer
Yes	ProjectId	ProjectId	integer
Yes	ProjectTitle	ProjectName	string 50
No	PriorityCode	PriorityCode	integer
No	Priority	Priority	string 10

Notice that the PriorityCode and Priority attributes are defined in the interface but marked as “Do not collect”.

In the interface above, the Parameter Replacement Field (i.e.! SETTAG!) contains a complete SQL statement, which is valid on the source (with the exception of the! ITEM! tag which is filled in by the Collector at run-time):

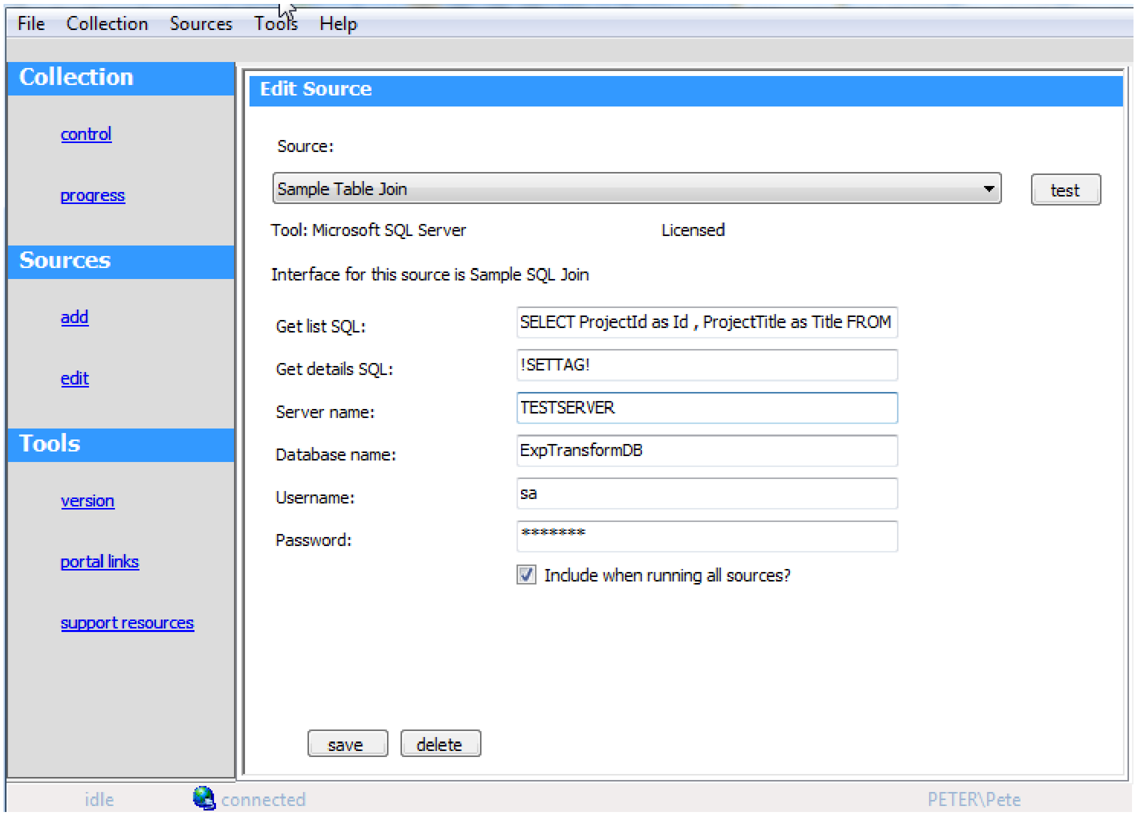
```
SELECT i.IssueTitle, i.ProjectId, p.ProjectTitle
FROM IssueData as i, luProjects as p
WHERE (i.ProjectId = p.ProjectId) AND (i.ProjectID =! ITEM!)
```

In the SQL statement above, notice that the ProjectId field from the IssueData table is joined to the ProjectId field of the luProjects table so that the query can return the ProjectTitle as a field. This type of SQL statement is called a two-table join.

To finish the sample, let’s look at how the Collector is configured.

5.6.3 Configuring the Collector to Perform a Join

To configure the collector for a join, you must remove the! FIELDS! tag and use the! SETTAG!. As shown in the image below, the “Get details SQL” textbox is simply filled in with “! SETTAG!”



Edit Source

Source: Sample Table Join

Tool: Microsoft SQL Server   Licensed

Interface for this source is Sample SQL Join

Get list SQL: SELECT ProjectId as Id , ProjectTitle as Title FROM

Get details SQL: !SETTAG!

Server name: TESTSERVER

Database name: ExpTransformDB

Username: sa

Password: ********

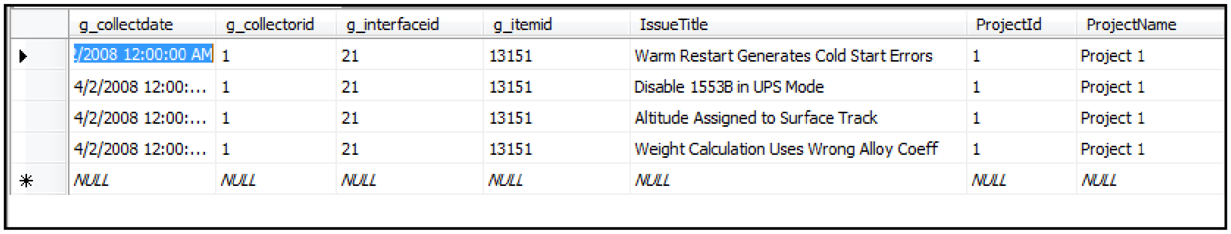
☑ Include when running all sources?

The “Get list SQL” generally does not change as a result of using either the! FIELDS! or! SETTAG! in the details SQL. For example, in the above source, the Get List SQL is:

```
SELECT ProjectId as Id, ProjectTitle as Title FROM luProjects
```

5.6.4 The Resulting Joined Data

The results of the join query using the SETTAG is shown below. The resulting data is stored in a table called “JoinResults”, as specified in the Interface Definition (see previous image).



	g_collectdate	g_collectorid	g_interfaceid	g_itemid	IssueTitle	ProjectId	ProjectName
▶	/2008 12:00:00 AM	1	21	13151	Warm Restart Generates Cold Start Errors	1	Project 1
	4/2/2008 12:00:...	1	21	13151	Disable 1553B in UPS Mode	1	Project 1
	4/2/2008 12:00:...	1	21	13151	Altitude Assigned to Surface Track	1	Project 1
	4/2/2008 12:00:...	1	21	13151	Weight Calculation Uses Wrong Alloy Coeff	1	Project 1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

The ProjectId and IssueTitle fields were captured directly from the IssueData table. The ProjectTitle field data was captured from the luProjects table, using a JOIN clause in the SQL statement to obtain the correct value from the lookup table. Queries in this Interface can use any of these fields to filter, sort or select records of interest.

Notice that the JoinResults table contains a number of fields that start with "g_". These global fields are created automatically for you and are used by IBM Rational Dashboard. The Collector inserts the correct data for these fields as it collects data.

5.7 Maintenance Limitations of Information Needs

Currently, any changes made to Information Needs in the Library tab are not propagated to units where the Information Need has already been assigned. This is something that will be changed in the future, but currently there is no way to propagate changes. For graphing items, like adding series and changing sources, this should not present too much of a problem. The main problem is found if an Information Need was created and assigned without choosing series to be the Current statuses for the View grids. There is no way to change this once the Information Need has been assigned.

The only way to apply a change to an Information Need is to un-assign and reassign it to the unit. This however will cause a loss of formatting and other changes if any were made to the graphs in the Information Need.

5.8 Tags Used in Queries

Text in the format of %ZZZ% are called tags, which allow queries to apply to all items and all periods without having to write a new query for each instance. Each tag is evaluated at run-time and replaced with the appropriate value.

Note that a query containing a tag, such as %TABLE% is not a valid SQL command – if you run a query which contains a tag in a SQL query tool, the SQL statement will generate an error. The Portal Query Test page allows you to run SQL statements containing tags. In addition, it helps you uncover valid values for tags and also to substitute different values for the tags.

The following table describes allowable tags:

Tag	Purpose	Notes
%TABLE%	database table name	The table name of the first field set in the interface. If there is one field set, this tag can be used in all queries. If there is more than one field set (and more than one table in the interface), then this tag contains the name of the first table. Example: SELECT … FROM %TABLE%
%COLDATE%	collection date of schedule period	This tag allows queries to use the collection date of the schedule period being evaluated. This tag can be compared to field in the table, for example to select records before or after the current collection date. This tag is usually compared to the g_collectdate field, whose data is automatically populated by IBM Rational Dashboard when each record is stored. Example: SELECT COUNT(*) FROM %TABLE% WHERE g_collectdate=%COLDATE%
%STARTDATE%	Start date of schedule period	This tag allows queries to use the start date of the schedule period being evaluated. This tag can be compared to data field(s) in the table, for example to select records before or after the start of the schedule period. When used with the start date tag, you can select records of interest in the current period. Example: SELECT COUNT(*) FROM %TABLE% WHERE change_date BETWEEN %STARTDATE% and %ENDDATE%
%ENDDATE%	End date of schedule period	This tag allows queries to use the end date of the schedule period being evaluated. This tag can be compared to data field(s) in the table, for example to select records before or after the end of the schedule period. When used with the start date tag, you can select records of interest in the current period. Example:

		SELECT Count(*) FROM %TABLE% WHERE change_date BETWEEN %STARTDATE% AND %ENDDATE%
%UNITID%	Unit id	Integer identifier of the current unit
%ITEMID%	item id	Integer identifier of the current item This tag is used to select records applicable to one item. Commonly, a single table in a transform database is used to store data for multiple items, for example requirement objects from multiple DOORS modules, or tasks from multiple Microsoft Project schedules. Example: SELECT COUNT(*) FROM MsPTasks WHERE g_itemid=%ITEMID% AND task_actual_start > task_plan_start
%GRAPHSTARTDATE%	start date of graph	The period start date of the first schedule period in a graph Typically used to allow a SLIP chart to determine the correct value for actual series data points Example: SELECT DATEDIFF(day, %GRAPHSTARTDATE%, schedule_end_date) AS DaysToDelivery FROM %TABLE%
%A()%	attribute value	Retrieves an attribute value from the current unit Example: %A(Project Cost)%

6 Security

IBM Rational Dashboard uses a flexible user- and role-based model for security. This section describes the security options for securing tabs and units available in IBM Rational Dashboard. The final subsection in this section describes security features provided by the Windows operating system and .NET framework.

6.1 User Licenses and Security

There are three types of licenses that can be purchased for IBM Rational Dashboard. These are shown in the following table:

License	typical number	security and rights
Administrator	2 – 5	full access to all functions and tabs. configures access to functions by non-administrator users
Manager	10 – 50	access to all functions except administration function
Executive	50-1000	read-only access to status tab

Administrators must create a user account for each user in IBM Rational Dashboard. User Login Accounts are added to IBM Rational Dashboard through the Admin tab. Any number of login accounts may be added. Only Administrators may add Login Accounts to IBM Rational Dashboard because only Administrators may access the Admin tab in the Portal. A user must have a Login account to access IBM Rational Dashboard.

Once a user account has been added, that account can be added to a Role. A role is a named collection of one or more accounts. Roles consist of individual user accounts and users may be added to more than one role. Administrators can create as many roles as needed – there is no limit. Administrators can only add individual login accounts to a role. Administrators cannot add a role as a member of another role.

Roles are used to grant a group of users access to information within the tool. Roles are indicated when numerous users share a common interest and need to access the same information. These users are grouped together in a role and the role is assigned permission to access data in the tool. Roles provide security for the tool.

There are four levels of security available in IBM Rational Dashboard.

- ***None***: Members have no access
- ***Read***: Members have read only access.
- ***Edit***: Members can modify existing items, including adding data or modifying the graph properties. Members cannot add or delete within the unit.
- ***Full***: Members are granted full control of the unit. They may add, edit and delete. They may also set modify rights.

6.2 Tab Security

In IBM Rational Dashboard access to the tabs (Status tab, Collection tab, Library tab, Report tab, and Admin tab) is based on the license type and role security. The following table shows how the license types are provided access to each tab.

	Status tab	**Collection tab**	**Library tab**	**Report tab**	**Admin tab**
Administrator	full	full	full	full	Full
Manager	Determined by unit rights	Set by "Collect Tab" role	Set by "Library Tab" role	Full	None
Executive	Read-only applied to Unit rights	None	none	full	None

Administrator users have access to all tabs and all functions in them. Administrators are also granted access to all units, regardless of whether they have been granted rights within the unit. All tabs are visible to all users. Tabs are disabled for users who do not have access.

Manager users can also be given access to additional tabs without being granted full administrative access. There are two reserved roles in the Portal "Collect Role" and "Library Role". Users with a manager license can be granted access to one of the tabs, by including them in the corresponding role. Notice that only administrators can modify role membership.

6.3 Create a User Account

To perform this task, you must be an administrator (as this task requires access to the admin tab).

To create an account:

1. Click on the **Admin** tab in the IBM Rational Dashboard Portal
2. Click **Accounts** from the left hand column
3. Click **add** on top of the grid. The *Edit Account* page will display.
4. Enter **login name**, **title and email address** of the user.
5. If this user should have an allocated license, assign the appropriate license. Note that without a license, the user will receive an error message when they access the Portal.
6. Click save to create the account.

The login name must match exactly the credentials that will be provided by the authentication method configured in the web.config file. For example, if you are using windows integrated authentication, then the login name must be of the form “MACHINE\login” or “DOMAIN\login”.

To review the number of licenses you have purchased, click on the “Version Info” command on the Admin tab. The Version Information page lists the number of each type of license purchased and the number that have been allocated. You can allocate up to the

In order to login, a user account requires a license.

6.4 Assign or Change the Owner of a Unit

To perform this task, you must be an administrator or a manager with full rights within the selected unit.

To assign or change the owner of a unit:

1. From the **Status tab** in the IBM Rational Dashboard Portal, select the **Unit** to apply security to
2. Click on Details button on the right side of the page
3. Click on the **Definition** section of the menu on the left
4. Click the Unit Properties link
5. Next to the “Owner” label, select a user from the drop down
6. Click the **save** button at the bottom of the page

6.5 *Grant Rights to a User or Role in a Unit*

To perform this task, you must be an administrator or a manager with full rights within the selected unit.

To assign security to unit:

7. From the **Status tab** in the IBM Rational Dashboard Portal, select the **Unit** to apply security to
8. Click on Details button on the right side of the page
9. Click on the **Definition** section of the menu on the left
10. Click the **Security** link
11. Click the **edit** button
12. Click the **add** button to add a new security role
13. From the **first line**, select the right the user or role should have
14. From the **second line**, select whether the right will be applied to a specific user or to a whole role
15. From the list, select the user or role that the right is to be applied to
16. Click the **add** button to add the security rule to the unit
17. Click the **apply** button to apply the settings to the unit.

6.6 *Create a Role*

To perform this task, you must be an administrator (as this task requires access to the admin tab).

To add a new role to the admin tab:

7. Click on the **Admin** tab in the IBM Rational Dashboard Portal
8. Click **Roles** from the left hand column
9. Click **add** on top of the grid. The *Edit Role* page will display.
10. Enter **role** name
11. Click **save** to create the new role. The role must be saves prior to adding users to a role.

To add users to a role:

1. From the **Admin** tab, click **Roles** from the left hand column
2. Click **edit** to the left of the role name. The *Edit Role* page will display.
3. Click the **available members** and click the arrow button to move the role to the left. Add as many members as needed.
4. Click **save**.

6.7 Security Using Windows File Security

In special cases, your organization may want to restrict access to certain pages based on Active Directory rights or membership. For example, perhaps only a few IBM Rational Dashboard administrators should be allowed to add, edit or delete accounts; however you would like to allow them administrative access to other pages and functions in IBM Rational Dashboard.

In such cases, you should consider using file system security to prevent access to pages where specific authenticated users should not have access. For example, in the Windows file system, if you allow only “DOMAIN\usera” to have access to the page “App_UI\Admin\EditAccount.aspx”, then other users will receive a “Not Authorized” error message when they attempt to display it. To use file system security, you must be using Windows authentication for your IBM Rational Dashboard portal and you must also not use impersonation.

7 Synchronizing an External List of Users (Using the UserSync Utility)

7.1 Overview of the User Synchronization Utility

This section describes a windows command line utility called UserSync. The utility synchronizes an external list of users with the list of users in the Portal. This utility allows a group of users from a corporate or organizational authentication repository, such as Active Directory or LDAP, to be automatically created in the Portal list of users.

This utility requires a text file containing a list of users, one per line. The list of users should be extracted from an authentication respository. You generate this text file using the appropriate API or tool. For Active Directory, you can use PowerShell, VBS or any .NET development system to create a text file needed by this application. The users in the text file are created (by the UserSync utility) in the list of users in the Portal (if they do not exist) and added to a role which is specified on the command line.

This utility can synchronize all types of licenses (executive, manager and administrator). You may run this utility three times, for example, once time to synchronize a file with the correct type of user.

7.2 Command Line Operation

The utility is called with the following parameters on the command line:

Parameter	**name**	**Values**	**description**
/file	file name	Folder path and file name of a file	a text file containing a list of users, where the list of users has been extracted from an authentication repository such as Active Directory or LDAP
/role	role name	name of role in IBM Rational Dashboard	name of a Portal role which will contain the users

/license	license type	se,sm,sa	One of:
			se = static (named) executive
			sm – static (named) manager
			sa = static (named) administrator
			fe = floating executive
			fm = floating manager

/action	action on delete	d,n,s,u	Indicates the action to take when a user has been deleted from the authentication

			repository but exists in the Portal list of users. One of: d = delete the user from IBM Rational Dashboard n = take no action s = suspend the user in IBM Rational Dashboard u = unlicense the user in IBM Rational Dashboard

The file specified by the “/file” parameter must be a comma seperated value file, where each row has the following format:

```
"login name" , "Written Name" , "email address" , "initials"
```

Notice that each user must be on one lines - you may not span rows with the information for a single user. Also, there is no header row in the file. Blank rows are ignored. A sample file with two entries is as follows:

```
"distributive\mark.chambers" , "Mark Chambers" , "mark.chambers@distributive.com" , "MC"
"distributive\lisa.lasalle" , "Lisa Lasalle" , "lisa.lasalle@distributive.com" , "LL"
```

To run the UserSync utility, you may start a DOS command window or use the “Run” command in the Windows Start bar. Sample command lines are shown below:

```
UserSync.exe /role="Sync Executives" /file="SyncExecutives.txt" /license="se" /action="n"

UserSync.exe /r="Sync Executives" /f="SyncExecutives.txt" /l=se /a=n
```

Notes:
1. If run with no arguements, the utility displayes a help page.
2. The URL of the IBM Rational Dashboard web services is stored in the app.exe.config file in the following location:

```
 <configuration>
   <applicationSettings>
     <UserSyncUtility.My.MySettings>
       <setting name="UserSyncUtility_ExpressWebServices_WSAPI">
```

3. The web services must have a \temp subfolder in which to create temporary files.

Tips and Troubleshooting

This section presents information that assists in configuring and troubleshooting the operation of the IBM Rational Dashboard. Note that the Installation Guide contains information for troubleshooting common problems encountered when installing and setting up IBM Rational Dashboard.

7.3 Use Only One Field Set for DOORS

For DOORS collections, make sure that there is only one field set defined for the interface. There should only be one entry in the 'Sets' pull down list in the Interface's Field's tab. You can navigate to this from the Collection Tab and then Collection > Interface > Fields. There is no reason to use more than one field set for DOORS collections. Sets exist for collecting from other tools.

7.4 Check the Collection Schedule

In the Unit Definition, there is an option to set the schedule mode to either "basic mode" or "advanced mode". Initially and in most circumstances, you should use "basic mode" which will set the unit, all items in the unit and all graphs to use the same schedule. If you select "advanced mode", then you can set graphs, items and the Unit to use different schedules.

Make sure that your scheduled collection dates are what you expected.

a. There are several options available when defining schedule dates so double check that collection periods and collection dates match your expectation. Look in the Admin tab and review the collection dates of your schedule.
b. The collection date can be set as the last day in the period. This is not always logical because then, the collection occurs during the period itself. It is more common to collect the day following the end date.
c. Collection will ONLY occur on the collection date. Therefore if you collect one day after the collection date, nothing will be collected! There are two ways around this. When performing a collection in the Collector, you can use the second radio button and specify the collection date, regardless of whether the present date is before or after the collection date or you can use the third radio button to recollect all collection dates from a previous date to the present.
d. Multiple units can be assigned the same schedule so use the Unit's date range to limit a schedule instead of deleting collection dates from the schedule.

7.5 Refresh in the Portal after Running Collection

After collecting, you may need to manually refresh a graph or refresh at the project level. Refreshing at the project level is simpler. To refresh from the project level, go to the unit details then use the vertical menu to navigate Definition > Unit Properties > Refresh. The easiest way to refresh is to set the start date for the previous year and the end date to the next year – the Portal ignores any dates outside the range of the valid data. To refresh a graph, click on the refresh icon next to the ruler and keyboard icon.

7.6 Refresh in the Portal after Changing an Equation

If you have just modified an equation that references a series or an equation that references equations that contain series, you will need to do a full refresh. Refreshes of data and calculations of data occur at different levels – data itself, equation of data, equation using equations of data, etc. To refresh a graph, click on the refresh icon next to the ruler and keyboard icon. Your best bet is to refresh at the Unit (i.e. project) level (explained above).

7.7 If No Data Points Are Displayed

There are several configuration things that can be checked if no data is seen in graphs or View grids. First, make sure that the collection has been completed. If there were errors in collection then the data may not be in the database. If the collection has completed, check the database to make sure viable data has been collected. This is especially good to check if it is only one graph that is not showing data while others in the same information need are showing data from the collection fine. If the data is in the database, or if you don't have access to the database, the next step would be to check the queries in the interface. Check the queries for table names, tags, and any fields that are used in the query. Again, this is most useful if the problem is only seen for one or two graphs in an information need.

If there is no data seen in any of the graphs, but there is data in the database, check the schedules. This is done in the Admin tab under Schedules. If the day the collection was run does not match up with a collect date of the schedule that is being used for the item, no data will appear. You will see a message in the Collector log file for each item as to whether a schedule was matched.

If there is data in the graphs without any problems and alarms have status colors, but there is still no colors or data in the various Views in the Unit Status page, the likelihood is that the Current Status, Current Plan, and Current Actual were not set in the Information Need. In an Information Need, go to the Graphs tab. There are drop down lists of viable series for the Current Status, Plan, and Actual. If there are no selections in the Information Need, there will be no data in the Views. Checking these conditions can also help to determine which series the View data is coming from if there is missing data for a graph that is showing in the View.

7.8 Check the Log Files

There are three log files that are available to do trouble shooting, one for the collector, one for the portal itself, and one for the Web Services that coordinate between the two. The Portal log file (PortalLog.txt) is found in the Logs folder of the Portal folders location. (The default installation is in C:\Program Files \ IBM \Rational \Dashboard \ Portal \ Logs.) The Web Services log file (wslog.txt) is found in the Logs folder in the WebServices folder. (The default installation is in C:\Program Files \ IBM \Rational \Dashboard \ WebServices \ Logs.) The Collector log file can be accessed from the Collector itself by clicking on File—Logging…, which will bring up the log file. It can also be accessed in the main Collector folder (logfile.txt).

For the Collector log, a successful item collection will look like the following:

```
2006-06-12 16:15:07.0286 INFO frmMain 1820 Distributive.frmMain.DoCollect Set to collect for
item TestModule0
2006-06-12 16:15:07.0286 INFO frmMain 1820
Distributive.WinApps.Tools.CToolDoors.GetDetails Calling DOORS for details....
2006-06-12 16:15:07.5442 INFO frmMain 1820
Distributive.WinApps.Tools.CToolDoors.GetDetails Returned from DOORS call for details....
2006-06-12 16:15:07.5442 INFO frmMain 1820 Distributive.frmMain.DoCollect Details collected
for item TestModule0
2006-06-12 16:15:07.9661 INFO frmMain 1820 Distributive.frmMain.DoCollect Sent data for item
through web services.
```

An unsuccessful item collection may look like the following:

```
2006-06-12 16:12:51.7612 INFO frmMain 2352 Distributive.frmMain.DoCollect Skipping
collection of item TestModule0. Not assigned or not collect date.
```

This specific error message indicates that either the item has not been assigned to a unit in the Portal, or that today's date is not a collect date in the schedule for the item.

7.9 Check the Graph Scale to More Easily See Wide Data Ranges

If a series is not showing up on a graph, it may be because you are plotting two series that are numerically far apart, in other words, you may be in thousands on one series and tens in another series. To fix this problem, go into the graph properties icon and assign one of the two series to a secondary y axis. This will allow each series to have its own scale so you will see them both on the same graph.

7.10 Set a Filter on Your Firewall for the Portal and Web Services

The Windows Firewall must be configured to allow port 80 access, or be turned off on the Dashboard server, for IIS to be able to render web pages to other PCs. The Collector requires access to Web Services which are hosted by IIS and users require access to the Portal which is also hosted by IIS. You can review the IIS log files to see why requests are being rejected.

7.11 Checking Configuration of DOORS Collection

If you are having difficulty getting modules and projects from DOORS, try the following:

- Dashboard collects DOORS data based on the module's default view. Note that the default view of the user takes precedence over the default view of the module.
- Make sure that the DOORS user name used in the Source of the Collector has access to the modules that you are trying to connect to and has access to the parent project of the modules that are being collected.
- Make sure that DashboardCollect is set to true for the modules that you want to collect. If you want to disable a module from being collected, just set the value to false.
- Make sure that the DashboardType is set to the exact string, including case and space characters, as the corresponding Interface's Type Identifier field. For example, "Module" and "module" is not an exact match.
- The Collector will collect a maximum of 50 characters from a DOORS attribute of type text. This limits the analysis of text based data. To work around this, create a DXL attribute in DOORS to perform the string calculation and then collect the DXL attribute.
- In the Collector, make sure that the 'Path to Client' field for the Source properties is correct.

Also, review the Collector's log file to see if any errors were encountered. If it appears that collection was attempted but no data was found, you can review the files in the Collector's \temp folder where the DXL that is run is located, as well as the output files for each module.

7.12 Working with Large Number of DOORS Modules

Collecting from DOORS can take a VERY long time if you have an endless list of modules but you are only collecting a subset. This is because Dashboard opens every module to see if the Dashboard Collect attribute equal true. DXL now has a feature to allow you to get access to an attribute without opening the module. Today a great workaround exists; have Dashboard login as a DOORS user that only has the rights to the modules that are configured to be collected by Dashboard.

7.13 Disable Virus Scanning on the Portal and Web Services Folders

The .NET framework provides a worker process which monitors the health and configuration of ASP.NET applications. When the worker process detects that a file has changed, it recycles, or stops and restarts itself, so that any configuration changes will be processed by ASP.NET applications. While this is generally very useful, when the worker process recycles, users lose any session information such as breadcrumbs or selected items.

Unfortunately, when a virus scanner runs, or is monitoring an ASP.NET application, it can cause the ASP.NET worker process to recycle on a periodic basis. For example, when the Portal writes a temporary graphic file to the file system, the virus scanner will open and scan the file, which the ASP.NET worker process interprets as a "foreign" system updating the file of an ASP.NET application, so it recycles ASP.NET. The end result is that ASP.NET

To work around this, we recommend that you add an entry to the exception list of your virus scanner to not actively monitor application folders for ASP.NET applications. Instead, schedule virus scanning when the application or IIS can be taken off-line.

7.14 Other Tips and Tricks

- If you assign a container instead of a module to a Unit, it will add all modules in the downward hierarchy.
- You can create a new Interface from an existing Interface. Just add an Interface and use the pull down box under the 'Copy' label.
- You cannot add a new graph to an Information Need that has been assigned to a Unit. You can add, change and remove a series from a graph but you cannot add a new graph to an instantiated Information Need. To do so, you need to modify the Information Need in the Library or create a new Information Need in the Library then reassign/assign the modified/new Information Need to the unit.
- To add a values table below the graph, use the graph format page. Graph Properties - Graph Display – Value Grid
- To quickly copy an image, a tool bar comes up in Internet Explorer when you hover over the graph to email or print the graph. You can also right click and copy a graph and paste it into PowerPoint.

- Run graph always shows over time, whether it is a bar chart or line. A non-run chart will only show the current value.

7.15 Maximum Request Length Exceeded ASP.NET Error

You may receive this error when you view a log file through the portal that is 3 Megabytes or higher. You may receive a different error that is less descriptive if you are connected to the Portal from a different machine other than the server. If this occurs you can archive your current portal log file by going to the default installation directory and locating \portal\log\portallog.txt and renaming or copying it elsewhere. You can then create a new text file called portallog.txt in place of it that is empty to start over.

7.16 "Graph Format" always says [DIFFERENT] after synchronization

You may see this because the schedule set on this graph in the library is different than the schedule set on your unit when you are in the synchronization screen. This is not a bug and is by design. As long as your graphs schedule in your unit matches the unit's defined schedule then your graph should display properly. It is also best to not update your graph because it can reset that graphs schedule and override the current unit's schedule and cause problems displaying data.

7.17 Executive users assigned to Summary Unit can see information about units they are not assigned too.

If you have executive users assigned to a summary unit then it is assumed that they will be able to see all units that the summary unit contains. This is by design.

8 Summary

Many organizations do not have an effective metric or measurement process, forcing managers to manage without data. Requirements metrics are key indicators of project scope, growth, stability, and progress. Managers who use requirements engineering metrics can spot trouble before a software project becomes a death march. By better managing requirements engineering, managers ensure that they deliver a quality product that completely satisfies customer expectations.

IBM Rational Dashboard puts software management best practices within reach of every organization and every manager. IBM Rational Dashboard contains a Built-In Library of Best Practices for project planning, Automated Collection for management reporting and Compliance Tracking capability to allow managers to manage by exception.

Contact Information

This chapter contains the following topics:

- Contacting IBM Rational Software Support
- Prerequisites
- Submitting problems
- Other information

Contacting IBM Rational Software Support

If the self-help resources have not provided a resolution to your problem, you can contact IBM Rational Software Support for assistance in resolving product issues.

Note: If you are a heritage Telelogic customer, you can go to http://support.telelogic.com/toolbar and download the IBM Rational Telelogic Software Support browser toolbar. This toolbar helps simplify the transition to the IBM Rational Telelogic product online resources. Also, a single reference site for all IBM Rational Telelogic support resources is located at http://www.ibm.com/software/rational/support/telelogic/

Prerequisites

To submit your problem to IBM Rational Software Support, you must have an active Passport Advantage® software maintenance agreement. Passport Advantage is the IBM comprehensive software licensing and software maintenance (product upgrades and technical support) offering. You can enroll online in Passport Advantage from http://www.ibm.com/software/lotus/passportadvantage/howtoenroll.html.

- To learn more about Passport Advantage, visit the Passport Advantage FAQs at http://www.ibm.com/software/lotus/passportadvantage/brochures_faqs_quickguides.html.
- For further assistance, contact your IBM representative.

To submit your problem online (from the IBM Web site) to IBM Rational Software Support, you must additionally:

- Be a registered user on the IBM Rational Software Support Web site. For details about registering, go to http://www-01.ibm.com/software/support/.
- Be listed as an authorized caller in the service request tool.

Submitting Problems

To submit your problem to IBM Rational Software Support:

1. Determine the business impact of your problem. When you report a problem to IBM, you are asked to supply a severity level. Therefore, you need to understand and assess the business impact of the problem that you are reporting.

   Use the following table to determine the severity level.

Severity	Description
1	The problem has a *critical* business impact: You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.
2	This problem has a *significant* business impact: The program is usable, but it is severely limited.
3	The problem has *some* business impact: The program is usable, but less significant features (not critical to operations) are unavailable.
4	The problem has *minimal* business impact: The problem causes little impact on operations or a reasonable circumvention to the problem was implemented.

2. Describe your problem and gather background information, When describing a problem to IBM, be as specific as possible. Include all relevant background information so that IBM Rational Software Support specialists can help you solve the problem efficiently. To save time, know the answers to these questions:

   - What software versions were you running when the problem occurred?

     To determine the exact product name and version, use the option applicable to you:

     - Start the IBM Installation Manager and select **File** > **View Installed Packages**. Expand a package group and select a package to see the package name and version number.

        - Start your product, and click **Help** > **About** to see the offering name and version number.

    - What is your operating system and version number (including any service packs or patches)?
    - Do you have logs, traces, and messages that are related to the problem symptoms?
        - Can you recreate the problem? If so, what steps do you perform to recreate the problem?
        - Did you make any changes to the system? For example, did you make changes to the hardware, operating system, networking software, or other system components?
        - Are you currently using a workaround for the problem? If so, be prepared to describe the workaround when you report the problem.

3. Submit your problem to IBM Rational Software Support. You can submit your problem to IBM Rational Software Support in the following ways:
    - **Online**: Go to the IBM Rational Software Support Web site at https://www.ibm.com/software/rational/support/ and in the Rational support task navigator, click **Open Service Request.** Select the electronic problem reporting tool, and open a Problem Management Record (PMR), describing the problem accurately in your own words.

      For more information about opening a service request, go to http://www.ibm.com/software/support/help.html

      You can also open an online service request using the IBM Support Assistant. For more information, go to http://www-01.ibm.com/software/support/isa/faq.html.
    - **By phone**: For the phone number to call in your country or region, go to the IBM directory of worldwide contacts at http://www.ibm.com/planetwide/ and click the name of your country or geographic region.
    - **Through your IBM Representative**: If you cannot access IBM Rational Software Support online or by phone, contact your IBM Representative. If necessary, your IBM Representative can open a service request for you. You can find complete contact information for each country at http://www.ibm.com/planetwide/.

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Rational Software Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Rational Software Support provides a workaround that you can implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the IBM Rational Software Support Web site daily, so that other users who experience the same problem can benefit from the same resolution.

Other Information

For Rational software product news, events, and other information, visit the IBM Rational Software Web site on http://www.ibm.com/software/rational/.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.
IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send written license inquiries to:

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

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

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions. Therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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

Intellectual Property Dept. for Rational Software
IBM Corporation
1 Rogers Street
Cambridge, Massachusetts 02142
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

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

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

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental. If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, ibm.com, DOORS, Passport Advantage, and Rational are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at www.ibm.com/legal/copytrade.html.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

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