HP SOA Systinet Workbench

Software Version: 3.10

Report Editor Guide

Document Release Date: January 2009
Software Release Date: January 2009

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Third-Party Web Sites

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

Copyright Notices

Contents

About this Guide

Welcome to the HP SOA Systinet Report Editor Guide. This guide describes how to use Report Editor to create and modify reports for use with SOA Systinet.

This guide contains the following chapters:

- Chapter 1, Report Editor

  Introduces Report Editor, the user interface, and the main use cases.

- Chapter 2, Getting Started

  Explains how to configure your environment and Report Editor.

- Chapter 3, Designing Reports

  Explains how to create and modify report definitions for use with SOA Systinet.

- Chapter 4, Exploring the Database

  Explains how to examine, optimize, and access the database.

- Chapter 5, Deploying Reports

  Explains how to deploy your report definitions to SOA Systinet.

- Chapter 6, Use Case Examples

  Walks you through the main use cases for Report Editor.

- Appendix A, Dialog Boxes

  Describes Report Editor dialog boxes in detail.

- Appendix B, Troubleshooting

Describes some common issues and their solutions.

Document Conventions

This document uses the following typographical conventions:

run.bat make	Script name or other executable command plus mandatory arguments.
[--help]	Command-line option.
either | or	Choice of arguments.
`replace_value`	Command-line argument that should be replaced with an actual value.
{arg1 | arg2}	Choice between two command-line arguments where one or the other is mandatory.
`java -jar hpsystinet.jar`	User input.
`C:\System.ini`	File names, directory names, paths, and package names.
`a.append(b);`	Program source code.
`server.Version`	Inline Java class name.
`getVersion()`	Inline Java method name.
Shift+N	Combination of keystrokes.
Service View	Label, word, or phrase in a GUI window, often clickable.
OK	Button in a user interface.
New→Service	Menu option.

Documentation Updates

This guide's title page contains the following identifying information:

- Software version number, which indicates the software version
- Document release date, which changes each time the document is updated
- Software release date, which indicates the release date of this version of the software

To check for recent updates, or to verify that you are using the most recent edition of a document, go to:

http://h20230.www2.hp.com/selfsolve/manuals

This site requires that you register for an HP Passport and sign-in. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

Or click the **New users - please register** link on the HP Passport logon page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. For details, contact your HP sales representative.

Support

You can visit the HP Software Support Web site at:

http://www.hp.com/go/hpsoftwaresupport

HP Software Support Online provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the HP Software Support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract.

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

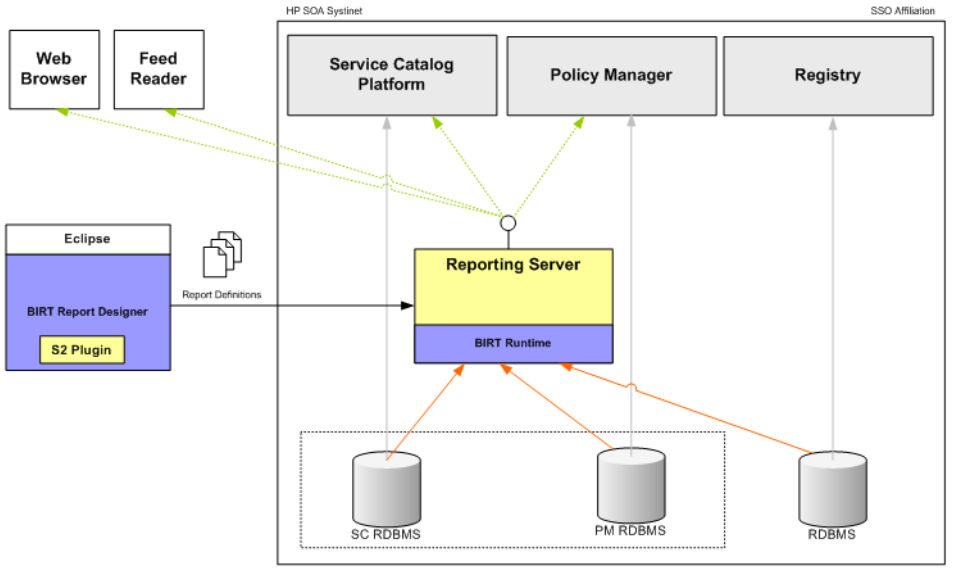
To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

1 Report Editor

Workbench includes Report Editor enabling you to create and deployment new reports easily and simply. The relationship between Report Editor and the SOA Systinet Reporting Service is shown in Figure 1.

Figure 1. SOA Systinet Reporting Framework



HP SOA Systinet Report Editor is an implementation of the Built-In Reporting Tool (BIRT), an open source Eclipse-based reporting system. The Help menu includes the BIRT documentation. The *BIRT Report Developer Guide* describes the general use of BIRT to create reports. The Report Editor Guide adds to this by explaining how to use this functionality in conjunction with SOA Systinet.

This chapter introduces Report Editor in the following sections:

Workbench Suite

HP SOA Systinet Workbench is a suite of editor tools enabling you to customize your deployment of SOA Systinet.

Workbench consists of the following editor tools, distributed as a single Eclipse development platform:

- **Customization Editor**

  Customizes the underlying SOA Definition Model (SDM) and the appearance of these artifacts within SOA Systinet.

- **Taxonomy Editor**

  Customizes the taxonomies used to categorize artifacts in SOA Systinet.

- **Assertion Editor**

  Customizes the conditions applied by your business policies within SOA Systinet.

- **Report Editor**

  Customizes report definitions for use with SOA Systinet.

Overview

Report Editor interacts with SOA Systinet, enabling you to access existing report definitions, create customized reports, and deploy them to the SOA Systinet server.

To access, create, and deploy reports with SOA Systinet, follow this process:

1 Create a reporting project.

For details, see Configuring a Reporting Project on page 32.

2 Create and modify reports.

For details, see Chapter 3, Designing Reports.

3 Deploy your reports to SOA Systinet.

For details, see Chapter 5, Deploying Reports.

4 Access your reports in SOA Systinet.

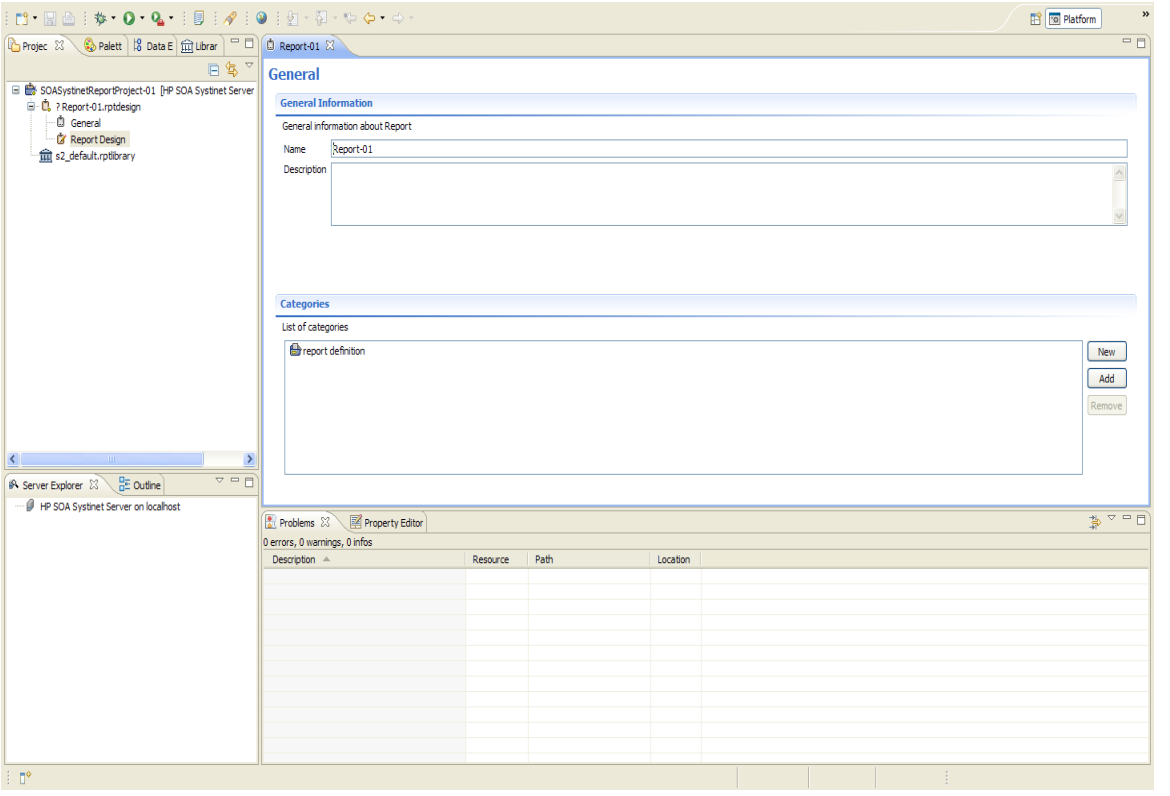
For details, see:

- Use Cases on page 20
- Accessing Reports on page 57
- Chapter 6, Use Case Examples
- *HP SOA Systinet User Guide*

User Interface

The default platform perspective is split into a number of views with menu options across the top, as shown in Figure 2.

Figure 2. Platform Perspective



The perspective consists of the following views:

- **Project Explorer**

  The tree view of your reporting project. For details, see Project Explorer on page 15.

- **Data Explorer**

  The view of the data sources and data sets for a particular report. For details, see Data Explorer on page 17.

- **Server Explorer**

  A list of SOA Systinet servers connected to Workbench. For details, see Server Explorer on page 19.

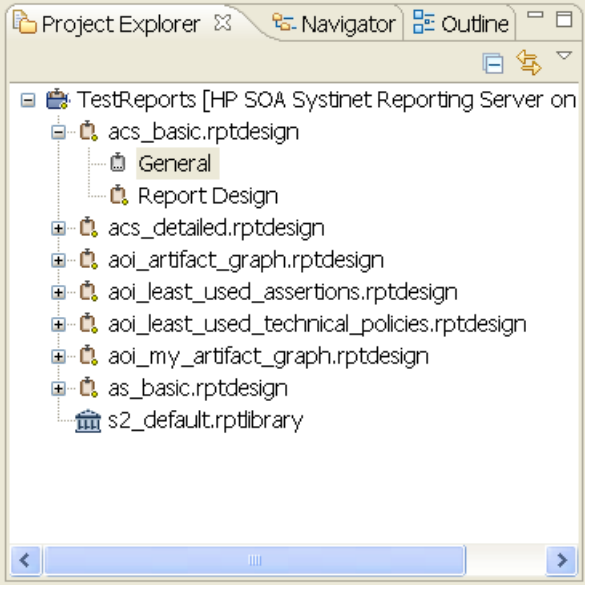
- **Editor Views**

  The main area of the perspective contains report editor views. In this view you can edit the layout and properties of your report definition. For details, see the *BIRT Report Developer Guide*, which you can access from the menu by selecting **Help→Help Contents**.

Project Explorer

The Project Explorer, as shown in Figure 3, is a view of the projects in your workspace and the report definitions they contain. This view is also an interaction point with SOA Systinet.

Figure 3. Project Explorer



Each project contains a set of report definitions and the SOA Systinet library.

Each report definition contains the following sections:

- **General**

  Double-click to open a properties editor view enabling you to change the name and description and to categorize the report definition.

- **Report Design**

  Double-click to open a layout editor view enabling you to change the appearance and content of the report definition.

The Project Explorer contains additional context menu options enabling you to interact with a running SOA Systinet SOA Systinet server. Right-click the project name or a particular report definition, and select **HP SOA Systinet** to view the options listed in Table 1 and Table 2.

Table 1. Project Context Menu Options

Option	Function
Download Report	Import report definitions from SOA Systinet. For details, see Importing Report Definitions on page 34.
Publish Report	Export a report to the default SOA Systinet server. For details, see Deploying a Report to SOA Systinet on page 51.
Publish To Other Server	Export a report to a specified SOA Systinet server.
Build Extension	Create a reporting extension for SOA Systinet containing all the reports in your project. For details, see Creating a Reporting Extension on page 52.

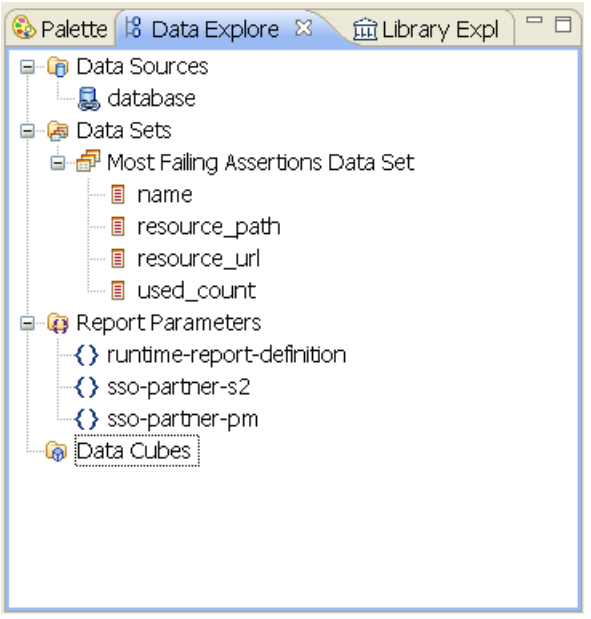
Table 2. Report Definition Context Menu Options

Option	Function
Publish Report	Export the report definition to the default SOA Systinet server. For details, see Deploying a Report to SOA Systinet on page 51.
Update Report	Updates the report with the current version on the server.
Remove Report From Server	Remove the report definition from the default SOA Systinet server with an option to delete it from the report project as well.
Publish to Other Server	Export the report definition to a specified SOA Systinet server.
Build Extension	Create a reporting extension for SOA Systinet containing the report. For details, see Creating a Reporting Extension on page 52.

Data Explorer

The Data Explorer displays the sources and sets of data used by the particular report you are working with, as shown in Figure 4.

Figure 4. Data Explorer View



Data Explorer contains the following elements:

- **Data Sources**

  Open the context menu and select **New Data Source** to access a database.

  For details, see Adding a Data Source to a Report on page 38.

- **Data Sets**

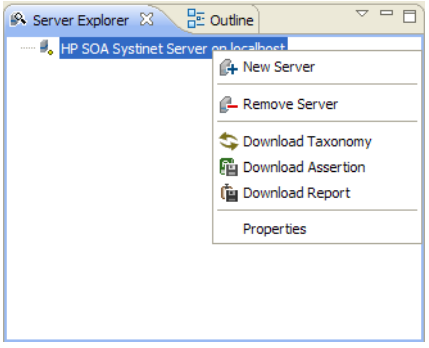
  Open the context menu and select **New Data Set** to define the data used by the report.

  For details, see Creating a Data Set on page 38.

Server Explorer

The Server Explorer displays the SOA Systinet servers connected to Workbench, as shown in Figure 5. The functionality is shared by all the Workbench editors.

Figure 5. Server Explorer View



Right-click a server in the Server Explorer to open the context menu described in Table 3.

Table 3. Server Explorer Context Menu Options

Option	Function
New Server	Add a server for downloading assertions and taxonomies (Assertion Editor, Taxonomy Editor, and Customization Editor).
Remove Server	Delete a server from the Server Explorer.
Download Taxonomy	Download a taxonomy from a platform server (Taxonomy Editor and Customization Editor).
Download Assertion	Download assertions from a platform server (Assertion Editor).
Download Report	Download reports from a reporting server (Report Editor).
Properties	View and edit the server name, URL, username, and password.

Use Cases

SOA Systinet uses report categories to associate reports with particular functionality as follows:

- **Platform – Dashboard Report**

  A dashboard report is designed to fit inside a portlet on the SOA Systinet dashboard, as described in Creating a Report for the Dashboard on page 22.

- **Platform – Tools Report**

  A tools report is intended for use with a reporting tool in SOA Systinet, as described in Creating a Report for the Reporting Tool on page 21.

- **Policy Manager – Artifact Compliance Report**

  Reports with the artifact compliance report category are available in the View menu of compliance report pages, as described in the "Document Reports" section of the *HP SOA Systinet User Guide*.

- **Policy Manager – Artifact Summary Report**

  Reports with the artifact summary report category are available in the View menu of compliance report pages, as described in the "Document Summary Reports" section of the *HP SOA Systinet User Guide*.

- **Policy Manager – Business Policy Compliance**

  Reports with the business policy summary category are available in the View menu of compliance report pages, as described in the "Business Policy Summary Reports" section of the *HP SOA Systinet User Guide*.

- **Policy Manager – Business Policy Summary**

  Reports with the business policy summary category are available in the View menu of compliance report pages, as described in the "Business Policy Reports" section of the *HP SOA Systinet User Guide*.

- **Policy Manager – Detail Report**

  Reports with the detail report category are available in the View menu of compliance report pages, as described in the "Document Reports" section of the *HP SOA Systinet User Guide*.

- **Policy Manager – Homepage Report**

  Selecting homepage report enables you to select the report as an area of interest to view on the main Policies tab in SOA Systinet. See the "Areas of Interest" section of the *HP SOA Systinet User Guide*.

Each use case describes the high-level steps required to implement them. The examples provided in Chapter 6, Use Case Examples can be used as a reference.

The main use cases are described in the following sections:

- Creating a Report for the Reporting Tool on page 21
- Creating a Report for the Dashboard on page 22
- Creating Reports for Policy Manager on page 23

Creating a Report for the Reporting Tool

The main use case for Report Editor is to create a new report definition designed for use with a reporting tool in SOA Systinet.

To create and use a reporting tool report definition:

1. Create a new report definition.

   For details, see Creating a Report Definition on page 37.

2. Add a data source for the report.

   For details, see Adding a Data Source to a Report on page 38.

3. Create a data set for the report,

   For details, see Creating a Data Set on page 38.

4. Design your report layout.

   For details, see the *BIRT Report Developer Guide*, which you can access from the main menu by selecting **Help→Help Contents**.

5 Categorize the report with **Platform - Tools** as the category.

For details, see Categorizing Reports on page 40.

6 Deploy the report to SOA Systinet.

For details, see Deploying a Report to SOA Systinet on page 51.

7 Create a reporting tool in SOA Systinet.

For details, see the "Creating a Reporting Tool" section of the *HP SOA Systinet User Guide*.

8 Execute the reporting tool in SOA Systinet.

For details, see the "Running a Reporting Tool" section of the *HP SOA Systinet User Guide*.

9 Optionally, create a scheduled task in SOA Systinet to use the new reporting tool.

For details, see the "Creating a Task" section of the *HP SOA Systinet User Guide*.

For a step-by-step walkthrough demonstrating this use case, see Example: Failure Impact Report for the Reporting Tool on page 59.

Creating a Report for the Dashboard

SOA Systinet contains functionality enabling you to display small reports on the **Dashboard**.

To create and use a dashboard report definition:

1 Create a new report definition.

For details, see Creating a Report Definition on page 37.

2 Add a data source for the report.

For details, see Adding a Data Source to a Report on page 38.

3 Create a data set.

For details, see Creating a Data Set on page 38.

4 Design your report layout.

For details, see the *BIRT Report Developer Guide*, which you can access from the main menu by selecting **Help→Help Contents**.

The dashboard portlet for a report has a width limitation of 211 pixels.

For an example of configuring your report layout to fit the portlet, see Example: Failure Impact Report Layout for the Dashboard on page 72.

5 Categorize the report with **Platform - Dashboard Report** as the category.

For details, see Categorizing Reports on page 40.

6 Deploy the report to SOA Systinet.

For details, see Deploying a Report to SOA Systinet on page 51.

7 Add the dashboard report to the SOA Systinet Dashboard.

For details, see the "Adding a Content Report" section of the *HP SOA Systinet User Guide*.

For a step-by-step walkthrough demonstrating this use-case, see Example: Failure Impact Report for the Dashboard on page 70.

Creating Reports for Policy Manager

Policy Manager supports the implementation of policy reports as alternative or additional views for policy pages in SOA Systinet.

Select a Policy Manager category to associate the report with a particular Policy Manager page. For details, see Use Cases on page 20.

2 Getting Started

SOA Systinet is distributed with a set of predefined reports and various ways to access them.

Report Editor provides a mechanism to customize these existing reports, and also enables you to create new report definitions from scratch.

This chapter describes the setup of Report Editor in the following sections:

Installing a Report Development Environment

This section describes a setup to develop and test reports on a single computer.

For supported platforms, see `readme.txt` in the installation folder.

A typical report development environment consists of the following:

- PC or notebook with at least 2GB RAM and at least 5GB free disk space
- Windows XP or Windows 2000
- Oracle eXpress Edition (XE)

  For deployment details, see Oracle XE Account Setup on page 26.

You do not need to install XE if you have dba-level access to deployed SOA Systinet environment using Oracle 10g.

- Java 1.5_9 or higher

  For deployment details, see Java Environment Setup on page 26.

- HP SOA Systinet deployed to JBoss

  For details, see *HP SOA Systinet Installation and Deployment Guide*.

You do not need to install SOA Systinet if you have access to a test environment.

- Oracle Developer

  HP Software recommends this tool for tuning SQL queries.

Oracle XE Account Setup

If you install Oracle XE, you must create a user account.

To create an Oracle XE account:

1 Open the XE web console and log on as the system user.

2 Create a new user, `platform`, with the password `changeit`, then grant the new user all available privileges.

Java Environment Setup

After installing Java, you must set an environment variable, JAVA_HOME.

To set the JAVA_HOME environment variable:

1 In Windows, click through **Control Panel→System→Advanced→Environment Variables**.

The Environment Variables dialog box opens.

2 In the User Variables section, click **New**.

3 Input Variable Name, `JAVA_HOME`, and set Variable Value to the installation folder for Java.

4 Click **OK** to exit each dialog box.

Installing Workbench

HP SOA Systinet Workbench is an Eclipse development platform distributed as a zip file, `hp-soa-systinet-workbench-3.10-win32.zip` or as a plugin for an existing Eclipse environment, `hp-soa-systinet-workbench-3.10-updates.zip`.

For supported platforms and known issues, see `readme.txt` alongside the archive.

To install HP SOA Systinet Workbench as a new Eclipse platform:

- Extract the archive to your required location, referred to in this document as `WB_HOME`.

The path must not be longer than 18 characters.

To install HP SOA Systinet Workbench to an existing Eclipse platform:

1 Install the following prerequisite plug-ins to your Eclipse development platform based on the required Workbench components:

- DTP SDK 1.5

  Required for Customization Editor and Report Editor.

  http://www.eclipse.org/datatools/downloads.php

- EMF SDO Runtime 2.3.0

Required for Customization Editor and Report Editor.

http://www.eclipse.org/modeling/emf/downloads/

- GEF Runtime 3.3

  Required for Customization Editor and Report Editor.

  http://archive.eclipse.org/tools/gef/downloads/drops/R-3.3-200706281000/index.php

- WTP 2.0

  Required for Customization Editor and Report Editor.

  http://www.eclipse.org/webtools/releases/2.0/

- XSD Runtime 2.3

  Required for Customization Editor and Report Editor

  http://www.eclipse.org/modeling/mdt/downloads/?project=xsd

- BIRT 2.2.0

  Required for Report Editor.

  http://download.eclipse.org/birt/downloads/build_list.php

2 In your current Eclipse SDK (3.3 or later), use the software updates feature to install HP SOA Systinet Workbench.

  Select **Help→Software Updates→Find and Install...**.

  The Install/Update dialog opens.

3 In the Install/Update dialog, select **Search for new features to install**, and click **Next**.

  The Install – Update Sites to Visit dialog opens.

4 In the Update Sites to Visit dialog, click **New Archived Site**.

The Select Local Archive Site dialog opens.

5 Locate and select `hp-soa-systinet-workbench-3.10-updates.zip`, and then click **Open**.

The Edit Local Site dialog opens.

6 In the Edit Local Site dialog, if required, rename the local archive name, and click **OK**.

7 In the Install – Update Sites to Visit dialog, select the new local archive, and then click **Finish**.

The Updates – Search Results dialog opens.

8 Select the modules from the archive that you want to install:

- **Workbench Extra 3.10**

  HP SOA Systinet Workbench splash screen.

- **Taxonomy Editor 3.10**

  Required for Customization Editor.

- **Customization Editor 3.10**
- **Assertion Editor 3.10**
- **Report Editor 3.10**
- **Common Plugin 3.10**

  Shared components used by the editors.

Click **Next**.

The Install – Feature License dialog opens.

9 In the Feature License dialog, select **I accept the terms in the license agreements**, and click **Next**.

The Install – Installation dialog opens.

10 In the Installation dialog, if required, change the installation location, and then click **Finish**.

11 If you install `Workbench Extra 3.10` make the following configuration changes:

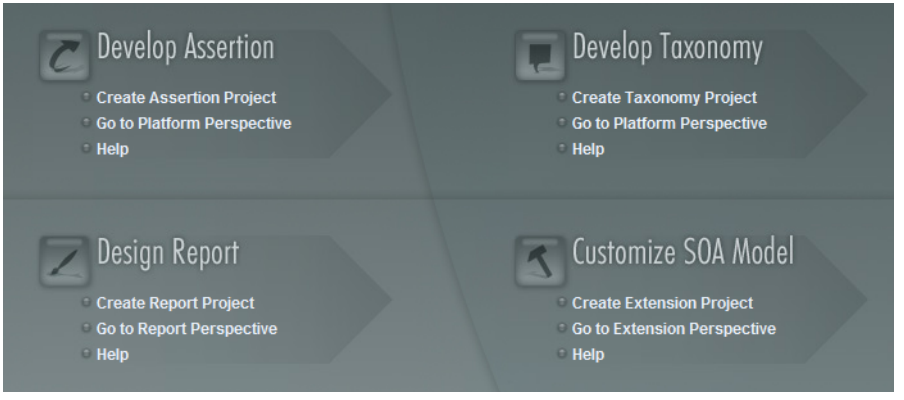
- Remove `-showsplash org.eclipse.platform` from `ECLIPSE_HOME/eclipse.ini`.
- Edit ECLIPSE_HOME/configuration/config.ini and make the following changes:
  - Set `osgi.splashPath=platform:/base/plugins/com.systinet.tools.workbench .`
  - Set `eclipse.product=com.systinet.tools.workbench.ide`

To start HP SOA Systinet Workbench:

- Execute **WB_HOME/systinet-workbench/start.exe**.

The first time you start Workbench, the welcome screen opens, as shown in Figure 6.

Figure 6. Workbench Welcome Screen



Select one of the options to open one of the editor tools, start a new editing project, or view the documentation set.

You can return to the welcome screen from any of the editor tools by selecting **Help→Welcome** from the menu options.

HP SOA Systinet Workbench requires Java SE Development Kit (JDK) 1.5.0 or higher. You must include the path to this version of the JDK in the *JAVA_HOME* environment variable.

HP SOA Systinet Workbench is memory-intensive. If you experience performance issues, HP recommends increasing the memory allocation.

To increase the memory allocation for HP SOA Systinet Workbench:

1. Open `WB_HOME/start.ini` for editing.
2. Set these new values:
   - `-Xms128m`
   - `-Xmx1024m`
3. Save your changes.
4. Restart Workbench.

SSL Configuration

By default, Workbench trusts all SOA Systinet server certificates. You may want Workbench to verify SOA Systinet certificates.

To verify SOA Systinet server certificates:

- Add the following options to `WB_HOME/start.ini`:

```
-Dcom.hp.systinet.security.ssl.verifyCert=true
-Djavax.net.ssl.trustStore=USER_TRUSTSTORE
```

```
-Djavax.net.ssl.trustStorePassword=TRUSTSTORE_PASS
-Djavax.net.ssl.trustStoreType=TRUSTSTORE_FORMAT
```

If SOA Systinet is configured for 2-way SSL, you must provide Workbench certificates to SOA Systinet.

To provide Workbench client certificates to SOA Systinet:

- Add the following options to `WB_HOME/start.ini`:

```
-Djavax.net.ssl.keyStore=USER_KEYSTORE
-Djavax.net.ssl.keyStorePassword=KEYSTORE_PASS
-Djavax.net.ssl.keyStoreType=KEYSTORE_FORMAT
```

Configuring a Reporting Project

To use Report Editor with SOA Systinet, you must create or import a project, and import any report definitions you want to modify.

These procedures are described in the following sections:

- Creating a Reporting Project on page 32
- Importing a Reporting Project on page 34
- Importing Report Definitions on page 34

Creating a Reporting Project

The Report Editor organizes your work into project folders in your workspace.

To create a reporting project:

1. Do one of the following:
   - In the Workbench Welcome page, select **Create Report Project**.
   - From the menu, select **File→New→Report Project**.

- Press **Alt+Shift+N**, and then press **R**. Expand **HP SOA Systinet**, select **Report Project**, and then click **Next**.

The New Report Project Wizard opens.

2 Input a name for your project and, if needed, change the workspace location, and then click **Next**.

3 Do one of the following:

- Select **Create a New Server**, and then click **Next**.

  Continue to Step 4.

- Select **Use an Existing Server**, select the server from the list and input its credentials, and then click **Next**.

  Continue to Step 5.

4 In the New Server dialog box, add the parameters you require, and then click **Next**.

For parameters descriptions, see New Report Project Wizard: New Server on page 76.

5 If needed, in the New Data Source dialog box, click **Manage Drivers** to add a JDBC driver.

For details, see Adding a JDBC Driver on page 44.

6 In the New Data Source dialog box, add the parameters you require.

For parameters descriptions, see New Report Project Wizard: Default Library Configuration - Create a new data source on page 77.

If your database server is running, click **Test Connection** to check if your settings are correct.

7 Click **Next** to set project options.

8 Do one of the following:

- To create a generalized reporting project, click **Finish** and exit the wizard.
- To create a project associated with a specific report category, click **Next** and continue this procedure.

The SOA Systinet server must be running to access report definitions.

9 Select a report category for your reporting project, and then click **Next**.

10 Select report definitions from the selected category to download to the Report Editor, and then click **Finish**.

Importing a Reporting Project

You can import a reporting project created elsewhere into your Workbench installation.

To import a reporting project:

1 Copy the project folder to your local files ystem.

2 From the menu, select **File→Import**.

The Import dialog box opens.

3 Expand **General** and select **Existing Projects into Workspace**.

4 Use **Browse**, select the project from your local file system, and then click **OK**.

5 Click **Finish** to import the reporting project.

Importing Report Definitions

The SOA Systinet SOA Systinet server contains all the report definitions for SOA Systinet. To modify existing reports, you must import their definitions from the SOA Systinet server.

To import SOA Systinet report definitions:

1 In the Project Explorer, open the context menu for the project, and do one of the following:

- Select **HP SOA Systinet→Download Report** to import from the current SOA Systinet server.
- Select **Import→Report**, and then select a reporting server.

The SOA Systinet server must be running.

2 Select the report category, and then click **Next**.

3 Select the report definitions you require, and then click **Finish**.

3 Designing Reports

This chapter describes the functionality of Report Editor in conjunction with SOA Systinet. The standard functionality of Report Editor is described in the *BIRT Report Developer Guide*, which you can access from the main menu by selecting **Help→Help Contents**.

This chapter contains the following sections:

Creating a report:

Creating a Report Definition

The first step in defining a report is to create the report definition.

To create a report definition:

1. Select **File→New→Report** from the menu.
2. Input a name and description for the report, and then click **Next**.
3. Select the project folder, and then click **Finish** to create the report definition.
4. Press **Ctrl+S** to save the report definition.

See Step 1 of Example: Failure Impact Report for the Reporting Tool on page 59 for an example of this procedure.

Adding a Data Source to a Report

By default, new reports contain the data source setup during project creation. If you need to use a different one, you must add it to the report.

To add a data source to a report:

1 Open the report definition editor view.

2 In the Data Explorer, open the Data Sources context menu and select **New Data Source**.

The New Data Source dialog box opens.

3 Select **JDBC Data Source** and then click **Next**.

4 If needed, in the New JDBC Data Source dialog box, click **Manage Drivers** to add a JDBC driver.

For details, see Adding a JDBC Driver on page 44.

5 In the New JDBC Data Source dialog box, add the parameters you require.

For parameters descriptions, see New Report Project Wizard: Default Library Configuration - Create a new data source on page 77.

If your database server is running, click **Test Connection** to check if your settings are correct.

6 Click **Finish** to add the data source to your report.

7 Press **Ctrl+S** to save your changes to the report definition.

Creating a Data Set

Each report is associated with one or more queries that each return a data set.

To create a data set:

1 From the Report Design perspective, open the report definition view for the report requiring the new data set.

If you need to use an alternative data source, see Adding a Data Source to a Report on page 38.

2 In the Data Explorer, open the Data Sets context menu, and select **New Data Set**.

The **New Data Set** dialog opens.

3 In the New Data Set dialog box, add the parameters you require.

For parameters descriptions, see New Data Set on page 75.

4 Click **Next** to open the **Query** dialog box.

5 Do one of the following:

- Copy a query from your development tool to the query editor.

  Skip to Step 8.

  For details about the database structure and writing queries, see Chapter 4, Exploring the Database.

- Create your query in Report Editor as described in Step 6 to Step 7.

6 Use **Schema**, **Filter**, and **Type**, and then click **Apply Filter** to focus on the data you require in Available Items.

Workbench limits the amount of data collected. To change the settings, see Changing the Data Collection Settings on page 44.

7 Browse Available Items to find the data items you require. Drag and drop items from Available Items to the query editor and construct your query.

For portability to different SOA Systinet installations using different schemas, remove the schema name from the variables in your query if your database schema does not require them.

8 Click **Finish** to add the data set to the report definition.

9 Press **Ctrl+S** to save your changes.

For an example of this procedure, see Example: Failure Impact Data Set on page 61.

If possible, use only one dataset per report definition. If you need more datasets in one report (for example, multiple properties in one table row), use JavaScript to build one table.

Designing a Report Layout

The *BIRT Report Developer Guide* contains all the information required to design the layout and content of your reports. Select **Help→Help Contents** to view the guides included with Workbench.

For an example, see Example: Failure Impact Report Layout for the Reporting Tool on page 64.

Categorizing Reports

The Report Editor enables you to categorize reports. Each category corresponds to a particular page or function in SOA Systinet.

To categorize a report:

1 In the **Project Explorer**, expand the report you want to categorize and double-click **General** to open a view of the report definition containing categorization.

2 The Categories section displays the current categories that apply to your report.

Click **Add** to open the **Add Category** dialog.

3 Select the category you require.

For details about categories, see Use Cases on page 20.

4 Click **OK** to confirm your categorization.

5 Press **Ctrl+S** to save your changes.

See Step 4 of Example: Failure Impact Report for the Reporting Tool on page 59 for an example of this procedure.

Modifying Existing Report Definitions

Report definitions that already exist in SOA Systinet can be modified.

To modify existing report definitions:

1 Import the report definition.

For details, see Importing Report Definitions on page 34.

2 To open the report definition, in the Project Explorer, double-click the report definition *rptdesign*.

3 Make your modifications, as described in Creating a Report Definition on page 37.

4 Press **Ctrl+S** to save your changes.

5 Redeploy the report definition to SOA Systinet.

For details, see Deploying a Report to SOA Systinet on page 51.

4 Exploring the Database

The functionality of Report Editor relies on interaction with the database.

This chapter describes how to examine and setup the database in the following sections:

SDM to Database Mapping Tool

Artifacts in SOA Systinet are stored in the form of XML documents. Their structure is defined by the SOA Definition Model (SDM). Artifacts are serialized into a database over a standard serialization layer. The serialization of data may differ from the norm, based on customer specific extensions or modifications.

The sdm2dbmap tool is a mapping tool that generates a report containing the mapping between your SDM and database tables.

To generate the report, execute the following command:

SOA_HOME/lib/sdm/bin/sdm2dbmap

The mapping report is output to the following file:

```
SOA_HOME/lib/sdm/build/sdm2dbmap.html
```

The output consists of the following parts:

- A top level 1:1 mapping between SDM artifacts and DB tables. Each artifact listed, maps directly to one table.
- A list of artifacts. Each artifact in the report maps each SDM property to a specific column in the table. There are also associated tables and foreign keys, joined using the primary key of the artifact table.
- A report documenting the DB schema for all database tables coming from the SDM. Tables with names ending in _Rev are used to store older revisions.

Adding a JDBC Driver

To connect to a database you must specify a JDBC driver.

To add a JDBC driver:

1 In the Database Connections dialog box, click **Manage Drivers**.

The Manage JDBC Drivers dialog box opens.

2 Click **Add** to browse your filesystem for a driver.

HP Software recommends using the same JDBC driver used during SOA Systinet installation.

3 Select the driver, and then click **Open** to add it to the list of drivers in the **Manage JDBC Drivers** dialog box.

Changing the Data Collection Settings

By default, the editor limits the amount of data collected to 20 schemas and 100 tables in the Query dialog box.

To change the data collection settings:

1 From the menu, select **Windows→Preferences**.

2 Expand **Report Design→Data Set Editor**, and then select **JDBC Data Set**.

3 Change the number of schemas and tables as required, and then click **OK**.

Optimizing the Database

If you frequently use particular fields for joins in your queries, create an index for that field to improve performance.

To create an Oracle index:

1 Log on to the sqlplus or isqlplus console as the system user.

2 Enter the following command:

create index ix_docrel_optimize2 on ry_docRel(sourceId, rtype);

As a minimum optimization, HP Software recommends the following:

- **create index ix_res_optimize1 on ry_resource(fk_artifactType, m_deleted, id, m_path);**
- **create index ix_docrel_optimize2 on ry_docRel(sourceId, rtype);**

Writing Report Queries

Although it is possible to use the query editor in Workbench, it is often more convenient to use a specialized tool to develop queries.

This section describes some typical methods and provides examples to help you design your queries.

These examples were created using Oracle Developer for use with an Oracle Database. Changes may be required to use these queries with other database types.

This section contains the following methods and examples:

Joining Tables in Queries

To join tables in your queries, add the following to the WHERE clause of your SELECT statement:

- Key or foreign key condition
- Discriminator expression
- Check the deleted flag (0 means false)

Example:

```
WHERE ryga_bsnService.id = rygt_ctgryPty.fk_resource_id
    AND rygt_ctgryPty.discriminator = 'bsnPolicy_iBag'
    AND bsnRes.m_deleted = '0'
```

Example: Business Service SELECT

This example query returns details for business services that are marked as ready for consumption:

```
SELECT
    bsn.id bsn_id,
    bsn.name_val bsn_name,
    bsnRes.m_path bsn_path,
    bsn.serviceVersion_val bsn_version,
    bsn.description_val bsn_description,
    bsn.productionStage_name bsn_productionStage

FROM ryga_bsnService bsn
    INNER JOIN ry_resource bsnRes ON
    (
        bsnRes.id = bsn.id
        AND bsnRes.m_deleted = '0'
    )
```

```
WHERE bsn.readyForConsumption_val ='1'
ORDER BY lower(bsn.name_val)
```

Table ry_bsnService contains the main fields for business services.

Table >ry_resource contains general information about every artifact (for all artifact types).

The inner join to ry_resource in the FROM clause excludes deleted artifacts.

Example: Contract SELECT

This example query returns details for contracts that have been accepted:

```
SELECT
    contract.id ctrct_id,
    contract.name_val ctrct_name,
    contract.description_val ctrct_description,
    contract.contractstate_name ctrct_state,
    providerContractRel.targetpath ctrct_targetpath

FROM ry_contract contract
    INNER JOIN ry_docrel providerContractRel ON
    (
        providerContractRel.rtype = '{http://systinet.com/2005/05/soa/model/property}providerContractor'
        AND providerContractRel.obsolete='0'
        AND providerContractRel.deleted='0'
        AND providerContractRel.targetDeleted='0'
        AND providerContractRel.sourceId = contract.id
    )
    INNER JOIN ry_resource contractRes ON
    (
        contractRes.id =contract.id
        AND contractRes.m_deleted='0'
    )

WHERE contract.contractState_val = 'uddi:systinet.com:soa:model:taxonomies:contractAgreementStates:accepted'
```

Table ry_contract contains the main fields for contracts.

Table ry_docrel contains artifact relationship details.

The inner join to ry_docrel in the FROM clause excludes contracts where the provider-contract relationship is obsolete, deleted, or where the provider does not exist.

Example: Combined Business Service and Contract SELECT

Example: Business Service SELECT on page 46 and Example: Contract SELECT on page 47 can be combined to list the names of the contracts and the business service to which they apply.

```
SELECT
    contract.id ctrct_id,
    contract.name_val ctrct_name,
    bsn.id bsn_id,
    bsn.name_val bsn_name

FROM ryga_bsnService bsn
    INNER JOIN ry_resource bsnRes ON
    (
        bsnRes.id = bsn.id
        AND bsnRes.m_deleted = '0'
    ),
    ry_contract contract
    INNER JOIN ry_docrel providerContractRel ON
    (
        providerContractRel.rtype = '{http://systinet.com/2005/05/soa/model/property}providerContractor'
        AND providerContractRel.obsolete='0'
        AND providerContractRel.deleted='0'
        AND providerContractRel.targetDeleted='0'
        AND providerContractRel.sourceId = contract.id
    )
    INNER JOIN ry_resource contractRes ON
    (
        contractRes.id =contract.id
        AND contractRes.m_deleted='0'
    )

WHERE bsn.readyForConsumption_val ='1'
   AND  contract.contractState_val = 'uddi:systinet.com:soa:model:taxonomies:contractAgreementStates:accepted'

    AND providerContractRel.targetId = bsn.id
ORDER BY lower(bsn.name_val)
```

By merging the queries you create an n:1 relationship between contracts and business services.

Example: Accepted Contract Count for a Business Service

This example returns business service details and the number of accepted contracts for each service:

```
SELECT
    bsn.id bsn_id,
    bsn.name_val bsn_name,
    bsnRes.m_path bsn_path,
    bsn.serviceVersion_val bsn_version,
    bsn.description_val bsn_description,
    bsn.productionStage_name bsn_productionStage,

    ( -- Contracts --
    SELECT count (0)
    FROM ry_contract contract
        INNER JOIN ry_docrel providerContractRel ON
        (
          providerContractRel.rtype = '{http://systinet.com/2005/05/soa/model/property}providerContractor'

            AND providerContractRel.obsolete='0'
            AND providerContractRel.deleted='0'
            AND providerContractRel.targetDeleted='0'
            AND providerContractRel.sourceId = contract.id
        )
        INNER JOIN ry_resource contractRes ON
        (
            contractRes.id =contract.id
            AND contractRes.m_deleted='0'
        )
    WHERE  contract.contractState_val =
'uddi:systinet.com:soa:model:taxonomies:contractAgreementStates:accepted'
        AND providerContractRel.targetId = bsn.id
    ) ctrct_count

    FROM ryga_bsnService bsn
        INNER JOIN ry_resource bsnRes ON
        (
            bsnRes.id = bsn.id
            AND bsnRes.m_deleted = '0'
        )

WHERE bsn.readyForConsumption_val ='1'
ORDER BY lower(bsn.name_val)
```

5 Deploying Reports

When your report is ready, you must deploy it to the SOA Systinet server.

Report Editor offers two methods:

- Deploy a report directly to the SOA Systinet server.

  For details, see Deploying a Report to SOA Systinet on page 51.

- Deploy a set of reports as an extension to the SOA Systinet server.

  This process consists of the following steps:

  1 Creating a Reporting Extension on page 52

  2 Applying Extensions on page 53

  3 Redeploying the EAR File on page 56

After deployment, you can access your report in various ways.

For details, see Accessing Reports on page 57.

You can remove custom reports from the SOA Systinet server.

For details, see Removing Report Definitions from SOA Systinet on page 58.

Deploying a Report to SOA Systinet

The SOA Systinet plug-ins for Report Editor enable you to deploy your report directly to the SOA Systinet server.

To deploy a report to SOA Systinet:

1 Make sure that the SOA Systinet server is running.

2 In the Project Explorer or Navigator view, open the context menu for the report design you want to deploy, and select **HP SOA Systinet→Publish Report**.

You will be asked if you want to publish the default library. If you do not want to be asked again, you can set this preference in the Preference menu of Report Editor.

Expand **Window→Preferences→HP SOA Systinet→Report Editor** and select whether you want Assertion Editor to prompt you every time you publish, always publish the default library, or never publish the default library.

Creating a Reporting Extension

You can also deploy a set of reports as an extension.

To create a reporting extension:

1 In the Project Explorer, open the context menu for the project you want to deploy and select **HP SOA Systinet→Build Extension**.

The Build Extension Wizard opens.

2 In the Build Extension Wizard, input the parameters you require.

For parameter descriptions, see Build Extension Wizard on page 74.

3 Click **Finish** to create the extension.

Deploy the extension to the SOA Systinet server using the Setup Tool.

For details, see Applying Extensions on page 53.

Applying Extensions

You can extend SOA Systinet by adding libraries or JSPs to the deployed EAR files, by modifying the data model, by configuring the appearance of the UI, and by importing prepackaged data.

Extensions to SOA Systinet come from the following sources:

- **Customization Editor**

  Typical extensions created by Customization Editor contain modifications to the data model and artifact appearance, and possibly data required by the customization (taxonomies). They may also contain new web components, which may include custom JSP and Java code.

- **Assertion Editor, Report Editor, and Taxonomy Editor**

  These extensions contain assertion, reporting, and taxonomy data only. They do not involve changes to the data model.

The Setup Tool opens the EAR files, applies the extensions, and then repacks the EAR files.

Apply extensions according to one of the following scenarios:

- Single-Step Scenario on page 53

  The Setup Tool performs all the processes involved in applying extensions, including any database alterations, as a single step.

- Decoupled DB Scenario on page 55

  Database SQL scripts are run manually. The Setup Tool performs the other processes as individual steps that are executable on demand. This scenario is useful in organizations where the user applying extensions does not have the right to alter the database, which is done by a database administrator.

Single-Step Scenario

Follow this scenario if you have permission to alter the database used for SOA Systinet.

To apply extensions to SOA Systinet in a single step:

1 Make sure that all extensions are in the following directory:

```
SOA_HOME/extensions
```

The Setup Tool automatically applies all extensions in that directory.

If you are applying extensions to another server, substitute the relevant home directory for `SOA_HOME`.

2 Stop the server.

3 Start the Setup Tool by executing the following command:

SOA_HOME/bin/setup.bat(sh)

4 Select the **Apply Extensions** scenario, and click **Next**.

The Setup Tool automatically validates the step by connecting to the server, copying the extensions, and merging the SDM configuration.

If you extension does not contain data model changes, select **Apply Extensions Don't Touch DB**.

5 Click **Next** for each of the validation steps and the setup execution.

This process takes some time.

6 Click **Finish** to end the process.

7 Deploy the EAR file:

- **JBoss**

  The Setup Tool deploys the EAR file automatically.

  If you need to deploy the EAR file to JBoss manually, see Redeploying the EAR File on page 56.

- **Other Application Servers**

  You must deploy the EAR file manually.

  For application server-specific details, see "Deploying the EAR File" in the *HP SOA Systinet Installation and Deployment Guide*.

8 Restart the server.

The Setup Tool normally applies ALTER scripts if database changes are required for an extension. If the ALTER script cannot be used, then a DROP and CREATE process may be used instead. In these cases, you must recreate indices on the database.

`SOA_HOME/log/setup.log` contains the following line in these cases:

```
Could not apply alteration scripts, application will continue with slower DB drop/create/restore scenario. ... .
```

Decoupled DB Scenario

Follow this scenario if the user who applies extensions does not have permission to modify the database.

To apply extensions and modify the database separately:

1 Make sure that all extensions are in the following directory:

```
SOA_HOME/extensions
```

The Setup Tool automatically applies all extensions in that directory.

2 Stop the server.

3 Start the Setup Tool by executing the following command:

SOA_HOME/bin/setup -a.

4 Select the **Apply Extensions** scenario, and click **Next**.

5 Click **Next**, to execute the extension application, and exit the Setup Tool.

6 Provide the scripts from `SOA_HOME/sql` to the database administrator.

The database administrator can use `all.sql` to execute the scripts that drop and recreate the database schema.

7 Execute the Setup Tool in command-line mode to finish the extension application:

SOA_HOME/bin/setup -c

8 Redeploy the EAR file:

- **JBoss**

  The Setup Tool deploys the EAR file automatically.

  If you need to deploy the EAR file to JBoss manually, see Redeploying the EAR File on page 56.

- **Other Application Servers**

  You must deploy the EAR file manually.

  For application server-specific details, see "Deploying the EAR File" in the *HP SOA Systinet Installation and Deployment Guide*.

Redeploying the EAR File

After using the Setup Tool to apply extensions or updates, you must redeploy the EAR file to the application server. For JBoss, you can do this using the Setup Tool.

For other application servers, follow the EAR deployment procedures described in the "Deploying the EAR File" in the *HP SOA Systinet Installation and Deployment Guide*.

To redeploy the EAR file to JBoss:

1. Stop the application server.
2. Start the Setup Tool by executing the following command:

   SOA_HOME/bin/setup.bat(sh).
3. Select the **Advanced** scenario, and click **Next**.
4. Scroll down, select **Deployment**, and then click **Next**.

   When the Setup Tool validates the existence of the JBoss Deployment folder, click **Next**.
5. Click **Finish** to close the Setup Tool.
6. Restart the application server.

Accessing Reports

Use Cases on page 20 describes how SOA Systinet uses specific functionality in the UI for each report category.

It is also possible to directly access the reports in the SOA Systinet server and obtain alternative outputs.

In your browser, the following URLs use the Service Portfolio report as an example:

- `https://hostname:port/reporting/reports/`
- `https://hostname:port/reporting/reports/service_portfolio?alt=text/plain`
- `https://hostname:port/reporting/reports/service_portfolio/documents/`
- `https://hostname:port/reporting/reports/service_portfolio/documents/1/?alt=text/plain`

The last URL is an instance of the report. The page source contains alternate URLs. Change the URL to return a comma separated output of the report as follows:

- `https://hostname:port/reporting/reports/service_portfolio/documents/1/content?alt=text/csv`

Removing Report Definitions from SOA Systinet

You can remove custom report definitions from SOA Systinet.

To remove a single report, use the context menu option.

To remove custom reporting extensions:

1. Remove any custom report extensions from `SOA_HOME/extensions`.
2. Execute the Setup Tool using the apply extensions scenario as described, in "Setup Tool" in the *HP SOA Systinet Administrator Guide*.

6 Use Case Examples

This chapter describes the following use cases:

- Example: Failure Impact Report for the Reporting Tool on page 59

  Explains how to create a failure impact report for use with the reporting tool in SOA Systinet.

- Example: Failure Impact Report for the Dashboard on page 70

  Explains how to create a failure impact report for the SOA Systinet Dashboard.

Each example is based on use cases described in Use Cases on page 20.

Example: Failure Impact Report for the Reporting Tool

To demonstrate the reporting tool use case described in Creating a Report for the Reporting Tool on page 21, follow this example:

To create a Failure Impact Report and execute it with a Reporting Tool:

1 Create the failure impact report definition.

Follow the procedure described in Creating a Report Definition on page 37 with the following inputs:

Parameter	Definition
Name	Failure Impact.
Description	Summary of Business Service Failure Impact Status.

2 Create the Failure Impact data set.

For details, see Example: Failure Impact Data Set on page 61.

3 Create the Failure Impact Report layout.

For details, see Example: Failure Impact Report Layout for the Reporting Tool on page 64.

4 The Failure Impact Report must be categorized for use with an SOA Systinet Reporting Tool.

Follow the procedure described in Categorizing Reports on page 40, and select category **Platform - Tools Report**.

5 Deploy the Failure Impact Report to SOA Systinet.

For details, see Deploying a Report to SOA Systinet on page 51.

6 In SOA Systinet, create a Failure Impact Reporting Tool.

In the Publish Reporting Tool page, input the following parameters:

Parameter	**Definition**
Name	Failure Impact Tool.
Description	Overview of the Failure Impact Status of Business Services.
Report Definition	Select **Failure Impact** from the drop-down list.

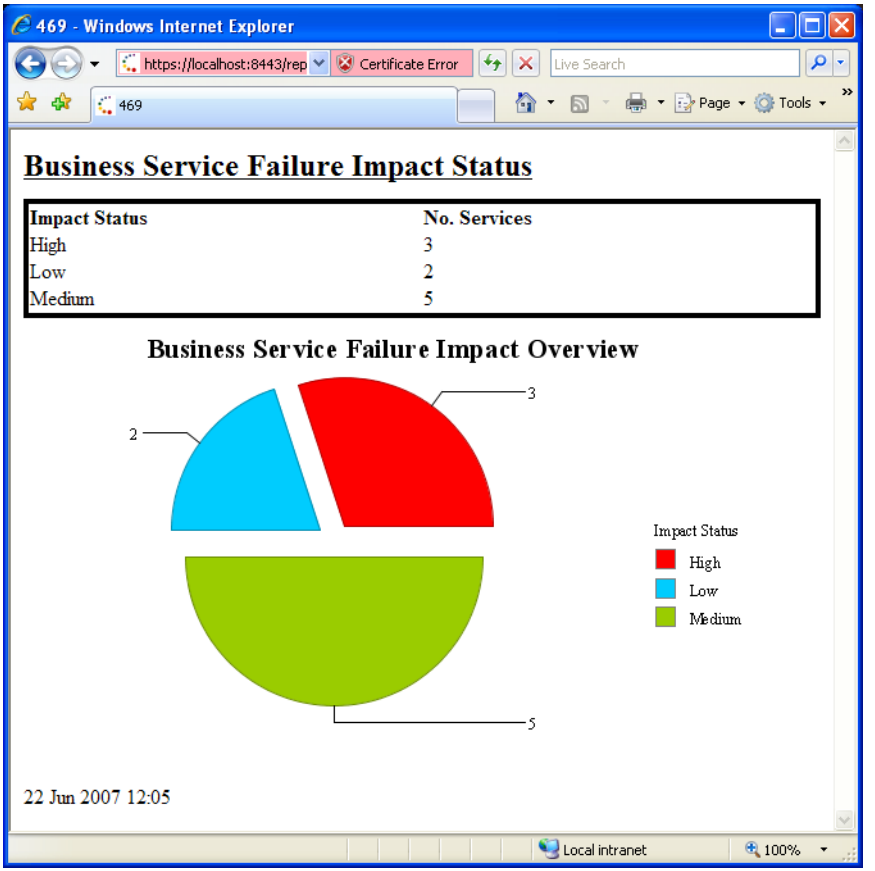
For details, see the "Creating a Reporting Tool" section of the *HP SOA Systinet User Guide*.

7 In SOA Systinet, execute the Failure Impact Reporting Tool.

For details, see the "Running a Reporting Tool" section of the *HP SOA Systinet User Guide*.

8 In SOA Systinet, access the Failure Impact Report.

In the SOA Systinet Tools tab Report Launcher portlet, click **Failure Impact Tool** to view the HTML version of the report, or one of the other format links to view the report in that format.



For details, see the "Report Launcher Portlet" section of the *HP SOA Systinet User Guide*.

Example: Failure Impact Data Set

To demonstrate the reporting tool use case described in Creating a Data Set on page 38, follow this example.

To create the Failure Impact data set:

1 Follow the procedure described in Creating a Data Set on page 38 with the following inputs.

2 In the New Data Set dialog box, input the following parameters:

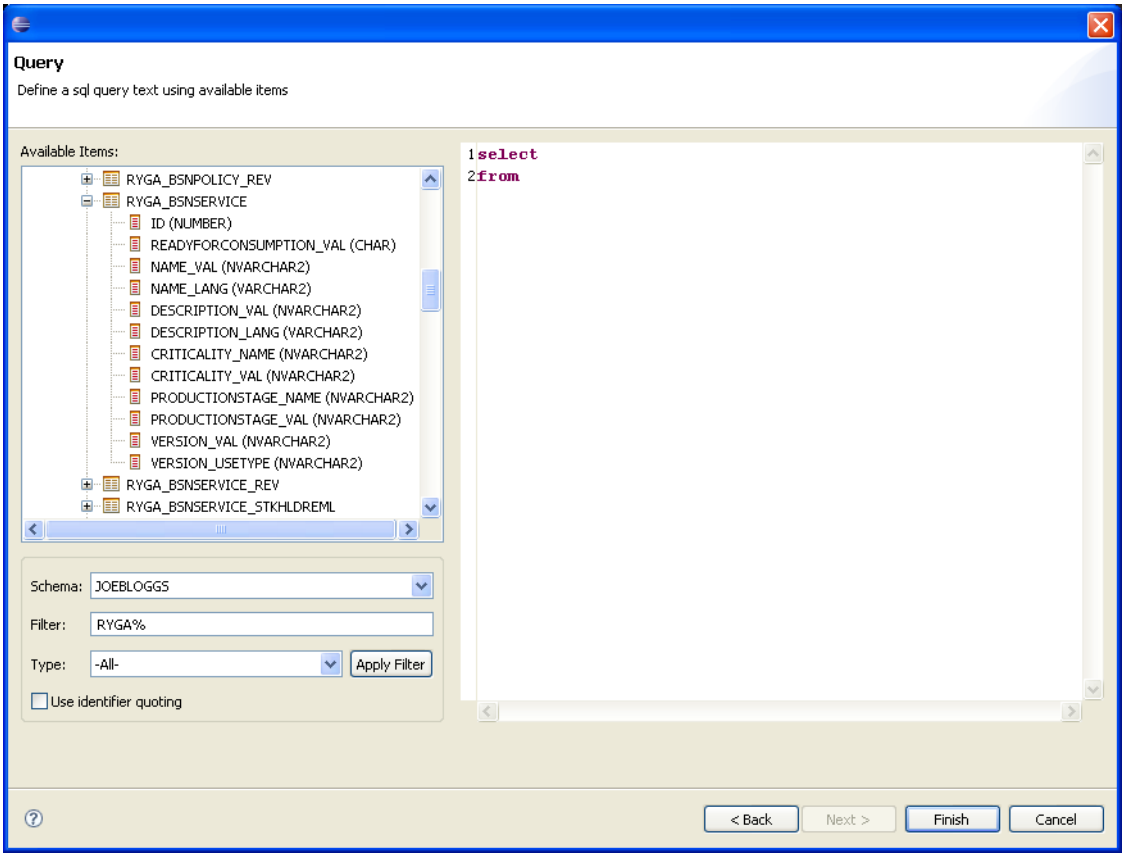
Parameter	Definition
Data Set Name	Failure Impact
Data Source	Database
Data Set Type	SQL Select Query

3 Use the filter to only display your schema and tables that contains the data that you want to use.

All standard tables in SOA Systinet begin with RYGA.

4 Use Available Items to explore the data and identify the tables and columns you need for your query.

For this report all the information required is in table **RYGA_BSNSERVICE**.

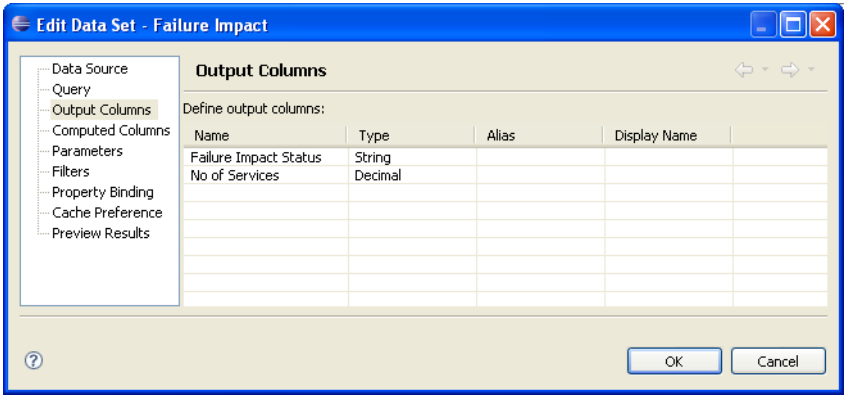


5 Construct your query by typing in the query input area and dragging and dropping items from **Available Items**.

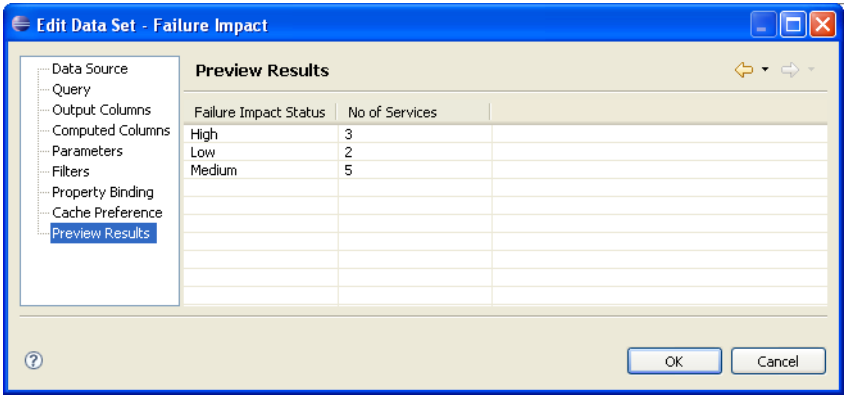
For this report the query is:

```
select RYGA_BSNSERVICE.CRITICALITY_NAME as "Failure Impact Status",
           count(RYGA_BSNSERVICE.CRITICALITY_NAME) as "No of Services"
 from RYGA_BSNSERVICE
 group by RYGA_BSNSERVICE.CRITICALITY_NAME
```

6 Click **Finish** to complete your query and view the Output Columns for the data set.



7 Click **Preview Results** to view the actual data in your database.



8 Click **OK** and press **Ctrl+S** to save the data set as part of the report.

Example: Failure Impact Report Layout for the Reporting Tool

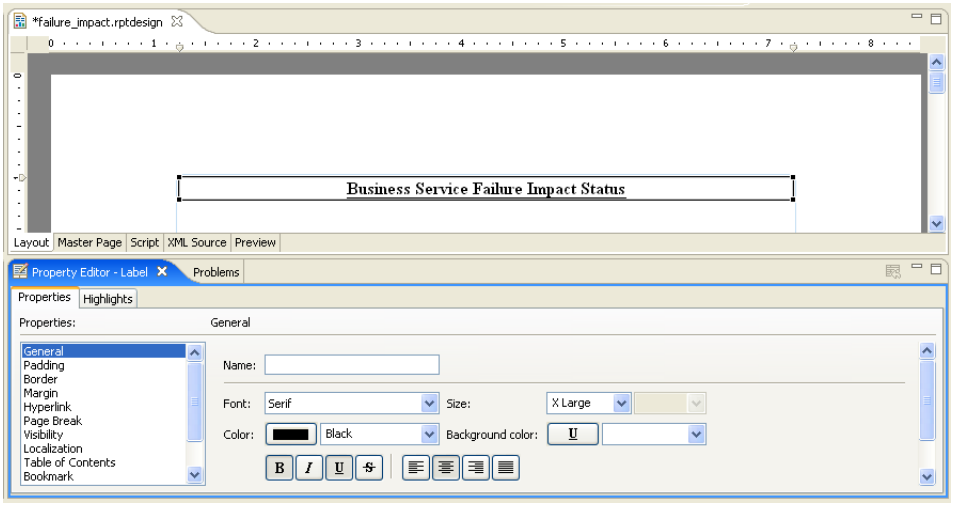
To demonstrate the reporting tool use case referred to in Designing a Report Layout on page 40, follow this example.

To Create the Layout of the Failure impact Report:

1 In the failure_impact.rptdesign editor view, right-click and select **Insert→Label**.

2 Type the report heading `Business Service Failure Impact Status`.

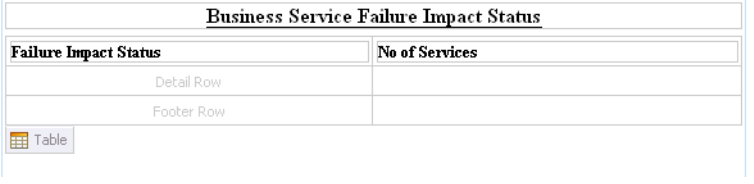
3 Use the Property Editor view to change the format of the heading.



4 In a blank area of the failure_impact.rptdesign editor view, right-click and select **Insert→Table**.

5 Input the table parameters and select the data set. In this case, 2 columns and the **Failure Impact** data set.

6 Select the left column header row, right-click and select **Insert→Text**. Input column header `Failure Impact Status`, and then click **OK**. Repeat for the right column header, `No of Services`.

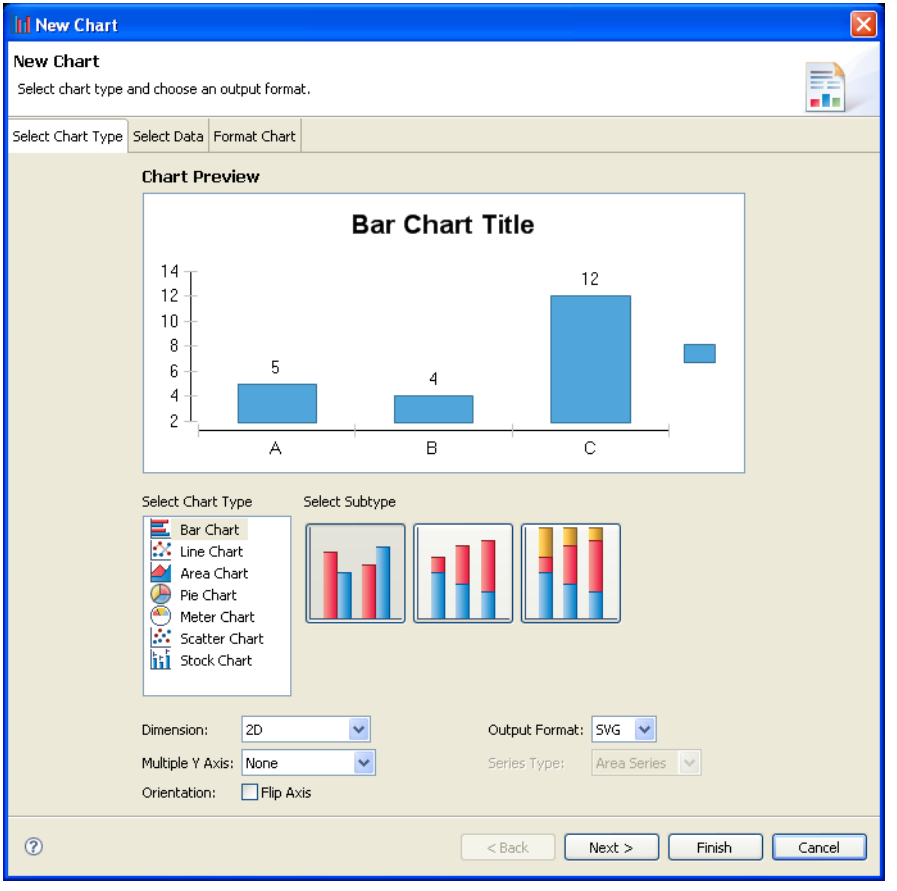


7 In the Data Explorer, expand **Data Sets→Failure Impact**.

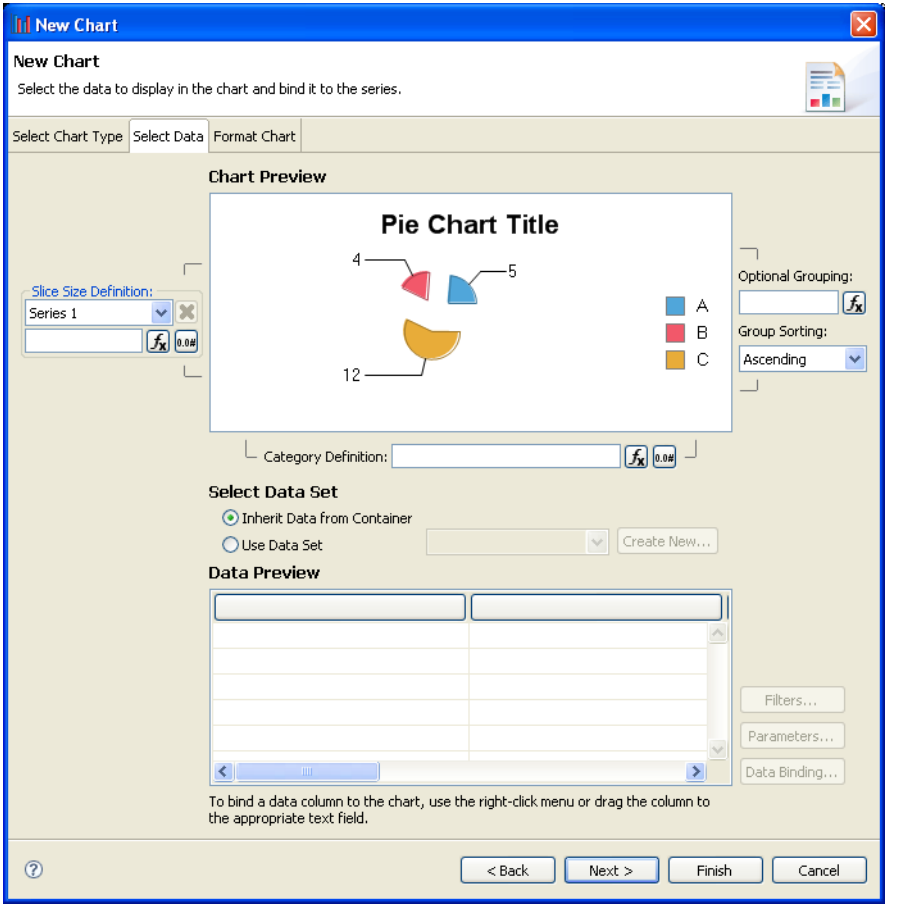
8 Drag **Failure Impact Status** from the Data Explorer to the left detail cell. Repeat for **No of Services** and the right detail cell.

Business Service Failure Impact Status	
Failure Impact Status	**No of Services**
[Failure Impact S...]	[No of Services]
Footer Row	

9 In a blank area of the failure_impact.rptdesign view, right-click and select **Insert→Chart** to open the New Chart dialog box.



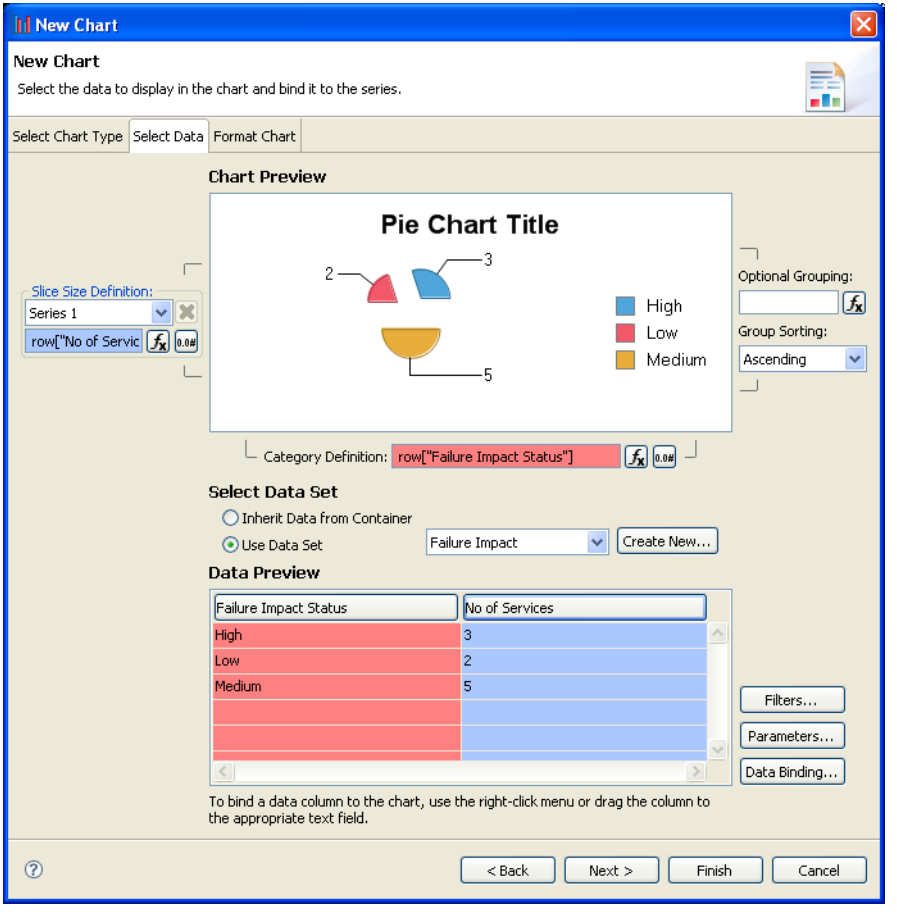
10 This report requires a pie chart, select **Pie Chart**, and then click **Next**.



11 Under Select Data Set, select **Use Data Set** and **Failure Impact** from the drop-down list.

12 Drag **Failure Impact Status** from **Data Preview** to the **Category Definition** input.

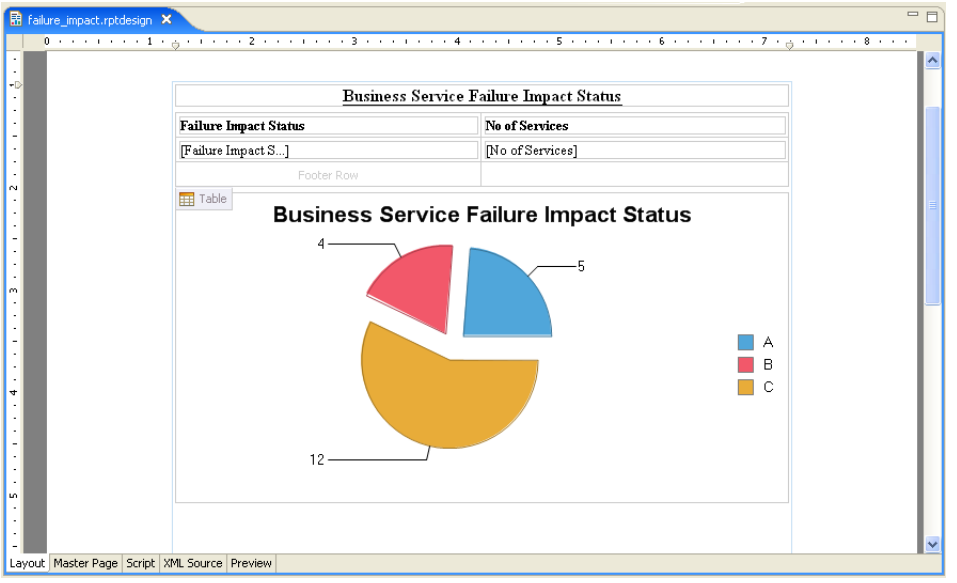
13 Drag **No of Services** from **Data Preview** to the **Slice Size Definition** input.



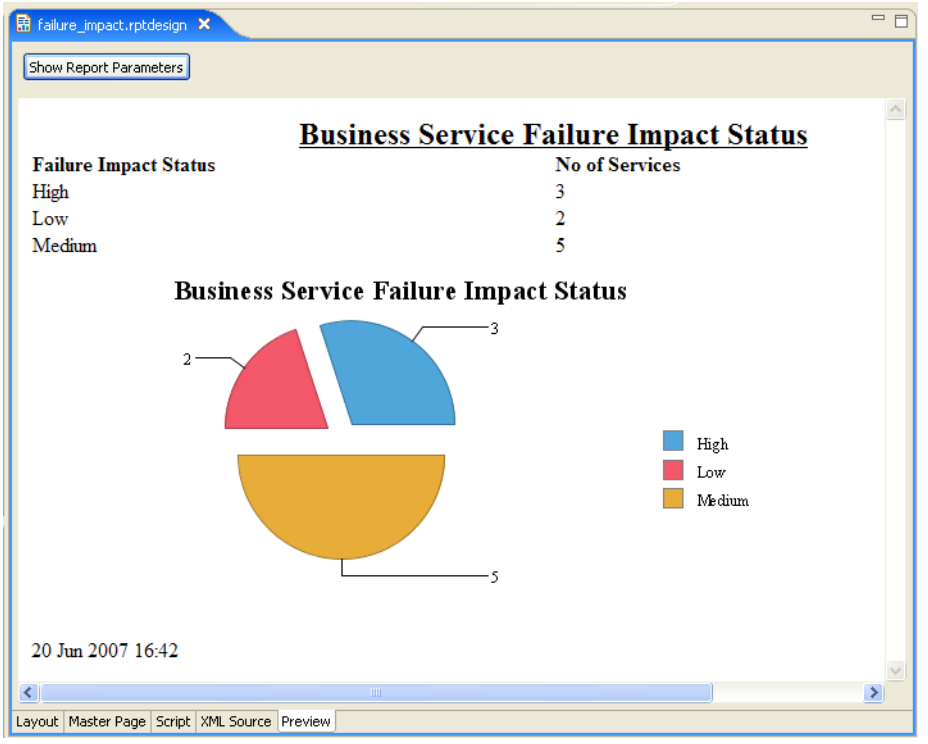
14 Click **Next** to format the chart appearance.

15 Select **Chart Area**, amend **Chart Title** to `Business Service Failure Impact Status`, and then click **Finish** to add the chart to the report definition.

16 Drag the chart area outline to the size you require, and then press **Ctrl+S** to save the report definition.



17 Select the **Preview** tab to view the report with data from your data source.



Example: Failure Impact Report for the Dashboard

SOA Systinet enables you to add small reports to the Dashboard. These reports have a limited width so some layout configuration is required.

This example uses the data set and table used in Example: Failure Impact Report for the Reporting Tool on page 59.

To create the failure impact report for the SOA Systinet Dashboard:

1 Create the failure impact dashboard report definition.

Follow the procedure described in Creating a Report Definition on page 37, with the following inputs:

Parameter	Definition
Name	Failure Impact Dashboard.
Description	Graph of Business Service Failure Impact Status.

2 Do one of the following:

- Create the failure impact data set.

  For details, see Example: Failure Impact Data Set on page 61.

- Copy the data set from the tools Failure Impact Report to the dashboard Failure Impact Report.

  For details, see Example: Copying the Failure Impact Data Set on page 72.

3 Create the Failure Impact Dashboard Report layout.

For details, see Example: Failure Impact Report Layout for the Dashboard on page 72.

4 The Failure Impact Dashboard Report must be categorized for use with the SOA Systinet Dashboard.

Follow the procedure described in Categorizing Reports on page 40, and select category **Platform - Dashboard Report**.

5 Deploy the Failure Impact Dashboard Report to SOA Systinet.

For details, see Deploying a Report to SOA Systinet on page 51.

6 Add the Failure Impact Dashboard Report to the SOA Systinet Dashboard.

For details, see the "Adding a Content Report" section of the *HP SOA Systinet User Guide*.

Select **Failure Impact Dashboard** from the list.

Example: Copying the Failure Impact Data Set

This example forms part of the procedure for the use case Example: Failure Impact Report for the Dashboard on page 70.

To copy the Failure Impact data set:

1. Double-click the tools Failure Impact Report design to open and select it.
2. In the Data Explorer, expand **Data Sets**.
3. Right-click **Failure Impact** and select **Copy**.
4. Click the **FailureImpactDashboard.rptdesign** editor to select it.
5. In the Data Explorer, right-click **Data Sets**, and select **Paste**.
6. Press **Ctrl+S** to save your changes.

Example: Failure Impact Report Layout for the Dashboard

This example forms part of the procedure for the use case Example: Failure Impact Report for the Dashboard on page 70.

To create the layout of the Failure Impact Dashboard Report:

1. In the **failure_impact_dashboard.rptdesign** view, select the **Master Page** tab.
2. In the Property Editor - Master Page view, select **Advanced**.
3. Scroll down the Property table until Type is visible.
4. Click **Type** to activate the drop-down value list, and then select **Custom**.
5. In the Property Editor - Master Page, select **General** to view the page size.
6. Change the **Header** and **Footer** to 5 points, the **Width** to 2 inches, and the **Height** to 3 inches.

The maximum width of a Dashboard Report is 211 pixels. There is no maximum height. Report Editor has problems dealing with pixel sizing so ins and points are used instead for this example.

7 In the Property Editor - Master Page, select **Margins** to view the page margins.

8 Set all the margins to 5 points.

9 In the failure_impact_dashboard.rptdesign editor view, select the **Layout** tab.

10 Do one of the following:

- Follow the procedure described in Example: Failure Impact Report Layout for the Reporting Tool on page 64 from Step 4 to Step 8 to add the table to your dashboard report.
- Copy the table from **failure_impact.rptdesign** and paste it to **failure_impact_dashboard.rptdesign**.

11 Press **Ctrl+S** to save the report definition.

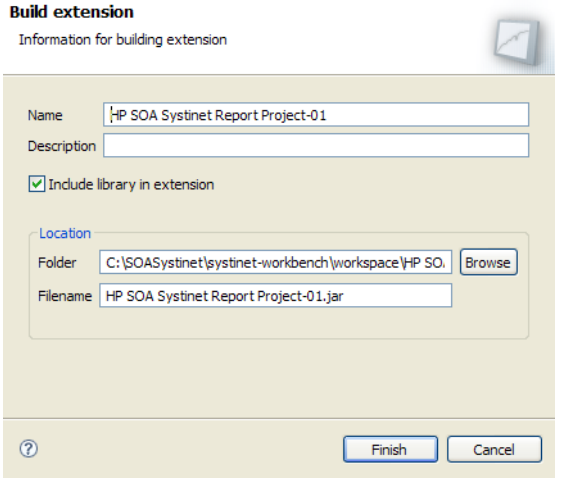
A Dialog Boxes

Each Report Editor input dialog is described in the following sections:

- Build Extension Wizard on page 74. Creating a report extension.
- New Data Set on page 75 Creating a new data set.
- New Report Project Wizard on page 76 Creating a new report project.

Build Extension Wizard

Enter general parameters for the report extension.

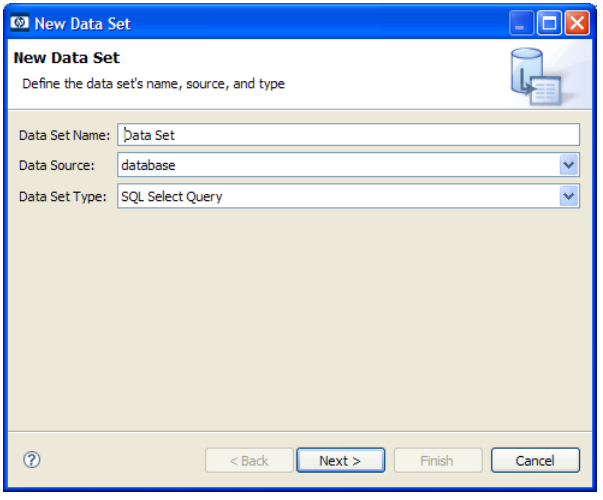


Option	Definition
Name	The name of the reporting extension.

Option	Definition
Description	A description for the reporting extension.
Include library in extension	Select this check-box if you want to include the default library in the extension.
Folder	The location of the reporting extensions folder.
Filename	The filename of the reporting extension.

New Data Set

Enter a name, then enter a source and type for the new data set.



Parameter	Definition
Data Set Name	Input a name for the data set.
Data Source	Select the data source from the drop-down list.

Parameter	Definition
Data Set Type	Choose one of the options from the drop-down list: • SQL Select Query • SQL Stored Procedure Query

New Report Project Wizard

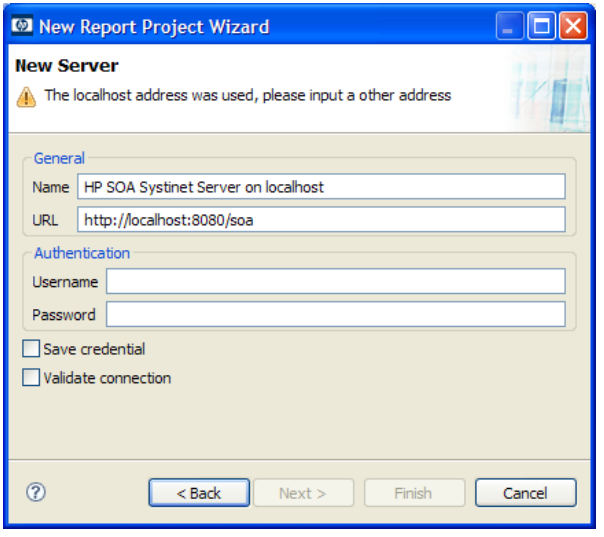
The New Report Project Wizard consists of the following stages depending on the options you select:

1 New Report Project Wizard: New Server on page 76

2 New Report Project Wizard: Default Library Configuration - Create a new data source on page 77

New Report Project Wizard: New Server

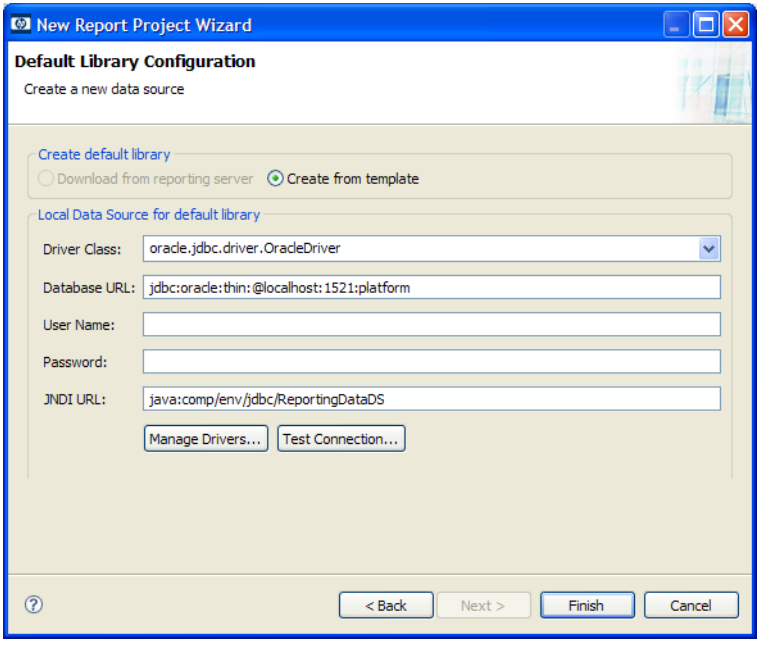
Create a new server for the project and optionally specify your authentication credentials.



Parameter	Definition
Name	The name of the server to display in the report editor.
URL	The location of the SOA Systinet server.
Username and Password	Credentials for access to the SOA Systinet server.
Save credentials	Select to store the input credentials.
Validate connection	Select to validate the server connection. The SOA Systinet server must be running to validate.

New Report Project Wizard: Default Library Configuration - Create a new data source

Configure the new data source and specify your login credentials.



Parameter	Definition
Create default library	Select whether to download from SOA Systinet server or create from template.
Driver Class	Select the driver class from the drop-down list.
Database URL	Input the location of the database as a JDBC connection string.
User Name and Password	Input a user name and password to access the database.
JNDI URL	The JNDI name of the data source. For example, `java:comp/env/jdbc/ReportingDataDS`.
Manage Drivers	Add, delete, or restore JAR files, and edit drivers.
Test Connection	Test the server connection using the specified parameters.

B Troubleshooting

This appendix describes some common problems and their resolutions:

Workbench Looks Different from the Pictures in the Documentation

Problem

Workbench does not look exactly as depicted in the user documentation. The most likely reason for this disparity is that you are in the wrong perspective.

Solution

Switch to the Platform perspective:

- From the menu, select **Window→Open Perspective→Platform**.

The Platform Perspective is Missing Views

Problem

There are views missing from the Platform perspective. The most likely reason for this is that some of the views are closed.

Solution

Reset the perspective:

- From the menu, select **Window→Reset Perspective**.

Incomplete Table List

Problem

The list of tables is incomplete. Workbench limits the number of schemas and tables accessed for performance reasons.

Solution

Change the settings, as described in Changing the Data Collection Settings on page 44.

Oracle Developer Queries Cause Errors in Workbench

Problem

Workbench returns errors for imported queries. The most likely reason is that your table names do not include the schema.

Solution

Include the schema name in the **FROM** clause of your query. For example, `soadb.ry_docrel`.

Workbench Displays Scrambled Table Names

Problem

The tables names are unreadable. These are most likely to be deleted tables stored in Oracle 10gR2.

Solution

Either change the maximum number of tables to 3000, as described in Changing the Data Collection Settings on page 44, or purge the deleted tables.

To purge deleted tables:

1. Log on to the sqlplus ot isqlplus console as the system user.
2. Enter the following commands:
   - **PURGE RECYCLEBIN**
   - **PURGE DBA_RECYCLEBIN**

The Report Result does not Appear

Problem

The report result is not deployed to the SOA Systinet server.

Solution

Check the report result:

1. In your browser, navigate to the latest report instance page. For example, `https://hostname:port/reporting/reports/service_portfolio/documents/1/?alt=text/plain`.
2. Check the following status elements:
   - `<category label="finished" scheme="urn:com:systinet:reporting:execution:status" term="finished"/>`
   - `<category label="report document" scheme="urn:com:systinet:reporting:kind" term="urn:com:systinet:reporting:kind:document"/>`

   A correctly finished report contains the *`finished`* value.

If the report execution does not finish, check the JMS settings, JMS queue, or the reporting service logs to analyze the bottleneck.