HP SOA Systinet Workbench

Software Version: 3.20

Customization Editor Guide

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.

Copyright Notices

Contents

About this Guide

Welcome to the *Customization Editor Guide*. This guide describes how to use Customization Editor to customize an installation of HP SOA Systinet.

This guide contains the following chapters:

- Chapter 1, Customization Editor

  Introduces Customization Editor, its main features, and the customizable elements of SOA Systinet.

- Chapter 2, Getting Started

  Explains how to configure Customization Editor, search an extension, and save your changes.

- Chapter 3, Manipulating Artifact Types

  Explains how to customize the attributes, properties, and appearance of artifacts.

- Chapter 4, Manipulating Properties

  Explains how to create and modify properties and property groups.

- Chapter 5, Configuring the User Interface

  Explains how to customize the main tabs of SOA Systinet, and create new UI elements.

- Chapter 6, Creating and Using Components

  Explains how to create customized Java and JSPs, and use these components with SOA Systinet.

- Chapter 7, Modifying Taxonomies

  Explains how to modify the set of available taxonomies.

- Chapter 8, Deploying Extensions

Explains how to prepare and deploy your extension to SOA Systinet.

- Appendix A, Views

Describes Customization Editor views in detail.

- Appendix B, Dialog Boxes

Describes Customization Editor dialog boxes in detail.

- Appendix C, Eclipse Plug-in Requirements

Required plug-ins when installing Workbench as an update.

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 Customization Editor

HP SOA Systinet Customization Editor is a set of Workbench features that enable you to edit the underlying SOA Definition Model (SDM) configuration and the user interface of an installation of SOA Systinet.

This chapter introduces Customization Editor in the following sections:

- Workbench Suite on page 13

- Overview on page 14

- User Interface on page 15

- Customizable Elements on page 26

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

SOA Systinet is distributed with a preconfigured data model and user interface. For typical deployment scenarios, both need to be customized to fit an organization-specific environment.

HP SOA Systinet Customization Editor provides a mechanism to customize both the model and the appearance of the user interface, implement those changes, and keep them isolated in a so-called extension package that can then be re-applied during a re-installation or upgrade of SOA Systinet.

To customize SOA Systinet, follow this process:

1 Create an extension project.

    For details, see Creating an Extension Project on page 40.

2 Modify the data model and user interface.

    For details, see the following:

    - Chapter 3, Manipulating Artifact Types

    - Chapter 4, Manipulating Properties

    - Chapter 5, Configuring the User Interface

    - Chapter 6, Creating and Using Components

    - Chapter 7, Modifying Taxonomies

3 Deploy the customization to SOA Systinet.

    For details, see Chapter 8, Deploying Extensions.

Be very careful when modifying artifacts that already have instances in the repository. For example, a new installation already contains the person artifact *administrator*. Removing properties from an artifact type with instances, or adding a new required property without a default value, may cause instability in SOA Systinet. Customization Editor should prevent this if the SOA Systinet server is running during modification.

User Interface

The default **Extension** perspective is split into a number of sections with menu options across the top, as shown in Figure 1, "Customization Editor UI".

Figure 1. Customization Editor UI
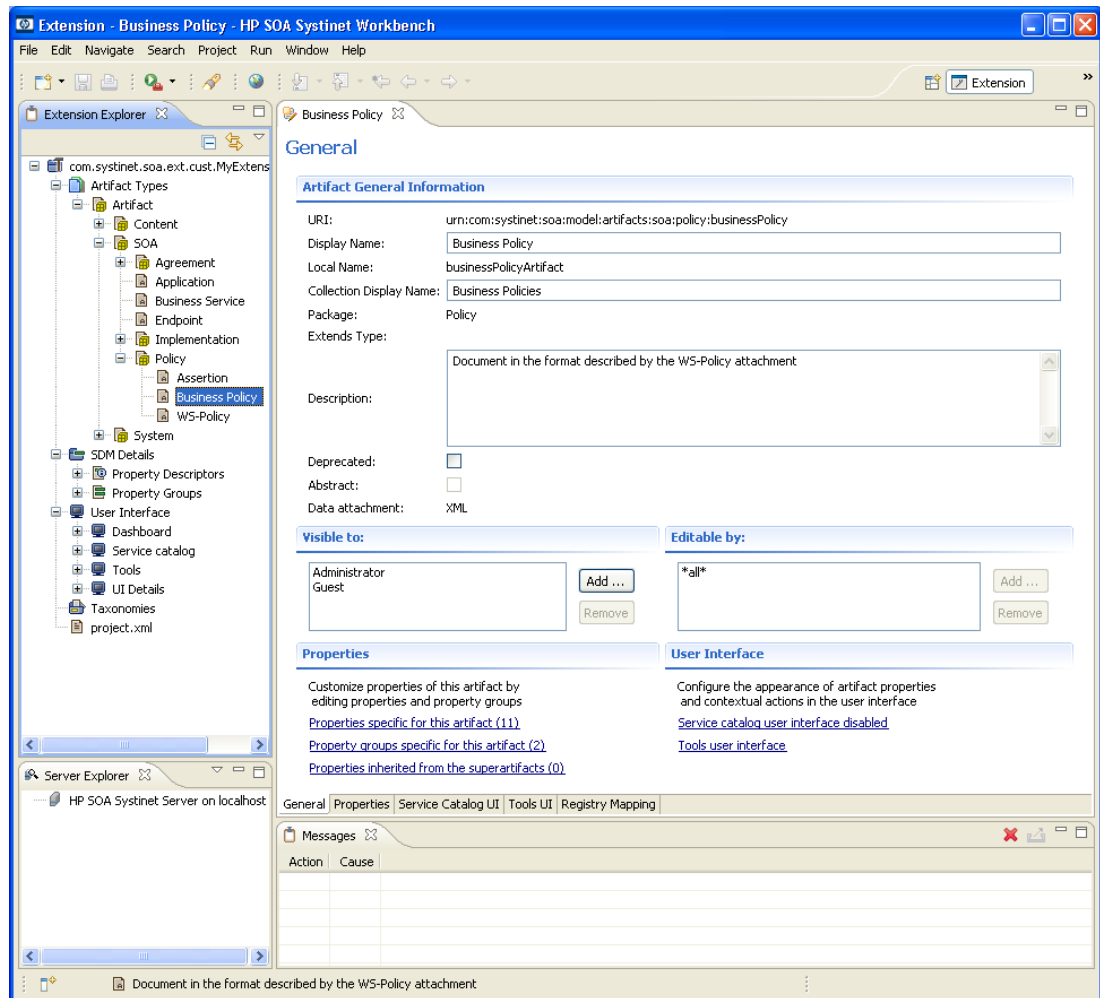
The perspective consists of the following elements:

• **Project Explorer**

The tree view of the SDM configuration represented by your project.

For details, see Extension Explorer on page 17.

- **Server Explorer**

  The view listing SOA Systinet server connections to Workbench.

  For details see, Server Explorer on page 24.

- **Editor Views**

  The main area of the perspective contains editor views of project elements. You can open multiple editors. The functionality of each editor varies depending on the project element.

  For details, see Appendix A, Views.

- **Information Views**

  Messages and search results open their own views.

  For details, see Appendix A, Views.

- **Menu Items**

  The standard Eclipse menu options plus additional Workbench options.

  For details about the additional options for Customization Editor, see Menu Options on page 25.

Extension Explorer

The Extension Explorer, as shown in Figure 2, "Extension Explorer View", represents a tree view of the extension project that you are currently working with. It includes all the elements from extensions that it depends on as well.

Figure 2. Extension Explorer View
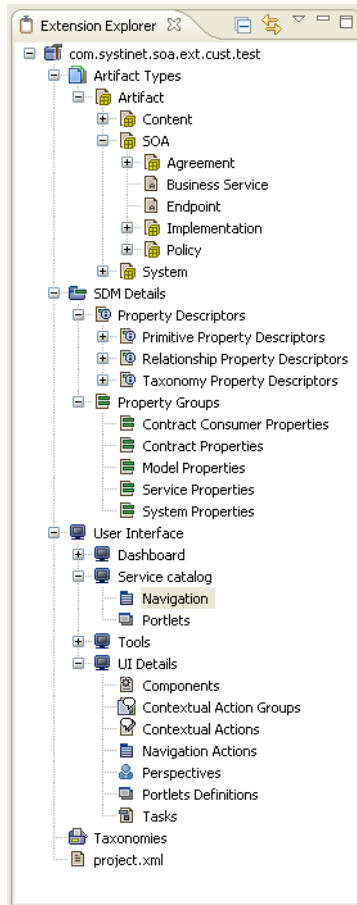
The tree is split into the extension root and five branches representing different elements of your extension.

Double-click an item in the tree to open the relevant editor, ot to open or close a branch.

For details about each editor, see Appendix A, Views.

Right-click an element to open a context menu that contains options for the element.

Each element and its contextual functionality is described in the following sections:

Extension Name

The extension name is the root element of the project.

The context menu contains the option to **Build Extension**, which creates a jar file that contains your extension. This extension is ready to import to SOA Systinet.

For deployment details, see Chapter 8, Deploying Extensions.

Artifact Types

The Artifact Types branch contains the artifact types organized into a hierarchical structure reflecting the SOA Definition Model (SDM). Each element within **Artifact Types** is a type of artifact or artifact package.

> An artifact package is an artifact type that serves as a group of other artifacts. They do not have instances in the repository and just serve to organize a hierarchy of artifact types in the SDM.

For details, see Artifacts on page 27.

The context menu for the Artifact Types branch contains the following functionality:

- **New Artifact Type** or **Package**

  Creates an artifact type as described in Creating an Artifact Type or Package on page 45.

- **Open Artifact Type** or **Package**

  Opens the artifact editor for the selected element with the name of the element as the editor title.

  For details, see Artifact Editor on page 93.

- **Deprecate Artifact Type or Package**

  Deactivates the artifact type in the extension project and hides it in the SOA Systinet UI.

- **Undeprecate Artifact Type** or **Package**

  Reactivates a deprecated artifact type in the extension project.

- **Delete Artifact Type** or **Package**

  Removes the artifact type from the extension project.

SDM Details

The SDM Details branch contains two element types described in the following sections:

Property Descriptors

Property descriptors are the attributes of artifact types.

For details, see Properties on page 27.

In the Extension Explorer, properties are divided into the following branches:

- Primitive Property Descriptors
- Relationship Property Descriptors
- Taxonomy Property Descriptors

Each property type branch contains one context menu option: New Property.

For details, see Creating a Property on page 61.

Each property type branch contains a list of the properties of that type.

The context menu for properties contains the following options:

- **Open Property**

  Opens the property editor for the selected element with the name of the element as the editor title.

  For details, see Property Editor on page 119.

- **Deprecate Property**

  Deactivates the property in the extension project and hides it in the SOA Systinet UI.

- **Undeprecate Property**

  Reactivates the deprecated property in the extension project.

- **Delete Property**

  Removes the property from the extension project.

- **Find Usage**

  Opens a search view displaying all the artifacts and property groups containing this property.

  For details, see Search View on page 125.

Property Groups

Property groups are sets of properties that can be added collectively to an artifact type.

For details, see Property Groups on page 29.

The context menu of the Property Group branch contains one option: New Property Group.

For details, see Creating a Property Group on page 63.

The context menu for each property group contains the following options:

- **Open Property Group**

  Opens the property group editor for the selected element with the name of the element as the editor title.

  For details, see Property Group Editor on page 123.

- **Delete Property Group**

  Removes the property group from the extension project.

- **Find Usage**

  Opens a search view displaying all the artifacts that contain this property group.

  For details, see Search View on page 125.

User Interface

The User Interface branch contains two element types described in the following sections:

- SOA Systinet Tabs on page 22
- UI Details on page 23

SOA Systinet Tabs

Each customizable tab in the SOA Systinet UI has its own branch.

For details, see SOA Systinet Tabs on page 29.

Each branch contains the following elements:

- Navigation
- Portlets

Open the editor to view and edit the elements for that SOA Systinet tab.

For details, see Navigation Editor on page 109 and Portlets Editor on page 113.

UI Details

UI Details contains the customizable elements of the SOA Systinet UI.

For details, see UI Elements on page 29.

Each element context menu contains the option to create a new element, as described in the following sections:

- Creating a Component on page 74

- Creating a Context Action Group on page 69

- Creating a Contextual Action on page 68

- Creating a Navigation Action on page 68

- Creating a Perspective on page 70

- Creating a Portlet on page 70

- Creating a Task on page 74

Taxonomies

Taxonomies are categorization groups.

For details, see Taxonomies on page 29.

Open the Taxonomies editor to control the taxonomies in your extension project.

For details, see Taxonomies Editor on page 127.

You can create a new taxonomy from the context menu.

Open the Taxonomies context menu and select **New→Taxonomy.**

The New Taxonomy dialog opens. For details see New Taxonomy on page 158.

For more details about creating and working with taxonomies, see the *HP SOA Systinet Taxonomy Editor Guide*.

project.xml

Open the project.xml view to configure the current extension project.

For details, see project.xml Editor on page 116.

Server Explorer

The Server Explorer displays the SOA Systinet servers connected to Workbench, as shown in Figure 3, "Server Explorer View". The functionality is shared by all the Workbench editors.

Figure 3. Server Explorer View
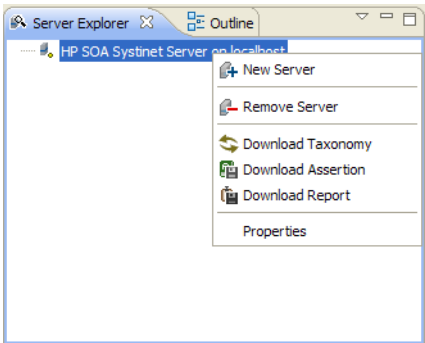
Right-click a server in the Server Explorer to open the context menu described in Table 1, "Server Explorer Context Menu Options".

Table 1. 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.

Option	Function
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.

Menu Options

The menus in Workbench are standard Eclipse menus, with the addition of specific functionality for Workbench editors.

The additional menu options for Customization Editor include:

- **File→New→HP SOA Extension Project**

  Create a new project, as described in Creating an Extension Project on page 40.

- **File→New→Artifact Type**

  Create an artifact type, as described in Creating an Artifact Type or Package on page 45.

- **File→New→Artifact Package**

  Create an artifact package, as described in Creating an Artifact Type or Package on page 45.

- **File→New→Property**

  Create a property, as described in Creating a Property on page 61.

- **File→New→Property Group**

  Create a property group, as described in Creating a Property Group on page 63.

- **File→New→Perspective**

  Create a perspective, as described in Creating a Perspective on page 70.

- **File→New→Component**

  Create a component, as described in Creating a Component on page 74.

- **File→New→Task**

  Create a task, as described in Creating a Task on page 74.

- **Navigate→Go To→Open Artifact**

  Open an artifact editor, as described in Artifact Editor on page 93.

  The keyboard shortcut **Ctrl**+**Alt**+**A** also performs this function.

- **Navigate→Go To→Open Property**

  Open a property editor, as described in Property Editor on page 119.

  The keyboard shortcut **Ctrl**+**Alt**+**P** also performs this function.

- **Navigate→Go To→Open Property Group**

  Open a property group editor, as described in Property Group Editor on page 123.

  The keyboard shortcut **Ctrl**+**Alt**+**R** also performs this function.

- **Search→Search**

  Open a search dialog box, as described in Searching the Extension on page 43.

- **Help→Dynamic Help**

  Access context-sensitive help, directing you to the relevant section of this document based on the active view in Customization Editor.

Customizable Elements

The Customization Editor enables you to modify almost any aspect of an installation of SOA Systinet.

These elements are described in the following sections:

SDM Elements

The Customization Editor manages four types of entities in the SDM configuration of SOA Systinet.

These entity types are described in the following sections:

Artifacts

Artifacts are the basic building blocks of SOA. Every entity in the repository is an instance of an artifact. Each type of artifact is defined by an artifact type in the SDM. Artifact types are further categorized into artifact packages. These packages are abstract artifact types that do not have instances in the repository but instead define groups of artifacts.

You can explore artifact types and packages in the Artifact Types branch of the Extension Explorer.

For details, see Artifact Types on page 19.

For artifact type procedures, see Chapter 3, Manipulating Artifact Types.

Properties

Properties are the attributes of artifacts, the labels that distinguish one instance of an artifact from another. For example, all artifact instances have the `name` property, but every artifact has a different name.

There are three categories of property:

- **Primitive Properties**

The basic labels for artifacts such as name and description.

Basic properties are of the following types:

Property Type	Description
addressPropertyType	A full postal address.
booleanPropertyType	True or false.
categoryBagPropertyType	A taxonomic categorization.
dailyIntervalPropertyType	A time period with a start and end day, for example Monday to Friday.
dateTimePropertyType	A specific date and time.
doublePropertyType	A double precision floating point number.
encryptedPasswordPropertyType	An encrypted password.
identifierBagPropertyType	A taxonomic identification.
integerPropertyType	An integer number.
nameUrlPairPropertyType	A URL with an optional name assigned.
nameValuePairPropertyType	A name and value pair.
plainTextPropertyType	One-line text, suitable for properties such as name.
textareaPropertyType	Multi-line text, suitable for properties such as description.

- **Relationship Properties**

  The nature and direction of a relationship with another artifact such as ImplementationOf.

- **Taxonomy Properties**

  The categories used to define an artifact based on taxonomies such as the High category in the Failure Impact taxonomy.

You can explore properties in the SDM Details branch of the Extension Explorer.

For details, see SDM Details on page 20.

For property procedures, see Chapter 4, Manipulating Properties.

Property Groups

You can organize properties into groups, and add them to artifacts collectively instead of individually. For example, you can organize name, address, and telephone number into a contact details property group. Then, whenever a new artifact type is created that represents a person, the group can be added to it instead of the individual properties.

You can explore property groups in the SDM Details branch of the Extension Explorer.

For details, see SDM Details on page 20.

For property group procedures, see Chapter 4, Manipulating Properties.

Taxonomies

Taxonomies are categorization groups, each of which contain a set of values within a single category. For example, office location can be a taxonomic group containing a set of values representing each office in an organization.

You can explore taxonomies in the Taxonomies branch of the Extension Explorer.

For details, see Taxonomies on page 23.

For taxonomy procedures, see Chapter 7, Modifying Taxonomies.

UI Elements

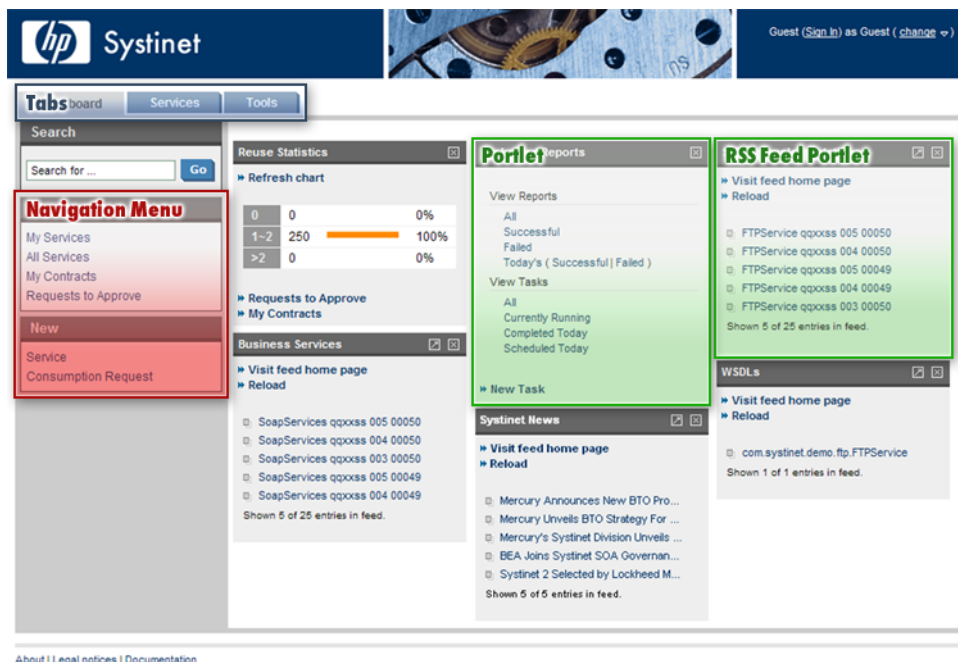You can configure two types of SOA Systinet pages and their UI elements.

These are described in the following sections:

- SOA Systinet Tabs on page 29

- Artifact Pages on page 31

SOA Systinet Tabs

The main SOA Systinet tabs consist of the tab heading, a menu on the left and a set of portlets, as shown in Figure 4, "SOA Systinet Tab".

Figure 4. SOA Systinet Tab

The configurable elements of SOA Systinet tabs are described in the following sections:

- Navigation Menus on page 30
- Portlets and Feeds on page 31

The SOA Systinet Policies tab is not customizable.

Navigation Menus

The navigation menu is organized into navigation groups and actions.

Access the details for each SOA Systinet tab from the User Interface→*Tab Name*→Navigation branch in the Extension Explorer.

Access the navigation actions in your extension from the User Interface→UI Details→Navigation Actions branch in the Extension Explorer

For more information about configuring the navigation actions of an SOA Systinet tab, creating and modifying navigation actions, and organizing navigation actions into groups, see Chapter 5, Configuring the User Interface.

Portlets and Feeds

SOA Systinet tabs contain portlets in the main area of the screen.

Access the details for each SOA Systinet tab from the User Interface→*Tab Name*→Portlets branch of the Extension Explorer.

Access the portlets in your extension from the User Interface→UI Details→Portlets Definitions branch in the Extension Explorer.

For more information about configuring the portlets of an SOA Systinet tab and creating new portlets, see Chapter 5, Configuring the User Interface.

Artifact Pages

Artifact pages consist of the page heading, a set of contextual menus, and the main layout of the page, as shown in Figure 5, "SOA Systinet Artifact Page".

Figure 5. SOA Systinet Artifact Page

The configurable elements of artifact pages are described in the following sections:

- Contextual Action Menus on page 33

- Page Layout on page 33

Contextual Action Menus

The contextual menus are organized into UI groups and contextual actions.

Access the details for each artifact type from the Service Catalog UI and Tools UI tabs of the artifact editor.

Access the contextual actions and groups of actions in your extension from the User Interface→UI Details→Contextual Actions and Contextual Action Groups branches in the Extension Explorer.

For details about configuring the contextual menus of an artifact page, see Chapter 3, Manipulating Artifact Types.

For details about creating context actions and organizing them into groups, see Chapter 5, Configuring the User Interface.

Page Layout

You can configure the page layout of artifact detail pages by adding or modifying the UI elements they contain.

Access the details for each artifact type from the Service Catalog UI and Tools UI tabs of the relevant artifact editor.

For details about configuring artifact page layouts, see Chapter 3, Manipulating Artifact Types.

2 Getting Started

You can use Customization Editor to create extensions to the core SOA Systinet configuration. The core configuration itself is not modified. Instead, SOA Systinet includes a utility for adding extensions to the configuration.

This chapter contains the following sections:

Installing Workbench

HP SOA Systinet Workbench is an Eclipse development platform distributed as a zip file, `hp-soa-systinet-workbench-3.20-win32.zip` or as a plugin for an existing Eclipse environment, `hp-soa-systinet-workbench-3.20-plugin.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   Ensure that your Eclipse platform contains the necessary plug-ins. For details, see Appendix C, Eclipse Plug-in Requirements.

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-3.20-plugin.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 3.20**

      HP SOA Systinet Workbench splash screen.

- **Taxonomy Editor 3.20**

 Required for Customization Editor.

- **Customization Editor 3.20**

- **Assertion Editor 3.20**

- **Report Editor 3.20**

- **Common Plugin 3.20**

  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 3.20` 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, "Workbench Welcome Screen".

Figure 6. Workbench Welcome Screen
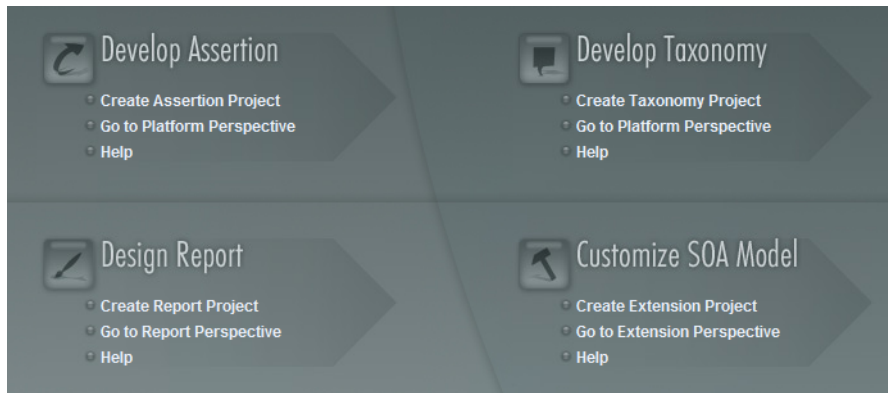
Select one of the options to open one of the editor tools, start a new editing project, or view the documentation set.

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
```

Creating an Extension Project

The first step in customizing your SOA Systinet installation is to create an extension project in the Customization Editor. This is the workspace where you make all your modifications.

 supports the separation of projects based on their type. The following extension project types are supported:

- **model**

  Extensions for changes to the data model and the appearance of existing UI components.

- **code**

  Extensions for custom components are their related UI configuration.

- **mixed**

  Combined model and code extensions.

HP Software recommend separating extensions into separate types because SOA Systinet does not guarantee backward compatibility for the Java API. This may make it difficult to migrate custom components to newer versions.

model extensions are easily migrated by opening them in the latest version of Customization Editor, building a new extension and redeploying them.

You can separate older or **mixed** extensions by opening them in the latest version of Customization Editor and selecting the **File→Split Mixed Extension**.

To create an extension project:

1 In the menu, select **File→New→Project**.

   The New Project wizard opens.

2 Expand **HP SOA SystinetExtension Project**, and click **Next**.

3 In the New Extension Project dialog box, add the parameters you want.

For parameter descriptions, see New Extension Project: Specify HP SOA Systinet Server Installation on page 144.

4   Click **Next** to select or create a server.

If no servers are currently configured the dialog continues to Step 6.

5   Do one of the following:

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

    Continue to Step 6.

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

    Continue to Step 7.

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

    For parameter descriptions, see New Extension Project: New Server on page 146.

7   Do one of the following:

  • Select **Create a New Extension Project from Scratch** to create an empty extension containing no elements at all.

    Click **Next**.

  • Select **Edit an Existing Extension** to open an existing extension for modification.

    Select an extension from the extension folder, or use **Add Extension** to select one from another location.

    Click **Next** to continue to Step 10.

- Select **Create a New Extension from an Existing One** to use an existing extension as the basis of a new one.

  Select an extension from the extension folder or use **Add Extension** to select one from another location.

  Click **Next**.

  > ⚠️ If you use this option, you must remove the existing extension from the extensions folder before you deploy the new modified extension.

8   In the Create Extension or Project dialog box, add the parameters you want, and then click **Next**.

  For parameter descriptions, see New Project: Create a New Extension on page 149.

9   In the Add Dependencies dialog box, optionally select or add extensions to depend on, and then click **Next**.

10   In the HP SOA Systinet Extension Project dialog box, add the parameters you want, and then select available projects to reference.

  > ▶ Adding referenced projects enables you to use assertions or taxonomies from the referenced projects without the need to copy them manually to the extension project.

11   Click **Finish** to open the project.

In the Platform perspective, the Project Explorer now displays a view of your extension that contains not only the elements contained in your extension project but all the elements from any other extensions that your project depends on.

For details about the contents of your extension project, see Extension Explorer on page 17.

Sharing an Extension Project

SOA Systinet enables the sharing of an extension project in the CVS repository.

To share an extension project:

1 In Extension Explorer, right-click the project you need to open its context menu, and select **Team→Share Project** to open the **Share Project** dialog box. For details, see Share Project on page 165.

2 Enter the required parameters and click **Finish**.

The project is published to the CVS repository.

Referencing Other Workbench Projects

You can include assertions and taxonomies from an Assertion or Taxonomy Project in your current Extension Project.

To reference an Assertion or Taxonomy Project:

1 From the platform perspective, right-click your extension project and select **Properties** to open the **Preferences** dialog box, and then select **Project References** to view projects available to reference.

2 Select the projects you want to reference and click **OK**.

The assertions or taxonomies from the referenced project are copied to your extension project.

Searching the Extension

The Customization Editor contains a customized search function, which enables you to find entities in your extension project.

To search your extension project:

1 From the menu, select **Search→Search**.

The Search dialog box opens.

2   In the SDM Search tab of the Search dialog box, add the parameters for your search.

For parameter descriptions, see Search on page 164.

3   Click **Search**.

The search results appear in the Search view.

For details, see Search View on page 125.

Saving Modifications

As you modify an entity in your extension project, the tab label of the entity editor is marked with an asterisk (*).

To save your changes to the project, select **File→Save** from the menu, or use keyboard shortcut **Ctrl+S**.

If you close an editor or the application with unsaved changes, you are prompted to save these changes.

Your modifications are made to your extension project and not to the configuration of SOA Systinet. To deploy your customization to SOA Systinet, see Chapter 8, Deploying Extensions.

3 Manipulating Artifact Types

You can use Customization Editor to create, modify, and delete artifact types and packages in your extension project. You can also change the format of artifact pages in the SOA Systinet UI.

For more details about artifact types and packages, see Artifacts on page 27.

The chapter explains the following procedures:

- Creating an Artifact Type or Package on page 45

- Modifying the Attributes of an Artifact Type on page 46

- Modifying the Properties of an Artifact Type on page 47

- Modifying an Artifact Type in the Services UI on page 49

- Modifying an Artifact Type in the Tools UI on page 50

- Modifying the Layout of View Artifact Pages on page 51

- Modifying the Contextual Actions of an Artifact on page 53

- Mapping an Artifact Type to a Registry on page 55

- Example: Adding the Department Property to Business Services on page 57

Creating an Artifact Type or Package

Customization Editor enables you to create a new artifact type or package in your extension project.

To create an artifact type or package:

1   Do one of the following:

- From the main menu, select **New→Artifact Type** or **Package**.

- In the Extension Explorer, open the context menu of an artifact package branch and select **New Artifact Type** or **Package**.

The New Artifact Type or New Artifact Package dialog box opens.

2   In the dialog box, add the artifact parameters you want.

For parameter descriptions, see New Artifact: Create on page 139.

3   Do one of the following:

- Click **Finish** to create the artifact type or package and exit the dialog box.

- Click **Next** and continue to Step 4 to set visibility parameters for the artifact in the SOA Systinet UI.

4   In the dialog box, add the visibility parameters you want.

For parameter descriptions, see New Artifact: Visibility on page 140.

5   Do one of the following:

- Click **Finish** to create the artifact type or package and exit the dialog box.

- Click **Next** and continue to Step 6 to set database parameters.

6   In the dialog box, add the database parameters you want.

For parameter descriptions, see New Artifact: Database Settings on page 141.

7   Click **Finish** to create the artifact type or package.

Modifying the Attributes of an Artifact Type

Customization Editor enables you to edit the main attributes of artifact types.

To edit the attributes of an artifact type:

1   Open the artifact editor and select the **General** tab.

2   In the General tab, make any changes you want.

> You cannot change the local name of the artifact if the extension project is in production mode.

For a description of the General tab, see Artifact Editor: General Tab on page 93.

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

Modifying the Properties of an Artifact Type

The main attributes of artifact types are properties. You can modify these properties, add them individually, or add them as a group within the artifact editor.

To add or modify the properties and property groups of an artifact type:

1   Open the artifact editor and select the **Properties** tab.

2   In the Properties tab, make any changes you want.

For a description of the Properties tab, see Artifact Editor: Properties Tab on page 95.

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

You can change the local name of the property descriptor defined in the current extension project.

To change the local name of a property descriptor:

1   From the tree menu of Extension Explorer, expand **SDM Details**→**Property Descriptors**.

2   Browse for the descriptor you want, and double-click to open the **General** view.

3   In the **Local Name** field, enter the parameters you require, and then save your changes.

You cannot change the local name of the property descriptor if the extension project is in production mode.

You can also change the cardinality of an artifact defined in the current extension project.

To change the cardinality of an artifact type:

1   Browse for the artifact you need and open the Editor view, and then switch to the **Properties** tab.

2   In the **Cardinality** field, click to open the drop-down menu, and select the cardinality you need.

> For taxonomic properties, the cardinality affects the way the property displays in the SOA Systinet UI.
>
> • Required taxonomic properties display as an **Add Category** function.
>
> • Optional taxonomic properties display as a drop-down menu.
>
> • Multiple taxonomic properties display as a list box.

> You cannot change the cardinality of an artifact if the extension project is in production mode.

You can also change the direction of relationship of a property descriptor which is defined in the current extension project.

To change the direction of a relationship property descriptor:

1   In Extension Explorer, expand **SDM Details→Property Descriptors→Relationship Property Descriptors** and double-click the descriptor you need to open its Editor view.

2   In the **Direction** field, click the link next to **Inverse Display Name** to toggle the relationship between **incoming** or **outgoing**, and then save your changes.

SOA Systinet also enables you to modify the taxonomical properties of an artifact type.

To modify the taxonomical properties of an artifact:

1   In Extension Explorer, browse for the artifact you need and open its Editor view, and then switch to the **Properties** tab.

2   In the Properties window, click **New** to open the **New Property** dialog box, and select **Taxonomy Property**, and then click **Next**.

The **New Taxonomy Property** dialog box opens. For details, see New Property: Taxonomy on page 154.

3   Enter general parameters and in the **Taxonomy Name** field, do one of the following:

-   Click **Browse** to browse for and select from a list of available taxonomies.

-   Click **Create** to open the Create a New Taxonomy dialog box,. For details, see New Taxonomy on page 158.

4   Enter the parameters you want and click **Finish**.

The created taxonomy is now visible in the **New Taxonomy Property** dialog box.

5   Click **Finish**.

Modifying an Artifact Type in the Services UI

SOA Systinet enables you to configure the appearance of an artifact type in the Services UI.

To modify the appearance of Service UI pages:

1   Open the artifact editor and select the **Service Catalog UI** tab.

2   Select **Is Artifact Available in Service Catalog** to make the artifact type visible in the Services UI.

3   In the Service Catalog UI tab, make any changes you want.

For a description of the Service Catalog UI tab, see .

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

To improve usability, SOA Systinet enables you to move artifacts between groups of artifact types.

To move artifacts between groups:

*   In the Service Catalog UI tab, select the artifact you need from the **Layout** pane and do one of the following:

  • Click **Up** or **Down** repeatedly until the artifact is in the group you want.

  • **Drag** the artifact you want from the resource group and **Drop** the artifact into the target group.

> You cannot move artifacts from the **Layout** pane to the **Contextual Action** pane.

Modifying an Artifact Type in the Tools UI

SOA Systinet enables you to modify the appearance of an artifact type in the view artifact and browse artifact pages of the Tools UI.

To modify the appearance of Tools UI pages:

1   Open the artifact editor and select the **Tools UI** tab.

2   In the Tools UI tab, make any changes you want.

  For a description of the Tools UI tab, see .

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

To improve usability, SOA Systinet enables you to move artifacts between groups of artifact types.

To move artifacts between groups:

- In the Tools UI tab, select the artifact you need from the **Layout** pane and do one of the following:

  - Click **Up** or **Down** repeatedly until the artifact is in the group you want.

  - **Drag** the artifact you want from the resource group and **Drop** the artifact into the target group.

    You cannot move artifacts from the **Layout** pane to the **Contextual Action** pane.

Modifying the Layout of View Artifact Pages

The View Artifact page shows the details of an artifact in both the Services and Tools UI of SOA Systinet. You can modify the elements that appear there with Customization Editor.

To modify the layout of view artifact pages:

1 Open the artifact editor and select the tab for the view you want to modify:

   - **Service Catalog UI**

   - **Tools UI**

2 In the selected UI tab Layout pane, make any changes you want.

   For a description of the UI tabs, see the following:

   - Artifact Editor: Service Catalog UI Tab on page 98

   - Artifact Editor: Tools UI Tab on page 100

   To add a group in the Service Catalog UI or the Tools UI, click **Add** in the layout pane to open the Choose Type dialog box, select **Group** and click **Next**, and then enter the name you want and click **Finish.**

To add a property, see Adding a Property to View Artifact Pages on page 52.

To add a component, see Adding a Component to View Artifact Pages on page 52.

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

> To add a table of related artifacts to the View Service page in the Services tab, use the generic association component.
>
> For details, see Generic Association Control on page 76 for details.

Adding a Property to View Artifact Pages

View Artifact pages usually display a set of properties for that artifact type.

To add a property to view artifact pages:

1   Open the Artifact Editor and select the UI tab for the view you want to modify.

2   In the Layout pane, click **Add**, select **Property**, and then click **Next**.

The Layout Property Details dialog box opens.

3   In the Layout Property Details dialog box, add the parameters you want.

For parameter descriptions, see Layout: Property Details on page 132.

4   Click **Finish** to add the property to the layout.

Adding a Component to View Artifact Pages

View Artifact pages can contain components to perform specific functionality.

To add a component to view artifact pages:

1   Open the Artifact Editor and select the UI tab for the view you want to modify.

2   In the Layout pane, click **Add**, select **Component**, and then click **Next**.

The Layout Component Details dialog box opens.

3   In the Layout Component Details dialog box, add the parameters you want.

    For parameter descriptions, see Layout: Component Details on page 133.

4   Click **Finish** to add the component to the layout.

Modifying the Contextual Actions of an Artifact

At the top of the View Artifact page (Services and Tools UI) and the Browse Artifact page (Tools UI) is a section containing sets of contextual actions. You can modify the actions that appear there with Customization Editor.

For details, see Artifact Pages on page 31.

To modify the contextual actions in view and browse artifact pages:

1   Open the artifact editor and click the tab for the view you want to modify:

    • **Service Catalog UI**

    • **Tools UI**

2   In the selected UI tab Contextual Actions pane, make any changes you want.

    For a description of the UI tabs, see the following:

    • Artifact Editor: Service Catalog UI Tab on page 98.

    • Artifact Editor: Tools UI Tab on page 100.

    For details on how to add a contextual action, see Adding a Contextual Item to Artifact Pages on page 54.

    For details on how to add a set of contextual items, see Adding a Contextual Item Group to an Artifact Page on page 54.

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

Adding a Contextual Item to Artifact Pages

You can modify the contextual action menus for artifact pages.

To add a contextual item to an artifact page:

1   Open the Artifact Editor and select the UI tab for the view you want to modify.

2   For the Tools UI, in the Contextual Actions pane, select which view to modify:

   • **Artifact Detail** to modify the View Artifact page.

   • **Collection** to modify the Browse Artifact page.

3   In the Contextual Actions pane, click **Add**, select **Contextual action item**, and then click **Next**.

   The Contextual Action Item Details dialog box opens.

4   In the Contextual Actions Item Details dialog box, add the parameters you want.

   For parameter descriptions, see Contextual Actions: Item Details on page 130.

5   Click **Finish** to add the element to the context action menu.

Adding a Contextual Item Group to an Artifact Page

You can organize contextual actions into groups and add them to artifact pages collectively.

To add a contextual action group to artifact pages:

1   Open the Artifact Editor and select the UI tab for the view you want to modify.

2   For the Tools UI, in the Contextual Actions pane, select which view to modify:

   • **Artifact Detail** to modify the View Artifact page.

   • **Collection** to modify the Browse Artifact page.

3   In the Contextual Actions pane, click **Add**, select **Contextual Action Group Reference**, and click **Next**.

    The Group Reference Detail dialog box opens.

4   Select the **Contextual Action Group** for the required set of actions and the **UI group** to add them to.

5   Click **Finish** to add the set of actions to the selected UI group.

Mapping an Artifact Type to a Registry

You can map artifacts in SOA Systinet to registry entities in a UDDI registry.

The artifact types listed in Table 2, "Predefined Registry Mapping" have predefined mappings to registry and cannot be amended.

Table 2. Predefined Registry Mapping

SDM Artifact	UDDI Entity	Direction
Organizational Unit	BusinessEntity	Bi-directional
Business Service	tModel	Bi-directional
Implementation Artifacts	Business Service	Bi-directional
Endpoint	BindingTemplate	Bi-directional

The mapping of an artifact to a registry entity is illustrated in Figure 7, "Registry Mapping".

Figure 7. Registry Mapping
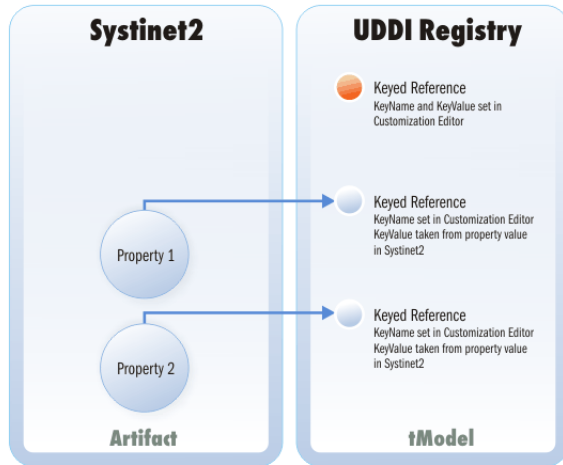
To add or modify artifact type mappings to registry entities:

1   Open the artifact editor and select the **Registry Mapping** tab.

2   In the Registry Mapping tab, make any changes you want.

     For a description of the Registry Mapping tab, see Artifact Editor: Registry Mapping Tab on page 103.

     For details on how to add an artifact mapping, see Adding a Registry Mapping for an Artifact Type on page 56.

     For details on how to add a property mapping, see Adding a Registry Mapping for a Property on page 57.

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

Adding a Registry Mapping for an Artifact Type

You can map artifact types to registry entities.

To add an artifact mapping:

1 Open the Artifact Editor and select the **Registry Mapping** tab.

2 In the Artifact Mapping pane, click **Add**.

  The New Artifact Mapping dialog box opens.

3 In the New Artifact Mapping dialog box, add the parameters you want.

  For parameter descriptions, see Registry Mapping: New Artifact Mapping on page 162.

4 Click **Finish** to add the mapping.

Adding a Registry Mapping for a Property

You can map properties to registry entity keyed references.

To add a property mapping:

1 Open the Artifact Editor and select the **Registry Mapping** tab.

2 In the Property Mapping pane, click **Add**.

  The New Property Mapping dialog box opens.

3 In the New Property Mapping dialog box, add the parameters you want.

  For parameter descriptions, see Registry Mapping: New Property Mapping on page 163.

4 Click **Finish** to add the mapping.

Example: Adding the Department Property to Business Services

In this example, you add a new taxonomic property, department, to the business service artifact, and make it visible in the Service View of a business service. The Service Properties section of the Service View is shown in Figure 8, "Default Service View of a Business Service".

Figure 8. Default Service View of a Business Service

To follow this example, you must first do the following:

• In Customization Editor, create a new extension dependent on the core extension, as described in Creating an Extension Project on page 40.

• In Taxonomy Editor, create a new department taxonomy and deploy it to SOA Systinet, as described in the "Example: Creating and Publishing a Department Taxonomy" section of the *HP SOA Systinet Taxonomy Editor Guide*.

To add the department property to business services:

1   Open the Business Service editor and select the **Properties** tab.

2   In the Properties segment, click **New**.

3   Select **Taxonomy Property** and click **Next**.

4   Input `Department` as the **Display Name**.

5   For the **Taxonomy**, click **Browse** and import the **Departments** taxonomy.

6   Click **Next** to set the visibility of the department property.

7   For the Service Catalog UI group, select **Service Properties**, and then click **Finish**.

8   Press **Ctrl+S** to save your changes to the business service artifact type.

9   Deploy your extension, as described in Chapter 8, Deploying Extensions.

10 Start your installation of SOA Systinet.

After you deploy this customization, the service properties section of the service view of a business service includes the new department property.

You can now set this property during business service creation, or by editing a business service.

4 Manipulating Properties

Customization Editor enables you to create, modify, and delete properties and property groups in your extension project.

For more details about properties and property groups, see Properties on page 27 and Property Groups on page 29.

This chapter describes the following procedures:

- Creating a Property on page 61

- Modifying a Property on page 62

- Creating a Property Group on page 63

- Modifying a Property Group on page 63

In addition, Chapter 3, Manipulating Artifact Types includes the following procedures related to properties:

- Modifying the Properties of an Artifact Type on page 47

- Mapping an Artifact Type to a Registry on page 55

Creating a Property

You can create properties individually and then add them to artifact types later.

To create a property:

1   Do one of the following:

- From the menu, select **File→New→Property**.

- In the Extension Explorer, open the context menu for the Property Descriptors branch, select **New Property**, and then select the property type.

- In the Extension Explorer, open the context menu for a property type branch and select **New *Type* Property** to create a property of that type.

The New Property dialog box opens.

2   In the New Property dialog box, add the property parameters you want.

For parameter descriptions of each property type, see the following:

3   Do one of the following:

- Click **Finish** to create the property and end the procedure without setting visibility parameters.

- Click **Next** and continue to Step 4 to set property visibility parameters.

4   In the New Property dialog box, add the visibility parameters you want.

For parameter descriptions, see New Property: Visibility on page 156.

5   Click **Finish** to create the new property.

Modifying a Property

Customization Editor enables you to modify properties.

To edit a property:

1   Open the Property Editor.

2   In the Property Editor, make any changes you want.

For a description of the Property Editor, see Property Editor on page 119.

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

Creating a Property Group

Many artifact types share sets of the same properties. It is useful to create groups of properties and add these to artifact types instead of adding each property individually.

To create a property group:

1   Do one of the following:

- From the menu, select **File→New→Property Group.**

- In the Extension Explorer, open the context menu for the Property Groups branch, and select **New Property Group**.

- Open the Artifact Editor and select the Properties tab. In the Property Groups pane, click **New** to create a new property group as a set of attributes for an artifact type.

The New Property Group dialog box opens.

2   In the New Property Group dialog box, add the parameters you want.

For parameter descriptions, see New Property Group on page 156.

3   Click **Finish** to create the property group.

Modifying a Property Group

Customization Editor enables you to modify property groups.

To edit a property group:

1   Open the property group editor.

2   In the Property Group Editor, make any changes you want.

    For a description of the Property Group Editor, see Property Group Editor on page 123.

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

5 Configuring the User Interface

Customization Editor enables you to create and modify individual UI elements. It also enables you to modify the main SOA Systinet tabs.

This chapter explains the following procedures:

In addition, Chapter 3, Manipulating Artifact Types includes procedures for modifying the appearance of artifact pages.

Adding Portlets to a Tab

You can use Customization Editor to configure which portlets are available in each tab in SOA Systinet and where they appear on the tab.

To add a portlet to an SOA Systinet tab:

1   Open the Portlets editor for the SOA Systinet tab you want to modify.

2   Click **Add** to open the Portlet Placement dialog box.

3   In the Portlet Placement dialog box, add the parameters you want.

    For parameter descriptions, see Portlet Placement on page 160.

4   Click **Finish** to add the portlet to the SOA Systinet tab.

Adding Navigation Groups to a Tab

You can use Customization Editor to configure the menu in each SOA Systinet tab. The items in the menu are organized into sections referred to as navigation groups.

To add a navigation group to an SOA Systinet tab:

1   Open the Navigation editor for the SOA Systinet tab you want to modify.

2   Click **Add** to open the Choose Type dialog box.

3   Select **Group** and click **Next** to open the Menu: Group Details dialog box. For details, see Menu: Group Details on page 134.

4   Add the parameters you want, and click **Finish** to add the group to the SOA Systinet tab menu.

Adding Navigation Actions to a Tab

You can use Customization Editor to configure the menu in each SOA Systinet tab. Most items in the menu correspond to navigation actions linking to another page in SOA Systinet.

To add a navigation item to an SOA Systinet tab:

1 Open the Navigation editor for the SOA Systinet tab you want to modify.

2 Click **Add** to open the Choose Type dialog box.

3 Select **Item** and click **Next**.

  The Menu Item dialog box opens.

4 In the Menu Item the dialog box, add the parameters you want.

  For parameter descriptions, see Menu: Menu Item on page 135.

5 Click **Finish** to add the item to the SOA Systinet tab menu.

Adding Components to a Tab

You can use Customization Editor to configure the menu in each SOA Systinet tab. Some items in the menu are links to a component that performs specific functionality.

To add a component to an SOA Systinet tab:

1 Open the Navigation editor for the SOA Systinet tab you want to modify.

2 Click **Add** to open the Choose Type dialog box.

3 Select **Component** and click **Next**

  The Component Details dialog box opens.

4 In the Component Details dialog box, add the parameters you want.

  For parameter descriptions, see Menu: Component Details on page 135.

5 Click **Finish** to add the component to the SOA Systinet tab menu.

Creating a Navigation Action

Navigation actions are links from the tab menus to pages in SOA Systinet.

To create a navigation action:

1   Do one of the following:

   - In the Extension Explorer, open the context menu for the Navigation Actions branch, and select **New Navigation Action**.

   - In the Navigation Actions editor, click **New**.

   The New Action dialog box opens.

2   In the New Action dialog box, add the parameters you want.

   For parameter descriptions, see .

3   Click **Finish** to create the new action.

Creating a Contextual Action

Contextual actions are links from the context menus in artifact view and browse pages.

To create a contextual action:

1   Do one of the following:

   - In the Extension Explorer, open the context actions for the Contextual Actions branch, and select **New Contextual Action**.

   - In the **Contextual Actions** editor, click **New**.

   The New Action dialog box opens.

2   In the New Action dialog box, add the parameters you want.

   For parameter descriptions, see .

3   Click **Finish** to create the new action.

Creating a Context Action Group

Many artifact types share sets of context actions. It is useful to make groups of actions, and then add these to artifact types.

To create a context action group:

1   Do one of the following:

   • In the Extension Explorer, open the context menu for the Contextual Action Groups branch, and select **New Context Action Group**.

   • In the Contextual Action Groups editor, click **Add**.

   The Contextual Actions dialog box opens.

2   Select **Group of Contextual Actions** and click **Next**.

3   Type the display name for the group and click **Finish** to create the contextual action group.

Adding Context Actions to a Group

Many artifact types share sets of the context actions. It is useful to make groups of actions, and then add these to artifact types.

To add a context action to a group:

1   Do one of the following:

   • In the Extension Explorer, open the context menu for the Contextual Action Groups branch, and select **New Context Action Group**.

   • In the Contextual Action Groups editor, click **New**.

   The Contextual Actions dialog box opens.

2   Select **Action Item** and click **Next**.

The New Contextual Action Item dialog box opens.

3   In the New Contextual Action Item dialog box, add the parameters you want.

For parameter descriptions, see .

4   Click **Finish** to add the action to the group.

Creating a Perspective

Perspectives control the views of SOA Systinet that are available to each user.

To create a perspective:

1   Do one of the following:

- In the Extension Explorer, open the context menu for the Perspectives branch and select **New Perspective**.

- In the Perspectives Editor, click **New**.

The New Perspective dialog box opens.

2   In the New Perspective dialog box, add the parameters you want.

For parameter descriptions, see .

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

Creating a Portlet

Portlets are components in the main section of SOA Systinet tabs. The functionality of these components depends on the portlet.

To create a portlet:

1   In the Extension Explorer, double-click the Portlets Definitions branch to open the Portlets Definitions editor.

2   Click **Add** and select **Portlet** to open the Add Portlet dialog box.

3   Type a name and URL for the portlet, and click **Finish** to create the portlet.

Creating an RSS Feed Portlet

The RSS feed portlets in the main section of SOA Systinet tabs display up-to-date views of repository data or links to external RSS feeds.

To create an RSS Feed Portlet:

1   In the Extension Explorer, double-click the Portlets Definitions branch to open the Portlets Definitions editor.

2   Click **New** and select **Feed** to open the Add Feed dialog box.

3   In the Add Feed dialog box, add the parameters you want.

   For parameter descriptions, see Portlet Definitions: Add Feed on page 159.

4   Click **Finish** to create the feed.

6 Creating and Using Components

Customization Editor enables you to create custom components, deploy them directly to the SOA Systinet UI, and to debug your code.

To create and use components:

1 Create a new extension project, selecting **Develop JSPs**, and setting the appropriate application server settings.

   For details, see as described in Creating an Extension Project on page 40.

2 Create a component.

   For details, see Creating a Component on page 74.

3 Create a task to use your component.

   For details, see Creating a Task on page 74.

4 Develop the Java and JSP that the component uses.

   For details, see Developing a Component on page 75.

5 Debug your code in Customization Editor.

   For details, see Debugging Your Code on page 76.

Example: Adding a Component to the Tools Menu on page 79 is a step-by-step walkthrough of this process.

Generic Association Control on page 76 describes how to use a component provided with SOA Systinet to display artifacts related to a business service in the Services UI.

Creating a Component

Components are functional elements that you can call using a number of different elements in the SOA Systinet UI.

To create a component:

1   Do one of the following:

    • In the Extension Explorer, open the context menu for the Components branch and select **New Component**.

    • In the Components Editor, click **Add**.

    The New Component dialog box opens.

2   In the New Component dialog box, add the parameters you want.

    For parameter definitions, see New Component on page 142.

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

Creating a Task

Tasks are top level components accessible via their URIs. The addition of a URI allows you to access the component in the SOA Systinet UI, and to make more sophisticated multi-layered components.

To create a task:

1   Do one of the following:

    • In the Extension Explorer, open the context menu for the Tasks branch and select **New Task**.

    • In the Tasks editor, click **Add**.

    The New Task dialog box opens.

2   In the New Task dialog box, add the parameters you want.

For parameter descriptions, see .

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

Developing a Component

When you create a component, as described in Creating a Component on page 74, the templates for the Java class and JSP page are created automatically.

To view this code, switch to the Java or Debug perspective, as shown in Figure 9, "Debug Perspective".

Figure 9. Debug Perspective
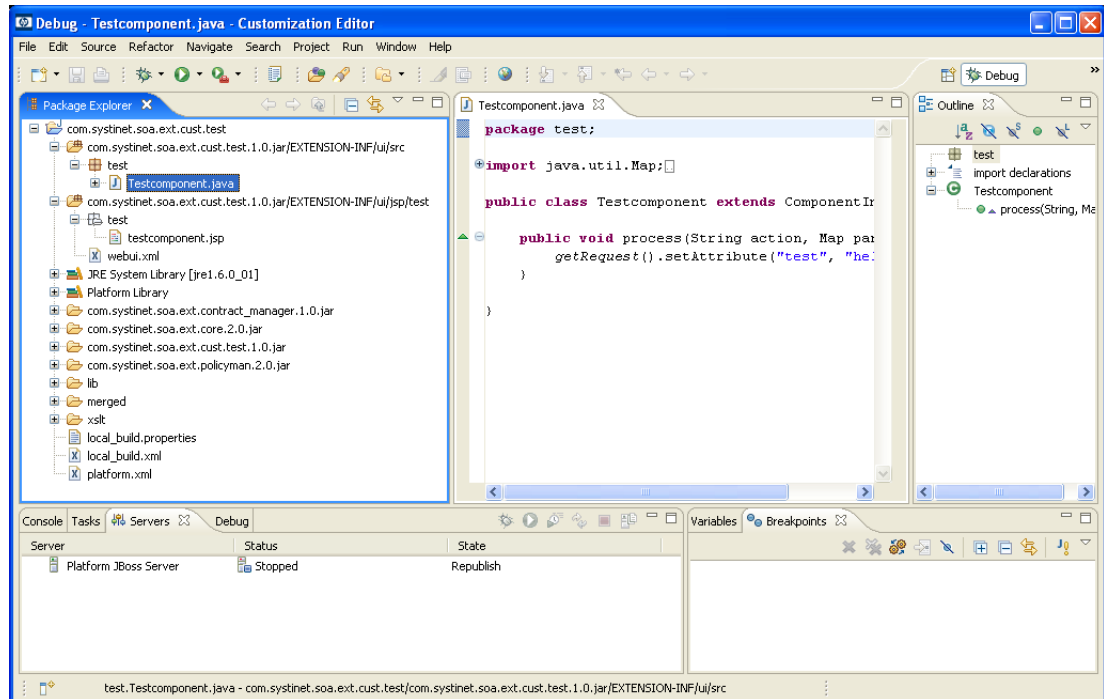
The project contains two source folders containing the Java and JSP files, respectively.

To open an edit view, double-click the source file you want to edit. In the edit view, you can write a custom component.

Debugging Your Code

Only changes to the UI or the SDM configuration require the extension to be deployed to SOA Systinet. After you deploy a navigation action that uses your custom component, you can edit your Java and JSP. The new functionality is applied to SOA Systinet directly.

To start SOA Systinet from Customization Editor:

1   From the main menu, select **Run→Debug**.

    The Debug dialog box opens.

2   Expand Generic Servers and select your server deployment.

3   Click **Debug** to start the server in debug mode.

The Console view displays the progress of the server start.

When you start the server, the Server view displays your server with the status `Debugging`.

Make changes to your Java and JSP, as required. Customization Editor applies these changes to SOA Systinet automatically. You do not need to reapply the extension or restart the server.

Generic Association Control

SOA Systinet includes a component that enables you to display artifacts related to a business service in the same way that implementations and documentation are displayed in the Services UI.

Functionality for documents is shown in Figure 10, "Document Association Control".

Figure 10. Document Association Control

To use generic association control:

1 Add a group to the View Artifact page using the name of the related artifact type (for example, Documentation).

2 Add a component to the View Artifact page with the following options:

   - Use `/sm/references/genericAssociationComponent` as the component.

   - Use the newly created group as the layout group.

   - Click **Add** to add a component parameter, select **relationshipType** as the key, and then input the name of the relationship from the business service to your artifact type.

   For the full procedure, see Adding a Component to View Artifact Pages on page 52.

If the generic association component does not provide the functionality you need, you can create and use a new component.

For details about customized association control, see the following sections:

Customizing the Table

If you need to modify the columns in the table, you must create a new component with a new table definition.

For examples, see `sm/references/documentationTableDefinition` and `/sm/references/implementationTableDefinition`.

If you base your component on `com.systinet.contractmgr.ui.references.GenericTableDefinition`, you can use the SDM property names and descriptions through a list of property descriptors of the artifacts that can be listed. You can access the values using `propertyDescriptors['name'].label` or `propertyDescriptors['name'].description`.

Customizing the Add Section

The default behavior displays one drop-down option, Add Artifact, containing the New Artifact and From Catalog options.

If you need to change the drop-down lists for particular artifacts, create a new component for the Add Section and one component for each drop-down list.

For details, see `/sm/references/implementationAddSection` where a custom component (`/sm/references/webServiceAddArtifact`) is used for the SOAP Service option, and the drop-down lists are ordered.

If this approach is not sufficient (for example, you need more than one drop-down list for an artifact type), you can create a customized Add Section.

For an example of this approach, see `/sm/onPlaceDocSelect`.

Customizing the Action Links

The default behavior displays Detail, Edit, and Remove actions.

If you want to change the actions or add new ones, see /sm/references/documentationActionRenderer where the edit action is replaced with a custom one.

Change it by passing the generic component the name of the custom action component under one of the customDetailComponent, customEditComponent, or customRemoveComponent parameter keys.

Example: Adding a Component to the Tools Menu

In this example, you add a new component to the Tools menu in the SOA Systinet UI, as shown in Figure 11, "Default Tools Menu View Section".

Figure 11. Default Tools Menu View Section
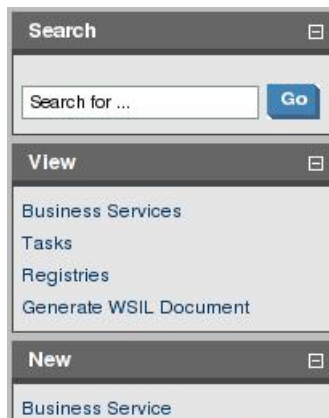
To follow this example, you must first do the following:

• Create a new extension dependent on the core extension with **Develop JSPs** selected.

For the full procedure, see Creating an Extension Project on page 40.

To create the new component and add it to the tools menu:

1   Create a component using the following parameters:

Parameter	Input
Name	testComponent
Page	/test/testComponent.jsp
Class name	test.TestComponent

For the full procedure, see Creating a Component on page 74.

2   Create a task using the following parameters:

Parameter	Input
Uri	/test/testUri
Component	testComponent
Caption	testTask
Context	/index/repository
Wrap	Selected
Requires Authentication	[Select if you want the user to sign-in to use the component]

For the full procedure, see Creating a Task on page 74.

3   Create a navigation action using the following parameters:

Parameter	Input
Action ID	testNavAction
Task Location	/test/testUri

For the full procedure, see Creating a Navigation Action on page 68.

4   Add the navigation action to the Tools tab using the following parameters:

Parameter	Input
Item Name	testNavItem
Action ID	testNavAction

Parameter	Input
Group Name	View

For the full procedure, see Adding Navigation Actions to a Tab on page 66.

5 Create your Java class and JSP using the following code:

Editor	Input
TestComponent.java	public void process(String arg0, Map arg1) { getRequest().setAttribute("test", "created and deployed your test component."); }
testComponent.jsp	You have succesfully <c:out value="${test}"/>

For details, see Developing a Component on page 75.

6 Apply the extension.

For details, see Deploying Components to SOA Systinet on page 91.

7 Start the SOA Systinet server to test the component.

For details, see Debugging Your Code on page 76.

After you deploy this customization, the Tools menu View section includes the new navigation action, as shown in Figure 12, "Customized Tools Menu View Section".

Figure 12. Customized Tools Menu View Section
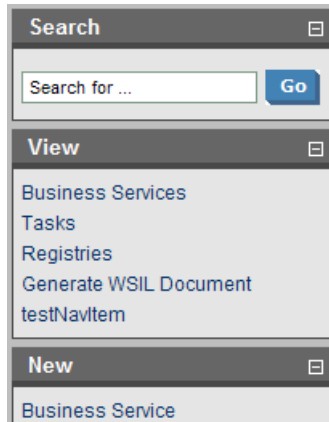
Click **testNavItem** to execute the task.

7 Modifying Taxonomies

Customization Editor controls which taxonomies are available in the extension project. You can create and modify taxonomies from your extension project.

Creating and modifying taxonomies uses the Taxonomy Editor functionality.

For details, see the *HP SOA Systinet Taxonomy Editor Guide*.

To control taxonomies in your extension project:

1   In the Extension Explorer, double-click the Taxonomies branch to open the Taxonomies editor.

2   To modify the available taxonomies in your extension project, do one of the following:

   • Open the taxonomy from the extension project, and edit the parameters as required.

   The modified taxonomy is then indicated by a **>** next to the taxonomy.

   • To import taxonomies from SOA Systinet and add them to the extension, click **Import**. Select multiple taxonomies by holding **Ctrl**.

   You can also import taxonomies in the Server Explorer view. For more details, see Server Explorer on page 24.

   • To update taxonomies in your extension to newer versions from SOA Systinet, click **Update**. Select multiple taxonomies by holding **Ctrl**.

   Alternatively, in the Extension Explorer, expand the branches in the Taxonomy branch, select the taxonomy (use **Ctrl** to select multiple taxonomies), open the context menu, and select **Update Taxonomy**.

   The selected taxonomies are added as separate items for modification under the Taxonomy branch.

- To remove a taxonomy from the extension, select the taxonomy and click **Remove**. Select multiple taxonomies by holding **Ctrl**.

> To enable the import or update of taxonomies from SOA Systinet, you must specify the correct server URL during configuration.
>
> For details, see .

You can change the taxonomy in a taxonomy based property descriptor which is used in the current extension project.

To change the taxonomy in a taxonomy based property descriptor:

1. In the tree menu of Extension Explorer, expand **SDM Details**→**Taxonomy Property Descriptors** and double-click the descriptor you need to open its Editor in the General view.

2. In the **Taxonomy Name** field, click **Browse**.

3. Select the taxonomy you want, and click **Select**.

   The new taxonomy overrides the previous taxonomy in the property descriptor.

> You cannot change the taxonomy in a taxonomy based property descriptor if the extension project is in production mode.

8 Deploying Extensions

Deploying an extension to SOA Systinet is a three-part process:

1 Exporting the Extension Project on page 85

2 Applying Extensions on page 86

3 Redeploying the EAR File on page 90

Alternatively, if you are developing components in Customization Editor, you can directly deploy an extension from Customization Editor.

For details, see Deploying Components to SOA Systinet on page 91.

Exporting the Extension Project

Customization Editor displays the entire configuration of your extension plus other extensions it depends on. However, your extension consists only of any modifications and additions that you have made. Only these changes are stored when you save your project as an extension JAR file, and only these changes are deployed to SOA Systinet.

To create your extension package:

1 In the Extension Explorer, open the context menu of the extension name, and select **Build Extension**.

   The Export Extension dialog box opens.

2 Choose a save location and file name, and then click **Save**.

   By default, the Extension Folder set during configuration is selected.

   For configuration details, see Creating an Extension Project on page 40.

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:

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

  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.

> In some specific circumstances (underscores and numbers in property names), extension application may fail because SOA Systinet cannot create short enough database table names (31 character maximum for most databases).
>
> The error in setup.log resembles the following:

```
[java] --- Nested Exception ---
[java] java.lang.RuntimeException: cannot reduce length of identifier
  'ry_c_es_Artifact02s_c_priEspPty01Group_c_priEspPty01',
  rename identifier elements or improve the squeezing algorithm
[java] at com.systinet.platform.rdbms.design.decomposition.naming.impl.
  BlizzardNameProviderImpl.getUniqueLimitedLengthName(BlizzardNameProviderImpl.java:432)
[java] at com.systinet.platform.rdbms.design.decomposition.naming.impl.
  BlizzardNameProviderImpl.filterTableName(BlizzardNameProviderImpl.java:374)
```

This is due to SOA Systinet using an older table naming algorithm in order to preserve backward compatibility with SOA Systinet 3.00 and older versions.

If you do not require backwards compatibility with these older versions, you can change the table naming algorithm.

To change the table naming algorithm:

1 Open `SOA_HOME/lib/pl-repository-old.jar#META-INF/rdbPlatformContext.xml` with a text editor.

2 In the `rdb-nameProvider` bean element, edit the following property element:

   <property name="platform250Compatible" value="false"/>

3 Save `rdbPlatformContext.xml`

This solution only impacts properties with multiple cardinality. If the problem persists or you need to preserve backwards compatibility, then review the property naming conventions in your extension.

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 your 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 .

- **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.

⚠️   Applying an extension that modifies the SDM model may drop your full text indices.

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. ... .

In these cases, reapply full text indices as described in the "Enabling Full Text Search" section of the *HP SOA Systinet Installation and Deployment Guide*.

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 90.

     • **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.

Deploying Components to SOA Systinet

If you are creating components in Customization Editor, it is not convenient to continually build and apply extensions to SOA Systinet. Customization Editor enables you to deploy your extension directly to SOA Systinet.

Only SOA Systinet deployed to a JBoss application server is supported by this release.

To directly deploy an extension from Customization Editor:

1   Stop the application server.

2   Switch to the Debug perspective in Customization Editor.

3   From the main menu, select **Run→External Tools→Open External Tools Dialog**.

    The External Tools dialog box opens.

4   Select **Ant Build→Customization Editor – Apply Extension**.

5   Click **Run** to apply the extension.

The Console view displays the output of the deployment process.

A Views

Each Customization Editor view in the main and bottom-right section of the editor is described in the following sections:

- Customize the taxonomies available in SOA Systinet.

Artifact Editor

Artifacts are the basic building blocks of SOA and represent all types of entity. This editor enables you to modify the parameters, properties, and appearance of an artifact type within SOA Systinet. Each artifact type and package has an editor with the artifact name as the title.

Each editor contains five tabs described in the following sections:

-

  The main attributes of the artifact type.

-

  The properties of the artifact type.

-

  The format of the View Artifact page in the SOA Systinet UI.

-

  The format of the View Artifact and Browse Artifact pages in the SOA Systinet UI.

-

  The mapping of the artifact to registry entities.

Artifact Editor: General Tab

The **General** tab displays the general attributes of the artifact and allows you to modify some of them.

The General tab for the Business Service artifact type is shown in .

Figure 13. Business Service Editor: General Tab

The general tab contains the following segments:

- **Artifact General Information**. The following parameters related to artifacts shown in this pane. They are described in the table below:

Parameter	Definition
URI	The identifier for the artifact descriptor in the configuration.
Display Name	The name of the artifact as it appears in the SOA Systinet UI.

Parameter	Definition
Local Name	The name of the artifact as it is stored in the extension.
Collection Display Name	The plural version of the name as it appears in the SOA Systinet UI.
Package	The parent artifact type that this artifact type belongs to.
Extends Type	The artifact type that this artifact type inherits the properties of.
Description	The description of the artifact type as it appears in the SOA Systinet UI.
Deprecated	A check box indicating whether the artifact is currently active in the extension.
Abstract	A check box indicating whether the artifact is an artifact package which does not have actual instances in the repository.

- **Visible to**

  A simple list of perspectives with **Add** and **Remove** functionality to select which perspectives can see this artifact type within SOA Systinet.

- **Editable by**

  A simple list of perspectives with **Add** and **Remove** functionality to select which perspectives can amend this artifact type within SOA Systinet.

- **Properties**

  A summary of the different properties that the artifact has. Each of the links opens the **Properties** tab described in Artifact Editor: Properties Tab on page 95.

- **User Interface**

  A pair of links to the **Service Catalog UI** and **Tools UI** tabs where the appearance of the artifact in the SOA Systinet UI can be amended.

Artifact Editor: Properties Tab

The **Properties** tab displays all the properties of the artifact and enables you to add and remove them:

Figure 14. Business Service Editor: Properties Tab
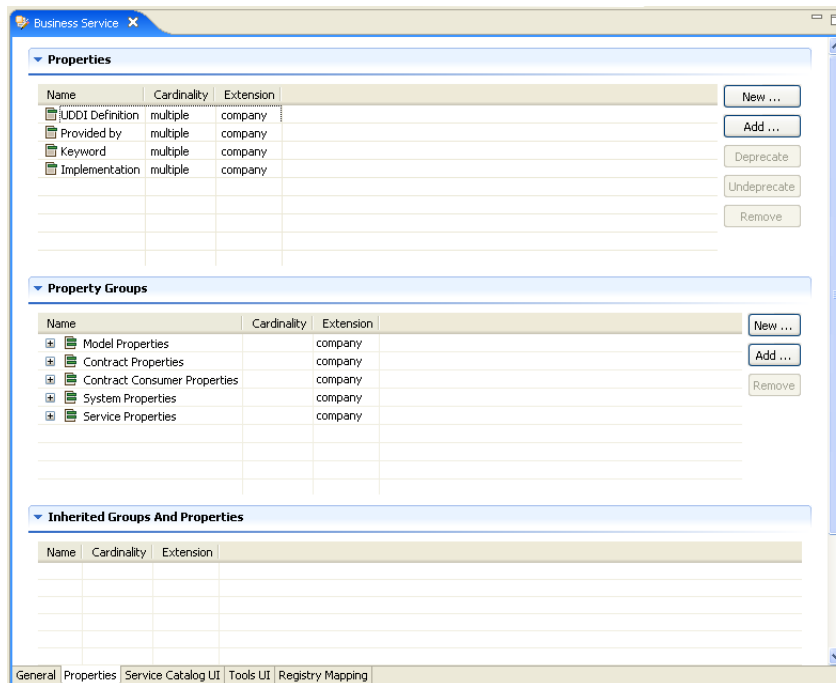
This tab contains the following panes:

- **Properties**

  This is the list of individual properties of the artifact. On the right are the following buttons:

  - **New** opens the **New Property** dialog described in Creating a Property on page 61.

  - **Add** adds a property to the artifact type from the list of available property descriptors, set its cardinality, and its location on the **View Artifact** page.

  - **Deprecate** deactivates the property in the artifact type.

  - **Undeprecate** reactivates the property in the artifact type.

- **Remove** removes the selected property from the artifact type.

- **Property Groups**

  This is the list of the property groups of the artifact. On the right are the following buttons:

  - **New** opens the **New Property Group** dialog. For details, see Creating a Property Group on page 63.

  - **Add** adds a property group to the artifact type from the list of available property groups.

  - **Remove** removes the selected property group from the artifact type.

- **Inherited Groups and Properties**

  If the artifact is based on another artifact then this segment displays the inherited artifact and its properties and groups.

The following property parameters are displayed in these panes:

Parameter	Definition
Name	The name of the property as it appears in the SOA Systinet UI.

Cardinality	The occurrence of the property in an artifact with options:
	
	• **Optional**
	
	  The property is not required to be populated.
	
	• **Required**
	
	  The property must be populated.
	
	• **Multiple**
	
	  The cardinality is determined by the MinOccurs/MaxOccurs defined in the Property Editor: Advanced Tab. For details, see Property Editor: Advanced Tab on page 122.

Extension	Which extension this property is part of.

Artifact Editor: Service Catalog UI Tab

The **Service Catalog UI** tab displays all the attributes of the artifact as they appear in the **Service View** page of the Service Catalog in SOA Systinet and allows them to be amended:

Figure 15. Business Service Editor: Service Catalog UI Tab
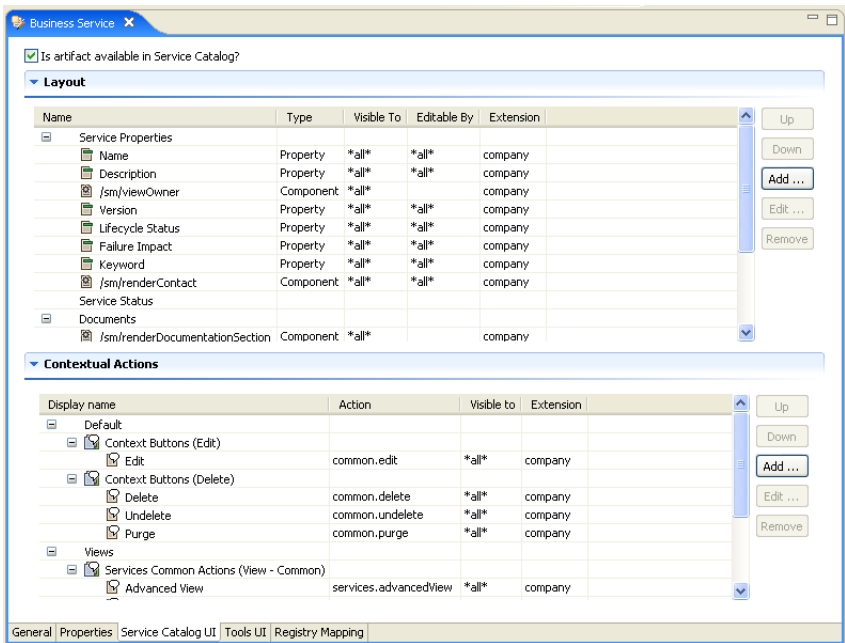
There is a check box to control whether the artifact is available within the Service Catalog and if this is checked there are two panes you can use to control its appearance:

• **Layout**

  This is the list of elements and properties of the artifact as they appear in the **Service View** page in the Service Catalog. The elements are organized into groups that match the sub-headings on the **Service View** page. On the right are the following buttons:

  • **Up** moves the element up in the **Service View** page.

- **Down** moves the element down in the **Service View** page.

- **Add** adds a property, component, or layout group to the **Service View** page, as described in Modifying the Layout of View Artifact Pages on page 51.

- **Edit** enables you to edit the name of a sub-heading group or the editable by and visible to attributes of a property.

- **Remove** removes the element from the **Service View** page.

The following page element parameters are displayed in this pane:

Parameter	Definition
Name	The name of the page element as in appears in the SOA Systinet UI.
Type	The type of page element with options: • **Property** is a property of the artifact type. • **Component** is an SOA Systinet UI component. • **Relationship** is a relationship property of the artifact type.
Visible to	The perspectives that can see this page element.
Editable by	The perspectives that can edit this page element in SOA Systinet.
Extension	Which extension this property is part of.

- **Contextual Actions**. These are the options that appear in the context action menus at the top of the **Service View** page. On the right are the following buttons:

  - **Up** moves the action up in the list.

  - **Down** moves the action down in the list.

  - **Add** adds a new context action to this artifact as described in Modifying the Contextual Actions of an Artifact on page 53.

  - **Edit** enables you to edit the context action with the same parameters as **Add** functionality.

• **Remove** removes the action from context action menus.

The following contextual action parameters related to properties are shown in this pane:

Parameter	Definition
Display Name	The name of the action as it appears in the SOA Systinet UI.
Action	Contextual action identification in the extension.
Visible to	The perspectives that can see this action.
Extension	Which extension this property is part of.

Artifact Editor: Tools UI Tab

The **Tools UI** tab displays all the attributes of the artifact as they appear in the **View Artifact** and **Browse Artifact** pages in Tools in SOA Systinet and allows them to be amended:

Figure 16. Business Service Editor: Tools UI Tab
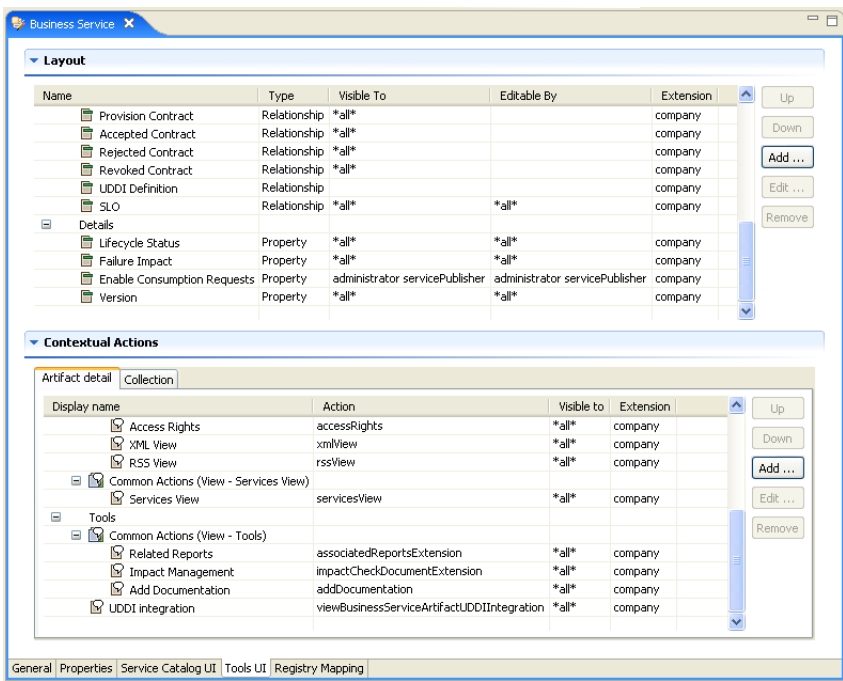
The following panes control its appearance:

- **Layout**

  This is the list of elements and properties of the artifact as they appear in the View Artifact page in Tools. The elements are organized into groups that match the sub-headings on the View Artifact page. On the right are the following buttons:

  - **Up** moves the element up in the **View Artifact** page.

  - **Down** moves the element down in the **View Artifact** page.

  - **Add** adds a property, component or layout group to the **View Artifact** page, as described in Modifying the Layout of View Artifact Pages on page 51.

- **Edit** enables you to edit the name of a sub-heading group or the editable by and visible to attributes of a property.

- **Remove** removes the element from the **View Artifact** page.

The following page element parameters are displayed in this pane:

Parameter	Definition
Name	The name of the page element as in appears in the SOA Systinet UI.
Type	The type of page element with options: • **Property** is a property of the artifact type. • **Component** is an SOA Systinet UI component. • **Relationship** is a relationship property of the artifact type.
Visible to	The perspectives that can see this page element.
Editable by	The perspectives that can edit this page element in SOA Systinet.
Extension	Which extension this page element is part of.

- **Contextual Actions**. These are the options that appear in context actions at the top of artifact pages. In the **Tools UI** tab there are two tabs in this pane. **Artifact Detail** shows the actions for the **View Artifact** page and **Collection** shows the actions for the **Browse Artifact** page. On the right are the following buttons:

  - **Up** moves the action up in the list.

  - **Down** moves the action down in the list.

  - **Add** adds a new context action to this artifact, as described in Modifying the Contextual Actions of an Artifact on page 53.

  - **Edit** enables you to edit the context action with the same parameters as the **Add** functionality.

  - **Remove** removes the action from context action menus.

The following contextual action parameters are displayed in this pane:

Parameter	Definition
Display Name	The name of the action as it appears in the SOA Systinet UI.
Action	Contextual action identification in the extension.
Visible to	The perspectives that can see this action.
Extension	Which extension this context action is part of.

Artifact Editor: Registry Mapping Tab

The **Registry Mapping** tab displays any mappings between the artifact and entities in registries and allows you to create, edit, and remove them:

Figure 17. Business Service: Registry Mapping Tab
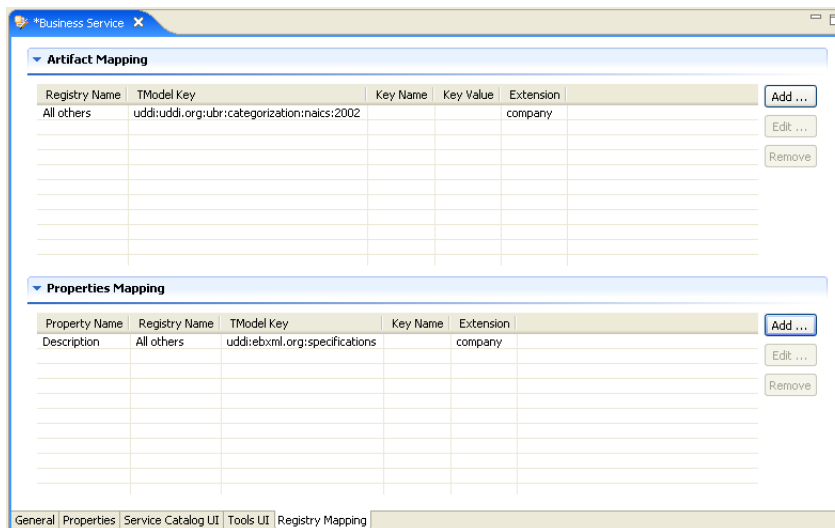
It is split into the following panes:

- **Artifact Mapping**

This is the list of registry entities that represent the artifact in registries. On the right are the following buttons:

- **Add** adds a registry mapping, as described in Mapping an Artifact Type to a Registry on page 55.

- **Edit** edits the mapping, as described in Mapping an Artifact Type to a Registry on page 55.

- **Remove** removes the mapping from the artifact.

The following registry mapping parameters are displayed in this pane:

Parameter	Definition
Registry Name	The registry that the artifact type is mapped to.
TModel Key	The taxonomy key used to categorize the UDDI entity in the registry.
Key Name	The name used to categorize the UDDI entity in the registry.
Key Value	The value used to categorize the UDDI entity in the registry.
Extension	Which extension this mapping is part of.

- **Property Mappings**

This is the list of the properties of the artifact and if they are mapped to a registry. On the right are the following buttons:

- **Add** adds a new registry mapping for the selected property, as described in Mapping an Artifact Type to a Registry on page 55.

- **Edit** edits a property mapping, as described in Mapping an Artifact Type to a Registry on page 55.

- **Remove** removes a mapping from the selected property.

The following registry mapping parameters are displayed in this pane:

Parameter	Definition
Property Name	The name of the property.
Registry Name	The registry that the artifact type property is mapped to.

Parameter	Definition
TModel Key	The taxonomy key used to categorize the UDDI entity in the registry.
Key Name	The name used to categorize the UDDI entity in the registry.
Extension	Which extension this property is part of.

Components Editor

Components are functional elements which can be added to the SOA Systinet UI.

Figure 18. Components Editor
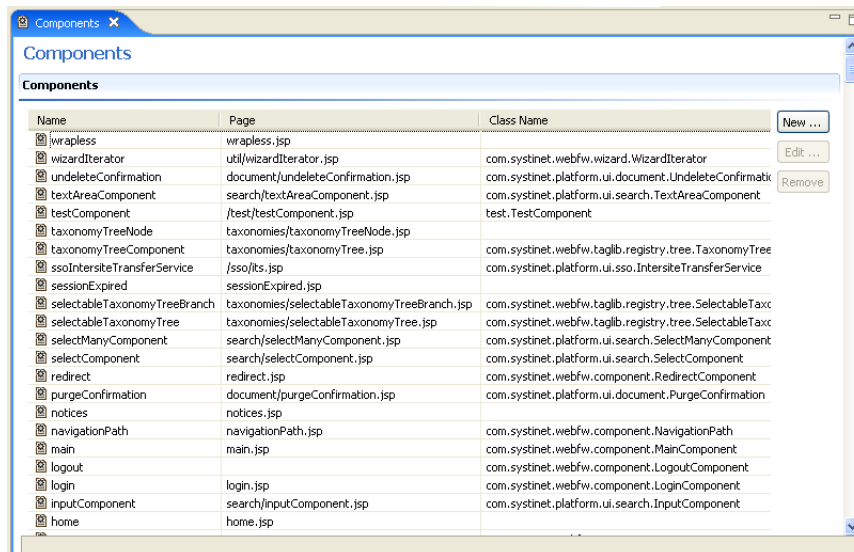
On the right are the following buttons:

- **New** opens the **New Component** dialog, as described in Creating a Component on page 74.

- **Edit** opens the **Edit Component** dialog with the same parameters described in Creating a Component on page 74.

- **Remove** deletes the selected component from your extension project.

The following component parameters are displayed:

Parameter	Definition
Name	The name of the component as it is stored in the extension.
Page	The JSP that the component uses.
Class Name	The Java class the component uses.
Extension	Which extension this component is part of.

Contextual Actions Editor

Context actions are options available in the SOA Systinet UI for artifact and service pages. The contextual actions editor allows you to create, edit, and remove context actions from your extension project:

Figure 19. Contextual Actions Editor
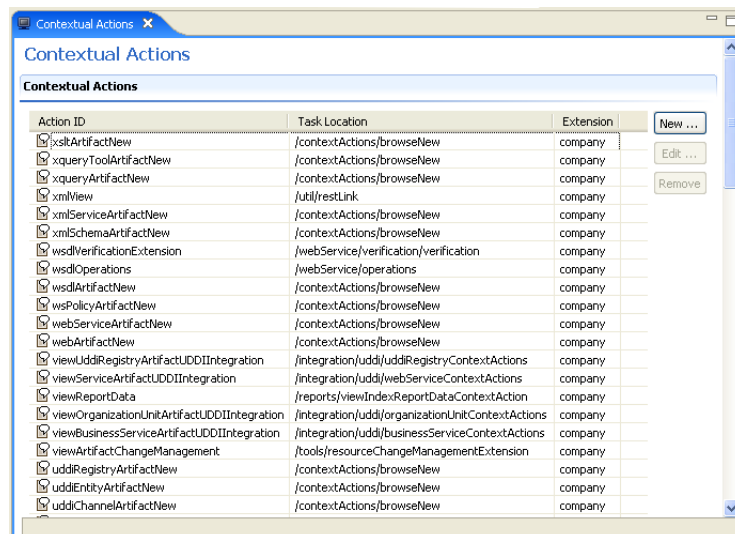
On the right are the following buttons:

- **New** opens the **New Action** dialog, as described in Creating a Contextual Action on page 68.

- **Edit** opens the **Edit Action** dialog with the same parameters described in Creating a Contextual Action on page 68.

- **Remove** deletes the selected context action from your extension project.

The following contextual action parameters are displayed:

Parameter	Definition
Action ID	The identification of the contextual action as it is stored in the extension.
Task Location	The identification of SOA Systinet UI task invoked by this contextual action.
Extension	Which extension this perspective is part of.

Contextual Action Groups Editor

Context actions can be grouped in order to add many actions to a UI element at once. The contextual action groups editor allows you to create, edit, and remove context action groups from your extension project:

Figure 20. Contextual Action Groups Editor
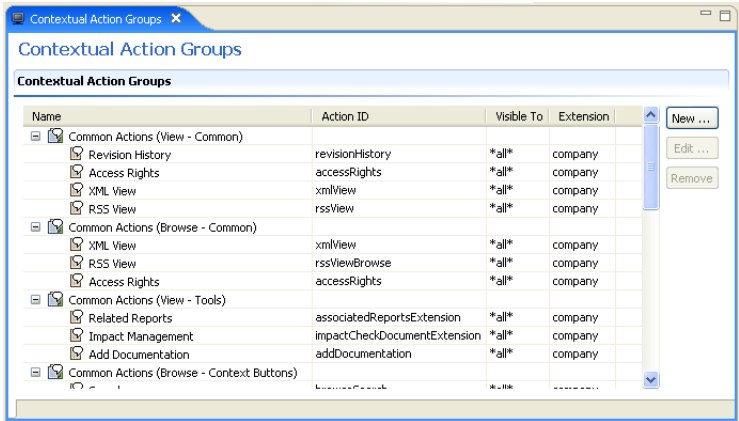
On the right are the following buttons:

- **Add** contains the following options:

  - **Action Item** adds a context action to a group, as described in Adding Context Actions to a Group on page 69.

  - **Group of Contextual Actions** adds a new context action group, as described in Creating a Context Action Group on page 69.

- **Edit** has the following functions depending on which item in the editor is highlighted:

  - For a group you can edit the name.

  - For an item the **Edit Item** dialog with the same parameters, as described in Creating a Contextual Action on page 68.

- **Remove** deletes the selected context action item from your extension project.

The following contextual action group parameters are displayed:

Parameter	Definition
Name	The name of the action group as it is stored in the extension.
Action	The reference to the contextual action in the extension.
Visible to	Which perspectives can see this item.
Extension	Which extension this perspective is part of.

Messages View

The **Messages** view tracks the changes that you make to extension entities and displays any warnings and problems that may occur as a result of those actions:

Figure 21. Messages View
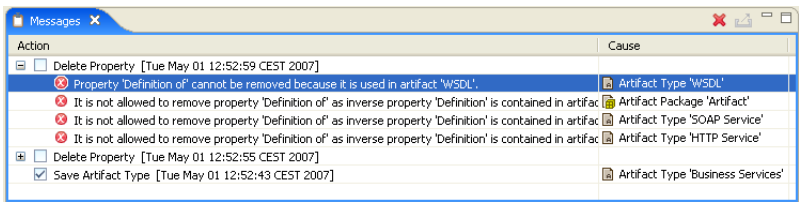
Double-clicking an item in the tree opens the relevant editor for that object.

Navigation Editor

Navigation actions are the links between pages in SOA Systinet. The navigation editor allows you to configure the left menu and title of SOA Systinet tabs:

Figure 22. Dashboard Navigation Editor
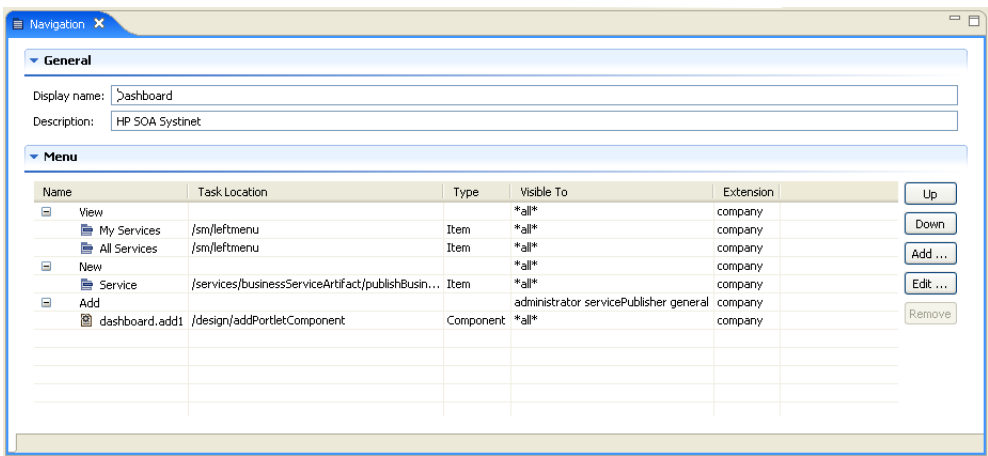
The editor contains the following collapsible panes:

• **General** enables you to change the **Display Name** and **Description** of the SOA Systinet tab.

- **Menu** enables you to configure the left menu of the SOA Systinet tab. The navigation items are arranged into headings and links.

  On the right are the following buttons:

  - **Up** moves the selected item up in the menu.

  - **Down** moves the selected item down in the menu.

  - **Add** contains the following options:

    - **Item** adds a navigation item to the menu, as described in Adding Navigation Actions to a Tab on page 66.

    - **Component** adds a component to the menu, as described in Adding Components to a Tab on page 67.

    - **Group** adds a new heading to the menu, as described in Adding Navigation Groups to a Tab on page 66.

  - **Edit** opens one of the following edit dialogs depending on the item selected:

    - **Item** edits a navigation item with the same parameters described in Adding Navigation Actions to a Tab on page 66.

    - **Component** edits a component with the same parameters described in Adding Components to a Tab on page 67.

    - **Group** edits a menu heading with the same parameters described in Adding Navigation Groups to a Tab on page 66.

  - **Remove** removes the selected navigation item from the SOA Systinet tab menu.

  This pane contains the following navigation action parameters:

Parameter	Definition
Name	The name of the navigation item as it is stored in the extension.
Task Location	The identification of SOA Systinet UI task invoked by this menu item.

Parameter	Definition
Type	Item or Component.
Visible to	Which perspectives can see this item.
Extension	Which extension this navigation item is part of.

Navigation Actions Editor

Navigation actions are the links available in the menus of SOA Systinet tabs. The navigation actions editor allows you to create, edit, and remove navigation actions from your extension project:

Figure 23. Navigation Actions Editor
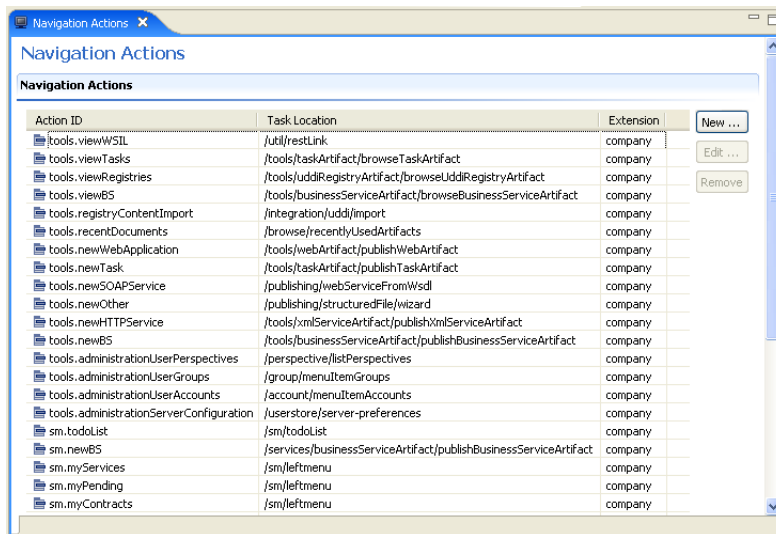
On the right are the following buttons:

- **New** opens the **New Action** dialog, as described in .

- **Edit** opens the **Edit Action** dialog with the same parameters described in .

• **Remove** deletes the selected navigation action from your extension project.

The following navigation action parameters are displayed:

Parameter	Definition
Action ID	The identification of the navigation action as it is stored in the extension.
Task Location	The identification of SOA Systinet UI task invoked by this menu item.
Extension	Which extension this perspective is part of.

Perspectives Editor

Perspectives are the way SOA Systinet controls what users can view. The perspectives editor allows you to create, edit, and remove perspectives from your extension project:

Figure 24. Perspectives Editor
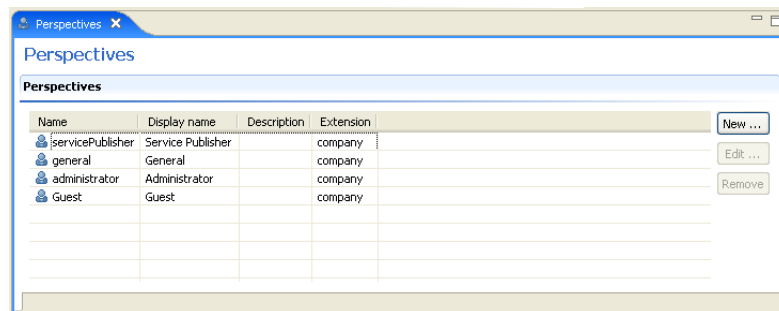
On the right are the following buttons:

• **New** opens the **New Perspective** dialog box, as described in Creating a Perspective on page 70.

• **Edit** opens the **Edit Perspective** dialog box for the selected perspective with the same parameters described in Creating a Perspective on page 70.

• **Remove** deletes the perspective from your extension project.

The following perspective parameters are displayed:

Parameter	Definition
Name	The name of the perspective as it is stored in the extension.
Display Name	The name of the perspective as it appears in the SOA Systinet UI.
Description	The description of the perspective as it appears in the SOA Systinet UI.
Extension	Which extension this perspective is part of.

Portlets Editor

There is a separate portlets editor for each tab within SOA Systinet. These allow you to control which portlets are available in each tab and where they appear.

Figure 25. Dashboard Portlets Editor
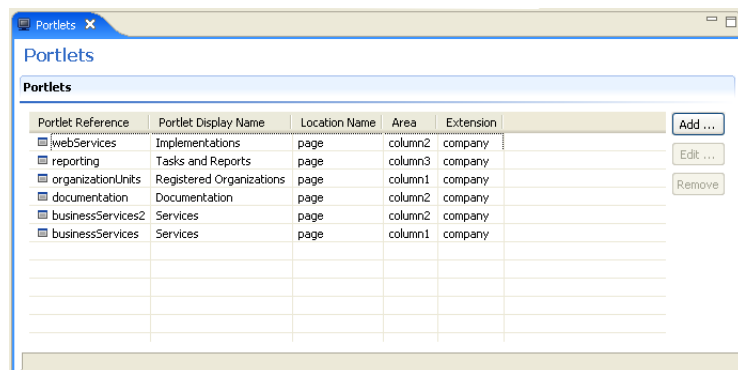
On the right are the following buttons:

- **Add** opens the **New Portlet Placement** dialog, as described in Adding Portlets to a Tab on page 66.

- **Edit** opens the **Edit Portlet Placement** dialog for the selected portlet with the same parameters described in Adding Portlets to a Tab on page 66.

- **Remove** removes the selected portlet from the SOA Systinet tab.

The following portlet placement parameters are displayed:

Parameter	Definition
Portlet Reference	The identification of the portlet in the extension.
Portlet Display Name	The name of the portlet as it appears on the SOA Systinet tab.
Location Name	There are two options: • **page** 　The portlet is open in the main area of the SOA Systinet tab. • **menu** 　The portlet is closed but available to add in the left menu.
Area	If the portlet is open this specifies which column it appears in on the SOA Systinet tab.
Extension	Which extension this portlet is part of.

Portlets Definitions Editor

Portlets are the components that appear in the main area of SOA Systinet tabs. The portlets definitions editor allows you to add, edit, and remove portlet definitions from your extension project:

Figure 26. Portlets Definitions Editor
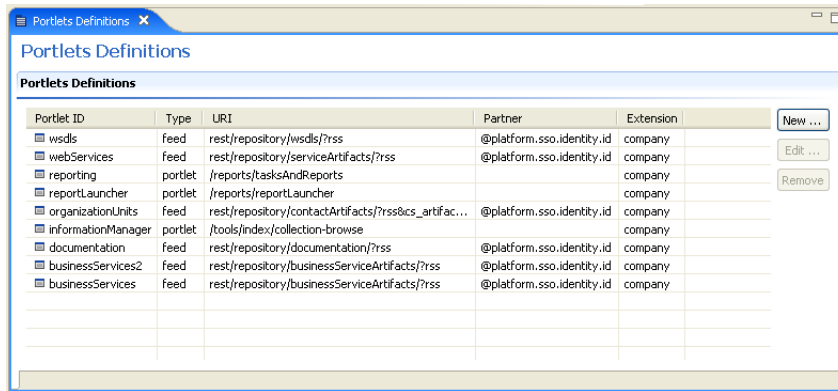
On the right are the following buttons:

- **Add** contains the following options:

  - **Feed** adds a new RSS feed portlet, as described in Creating an RSS Feed Portlet on page 71.

  - **Portlet** adds a new portlet, as described in Creating a Portlet on page 70.

- **Edit** opens the **Edit Feed** or **Edit Portlet** dialog box for the selected item with the same parameters as the **Add** functionality.

- **Remove** deletes the selected portlet from your extension project.

The following portlet parameters are displayed and described in the table below:

Parameter	Definition
Name	The name of the portlet as it is stored in the extension.
Type	The type of portlet with options: • Feed • Portlet

Parameter	Definition
URI	The identifier for the portlet descriptor in the configuration.
Partner	The SSO partner identification.
Extension	Which extension this perspective is part of.

project.xml Editor

The project editor allows you to configure your extension project. It contains three tabs described in the following sections:

Project Editor: Overview Tab

The **Overview** tab enables you to view and amend the basic parameters of your extension project:

Figure 27. Project Editor: Overview Tab
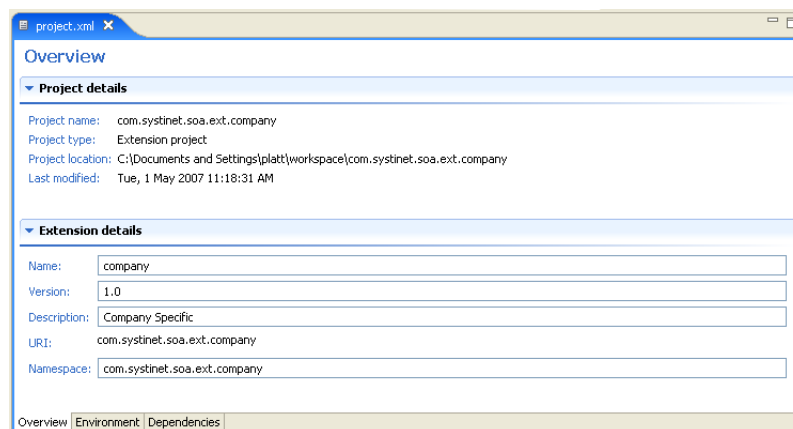
The **Overview** tab contains the following collapsible panes:

- **Project details** displays parameters for the extension project:

Parameter	Definition
Project name	The name of the extension project.
Project type	The project is an extension project.
Project location	The workspace folder that contains the extension project.
Last modified	The last time the project was changed.

- **Extension details** enables you to configure some of the extension project parameters:

Parameter	Definition
Name	The display name of the extension.
Version	The version number of the extension project.
Description	A description of the extension project.
URI	The identifier for the extension in the configuration (not editable).
Namespace	The prefix used for the URI when you create a new artifact type or property.

Project Editor: Environment Tab

The **Environment** tab displays the parameters of the SOA Systinet server that the Customization Editor is configured for:

Figure 28. Project Editor: Environment Tab
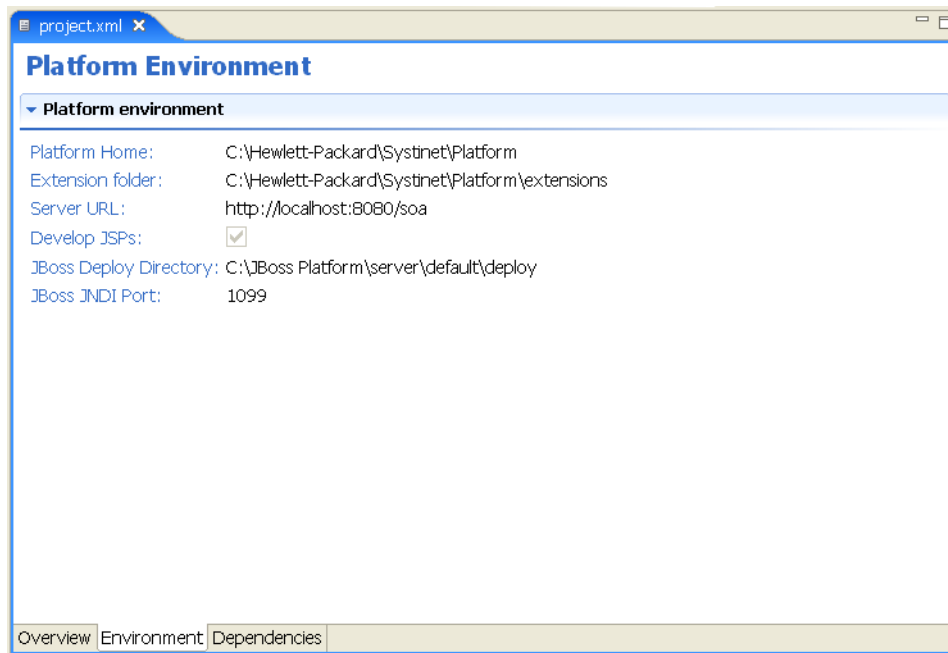
The **Environment** tab contains the following parameters:

Parameter	Definition
Platform Home	Your SOA Systinet installation folder.
Extension Folder	The location of the extension folder in your SOA Systinet installation.
Server URL	The URL used to access SOA Systinet.
Deploy JSPs	Indicates if component and JSP development is configured for your extension project.
JBoss Deploy Directory	Deployment directory for your JBoss application server.
JBoss JNDI Port	JNDI port configured in your JBoss application server.

Project Editor: Dependencies Tab

The **Dependencies** tab displays general information about your extension project and any dependencies that is has:

Figure 29. Project Editor:Dependencies Tab
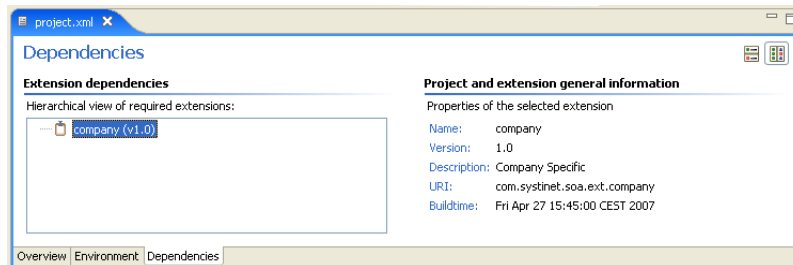
The **Dependencies** tab contains the following panes:

- **Extension dependencies** displays a tree of extensions that the extension project is dependent on.

- **Project and extension general information** displays parameters of the extension project:

Parameter	Definition
Name	The name of the extension.
Version	The version number of the extension project.
Description	A description of the extension project.
URI	The identifier for the extension project in the configuration (not editable).
Buildtime	The creation date and time of the extension.

Property Editor

Properties are attributes of artifacts. The property editor allows you to modify the properties in your extension project. Each property has an editor with the property name as the title containing the tabs described in the following sections:

- Property Editor: General Tab on page 120

  The main attributes of the property.

- Property Editor: Advanced Tab on page 122

  The advanced attributes of the property.

Property Editor: General Tab

The **General** tab displays the general attributes of the artifact and enables you to modify some of them:

Figure 30. Description Editor: General Tab

This tab contains the following panes:

- **General**

  The following property parameters are displayed in this segment:

Parameter	Definition
Display Name	The name of the property as it appears in the SOA Systinet UI.
Local Name	The name of the property as it is stored in the extension.
URI	The identifier for the property descriptor in the configuration.
Description	The description of the property as it appears in the SOA Systinet UI.
Collection Display Name	The plural version of the name as it appears in the SOA Systinet UI.
Deprecated	A check box indicating whether the property is currently active in the extension.

- **Property Type**, **Relationship**, or **Taxonomy**

  There are three types of property and this segment is specific to each:

  - **Property Type**

    The property type for primitive properties.

  - **Relationship**

    This pane displays the source and target artifacts that this relationship links and some parameters of the inverse relationship. On the right are two sets of two buttons for altering the source and target artifacts:

    - **Add** selects an artifact in the extension to source or target.

    - **Remove** removes the relationship from a source or target artifact.

  - **Taxonomy**

    The taxonomy that contains the available options for this property with **Browse** to select a new taxonomy from the extension project.

- **Visible to**

  A list of perspectives with **Add** and **Remove** functionality to select which perspectives can see this property within SOA Systinet.

- **Editable by**. A simple list of perspectives with **Add** and **Remove** functionality to select which perspectives can amend this property within SOA Systinet.

Property Editor: Advanced Tab

The **Advanced** tab displays the advanced attributes of the property and allows you to modify them:

Figure 31. Description Editor: Advanced Tab

This tab contains the following panes:

- **Multiple Cardinality:**

Parameter	Definition
MinOccurs	If a property has multiple cardinality this parameter is the minimum occurrences of the property in an artifact.
MaxOccurs	If a property has multiple cardinality this parameter is the maximum occurrences of the property in an artifact. Use `unbounded` if there is no limit.

> Properties that inherit `urn:com:systinet:soa:model:propertyTypes:text`, such as **Name**, cannot have multiple instances. If you need a text property with multiple cardinality, use one that inherits `urn:com:systinet:soa:model:propertyTypes:plainText`.

> Cardinality for incoming relationship properties cannot be changed.

- **Default Values:**

Parameter	Definition
Default Value	An XML extract defining the default value.
Multiple Default Value	An XML extract defining default values in the case of multiple cardinality.

- **Database Sizes**

Displays optional database sizing for property elements with **Add**, **Edit**, and **Remove** functionality.

Property Group Editor

Properties can be organized into groups. The property group editor allows you to add and remove properties from a property group. Each property group has an editor with the property group name as the title:

Figure 32. System Properties Editor
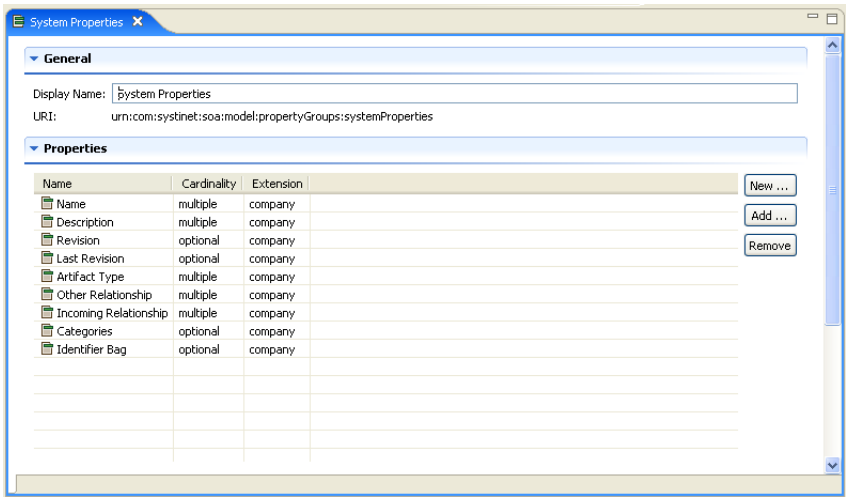
The editor contains the following collapsible segments:

- **General** enables you to change the **Display Name** and view the **URI** of the property group.

- **Properties** enables you to manage the property group.

  On the right are three buttons:

  - **New** creates a new property, as described in Creating a Property Group on page 63.

  - **Add** adds a property to the group from the list of available properties.

  - **Remove** removes the selected property from this property group.

The following property parameters are displayed and described in the table below:

Parameter	Definition
Name	The name of the property.

Parameter	Definition
Cardinality	The occurrence of the property in an artifact with options: • **Optional** The property is not required to be populated. • **Required** The property must be populated. • **Multiple** The property can occur multiple times with different values.
Extension	Which extension this property is part of.

Search View

The **Search** view displays the results of a search or find usage query:

Figure 33. Search View
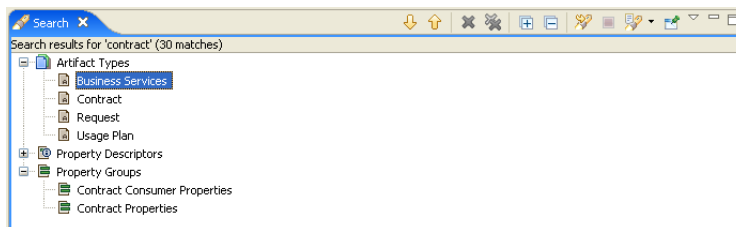
The results are shown as a tree of artifact types, properties, and property groups containing the requested item.

Double-click an artifact type, property, or group to open the relevant editor.

125

Tasks Editor

Tasks are top level SOA Systinet UI components accessible via their URIs. The tasks editor allows you to create, edit, and remove tasks from your extension project.

Figure 34. Tasks Editor
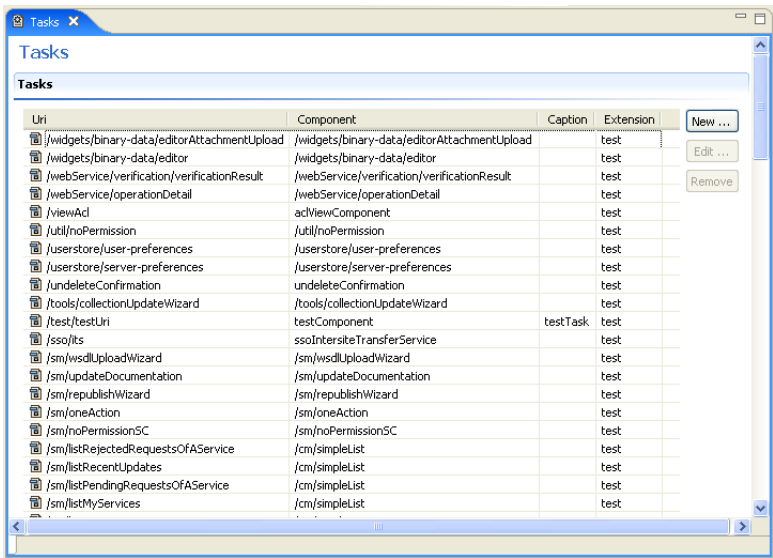
On the right are the following buttons:

- **New** opens the **New Task** dialog box, as described in Creating a Task on page 74.

- **Edit** opens the **Edit Task** dialog box with the same parameters described in Creating a Task on page 74.

- **Remove** deletes the selected task from your extension project.

The following parameters relate to tasks:

Parameter	Definition
URI	The identifier for the task.

Parameter	Definition
Component	The component the task uses.
Caption	A name for the task.
Extension	Which extension this task is part of.

Taxonomies Editor

Taxonomies are category groups that allow you to organize your services. The taxonomies editor allows the import and removal of taxonomies from your extension project:

Figure 35. Taxonomies Editor
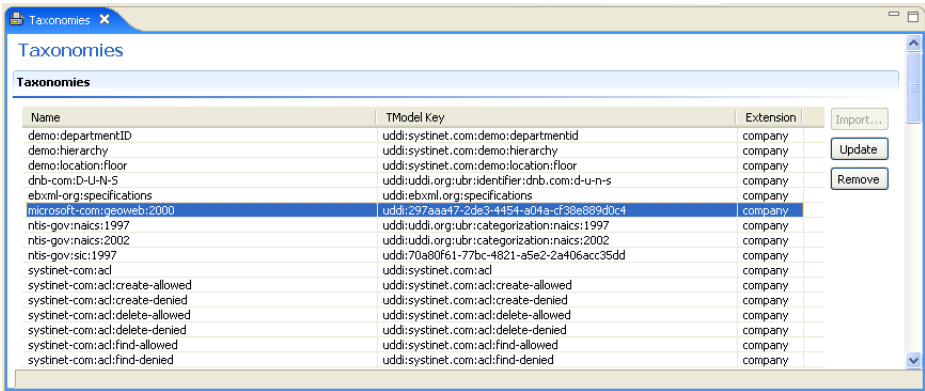
On the right are the following buttons:

• **Import** imports taxonomies from the active SOA Systinet server.

> The referenced SOA Systinet server must be running during import.

• **Update** refreshes the taxonomy list with any changes from the SOA Systinet server.

- **Remove** deletes the selected taxonomy from the extension project.

The following parameters relate to taxonomies:

Parameter	Definition
Name	The name of the taxonomy as it is stored in the extension.
TModel Key	The taxonomy key identifier.
Extension	Which extension this taxonomy is part of.

B Dialog Boxes

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

Contextual Actions

-

-

Contextual Actions: Item Details

Enter general parameters of the new contextual action item:

Parameter	Definition
Display Name	The name of the action as it appears in the SOA Systinet UI
Description	The description of the context action
Action	Select the action from the list
UI Group	Select the action menu that the action is part of
Visible to	Use **Add** and **Remove** to select the perspectives that can see the property

Contextual Action Groups

Enter general parameters of the new contextual action item group:

Parameter	Definition
Display Name	The name of the context action item.
Description	A description of the item.
Action	Browse for the action to add.
Group	From the drop-down list, select the group to which you want to add the action.
Visible to	Use **Add** and **Remove** to select the perspectives that can see the action.

Layout

Layout: Property Details

Specify general details for the new property view:

Parameter	Definition
Property	Use **Browse** to select a property to be displayed in the layout
Layout Group	Select the layout heading for the property to be displayed under from the drop-down list
Visible to	Use **Add** and **Remove** to select the perspectives that can see the property
Editable by	Use **Add** and **Remove** to select the perspectives that can edit the property

Layout: Component Details

Specify general details for the new component view:

Parameter	Definition
Component Name	Use **Browse** to select the component
Layout Group	Select the layout heading that the component is part of
Visible to	Use **Add** and **Remove** to select the perspectives that can see the component
Editable by	Use **Add** and **Remove** to select the perspectives that can edit the component
Component parameters	Use **Add** and **Remove** to select parameters to use with the component

Menu

- Menu: Group Details on page 134

- Menu: Menu Item on page 135

- Menu: Component Details on page 135

Menu: Group Details

Enter general parameters for the new navigation group:

Parameter	Definition
Group Name	The name of the group heading as it appears in the SOA Systinet UI
Description	A description of the new navigation group
Visible to	Use **Add** and **Remove** to select viewing permissions in the SOA Systinet UI

Menu: Menu Item

Enter general parameters for the new navigation action:

Parameter	Definition
Item Name	The name of the navigation link as it appears in the SOA Systinet UI.
Action ID	Browse for the Navigation Action ID.
Group Name	Select the menu heading that the item is part of from the drop-down list.
Description	A description of the new navigation item.
Visible to	Use **Add** and **Remove** to select the perpectives that can see the item

Menu: Component Details

Specify properties for the new component:

Parameter	Definition
Component Action	The identification of the component
Group Name	Select the menu heading that the component is part of from the drop-down list
Visible to	Use **Add** and **Remove** to select the perspectives that can see the component
Component parameters	Use **Add** and **Remove** to select parameters to use with the component

New Action

New Action: Navigation Action

Enter general parameters for the new navigation action:

Parameter	Definition
Action ID	The identification of the contextual or navigation action
Task Location	The name of the action that for is called when the user clicks the link.
Task Parameters	Use **Add** and **Remove** to select the task parameters for the navigation or contextual action specified in **Task Location**. The task parameters are: • **Key**. The name of the parameter • **Value**. The value of the parameter
Partner	The name of the SSO partner
Partner Path	A relative URL pointing to the resource, that is accessed via SSO

New Action: Contextual Action

Enter general parameters for the new contextual action:

Parameter	Definition
Action ID	The identification of the contextual or navigation action.
Use predefined link renderer	De-select if you want to use a specific component instead of a task.
Target Task	Use **Browse** to select the task called by the new context action.
Component name	If **Use predefined link renderer** is not selected use **Browse** to select the component.
Task Parameters	Use **Add** and **Remove** to select the task parameters for the navigation or contextual action specified in **Task Location** which consist of: • **Key**. The name of the parameter • **Value**. The value of the parameter
Partner	The name of the SSO partner.
Partner Path	A relative URL pointing to the resource, that is accessed via SSO.
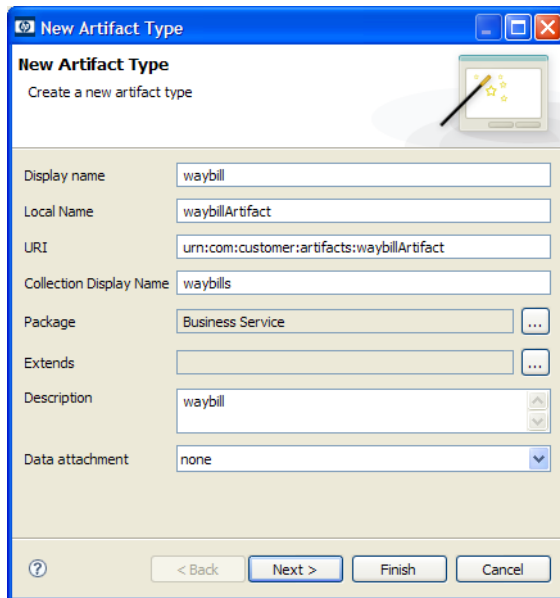
New Artifact

The New Artifact Type and New Artifact Package dialogs are identical and consist of the following ordered stages:

New Artifact: Create

Enter general parameters of the new artifact type:

Parameter	Definition
Display Name	The name of the artifact as it appears in the SOA Systinet UI.

Parameter	Definition
Local Name	The name of the artifact as it is stored in the extension.
URI	The identifier of the artifact descriptor in the configuration.
Collection Display Name	The plural name of the artifact as it appears in the SOA Systinet UI.
Package	Click **Browse** to select a parent artifact package to which the artifact belongs. The default is SOA if the new artifact is not created from within an existing artifact package.
Extends	Click **Browse** to select an artifact type to inherit the properties from If you select **Extends**, the new artifact inherits any future changes made to the chosen parent artifact.
Description	The description of the artifact type as it appears in the SOA Systinet UI.
Data Attachment	If the artifact type normally has attached data content, select the content type from the drop-down list.

New Artifact: Visibility

Specify visibility of the new artifact type:

Parameter	Definition
Copy User Interface configuration from parent artifact type?	If the new artifact type extends an artifact with an existing UI configuration, check the box if you want it to inherit these attributes
Is the artifact available in Service Catalog?	Check the box if the artifact type is visible in the Service UI
Visible to	Use **Add** and **Remove** to select viewing permissions in the SOA Systinet UI
Editable by	Use **Add** and **Remove** to select edit permissions in the SOA Systinet UI

New Artifact: Database Settings

Specify database settings for the new artifact type:

Parameter	Definition
Collection name	The name of collection where the artifact instances are stored
Database table name	The name of the database table where instances of this artifact are stored

You cannot change these parameters if the new artifact extends an artifact type that already defines them.

New Component

Enter general parameters for the new component:

Parameter	Definition
Name	The name of the component as it is stored in the extension.
Page	The JSP used by the component.
Class name	The Java class used by the component.
Component parameters	Use **Add** and **Remove** to select parameters to use with the component.

New Perspective

Enter general parameters for the new perspective:
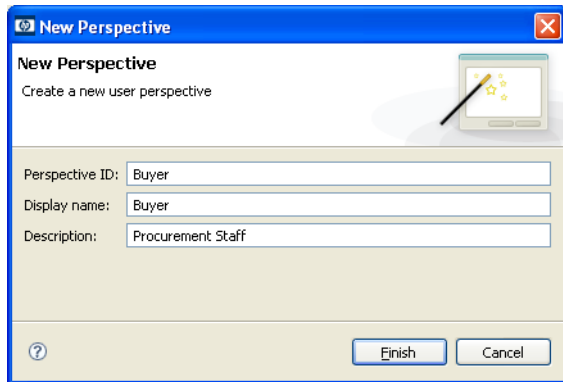
Parameter	Definition
Perspective ID	The name of the perspective in the extension project
Display Name	The name of the perspective as it appears in the SOA Systinet UI
Description	The description of the perspective as it appears in the UI

New Extension Project

The New Extension Project dialog consists of the following stages depending on the options you select:

New Extension Project: Specify HP SOA Systinet Server Installation

Specify configuration details for an HP SOA Systinet server:

Parameter	Definition
Platform Home	Use **Browse** to select the SOA Systinet platform installation folder
Extension Folder	Use **Browse** to select the extension folder in your SOA Systinet installation (populated by default based on the **Platform Home** input
Production Mode	Select this check-box to create the project in production mode (selected by default). This disables customizations that would require the deletion of server data during the application of the extension to SOA Systinet.
Project Type	Select from the following options: • **model** for changes to the SDM model. • **code** for custom code components and UI customizations. • **mixed** for both types of changes (not recommended).

Parameter	Definition
JBoss Deploy Directory	Use **Browse** to select the deployment directory for your application server (usually `JBOSS_HOME\server\default\deploy`)
JBoss JNDI Port	Input the JNDI Port for your application server.

Only JBoss is supported for the development of components and tasks in this release.

New Extension Project: New Server

Specify parameters for the new server:

Parameter	Definition
Name	The name you want to use for the new server.
URL	The endpoint URL of the server you want to use.
Username and Password	Your login credentials for the specified server.
Save password	Select this check-box to store your login credentials.
Validate connection	Select this check-box to validate connection to the server. Otherwise, you can validate connection the next time you log in.

New Project: Select Working Extensions

Choose how to create your working extensions:
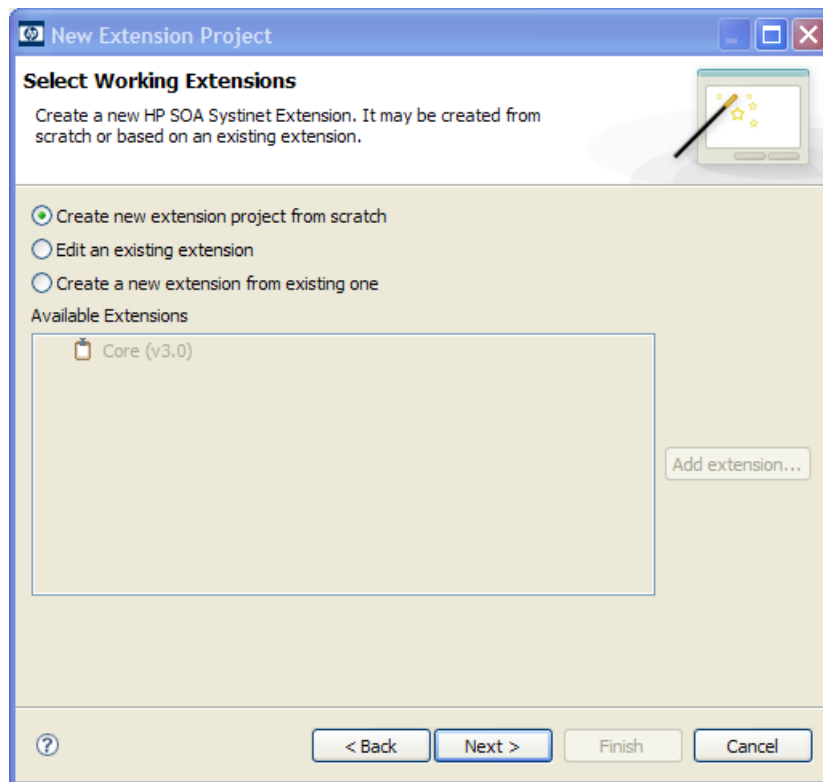
Selection	Action
Create a new extension from scratch	Click **Next**.
Edit an existing extension	Do one of the following: • Select from the available extensions and click **Next**. • Click **Add extension** to browse for and select a different extension, and click **Next**.

	You cannot edit the core extension.

Selection	Action
Create a new extension from an existing one	Do one of the following: • Select from the available extensions and click **Next**. You cannot use the core extensions. • Click **Add extension** to browse for and select a different extension, and click **Next**.

New Project: Create a New Extension

Enter general parameters for the new extension:

Parameter	Definition
Name	The name of your extension project
Namespace	The prefix used for the URI when creating a new artifact type or property
Description	A description of your extension project
Version	The extension version number
URI	The unique identifier of the extension

New Project: Create Dependencies

Select extensions to declare dependencies:
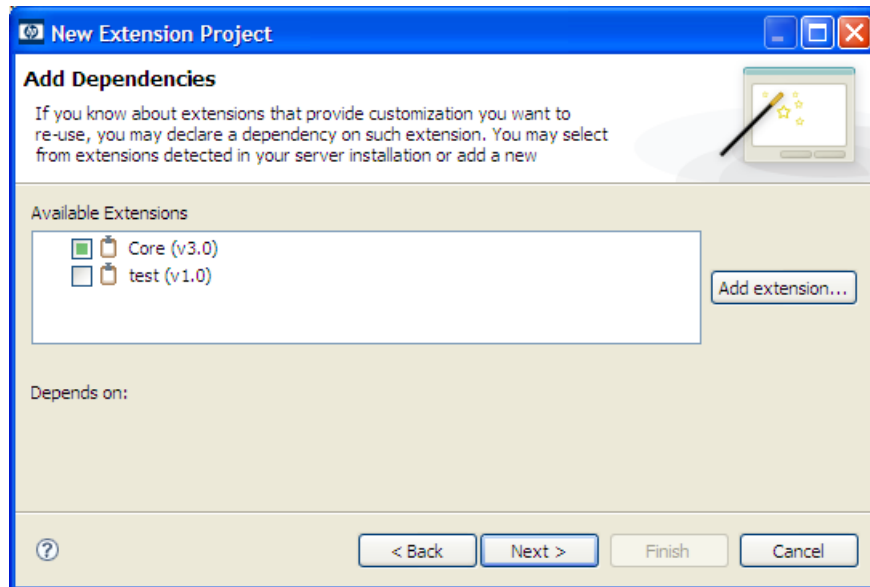
Your changes are stored in your extension. You can change the labels and descriptions of elements from dependent extensions but you cannot delete them or make any other changes. If you want to delete an element from a dependent extension then deprecate it.

New Property

The New Property dialog consist of the following ordered stages:

1 Property parameters according to the property type:

- New Property: Primitive on page 152

- New Property: Relationship on page 153

- New Property: Taxonomy on page 154

For details about property types, see Properties on page 27.

New Property: Primitive

Enter general parameters for the new primitive property:
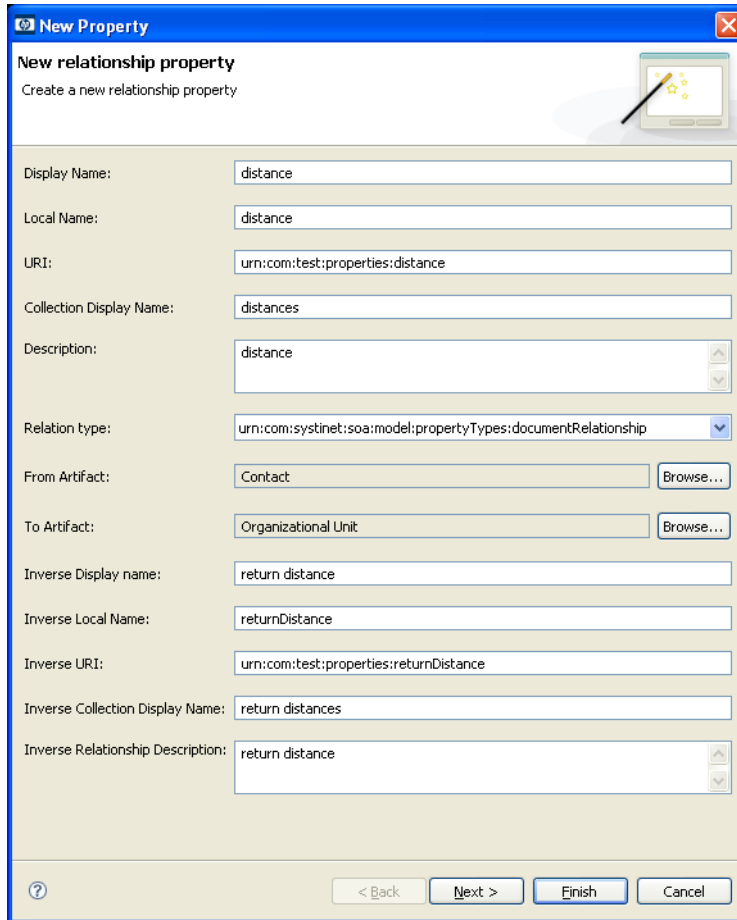
Parameter	Definition
Display Name	The name of the property as it appears in the SOA Systinet UI
Local Name	The name of the property as it is stored in the extension
URI	The identifier for property in the configuration
Collection Display Name	The plural name of the property as it appears in the SOA Systinet UI
Description	The description of the property as it appears in the SOA Systinet UI
Property Type	Use **Browse** to select a property from the available primitive property types. For details, see Properties on page 27.

New Property: Relationship

Enter general parameters for the new relationship property:

Parameter	Definition
Display Name	The name of the property as it appears in the SOA Systinet UI

Parameter	Definition
Local Name	The name of the property as it is stored in the extension
URI	The identifier for the property in the configuration
Collection Display Name	The plural name of the property as it appears in the SOA Systinet UI
Description	The description of the property as it appears in the SOA Systinet UI
Relation Type	Select a type from the drop-down list
From Artifact	Use **Browse** to select the source artifact type of the relationship
To Artifact	Use **Browse** to select the target artifact type of the relationship
Inverse Display Name	The name of the inverse relationship property as it appears in the SOA Systinet UI
Inverse Local Name	The name of the inverse relationship property as it is stored in the extension
Inverse URI	The identifier for the inverse property descriptor in the configuration
Inverse Collection Display Name	The plural name of the inverse relationship property as it appears in the SOA Systinet UI
Inverse Relationship Description	The description of the inverse relationship property as it appears in the SOA Systinet UI

New Property: Taxonomy

Enter general parameters for the new taxonomy property:

Parameter	Definition
Display Name	The name of the property as it appears in the SOA Systinet UI
Local Name	The name of the property as it is stored in the extension
URI	The identifier for the property in the configuration
Collection Display Name	The plural name of the property as it appears in the SOA Systinet UI
Description	The description of the property as it appears in the SOA Systinet UI
Taxonomy	Use **Browse** to select a taxonomy from the available taxonomies with an option to import taxonomies available in SOA Systinet that are not in your extension.

New Property: Visibility

This optional stage contains property visibility parameters.

Parameter	Definition
Visible to	Use **Add** and **Remove** to select viewing permissions in the SOA Systinet UI
Editable by	Use **Add** and **Remove** to select edit permissions in the SOA Systinet UI

New Property Group

Enter general parameters for the new property group:

Parameter	Definition
Display Name	The name of the property group as it appears in the SOA Systinet UI
URI	The identifier for the property group descriptor in the configuration
Properties	Use **Add** and **Remove** to select the properties in the group

New Task

Enter general parameters for the new task:

Parameter	Definition
Uri	The identifier for the task
Component	Use **Browse** to select the component to associate with the task
Caption	A name for the task
Context	Use **Browse** to select the SOA Systinet context associated with the task. Context specifies, for example, the content of the left menu.
Task parameters	Use **Add** and **Remove** to select parameters to use with the task
Wrap	Select **Wrap** to contain the task output in the SOA Systinet UI
Requires Authentication	Select **Requires Authentication** if the task requires the user to log in before execution

New Taxonomy

Enter general parameters for the new taxonomy:

Parameter	Definition
Taxonomy Name	The name of the new taxonomy as it will appear in the Systinet UI.
Taxonomy ID	The system identifier of the new taxonomy.
Filename	The name of the taxonomy as it appears in the repository.

Portlet Definitions

- Portlet Definitions: Add Feed on page 159

Portlet Definitions: Add Feed

Enter general parameters for the portlet:

Parameter	Definition
Portlet ID	The identification of the portlet definition
Url	The link to the RSS source
Partner	The name of the SSO partner
Multiple Instances	Check the box to allow multiple instances of the feed

Portlet Placement

Enter parameters for the portlet placement:

Parameter	Definition
Portlet Reference	Click **Browse** to select from the available portlets
Display Name	The name of the portlet as it appears in the SOA Systinet UI
Location	There are two options: • **Page.** The portlet is open in the main area of the SOA Systinet tab • **Menu.** The portlet is closed but available to add in the left menu
Column	If the portlet is located on the page then use the drop-down list to select which column it appears in
Order	Type a number into this field if this portlet occupies the same column as another portlet. The number defines the order of portlets in the column. The portlet with the lowest order is first (towards the top).
Permanent	If **Menu** is selected check this box to enable multiple instances of the portlet.

Parameter	Definition
Visible to	Use **Add** and **Remove** to select the perspectives that can see the portlet

Registry Mapping

Registry Mapping: New Artifact Mapping

Enter parameters for the new registry mapping of the artifact:

Parameter	Definition
Registry	Use **Select** to select from the available registries

Parameter	Definition
T-ModelKey	Use **Add** to select from the available taxonomies or input one
T-ModelKey version 2	If you are mapping to a UDDI version 2 registry input a taxonomy key
Key Name	The name used to categorize the UDDI entity in the registry
Key Value	The value used to categorize the tModel in the registry
Display Name	The name of the mapping as it appears in the SOA Systinet UI
Cache content	If selected, the content of documents referenced from the UDDI entity are cached in SOA Systinet
Direction	Select the direction of the mapping from the drop-down list

Registry Mapping: New Property Mapping

Enter parameters for the new registry mapping of the property:

Parameter	Definition
Property	Use **Select** to select from the available properties
Direction	Select the direction of the mapping from the drop-down list
Registry	Use **Select** to select from the available registries
T-ModelKey	Use **Add** to select from the available taxonomies or input one
T-ModelKey version 2	If you are mapping to a UDDI version 2 registry input a taxonomy key
Key Name	The name used to categorize the UDDI entity in the registry

Search

Enter parameters to search your extension project:

Parameter	Definition
Containing text	The parameter to search for

Parameter	Definition
Case sensitive	Check the box to make the search case sensitive
Regular expression	Enables more sophisticated search parameters (for example, to find everything with more than one word). This is an advanced topic beyond the scope of this guide.
Search For	Select the SDM entities to search
Search In	Select the entity attributes to search
Scope	Select the extension to search

Share Project

Define parameters for the CVS repository:

Parameter	Definition
Host	The name of the host server.
Repository path	The location of the host server.
User and Password	Your login credentials for the host server.
Connection type	The communication protocol for the CVS host server.

C Eclipse Plug-in Requirements

Ensure that the following prerequisite plug-ins are added 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