HP SOA Systinet Customization Editor

Software Version: 2.51

User Guide

Document Release Date: August 2007
Software Release Date: August 2007

Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Third-Party Web Sites

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

Copyright Notices

Trademark Notices

Java™ is a US trademark of Sun Microsystems, Inc. Microsoft®, Windows® and Windows XP® are U.S. registered trademarks of Microsoft Corporation. IBM®, AIX® and WebSphere® are trademarks or registered trademarks of International Business Machines Corporation in the United States and/or other countries. BEA® and WebLogic® are registered trademarks of BEA Systems, Inc.

Contents

Welcome to This Guide

This guide describes how to use the HP SOA Systinet Customization Editor to edit the SDM configuration and the user interface of an installation of SOA Systinet. You can use the Customization Editor to modify the model, its registry mappings and certain UI elements.

How This Guide is Organized

This guide explains how to use the Customization Editor to create an extension to the configuration of HP SOA Systinet, how to configure the individual elements of SOA Systinet and then deploy them to an installation of SOA Systinet.

The guide explains:

Getting Started on page 11. Introducing and configuring the Customization Editor, its main features, and the customizable elements of SOA Systinet.

Manipulating Artifact Types on page 31. Customizing the attributes, properties and appearance of artifacts.

Manipulating Properties on page 51. Creating and modifying properties and property groups.

Configuring the SOA Systinet UI on page 63. Customizing the main tabs of SOA Systinet and creating new UI elements.

Creating and Using Components on page 79. Creating customized Java and JSPs and using these components with SOA Systinet.

Modifying Taxonomies on page 89. Modifying the set of available taxonomies.

Deploying an Extension to SOA Systinet on page 91. Preparing and deploying your extension to SOA Systinet.

User Interface Reference on page 99. A guide to the Customization Editor user interface.

Document Conventions

The typographic conventions used in this document are:

run.bat make	Script name or other executable command plus mandatory arguments.
[--help]	A command-line option.
either | or	A choice of arguments.
replace_value	A command-line argument that should be replaced with an actual value.
{arg1 | arg2}	A choice between two command-line arguments where one or the other is mandatory.
`rmdir /S /Q System32`	Operating system commands and other user input that you can type on the command line and press **Enter** to invoke. Items in *italics* should be replaced by actual values.
`C:\System.ini`	Filenames, directory names, paths and package names.
`a.append(b);`	Program source code.
`server.Version`	An inline Java or C++ class name.
`getVersion()`	An inline Java method name.
Shift-N	A combination of keystrokes.
Service View	A label, word or phrase in a GUI window, often clickable.
New->Service	Menu choice.

Documentation Updates

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

- Software version number
- 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://ovweb.external.hp.com/lpe/doc_serv/

Support

Mercury Product Support

You can obtain support information for products formerly produced by Mercury as follows:

- If you work with an HP Software Services Integrator (SVI) partner (**http://h20230.www2.hp.-com/svi_partner_list.jsp**), contact your SVI agent.
- If you have an active HP Software support contract, visit the HP Software Support Web site and use the Self-Solve Knowledge Search to find answers to technical questions.
- For the latest information about support processes and tools available for products formerly produced by Mercury, we encourage you to visit the Mercury Customer Support Web site at: **http://hp.-com/go/hpsoftwaresupport**.
- For the latest information about support processes and tools available for products formerly produced by Systinet, we encourage you to visit the Systinet Online Support Web site at: **http://www.systinet.-com/support/index**.
- If you have additional questions, contact your HP Sales Representative.

HP Software Support

You can visit the HP Software Support Web site at:

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

HP Software online support provides an efficient way to access interactive technical support tools. As a valued support customer, you can benefit by using the support 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 Getting Started

The Customization Editor is used 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 describes:

- Introduction to the Customization Editor on page 11 gives an overview of the use of the Customization Editor.
- Customizable Elements of SOA Systinet on page 12 describes the elements of SOA Systinet that are configurable using the Customization Editor.
- Creating an Extension Project on page 17 describes how to start a new project to configure SOA Systinet.
- Introduction to the User Interface on page 19 explains the main features of the Customization Editor user interface.
- Navigating the Customization Editor on page 21 describes the layout of the Extension Explorer – the main navigation view in the editor.
- Customization Editor Menu Options on page 27 describes the Customization Editor options in the Eclipse drop-down menus.
- Searching the Extension on page 28 explains how to use the search function to find elements in the extensions.
- Saving Modifications on page 29 describes how to save your modifications.

Introduction to the Customization Editor

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.

The overall process of customizing SOA Systinet is as follows:

1. Use the Customization Editor to:
   - Modify the data model and user interface as described in:
     - Manipulating Artifact Types on page 31
     - Manipulating Properties on page 51
     - Configuring the SOA Systinet UI on page 63
     - Creating and Using Components on page 79
     - Modifying Taxonomies on page 89
   - Save your modifications to a stand-alone extension package file as described in Exporting the Extension Project on page 91.
2. Use the Setup Tool provided with SOA Systinet to apply the extension package to an installation of SOA Systinet as described in Applying Extensions on page 92.

Special care must be taken 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. The Customization Editor should prevent this if the SOA Systinet server is running during modification.

Customizable Elements of SOA Systinet

The Customization Editor enables you to modify almost any aspect of an installation of SOA Systinet. These aspects consist of two main groups:

- **SDM Entities** are the building blocks that represent the types of entity that exist in the repository defined by the SOA Definition Model. These are described in more detail in SDM Elements on page 13.
- **UI Entities** are the elements of the SOA Systinet user interface. These are described in UI Elements on page 14.

SDM Elements

The Customization Editor deals with four types of entity in the SDM configuration of SOA Systinet:

- **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.

  Artifact types and packages are visible in the **Artifact Types** branch of the **Extension Explorer** (see Artifact Types on page 23) and artifact type procedures are described in Manipulating Artifact Types on page 31.

- **Properties** are the attributes of artifacts. Properties are 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.

  Properties are visible in the **SDM Details** branch of the **Extension Explorer** (see SDM Details on page 24) and property procedures are described in Manipulating Properties on page 51.

- **Property Groups** enable properties to be organized and added to artifacts as a group instead of individually. For example; name, address, and telephone number could be organized into a contact details property group and then whenever a new artifact type is created that represents a person, the group can be added to it instead of the individual properties.

  Property Groups are visible in the **SDM Details** branch of the **Extension Explorer** (see SDM Details on page 24) and property group procedures are described in Manipulating Properties on page 51.

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

Taxonomies are visible in the **Taxonomies** branch of the **Extension Explorer** (see Taxonomies on page 26) and taxonomy procedures are described in Modifying Taxonomies on page 89.

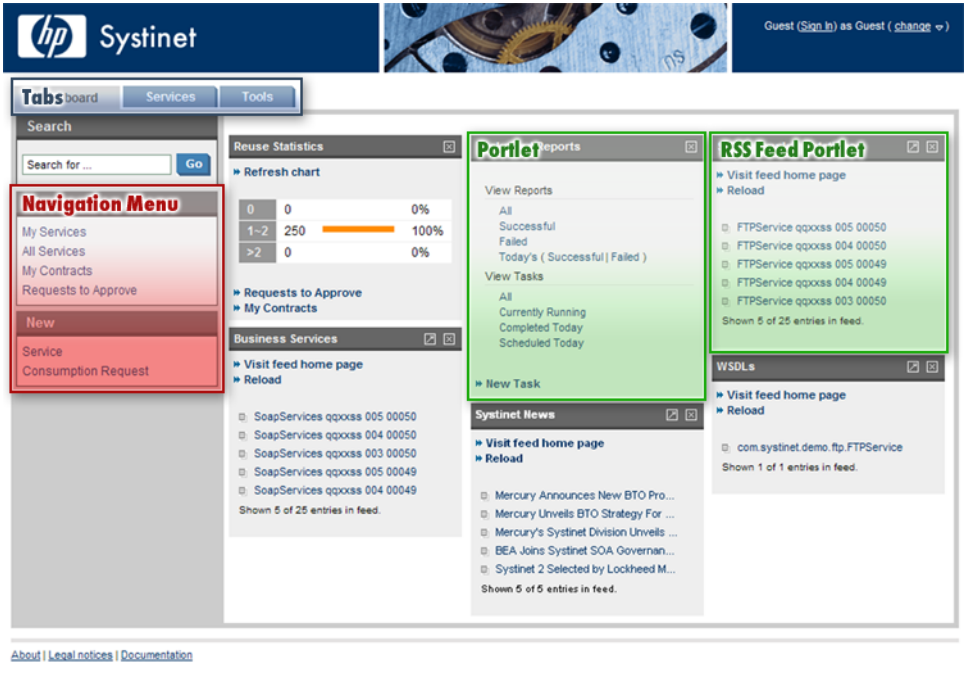
UI Elements

Two types of SOA Systinet pages are configurable:

- The main SOA Systinet tabs.
- The artifact detail and browse pages.

The main SOA Systinet tabs consist of the tab heading, a menu on the left and a set of portlets in the main area of the screen:

Figure 1. SOA Systinet Tab



The two configurable elements on SOA Systinet tabs are:

- **Navigation Menu**

  The navigation menu is organized into navigation groups and actions.

  The details for each SOA Systinet tab are accessible in the **User Interface->*SOA Systinet tab*->Navigation** branch of the **Extension Explorer** (see User Interface on page 25).

  All the navigation actions in your extension are accessible in the **User Interface->UI Details->Navigation Actions** branch in the **Extension Explorer** (see User Interface on page 25).

  Configuring the navigation actions of a SOA Systinet tab, creating and modifying navigation actions, and organizing navigation actions into groups are described in Configuring the SOA Systinet UI on page 63.

- **Portlets and Feeds**

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

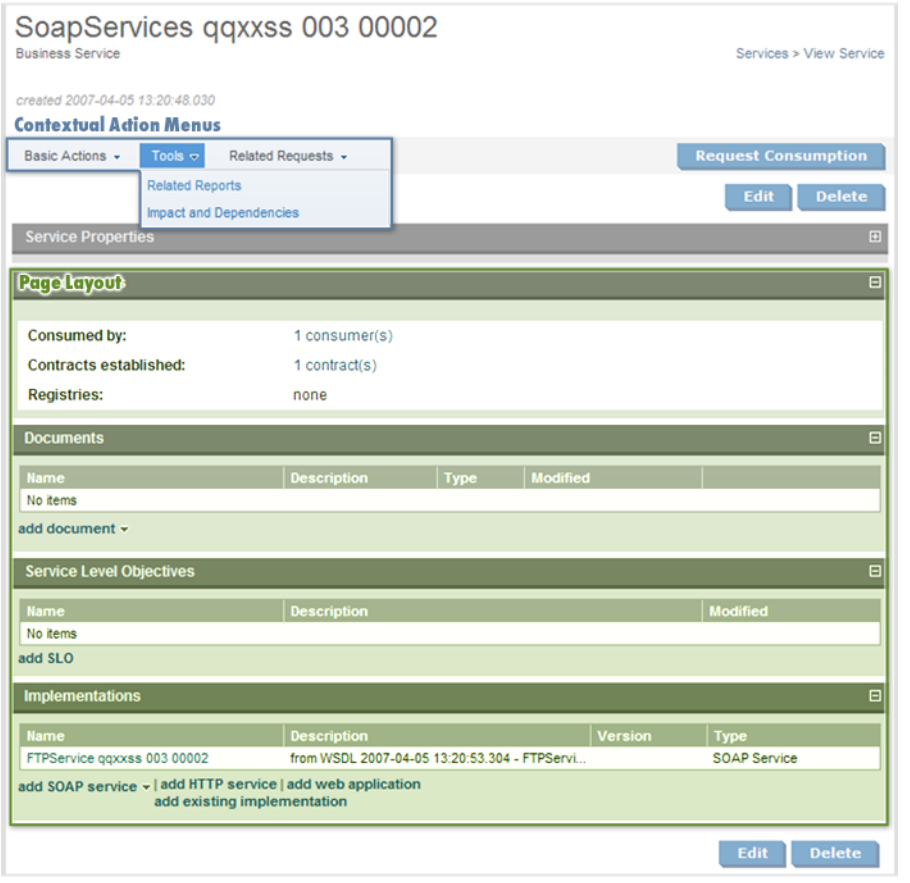
  The details for each SOA Systinet tab are accessible in the **User Interface->*SOA Systinet tab*->Portlets** branch of the **Extension Explorer** (see User Interface on page 25).

  All the portlets in your extension are accessible in the **User Interface->UI Details->Portlets Definitions** branch in the **Extension Explorer** (see User Interface on page 25).

  Configuring the portlets of a SOA Systinet tab and creating new portlets are described in Configuring the SOA Systinet UI on page 63.

Artifact pages consist of the page heading, a set of contextual menus and the main layout of the page:

Figure 2. SOA Systinet Artifact Page



The two configurable elements on artifact pages are:

- **Contextual Action Menus**

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

The details for each artifact type are accessible in the **Service Catalog UI** and **Tools UI** tabs of the relevant artifact editor (see Artifact Types on page 23).

All the contextual actions and groups of actions in your extension are accessible in the **User Interface->UI Details->Contextual Actions** and **Contextual Action Groups** branches in the **Extension Explorer** (see User Interface on page 25).

Configuring the contextual menus of an artifact page is described in Manipulating Artifact Types on page 31. Creating context actions and organizing them into groups is described in Configuring the SOA Systinet UI on page 63.

- **Page Layout**

  The page layout of artifact detail pages can be configured.

  The details for each artifact type are accessible in the **Service Catalog UI** and **Tools UI** tabs of the relevant artifact editor (see Artifact Types on page 23).

  Configuring the layout of an artifact page is described in Manipulating Artifact Types on page 31.

Creating an Extension Project

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

To create an extension project:

1. In the **File** menu, select **New–>HP SOA Systinet Extension Project**.
2. Complete the dialog with parameters:

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

Parameter	Definition
Server URL	The URL used to access SOA Systinet in the form - `http://localhost:8080/soa`
Develop JSPs	Select **Develop JSPs** to enable component development in Customization Editor
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.

3 Click **Next** to select the type of extension project:

4 Select one of the three options:

- Create a new extension project from scratch – this creates an empty extension containing no elements at all. This is the default option for most uses of the Customization Editor.
- Edit an existing extension – open an extension for modification.
- Create a new extension from an existing one – use an existing extension as the basis of a new one. If you use this option, the existing extension must be removed from the extensions folder before you deploy the new extension.

For the last two options select an extension from the extension folder or use **Add Extension** to select one from another location.

Click **Next**.

Edit extension skips to Step 7.

5 For the create options complete the dialog with parameters:

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

Click **Next** to select an extension dependency.

6 For the create options you can optionally add dependencies on other extensions. All extensions are dependent on the core extension.

Select extensions from the list or use **Add extension** to add an extension from outside the default extensions directory to the list. Click **Next** to set the extension properties.

7 Complete the dialog with the following parameters:

Parameter	Definition
Project Name	The name of the new extension project
Location	Optionally, deselect **Default Location** and use **Browse** to select a new workspace directory

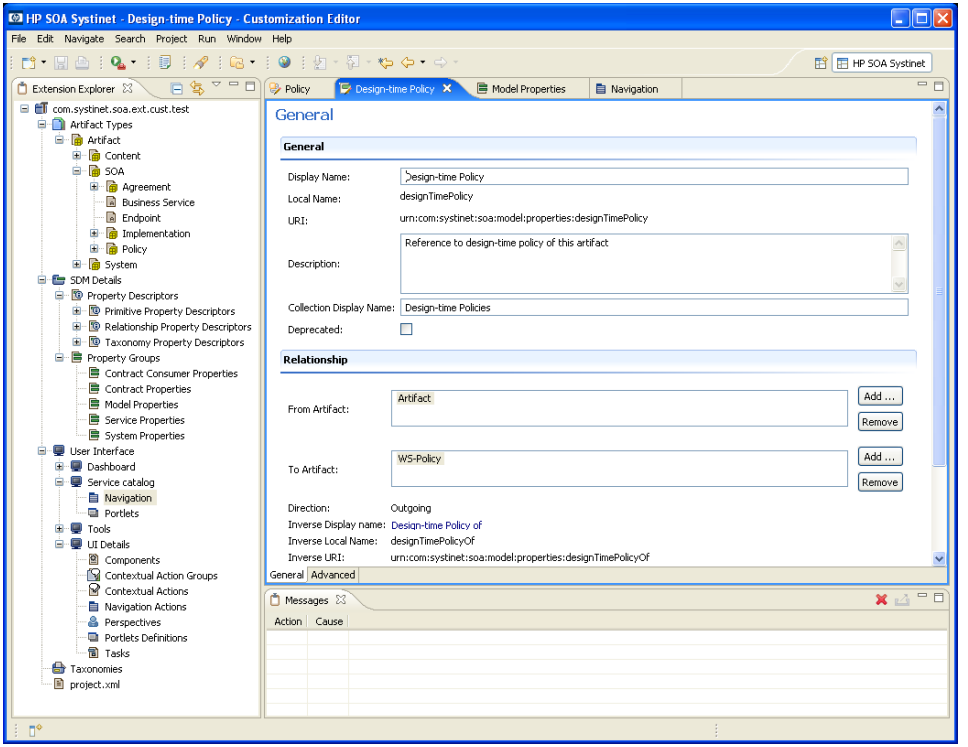
8 Click **Finish** to create the project.

In the **HP SOA Systinet** perspective, the **Extension Explorer** (see Navigating the Customization Editor on page 21) now displays a view of your extension which contains not only the elements contained in your extension project but all the elements from any other extensions that your project depends on.

Introduction to the User Interface

The default **HP SOA Systinet** perspective is split into three sections with menu options across the top. On the left is a tree view of the SDM configuration. The main section contains the details of the item highlighted on the left. Multiple editors can be open in this main section and the functionality varies depending on the editor. In the bottom-right are views for tracking problems, actions, and searches.

Figure 3. Customization Editor UI



The details of the user interface are described in:

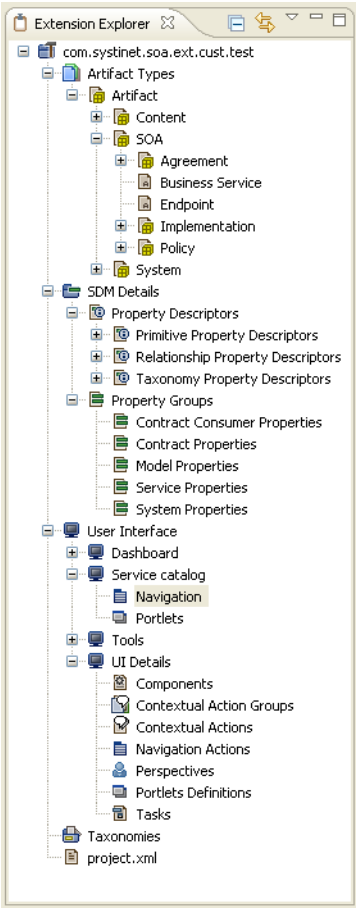
- Navigating the Customization Editor on page 21. The **Extension Explorer** is the main navigation view for the Customization Editor.
- User Interface Reference on page 99 describes the functionality of each editor and view in the main and bottom-right sections.

Navigating the Customization Editor

The **Extension Explorer** 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.

Note: Your changes are stored in your extension. You can change the labels abd 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.

Figure 4. Extension Explorer View



Double-clicking an item in the tree will open the relevant editor in the main section and also open/close a branch.

Right-clicking an element opens a context menu of options relevant to the particular element.

The tree is split into the extension root and five branches representing different elements of your extension. The functionality within each element is described in the following sections:

- Your Extension on page 23 describes the top level object in the extension.
- Artifact Types on page 23 contains the hierarchical structure of the artifact types and packages within the extension.
- SDM Details on page 24 contains properties and property groups.
- User Interface on page 25 contains the customizable elements of the SOA Systinet user interface.
- Taxonomies on page 26 accesses the categorization groups available within this extension.
- project.xml on page 26 is configuration of the current extension project.

Your Extension

The extension name is the root element of the project.

The right-click context menu contains the option to **Build Extension** which creates a `jar` containing your extension ready to import to SOA Systinet (see Deploying an Extension to SOA Systinet on page 91).

Artifact Types

The **Artifact Types** branch contains the artifact types organized into a hierarchical structure reflecting the SOA Model.

Each element below **Artifact Types** is a type of artifact or artifact package and the right-click context menu has the following functionality:

- **New Artifact Type/Package** creates a new artifact type as described in Creating an Artifact Type or Package on page 31.
- **Open Artifact Type/Package** opens the artifact editor for the selected element with the name of the element as the editor title. The functionality of this view is described in Artifact Editor on page 100.
- **Deprecate Artifact Type/Package**. After confirmation, the artifact type becomes inactive in the extension project and is hidden in the SOA Systinet UI.

- **Undeprecate Artifact Type/Package**. After confirmation, the artifact type reactivates in the extension project.
- **Delete Artifact Type/Package**. After confirmation, the artifact type is removed from the extension project.

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.

SDM Details

The **SDM Details** branch contains two element types:

- **Property Descriptors** of which there are three types:
  - Primitive Property
  - Relationship Property
  - Taxonomy Property

  The parent branches contain one context menu option:

  - **New Property** opens a dialog described in Creating a Property on page 51.

  Within each parent element is the list of 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. The functionality of this view is described in Property Editor on page 111.
  - **Deprecate Property**. After confirmation, the property becomes inactive in the extension project and is hidden in the SOA Systinet UI.
  - **Undeprecate Property**. After confirmation, the property reactivates in the extension project.
  - **Delete Property**. After confirmation, the property is removed from the extension project.

    - **Find Usage** opens a search view displaying all the artifacts and property groups that contain this property as described in Search View on page 134.

- **Property Groups**

  The context menu of the parent element only has one option:

    - **New Property Group** opens a dialog described in Creating a Property Group on page 60.

  The context menu for each property group has these options:

    - **Open Property Group** opens the property group editor for the selected element with the name of the element as the editor title. The functionality of this view is described in Property Group Editor on page 115.

    - **Delete Property Group**. After confirmation, the property group is removed from the extension project.

    - **Find Usage** opens a search view displaying all the artifacts that contain this property group as described in Search View on page 134.

User Interface

The **User Interface** branch contains a branch for each tab in the SOA Systinet UI and a **UI Details** branch containing UI elements.

The branch for each SOA Systinet tab contains two elements:

- **Navigation**. Double-click to open the view for that SOA Systinet tab allowing you to customize the menus in SOA Systinet. The functionality of the **Navigation** editor is described in Navigation Editor on page 116.

- **Portlets**. Double-click to open the view for that SOA Systinet tab allowing you to customize the portlets that are available on the selected SOA Systinet tab. The functionality of the **Portlets** editor is described in Portlets Editor on page 118.

UI Details contains the following elements:

- **Components** are elements called by the SOA Systinet UI to perform a particular function. Double click to open the view described in Components Editor on page 120 or right-click and select **New Component** to start the dialog described in Creating a Component on page 79.
- **Contextual Action Groups** allow you to group context actions together and add them to the context options in the SOA Systinet UI. Double-click to open the view described in Contextual Action Groups Editor on page 121 or right click and select **New Context Action Group** to start the dialog described in Creating a Context Action Group on page 72.
- **Contextual Actions** are menu options available within artifact and service pages in the SOA Systinet UI. Double click to open the view described in Contextual Actions Editor on page 122 or right click and select **New Context Action** to start the dialog described in Creating a Contextual Action on page 70.
- **Navigation Actions** are the links in SOA Systinet that navigate between pages. Double click to open the view described in Navigation Actions Editor on page 124 or right click and select **New Navigation Action** to start the dialog described in Creating a Navigation Action on page 68.
- **Perspectives** control the views available to users in SOA Systinet. Double click to open the view described in Perspectives Editor on page 125 or right click and select **New Perspective** to start the dialog described in Creating a Perspective on page 74.
- **Portlets Definitions** are the components in the main section of SOA Systinet pages. Double click to open the view described in Portlets Definitions Editor on page 126 or right click and select **New Portlet** to start the dialog described in Creating a Portlet on page 75.
- **Tasks** are UI wrappers for components. Double click to open the view described in Tasks Editor on page 127 or right-click and select **New Task** to start the dialog described in Creating a Task on page 81.

Taxonomies

The **Taxonomies** branch allows you to control the taxonomies that are available within your extension project. Double-click to open the view described in Taxonomies Editor on page 129.

project.xml

The **project.xml** branch accesses the view allowing you to configure the current extension project as described in project.xml Editor on page 130.

Customization Editor Menu Options

The menus in Customization Editor are the normal Eclipse menus with the addition of specific functionality for the Customization Editor.

The extra menu options in the Customization Editor include:

- **File->New->HP SOA Extension Project** to start a new project as described in Creating an Extension Project on page 17.
- **File->New->Artifact Type** to create an artifact type as described in Creating an Artifact Type or Package on page 31.
- **File->New->Artifact Package** to create an artifact package as described in Creating an Artifact Type or Package on page 31.
- **File->New->Property** to create a property as described in Creating a Property on page 51.
- **File->New->Property Group** to create a property group as described in Creating a Property Group on page 60.
- **File->New->Perspective** to create a perspective as described in Creating a Perspective on page 74.
- **File->New->Component** to create a component as described in Creating a Component on page 79.
- **File->New->Task** to create a task as described in Creating a Task on page 81.
- **Navigate->Go To->Open Artifact** to open an artifact editor as described in Artifact Editor on page 100. The keyboard shortcut **Ctrl+Alt+A** also performs this function.
- **Navigate->Go To->Open Property** to open a property editor as described in Property Editor on page 111. The keyboard shortcut **Ctrl+Alt+P** also performs this function.
- **Navigate->Go To->Open Property Group** to open a property group editor as described in Property Group Editor on page 115. The keyboard shortcut **Ctrl+Alt+R** also performs this function.
- **Search->Search** opens a search dialog as described in Searching the Extension on page 28.

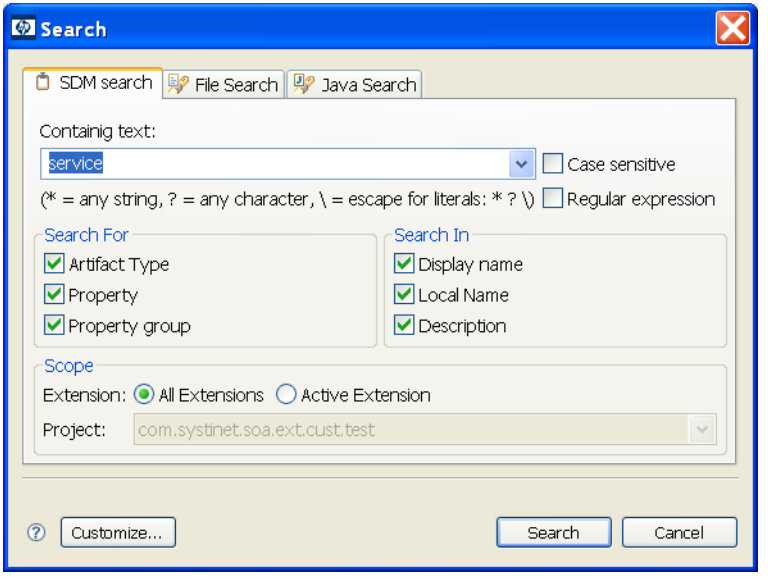
- **Help->Dynamic Help** accesses context sensitive help, directing you to the relevant section of this document based on the active view in the Customization Editor.

Searching the Extension

The Customization Editor contains a customized search function that enables you to find entities in the extension project.

To search your extension project:

1 In the **Search** menu select **Search** to open the **Search** dialog:



2 Complete the **SDM search** dialog containing parameters:

Parameter	Definition
Containing text	The parameter to search the extension for
Case sensitive	Check the box to make the search case sensitive

Parameter	Definition
Regular expression	Allows 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

3 Click **Search** to execute the search and view the results in the **Search** view (see Search View on page 134).

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 **File** menu or use the keyboard shortcut **Ctrl+S**.

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

These changes are made to your extension project and not to the configuration of SOA Systinet. To deploy your modifications to SOA Systinet see Deploying an Extension to SOA Systinet on page 91.

2 Manipulating Artifact Types

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

The chapter describes the following procedures:

Creating an Artifact Type or Package

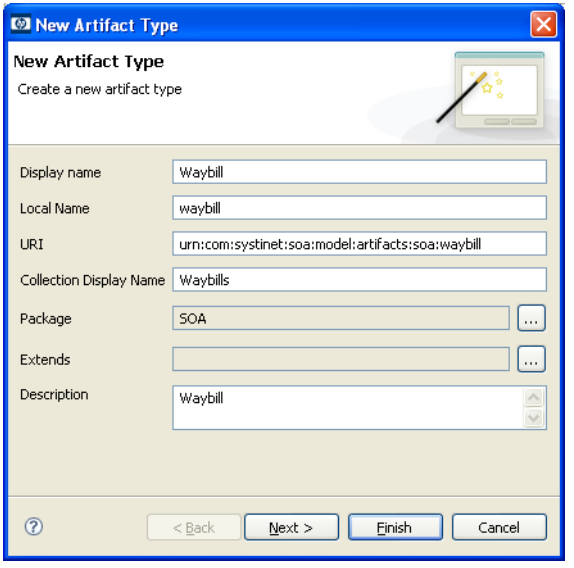
This procedure describes how to create a new artifact type or package in your extension project. For more information on artifact types and packages, see SDM Elements on page 13.

To create an artifact type:

1 To start the **New Artifact Type** dialog:

- In the **File** menu select **New->Artifact Type/Package.**
- Alternatively, in the **Extension Explorer** right-click **Artifact Types** and select **New Artifact Type/Package**.
- Alternatively, in the **Extension Explorer** right-click an artifact package branch and select **New Artifact Type/Package** to create a new artifact type as part of that package.

The **New Artifact Type** or **New Artifact Package** dialog appears:



2 Complete the dialog which contains the following parameters:

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

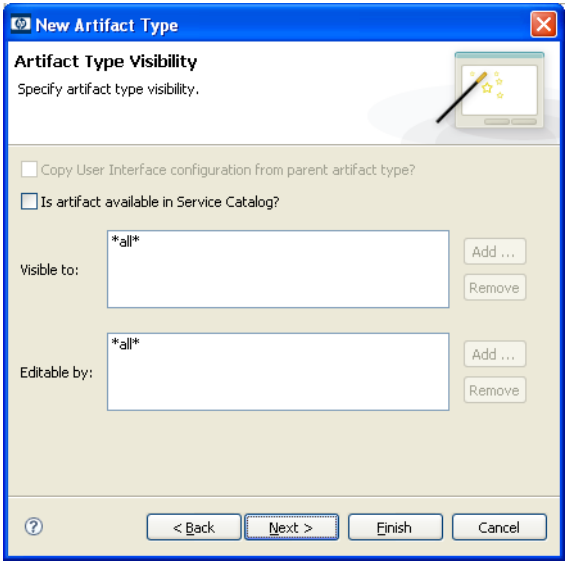
Parameter	Definition
Package	A browse button allows you to select the parent artifact type that this artifact type belongs to
Extends	A browse button allows you to select an artifact type to inherit the properties from **Note:** If you select **Extends**, any future changes to the chosen parent artifact are inherited by the new artifact.
Description	The description of the artifact type as it appears in the SOA Systinet UI

3 Do one of the following:

- Click **Finish** to create the artifact type.
- Continue to Step 4 to set the visibility of the artifact in the SOA Systinet UI.

4 Click **Next**.

The **New Artifact Type/Package** displays visibility parameters:



5 Complete the dialog which contains the following parameters:

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

6 Do one of the following:

- Click **Finish** to create the artifact type.
- Continue to Step 7 to set database parameters.

7 Click **Next**.

The **New Artifact Type/Package** displays database settings.

These settings cannot be changed if the new artifact extends an artifact type that already defines them.

Complete the dialog which contains the following parameters:

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

8 Click **Finish** to create the artifact type.

Modifying the Attributes of an Artifact Type

HP SOA Systinet 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 in the **General** tab.

2 The **General** tab displays three editable segments:

- **Artifact General Information** contains four editable parameters:

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

Parameter	Definition
Deprecated	A checkbox indicating whether the artifact is currently active in the extension

- **Visible to**

  Use **Add** and **Remove** to select the perspectives that can see this artifact type within SOA Systinet.

- **Editable by**

  Use **Add** and **Remove** to select the perspectives that can amend this artifact type within SOA Systinet.

Modifying the Properties of an Artifact Type

The main attributes of artifact types are properties. These can be modified, added individually, or added 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 click the **Properties** tab.

2 The **Properties** tab contains two modifiable segments:

- **Properties**:
  - Click **New** to create a new property as an attribute of the artifact type and create the property as described in Creating a Property on page 51.
  - Click **Add** to select a property to add to the artifact type with a selected cardinality and select where the property appears in the Tools and Services artifact detail pages.
  - Click **Deprecate** to deactivate the property in the artifact type.
  - Click **Undeprecate** to reactivate a deprecated property in the artifact type.
  - Select a property and click **Remove** to remove the property from the artifact type.

- **Property Groups**:
  - Click **New** to create a new property group as a set of attributes of the artifact type as described in Creating a Property Group on page 60.
  - Click **Add** to add a property group to the artifact type.
  - Select a property group and click **Remove** to remove the group from the artifact type.

Modifying an Artifact Type in the Services UI

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

To modify the appearance of Services UI pages:

1. Open the artifact editor and click the **Service Catalog UI** tab.
2. Check the **Is artifact available in Service Catalog?** box to make the artifact type visible in the Services UI.
3. If the artifact type is available in the **Services** UI there are two elements of the view artifact page that can be modified:
   - To modify the layout of the view artifact page see Modifying the Layout of the View Artifact Page on page 38.
   - To modify the contextual actions in the view artifact page see Modifying the Contextual Actions of an Artifact on page 41.

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 click the **Tools UI** tab.

2 There are two elements of the **View Artifact** and **Browse Artifact** pages in the **Tools** UI that can be modified:

- To modify the layout of the **View Artifact** page see Modifying the Layout of the View Artifact Page on page 38.
- To modify contextual actions in the **View Artifact** page or the **Browse Artifact** page see Modifying the Contextual Actions of an Artifact on page 41.

Modifying the Layout of the View Artifact Page

The **View Artifact** page shows the details of an artifact in both the **Services** and **Tools** tabs of SOA Systinet. The elements that appear there can be modified with the Customization Editor.

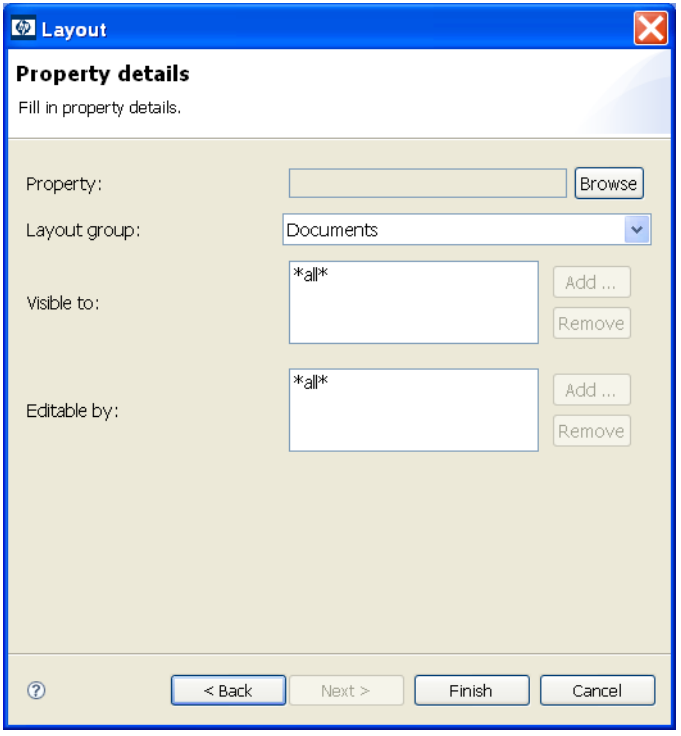
To modify the layout of view artifact pages:

1 Open the artifact editor and click the **Service Catalog UI** or **Tools UI** tab according to which view you want to modify.

2 In the **Layout** segment:

- To change the order of elements, select the element and click **Up** and **Down**.
- To remove an element select it and click **Remove**.
- To edit an element select it, click **Edit** and modify the same parameters as appear in the **Add** functions.
- To add an element click **Add**:
  - To add a new layout heading, in which to group properties and components, select **Group**, input a name for the group and click **Finish** to add it to the layout.
  - To add a property:

1 Select **Property** and click **Next** to open the **Layout Property Details** dialog:



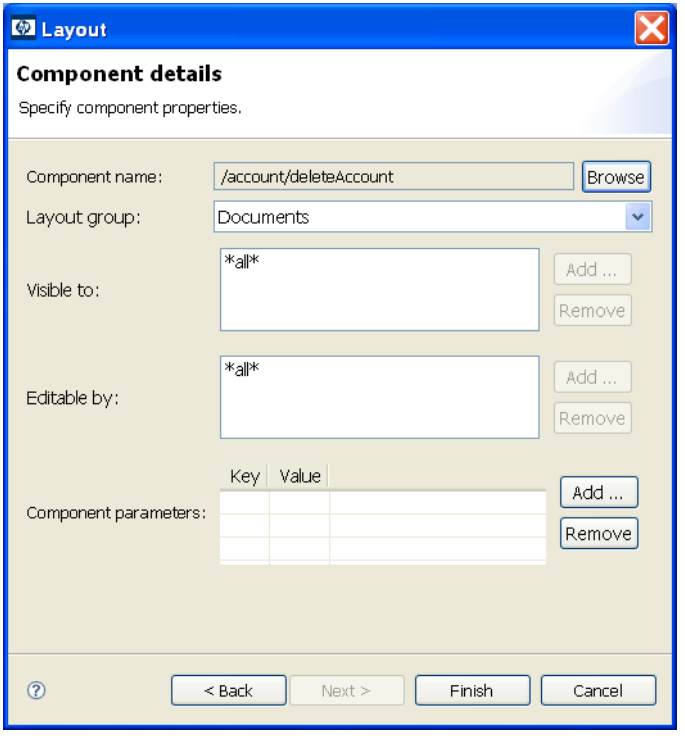
2 Complete the dialog with the following parameters:

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

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

- To add a new functional UI component:

1 Select **Component** and click **Next** to open the **Layout Component Details** dialog:



2 Complete the dialog with the following parameters:

Parameter	Definition
Component Name	Use **Browse** to select the component
Layout Group	Select the layout heading that the component is part of

Parameter	Definition
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

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

This component must also be created using the procedure described in Creating a Task on page 81.

Modifying the Contextual Actions of an Artifact

At the top of the **View Artifact** page in both **Services** and **Tools** and the **Browse Artifact** page in **Tools** is a section containing sets of **Contextual Actions**. The actions that appear there can be modified with the Customization Editor.

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

1 Open the artifact editor and click the **Service Catalog UI** or **Tools UI** tab according to which view you want to modify.

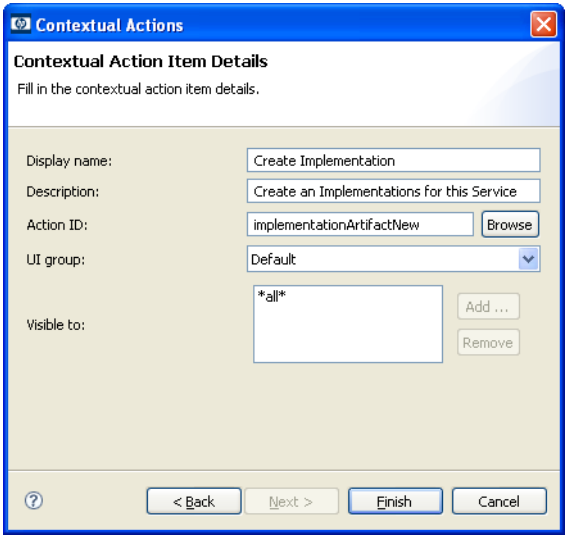
2 In the **Contextual Actions** segment:

In the **Tools UI** tab this segment contains two tabs:

- **Artifact Detail** refers to the **View Artifact** page.

- **Collection** refers to the **Browse Artifact** page.

- To change the order of actions, select the action and click **Up** and **Down**.
- To remove an action select it and click **Remove**.
- To edit an action element select it, click **Edit** and modify the same parameters as appear in the **Add** functions.
- To add an action element click **Add**:
  - To add a contextual menu heading select **UI Group**, input a display name for the group and click **Finish** to add it to the context menu.
  - To add a contextual action item:
    1. Select **Contextual action item** and click **Next** to open the **Contextual Action Item Details** dialog:



    2. Complete the dialog which contains the following parameters:

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

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

- To add a set of contextual actions to the menu:

  1 Select **Contextual action group reference** and click **Next** to open the **Group reference detail** dialog.

  2 Select the **Contextual action group** for the required set of actions and the **UI group** that the actions appears in and click **Finish** to add the set of actions to the contextual menus.

Mapping an Artifact Type to a Registry

Artifacts in SOA Systinet can be mapped to registry entities in a UDDI registry.

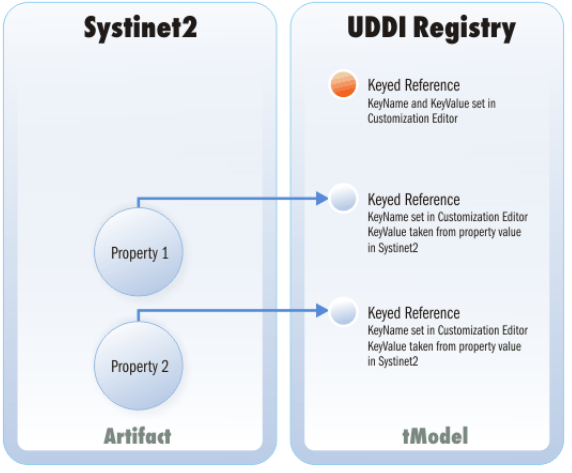
The following artifact types have predefined mappings to registry and cannot be amended:

Table 1. 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

Figure 5 illustrates the mapping of an artifact to a registry entity:

Figure 5. Registry Mapping



To add or modify artifact type mappings to registry entities:

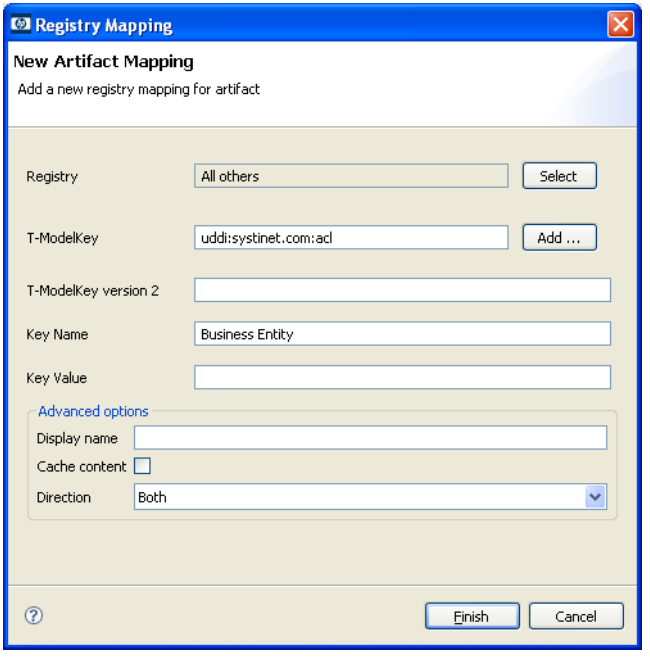
1 Open the artifact editor and click the **Registry Mapping** tab.

2 The functionality of the **Artifact Mapping** segment and **Property Mapping** is the same:

- To edit a mapping, select the mapping, click **Edit** and modify the same parameters as appear in the **Add** functionality.

- To remove a mapping select the mapping and click **Remove**.
- To add a registry mapping for an artifact type:

1. In the **Artifact Mapping** segment click **Add** to open the **New Artifact Mapping** dialog:



2. Complete the dialog with the following parameters:

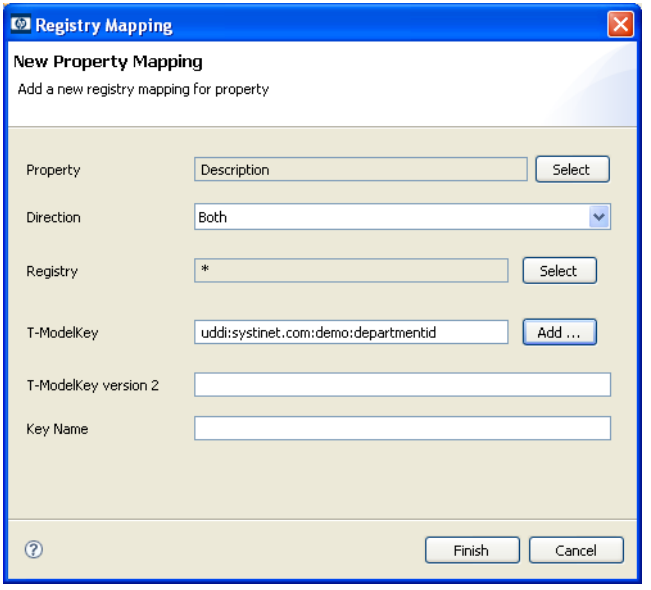
Parameter	Definition
Registry	Use **Select** to select from the available registries
T-ModelKey	Use **Add** to select from the available taxonomies or input one
T-ModelKey version 2	If you are mapping to a UDDI version 2 registry input a taxonomy key
Key Name	The name used to categorize the UDDI entity in the registry
Key Value	The value used to categorize the tModel in the registry

Parameter	Definition
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

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

- To add a registry mapping for a property:

1 In the **Property Mapping** segment click **Add** to open the **New Property Mapping** dialog:



2 Complete the dialog with the following parameters:

Parameter	Definition
Property	Use **Select** to select from the available properties

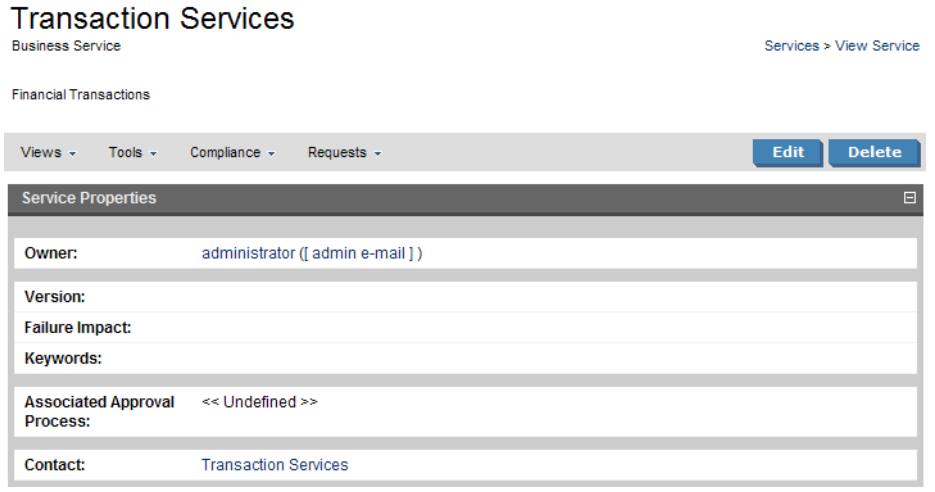
Parameter	Definition
Direction	Select the direction of the mapping from the drop-down list
Registry	Use **Select** to select from the available registries
T-ModelKey	Use **Add** to select from the available taxonomies or input one
T-ModelKey version 2	If you are mapping to a UDDI version 2 registry input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

3 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.

Figure 6. Default Service View of a Business Service



To follow this example requires the following prerequisites:

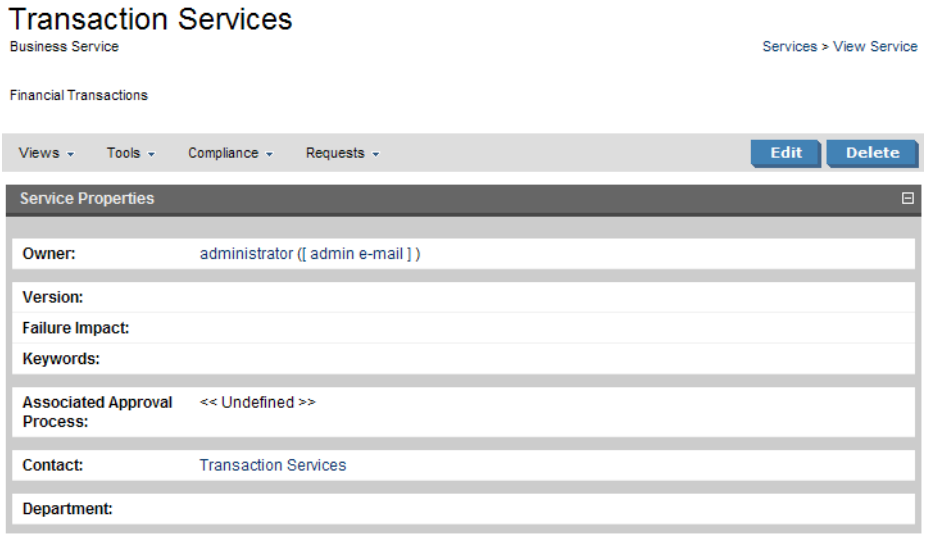
- Create a new extension dependent on the core extension, as described in Creating an Extension Project on page 17.
- Create the new department taxonomy and deploy it to SOA Systinet, as described in the Example: Creating and Publishing a Department Taxonomy section of the HP SOA Systinet Taxonomy Editor Guide.

To add the department property to business services:

1. Open the business service editor and navigate to the **Properties** tab.
2. Click **New** to add a new property as described in Modifying the Properties of an Artifact Type on page 36.
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 Deploying an Extension to SOA Systinet on page 91.
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.

Figure 7. Customized Service View of a Business Service



This property can now be set during business service creation.

3 Manipulating Properties

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

The property procedures are:

- Creating a Property on page 51 describes how to create properties of types:
  - Primitive
  - Relationship
  - Taxonomy
- Modifying a Property on page 57
- Creating a Property Group on page 60
- Modifying a Property Group on page 61

Manipulating Artifact Types on page 31 includes these procedures which are also related to properties:

- Modifying the Properties of an Artifact Type on page 36 describes how to add properties to an artifact.
- Mapping an Artifact Type to a Registry on page 43 describes how to add registry mappings to an artifact and its properties.

Creating a Property

This procedure describes how to create a new property descriptor in your extension project. For more information on properties, see SDM Elements on page 13.

To create a property:

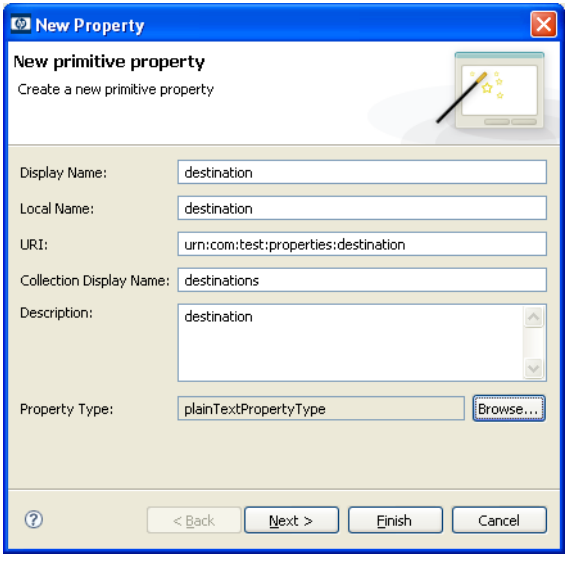
1 Do one of the following:

- In the **File** menu select **New->Property**.
- Alternatively, in the **Extension Explorer** right-click **Property Descriptors** and select **New Property** and the property type.
- Alternatively, in the **Extension Explorer** right-click a property type branch and select **New *Type* Property** to create a property of that type.

The **New Property** dialog appears.

2 A different dialog opens depending on the property type:

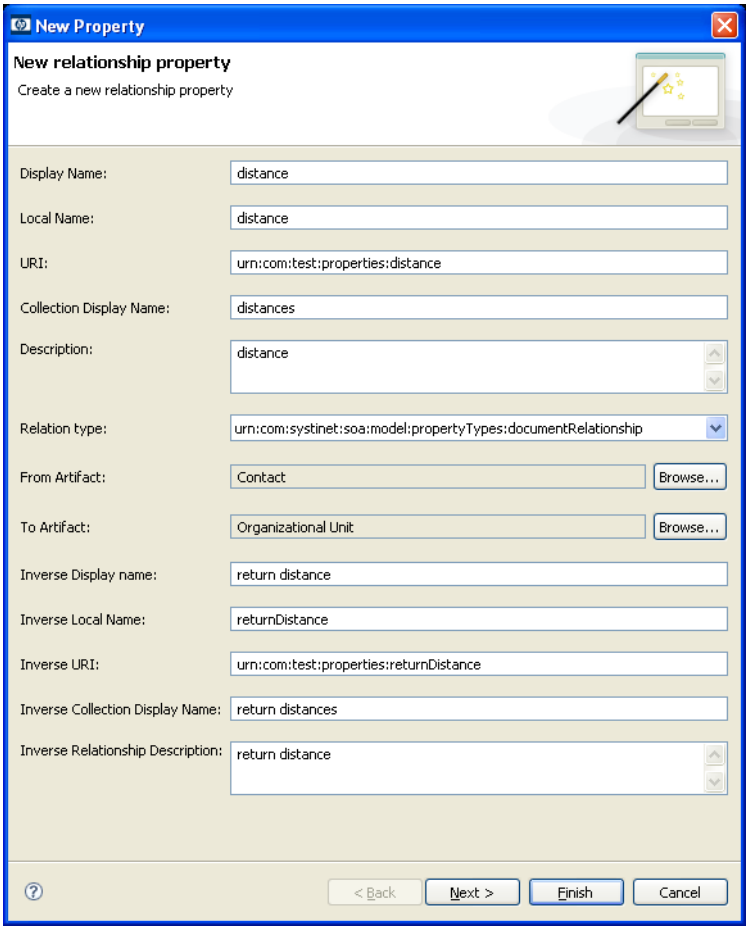
- For primitive properties:



Complete the dialog which contains the following parameters:

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 relationship properties which are created in pairs:

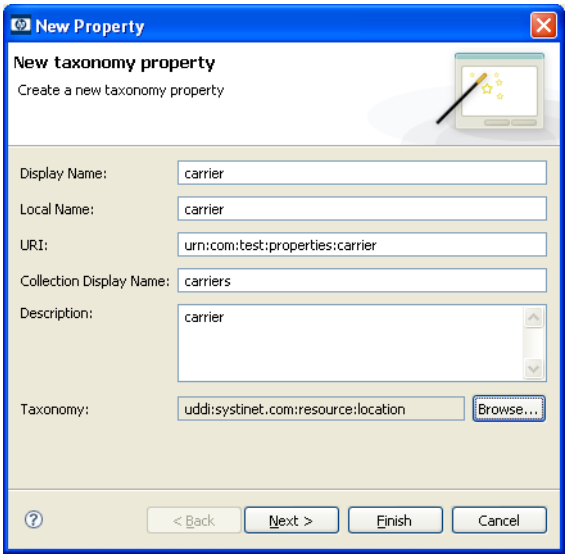


Complete the dialog which contains the following parameters:

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

Parameter	Definition
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

- For taxonomy properties:



Complete the dialog which contains the following parameters:

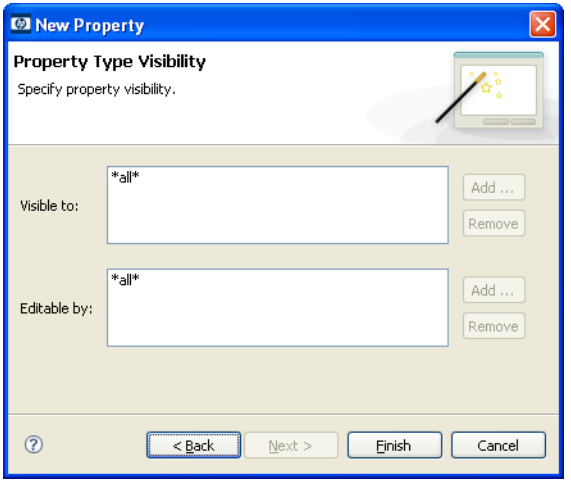
Parameter	Definition
Display Name	The name of the property as it appears in the SOA Systinet UI
Local Name	The name of the property as it is stored in the extension
URI	The identifier for the property in the configuration
Collection Display Name	The plural name of the property as it appears in the SOA 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.

3 Do one of the following:

- Click **Finish** to create the property.
- Continue to Step 4 to set property visibility.

4 Click **Next**.

The **New Property** dialog displays visibility parameters:



5 Complete the dialog which contains the following 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

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

Modifying a Property

HP SOA Systinet Customization Editor enables you to modify properties.

To edit a property:

1 Open the property editor.

2 Edit the general attributes of the property in the **General** tab and the advanced attributes in the **Advanced** tab.

- The **General** tab contains the following editable segments:
  - **General** contains the following editable parameters:

Parameter	Definition
Display Name	The 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
Collection Display Name	The plural version of the name as it appears in the SOA Systinet UI
Deprecated	A checkbox indicating whether the property is currently active in the extension

  - There are three types of property and the second segment is specific to each:
    - **Property Type**

      Primitive properties are not editable and just display the property type.

    - **Relationship**

      Use **Add** and **Remove** to select the endpoints of the relationship.

    - **Taxonomy**

      Use **Browse** to select a new taxonomy from the extension project.

  - **Visible to**

    Use **Add** and **Remove** to select the perspectives that can see this property within SOA Systinet.

- **Editable by**

  Use **Add** and **Remove** to select the perspectives that can amend this property within SOA Systinet.

- The **Advanced** tab contains the following editable segments:

  - **Multiple Cardinality** contains the following editable parameters:

Parameter	Definition
MinOccurs	For multiple cardinality enter a minimum number of instances of the property
MaxOccurs	For multiple cardinality enter a maximum number of instances of the property. Use `unbounded` if there is no limit.

  - **Default Values** contains the following editable parameters:

Parameter	Definition
Default Value	Default value is described by an XML element which must follow the rules defined by XML schema for the selected property and property type. For example, a primitive plain text property with local name email is defined by `<p:email xmlns:p="http://systinet.com/2005/05/soa/model/property">value</p:email>`
Multiple Default Value	If the property has multiple cardinality then specify more than one default value. For example, a plain text property with local name email and multiple cardinality is defined by `<g:emailGroup xmlns:g="http://systinet.com/2005/05/soa/model/propertyGroup" xmlns:p="http://systinet.com/2005/05/soa/model/property"> <p:email>value1</p:email> <p:email>value2</p:email> </g:emailGroup>`

- **Database Sizes** enables you to optionally set the storage size of property elements with **Add**, **Edit** and **Remove** functionality.

Creating a Property Group

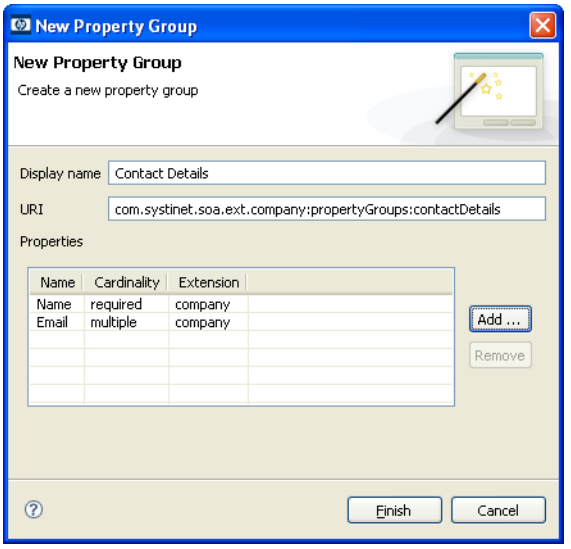
As many artifacts share sets of the same properties it is useful to create groups of properties and add these to the artifacts instead.

To create a property group:

1 Do one of the following:

- In the **File** menu select **New->Property Group.**
- Alternatively, in the **Extension Explorer** right-click **Property Groups** and select **New Property Group**.
- Alternatively, in the artifact **Properties** tab click **New** in the **Property Groups** segment to create a new group as a set of attributes for an artifact type.

The **New Property Group** dialog appears:



2 Complete the dialog which contains the following parameters:

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

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

Modifying a Property Group

HP SOA Systinet Customization Editor enables you to modify property groups.

To edit a property group:

1 Open the property group editor.

2 Modify the property group with one of the following functions:

- Edit the **General** segment to change the display name of the property group.
- Click **New** under **Properties** to create a new property as part of the group as described in Creating a Property on page 51.
- Click **Add** to add a property to the group from the list of available properties.
- Click **Remove** to remove a property from the group.

4 Configuring the SOA Systinet UI

The main tabs in SOA Systinet can be modified and also the appearance of artifact pages can be altered using the Customization Editor

The procedures for altering the appearance of artifact pages are in Manipulating Artifact Types on page 31. This section deals with creating and modifying UI elements and modifying the appearance of the main SOA Systinet tabs.

The available procedures are:

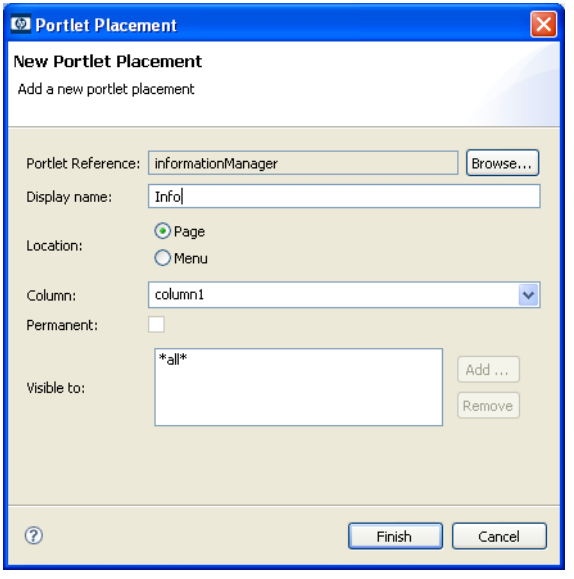
Adding Portlets to a SOA Systinet Tab

The Customization Editor can be used to configure the portlets that are available in each tab in SOA Systinet and where they appear on the tab.

To add portlet to a SOA Systinet tab:

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

2 Click **Add** to open the **New Portlet Placement** dialog:



3 Complete the dialog which contains the following parameters:

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

Parameter	Definition
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
Permanent	If **Menu** is selected check this box to enable multiple instances of the portlet.
Visible to	Use **Add** and **Remove** to select the perspectives that can see the portlet

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

Adding Navigation Groups to a SOA Systinet Tab

The Customization Editor can be used to configure the menu in each SOA Systinet tab. The items in the menu are organized into sections.

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.

3 Select **Group** and click **Next** to open the **Group Details** dialog.

4 Complete the dialog which contains the following parameters:

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

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

Adding Navigation Actions to an SOA Systinet Tab

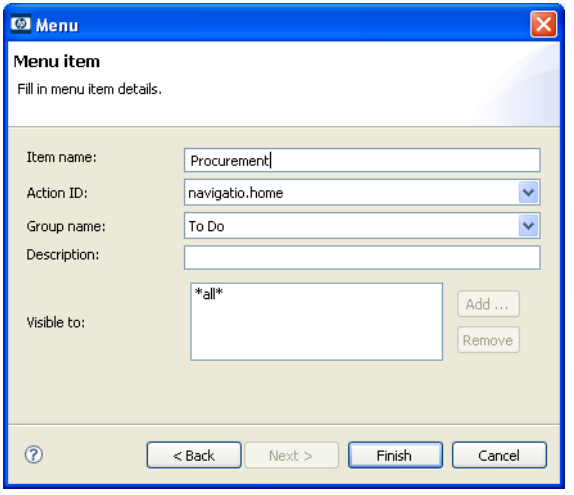
The Customization Editor can be used to configure the menu in each SOA Systinet tab. Each item in the menu corresponds to a navigation action linking to another page in SOA Systinet.

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.

3 Select **Item** and click **Next** to open the **Menu Item** dialog:



4 Complete the dialog which contains the following parameters:

Parameter	Definition
Item Name	The name of the navigation link as it appears in the SOA Systinet UI
Action ID	Select the Navigation Action from the drop-down list
Group Name	Select the menu heading that the item is part of from the drop-down list

Parameter	Definition
Description	A description of the new navigation item
Visible to	Use **Add** and **Remove** to select the perpectives that can see the item

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

Adding Components to a SOA Systinet Tab

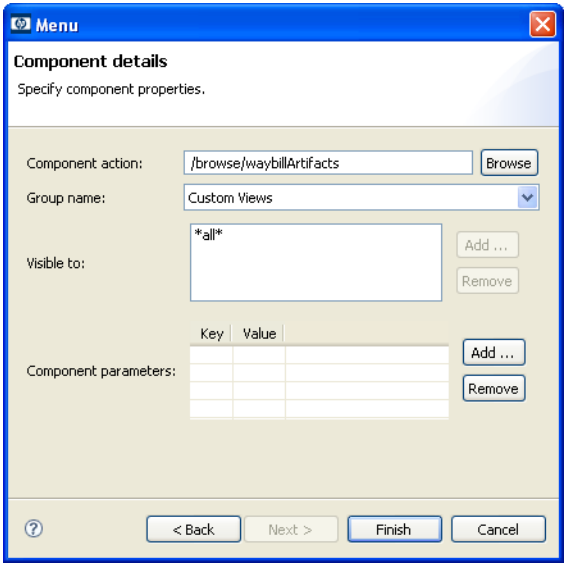
The Customization Editor can be used to configure the left-menu in each tab in SOA Systinet. Some items in the menu are links to a component that performs specific functionality.

To add a component to a 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.

3 Select **Component** and click **Next** to open the **Component Details** dialog:



4 Complete the dialog which contains the following parameters:

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

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

Creating a Navigation Action

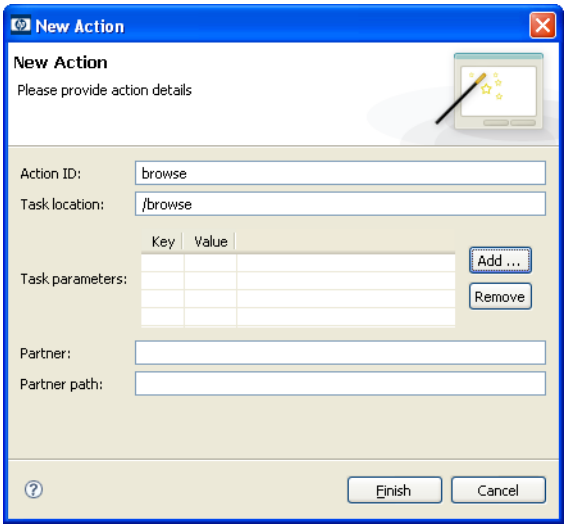
Navigation actions are links from the tab menus to pages in SOA Systinet.

To create an action:

1 Do one of the following:

- Right-click the **Navigation Actions** branch in the **Extension Explorer** and select **New Navigation Action**.
- Alternatively, click **New** in the **Navigation Actions** editor.

The **New Action** dialog appears:



2 Complete the dialog which contains the following parameters:

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** 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

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

- The full URL is compiled from the base URL retrieved from the DB (where the key is the name of the SSO Partner) and the relative URL.

If SSO is not needed to render the menu item, then only the task location and task parameters are used. If SSO is required (the menu item references a task/resource in a different product than it is rendered in) then the target URL is composed from partner, partner path and task parameters.

For example:

```
<action id="tools.newOther">
  <partner>@platform.sso.identity.id</partner>
  <partnerPath>web/publishing/structuredFile/wizard</partnerPath>
  <task location="/publishing/structuredFile/wizard">
    <parameter key="FirstUIStateID">fileDetails</parameter>
  </task>
</action>
```

The SSO URL is composed from e.g. http://johngalt:8080/soa/systinet/platform/ + web/publishing/structuredFile/wizard + ? + FirstUIStateID=fileDetails to final form http://johngalt:8080/soa/systinet/platform/web/publishing/structuredFile/wizard?FirstUIStateID=fileDetails

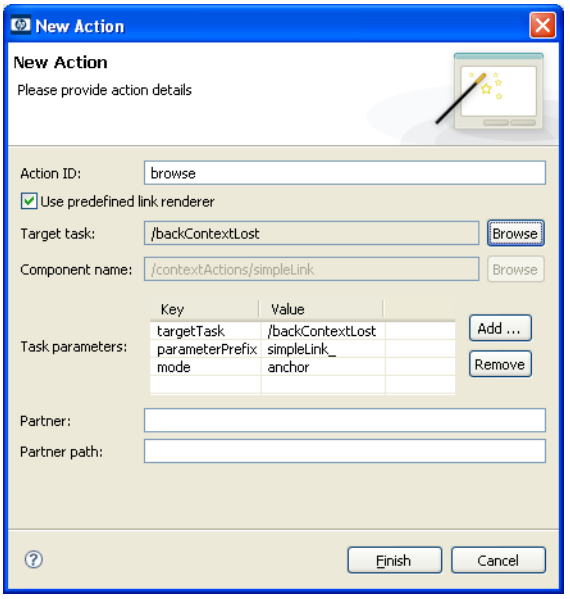
Creating a Contextual Action

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

To create an action:

1. Do one of the following:
   - Right-click the **Contextual Actions** branch in the **Extension Explorer** and select **New Contextual Action**.
   - Alternatively, click **New** in the **Contextual Actions** editor.

   The **New Action** dialog appears:



2 Complete the dialog which contains the following parameters:

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

Parameter	Definition
Partner	The name of the SSO partner
Partner Path	A relative URL pointing to the resource, that is accessed via SSO

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

The full URL is compiled from the base URL retrieved from the DB (where the key is the name of the SSO Partner) and the relative URL.

If SSO is not needed to render the contextual item, then only the task location and task parameters are used. If SSO is required (the contextual item references a task/resource in a different product than it is rendered in) then the target URL is composed from partner, partner path and task parameters.

For example:

```
<action id="tools.newOther">
  <partner>@platform.sso.identity.id</partner>
  <partnerPath>web/publishing/structuredFile/wizard</partnerPath>
  <task location="/publishing/structuredFile/wizard">
    <parameter key="FirstUIStateID">fileDetails</parameter>
  </task>
</action>
```

The SSO URL is composed from e.g. http://johngalt:8080/soa/systinet/platform/ + web/publishing/structuredFile/wizard + ? + FirstUIStateID=fileDetails to final form http://johngalt:8080/soa/systinet/platform/web/publishing/structuredFile/wizard?FirstUIStateID=fileDetails

Creating a Context Action Group

As many artifact types share sets of the context actions it is useful to make groups of actions and add these to artifact types instead.

To create a context action group:

1 Do one of the following:

- In the **Extension Explorer** right-click the **Contextual Action Groups** branch and select **New Context Action Group**.
- Alternatively, click **Add** in the Contextual Action Groups editor.

The **Contextual Actions** dialog appears.

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

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

Adding Context Actions to a Group

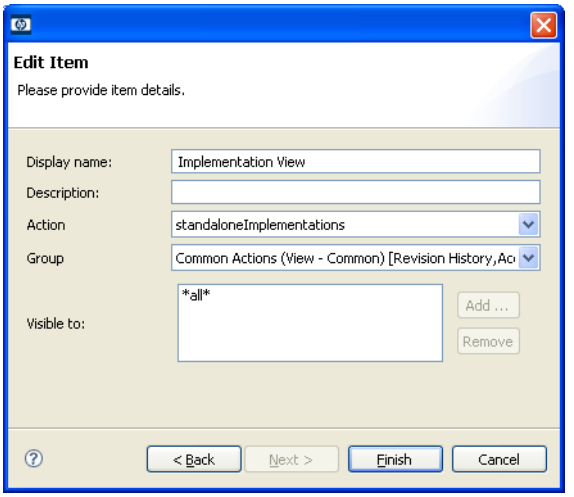
As many artifact types share sets of the context actions it is useful to make groups of actions and add these to artifact types instead.

To add a context action to a group:

1
- In the **Extension Explorer** right-click the **Contextual Action Groups** branch and select **New Context Action Group**.
- Alternatively, click **New** in the Contextual Action Groups editor.

The **Contextual Actions** dialog appears.

2 Select **Action Item** and click **Next** to open the **Edit Item** dialog:



3 Complete the dialog which contains the following parameters:

Parameter	Definition
Display Name	The name of the context action item
Description	A description of the item
Action	Select the action to add from the drop-down list
Group	Select the group to add the action to from the drop-down list
Visible to	Use **Add** and **Remove** to select the perspectives that can see the action

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

Creating a Perspective

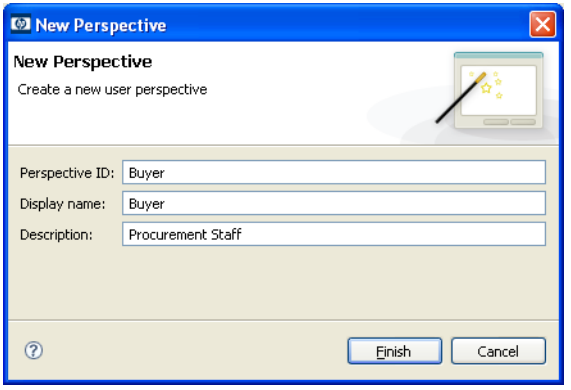
Perspectives control the views of SOA Systinet that are available to each user.

To create a perspective:

1 Do one of the following:

- In the **Extension Explorer** right-click the **Perspectives** branch and select **New Perspective**.
- Alternatively, click **New** in the **Perspectives** editor.

The **New Perspective** dialog appears:



2 Complete the dialog which contains the following parameters:

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

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

Creating a Portlet

Portlets are the components in the main section of SOA Systinet tabs with varying functionality depending 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.

3 Input a name and url for the portlet and click **Finish** to create the portlet.

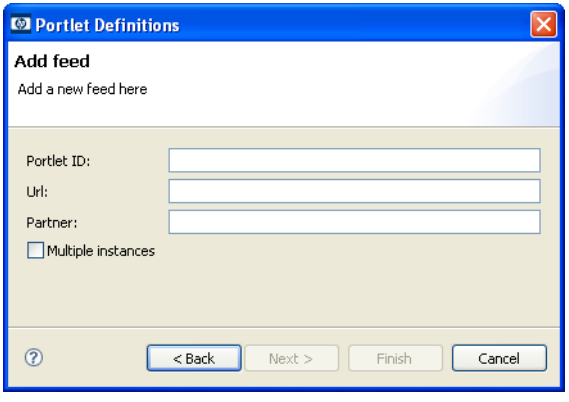
Creating an RSS Feed Portlet

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:



3 Complete the dialog which contains the following parameters:

Parameter	Definition
Portlet ID	The identification of the portlet definition

Parameter	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

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

5 Creating and Using Components

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

The complete process is:

1. Create a new extension project, as described in Creating an Extension Project on page 17, selecting **Develop JSPs**, and setting the appropriate application server settings.
2. Create a component, as described in Creating a Component on page 79.
3. Create a task to use your component, as described in Creating a Task on page 81.
4. Create a navigation action to use your task, as described in Creating a Navigation Action on page 68.
5. Add the navigation item to the UI, as described in Adding Navigation Actions to an SOA Systinet Tab on page 66.
6. Develop the Java and JSP that the component uses in Customization Editor, as described in Developing the Component on page 82.
7. Deploy the extension directly from Customization Editor, as described in Direct Deployment to SOA Systinet on page 96.
8. Debug your code in Customization Editor, as described in Debugging Your Code on page 83.

Example: Adding a Component to the Tools Menu on page 84 is a step-by-step walkthrough of this process.

Creating a Component

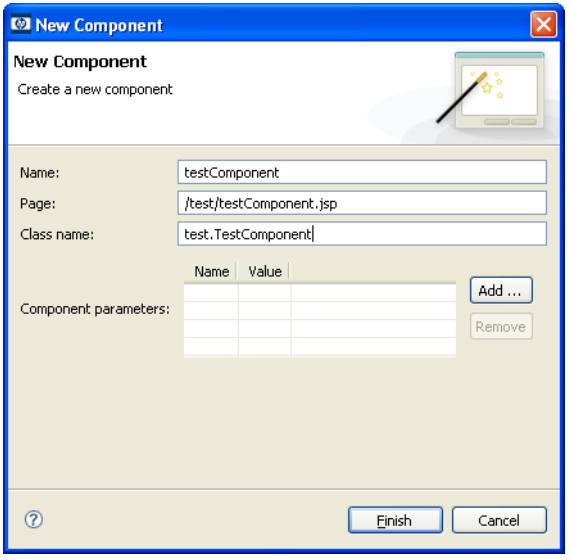
Components are functional elements which can called by a number of different elements in the SOA Systinet UI.

To create a component:

1 Do one of the following:

- In the **Extension Explorer** right-click the **Components** branch and select **New Component**.
- Alternatively, click **Add** in the Components editor.

The **New Component** dialog appears:



2 Input the following parameters:

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

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

Creating a Task

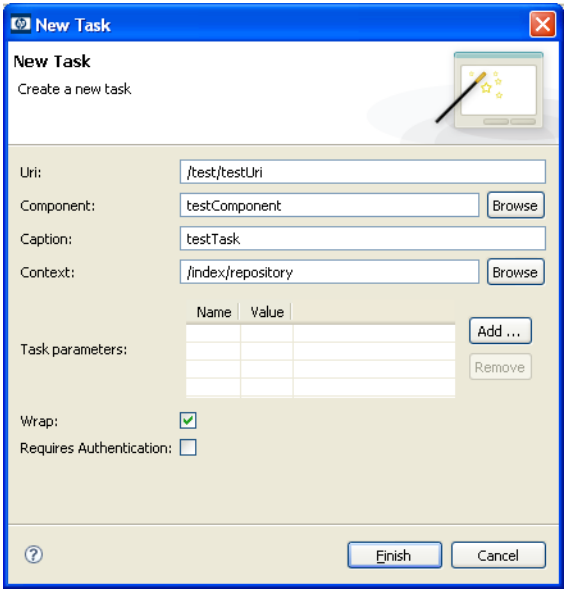
Tasks are top level components accessible via their uris. The addition of a uri allows you to access the component in the SOA Systinet UI and make more sophisticated multi-layered components.

To create a task:

1 Do one of the following:

- In the **Extension Explorer** right-click the **Tasks** branch and select **New Task**.
- Alternatively, click **Add** in the Tasks editor.

The **New Task** dialog appears:



2 Input the following parameters:

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

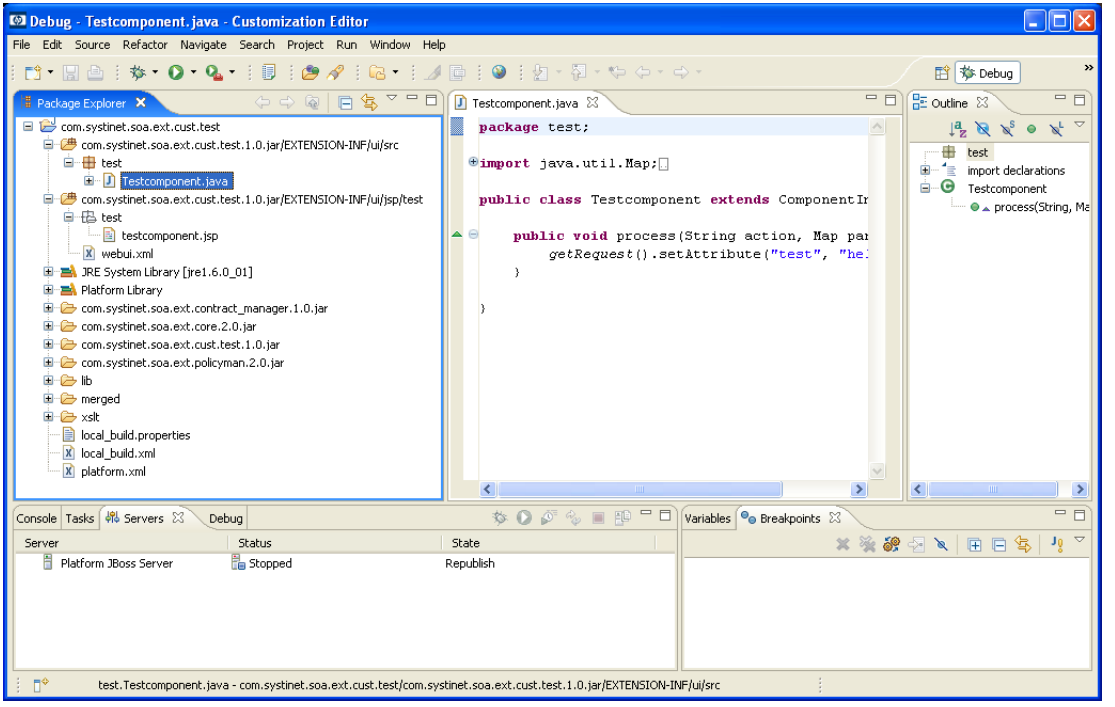
3 Click **Finish** to create the task.

Developing the Component

When you create a component, as described in Creating a Component on page 79, the templates for the Java class and JSP page are automatically created.

To view this code you must switch to the **Java** or the **Debug** perspective:

Figure 8. Debug Perspective



The project contains two source folders containing the Java and JSP files respectively.

Double-click the source file you want to edit to open an edit view enabling you to write your 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 and the new functionality is applied to SOA Systinet directly.

To start SOA Systinet from Customization Editor:

1 In the **Run** menu, select **Debug** to open the **Debug** dialog.

2 Select your server deployment under **Generic Servers**, and then click **Debug** to start the server in debug mode.

The **Console** view displays the progress of the server start.

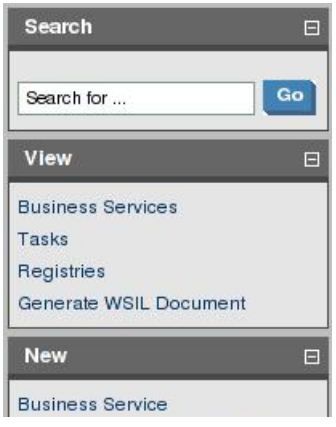
When the server is started the **Server** view displays your server with status **Debugging**.

Make changes to your Java and JSP as required and these changes are applied to SOA Systinet without the need to reapply the extension of restart the server.

Example: Adding a Component to the Tools Menu

In this example, you add a new component, to **Tools** menu in the SOA Systinet UI.

Figure 9. Default Tools Menu View Section



To follow this example requires the following prerequisites:

- Create a new extension dependent on the core extension with **Develop JSPs** selected, as described in Creating an Extension Project on page 17.

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

1 Create a component as described in Creating a Component on page 79 using the following inputs:

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

2 Create a task as described in Creating a Task on page 81 using the following inputs:

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

3 Create a navigation action as described in Creating a Navigation Action on page 68 using the following inputs:

Parameter	Input
Action ID	testNavAction
Task Location	/test/testUri

4 Add the navigation action to the **Tools** tab as described in Adding Navigation Actions to an SOA Systinet Tab on page 66. Select **Item** and use the following inputs:

Parameter	Input
Item Name	testNavItem
Action ID	testNavAction
Group Name	View

5 Create your Java class and JSP as described in Developing the Component on page 82 adding the following code:

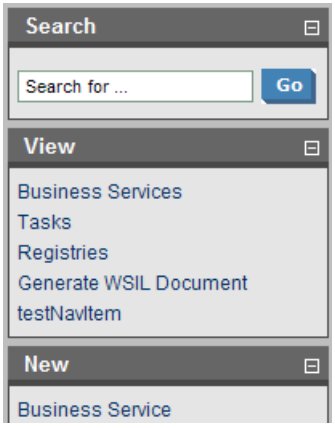
Editor	Input
TestComponent.java	public void process(String arg0, Map arg1) { getRequest().setAttribute("test", "created and deployed your test component."); }
testComponent.jsp	You have succesfully <c:out value="${test}"/>

6 Apply the extension as described in Direct Deployment to SOA Systinet on page 96.

7 Start the SOA Systinet server as described in Debugging Your Code on page 83.

After you deploy this customization, the **View** section **Tools** menu includes the new navigation action.

Figure 10. Customized Tools Menu View Section



Click **testNavItem** to execute the task.

6 Modifying Taxonomies

The Customization Editor controls which taxonomies are available in the extension project. You can create and modify taxonomies using HP SOA Systinet Taxonomy Editor.

To control taxonomies in your extension project:

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

2 To modify the available taxonomies in your extension project:

- To import taxonomies from SOA Systinet and add them to the extension click **Import**.
- To update taxonomies in your extension newer versions from SOA Systinet click **Update**.
- To remove a taxonomy from the extension, select the taxonomy and click **Remove**.

To enable the import or update of taxonomies from SOA Systinet you must specify the correct server URL during configuration (see Creating an Extension Project on page 17).

7 Deploying an Extension to SOA Systinet

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

1. Save your new extension project as a jar file in the SOA Systinet extension folder as described in Exporting the Extension Project on page 91.
2. Deploy the extension to SOA Systinet using the Setup Tool as described in Applying Extensions on page 92.
3. Redeploy the EAR file to your application server as described in Redeploying the EAR to JBoss on page 95.

Alternatively, if you are developing components in Customization Editor, you can directly deploy the extension from Customization Editor as described in Direct Deployment to SOA Systinet on page 96.

Exporting the Extension Project

Customization Editor displays the entire configuration of your extension plus other extensions it depends on. However, your extension only consists 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. Right-click the extension name in the **Extension Explorer** (see Navigating the Customization Editor on page 21 and select **Build Extension** to open the export dialog.
2. By default, the **Extension Folder** set during configuration (see Creating an Extension Project on page 17) is selected. Choose a save location and file name and click **Save**.

Applying Extensions

SOA Systinet can be extended with libraries that are added into the deployed .ear files. The setup tool opens the .ear files, applies the extensions and repacks the .ear files.

Extensions to SOA Systinet come from the following sources:

- The HP SOA Systinet Customization Editor
- HP SOA Systinet Policy Manager extensions used to create assertions in HP SOA Systinet Assertion Editor. These extensions are custom validation handlers. You must apply them to the Policy Manager server in order for Policy Manager to use assertions created with them.
- Upgrading from the Visibility Edition to the Standard Edition, which requires Policy Manager and Contract Manager extensions to be added to SOA Systinet.

Apply extensions according to one of the following scenarios:

- **Single-step (default)** – In this scenario, the setup tool performs all the processes involved in applying extensions, including database alterations, as a single, seamless step.
- **Decoupled Database** – In this scenario, database SQL scripts are run manually, and the setup tool performs the other processes as individual steps that are executable on demand. This is useful in organizations where the user who applies extensions does not have the right to alter the database, which is done by a database administrator.

Single-Step Scenario

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

To apply extensions to SOA Systinet in a single step:

1 Ensure that all extensions are in the `PLATFORM_HOME/extensions` directory. The setup tool automatically applies all extensions in that directory.

Note: If you are applying extensions to another server, substitute `PLATFORM_HOME` with the relevant home directory.

2 Stop the server.

3 Start the setup tool by executing **PLATFORM_HOME/bin/setup.bat(sh)**.

Important: The application server to which SOA Systinet is deployed must not be running.

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.

5 Click **Next** for each of the validation steps and the setup execution. Note that this takes some time.

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

7 For JBoss implementations the EAR file id deployed automatically. For WebLogic and WebSphere, redeploy the EAR file as described in:

- For WebLogic follow the procedure described in the Deploying SOA Systinet Components section of the HP SOA Systinet Installation Guide.
- For WebSphere follow the procedure described in the Deploying the SSO Service and Creating EAR Files section of the HP SOA Systinet Installation Guide.

8 Restart the server.

Decoupled DB Scenario

Follow this scenario if the user who applies extensions does not also modify the database.

To apply extensions and modify the database separately:

1 Ensure that all extensions are in the `PLATFORM_HOME/extensions` directory. The setup tool automatically applies all extensions in that directory.

Note: If you are applying extensions to another server, substitute `PLATFORM_HOME` with the relevant home directory.

2 Stop the server.

3 Start the setup tool by executing **PLATFORM_HOME/bin/setup.bat(sh)**.

Important: The application server to which SOA Systinet is deployed must not be running.

4 Select the **Advanced** scenario and click **Next**.

5 Select the following steps:

- Extensions environment setup
- Extensions model execution
- Extensions database prepare
- Extensions database backup

6 Click **Next**, execute the chosen steps, and exit the setup tool.

7 The database administrator runs the following SQL scripts. INSTALL_HOME refers to the installation directory of the SOA Systinet component to which you are applying the extensions. **The SQL scripts must be run in this order:**

a INSTALL_HOME\conf\sql\dbschema_YOUR-DATABASE_drop.sql

b INSTALL_HOME\working\sdm-gen\build\rdbms\sql\dbschema_YOUR-DATABASE_create.sql

Note: For DB2, the token &1 must be replaced by the tablespace name.

c INSTALL_HOME\working\sdm-gen\build\rdbms\sql\dbschema_YOUR-DATABASE_insert.sql

8 Start the setup tool by executing **PLATFORM_HOME/bin/setup.bat(sh)**.

Important: The application server to which SOA Systinet is deployed must not be running.

9 Select the **Advanced** scenario and click **Next**.

10 Select the following steps:

- Extensions database SQL sync
- Extensions database restore
- Extensions nonmodel execution
- Extensions environment finalize
- Client package creation

11 Click **Next**, execute the chosen steps, and exit the setup tool.

12 For WebLogic and WebSphere, deploy the new EAR file in the `INSTALL_HOME\deploy` directory.

Redeploying the EAR to JBoss

After using the setup tool to apply extensions or updates it is necessary to redeploy the EAR file to the application server. For JBoss this can be done using the setup tool.

For WebLogic and WebSphere follow the procedures described in the Deploying SOA Systinet Components sections of the HP SOA Systinet Installation Guide.

To redeploy the EAR to JBoss:

1 Stop the application server.

2 Start the setup tool by executing **PLATFORM_HOME/bin/setup.bat(sh)**.

Important: The application server to which SOA Systinet is deployed must not be running.

3 Choose the **Advanced** scenario and click **Next**.

4 Select **Deployment**, and then click **Next**.

5 When the setup tool validates the existence of the JBoss Deployment folder, click **Next**.

6 Click **Finish** to close the setup tool.

7 Restart the application server.

Direct Deployment 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 Select **Run->External Tools->External Tools** from the menu to open the **External Tools** dialog.

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.

8 User Interface Reference

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

- Artifact Editor on page 100. Manage your SOA artifacts and customize their properties, appearance and registry mapping.
- Property Editor on page 111. Manage your SOA properties and which perspectives can see and edit them.
- Property Group Editor on page 115. Organize your properties into groups.
- Navigation Editor on page 116. Customize the left-menu navigation links in SOA Systinet tabs.
- Portlets Editor on page 118. Customize which portlets appear in SOA Systinet tabs.
- Components Editor on page 120. Manage the components in SOA Systinet.
- Contextual Action Groups Editor on page 121. Organize your context actions into groups.
- Contextual Actions Editor on page 122. Manage the context actions available in the SOA Systinet UI.
- Navigation Actions Editor on page 124. Manage the navigation links available in SOA Systinet.
- Portlets Definitions Editor on page 126. Manage the portlets available in SOA Systinet.
- Tasks Editor on page 127. Manage tasks in SOA Systinet.
- Perspectives Editor on page 125. Customize the perspectives available in SOA Systinet.
- Taxonomies Editor on page 129. Customize the taxonomies available in SOA Systinet.
- project.xml Editor on page 130. Manage your extension project.
- Messages View on page 133. View the action log as you customize your extension project.

- Search View on page 134. View the results of usage and entity searches.

Artifact Editor

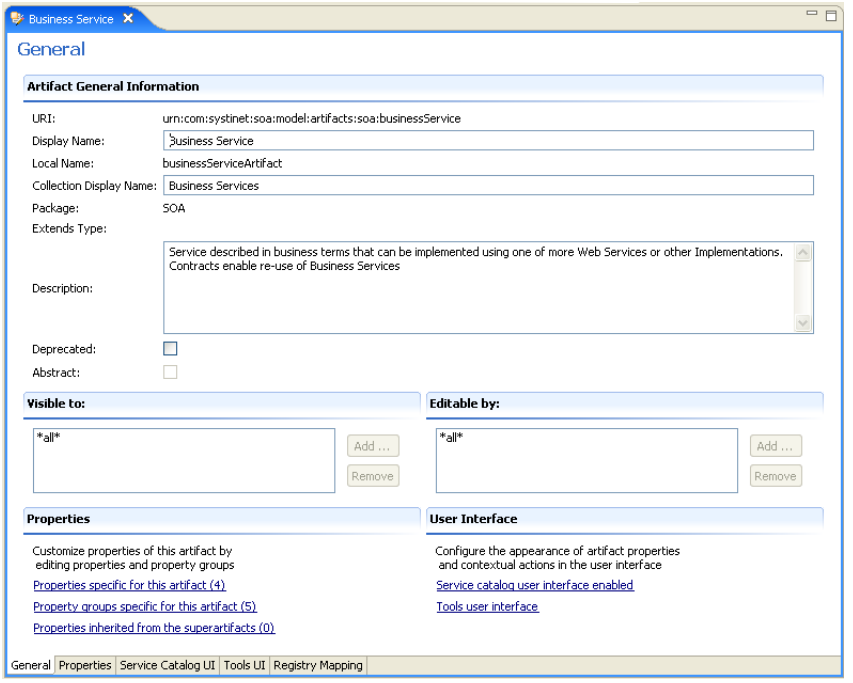
Artifacts are the basic building blocks of SOA and represent all types of entity. This editor allows you to modify the parameters, properties and appearance of an artifact type within SOA Systinet. Each artifact type and package has an editor with the artifact name as the title containing five tabs described in the following sections:

- **Artifact – General Tab on page 100 .** The main attributes of the artifact type
- **Artifact – Properties Tab on page 102 .** The properties of the artifact type
- **Artifact – Service Catalog UI Tab on page 104 .** The format of the View Artifact page in the SOA Systinet UI
- **Artifact – Tools UI Tab on page 107 .** The format of the View Artifact and Browse Artifact pages in the SOA Systinet UI
- **Artifact – Registry Mapping Tab on page 109 .** The mapping of the artifact to registry entities

Artifact – General Tab

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

Figure 11. Business Service Editor – General Tab



The general tab contains the following segments:

- **Artifact General Information**. The following parameters related to artifacts shown in this segment. 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
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

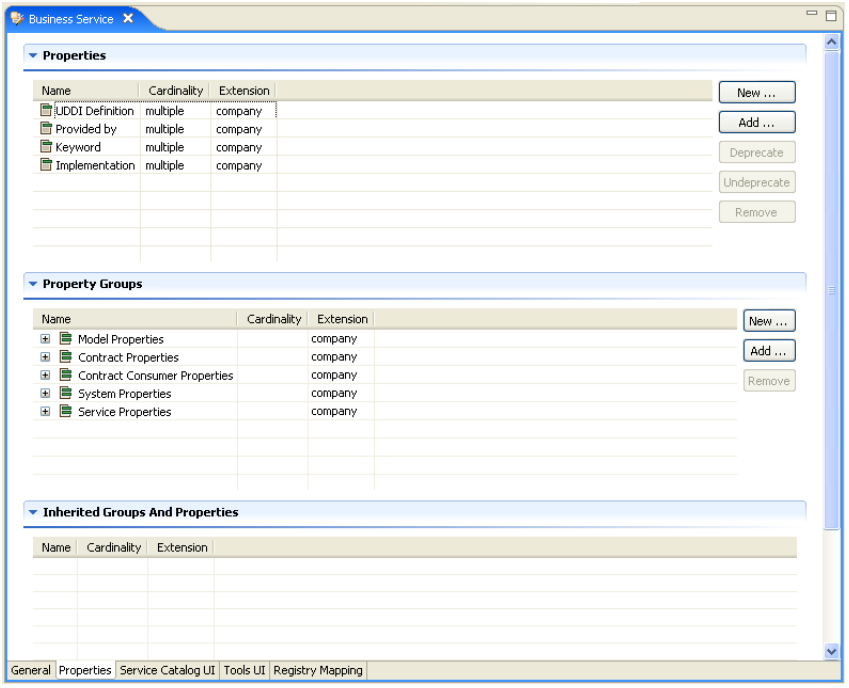
Parameter	Definition
Package	The parent artifact type that this artifact type belongs to
Extends Type	The artifact type that this artifact type inherits the properties of
Description	The description of the artifact type as it appears in the SOA Systinet UI
Deprecated	A checkbox indicating whether the artifact is currently active in the extension
Abstract	A checkbox 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 – Properties Tab on page 102.
- **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 – Properties Tab

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

Figure 12. Business Service Editor – Properties Tab



This tab contains the following segments:

- **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 51.
  - **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 described in Creating a Property Group on page 60
  - **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.

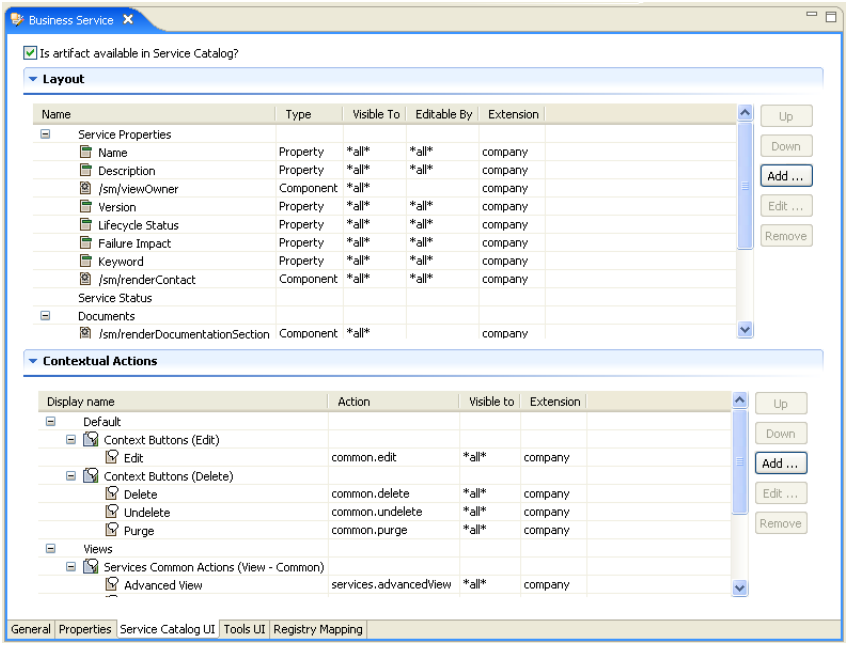
The following property parameters are displayed in these segments:

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 property can occur multiple times with different values
Extension	Which extension this property is part of

Artifact – 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 13. Business Service Editor – Service Catalog UI Tab



There is a checkbox to control whether the artifact is available within the Service Catalog and if this is checked there are two segments 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 the View Artifact Page on page 38.

- **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 segment:

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 a 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 41.
  - **Edit** enables you to edit the context action with the same parameters as **Add** functionality.
  - **Remove** removes the action from context action menus.

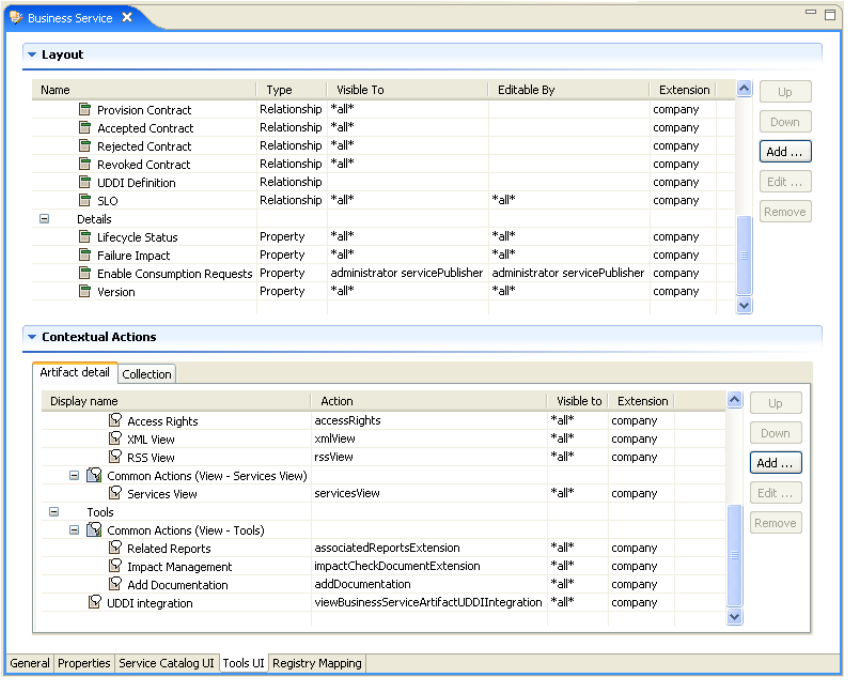
  The following contextual action parameters related to properties are shown in this segment:

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 – 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 14. Business Service Editor – Tools UI Tab



The following segments 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 the View Artifact Page on page 38.
  - **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 segment:

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 a 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 segment. **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 41.
- **Edit** enables you to edit the context action with the same parameters as **Add** functionality.
- **Remove** removes the action from context action menus.

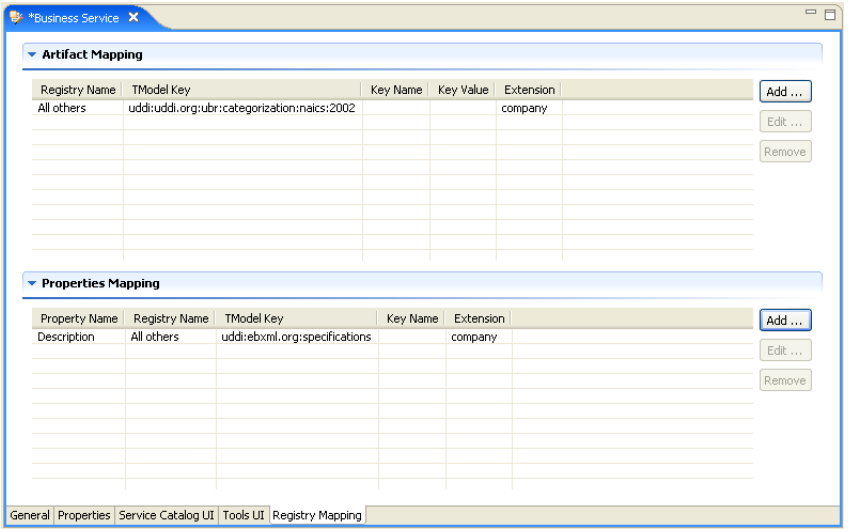
The following contextual action parameters are displayed in this segment:

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 – 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 15. Business Service – Registry Mapping Tab



It is split into the following segments:

- **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 43.
  - **Edit** edits the mapping as described in Mapping an Artifact Type to a Registry on page 43.
  - **Remove** removes the mapping from the artifact.

  The following registry mapping parameters are displayed in this segment:

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

Parameter	Definition
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 43.
  - **Edit** edits a property mapping as described in Mapping an Artifact Type to a Registry on page 43.
  - **Remove** removes a mapping from the selected property.

  The following registry mapping parameters are displayed in this segment:

Parameter	Definition
Property Name	The name of the property
Registry Name	The registry that the artifact type property 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
Extension	Which extension this property is part of

Property Editor

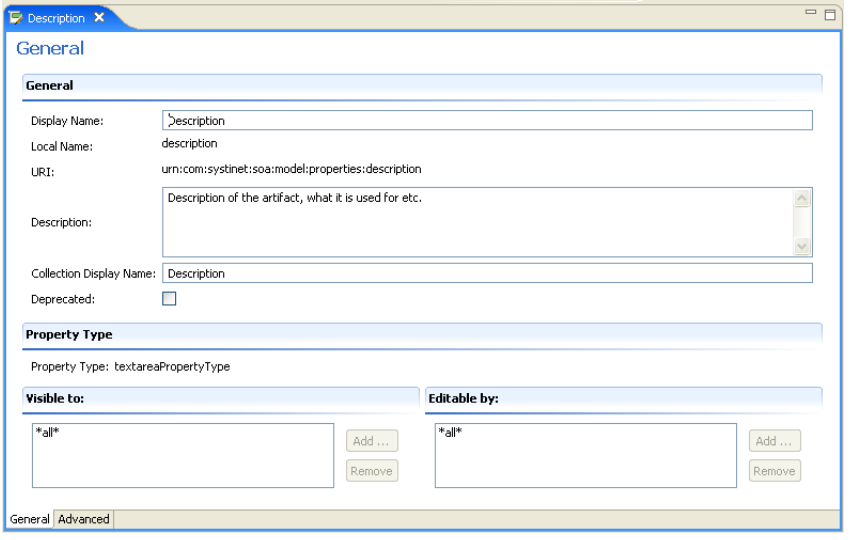
Properties are attributes of artifacts. The property editor allows you to modify the properties in your extension project. Each property has an editor with the property name as the title containing the following tabs described in sections:

- **Property – General Tab on page 112.** The main attributes of the property
- **Property – Advanced Tab on page 113.** The advanced attributes of the property

Property – General Tab

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

Figure 16. Description Editor – General Tab



This tab contains the following segments:

- **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

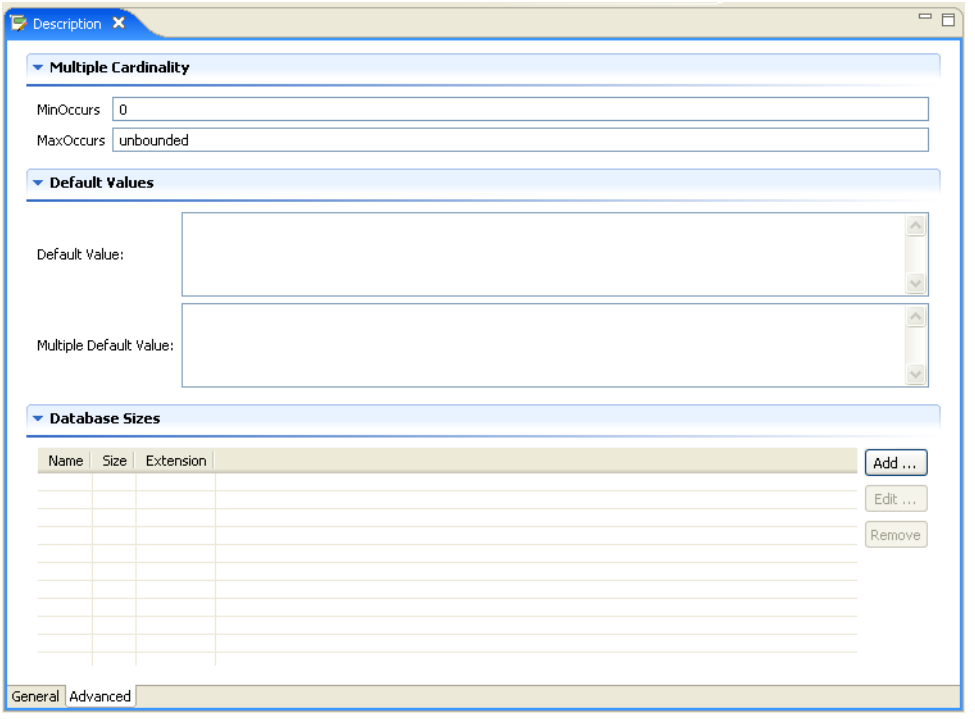
Parameter	Definition
Deprecated	A checkbox indicating whether the property is currently active in the extension

- **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 segment 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 – Advanced Tab

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

Figure 17. Description Editor Advanced Tab



This tab contains the following segments:

- **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.

- **Default Values:**

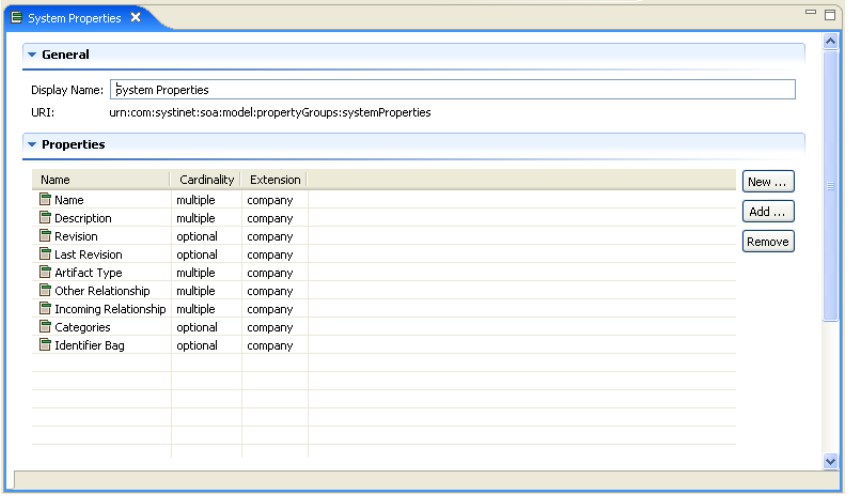
Parameter	Definition
Default Value	An XML extract defining the default value
Multiple Default Value	An XML extract defining default values in the case of multiple cardinality

- **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 18. 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 60.
  - **Add** adds a property to the group from the list of available properties.
  - **Remove** removes the selected property from this property group.

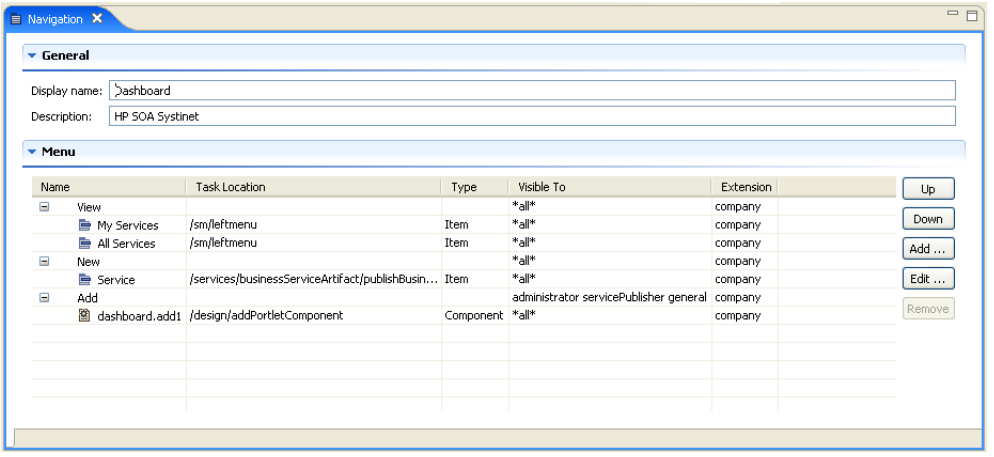
  The following property parameters are displayed and described in the table below:

Parameter	Definition
Name	The name of the property
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

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 19. Dashboard Navigation Editor



The editor contains the following collapsible segments:

- **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 an SOA Systinet Tab on page 66.
    - **Component** adds a component to the menu as described in Adding Components to a SOA Systinet Tab on page 67.

    - **Group** adds a new heading to the menu as described in Adding Navigation Groups to a SOA Systinet Tab on page 65.

- **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 an SOA Systinet Tab on page 66.
    - **Component** edits a component with the same parameters described in Adding Components to a SOA Systinet Tab on page 67.
    - **Group** edits a menu heading with the same parameters described in Adding Navigation Groups to a SOA Systinet Tab on page 65.

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

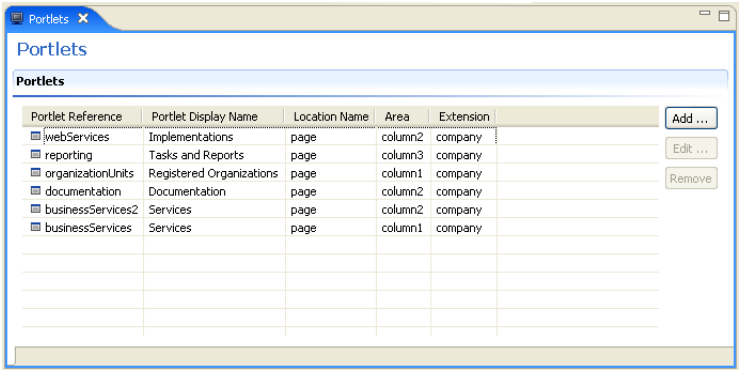
This segment contains the following navigation action parameters:

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
Type	Item or Component
Visible to	Which perspectives can see this item
Extension	Which extension this navigation item 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 20. Dashboard Portlets Editor



On the right are the following buttons:

- **Add** opens the **New Portlet Placement** dialog described in Adding Portlets to a SOA Systinet Tab on page 64.
- **Edit** opens the **Edit Portlet Placement** dialog for the selected portlet with the same parameters described in Adding Portlets to a SOA Systinet Tab on page 64.
- **Remove** removes the selected portlet from the SOA Systinet tab.

The following portlet placement parameters are displayed:

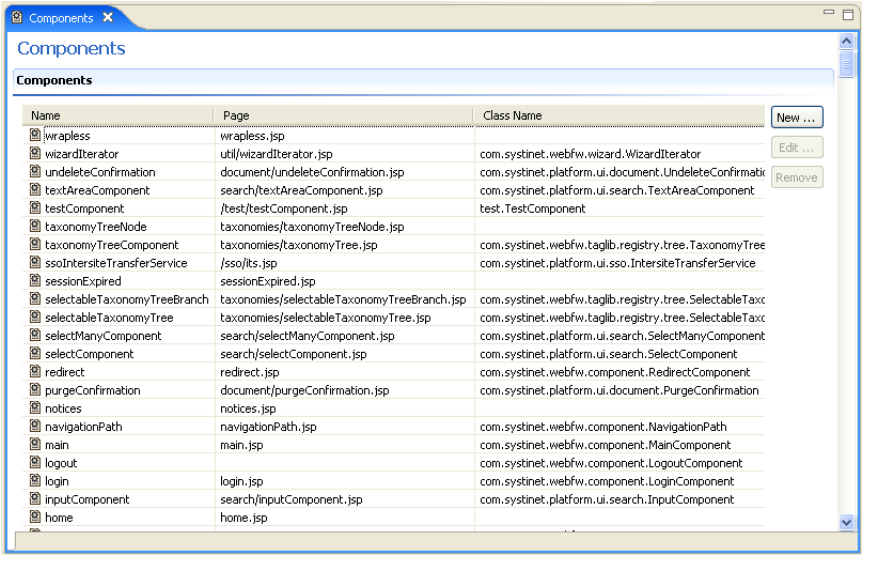
Parameter	Definition
Portlet Reference	The identification of the portlet in the extension
Portlet Display Name	The name of the portlet as it appears on the SOA Systinet tab
Location Name	There are two options: • **page**. The portlet is open in the main area of the SOA Systinet tab • **menu**. The portlet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

Parameter	Definition
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

Components Editor

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

Figure 21. Components Editor



On the right are the following buttons:

- **New** opens the **New Component** dialog as described in Creating a Component on page 79.
- **Edit** opens the **Edit Component** dialog with the same parameters described in Creating a Component on page 79.

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

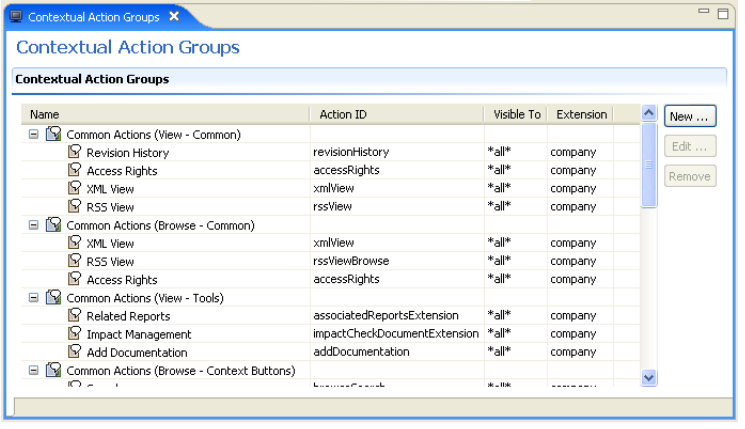
The following component parameters are displayed:

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 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 22. 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 73.
- **Group of Contextual Actions** adds a new context action group as described in Creating a Context Action Group on page 72.

- **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 70.

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

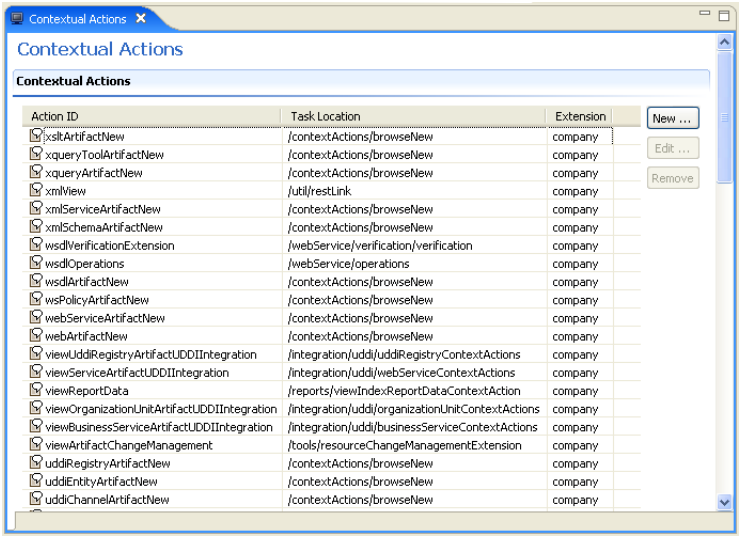
The following contextual action group parameters are displayed:

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

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 23. Contextual Actions Editor



On the right are the following buttons:

- **New** opens the **New Action** dialog described in Creating a Contextual Action on page 70.
- **Edit** opens the **Edit Action** dialog with the same parameters as described in Creating a Contextual Action on page 70.
- **Remove** deletes the selected context action from your extension project.

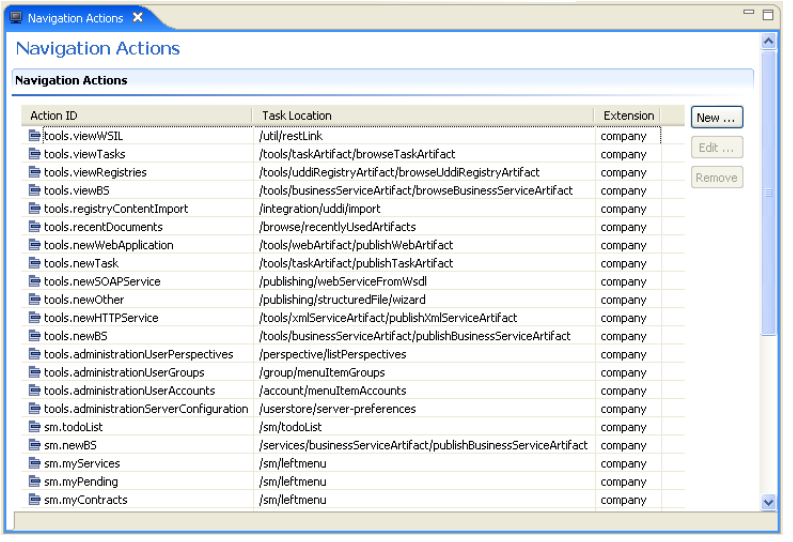
The following contextual action parameters are displayed:

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

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 24. Navigation Actions Editor



On the right are the following buttons:

- **New** opens the **New Action** dialog described in Creating a Navigation Action on page 68.
- **Edit** opens the **Edit Action** dialog with the same parameters as described in Creating a Navigation Action on page 68.
- **Remove** deletes the selected navigation action from your extension project.

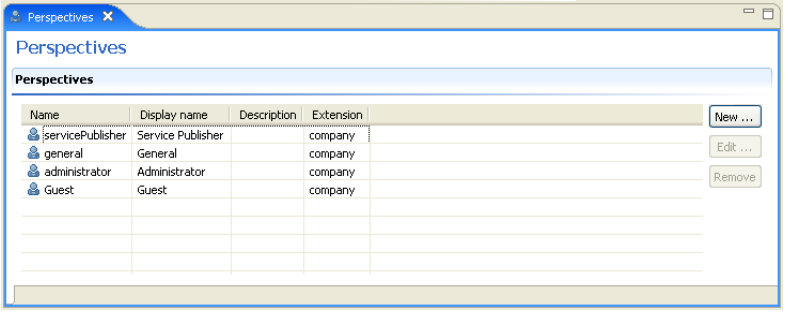
The following navigation action parameters are displayed:

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 25. Perspectives Editor



On the right are the following buttons:

- **New** opens the **New Perspective** dialog described in Creating a Perspective on page 74.
- **Edit** opens the **Edit Perspective** dialog for the selected perspective with the same parameters described in Creating a Perspective on page 74.
- **Remove** deletes the perspective from your extension project.

The following perspective parameters are displayed:

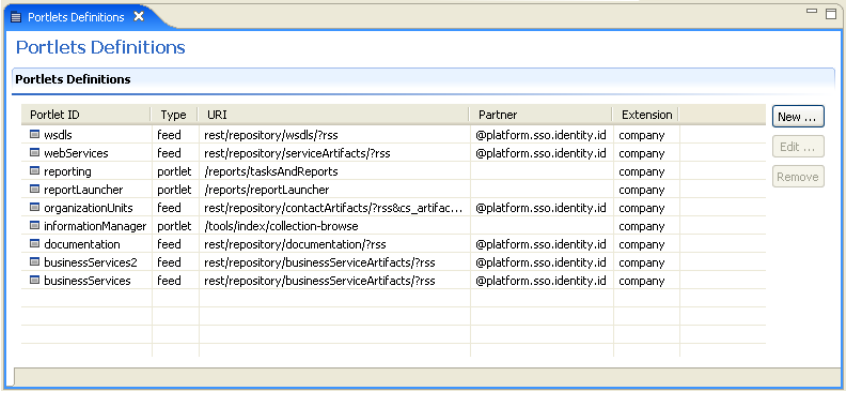
Parameter	Definition
Name	The name of the perspective as it is stored in the extension

Parameter	Definition
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 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 76.
  - **Portlet** adds a new portlet as described in Creating a Portlet on page 75.
- **Edit** opens the **Edit Feed** or **Edit Portlet** dialog for the selected item with the same parameters as **Add** functionality.

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

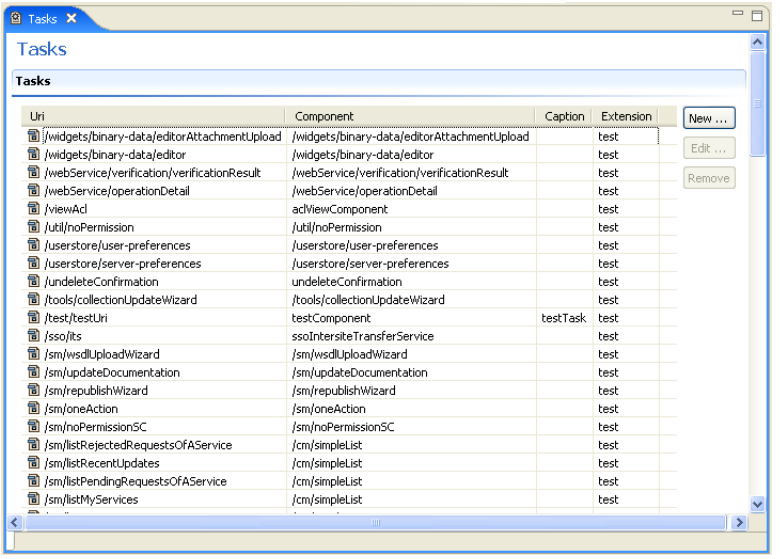
The following portlet parameters are displayed and described in the table below:

Parameter	Definition
Name	The name of the portlet as it is stored in the extension
Type	The type of portlet with options: • Feed • Portlet
URI	The identifier for the portlet descriptor in the configuration
Partner	The SSO partner identification
Extension	Which extension this perspective is part of

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 27. Tasks Editor



On the right are the following buttons:

- **New** opens the **New Task** dialog described in Creating a Task on page 81.
- **Edit** opens the **Edit Task** dialog with the same parameters described in Creating a Task on page 81.
- **Remove** deletes the selected task from your extension project.

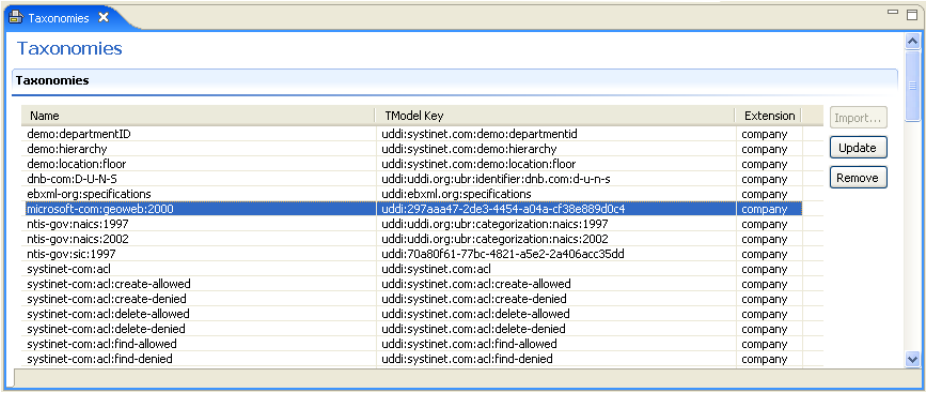
There are the following parameters related to tasks:

Parameter	Definition
Uri	The identifier for the task
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 28. Taxonomies Editor



On the right are the following buttons:

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

  Note: 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.

There are the following parameters related to taxonomies described in the table below:

Parameter	Definition
Name	The name of the taxonomy as it is stored in the extension

Parameter	Definition
TModel Key	The taxonomy key identifier
Extension	Which extension this taxonomy is part of

project.xml Editor

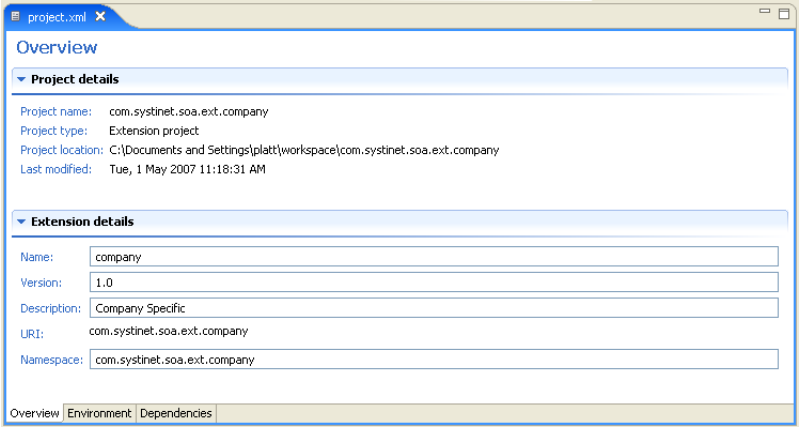
The project editor allows you to configure your extension project. It contains three tabs described in the following sections:

- Project – Overview Tab on page 130
- Project – Environment Tab on page 131
- Project – Dependencies Tab on page 133

Project – Overview Tab

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

Figure 29. Project Overview Tab



The tab contains the following collapsible segments:

- **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

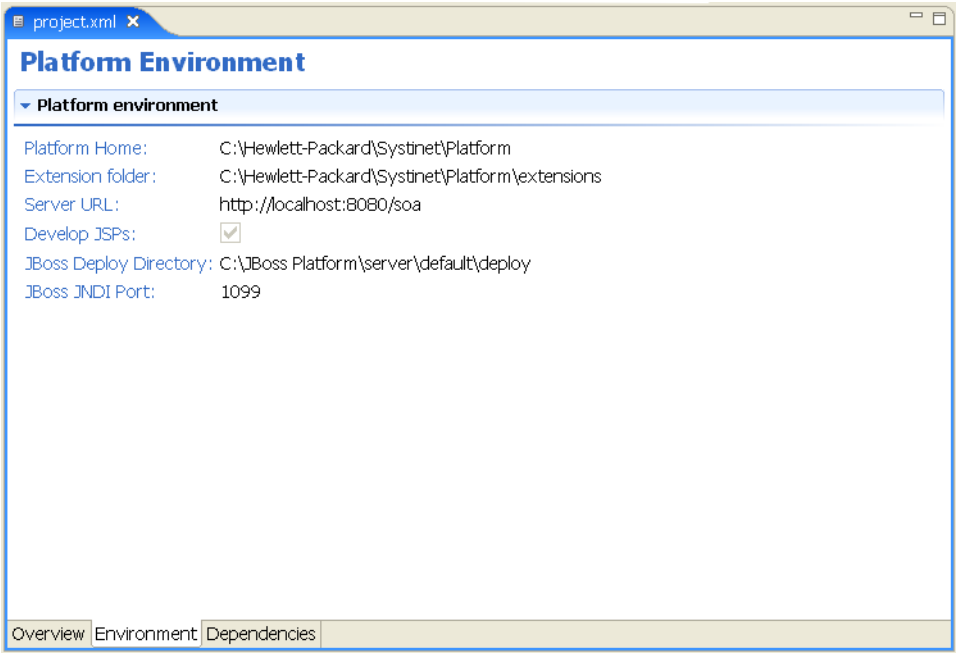
- **Extension details** enables you to configure some of the extension project parameters:

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 – Environment Tab

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

Figure 30. Project Environment Tab



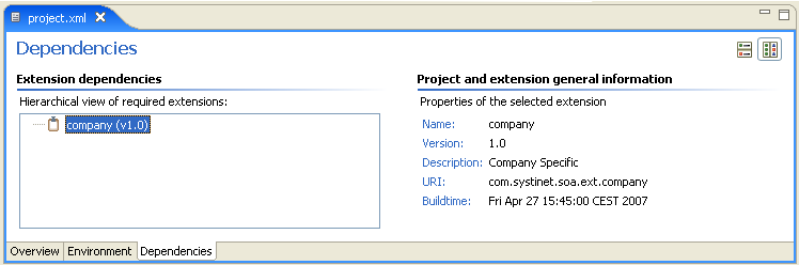
This tab contains parameters:

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 – Dependencies Tab

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

Figure 31. Project Dependencies Tab



The tab contains the following segments:

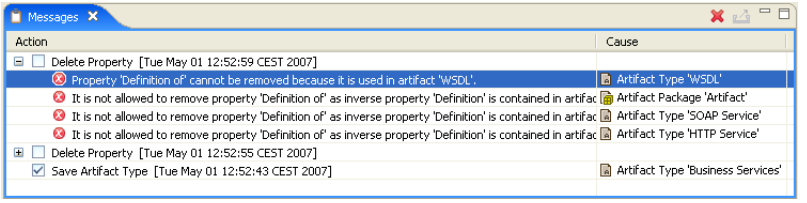
- **Extension dependencies** displays a tree of extensions that the extension project is dependent on.
- **Project and extension general information** displays parameters of the extension project:

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

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 32. Messages View

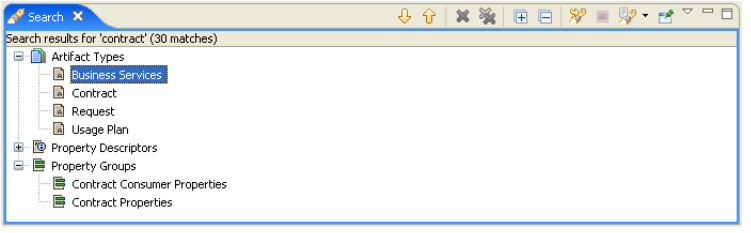


Double-clicking an item in the tree opens the relevant editor for that object.

Search View

The **Search** view displays the results of a search (see Searching the Extension on page 28) or find usage query (see SDM Details on page 24):

Figure 33. Search View



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

Double-clicking an artifact type, property or group opens the relevant editor.

Index

I

M

N

P

R

S

T

U

V