HP SOA Systinet Workbench

Software Version: 4.00

Customization Editor Guide

Document Release Date: September 2010
Software Release Date: September 2010

Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Copyright Notices

Contents

About this Guide

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

This guide contains the following chapters:

- Chapter 1, Customization Editor

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

- Chapter 2, Getting Started

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

- Chapter 3, Manipulating Artifact Types

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

- Chapter 4, Manipulating Properties

  Explains how to create and modify properties and property groups.

- Chapter 5, Creating and Using Components

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

- Chapter 6, Modifying Taxonomies

  Explains how to modify the set of available taxonomies.

- Chapter 7, Deploying Extensions

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

- Appendix A, Views

Describes Customization Editor views in detail.

- Appendix B, Dialog Boxes

  Describes Customization Editor dialog boxes in detail.

- Appendix C, Localname Naming Rules

  Explains localname naming conventions.

- Appendix D, Eclipse Plug-in Requirements

  Required plug-ins when installing Workbench as an update.

Document Conventions

This document uses the following typographical conventions:

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

Documentation Updates

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

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

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

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

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

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

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

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

Support

You can visit the HP Software Support Web site at:

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

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

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

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

To find more information about access levels, go to:

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

To register for an HP Passport ID, go to:

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

1 Customization Editor

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

This chapter introduces Customization Editor in the following sections:

- Workbench Suite on page 11
- Overview on page 12
- User Interface on page 13
- SDM Elements on page 22

Workbench Suite

HP SOA Systinet Workbench is a suite of editor tools enabling you to customize your deployment of HP SOA Systinet 4.00.

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

- **Customization Editor**

  Customizes the underlying SOA Definition Model (SDM) within HP SOA Systinet.

- **Taxonomy Editor**

  Customizes the taxonomies used to categorize artifacts in HP SOA Systinet.

- **Assertion Editor**

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

- **Report Editor**

Customizes report definitions for use with HP SOA Systinet.

Overview

HP SOA Systinet is distributed with a preconfigured data model. For typical deployment scenarios, the model should be customized to fit an organization-specific environment.

HP SOA Systinet Customization Editor provides a mechanism to customize the model, 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 HP SOA Systinet.

To customize HP SOA Systinet, follow this process:

1. Create an extension project.

   For details, see Creating an Extension Project on page 32.

2. Modify the data model.

   For details, see the following:

   - Chapter 3, Manipulating Artifact Types
   - Chapter 4, Manipulating Properties
   - Chapter 5, Creating and Using Components
   - Chapter 6, Modifying Taxonomies

3. Deploy the customization to HP SOA Systinet.

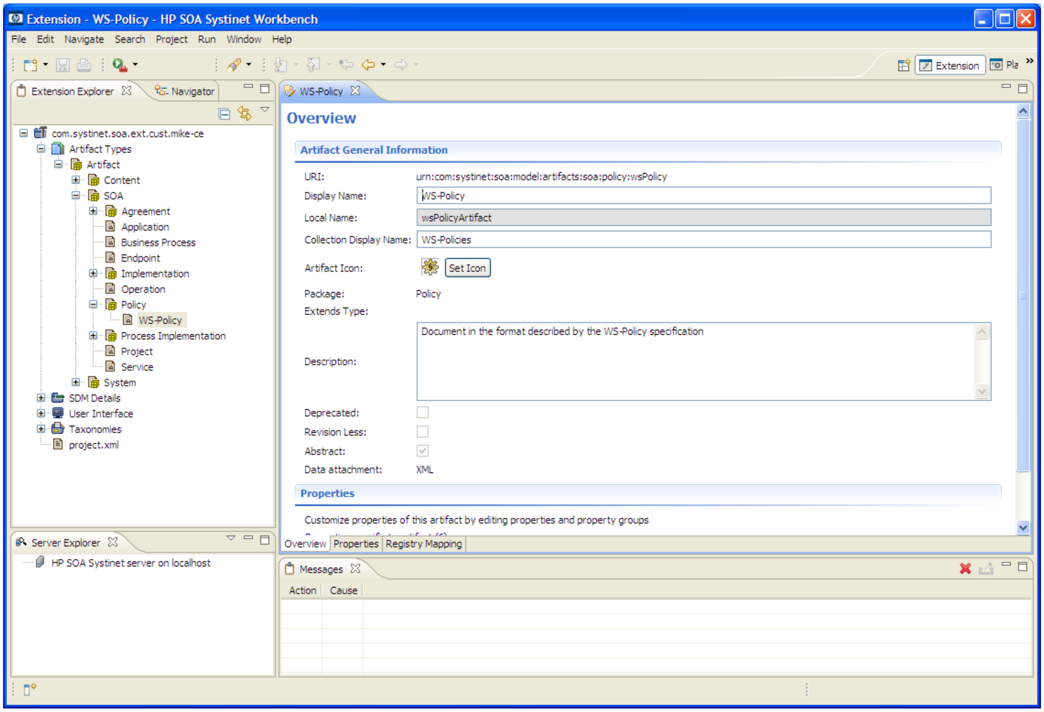
   For details, see Chapter 7, Deploying Extensions.

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

User Interface

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

Figure 1. Customization Editor UI



URI: urn:com:systinet:soa:model:artifacts:soa:policy:wsPolicy

Display Name: WS-Policy

Local Name: wsPolicyArtifact

Collection Display Name: WS-Policies

Artifact Icon:

Package: Policy

Extends Type:

Description: Document in the format described by the WS-Policy specification

Deprecated: □

Revision Less: □

Abstract: ☑

Data attachment: XML

Customize properties of this artifact by editing properties and property groups

The perspective consists of the following elements:

- **Project Explorer**

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

  For details, see Extension Explorer on page 14.

- **Server Explorer**

The view listing HP SOA Systinet server connections to Workbench.

For details see, Server Explorer on page 20.

- **Editor Views**

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

  For details, see Appendix A, Views.

- **Information Views**

  Messages and search results open their own views.

  For details, see Appendix A, Views.

- **Menu Items**

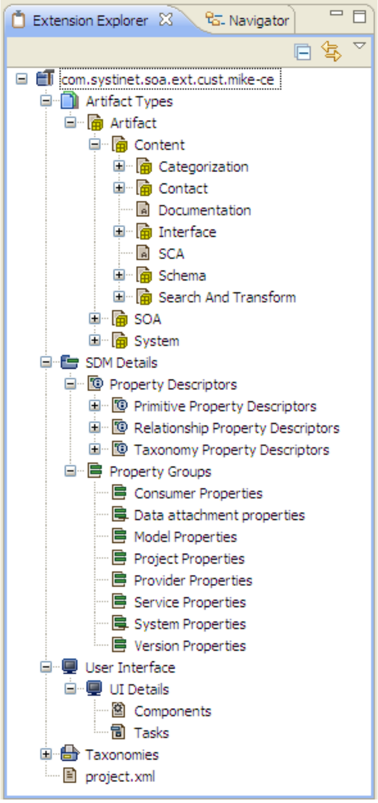
  The standard Eclipse menu options plus additional Workbench options.

  For details about the additional options for Customization Editor, see Menu Options on page 21.

Extension Explorer

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

Figure 2. Extension Explorer View



- com.systinet.soa.ext.cust.mike-ce
  - Artifact Types
    - Artifact
      - Content
        - Categorization
        - Contact
        - Documentation
        - Interface
        - SCA
        - Schema
        - Search And Transform
      - SOA
      - System
  - SDM Details
    - Property Descriptors
      - Primitive Property Descriptors
      - Relationship Property Descriptors
      - Taxonomy Property Descriptors
    - Property Groups
      - Consumer Properties
      - Data attachment properties
      - Model Properties
      - Project Properties
      - Provider Properties
      - Service Properties
      - System Properties
      - Version Properties
  - User Interface
    - UI Details
      - Components
      - Tasks
  - Taxonomies
  - project.xml

The tree is split into the extension root and five branches representing different elements of your extension.

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

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

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

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

- Extension Name on page 16
- Artifact Types on page 16
- SDM Details on page 17
- User Interface on page 19
- Taxonomies on page 19
- project.xml on page 20

Extension Name

The extension name is the root element of the project.

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

For deployment details, see Chapter 7, Deploying Extensions.

Artifact Types

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

⚠ HP SOA Systinet separates the SDM into two models, public and system. Customization Editor hides system artifacts, preventing you from modifying them.

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

For details, see Artifacts on page 23.

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

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

Creates an artifact type as described in Creating an Artifact Type or Package on page 37.

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

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

  For details, see Artifact Editor on page 63.

- **Deprecate Artifact Type or Package**

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

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

  Reactivates a deprecated artifact type in the extension project.

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

  Removes the artifact type from the extension project.

SDM Details

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

- Property Descriptors on page 17
- Property Groups on page 18

HP SOA Systinet separates the SDM into two models, public and system. Customization Editor shows properties and property groups used by system artifacts, but prevents you from modifying them.

Property Descriptors

Property descriptors are the attributes of artifact types.

For details, see Properties on page 23.

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

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

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

For details, see Creating a Property on page 45.

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

The context menu for properties contains the following options:

- **Open Property**

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

  For details, see Property Editor on page 75.

- **Deprecate Property**

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

- **Undeprecate Property**

  Reactivates the deprecated property in the extension project.

- **Delete Property**

  Removes the property from the extension project.

- **Find Usage**

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

  For details, see Search View on page 80.

Property Groups

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

For details, see Property Groups on page 24.

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

For details, see Creating a Property Group on page 47.

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

- **Open Property Group**

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

  For details, see Property Group Editor on page 78.

- **Delete Property Group**

  Removes the property group from the extension project.

- **Find Usage**

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

  For details, see Search View on page 80.

User Interface

UI Details enables you to create new components and tasks for use in the HP SOA Systinet UI.

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

- Creating a Component on page 49
- Creating a Task on page 50

Taxonomies

Taxonomies are categorization groups.

For details, see Taxonomies on page 25.

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

For details, see Taxonomies Editor on page 82.

You can create a new taxonomy from the context menu.

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

The New Taxonomy dialog opens. For details see New Taxonomy on page 101.

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

project.xml

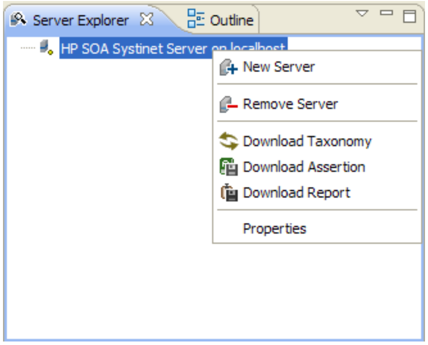
Open the project.xml view to configure the current extension project.

For details, see project.xml Editor on page 71.

Server Explorer

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

Figure 3. Server Explorer View



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

Table 1. Server Explorer Context Menu Options

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

Menu Options

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

The additional menu options for Customization Editor include:

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

  Create a new project, as described in Creating an Extension Project on page 32.

- **File>New>Artifact Type**

  Create an artifact type, as described in Creating an Artifact Type or Package on page 37.

- **File>New>Artifact Package**

  Create an artifact package, as described in Creating an Artifact Type or Package on page 37.

- **File>New>Property**

  Create a property, as described in Creating a Property on page 45.

- **File>New>Property Group**

  Create a property group, as described in Creating a Property Group on page 47.

- **File>New>Component**

  Create a component, as described in Creating a Component on page 49.

- **File>New>Task**

  Create a task, as described in Creating a Task on page 50.

- **Navigate>Go To>Open Artifact**

  Open an artifact editor, as described in Artifact Editor on page 63.

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

- **Navigate>Go To>Open Property**

  Open a property editor, as described in Property Editor on page 75.

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

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

  Open a property group editor, as described in Property Group Editor on page 78.

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

- **Search>Search**

  Open a search dialog box, as described in Searching the Extension on page 35.

- **Help>Dynamic Help**

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

SDM Elements

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

These entity types are described in the following sections:

- Artifacts on page 23
- Properties on page 23
- Property Groups on page 24
- Taxonomies on page 25

Artifacts

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

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

For details, see Artifact Types on page 16.

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

Properties

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

There are three categories of property:

- **Primitive Properties**

  The basic labels for artifacts such as name and description.

  Basic properties are of the following types:

Property Type	Description
addressPropertyType	A full postal address.
booleanPropertyType	True or false.

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

- **Relationship Properties**

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

- **Taxonomy Properties**

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

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

For details, see SDM Details on page 17.

For property procedures, see Chapter 4, Manipulating Properties.

Property Groups

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

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

For details, see SDM Details on page 17.

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

Taxonomies

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

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

For details, see Taxonomies on page 19.

For taxonomy procedures, see Chapter 6, Modifying Taxonomies.

2 Getting Started

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

This chapter contains the following sections:

Installing Workbench

HP SOA Systinet Workbench is an Eclipse development platform distributed as a zip file, `hp-soa-systinet-workbench-4.00-win32.zip` or as a plugin for an existing Eclipse environment, `hp-soa-systinet-workbench-4.00-plugin.zip`.

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

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

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

  The path must not be longer than 97 characters.

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

1 Ensure that your Eclipse platform contains the necessary plug-ins. For details, see Appendix D, Eclipse Plug-in Requirements.

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

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

   The Install/Update dialog opens.

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

   The Install – Update Sites to Visit dialog opens.

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

   The Select Local Archive Site dialog opens.

5 Locate and select `hp-soa-systinet-workbench-4.00-plugin.zip`, and then click **Open**.

   The Edit Local Site dialog opens.

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

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

   The Updates – Search Results dialog opens.

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

- **Workbench Extra 4.00**

  HP SOA Systinet Workbench splash screen.

- **Taxonomy Editor 4.00**

  Required for Customization Editor.

- **Customization Editor 4.00**
- **Assertion Editor 4.00**
- **Report Editor 4.00**
- **Common Plugin 4.00**

  Shared components used by the editors.

Click **Next**.

The Install – Feature License dialog opens.

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

   The Install – Installation dialog opens.

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

11. If you install `Workbench Extra 4.00` make the following configuration changes:

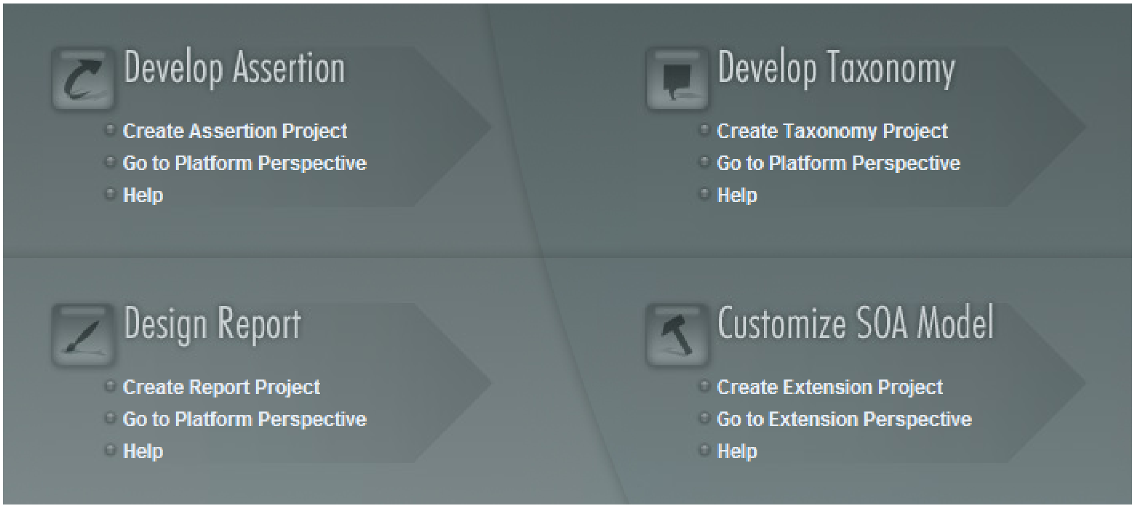
    - Remove `-showsplash org.eclipse.platform` from `ECLIPSE_HOME/eclipse.ini`.
    - Edit ECLIPSE_HOME/configuration/config.ini and make the following changes:
      - Set `osgi.splashPath=platform:/base/plugins/com.systinet.tools.workbench` .
      - Set `eclipse.product=com.systinet.tools.workbench.ide`

To start HP SOA Systinet Workbench:

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

The first time you start Workbench, the welcome screen opens, as shown in Figure 4, “Workbench Welcome Screen”.

Figure 4. Workbench Welcome Screen



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

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

By default, Workbench runs in 'normal' mode which prevents users from uploading system taxonomies (IDs start with *uddi:systinet.com:soa:model:taxonomies*) and the Report Editor .rptlibrary file to HP SOA Systinet servers. If you need to work with system taxonomies or want to upload the .rptlibrary file you can switch Workbench into 'admin' mode.

Be extremely careful when working with system taxonomies, HP SOA Systinet uses some hard-coded values from system taxonomies, changing or removing them may cause errors.

To Switch Workbench to Admin Mode

1 Open `WB_HOME/configuration/config.ini` with a text editor.

2 Add *mode=admin* to `config.ini`.

3. Restart Workbench.

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

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

To increase the memory allocation for HP SOA Systinet Workbench:

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

SSL Configuration

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

To verify HP SOA Systinet server certificates:

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

```
-Dcom.hp.systinet.security.ssl.verifyCert=true
-Djavax.net.ssl.trustStore=USER_TRUSTSTORE
-Djavax.net.ssl.trustStorePassword=TRUSTSTORE_PASS
-Djavax.net.ssl.trustStoreType=TRUSTSTORE_FORMAT
```

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

To provide Workbench client certificates to HP SOA Systinet:

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

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

Creating an Extension Project

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

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

- **model**

  Extensions for changes to the data model.

- **code**

  Extensions for custom components.

- **mixed**

  Combined model and code extensions.

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

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

To create an extension project:

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

   The New Project wizard opens.

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

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

   For parameter descriptions, see New Extension Project: Specify HP SOA Systinet Server Installation on page 88.

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

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

5. Do one of the following:

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

     Continue to Step 6.

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

     Continue to Step 7.

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

   For parameter descriptions, see New Extension Project: New Server on page 90.

7. Do one of the following:

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

     Click **Next**.

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

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

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

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

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

  Click **Next**.

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

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

   For parameter descriptions, see New Project: Create a New Extension on page 93.

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

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

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

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

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

For details about the contents of your extension project, see Extension Explorer on page 14.

Sharing an Extension Project

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

To share an extension project:

1. In Extension Explorer, right-click the project you need to open its context menu, and select **Team>Share Project** to open the **Share Project** dialog box. For details, see Share Project on page 105.

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

   The project is published to the CVS repository.

Referencing Other Workbench Projects

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

To reference an Assertion or Taxonomy Project:

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

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

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

Searching the Extension

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

To search your extension project:

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

   The Search dialog box opens.

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

   For parameter descriptions, see Search on page 104.

3. Click **Search**.

The search results appear in the Search view.

For details, see Search View on page 80.

Saving Modifications

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

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

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

Your modifications are made to your extension project and not to the configuration of HP SOA Systinet. To deploy your customization to HP SOA Systinet, see Chapter 7, Deploying Extensions.

3 Manipulating Artifact Types

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

⚠ HP SOA Systinet separates the SDM into two models, public and system. Customization Editor hides system artifacts, preventing you from modifying them.

For more details about artifact types and packages, see Artifacts on page 23.

The chapter explains the following procedures:

- Creating an Artifact Type or Package on page 37
- Modifying the Attributes of an Artifact Type on page 38
- Modifying the Properties of an Artifact Type on page 39
- Mapping an Artifact Type to a Registry on page 41
- Example: Adding the Department Property to Services on page 43

Creating an Artifact Type or Package

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

To create an artifact type or package:

1. Do one of the following:
   - From the main menu, select **New>Artifact Type** or **Package**.
   - In the Extension Explorer, open the context menu of an artifact package branch and select **New Artifact Type** or **Package**.

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

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

   For parameter descriptions, see New Artifact: Create on page 84.

   For localname naming conventions, see Appendix C, Localname Naming Rules.

3. Do one of the following:

   - Click **Finish** to create the artifact type or package and exit the dialog box.
   - Click **Next** and continue to Step 4 to set database parameters.

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

   For parameter descriptions, see New Artifact: Database Settings on page 86.

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

Modifying the Attributes of an Artifact Type

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

To edit the attributes of an artifact type:

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

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

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

   For a description of the Overview tab, see Artifact Editor: Overview Tab on page 64.

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

Modifying the Properties of an Artifact Type

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

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

1. Open the artifact editor and select the **Properties** tab.
2. In the Properties tab, make any changes you want.

   For a description of the Properties tab, see Artifact Editor: Properties Tab on page 66.
3. Press **Ctrl+S** to save your changes.

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

To change the local name of a property descriptor:

1. From the tree menu of Extension Explorer, expand **SDM Details>Property Descriptors**.
2. Browse for the descriptor you want, and double-click to open the **Overview** view.
3. In the **Local Name** field, enter the parameters you require, and then save your changes.

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

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

To change the cardinality of an artifact type:

1. Browse for the artifact you need and open the Editor view, and then switch to the **Properties** tab.
2. In the **Cardinality** field, click to open the drop-down menu, and select the cardinality you need.

   ▶ For taxonomic properties, the cardinality affects the way the property displays in the HP SOA Systinet UI.

- Required taxonomic properties display as an **Add Category** function.
- Optional taxonomic properties display as a drop-down menu.
- Multiple taxonomic properties display as a list box.

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

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

To change the direction of a relationship property descriptor:

1. In Extension Explorer, expand **SDM Details>Property Descriptors>Relationship Property Descriptors** and double-click the descriptor you need to open its Editor view.
2. In the **Direction** field, click the link next to **Inverse Display Name** to toggle the relationship between **incoming** or **outgoing**, and then save your changes.

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

To modify the taxonomical properties of an artifact:

1. In Extension Explorer, browse for the artifact you need and open its Editor view, and then switch to the **Properties** tab.
2. In the Properties window, click **New** to open the **New Property** dialog box, and select **Taxonomy Property**, and then click **Next**.

   The **New Taxonomy Property** dialog box opens. For details, see New Property: Taxonomy on page 98.
3. Enter general parameters and in the **Taxonomy Name** field, do one of the following:
   - Click **Browse** to browse for and select from a list of available taxonomies.

- Click **Create** to open the Create a New Taxonomy dialog box,. For details, see New Taxonomy on page 101.

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

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

5 Click **Finish**.

Mapping an Artifact Type to a Registry

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

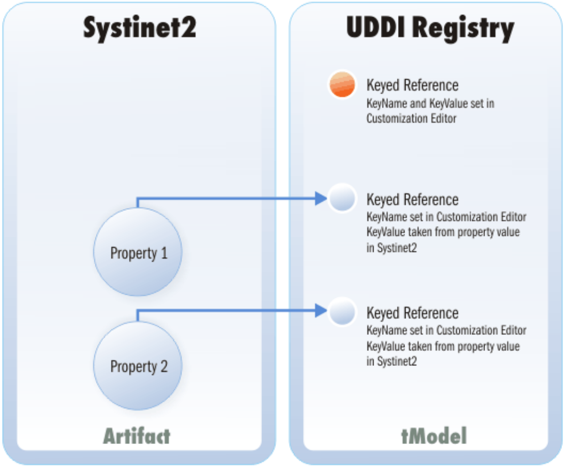
The artifact types listed in Table 2, "Predefined Registry Mapping" have predefined mappings to registry and cannot be amended.

Table 2. Predefined Registry Mapping

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

The mapping of an artifact to a registry entity is illustrated in Figure 5, “Registry Mapping”.

Figure 5. Registry Mapping


Systinet2
UDDI Registry
Keyed Reference
KeyName and KeyValue set in Customization Editor
Property 1
Keyed Reference
KeyName set in Customization Editor
KeyValue taken from property value in Systinet2
Property 2
Keyed Reference
KeyName set in Customization Editor
KeyValue taken from property value in Systinet2
Artifact
tModel

To add or modify artifact type mappings to registry entities:

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

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

   For a description of the Registry Mapping tab, see Artifact Editor: Registry Mapping Tab on page 69.

   For details on how to add an artifact mapping, see Adding a Registry Mapping for an Artifact Type on page 42.

   For details on how to add a property mapping, see Adding a Registry Mapping for a Property on page 43.

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

Adding a Registry Mapping for an Artifact Type

You can map artifact types to registry entities.

To add an artifact mapping:

1. Open the Artifact Editor and select the **Registry Mapping** tab.
2. In the Artifact Mapping pane, click **Add**.

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

   For parameter descriptions, see Registry Mapping: New Artifact Mapping on page 102.
4. Click **Finish** to add the mapping.

Adding a Registry Mapping for a Property

You can map properties to registry entity keyed references.

To add a property mapping:

1. Open the Artifact Editor and select the **Registry Mapping** tab.
2. In the Property Mapping pane, click **Add**.

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

   For parameter descriptions, see Registry Mapping: New Property Mapping on page 104.
4. Click **Finish** to add the mapping.

Example: Adding the Department Property to Services

In this example, you add a new taxonomic property, department, to the service artifact.

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

- In Customization Editor, create a new extension dependent on the core extension, as described in Creating an Extension Project on page 32.

- In Taxonomy Editor, create a new department taxonomy and deploy it to HP 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 services:

1. Open theService editor and select the **Properties** tab.
2. In the Properties segment, click **New**.
3. Select **Taxonomy Property** and click **Next**.
4. Input `Department` as the **Display Name**.
5. For the **Taxonomy**, click **Browse** and import the **Departments** taxonomy.
6. Press **Ctrl+S** to save your changes to the business service artifact type.
7. Deploy your extension, as described in Chapter 7, Deploying Extensions.
8. Start your installation of HP SOA Systinet.

Add the property to the artifact in the user interface. For details, see "UI Customization" in the *HP SOA Systinet Administration Guide*.

4 Manipulating Properties

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

HP SOA Systinet separates the SDM into two models, public and system. Customization Editor shows properties and property groups used by system artifacts, but prevents you from modifying them.

For more details about properties and property groups, see Properties on page 23 and Property Groups on page 24.

This chapter describes the following procedures:

- Creating a Property on page 45
- Modifying a Property on page 46
- Creating a Property Group on page 47
- Modifying a Property Group on page 47

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

- Modifying the Properties of an Artifact Type on page 39
- Mapping an Artifact Type to a Registry on page 41

Creating a Property

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

To create a property:

1 Do one of the following:

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

The New Property dialog box opens.

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

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

- New Property: Primitive on page 95
- New Property: Relationship on page 96
- New Property: Taxonomy on page 98

For localname naming conventions, see Appendix C, Localname Naming Rules.

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

Modifying a Property

Customization Editor enables you to modify properties.

To edit a property:

1 Open the Property Editor.

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

For a description of the Property Editor, see Property Editor on page 75.

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

Creating a Property Group

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

To create a property group:

1. Do one of the following:

   - From the menu, select **File>New>Property Group.**
   - In the Extension Explorer, open the context menu for the Property Groups branch, and select **New Property Group**.
   - Open the Artifact Editor and select the Properties tab. In the Property Groups pane, click **New** to create a new property group as a set of attributes for an artifact type.

   The New Property Group dialog box opens.

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

   For parameter descriptions, see New Property Group on page 100.

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

Modifying a Property Group

Customization Editor enables you to modify property groups.

To edit a property group:

1. Open the property group editor.
2. In the Property Group Editor, make any changes you want.

   For a description of the Property Group Editor, see Property Group Editor on page 78.

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

5 Creating and Using Components

Customization Editor enables you to create custom components, for use in the customizable HP SOA Systinet user interface.

To create and use components:

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

For details, see as described in Creating an Extension Project on page 32.

2 Create a component.

For details, see Creating a Component on page 49.

3 Create a task to use your component.

For details, see Creating a Task on page 50.

4 Develop the Java and JSP that the component uses.

For details, see Developing a Component on page 51.

5 Deploy the code extension to HP SOA Systinet.

For details, see Chapter 7, Deploying Extensions.

6 Perform the appropriate UI customization to use your component in HP SOA Systinet.

For details, see "UI Customization" in the *HP SOA Systinet Administration Guide*.

Creating a Component

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

To create a component:

1 Do one of the following:

   - In the Extension Explorer, open the context menu for the Components branch and select **New Component**.
   - In the Components Editor, click **Add**.

   The New Component dialog box opens.

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

   For parameter definitions, see New Component on page 87.

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 HP SOA Systinet UI, and to make more sophisticated multi-layered components.

To create a task:

1 Do one of the following:

   - In the Extension Explorer, open the context menu for the Tasks branch and select **New Task**.
   - In the Tasks editor, click **Add**.

   The New Task dialog box opens.

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

   For parameter descriptions, see New Task on page 100.

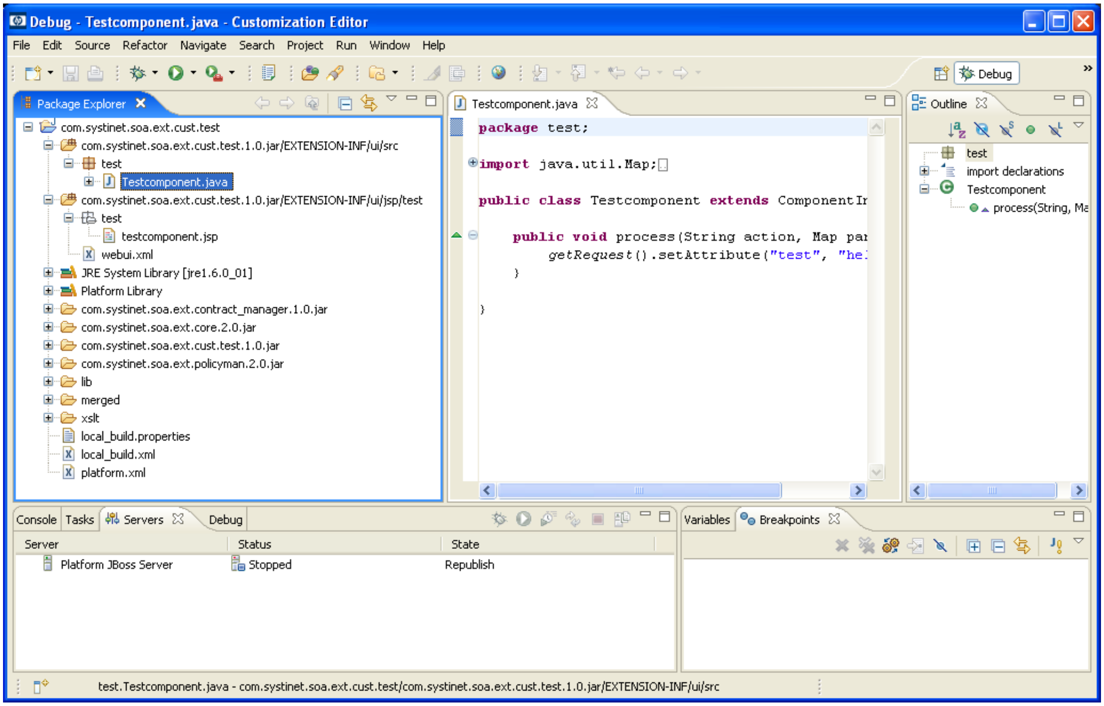
3 Click **Finish** to create the task.

Developing a Component

When you create a component, as described in Creating a Component on page 49, the templates for the Java class and JSP page are created automatically.

To view this code, switch to the Java or Debug perspective, as shown in Figure 6, “Debug Perspective”.

Figure 6. Debug Perspective



```
com.systinet.soa.ext.cust.test
    com.systinet.soa.ext.cust.test.1.0.jar/EXTENSION-INF/ui/src
        test
            Testcomponent.java
    com.systinet.soa.ext.cust.test.1.0.jar/EXTENSION-INF/ui/jsp/test
        test
            testcomponent.jsp
        webui.xml
    JRE System Library [jre1.6.0_01]
    Platform Library
    com.systinet.soa.ext.contract_manager.1.0.jar
    com.systinet.soa.ext.core.2.0.jar
    com.systinet.soa.ext.cust.test.1.0.jar
    com.systinet.soa.ext.policyman.2.0.jar
    lib
    merged
    xslt
    local_build.properties
    local_build.xml
    platform.xml
```

```
package test;

import java.util.Map;□

public class Testcomponent extends ComponentIr

    public void process(String action, Map par
        getRequest().setAttribute("test", "he]
    }

}
```

```
test
    import declarations
    Testcomponent
        process(String, Ma
```

Server	Status	State
Platform JBoss Server	Stopped	Republish

test.Testcomponent.java - com.systinet.soa.ext.cust.test/com.systinet.soa.ext.cust.test.1.0.jar/EXTENSION-INF/ui/src

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

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

6 Modifying Taxonomies

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

Creating and modifying taxonomies uses the Taxonomy Editor functionality.

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

To control taxonomies in your extension project:

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

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

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

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

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

     You can also import taxonomies in the Server Explorer view. For more details, see Server Explorer on page 20.

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

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

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

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

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

For details, see Creating an Extension Project on page 32.

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

To change the taxonomy in a taxonomy based property descriptor:

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

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

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

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

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

7 Deploying Extensions

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

1 Exporting the Extension Project on page 55

2 Applying Extensions on page 56

3 Redeploying the EAR File on page 60

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

For details, see Deploying Components to HP SOA Systinet on page 61.

Exporting the Extension Project

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

To create your extension package:

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

The Export Extension dialog box opens.

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

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

For configuration details, see Creating an Extension Project on page 32.

If the extension already exists, Customization Editor prompts you to confirm that you want to overwrite it.

Applying Extensions

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

Extensions to HP SOA Systinet come from the following sources:

- **Customization Editor**

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

  If your extension contains new artifact types, HP SOA Systinet does not create default ACLs for them. Set default ACLs for the new artifact types in HP SOA Systinet using the functionality described in "How to Manage Default Access Rights" in the *Administration Guide*.

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

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

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

Apply extensions according to one of the following scenarios:

- Single-Step Scenario on page 58

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

- Decoupled DB Scenario on page 59

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

In some specific circumstances (underscores and numbers in property names), extension application may fail because HP SOA Systinet cannot create short enough database table names (31 character maximum for most databases).

The error in `setup.log` resembles the following:

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

 BlizzardNameProviderImpl.getUniqueLimitedLengthName(BlizzardNameProviderImpl.java:432
[java] at com.systinet.platform.rdbms.design.decomposition.naming.impl.
  BlizzardNameProviderImpl.filterTableName(BlizzardNameProviderImpl.java:374)
```

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

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

To change the table naming algorithm:

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

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

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

3. Save `rdbPlatformContext.xml`

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

Single-Step Scenario

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

To apply extensions to HP SOA Systinet in a single step:

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

   `SOA_HOME/extensions`

   The Setup Tool automatically applies all extensions in that directory.

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

2. Stop the server.

3. Start the Setup Tool by executing the following command:

   SOA_HOME/bin/setup.bat(sh)

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

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

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

   This process takes some time.

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

7. Deploy the EAR file:

   - **JBoss**

The Setup Tool deploys the EAR file automatically.

If you need to deploy the EAR file to JBoss manually, see Redeploying the EAR File on page 60.

- **Other Application Servers**

  You must deploy the EAR file manually.

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

8 Restart the server.

⚠

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

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

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

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

Decoupled DB Scenario

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

To apply extensions and modify the database separately:

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

```
SOA_HOME/extensions
```

The Setup Tool automatically applies all extensions in that directory.

2 Stop the server.

3 Start the Setup Tool by executing the following command:

SOA_HOME/bin/setup -a.

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

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

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

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

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

   SOA_HOME/bin/setup -c

8. Redeploy the EAR file:

   - **JBoss**

     The Setup Tool deploys the EAR file automatically.

     If you need to deploy the EAR file to JBoss manually, see Redeploying the EAR File on page 60.

   - **Other Application Servers**

     You must deploy the EAR file manually.

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

Redeploying the EAR File

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

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

To redeploy the EAR file to JBoss:

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

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

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

Deploying Components to HP SOA Systinet

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

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

To directly deploy an extension from Customization Editor:

1. Stop the application server.
2. Switch to the Debug perspective in Customization Editor.
3. From the main menu, select **Run>External Tools>Open External Tools Dialog**.

   The External Tools dialog box opens.
4. Select **Ant Build>Customization Editor – Apply Extension**.

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

The Console view displays the output of the deployment process.

A Views

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

- Artifact Editor on page 63. Manage your SOA artifacts and customize their properties, appearance, and registry mapping.
- Messages View on page 71. View the action log as you customize your extension project.
- project.xml Editor on page 71. Manage your extension project.
- Property Editor on page 75. Manage your SOA properties and which perspectives can see and edit them.
- Property Group Editor on page 78. Organize your properties into groups.
- Search View on page 80. View the results of usage and entity searches.
- Tasks Editor on page 80. Manage tasks in HP SOA Systinet.
- Taxonomies Editor on page 82. Customize the taxonomies available in HP SOA Systinet.

Artifact Editor

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

Each editor contains the following tabs described in the following sections:

- Artifact Editor: Overview Tab on page 64

  The main attributes of the artifact type.

- Artifact Editor: Properties Tab on page 66

  The properties of the artifact type.

- Artifact Editor: Registry Mapping Tab on page 69

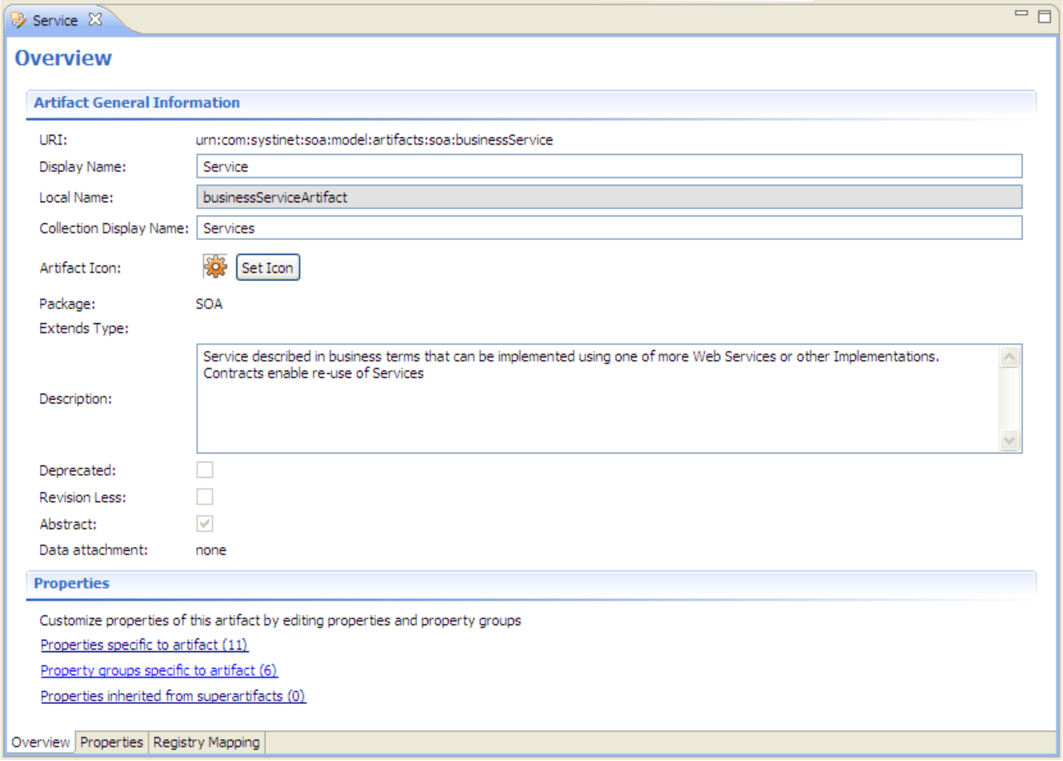
  The mapping of the artifact to registry entities.

Artifact Editor: Overview Tab

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

The Overview tab for the Business Service artifact type is shown in Figure 7, “Business Service Editor: Overview Tab”.

Figure 7. Business Service Editor: Overview Tab



Artifact General Information

URI: urn:com:systinet:soa:model:artifacts:soa:businessService

Display Name: Service

Local Name: businessServiceArtifact

Collection Display Name: Services

Artifact Icon:

Package: SOA

Extends Type:

Description: Service described in business terms that can be implemented using one of more Web Services or other Implementations. Contracts enable re-use of Services

Deprecated: ☐

Revision Less: ☐

Abstract: ☑

Data attachment: none

Properties

Customize properties of this artifact by editing properties and property groups

Properties specific to artifact (11)

Property groups specific to artifact (6)

Properties inherited from superartifacts (0)

The Overview tab contains the following segments:

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

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

Parameter	Definition
Collection Display Name	The plural version of the name as it appears in the HP SOA Systinet UI.
Artifact Icon	Displays the icon used for the artifact in the UI with an option to change it.
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 HP SOA Systinet UI.
Deprecated	A check box indicating whether the artifact is currently active in the extension.
Revisionless	Indicates whether the artifact creates a new revision when it updates.
Abstract	A check box indicating whether the artifact is an artifact package which does not have actual instances in the repository.
Data Attachment	Indicates the type of data content that may be attached to the artifact.

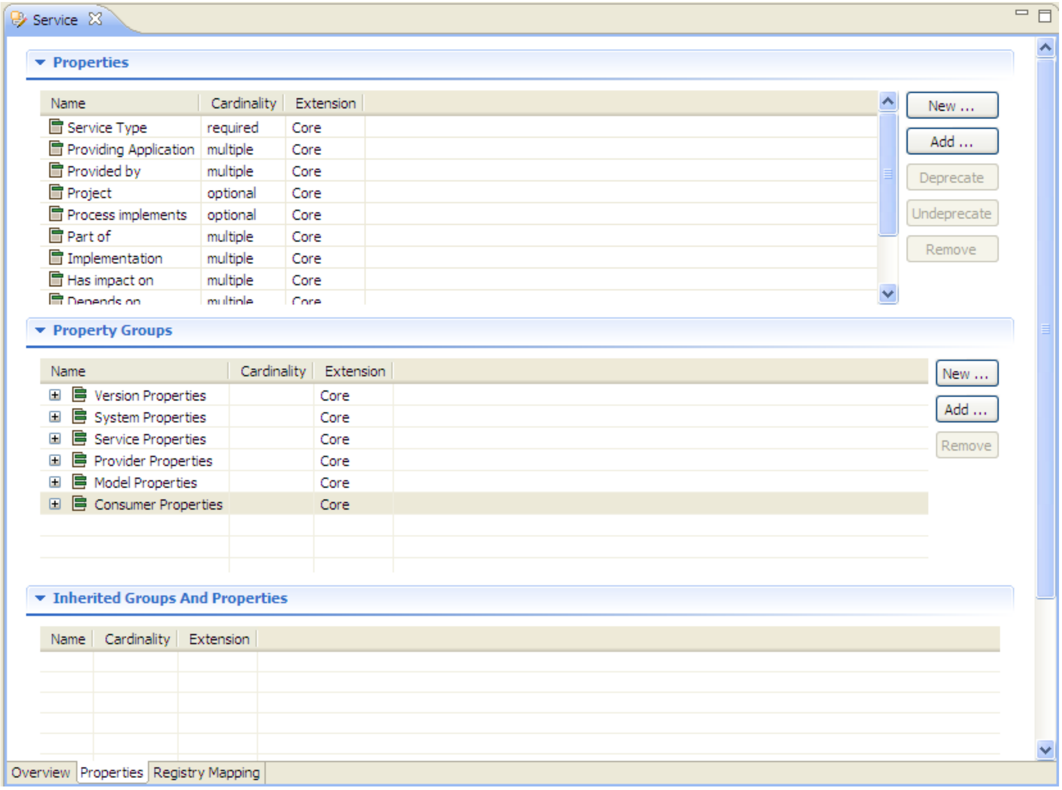
- **Properties**

  A summary of the different properties that the artifact has. Each of the links opens the **Properties** tab described in Artifact Editor: Properties Tab on page 66.

Artifact Editor: Properties Tab

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

Figure 8. Business Service Editor: Properties Tab



Name	Cardinality	Extension
Service Type	required	Core
Providing Application	multiple	Core
Provided by	multiple	Core
Project	optional	Core
Process implements	optional	Core
Part of	multiple	Core
Implementation	multiple	Core
Has impact on	multiple	Core
Depends on	multiple	Core

Name	Cardinality	Extension
Version Properties		Core
System Properties		Core
Service Properties		Core
Provider Properties		Core
Model Properties		Core
Consumer Properties		Core

Name	Cardinality	Extension

This tab contains the following panes:

- **Properties**

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

  - **New** opens the **New Property** dialog described in Creating a Property on page 45.
  - **Add** adds a property to the artifact type from the list of available property descriptors, set its cardinality, and its location on the **View Artifact** page.

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

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

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

- **Property Groups**

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

  - **New** opens the **New Property Group** dialog. For details, see Creating a Property Group on page 47.

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

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

- **Inherited Groups and Properties**

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

The following property parameters are displayed in these panes:

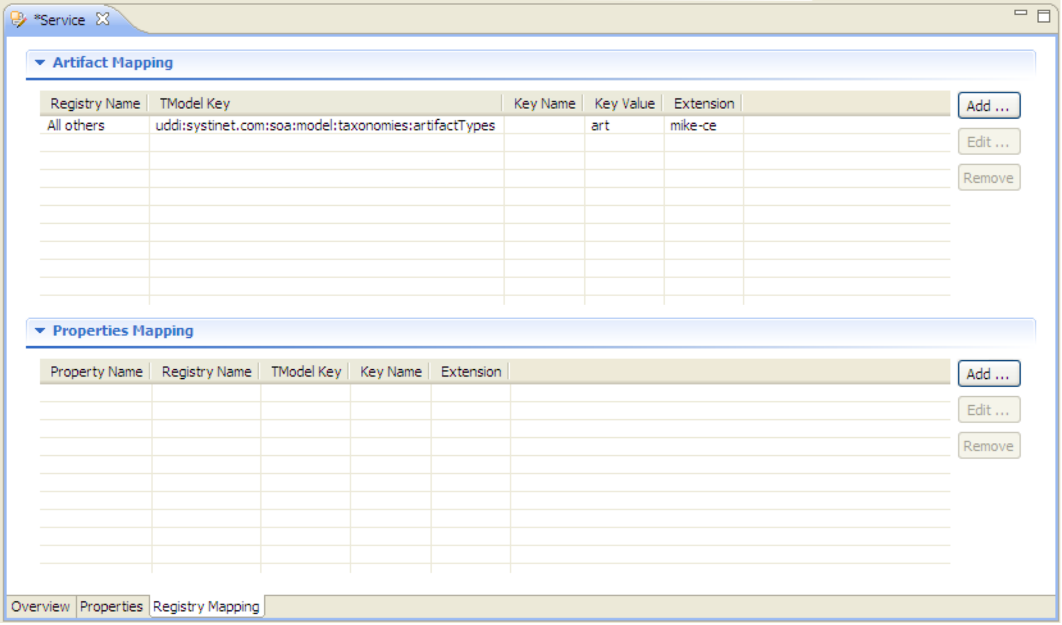
Parameter	**Definition**
Name	The name of the property as it appears in the HP SOA Systinet UI.
Cardinality	The occurrence of the property in an artifact with options: • **Optional** The property is not required to be populated. • **Required** The property must be populated. • **Multiple**

Parameter	Definition
	The cardinality is determined by the MinOccurs/MaxOccurs defined in the Property Editor: Advanced Tab. For details, see Property Editor: Advanced Tab on page 77.
Extension	Which extension this property is part of.

Artifact Editor: Registry Mapping Tab

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

Figure 9. Business Service: Registry Mapping Tab



Registry Name	TModel Key	Key Name	Key Value	Extension
All others	uddi:systinet.com:soa:model:taxonomies:artifactTypes		art	mike-ce

Property Name	Registry Name	TModel Key	Key Name	Extension

It is split into the following panes:

- **Artifact Mapping**

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

- **Add** adds a registry mapping, as described in Mapping an Artifact Type to a Registry on page 41.
- **Edit** edits the mapping, as described in Mapping an Artifact Type to a Registry on page 41.
- **Remove** removes the mapping from the artifact.

The following registry mapping parameters are displayed in this pane:

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

- **Property Mappings**

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

- **Add** adds a new registry mapping for the selected property, as described in Mapping an Artifact Type to a Registry on page 41.
- **Edit** edits a property mapping, as described in Mapping an Artifact Type to a Registry on page 41.
- **Remove** removes a mapping from the selected property.

The following registry mapping parameters are displayed in this pane:

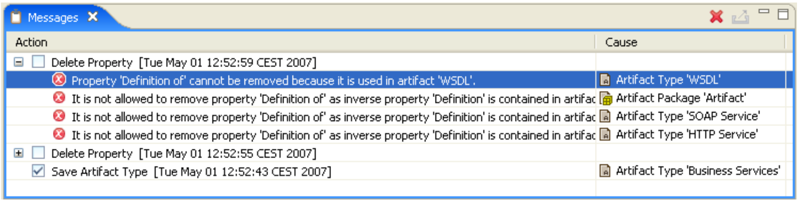
Parameter	Definition
Property Name	The name of the property.
Registry Name	The registry that the artifact type property is mapped to.

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

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 10. Messages View



Action	Cause
⊟ □ Delete Property [Tue May 01 12:52:59 CEST 2007]	
Property 'Definition of' cannot be removed because it is used in artifact 'WSDL'.	Artifact Type 'WSDL'
It is not allowed to remove property 'Definition of' as inverse property 'Definition' is contained in artifac	Artifact Package 'Artifact'
It is not allowed to remove property 'Definition of' as inverse property 'Definition' is contained in artifac	Artifact Type 'SOAP Service'
It is not allowed to remove property 'Definition of' as inverse property 'Definition' is contained in artifac	Artifact Type 'HTTP Service'
⊞ □ Delete Property [Tue May 01 12:52:55 CEST 2007]	
☑ Save Artifact Type [Tue May 01 12:52:43 CEST 2007]	Artifact Type 'Business Services'

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

project.xml Editor

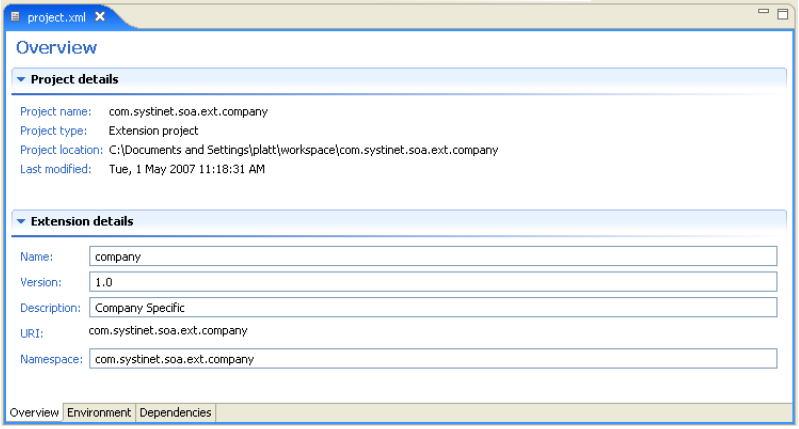
The project editor allows you to configure your extension project. It contains three tabs described in the following sections:

- Project Editor: Overview Tab on page 71
- Project Editor: Environment Tab on page 73
- Project Editor: Dependencies Tab on page 74

Project Editor: Overview Tab

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

Figure 11. Project Editor: Overview Tab



Overview

Project details

Project name: com.systinet.soa.ext.company
Project type: Extension project
Project location: C:\Documents and Settings\platt\workspace\com.systinet.soa.ext.company
Last modified: Tue, 1 May 2007 11:18:31 AM

Extension details

Name: company
Version: 1.0
Description: Company Specific
URI: com.systinet.soa.ext.company
Namespace: com.systinet.soa.ext.company

The **Overview** tab contains the following collapsible panes:

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

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

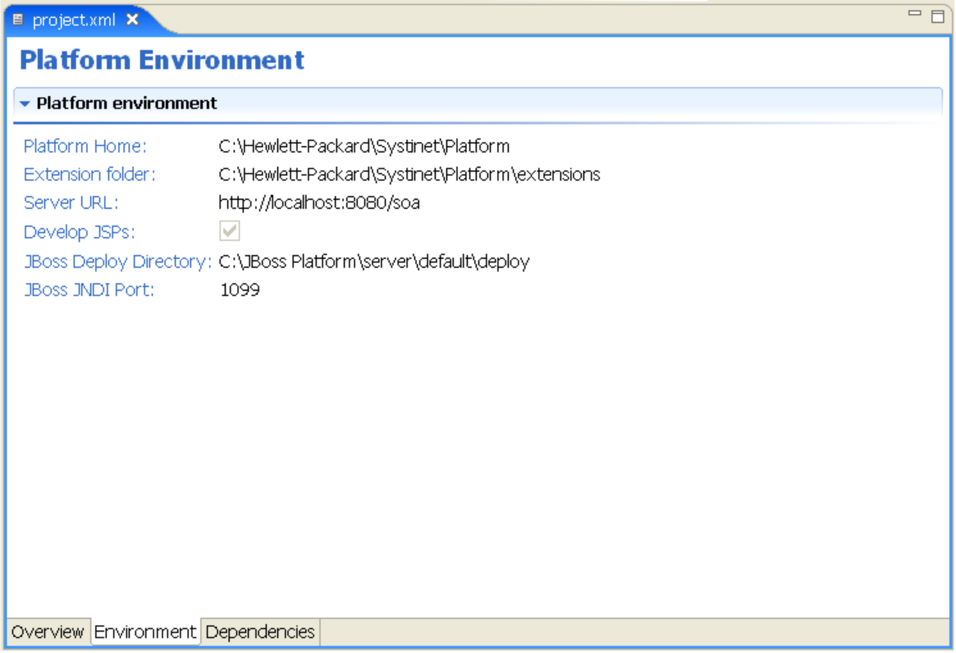
- **Extension details** enables you to configure some of the extension project parameters:

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

Project Editor: Environment Tab

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

Figure 12. Project Editor: Environment Tab



Platform Home: C:\Hewlett-Packard\Systinet\Platform
Extension folder: C:\Hewlett-Packard\Systinet\Platform\extensions
Server URL: http://localhost:8080/soa
Develop JSPs: ☑
JBoss Deploy Directory: C:\JBoss Platform\server\default\deploy
JBoss JNDI Port: 1099

The **Environment** tab contains the following parameters:

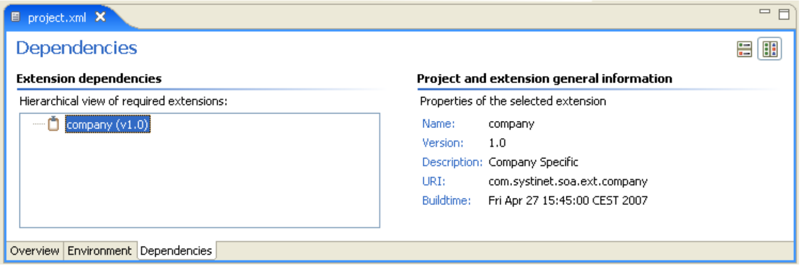
Parameter	Definition
Platform Home	Your HP SOA Systinet installation folder.
Extension Folder	The location of the extension folder in your HP SOA Systinet installation.
Server URL	The URL used to access HP SOA Systinet.
Deploy JSPs	Indicates if component and JSP development is configured for your extension project.

Parameter	Definition
JBoss Deploy Directory	Deployment directory for your JBoss application server.
JBoss JNDI Port	JNDI port configured in your JBoss application server.

Project Editor: Dependencies Tab

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

Figure 13. Project Editor:Dependencies Tab



Dependencies

Extension dependencies

Hierarchical view of required extensions:

company (v1.0)

Project and extension general information

Properties of the selected extension

Name:	company
Version:	1.0
Description:	Company Specific
URI:	com.systinet.soa.ext.company
Buildtime:	Fri Apr 27 15:45:00 CEST 2007

The **Dependencies** tab contains the following panes:

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

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

Property Editor

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

- Property Editor: Overview Tab on page 75

  The main attributes of the property.

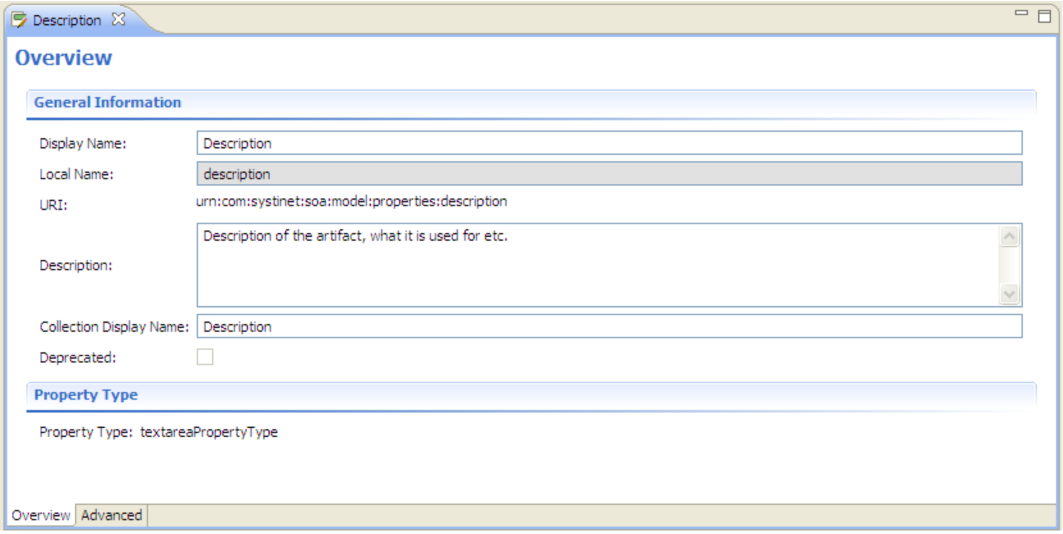
- Property Editor: Advanced Tab on page 77

  The advanced attributes of the property.

Property Editor: Overview Tab

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

Figure 14. Description Editor: Overview Tab



Overview

General Information

Display Name: Description

Local Name: description

URI: urn:com:systinet:soa:model:properties:description

Description: Description of the artifact, what it is used for etc.

Collection Display Name: Description

Deprecated: □

Property Type

Property Type: textareaPropertyType

This tab contains the following panes:

- **General**

  The following property parameters are displayed in this segment:

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

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

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

  - **Property Type**

    The property type for primitive properties.

  - **Relationship**

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

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

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

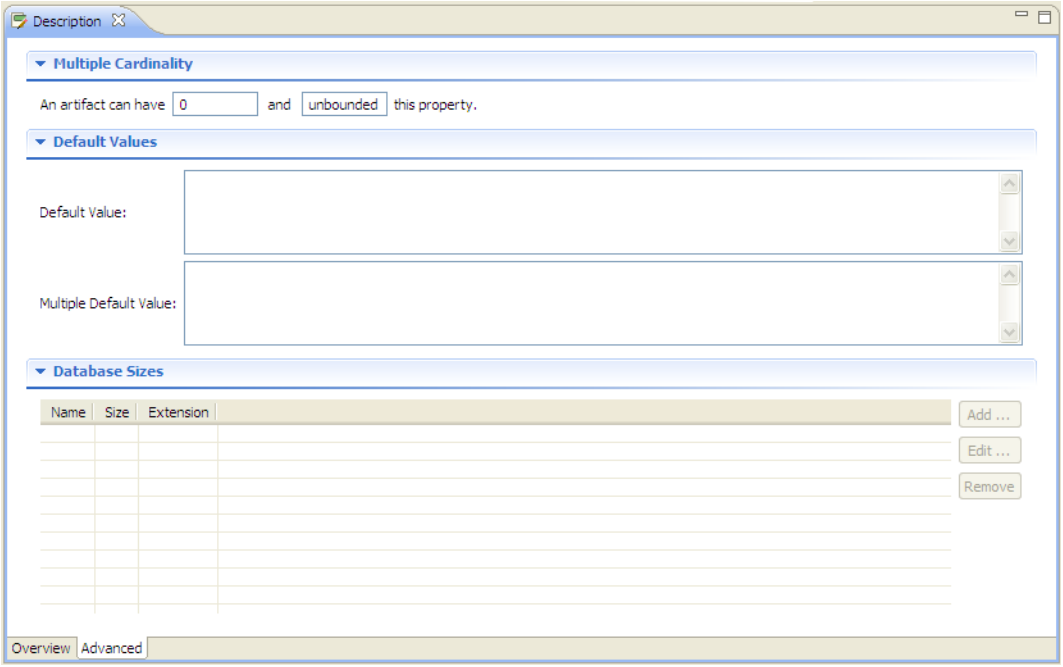
  - **Taxonomy**

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

Property Editor: Advanced Tab

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

Figure 15. Description Editor: Advanced Tab



Description

Multiple Cardinality

An artifact can have 0 and unbounded this property.

Default Values

Default Value:

Multiple Default Value:

Database Sizes

Name	Size	Extension

This tab contains the following panes:

- **Multiple Cardinality:**

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

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

Cardinality for incoming relationship properties cannot be changed.

- **Default Values:**

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

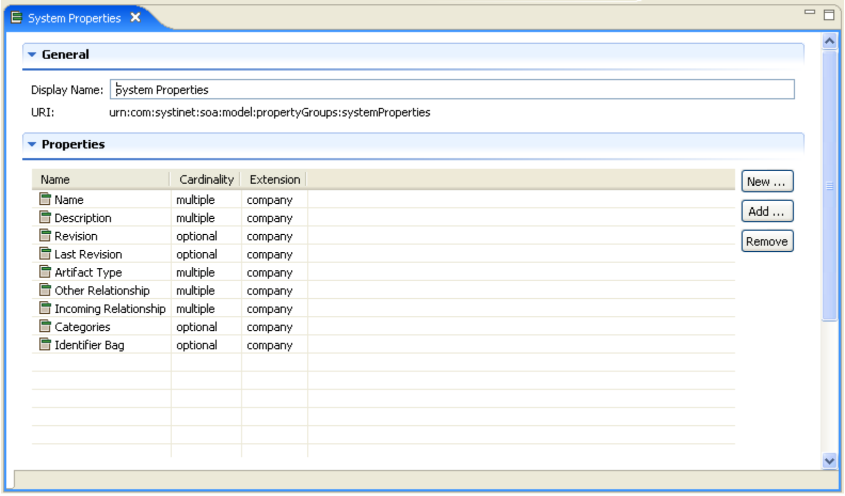
- **Database Sizes**

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

Property Group Editor

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

Figure 16. System Properties Editor



Display Name: System Properties

URI: urn:com:systinet:soa:model:propertyGroups:systemProperties

Name	Cardinality	Extension
Name	multiple	company
Description	multiple	company
Revision	optional	company
Last Revision	optional	company
Artifact Type	multiple	company
Other Relationship	multiple	company
Incoming Relationship	multiple	company
Categories	optional	company
Identifier Bag	optional	company

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 47.
  - **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:

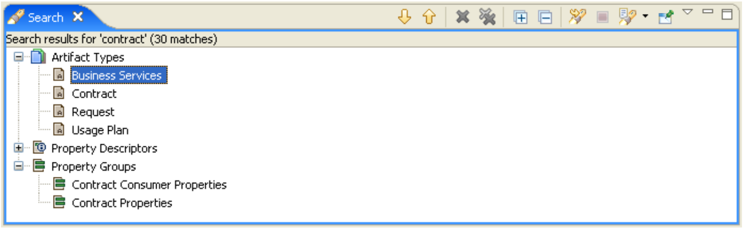
Parameter	Definition
Name	The name of the property.
Cardinality	The occurrence of the property in an artifact with options:

Parameter	Definition
	- **Optional** The property is not required to be populated. - **Required** The property must be populated. - **Multiple** The property can occur multiple times with different values.
Extension	Which extension this property is part of.

Search View

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

Figure 17. Search View



Search results for 'contract' (30 matches)

- Artifact Types
  - Business Services
  - Contract
  - Request
  - Usage Plan
- Property Descriptors
- Property Groups
  - Contract Consumer Properties
  - Contract Properties

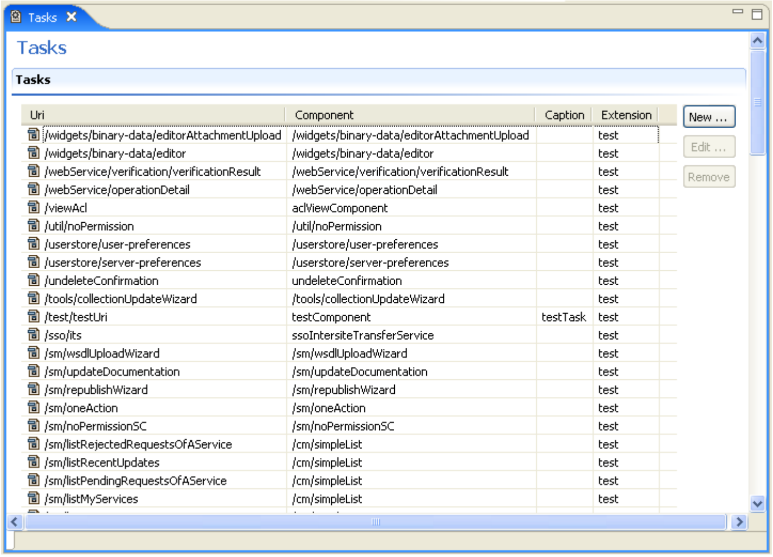
The results are shown as a tree of artifact types, properties, and property groups containing the requested item.

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

Tasks Editor

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

Figure 18. Tasks Editor



Uri	Component	Caption	Extension
/widgets/binary-data/editorAttachmentUpload	/widgets/binary-data/editorAttachmentUpload		test
/widgets/binary-data/editor	/widgets/binary-data/editor		test
/webService/verification/verificationResult	/webService/verification/verificationResult		test
/webService/operationDetail	/webService/operationDetail		test
/viewAcl	aclViewComponent		test
/util/noPermission	/util/noPermission		test
/userstore/user-preferences	/userstore/user-preferences		test
/userstore/server-preferences	/userstore/server-preferences		test
/undeleteConfirmation	undeleteConfirmation		test
/tools/collectionUpdateWizard	/tools/collectionUpdateWizard		test
/test/testUri	testComponent	testTask	test
/sso/its	ssoIntersiteTransferService		test
/sm/wsdlUploadWizard	/sm/wsdlUploadWizard		test
/sm/updateDocumentation	/sm/updateDocumentation		test
/sm/republishWizard	/sm/republishWizard		test
/sm/oneAction	/sm/oneAction		test
/sm/noPermissionSC	/sm/noPermissionSC		test
/sm/listRejectedRequestsOfAService	/cm/simpleList		test
/sm/listRecentUpdates	/cm/simpleList		test
/sm/listPendingRequestsOfAService	/cm/simpleList		test
/sm/listMyServices	/cm/simpleList		test

On the right are the following buttons:

- **New** opens the **New Task** dialog box, as described in Creating a Task on page 50.
- **Edit** opens the **Edit Task** dialog box with the same parameters described in Creating a Task on page 50.
- **Remove** deletes the selected task from your extension project.

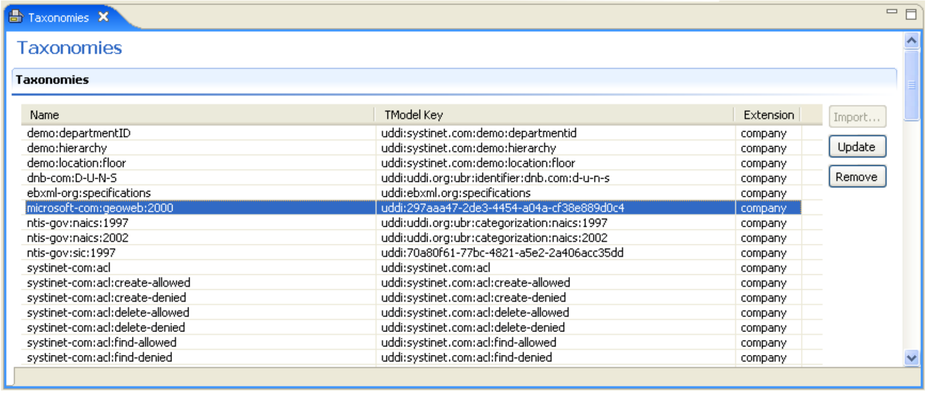
The following parameters relate to tasks:

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 19. Taxonomies Editor



Name	TModel Key	Extension
demo:departmentID	uddi:systinet.com:demo:departmentid	company
demo:hierarchy	uddi:systinet.com:demo:hierarchy	company
demo:location:floor	uddi:systinet.com:demo:location:floor	company
dnb-com:D-U-N-S	uddi:uddi.org:ubr:identifier:dnb.com:d-u-n-s	company
ebxml-org:specifications	uddi:ebxml.org:specifications	company
microsoft-com:geoweb:2000	uddi:297aaa47-2de3-4454-a04a-cf38e889d0c4	company
ntis-gov:naics:1997	uddi:uddi.org:ubr:categorization:naics:1997	company
ntis-gov:naics:2002	uddi:uddi.org:ubr:categorization:naics:2002	company
ntis-gov:sic:1997	uddi:70a80f61-77bc-4821-a5e2-2a406acc35dd	company
systinet-com:acl	uddi:systinet.com:acl	company
systinet-com:acl:create-allowed	uddi:systinet.com:acl:create-allowed	company
systinet-com:acl:create-denied	uddi:systinet.com:acl:create-denied	company
systinet-com:acl:delete-allowed	uddi:systinet.com:acl:delete-allowed	company
systinet-com:acl:delete-denied	uddi:systinet.com:acl:delete-denied	company
systinet-com:acl:find-allowed	uddi:systinet.com:acl:find-allowed	company
systinet-com:acl:find-denied	uddi:systinet.com:acl:find-denied	company

On the right are the following buttons:

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

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

- **Update** refreshes the taxonomy list with any changes from the HP SOA Systinet server.
- **Remove** deletes the selected taxonomy from the extension project.

The following parameters relate to taxonomies:

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

Parameter	**Definition**
Extension	Which extension this taxonomy is part of.

B Dialog Boxes

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

- New Artifact on page 84: Create a new artifact type or package.
- New Component on page 87: Create a new component.
- New Extension Project on page 88: Create an extension project.
- New Property on page 95: Create a new property.
- New Property Group on page 100: Create a new property group.
- New Task on page 100: Create a new task.
- Registry Mapping on page 102: Add registry mapping.
- Search on page 104: Search your extension project.

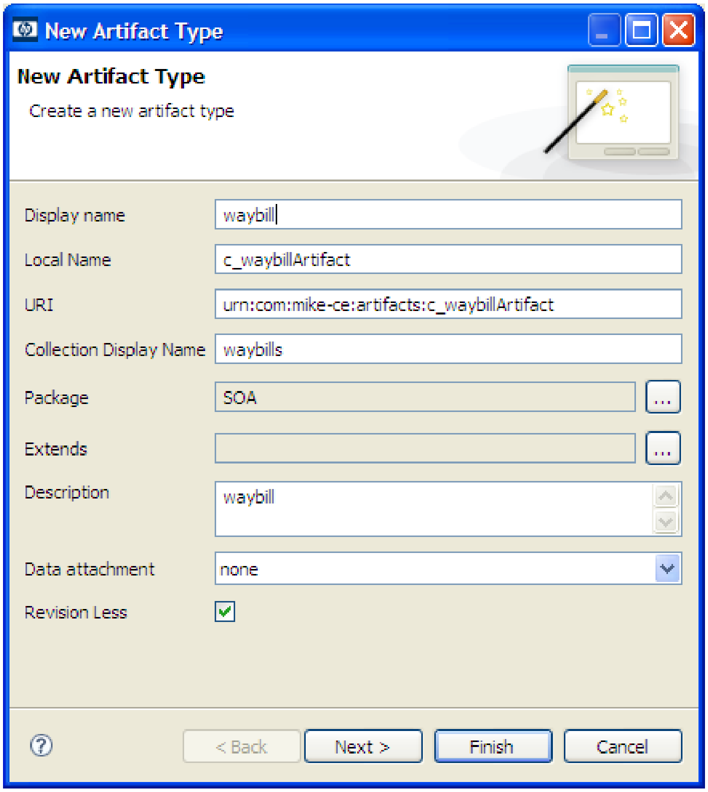
New Artifact

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

1. New Artifact: Create on page 84
2. New Artifact: Database Settings on page 86

New Artifact: Create

Enter general parameters of the new artifact type:



New Artifact Type

New Artifact Type

Create a new artifact type

Display name: waybill

Local Name: c_waybillArtifact

URI: urn:com:mike-ce:artifacts:c_waybillArtifact

Collection Display Name: waybills

Package: SOA

Extends:

Description: waybill

Data attachment: none

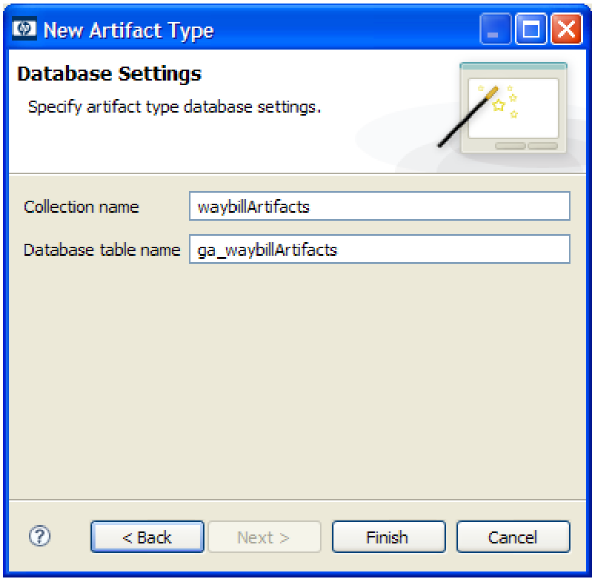
Revision Less: ☑

Parameter	Definition
Display Name	The name of the artifact as it appears in the HP 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 HP SOA Systinet UI.
Package	Click **Browse** to select a parent artifact package to which the artifact belongs. The default is SOA if the new artifact is not created from within an existing artifact package.
Extends	Click **Browse** to select an artifact type to inherit the properties from

Parameter	Definition
	► If you select **Extends**, the new artifact inherits any future changes made to the chosen parent artifact.
Description	The description of the artifact type as it appears in the HP SOA Systinet UI.
Data Attachment	If the artifact type normally has attached data content, select the content type from the drop-down list.
Revisionless	Specify the artifact to be revisionless. These type of artifacts update without creating a new revision.

New Artifact: Database Settings

Specify database settings for the new artifact type:



New Artifact Type

Database Settings

Specify artifact type database settings.

Collection name: waybillArtifacts

Database table name: ga_waybillArtifacts

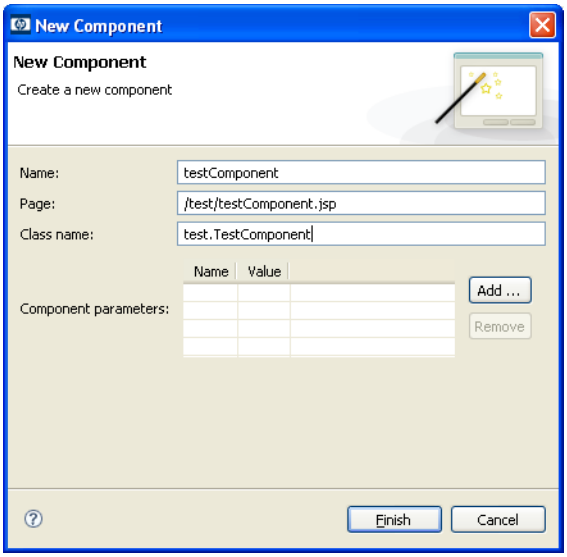
Parameter	Definition
Collection name	The name of collection where the artifact instances are stored

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

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

New Component

Enter general parameters for the new component:



New Component

New Component

Create a new component

Name: testComponent

Page: /test/testComponent.jsp

Class name: test.TestComponent

Component parameters:

Name	Value

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.

Parameter	Definition
Component parameters	Use **Add** and **Remove** to select parameters to use with the component.

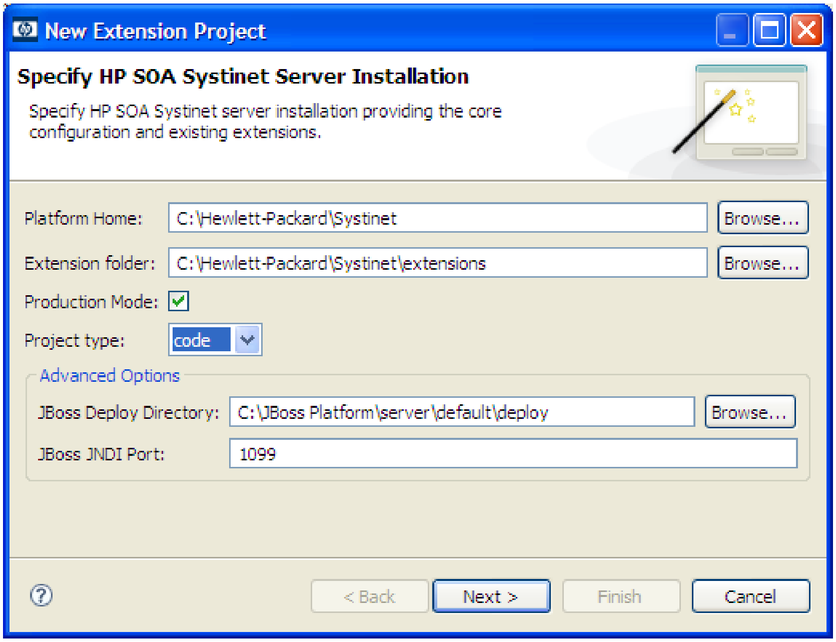
New Extension Project

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

1. New Extension Project: Specify HP SOA Systinet Server Installation on page 88
2. New Extension Project: New Server on page 90
3. New Project: Select Working Extensions on page 91
4. New Project: Create a New Extension on page 93
5. New Project: Create Dependencies on page 94

New Extension Project: Specify HP SOA Systinet Server Installation

Specify configuration details for an HP SOA Systinet server:



Specify HP SOA Systinet Server Installation

Specify HP SOA Systinet server installation providing the core configuration and existing extensions.

Platform Home: C:\Hewlett-Packard\Systinet

Extension folder: C:\Hewlett-Packard\Systinet\extensions

Production Mode: ☑

Project type: code

Advanced Options

JBoss Deploy Directory: C:\JBoss Platform\server\default\deploy

JBoss JNDI Port: 1099

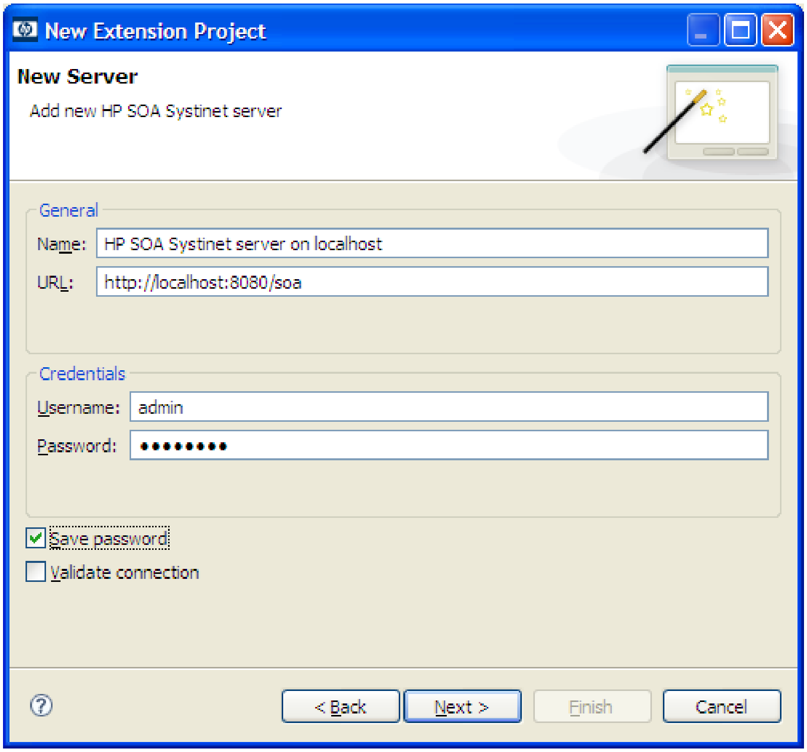
Parameter	**Definition**
Platform Home	Use **Browse** to select the HP SOA Systinet platform installation folder
Extension Folder	Use **Browse** to select the extension folder in your HP SOA Systinet installation (populated by default based on the **Platform Home** input
Production Mode	Select this check-box to create the project in production mode (selected by default). This disables customizations that would require the deletion of server data during the application of the extension to HP SOA Systinet.
Project Type	Select from the following options: • **model** for changes to the SDM model. • **code** for custom code components and UI customizations.

Parameter	Definition
	• **mixed** for both types of changes (not recommended).
JBoss Deploy Directory	Use **Browse** to select the deployment directory for your application server (usually `JBOSS_HOME\server\default\deploy`)
JBoss JNDI Port	Input the JNDI Port for your application server.

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

New Extension Project: New Server

Specify parameters for the new server:



New Server

Add new HP SOA Systinet server

General

Name: HP SOA Systinet server on localhost

URL: http://localhost:8080/soa

Credentials

Username: admin

Password: ••••••••

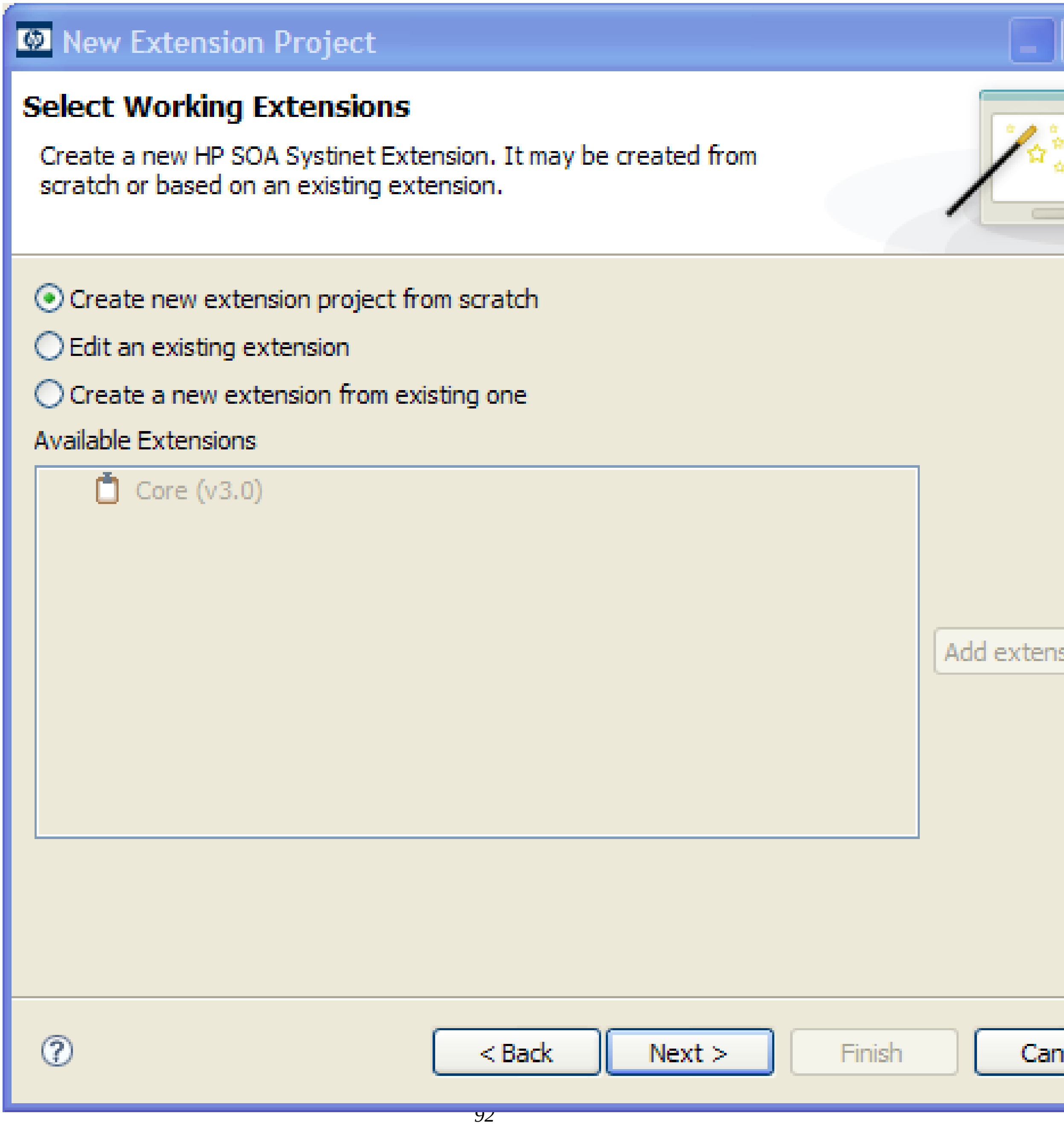
☑ Save password

□ Validate connection

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

New Project: Select Working Extensions

Choose how to create your working extensions:



New Extension Project

Select Working Extensions

Create a new HP SOA Systinet Extension. It may be created from scratch or based on an existing extension.

● Create new extension project from scratch

○ Edit an existing extension

○ Create a new extension from existing one

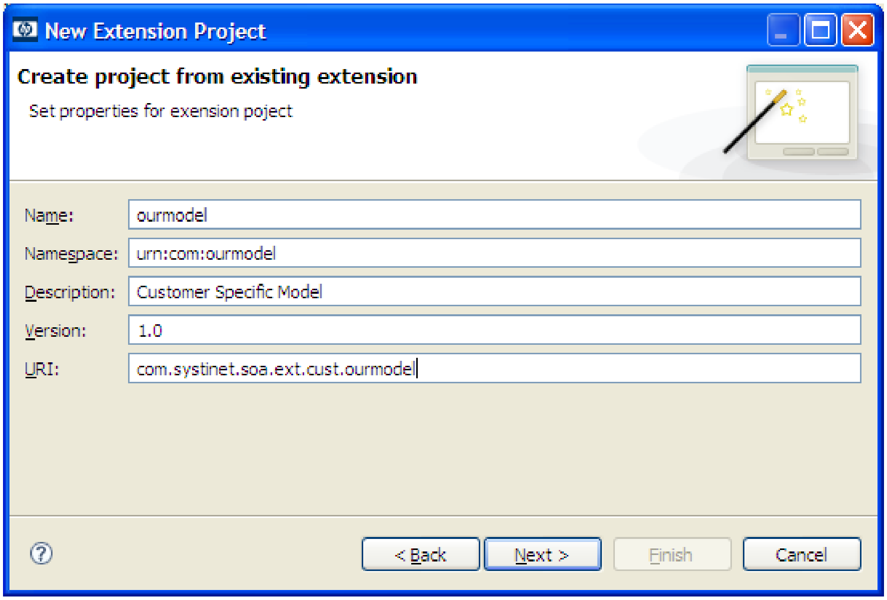
Available Extensions

Core (v3.0)

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

New Project: Create a New Extension

Enter general parameters for the new extension:



Create project from existing extension

Set properties for exension poject

Name: ourmodel

Namespace: urn:com:ourmodel

Description: Customer Specific Model

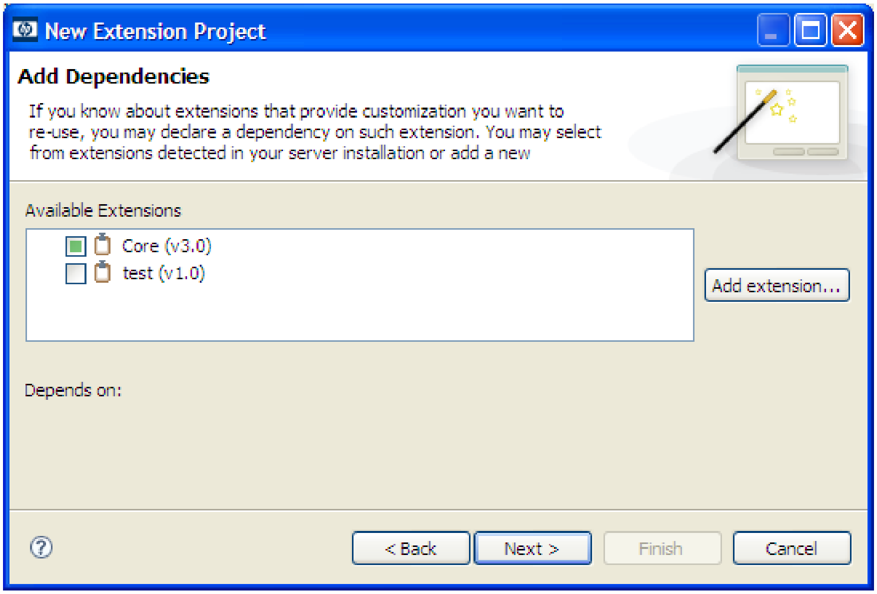
Version: 1.0

URI: com.systinet.soa.ext.cust.ourmodel

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

New Project: Create Dependencies

Select extensions to declare dependencies:



New Extension Project

Add Dependencies

If you know about extensions that provide customization you want to re-use, you may declare a dependency on such extension. You may select from extensions detected in your server installation or add a new

Available Extensions

- ☑ Core (v3.0)
- □ test (v1.0)

Depends on:

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

New Property

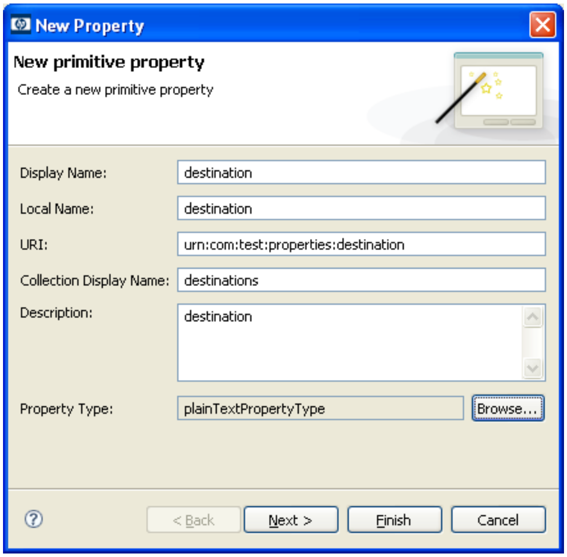
The New Property dialog contains parameters according to the property type:

- New Property: Primitive on page 95
- New Property: Relationship on page 96
- New Property: Taxonomy on page 98

For details about property types, see Properties on page 23.

New Property: Primitive

Enter general parameters for the new primitive property:



New Property

New primitive property

Create a new primitive property

Display Name: destination

Local Name: destination

URI: urn:com:test:properties:destination

Collection Display Name: destinations

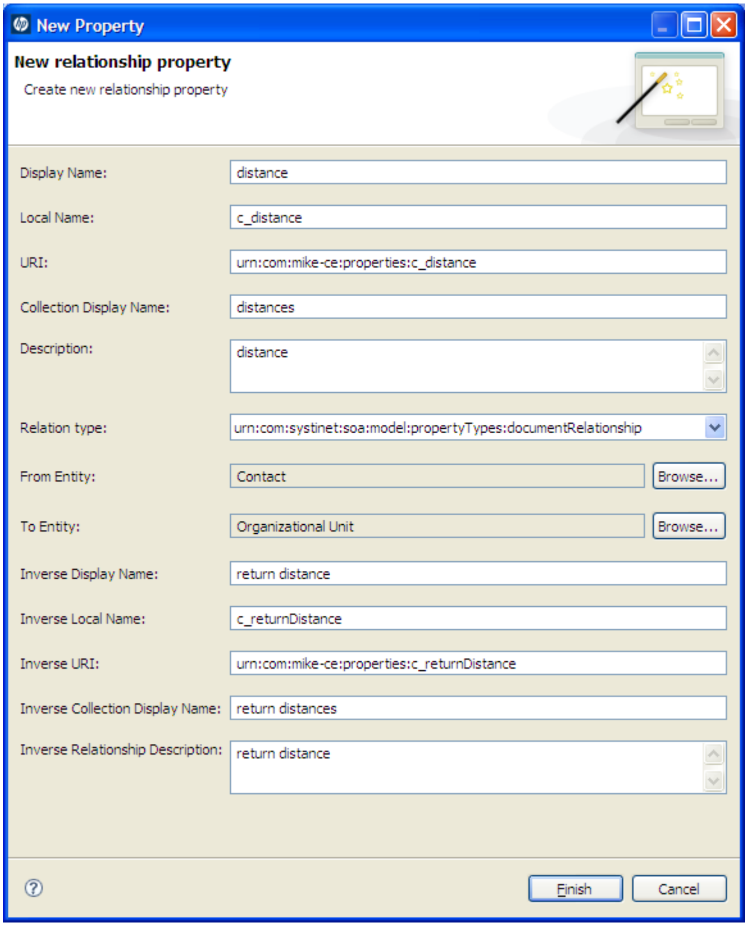
Description: destination

Property Type: plainTextPropertyType

Parameter	Definition
Display Name	The name of the property as it appears in the HP 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 HP SOA Systinet UI
Description	The description of the property as it appears in the HP SOA Systinet UI
Property Type	Use **Browse** to select a property from the available primitive property types. For details, see Properties on page 23.

New Property: Relationship

Enter general parameters for the new relationship property:



New Property

New relationship property

Create new relationship property

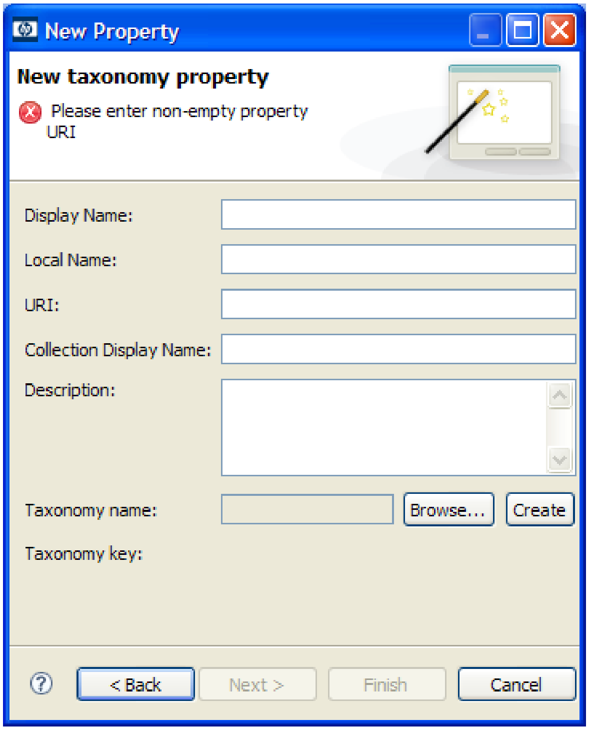
Display Name:	distance
Local Name:	c_distance
URI:	urn:com:mike-ce:properties:c_distance
Collection Display Name:	distances
Description:	distance
Relation type:	urn:com:systinet:soa:model:propertyTypes:documentRelationship
From Entity:	Contact
To Entity:	Organizational Unit
Inverse Display Name:	return distance
Inverse Local Name:	c_returnDistance
Inverse URI:	urn:com:mike-ce:properties:c_returnDistance
Inverse Collection Display Name:	return distances
Inverse Relationship Description:	return distance

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

Parameter	**Definition**
Description	The description of the property as it appears in the HP SOA Systinet UI
Relation Type	Select a type from the drop-down list
From Entity	Use **Browse** to select the source artifact type or property group of the relationship
To Entity	Use **Browse** to select the target artifact type or property group of the relationship
Inverse Display Name	The name of the inverse relationship property as it appears in the HP 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 HP SOA Systinet UI
Inverse Relationship Description	The description of the inverse relationship property as it appears in the HP SOA Systinet UI

New Property: Taxonomy

Enter general parameters for the new taxonomy property:


New Property

New taxonomy property

Please enter non-empty property URI

Display Name:

Local Name:

URI:

Collection Display Name:

Description:

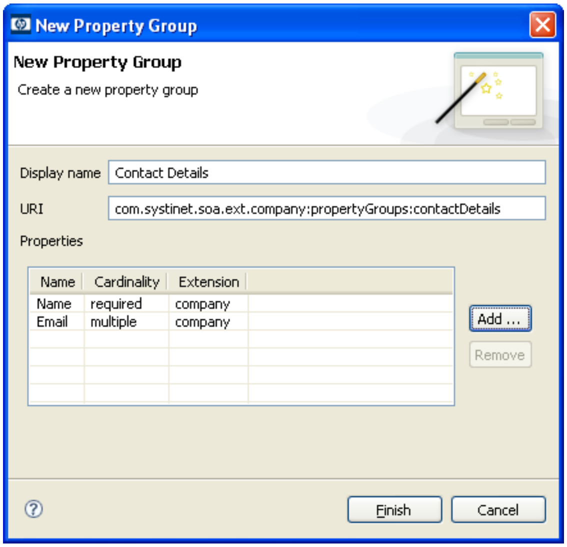
Taxonomy name:

Taxonomy key:

Parameter	Definition
Display Name	The name of the property as it appears in the HP 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 HP SOA Systinet UI
Description	The description of the property as it appears in the HP SOA Systinet UI
Taxonomy	Use **Browse** to select a taxonomy from the available taxonomies with an option to import taxonomies available in HP SOA Systinet that are not in your extension.

New Property Group

Enter general parameters for the new property group:



Display name: Contact Details

URI: com.systinet.soa.ext.company:propertyGroups:contactDetails

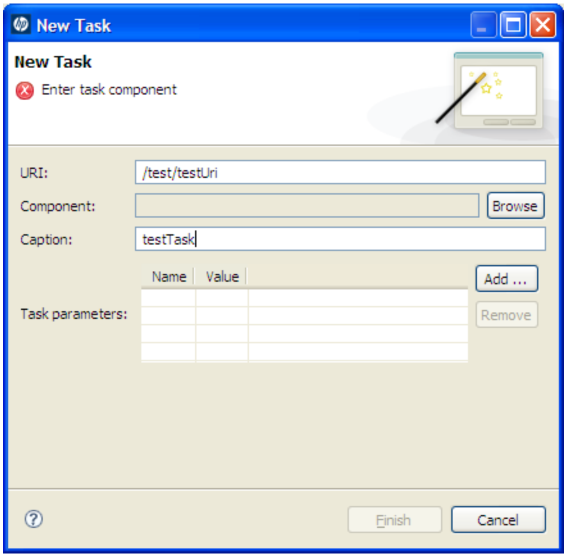
Properties

Name	Cardinality	Extension
Name	required	company
Email	multiple	company

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

New Task

Enter general parameters for the new task:



New Task

URI: /test/testUri

Component:

Caption: testTask

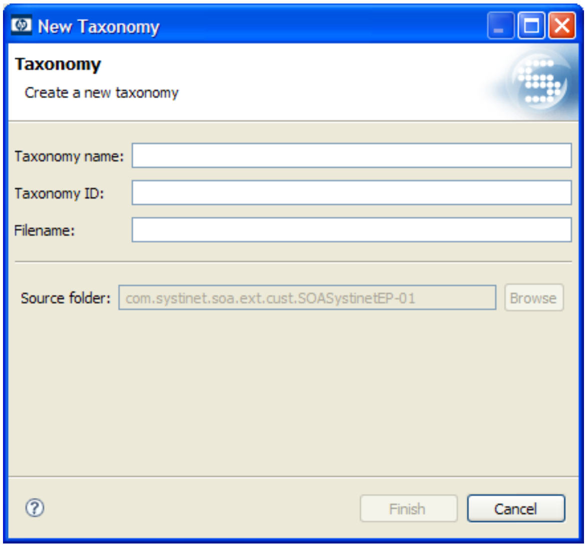
Task parameters:

Name	Value

Parameter	Definition
Uri	The identifier for the task
Component	Use **Browse** to select the component to associate with the task
Caption	A visible name for the task.
Task parameters	Use **Add** and **Remove** to select parameters to use with the task

New Taxonomy

Enter general parameters for the new taxonomy:



New Taxonomy

Taxonomy

Create a new taxonomy

Source folder: com.systinet.soa.ext.cust.SOASystinetEP-01

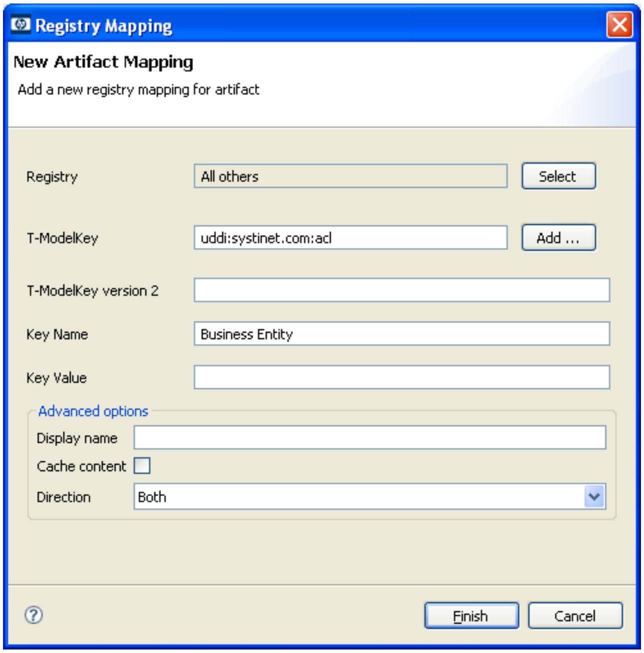
Parameter	Definition
Taxonomy Name	The name of the new taxonomy as it will appear in the SOA Systinet UI.
Taxonomy ID	The system identifier of the new taxonomy.
Filename	The name of the taxonomy as it appears in the repository.

Registry Mapping

- Registry Mapping: New Artifact Mapping on page 102
- Registry Mapping: New Property Mapping on page 104

Registry Mapping: New Artifact Mapping

Enter parameters for the new registry mapping of the artifact:



Registry Mapping

New Artifact Mapping

Add a new registry mapping for artifact

Registry: All others

T-ModelKey: uddi:systinet.com:acl

T-ModelKey version 2:

Key Name: Business Entity

Key Value:

Advanced options

Display name:

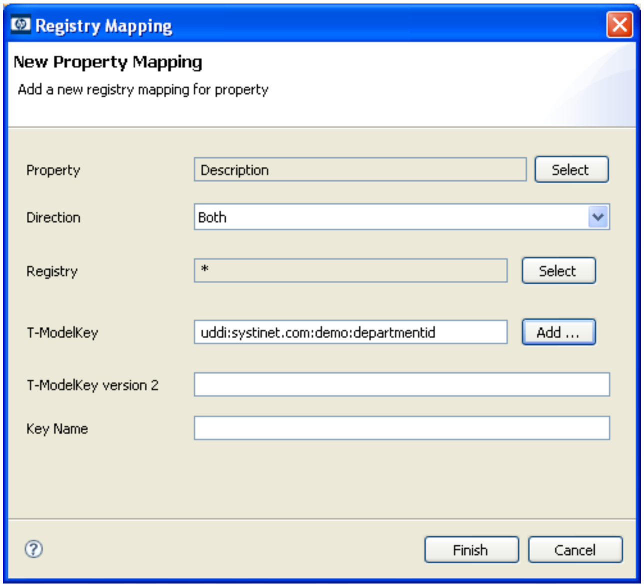
Cache content □

Direction: Both

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
Display Name	The name of the mapping as it appears in the HP SOA Systinet UI
Cache content	If selected, the content of documents referenced from the UDDI entity are cached in HP SOA Systinet
Direction	Select the direction of the mapping from the drop-down list

Registry Mapping: New Property Mapping

Enter parameters for the new registry mapping of the property:



New Property Mapping

Add a new registry mapping for property

Property: Description

Direction: Both

Registry: *

T-ModelKey: uddi:systinet.com:demo:departmentid

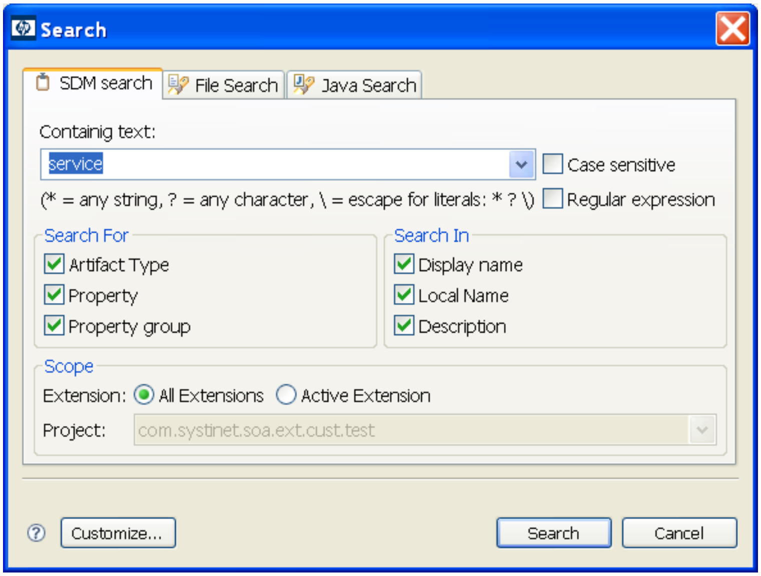
T-ModelKey version 2:

Key Name:

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

Search

Enter parameters to search your extension project:



Search

SDM search | File Search | Java Search

Containig text:

service

☐ Case sensitive

(* = any string, ? = any character, \ = escape for literals: * ? \)

☐ Regular expression

Search For

☑ Artifact Type

☑ Property

☑ Property group

Search In

☑ Display name

☑ Local Name

☑ Description

Scope

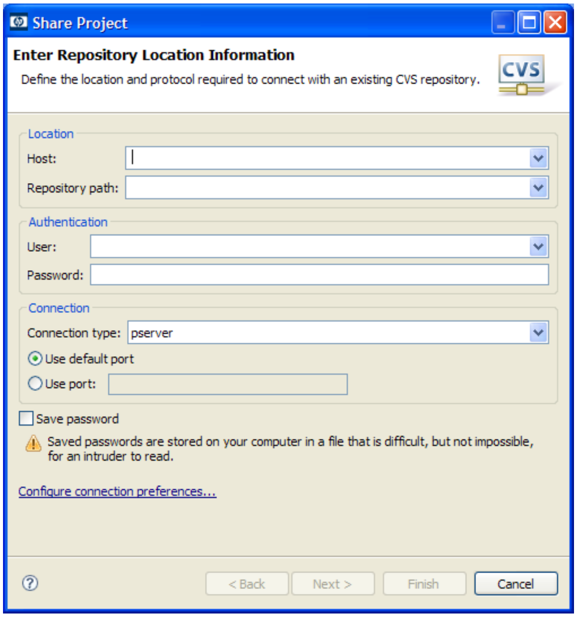
Extension: ● All Extensions ○ Active Extension

Project: com.systinet.soa.ext.cust.test

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

Share Project

Define parameters for the CVS repository:



Enter Repository Location Information

Define the location and protocol required to connect with an existing CVS repository.

Connection type: pserver

● Use default port

○ Use port:

□ Save password

Saved passwords are stored on your computer in a file that is difficult, but not impossible, for an intruder to read.

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

C Localname Naming Rules

The naming strategy used to generate the names of database tables and columns can only process a specific subset of XML identifiers (element and attribute names). This is because the existing database engines have limits for the length of tablename/columnname identifiers. XML identifiers are generated from the SDM configuration in a straightforward way. This section omits the XML layer for simplicity and summarizes the rules at the SDM level which is more user-friendly.

When creating an artifact type or property, for the localname, do the following:

- Use Latin characters where possible (A-Z, a-z)
- Use short identifiers where possible to improve the readabililty of the DB schema. For example, `stSchtronXslt` is a better localname than `searchAndTransformSchematronXsltUrlLink`.
- You can safely use underscores in localnames.

Do not use numbers in localnames where possible. They should not be used in artifact localnames at all and to avoid confusion, do not use numbers in property localnames when the property is in multiple cardinality.

Using numbers in localnames increases the chances that extension application fails.

There are no restrictions on property group localnames as they are not represented in the database schema.

It is difficult to generate separate validation rules for properties and artifacts because the generated DB identifiers are often influenced by a combination of both.

D Eclipse Plug-in Requirements

Ensure that the following prerequisite plug-ins are added to your Eclipse development platform based on the required Workbench components:

- DTP SDK 1.5

  Required for Customization Editor and Report Editor.

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

- EMF SDO Runtime 2.3.0

  Required for Customization Editor and Report Editor.

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

- GEF Runtime 3.3

  Required for Customization Editor and Report Editor.

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

- WTP 2.0

  Required for Customization Editor and Report Editor.

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

- XSD Runtime 2.3

  Required for Customization Editor and Report Editor

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

- BIRT 2.2.0

Required for Report Editor.

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